

MASTERARBEIT

ERFOLGSFAKTOREN FÜR DIE GENAUIGKEIT VON SCHÄTZUNGEN IN AGILEN SOFTWARE-ENTWICKLUNGSPROJEKTEN

Schätzungen als Basis für erfolgreiche Termin- und Kostenplanung

ausgeführt an der



am Studiengang
Software Engineering Leadership

Von: Thomas Marschall
Personenkennzeichen: 1440030006

Bad Tatzmannsdorf, am 21.04.2017

A handwritten signature in blue ink, appearing to read 'Th. Marschall', written over a horizontal dotted line.

Unterschrift

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.



.....
Unterschrift

DANKSAGUNG

Mein Dank gilt allen voran meiner Gattin Sabine, die mich nicht nur während dem Verfassen dieser Arbeit jederzeit unterstützt hat und mir viel Verständnis entgegengebracht hat, wenn ich meine Zeit am Computer statt mit meiner Familie verbracht habe, sondern die mich auch ermuntert hat, das Studium an der FH Campus 02 zu beginnen.

Des Weiteren danke ich Jürgen, dem Geschäftsführer meines Arbeitgebers und allen Kollegen, die mit ihren Inputs in den vielfältigen Diskussionen über Vorgehensmodelle, Planungen und Schätzungen für Softwareprojekten meine Arbeit inspiriert und unterstützt haben.

Nicht zuletzt gilt mein Dank dem Betreuer meiner Arbeit, Johannes Pusterhofer, sowie den Lektoren und den Kommilitonen des Masterstudiengangs „Software Engineering Leadership“, die mir geholfen haben, meinen beruflichen und privaten Horizont zu erweitern.

KURZFASSUNG

In Software-Entwicklungsprojekten ist, wie auch in anderen Projekten, die Qualität der Planung ein wesentlicher Faktor für den Projekterfolg. Um eine zuverlässige Termin- und Kostenplanung zu erstellen, ist es notwendig, den Aufwand für die Projektumsetzung mit hinreichender Genauigkeit zu kennen. Daher ist es wichtig, Faktoren zu identifizieren, welche die Genauigkeit der Schätzung in Software-Entwicklungsprojekten positiv beeinflussen.

In der Software-Entwicklung wird zwischen traditionellen (schwergewichtigen) und agilen (leichtgewichtigen) Vorgehensmodellen unterschieden. Wichtige Vorgehensmodelle werden in ihren Grundzügen beschrieben. Bei den traditionellen Vorgehensmodellen sind dies das Wasserfallmodell, der Unified Process und das Spiral-Modell. Bei den agilen Vorgehensmodellen, auf die sich die Betrachtungen im weiteren Verlauf dieser Arbeit konzentrieren, ist Scrum das am häufigsten eingesetzte, Extreme Programming XP (oder ein abgeleitetes Vorgehensmodell) das am zweithäufigsten angewendete.

Eigenschaften der Vorgehensmodelle und Unterschiede in den Zugängen zu einigen Aspekten des Vorgehens im Projekt werden beschrieben.

Neben dem Vorgehen im Projektablauf sind auch die Methoden für das Erstellen von Schätzungen unterschiedlich. Diese Methoden werden für traditionelle und agile Projekte betrachtet. In agilen Projekten wird häufig eine Trennung der Schätzung der Größe eines Entwicklungsprojekts und dem Aufwand vorgenommen. Verbreitete Methoden für dieses Vorgehen ist das Schätzen mit Story Points oder Idealen Tagen. Beide Methoden haben aber auch Nachteile, weshalb eine Alternative, das Schätzen mit tatsächlicher Zeit, betrachtet wird.

Sodann werden Einflussfaktoren auf die Genauigkeit von Schätzungen – sowohl positive als auch negative – beschrieben.

Im zweiten Teil der Arbeit erfolgt eine Prüfung der Einflussfaktoren auf Schätzungen in einem mittleren IT-Unternehmen daraufhin, ob diese eine signifikante Auswirkung auf die Qualität der Schätzung haben. Diese Prüfung erfolgt durch Interviews mit sechs Experten, einer Befragung aller Mitarbeiter in der Software-Entwicklung mittels Fragebogen und durch das Auswerten von historischen Daten. Diese Daten sind wesentliche Projekt-Kenngrößen wie Gesamtumfang in Stunden, Anzahl der im Projekt tätigen Mitarbeiter, sowie eine Gegenüberstellung des geschätzten Aufwands pro Backlog-Item und tatsächlichem Aufwand.

ABSTRACT

In software development projects, as well as in other projects, the quality of planning is an essential factor for project success. To create a reliable scheduling and cost planning, it is necessary to know the effort and cost of project implementation with sufficient accuracy. Therefore, it is important to identify factors that influence the accuracy of the estimation in software development projects positively.

In software development there is a distinction between traditional (heavyweight) and agile (lightweight) process models. Important process models are described in outline. In traditional process models that are the waterfall model, the Unified Process and the spiral model. Within agile process models, which are in the focus of the remainder of this work, Scrum is the most commonly used, Extreme Programming XP (or a derived process model), the second most commonly applied.

Features of the process models and differences in the approaches to some aspects of the procedure in the project are described.

Not only the proceedings in the project process models are different, also the methods for creating estimates are. These methods are considered for traditional and agile projects. In agile projects, it is common use to separate the estimate of the size of a development project and the effort. Common methods for this approach are estimating with story points or ideal days. Both methods have some disadvantages, which is why an alternative - estimating with actual time – is looked at in some detail.

Then, factors affecting the accuracy of estimates - both positive and negative – are described.

In the second part, we will examine the factors influencing estimates in a middle-sized IT-company. We will investigate whether these have a significant impact on the quality of the estimate. This examination is carried out through interviews with six experts, a survey of all employees in software development by means of a questionnaire, and by the evaluation of historical data. These data are essential project parameters, such as the total number of hours, the number of employees in the project, as well as a comparison of the estimated and actual effort per backlog item.

GLEICHHEITSGRUNDSATZ

Aus Gründen der Lesbarkeit wurde in dieser Arbeit darauf verzichtet, geschlechtsspezifische Formulierungen zu verwenden. Jedoch möchte ich ausdrücklich festhalten, dass die bei Personen verwendeten maskulinen Formen für beide Geschlechter zu verstehen sind.

INHALTSVERZEICHNIS

| | | |
|----------|--|-----------|
| 1 | EINLEITUNG | 1 |
| 1.1 | Ausgangssituation | 1 |
| 1.2 | Problemstellung | 1 |
| 1.3 | Forschungsfrage, Arbeitshypothese und Nullhypothese | 2 |
| 1.4 | Ziele und Nichtziele | 2 |
| 1.5 | Methodische Vorgangsweise..... | 2 |
| 1.6 | Aufbau der Arbeit..... | 4 |
| 2 | PLANUNG IN SOFTWAREPROJEKTEN..... | 6 |
| 2.1 | Unsicherheit von Schätzungen im Projektverlauf | 7 |
| 3 | VORGEHENSMODELLE FÜR SOFTWAREENTWICKLUNG..... | 10 |
| 3.1 | Traditionelle und agile Vorgehensmodelle | 10 |
| 3.2 | Traditionelle Vorgehensmodelle | 10 |
| 3.2.1 | Wasserfallmodell | 10 |
| 3.2.2 | Unified Process..... | 12 |
| 3.2.3 | Spiral Modell | 13 |
| 3.2.4 | Eigenschaften von traditionellen Vorgehensmodellen | 13 |
| 3.3 | Agile Vorgehensmodelle..... | 13 |
| 3.3.1 | Scrum | 14 |
| 3.3.2 | Extreme Programming XP | 15 |
| 3.3.3 | Eigenschaften von agilen Vorgehensmodellen | 15 |
| 4 | AUFWANDSSCHÄTZUNG IN TRADITIONELLEN SOFTWAREPROJEKTEN | 17 |
| 4.1 | Ableiten von Schätzungen aus Zählungen | 17 |
| 4.1.1 | Zusammenfassung und Kritik | 19 |
| 4.2 | Ableitung von Schätzungen aus Function Points | 19 |
| 4.2.1 | Anforderungen, die den Entwicklungsaufwand beeinflussen | 20 |
| 4.2.2 | Project Delivery Rate | 21 |
| 4.2.3 | Zusammenfassung und Kritik | 22 |

| | | |
|----------|---|-----------|
| 5 | AUFWANDSSCHÄTZUNG IN AGILEN SOFTWAREPROJEKTEN | 24 |
| 5.1 | Schätzung mit Story Points | 24 |
| 5.1.1 | Welche Größe wird mit Story Points gemessen? | 25 |
| 5.1.2 | Vorteile der Schätzung mit Story Points | 26 |
| 5.1.3 | Von Story Points zur Zeitdauer | 27 |
| 5.2 | Schätzung mit idealen Tagen | 28 |
| 5.2.1 | Vorteile der Schätzung mit idealen Tagen | 29 |
| 5.3 | Nachteile der vorangegangenen Methoden und Schätzen mit tatsächlicher Zeit | 29 |
| 5.4 | Zusammenfassung | 31 |
| 6 | EINFLUSSFAKTOREN AUF DIE GENAUIGKEIT VON SCHÄTZUNGEN..... | 32 |
| 6.1 | Schätzungen mit Story Points und Expertenschätzungen in agilen Projekten | 32 |
| 6.2 | Faktoren, die die Genauigkeit negativ beeinflussen..... | 33 |
| 6.2.1 | Änderungen von Anforderungen | 33 |
| 6.2.2 | Weitere negative Einflussfaktoren | 34 |
| 6.3 | Faktoren, die die Genauigkeit positiv beeinflussen | 34 |
| 6.3.1 | Methoden zur Reduzierung von situationsbedingten und persönlichen Verzerrungen..... | 34 |
| 6.3.2 | Methoden, um den Schätzprozess zu unterstützen | 37 |
| 6.3.3 | Feedback für die Schätzung und Trainingsmöglichkeiten zur Verfügung stellen | 39 |
| 6.3.4 | Berücksichtigung der (Un-)Genauigkeit der Schätzung | 40 |
| 6.3.5 | Planning Poker | 41 |
| 6.3.6 | Stufenweise Planung | 42 |
| 7 | ZUSAMMENFASSUNG TEIL 1 | 44 |
| 8 | FALLSTUDIE | 46 |
| 8.1 | Experteninterviews | 47 |
| 8.1.1 | Umfeld der Interviews | 48 |
| 8.1.2 | Die Interviewpartner | 49 |
| 8.1.3 | Gesprächsleitfaden..... | 49 |
| 8.1.4 | Die Transkriptionsregeln..... | 50 |
| 8.1.5 | Auswertung der Experteninterviews | 51 |
| 8.2 | Befragung der Mitarbeiter der Abteilung Technik..... | 61 |
| 8.2.1 | Erfahrung in Projekten, Methodik für die Schätzung und Evaluierung..... | 61 |

| | | |
|----------|--|------------|
| 8.2.2 | Meinung der Befragten zu positiven Einflussfaktoren | 65 |
| 8.2.3 | Meinung der Befragten zu negativen Einflussfaktoren..... | 72 |
| 8.3 | Auswertung der historischen Daten von mehreren Teams im Unternehmen rmDATA | 73 |
| 8.3.1 | Eingesetzte Metriken für die Schätzung der Größe von Backlog-Items..... | 75 |
| 8.3.2 | Auswirkungen von Erfolgsfaktoren in den untersuchten Projekten..... | 76 |
| 9 | FAZIT UND AUSBLICK | 83 |
| 9.1 | Fazit | 84 |
| 9.2 | Ausblick | 85 |
| | ANHANG A - PROJEKTSTECKBRIEFE | 86 |
| | ANHANG B - TRANSKRIPTIONEN DER EXPERTEN-INTERVIEWS..... | 104 |
| | ANHANG C - DAS AGILE MANIFEST | 140 |
| | ABKÜRZUNGSVERZEICHNIS..... | 141 |
| | GLOSSAR | 142 |
| | ABBILDUNGSVERZEICHNIS | 143 |
| | TABELLENVERZEICHNIS | 145 |
| | LITERATURVERZEICHNIS..... | 146 |

1 EINLEITUNG

1.1 Ausgangssituation

Die Aufwandsschätzung in agilen Entwicklungsprojekten, insbesondere wenn diese mit Scrum umgesetzt werden, erfolgt häufig mit Hilfe von Story Points. Story Points sind eine relative Maßeinheit für die Größe eines Backlog Items. Somit werden Stories untereinander vergleichbar – eine Story mit zwei Story Points ist doppelt so groß wie eine Story mit einem Story Point und etwa halb so groß wie eine Story mit fünf Story Points.

Story Points von verschiedenen Entwicklungsteams sind nicht vergleichbar, da jedes Team andere Maßstäbe für die Vergabe von Story Points verwendet.

Um auf Basis von Story Points Vorhersagen über den Zeitpunkt zu treffen, wann ein Software-Entwicklungsprojekt fertiggestellt sein wird, wird üblicher Weise die Velocity des Teams herangezogen. Die Velocity gibt an, wie viele Story-Points ein Team innerhalb eines Sprints durchschnittlich umsetzen kann. Durch Division des Gesamtaufwandes für ein Projekt durch die Velocity ergibt sich die Anzahl der Sprints, die benötigt werden, um das Projekt fertig zu stellen.

Des Weiteren steht in Software-Entwicklungsprojekten, die mit agilen Vorgehensmodellen umgesetzt werden, zu Beginn des Projekts üblicher Weise noch keine vollständige und endgültige Beschreibung der Anforderungen zur Verfügung. Anforderungen werden erst im Laufe des Projekts detailliert und auch angepasst.

Somit unterscheidet sich der Ansatz für die Termin- und Kostenplanung in agilen Projekten wesentlich von dem Ansatz in traditionellen Projekten.

In Projekten, die mit traditionellen Vorgehensmodellen umgesetzt werden, werden zu Beginn des Projekts alle Anforderungen möglichst genau beschrieben, der Aufwand in Stunden (Tagen, Wochen oder Monaten) geschätzt und daraus eine Meilenstein- und Terminplanung abgeleitet.

1.2 Problemstellung

Methoden, die dabei helfen, die Genauigkeit von Schätzung in traditionellen Projekten genauer zu machen, können nicht ohne weiteres auf agile Projekte umgelegt werden¹.

Dennoch ist es – vor allem für das Management – unabhängig vom verwendeten Vorgehensmodell für die Umsetzung von Software-Entwicklungsprojekten wichtig, den Termin und die Kosten für die Fertigstellung eines Release und den Inhalt des Releases zu kennen.

¹ Siehe auch (Choudhari & Suman, 2012)

Aussagen darüber können aber nur dann getroffen werden, wenn der Aufwand (in Zeit) für die Fertigstellung des Projekts bekannt ist. Diese Aussagen können in Projekten, die mit agilen Methoden umgesetzt werden, oft nicht mit der gewünschten Genauigkeit getroffen werden. Um diese Problematik zu minimieren, sollen Methoden gefunden werden, um die Genauigkeit der Aufwandsschätzung in agilen Software-Entwicklungsprojekten zu erhöhen.

1.3 Forschungsfrage, Arbeitshypothese und Nullhypothese

Forschungsfrage:

Gibt es in agilen, iterativen Software-Entwicklungsprojekten Faktoren, welche die Aufwandsschätzung positiv beeinflussen, sodass eine zuverlässige Aussage über den Aufwand und die Umsetzungsdauer des Vorhabens getroffen werden kann?

Arbeitshypothese:

Je bewusster und umfangreicher positive Erfolgsfaktoren für die Schätzgenauigkeit in Software-Entwicklungsprojekten eingesetzt werden, desto höher ist die Genauigkeit der Schätzungen im Projekt.

Nullhypothese:

Der gezielte und bewusste Einsatz von positiven Erfolgsfaktoren beeinflusst die Schätzgenauigkeit in Software-Entwicklungsprojekten nur marginal.

1.4 Ziele und Nichtziele

In der vorliegenden Arbeit soll untersucht werden, ob es Methoden und Faktoren gibt, welche die Genauigkeit von Schätzungen in agilen Projekten unterstützen, wie diese angewendet werden, welchen Aufwand sie verursachen und wie stark die jeweiligen Auswirkungen sind. Die in Frage kommenden Faktoren werden durch eine Literaturrecherche ermittelt.

Die Auswirkung dieser Faktoren soll durch Befragung von Projektmanagern mittels Experteninterviews, von Mitarbeitern in der Software-Entwicklung mittels Online-Befragung und die Auswertung von historischen Daten eines mittleren IT-Unternehmens mit hinreichender Signifikanz belegt werden.

Es erfolgt keine Auswertung von Daten von anderen Unternehmen, auch werden keine Daten aus Projektdatenbanken für die Verifizierung der Arbeitshypothese herangezogen.

1.5 Methodische Vorgangsweise

Ausgehend von der Arbeitshypothese, dass es Faktoren gibt, welche die Genauigkeit der Schätzung in agilen Softwareprojekten positiv beeinflussen, wird mittels eines deduktiven Ansatzes versucht, die Hypothese durch eine Literaturrecherche, Befragung der Experten und Mitarbeiter und mit dem Datenmaterial der Firma rmDATA zu verifizieren.

Im theoretischen Teil der Arbeit werden die Faktoren, die für die Beeinflussung der Genauigkeit der Schätzungen in Frage kommen, durch Literaturrecherche ermittelt, im praktischen Teil der Einfluss dieser Faktoren auf die Schätzungen bei der Firma rmDATA geprüft.

Bei der Firma rmDATA existieren für Projekte der vergangenen Jahre einerseits die Backlogs und die Projektdefinitionen mit den Rahmenbedingungen und Annahmen über den Gesamtaufwand sowie andererseits detaillierte Zeitaufzeichnungen über den tatsächlichen Umsetzungsaufwand. Für die Arbeitspakete, für die ausreichend Daten vorliegen, wird die Schätzung (in Story Points oder idealer Zeit) dem tatsächlichen Aufwand gegenübergestellt. Innerhalb der einzelnen Projekte wird der tatsächliche durchschnittliche Zeitaufwand für die Umsetzung eines Story Points sowie die Varianz für diese Wert ermittelt. Dadurch ergibt sich ein Bild, in welchem Maß die Schätzungen des Teams mit dem tatsächlichen Aufwand korrelieren.

Durch Interviews mit den Projektmanagern wird die Erfahrung dieser Experten hinsichtlich der Wirksamkeit von Erfolgsfaktoren erhoben. Die Auswertung der Interviews erfolgt mittels induktiver Kategorienbildung. Erfolgsfaktoren, die auf Basis der Literaturrecherche ermittelt wurden und die aus der Erfahrung der Projektmanager positive Auswirkung auf die Genauigkeit von Schätzungen in deren Projekten haben, können als wirksam eingestuft werden. Die Erfahrungen der Projektmanager werden durch eine Online-Befragung aller Mitarbeiter in der Software-Entwicklung des untersuchten Unternehmens ergänzt.

Abschließend erfolgt, soweit das auf Basis der verfügbaren Daten möglich ist, eine Analyse der Projektrahmenbedingungen und der Historie der User-Stories, um zu ermitteln, ob und welche der möglichen Einflussfaktoren auf die Schätzung der jeweiligen Story bzw. das jeweilige Projekt gewirkt haben.

Durch den Vergleich der relativen Abweichung zwischen den Stories und den gefundenen Einflussfaktoren wird ermittelt, ob die Einflussfaktoren die angenommene positive Wirkung auf die Genauigkeit der Schätzung haben. Wenn also die Schätzungen für Stories oder Projekte, auf die ein oder mehrere der gefundenen Erfolgsfaktoren zutreffen, signifikant genauer ist als die Schätzung für Stories oder Projekte, auf die keine der Erfolgsfaktoren zutreffen, kann die Nullhypothese verworfen werden.

Als Maß für die Genauigkeit der Schätzungen dienen die nach Größe der Abweichung kategorisierten Schätzungen, der Mittelwert der Abweichung der Schätzungen und die Standardabweichung derselben sowie der Medianwert der Abweichungen.

Je höher die Anzahl der Schätzungen mit geringen Abweichungen in einem Projekt ist, desto genauer sind die Schätzungen der einzelnen Stories.

Der Mittelwert der Abweichungen sollte, ebenso wie der Median, bei 0% Abweichung liegen, was als Indikator dafür dienen würde, dass die Schätzung des Gesamtaufwands für alle untersuchten Stories genau war.

Ebenso ist die Standardabweichung der Abweichung der Schätzungen vom tatsächlichen Aufwand ein Indikator für die Schätzgenauigkeit aller betrachteten Stories pro Projekt. Je

niedriger die Standardabweichung ist, desto genauer sind die Schätzungen innerhalb des Projekts.

1.6 Aufbau der Arbeit

In *Kapitel 2 Planung in Softwareprojekten* wird das Ziel und die Wichtigkeit von Planungen und Schätzungen für Projekte, sowohl für interne Projekte, als auch für verkaufte Projekte, beleuchtet. Des Weiteren wird die grundsätzliche Veränderung der Genauigkeit von Schätzungen im Verlauf des Projekts („Cone of Uncertainty“) beschrieben.

In *Kapitel 3 Vorgehensmodelle für Softwareentwicklung* werden wichtige traditionelle und danach verbreitete agile Vorgehensmodelle für die Softwareentwicklung beschrieben. Traditionelle und agile Vorgehensmodelle unterscheiden sich in vielen Aspekten voneinander, weshalb Erkenntnisse, die für traditionelle Entwicklungsprojekte zutreffend sind, nicht unmittelbar auch für agile Projekte Gültigkeit haben. Die Kenntnis der unterschiedlichen Vorgehensmodelle soll zum Verständnis dieser Unterschiede beitragen.

Im folgenden *Kapitel 4 Aufwandsschätzung in traditionellen Softwareprojekten* werden ausgewählte Schätzmethoden für traditionelle Projekte vorgestellt und deren Nützlichkeit betrachtet. Einflussfaktoren auf die Schätzungen, sowie Einflussfaktoren auf den Gesamtaufwand in Entwicklungsprojekten, werden beschrieben.

Kapitel 5 Aufwandsschätzung in agilen Softwareprojekten beschäftigt sich mit der Art der Aufwandsschätzung in agilen Projekten und den überwiegend verwendeten Einheiten (Story Points und Ideale Tage). Vor- und Nachteile der Schätzung mit diesen Einheiten und eine alternative Methode, die Schätzung in tatsächlicher Zeit, werden diskutiert.

Im *Kapitel 6 Einflussfaktoren auf die Genauigkeit von Schätzungen*, werden in der Literatur beschriebene Einflussfaktoren auf die Genauigkeit von Schätzungen – ohne Unterscheidung des Vorgehensmodells – zusammengefasst. Es werden sowohl Faktoren, die die Genauigkeit negativ beeinflussen, als auch jene, die die Genauigkeit der Schätzungen positiv beeinflussen, berücksichtigt.

Im *Kapitel 7 Zusammenfassung Teil 1* werden die gefundenen Einflussfaktoren nochmals übersichtlich aufbereitet und deren Anwendbarkeit auf Schätzungen in agilen Projekten auf Basis von Story Points überprüft.

Der zweite Teil der Arbeit, beginnend ab *Kapitel 8 Fallstudie* widmet sich dem Versuch, die Ergebnisse aus dem ersten Teil der Arbeit an Hand von Erhebungen in einem österreichischen IT-Unternehmen mit rund 70 Mitarbeitern zu verifizieren. Dafür werden Mitglieder der sechs Entwicklungsteams dieses Unternehmens in Experteninterviews zu ihren Erfahrungen mit agiler Projektentwicklung, mit Schätzungen und Erfolgsfaktoren für genaue Schätzungen befragt. Diese Interviews werden durch induktive Kategorienbildung ausgewertet.

Um die Ergebnisse der Auswertung der Experteninterviews zu stützen, werden alle Mitarbeiter in der Software-Entwicklung mittels Fragebogen zu ihrer Meinung, in wie weit die gefundenen Erfolgsfaktoren einen Einfluss auf Schätzungen haben, befragt. Schließlich erfolgt noch die

Auswertung von historischen Daten über Schätzungen und tatsächlichen Aufwand aus dem untersuchten Unternehmen.

Im abschließenden *Kapitel 9 Fazit und Ausblick* erfolgt eine Zusammenfassung der Ergebnisse der Arbeit und es wird eine Anregung für weiterführende Untersuchungen gegeben.

2 PLANUNG IN SOFTWAREPROJEKTEN

"Estimation is valuable when it helps you make a significant decision"

Martin Fowler, Autor und Mitunterzeichner des
Manifests für agile Software-Entwicklung

Die Planung von Kosten für und Terminen in Softwareprojekten ist wesentlich für den Erfolg von Entwicklungsprojekten jeglicher Größe. Vor Beginn eines Projekts hilft die Planung, eine Entscheidung für oder gegen die Durchführung eines Projekts zu treffen. Abhängig von den voraussichtlichen Kosten für ein Projekt wird dieses genehmigt oder abgelehnt. Nach Beginn des Projekts ist die Planung Grundlage für die Überprüfung von Meilensteinen. Sie dient somit als Richtwert dafür, wie weit ein Projekt zu einem bestimmten Zeitpunkt umgesetzt sein muss, damit die zeitgerechte Fertigstellung möglich ist. Abhängig von den Meilensteinen in Software-Entwicklungsprojekten sind einige andere Planungen: Innerhalb der Entwicklungsabteilung muss zum Beispiel geplant werden, welches Personal zu welchem Zeitpunkt verfügbar sein muss. Ebenso haben auch andere Abteilungen in einem Unternehmen ein Interesse an den Terminen in einem Softwareprojekt: Das Marketing für ein neues Produkt muss geplant werden, Schulungen für interne und externe Nutzer müssen organisiert werden und Erlösplanungen sind an die Verfügbarkeit von Produktversionen zu einem bestimmten Termin gekoppelt.

Die Planung (und damit die Schätzung, auf welcher die Planung beruht) hilft aber auch, wertvolle Produkte zu entwickeln. Abhängig von der Planung und dem tatsächlichen Fortschritt kann dieses oder jenes Feature für ein Produktrelease umgesetzt oder nicht umgesetzt werden. Alternativ wird der Versionsinhalt reduziert, um die Freigabe zu einem bestimmten Termin sicherzustellen.

Somit bilden die Schätzungen und die Planungen in einem Software-Entwicklungsprojekt die Basis für Aussagen, wann ein Projekt abgeschlossen sein wird, welche Inhalte umgesetzt werden können, wie hoch die Kosten für das Projekt sein werden und wie hoch der Aufwand für die Umsetzung des Projekts sein wird (Cohn, Agile Estimating and Planning, 2006).

Ebenso wie Mike Cohn weist auch Peter R. Hill in (Hill, 2011) darauf hin, dass die Schätzung einer der wesentlichen Erfolgsfaktoren für den Projekterfolg ist. Grundlage für diese Aussage sind die Ergebnisse des zweijährlich erscheinenden CHAOS-Reports der Standish Group. Hill bezieht sich dabei insbesondere auf parameter-basierte Schätzungen, die formale Tools und Methoden verwenden, in geringerem Maß aber auch auf Projekte, die informelle Methoden für Schätzungen verwenden.

Laut (Hill, 2011) sind Schätzungen und Planungen nicht nur für Projekte, die direkt verkauft werden, sondern auch für alle anderen Projekte wesentlich:

- Versäumte Freigabetermine können zu verlorenen Geschäften führen
- Wenn ein Projekt fehlschlägt, werden Ressourcen „verschwendet“ und stehen nicht für andere Projekte zur Verfügung
- Ein abgebrochenes Projekt führt üblicher Weise dazu, dass die Kosten für dieses Projekt nutzlos investiert wurden
- Wenn Kosten überschritten werden, ist unter Umständen der Business-Case für die Umsetzung eines Software-Projekts nicht mehr gegeben.

Andererseits dienen Schätzungen dazu, zu ermitteln, ob ein Plan für die Umsetzung eines Softwareprojekts realistisch ist (McConnell, 2006). Nur wenn die Ziele in einem Projekt realistisch sind, besteht die Möglichkeit, diese Ziele durch Steuerung des Projektverlaufs auch zu erreichen. Wenn die Geschäftsziele für ein Projekt von den Schätzungen und den zu Beginn eines Projekts geplanten Inhalten um mehr als 20 Prozent abweichen, ist es unwahrscheinlich, dass das Projekt diese Ziele auch erreichen kann. Eine Planung zu Beginn des Projekts kann solche Differenzen aufdecken und gibt den Projektbeteiligten somit die Möglichkeit, die Projekteinhalte, den Zeitplan oder die Ressourcen anzupassen, um die gesteckten Ziele auch erreichen zu können. Vgl. auch (Angermeier, 2009).

2.1 Unsicherheit von Schätzungen im Projektverlauf

Unabhängig vom gewählten Vorgehensmodell für die Projektentwicklung und von der gewählten Methode für die Schätzung in einem Projekt unterliegt die Schätzung einer gewissen Unsicherheit. Diese Unsicherheit ist zu Beginn eines Projekts am größten und verringert sich im Laufe eines gut geführten Projekts mit zunehmendem Fortschritt des Projekts. Die Ursache für die Verringerung der Unsicherheit ist, dass zu Beginn eines Projekts eine Vielzahl von Entscheidungen noch nicht getroffen ist. Im Laufe des Projekts werden immer mehr Entscheidungen getroffen und die Unsicherheit in der Schätzung somit reduziert. Diese Abnahme der Unsicherheit wird im Cone of Uncertainty abgebildet (McConnell, 2006).

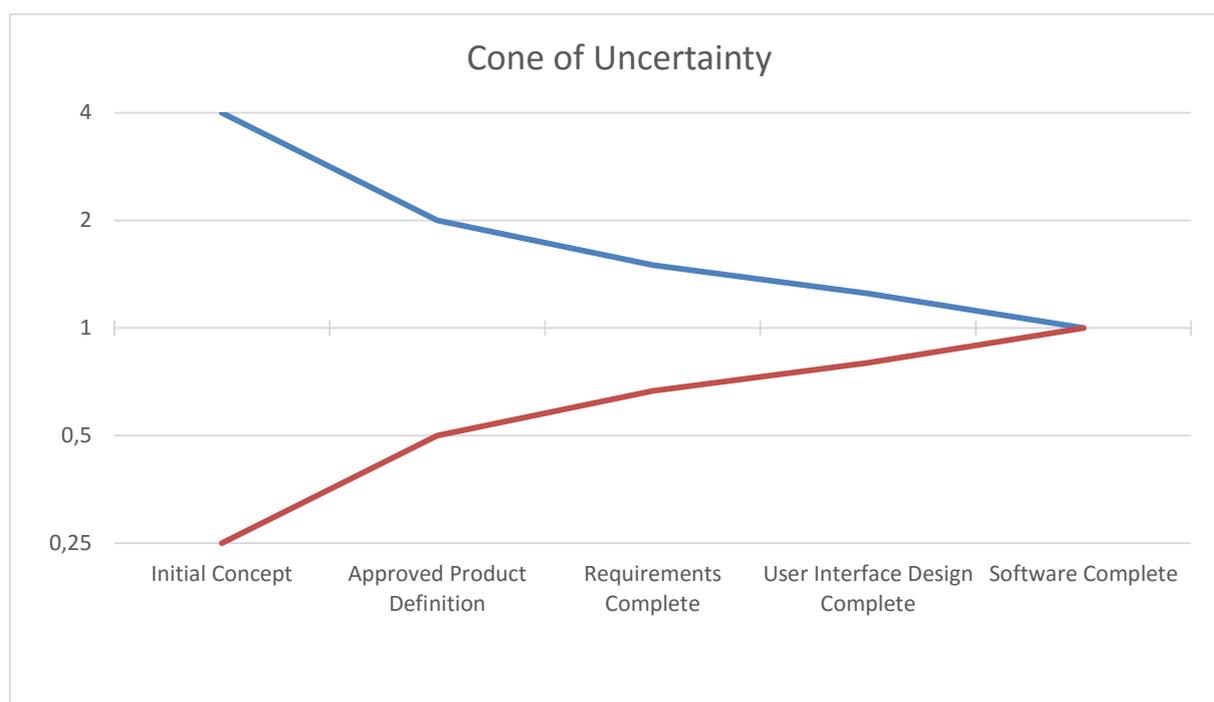


Abbildung 1: Der Cone of Uncertainty bezogen auf Projektmeilensteine, nach (McConnell, 2006)

Zu Beginn eines Projekts ist die Ungenauigkeit in den Schätzungen am höchsten. Die Schätzungen von erfahrenen Projektmanagern enthalten zu diesem Zeitpunkt Fehler im Bereich von 0,25 bis zum vierfachen des tatsächlichen Aufwands. Insgesamt sind zu diesem Zeitpunkt somit Abweichungen um das 16fache zwischen der niedrigsten und der höchsten Schätzung zu erwarten. Je weiter das Projekt voranschreitet und die Unsicherheit durch bereits getroffene Entscheidungen reduziert wird, desto mehr verringert sich der Fehler in den Schätzungen, um bis zum Ende des Projekts bei 1 zu konvergieren. Diese Konvergenz erfolgt jedoch nicht zwangsläufig, sondern durch aktives Projektmanagement und dem Auflösen von Quellen der Unsicherheit im Projektverlauf (McConnell, 2006).

Der Cone of Uncertainty ist sowohl für traditionelle (sequentielle) als auch – in angepasster Form – für iterative Projekt zutreffend. Bei vollständig iterativen Projekten, in denen in jeder Iteration ein vollständiger Entwicklungszyklus von der Anforderungsdefinition bis zur Freigabe durchlaufen wird, trifft der Cone für jeden Zyklus in „Miniaturform“ zu. Wenn in einem Projekt nur teilweise iterativ gearbeitet wird, d.h. die Anforderungen werden zu Beginn des Projekts definiert und die Umsetzung erfolgt danach iterativ, trifft der Cone ebenfalls zu. In diesem Fall erreicht man nach Festlegung der Anforderungen eine Genauigkeit in den Schätzungen von +/- 33% und somit schon wesentlich geringere Fehler als zu Projektbeginn (McConnell, 2006).

Laut (Coelho & Basu, 2012) ist nach der ersten Iteration der Meilenstein „Initial Concept“ (bzw. „Initial Product Definition“ lt. Barry Boehm) erreicht, nach der zweiten Iteration der Meilenstein „Approved Product Definition“, und so weiter.

Barry Boehm hat die Skala der Unsicherheit 1981 noch mit geringeren Schwankungsbereichen (von 0,6 bis 1,6) dargestellt, insgesamt jedoch die gleiche Abnahme der Unsicherheit im Projektverlauf illustriert (Cohn, Agile Estimating and Planning, 2006). Andere Darstellungen

gehen von einer nicht symmetrischen Veränderung der Unsicherheit aus. Diese Darstellungen berücksichtigen, dass es wahrscheinlicher ist, dass der tatsächliche Aufwand höher als der geschätzte ist, als dass der tatsächliche Aufwand geringer ist (Cohn, Agile Estimating and Planning, 2006), (McConnell, 2006).

Einerseits gibt es also eine Vielzahl von Argumenten für eine „gute“ Planung basierend auf genauen Schätzungen, andererseits ist die Schwankungsbreite von Schätzungen besonders zu Beginn eines Projekts beträchtlich. Die Planung wird somit erschwert, unabhängig davon, ob ein Projekt sequentiell oder iterativ umgesetzt wird. Wenn jedoch die Qualität der Schätzungen erhöht werden kann, kann damit auch die Qualität der Planung erhöht werden (McConnell, 2006).

3 VORGEHENSMODELLE FÜR SOFTWAREENTWICKLUNG

Im diesem Kapitel werden unterschiedliche wichtige Vorgehensmodelle für die Umsetzung von Software-Entwicklungsprojekten beschrieben. Eine übergreifende Gruppierung der Vorgehensmodelle ergibt sich durch die Unterteilung in traditionelle und in agile Vorgehensmodelle, wobei für beide Gruppen wichtige Beispiele aufgegriffen werden.

3.1 Traditionelle und agile Vorgehensmodelle

Im Jahr 1970 hat Winston W. Royce (Royce, 1970) ein Vorgehensmodell für das Management von großen Software-Projekten vorgestellt. Dieses Modell wird im Allgemeinen als Wasserfallmodell, bei dem die einzelnen Schritte von der Erhebung der Systemanforderungen über die Analyse, die Codierung und die Inbetriebnahme sequentiell durchlaufen werden, bezeichnet. Das Wasserfallmodell und andere Modelle werden als traditionelle oder schwergewichtige Vorgehensmodelle bezeichnet. Andere Beispiele für traditionelle Vorgehensmodelle sind der Unified Process oder das Spiralmodell (Awad, 2005), (Boehm, A Spiral Model of Software Development and Enhancement, 1988).

Demgegenüber stehen leichtgewichtige oder agile Vorgehensmodelle, die unter anderem mehr Wert auf Zusammenarbeit mit dem Auftraggeber, Änderungen in den Anforderungen auch spät im Entwicklungsprozess und laufende Auslieferung von funktionierender Software legen. (Beck, et al., Principles behind the Agile Manifesto, 2001), (Beck, et al., Manifesto for Agile Software Development, 2001) Einige Beispiele für agile Vorgehensmodelle in der Softwareentwicklung sind Scrum, Extreme Programming XP, Kanban, Agile Unified Process oder Crystal (Moniruzzaman & Hossain, 2013), (Steyer, 2010).

3.2 Traditionelle Vorgehensmodelle

3.2.1 Wasserfallmodell

Das Wasserfallmodell (Royce, 1970) basiert auf den einfachsten Implementierungsschritten der Analyse und der Codierung, die sequentiell aufeinanderfolgen.

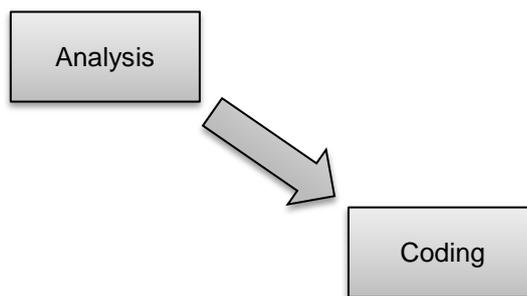


Abbildung 2: Implementierungsschritte für kleinste Softwareprojekte. Nach (Royce, 1970)

Diese einfache Aufteilung ist aber nur in kleinsten Softwareprojekten zielführend. In größeren Projekten besteht das Wasserfallmodell aus wesentlich mehr Schritten, wie in der folgenden Grafik abgebildet. Ein gewisses Maß an Iteration wohnt auch schon dem Modell von Royce inne, da er erkannt hat, dass Probleme, die in einem späteren Schritt des Wasserfalls auftreten, in einem der vorangehenden Schritte gelöst werden müssen. Der Rücksprung auf einen früheren Schritt sollte jedoch auf den unmittelbaren Vorgänger beschränkt sein, was Royce als stimmigen Ansatz beschreibt.

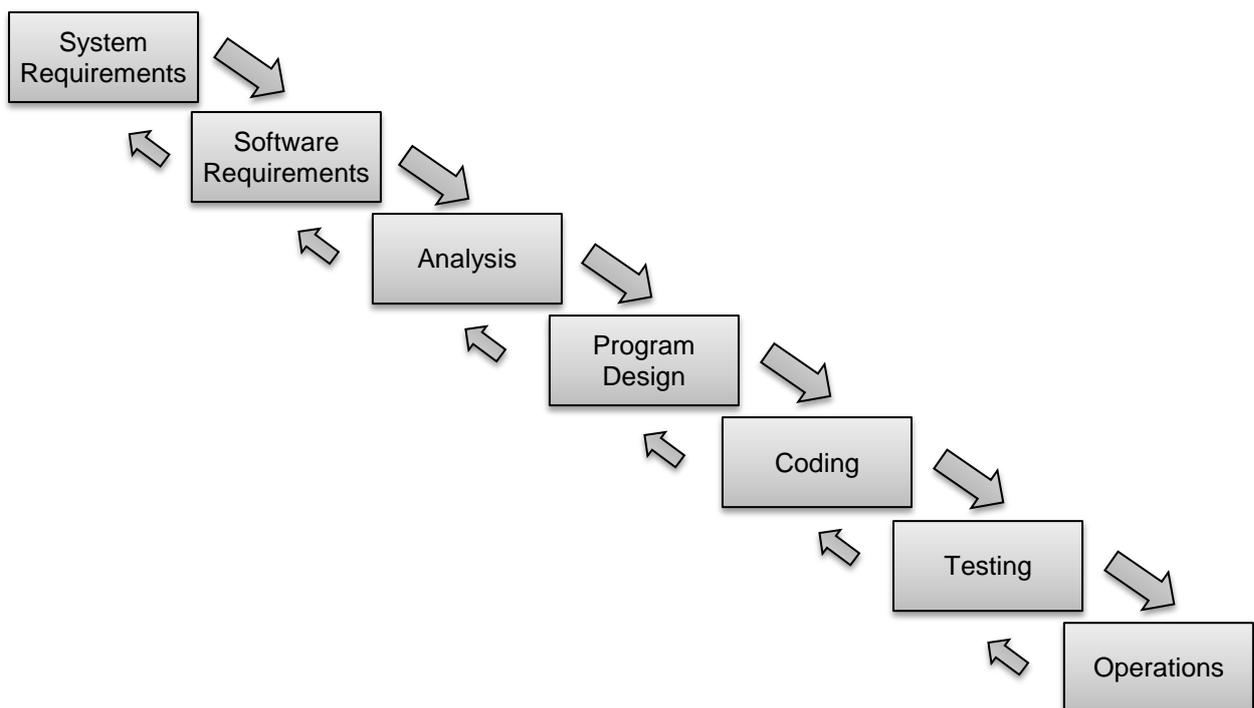


Abbildung 3: Das Vorgehensmodell nach Royce (Wasserfallmodell), wobei Iterationen auf aufeinanderfolgende Schritten beschränkt sein sollen (symbolisiert durch die Pfeile zwischen den Schritten). Nach (Royce, 1970).

Jeder Schritt in diesem Modell besteht aus einer definierten Anzahl von Aktivitäten, die Artefakte eines Schritts müssen fertiggestellt werden, bevor der nächste Schritt begonnen wird (Awad, 2005).

Im Ansatz von Royce erfolgen also die Anforderungserhebung und die Analyse, die maßgeblichen Eingangparameter für die Aufwandsschätzung, zu Beginn des Projekts. Bevor mit der Umsetzung begonnen wird, sind nach diesem Modell die Analyseschritte vollständig abgeschlossen.

3.2.2 Unified Process

Im Unified Process werden die notwendigen Aktivitäten in einem Entwicklungsprojekt in Workflows und Phasen eingeteilt. Alle Aktivitäten werden iterativ und inkrementell ausgeführt. Die Workflows wie die Geschäftsmodellierung und das Requirements-Engineering erstrecken sich über mehrere Phasen, wobei der Aufwand für diese beiden Workflows nach einem Anstieg und Maximum in der ersten Phase im weiteren Verlauf des Projekts abnehmen. Die Phasen im Unified Process Lifecycle sind

- Inception (Beginn): Mit dem Ende dieser Phase ist die Machbarkeit des Projekts sichergestellt, ein Businesscase für das Projekt ist vorhanden und das Ziel des Designs ist festgelegt.
- Elaboration (Ausarbeitung): Die Basisarchitektur wurde erstellt und der Plan für die Umsetzung gefasst.
- Construction (Erstellung): Nach dieser Phase ist ein ausführbares Produkt verfügbar, das für Tests genutzt werden kann.
- Transition (Übergang): In dieser Phase wird das System an die künftigen Nutzer übergeben. Am Ende sollten das Projektteam und die Stakeholder übereingekommen sein, dass die Projektziele erreicht wurden (Awad, 2005).

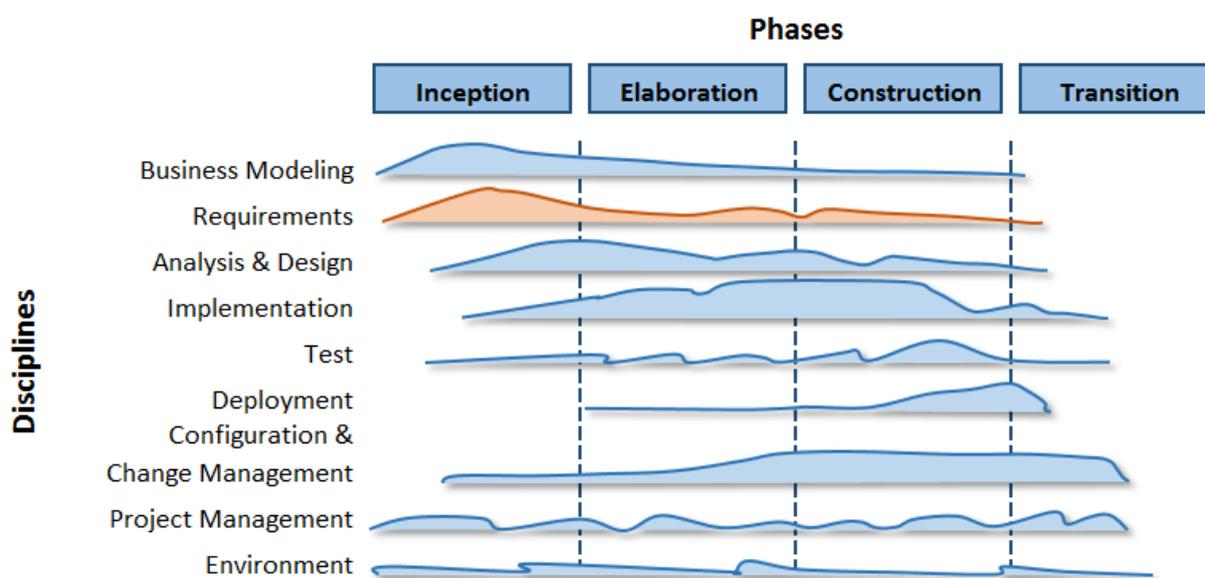


Abbildung 4: Die Phasen und Workflows (Disciplines) im Verlauf des Unified Process Lifecycles. Nach: (Rational Software Corporation, 2001)

Auch in Unified Process wird der höchste Anteil der Arbeiten für die Anforderungserhebung zu Beginn des Projekts erledigt. Anders als im Wasserfallmodell endet die Anforderungserhebung aber nicht zu einem festgelegten Zeitpunkt, sondern erstreckt sich von der Inception über die Elaboration und läuft mit der Construction bis in die Transition aus. Anforderungen werden in Form von Use-Cases und Szenarios verwaltet.

3.2.3 Spiral Modell

Das Spiralmodell ist als Weiterentwicklung des Wasserfallmodells entstanden. Die Schritte des Wasserfallmodells werden in iterativer Form mit zunehmender Detaillierung wiederholt. In einer ersten Schleife wird eine Machbarkeitsstudie erstellt, in der zweiten Schleife ein „Concept of Operations“, in der dritten Schleife eine Spezifikation der Top-Level Anforderungen und so fort (Boehm, A Spiral Model of Software Development and Enhancement, 1988).

Das Spiralmodell, ein ebenfalls schwergewichtiges Vorgangmodell, versucht durch seinen Aufbau Top-Down und Bottom-Up-Konzepte zu kombinieren (Awad, 2005).

3.2.4 Eigenschaften von traditionellen Vorgehensmodellen

Traditionelle oder schwergewichtige Vorgehensmodelle weisen laut (Awad, 2005, S. 6f) einige Charakteristika auf; unter anderem ist das die Tendenz dazu, zu Beginn des Projekts einen Großteil der Planung in hohem Detaillierungsgrad für einen längeren Zeitraum durchzuführen. Dies folgt dem Ansatz, dass das Entwicklungsprojekt wiederholbar und vorhersagbar ist.

Andere Charakteristika sind eine umfassende Dokumentation, eine Prozessorientierung mit definierten Tasks, und einem hohen Einsatz von Tools (Awad, 2005).

3.3 Agile Vorgehensmodelle

Das (im Jahr 2011) am häufigsten eingesetzte agile Vorgehensmodell ist Scrum (52%), gefolgt von Scrum/XP Hybrid (14%). Extreme Programming wird zu 2% eingesetzt (Moniruzzaman & Hossain, 2013) mit Bezug auf den State of agile Report 2011.

Vergleichbare Zahlen liefert der 10th Annual State of agile Report (VersionOne Inc., 2016): Im Jahr 2015 haben 58% der Teilnehmer an der Umfrage Scrum eingesetzt, 10% Scrum/XP Hybrid sowie weitere 1% XP. An dritter Stelle finden sich mit 8% hybride Vorgehensmodelle.

Agile Vorgehensmodelle unterscheiden sich in vielen Punkten von traditionellen Vorgehensmodellen, drei dieser Unterschiede sind in der folgenden Tabelle dargestellt:

| Aspekt | Traditioneller Zugang | Agiler Zugang |
|----------------------------------|---|---|
| Entwicklungs-Lebenszyklus | Linear, Lebenszyklus-Modell (Wasserfall, Spiral oder Variationen) | Iterativ; Evolutionary Delivery Model |
| Stil der Entwicklung | Vorausschauend | Adaptiv |
| Anforderungen | Früh bekannt, hochgradig stabil; klar definiert und dokumentiert | Entstehend, sich ändernd, unbekannt – ermittelt im Projektverlauf |

Tabelle 1: Vergleich von Aspekten in der traditionellen und agilen Software-Entwicklung; Angepasst und gekürzt aus (Moniruzzaman & Hossain, 2013)

3.3.1 Scrum

Das Rahmenwerk Scrum wurde von Ken Schwaber und Jeff Sutherland entwickelt, um die involvierten Personen in die Lage zu versetzen, produktiv und kreativ Produkte mit höchstmöglichem Wert auszuliefern (Schwaber & Sutherland, 2014).

Die Erstellung eines Produkts erfolgt in Sprints – einem Zeitraum von zwei bis vier Wochen, innerhalb dessen ein Produkt-Inkrement fertiggestellt wird. Die Anforderungen für die Produktentwicklung werden im Backlog in Form von sogenannten Backlog-Items verwaltet. Die Sammlung aller Anforderungen wird Product-Backlog genannt. Der Teil der Anforderungen, die innerhalb eines Sprints umgesetzt werden sollen, wird als Sprint-Backlog bezeichnet.

Das Product Backlog ist emergent, das heißt, es wird im Laufe des Projekts durch das Scrum-Team laufend weiterentwickelt. Es stellt eine Sammlung von allem, was künftig im Produkt enthalten sein soll, dar. Die gewünschten Eigenschaften des Product Backlogs werden oft (z.B. von Mike Cohn) mit dem Akronym „DEEP“ beschrieben:

- **Detailed appropriately** (angemessen detailliert): Je weiter oben ein Eintrag im Backlog steht, desto detaillierter ist er ausgearbeitet. Einträge, die weit unten im Backlog stehen, sind wenig detailliert und bestehen manchmal nur aus einem Titel.
- **Estimated** (geschätzt): Der Aufwand für jeden Backlog-Eintrag ist geschätzt. Die Schätzung ist eher grob und wird oft als relative Größe in Form von Story-Points oder in idealen Tagen angegeben.
- **Emergent** (entstehend): Das Product Backlog ändert sich laufend, Einträge werden verfeinert, neu hinzugefügt, anders priorisiert oder entfernt.
- **Prioritized** (priorisiert): Die wichtigsten (wertvollsten) Einträge stehen ganz oben im Backlog. Je weniger wichtig ein Eintrag ist, desto weiter unten steht er im Backlog.

In welcher Form Backlog-Items formuliert werden, wird von Schwaber und Sutherland nicht näher beschrieben. Eine vielfach genutzte Form der Beschreibung sind User-Stories; wenn die Items im Backlog noch nicht detailliert ausformuliert sind oder nur grobe Ziele enthalten, werden

diese auch als Epic bezeichnet. Ebenso wie User-Stories können auch andere Arten der Beschreibung, wie zum Beispiel Skizzen oder Prototypen, hilfreich sein (Pichler, 2010).

Zwei wesentliche Eigenschaften der Anforderungen für ein Produkt, das mit Scrum entwickelt wird, sind somit, dass diese geschätzt sind und dass sie emergent sind. Die Anforderungen sind somit zu Beginn eines Projekts nicht vollständig bekannt oder beschrieben, sie können sich im Laufe eines Projekts ändern, Anforderungen können dazukommen oder weggelassen werden. Erst zu dem Zeitpunkt, wenn ein Backlog-Item in das Sprint-Backlog aufgenommen wird, muss es spezifiziert sein und darf nicht mehr verändert werden (Steyer, 2010) und (Schwaber & Sutherland, 2014).

3.3.2 Extreme Programming XP

Extreme Programming, eine *„leichtgewichtige Methode für kleine bis mittelgroße Teams, die Software im Angesicht von vagen oder sich schnell ändernden Anforderungen entwickeln“* (Beck, Extreme Programming explained, 2000) wurde im Jahr 2000 von Kent Beck beschrieben. Ziel von Beck ist es, mit der von ihm vorgestellten Methode bestehende Probleme in der Software-Entwicklung zu lösen. Die wesentlichen Probleme sieht er in folgenden Risiken:

- Der Termin für die Auslieferung der fertigen Software kann nicht eingehalten werden.
- Eine erfolgreich eingeführte Software kann auf Dauer nicht gewartet werden, da die Kosten für Änderungen immer weiter ansteigen und die defect-rate steigt.
- Eine Software wird in Betrieb genommen, ohne das eigentliche Geschäftsproblem, für das sie entwickelt wurde, zu lösen oder das Geschäftsproblem, für welches die Software entwickelt wurde, hat sich verändert oder besteht in dieser Form nicht mehr.

Die angeführten Risiken sollen mit Extreme Programming durch Praktiken für die tägliche Arbeit des Software-Entwicklers reduziert werden. Diese zwölf Praktiken basieren auf fünfzehn Prinzipien und diese wiederum auf vier Werten (Kommunikation, Einfachheit, Feedback und Mut).

Beck schlägt vor, XP vor allem dann einzusetzen, wenn sich die Anforderungen an ein Produkt rasch entwickeln oder vage sind. Dabei geht Beck davon aus, dass der Kunde zu Beginn eines Projekts niemals exakt Auskunft darüber geben kann, welche Anforderungen er an eine zu entwickelnde Software hat. Des Weiteren ändern sich die Anforderungen an eine Software, sobald ein Release vorhanden ist oder eingesetzt wird. Die Anforderungen, die durch den Einsatz einer Software entstehen, bezeichnet Beck als wertvoll, da diese Anforderungen nicht auf Spekulationen basieren. Somit sind diese sich ändernden Anforderungen eine Chance, und kein Risiko (Beck, Extreme Programming explained, 2000), (Kuhmann, 2013).

3.3.3 Eigenschaften von agilen Vorgehensmodellen

Agile Vorgehensmodelle sind leichtgewichtig und versprechen dadurch eine höhere Flexibilität als traditionelle Vorgehensmodelle. Die Entwicklung soll früh in einem Projekt beginnen, die

künftigen Anwender sollen stark einbezogen werden und funktionsfähige Software soll zu einem frühest möglichen Zeitpunkt ausgeliefert werden (Kuhmann, 2013).

Laut (Awad, 2005) sind agile Vorgehensmodelle adaptiv, das bedeutet, dass in agilen Projekten Änderungen zu jedem Zeitpunkt als Chance, und nicht als Risiko gesehen werden (siehe auch (Beck, et al., Principles behind the Agile Manifesto, 2001)). Die involvierten Personen versuchen also nicht, Änderungen zu verhindern, sondern Wege zu finden, Veränderungen besser zu handhaben.

Des Weiteren sind agile Vorgehensmodelle in einem empirischen oder nichtlinearen Prozess organisiert, da es nicht wahrscheinlich ist, dass ein vordefinierter Prozess in der Softwareentwicklung tatsächlich zum gewünschten Ergebnis führt. Dies begründen (Williams & Cockburn, 2003) damit, dass es im Projektverlauf Änderungen in den Anforderungen, Änderungen von eingesetzten Technologien und Veränderungen im Team geben wird.

4 AUFWANDSSCHÄTZUNG IN TRADITIONELLEN SOFTWAREPROJEKTEN

„Eine gute Schätzung ist eine Schätzung, die einen ausreichend klaren Blick auf die Realität des Projekts gibt, um dem Projektmanagement gute Entscheidungen zu ermöglichen, wie das Projekt zu kontrollieren ist, um die Projektziele zu erreichen.“
(McConnell, 2006)

Steve McConnell sieht die Schätzung in Softwareprojekten als Basis für die Steuerung des Projekts. Dabei weist er auch auf den Unterschied zwischen Schätzungen, Zielen und Zusagen hin. Eine Schätzung in diesem Sinne ist eine Voraussage, wie lange ein Projekt dauern wird, während ein Ziel ein Statement über ein gewünschtes Geschäftsziel ist. Und eine Schätzung über den Projektaufwand ist keine Zusage (commitment), obwohl diese Unterscheidung in der Kommunikation oft nicht berücksichtigt wird.

Im Zuge der Schätzung – egal ob in traditionellen oder in agilen Projekten – stellt sich die Aufgabe, die Größe des Softwareprojekts auf Aufwand, Dauer und Kosten für das Projekt umzulegen.

4.1 Ableiten von Schätzungen aus Zählungen

Die von (McConnell, 2006) vorgeschlagene Methode der ersten Wahl ist es, etwas zu finden, das man zählen kann, um aus dieser Zählung mit verschiedenen Daten den Aufwand zu errechnen. Nur wenn es nicht möglich ist etwas zu zählen, soll das Ermessen als letzte Zuflucht dienen.

Laut McConell gibt es in Softwareprojekten viele Dinge, die gezählt werden können, einige Beispiele dafür sind:

Am Beginn eines Projekts können das marketing requirements, Features, Use Cases oder Stories sein. In der Mitte eines Projekts können das technische Anforderungen, Function-Points, Web-Seiten, Reports, Dialogboxen, Bildschirmseiten oder Datenbanktabellen sein. Gegen Ende eines Projekts können Codezeilen, Klassen, Tasks oder gemeldete Fehler gezählt werden. Neben diesen Beispielen gibt es noch viele weitere Möglichkeiten für Zählungen.

Für die Auswahl, was gezählt werden soll, gibt McConnell einige Ratschläge:

- *Finde etwas, das in hohem Maße mit der Größe der zu schätzenden Software korreliert.* In Projekten, in denen der Inhalt fixiert ist und Kosten und Terminplanung geschätzt werden sollen, sind marketing requirements, technische Anforderungen oder Function Points Größen, die in hohem Maße mit der endgültigen Größe des zu entwickelnden Systems zusammenhängen. Solche Größen eignen sich, um gezählt zu werden.
- *Finde etwas, das eher früher als später im Entwicklungszyklus zur Verfügung steht.* Je früher es möglich ist, etwas zu zählen, desto früher ist es möglich, Vorhersagen über das Projekt zu machen. Frühe Vorhersagen werden aber in aller Regel ungenauer sein, als Vorhersagen, die später im Projektverlauf gemacht werden.
- *Finde etwas, das ein statistisch sinnvolles Mittel als Ergebnis erzeugen wird.* Nur, wenn ein ausreichend großes Sample zur Verfügung steht, kann man statistisch sinnvolle Aussagen treffen. Eine Anzahl von 20 oder mehr Objekten sollte daher für die Zählung zur Verfügung stehen.
- *Verstehe, was du zählst.* Um aus einer Zählung eine Schätzung ableiten zu können, muss der Aufwand für eine Einheit der gezählten Objekte aus historischen Daten abgeleitet werden. Damit diese Ableitung aus historischen Daten zuverlässige Werte liefert, müssen die aktuell gezählten Objekte die gleiche Größe wie die gezählten Objekte aus historischen Daten haben. Ein Beispiel dafür sind User Stories. Wenn ein Team in der Vergangenheit 7 User Stories in der Woche umsetzen konnte und man daraus ableiten will, dass das Team im aktuellen Projekt 7 Stories pro Woche umsetzen kann, muss sichergestellt sein, dass das Team die gleiche Größe hat, die eingesetzte Technologie die gleiche ist, dass eine User Story in der Vergangenheit gleich definiert wurde wie im aktuellen Projekt und einiges mehr.
- *Finde etwas, das mit minimalem Aufwand gezählt werden kann.* Abhängig von der notwendigen Genauigkeit der Schätzung können Objekte, die mit wenig Aufwand gezählt werden können, ausreichend sein. McConnell schlägt mit Bezug auf Barry Böhm et al zum Beispiel vor, statt Function Points aus dem Cocomo II Projekt Object Points zu verwenden, da der Aufwand für die Ermittlung derselben rund 50% geringer ist als für Function Points.

Die folgende Tabelle gibt einen Überblick über Objekte, die sich für eine Zählung und damit für eine Aufwandsschätzung durch Umrechnung mit historischen Daten eignen:

Objekte für die Zählung

| |
|--------------------------|
| Marketing requirements |
| Features |
| Use Cases |
| Stories |
| Technische Anforderungen |

| |
|---|
| Function Points |
| Änderungsmeldungen |
| Webseiten |
| Berichte |
| Dialogboxen |
| Datenbanktabellen |
| Klassen |
| Gefundene Fehler |
| Einstellungen |
| Anzahl der bereits geschriebenen Codezeilen |
| Anzahl von Testfällen |

Tabelle 2: Beispiele für Objekte, die gezählt werden können, um Schätzungen abzuleiten, aus: (McConnell, 2006)

4.1.1 Zusammenfassung und Kritik

Für die Fragestellung dieser Arbeit sind besonders zwei der Vorschläge von McConnell interessant: *Finde etwas, das eher früher als später im Entwicklungszyklus zur Verfügung steht.* Je früher eine Vorhersage über den Aufwand in einem Projekt getroffen werden kann, desto wertvoller ist sie. Man sollte also möglichst früh im Projekt eine Schätzung erstellen können. Dies geht aber – wie auch der Cone of Uncertainty zeigt – in aller Regel mit einer geringeren Genauigkeit der Schätzung einher; einem Effekt, der insbesondere vom Management nicht gewünscht ist.

Des Weiteren (*Verstehe, was du zählst.*) stellt McConnell fest, dass für den Vergleich mit historischen Daten sichergestellt sein muss, dass die Annahmen in der Vergangenheit und in der Gegenwart die gleichen sind, da es sonst nicht legitim ist, aus historischen Daten Aussagen über das aktuelle Projekt abzuleiten. Genau diese Übereinstimmung der Annahmen ist ein wesentlicher Punkt, auf den in Folge in dieser Arbeit eingegangen wird, da das Herstellen dieser Übereinstimmung mit vielen Herausforderungen verbunden ist.

4.2 Ableitung von Schätzungen aus Function Points

Die International Software Benchmarking Standards Group (ISBSG) stellt mit dem ISBSG Repository eine Datenbank mit Daten zu über 5.500 Projekten zur Verfügung. Im Repository werden zu den Projekten Daten wie der Projektkontext, die Art des Projekts (Weiterentwicklung, Neuentwicklung), Größe der Entwicklungsteams, Architektur, Plattform, Programmiersprache und die Entwicklungsmethode (46% der Projekte wurden mit traditionellen oder Wasserfall-

Methoden umgesetzt) verwaltet. Ebenfalls enthalten sind Daten über die Projektgröße (gemessen in Function Points), den Gesamtaufwand und die Project Delivery Rate (Hill, 2011).

Peter R. Hill gibt in (Hill, 2011) eine Anleitung dafür, wie auf Basis von historischen Daten Schätzungen für die Größe, den Aufwand und die Dauer von Software-Entwicklungsprojekten erstellt werden können.

4.2.1 Anforderungen, die den Entwicklungsaufwand beeinflussen

Es gibt drei Arten von Anforderungen, die in einem Projekt berücksichtigt werden müssen: Funktionale Anforderungen (functional requirements), nichtfunktionale Anforderungen (nonfunctional requirements) und technische Anforderungen (technical requirements). Alle drei Arten von Anforderungen müssen berücksichtigt werden, um eine realistische Schätzung für den Aufwand einer Projektentwicklung erstellen zu können (Hill, 2011). Die Drei Arten der Anforderungen charakterisieren sich laut Hill wie folgt:

Funktionale Anforderungen repräsentieren, WAS eine Software leisten soll. Funktionale Anforderungen beschreiben, welche Geschäftsfälle die Anwender mit der jeweiligen Funktion abwickeln können. Gemäß der ISBSG-Definition wird die Größe von funktionalen Anforderungen in Function Points ausgedrückt. Funktionale Anforderungen sind die einzigen Arten von Anforderungen, die für das Ermitteln der funktionalen Größe einer Software berücksichtigt werden.

Nichtfunktionale Anforderungen beschreiben, WIE eine Software arbeiten soll. Beispiele für nichtfunktionale Anforderungen sind Genauigkeit, Qualität, Angemessenheit, Konformität (mit legislatischen Vorgaben), Interoperabilität, Sicherheit, Zuverlässigkeit, Wartbarkeit, Übertragbarkeit und Performance. Nichtfunktionale Anforderungen werden oft nicht explizit oder auch gar nicht formuliert, sind aber ebenfalls ein wesentlicher Teil der Projektanforderungen.

Technische Anforderungen werden nicht durch den Anwender, sondern durch Entwickler formuliert. Technische Anforderungen legen fest, wie eine Software entwickelt und gebildet wird.

Die folgende Abbildung zeigt die Arten und Zusammenfassung von Anforderungen in einem Software-Entwicklungsprojekt. Alle drei Arten von Anforderungen zusammengenommen ergeben die Anforderungen an ein Software-Entwicklungsprojekt.

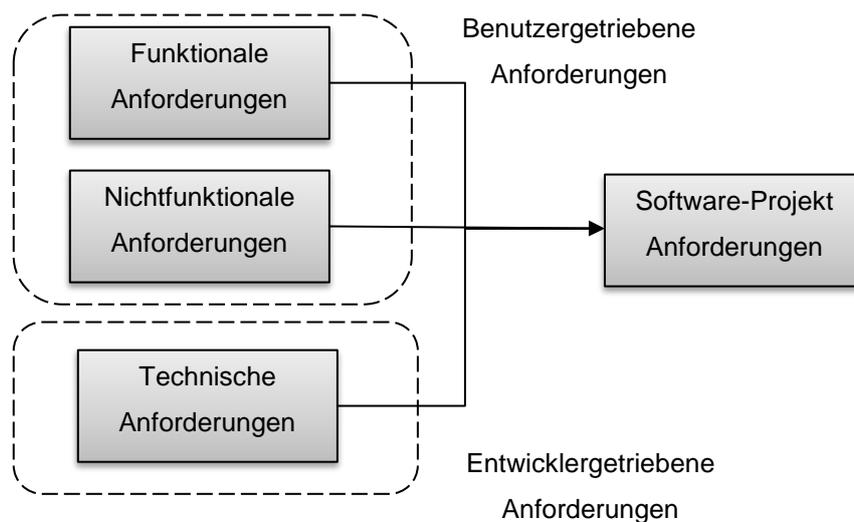


Abbildung 5: Arten von Anforderungen in einem Software-Entwicklungs-Projekt; nach: (Hill, 2011)

4.2.2 Project Delivery Rate

Alle von (Hill, 2011) vorgestellten Methoden zur Ermittlung des Aufwandes von Software-Entwicklungsprojekten leiten aus der funktionalen Größe eines Produkts durch Multiplikation mit der Project Delivery Rate (PDR) den Aufwand her:

$$\text{Aufwand} = \text{Größe} * \text{PDR}$$

Wobei die Größe die *funktionale Größe* gemessen in *Function Points* ist und PDR der Aufwand in Stunden pro *Function Point*.

Die Project Delivery Rate in einem Projekt, oder anders ausgedrückt die Produktivität, hängt von einigen Faktoren ab:

Einer der größten Faktoren, der die PDR beeinflusst, ist die **verwendete Programmiersprache**. Der Medianwert für die PDR variiert, abhängig von der verwendeten Programmiersprache in Midrange Development Environments von 8 bis 13 Stunden.

Der zweite große Faktor, der die PDR beeinflusst, ist die **Teamgröße**. Ab einer Teamgröße von 5 Personen sinkt die Produktivität. Des Weiteren steigt lt. Hill ab einer Teamgröße von fünf Personen auch der Fehler in der Schätzung. Hill geht aber nicht näher auf die Gründe für diese steigende Ungenauigkeit ein.

Eine Begründung für die sinkende Produktivität bei steigender Teamgröße gibt (McConnell, 2006) unter dem Schlagwort **Diseconomy of Scale**. Diese besagt, dass der Aufwand für ein Projekt nicht linear mit der Größe des Projekts steigt, sondern schneller. Ein Projekt, das zehnmal so groß wie ein anderes Projekt ist, verursacht nicht den zehnfachen Aufwand, sondern einen höheren. Die Begründung dafür ist in der Anzahl der möglichen Kommunikationswege zwischen den beteiligten Personen. Bei zwei Personen gibt es einen Kommunikationspfad, bei drei Personen drei, bei vier Personen 6, bei 5 Personen 10 und bei 10 Personen gibt es bereits 45 Kommunikationspfade. Diese quadratische wachsende Anzahl der

Kommunikationswege führt zu einem nicht linearen Anstieg des Aufwands mit wachsender Projekt- bzw. Teamgröße.

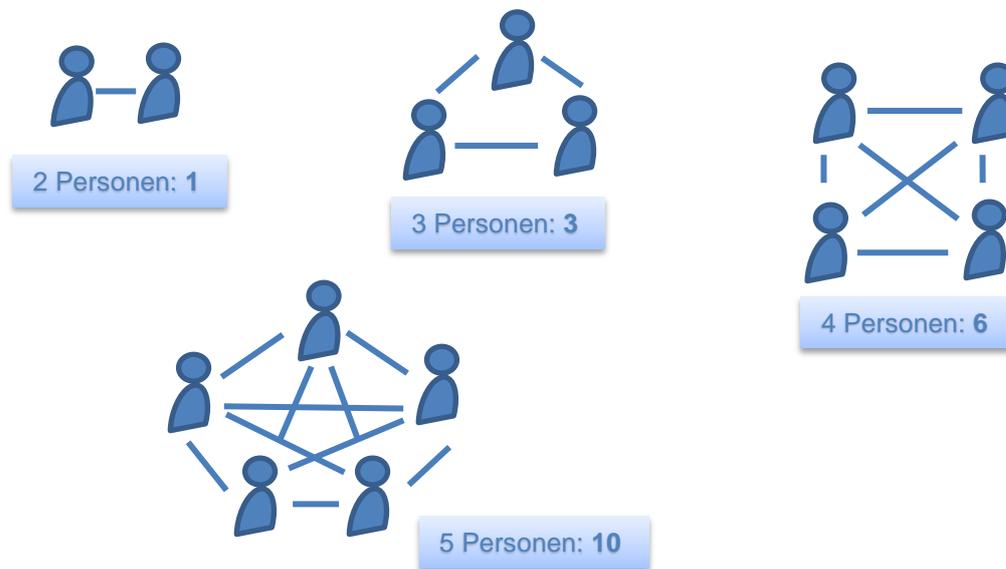


Abbildung 6: Die Anzahl der Kommunikationswege steigt nicht linear, sondern quadratisch proportional mit der Anzahl der Personen im Team. Übernommen und angepasst aus: (McConnell, 2006)

Weitere Faktoren, die auf die PDR einwirken, sind das verwendete Vorgehensmodell, die Art der entwickelten Anwendung, die Entwicklungsplattform und andere.

4.2.3 Zusammenfassung und Kritik

Die Funktionale Größe von Software berücksichtigt für die Ermittlung der Größe eines Software-Projekts nur einen Teil den Anforderungen, die funktionalen Anforderungen. Nicht funktionale Anforderungen und technische Anforderungen werden von der funktionalen Größe nicht erfasst. Somit ist die Messung der Funktionalen Größe nicht für alle Softwareprojekte passend.

(Hill, 2011) weist mehrfach darauf hin, dass bei der Ermittlung der Größe eines Software-Projekts mit Hilfe der Daten aus dem ISBSG-Repository Vorsicht geboten ist und dass einige Punkte zu berücksichtigen sind, wenn die Projektgröße mit dem ISBSG-Repository ermittelt wird, unter anderen:

- Die Projektdatenbank enthält nicht Daten von durchschnittlichen Projekten, sondern von Projekten, die produktiver als der Durchschnitt sind (bedingt dadurch, dass Unternehmen, die Projektdaten erfassen und ins Repository einbringen, einen höheren Prozessreifegrad als der Durchschnitt der Unternehmen besitzen).
- Projekte können aus Komponenten bestehen, die unterschiedlichen Kriterien gemäß der Charakterisierung in den Repository-Daten entsprechen, dadurch können sich die PDR's zwischen den Komponenten wesentlich unterscheiden, was bei der Ermittlung des Gesamtaufwands berücksichtigt werden muss.

- Wenn für die Ermittlung des Aufwands in einem geplanten Projekt die Daten von unpassenden Referenzprojekten verwendet werden, kann die Aufwandsschätzung wesentlich vom „richtigen“ Wert abweichen.
- In unterschiedlichen Unternehmen haben unterschiedliche Faktoren Einfluss auf die Produktivität. Diese Faktoren sind vielfältig und oft nicht einfach zu erkennen. Beispiele dafür sind die Arbeitsumgebung, die Haltung und das Mindset der Mitarbeiter, das Management, die unterschiedlichen Aufgaben der Mitarbeiter und einige mehr. Daher ist es besser, Vergleichsdaten aus dem eigenen Unternehmen als aus anderen Unternehmen (über das ISBSG-Repository) zu verwenden.

Die Basis für die Größenermittlung eines Software-Projekts mit der ISBSG-Methoden bilden Function Points und die Project Delivery Rate (=Aufwand in Stunden pro Function Point). Im Anhang zu seinem Buch präsentiert (Hill, 2011) eine Anzahl von Tabellen, wie hoch die Project Delivery Rate für Projekte in verschiedensten Kategorien ist:

Project Delivery Rate nach maximaler Teamgröße, PC-Plattform

| | N | Min | P10 | P25 | Median | P75 | P90 | Max | Mean | StdDev |
|------------|----|-----|-----|-----|--------|------|------|------|------|--------|
| 1-4 | 19 | 1,0 | 2,8 | 4,0 | 6,4 | 11,2 | 17,9 | 42,3 | 9,7 | 9,7 |
| 5-8 | 35 | 1,0 | 1,8 | 3,5 | 8,4 | 11,1 | 16,8 | 39,2 | 9,3 | 8,7 |
| 9-* | 29 | 1,2 | 4,2 | 7,5 | 9,8 | 17,6 | 32,9 | 49,8 | 15,0 | 13,1 |

Tabelle 3: Auswertung der Project Delivery Rate, kategorisiert nach Teamgröße, aus: (Hill, 2011)

Die in der Tabelle angegebenen Werte sind folgende:

- N ist die Anzahl der Projekte im Sample
- Min und Max sind die minimalen und maximalen Werte im Sample
- P10, P25, P75 und P90 sind die Werte für die 10, 25, ... Prozent der Projekte im Sample unterhalb dieses Wertes liegen
- Median (entspricht P50) ist der Wert, für den die Hälfte der Projekte im Sample unterhalb und die Hälfte oberhalb liegen
- Mean ist der Mittelwert
- StdDev (Standard Deviation oder Standardabweichung) ist ein Maß für die Bandbreite der Werte im Sample.

Die minimalen und maximalen Werte in dieser Beispieltabelle (1,0 bis 39,2 bzw. 49,8) sowie die Standardabweichung (9,7 bis 13,1 bei Medianwerten zwischen 6,4 und 9,8) zeigen, dass die Bandbreite für die PDR innerhalb eines Samples beträchtlich ist. Der geringste und höchste Aufwand variiert in diesem Beispiel etwa um den Faktor 50.

5 AUFWANDSSCHÄTZUNG IN AGILEN SOFTWAREPROJEKTEN

In agilen Projekten wird für die Aufwandsschätzung üblicherweise in zwei Schritten vorgegangen: Zuerst wird die Größe eines Projekts auf Basis der gewünschten Features geschätzt, danach wird aus der Größe der Aufwand und damit die Dauer ermittelt und die Planung erstellt. Um die Größe eines Projekts bzw. der einzelnen Features zu schätzen, sind zwei Einheiten gebräuchlich: *Story Points* und *Ideal Days* (Cohn, *Agile Estimating and Planning*, 2006).

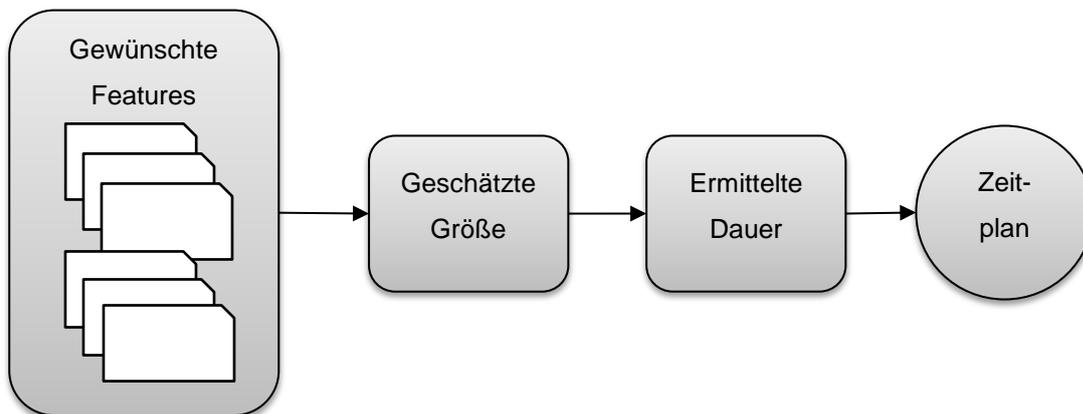


Abbildung 7: Bei der Aufwandsschätzung in agilen Projekten wird aus der geschätzten Größe der Summe der Features die Umsetzungsdauer ermittelt; aus: (Cohn, *Agile Estimating and Planning*, 2006).

Beide in Folge vorgestellten Methoden basieren auf der Schätzung von Experten. Expertenschätzung ist die dominierende Strategie, um den Aufwand von Software-Entwicklungsprojekten zu ermitteln (Jorgensen, 2004), (Basten, 2012); verschiedene Studien belegen, dass Expertenschätzungen („informal analogy“, „intuition and experience“, „expert judgement“) von 62% bis 86% der untersuchten Unternehmen eingesetzt werden. Darüber hinaus stellt (Jorgensen, 2004) fest, dass keine ausreichenden Belege darüber vorliegen, dass formale Modelle genauere Schätzungen als Expertenschätzungen liefern.

5.1 Schätzung mit Story Points

Die Schätzung der Größe von User Stories mit Story Points ist eine weit verbreitete Methode in agilen Softwareprojekten (Cohn, *Agile Estimating and Planning*, 2006), (Hill, 2011), (Coelho & Basu, 2012), (Pichler, 2010), (Bergsmann, 2014) u.a.

Story Points sind ein relatives Maß für die Größe. Eine User Story, die doppelt so groß wie eine andere ist, erhält daher als Schätzung doppelt so viele Story Points. (Cohn, Agile Estimating and Planning, 2006) schlägt vor, Story Points aus einer vorgegebenen Skala zu verwenden; üblich dafür ist eine angepasste Fibonacci-Folge mit den Werten 1, 2, 3, 5, 8, 13, 20, 40 und 100. Diese Folge mit immer größer werdenden Abständen zwischen den Werten drückt aus, dass eine Schätzung umso ungenauer wird, umso größer der zu schätzende Wert ist. Des Weiteren erlaubt diese nicht lineare Folge ein schnelleres Schätzen durch das Entwicklungsteam, da durch die vorgegebenen möglichen Werte keine kleinlichen Diskussionen über die „richtige“ Größe entstehen (Pichler, 2010).

5.1.1 Welche Größe wird mit Story Points gemessen?

Welche Art von Größe Story Points repräsentieren, wird von unterschiedlichen Autoren unterschiedlich gesehen:

(Fowler, et al., 2013) beschreiben den Story Point als Maß für den Aufwand, der für die Umsetzung einer User Story benötigt wird. Dies betrifft den gesamten Aufwand, bis eine Story fertiggestellt ist, also nicht nur die Umsetzung, sondern auch den Testaufwand. Dass ein Story Point ein Maß für die Komplexität einer Story sei, träge nur dann zu, wenn die Komplexität oder das Risiko, das einer Story innewohnt, tatsächlich den Umsetzungsaufwand beeinflusst.

Laut (Leffingwell, 2011) repräsentiert ein Story Point unterschiedliche Aspekte, welche die Größe einer Story beeinflussen:

- Verständnis: Versteht das Team, was der Inhalt der Story ist?
- Komplexität: Wie schwierig ist die Story zu implementieren?
- Volumen: Wie viel Aufwand ist es, die Story umzusetzen; wie lange wird es dauern?
- Ungewissheit: Was ist noch unbekannt und wie könnte dies die Schätzung verändern?

(Bergsmann, 2014) beschreibt Story Points mit Bezug auf das Scrum Glossar von (Kriegisch) als Maß für den Funktionsumfang und gegebenenfalls Komplexität einer Anforderung. Ein Story Point ist demnach kein absolutes Maß für den Aufwand, sondern eine relative Größe zum Vergleich unterschiedlicher Anforderungen. Ein Story Point lässt sich aber unter Berücksichtigung der Velocity eines Teams in Zeitaufwand umrechnen.

Ähnlich beschreibt (Rubin, 2012) Story Points als eine Kombination aus Faktoren wie der physischen Größe und der Komplexität eines Product Backlog Items. Diese Faktoren werden laut Rubin kombiniert und als eine einzelne relative Größe ausgedrückt. Er weist aber auch darauf hin, dass Story Points den Aufwand ausdrücken müssen, den ein Entwicklungsteam für ein Product Backlog Item investieren muss, da Story Points schlussendlich dafür verwendet werden, um eine Zeitdauer zu ermitteln.

Laut (Buglione & Abran, 2007) werden Story Points als Maß für die Zeit verwendet, die benötigt wird, um bestimmte Funktionalität herzustellen und an den Kunden auszuliefern. Die Autoren

(wie auch andere) weisen insbesondere darauf hin, dass Story Points zwischen unterschiedlichen Teams nicht vergleichbar sind, da jedes Team einen eigenen Bezugswert für einen Story Point definiert.

Mike Cohn führt in seinem Blog (Cohn, What are Story Points, 2016) aus, dass Story Points jedenfalls „Aufwand“ („effort“) repräsentieren. Dabei muss die Schätzung alles berücksichtigen, was Aufwand verursacht, wie zum Beispiel:

- Die Menge der Arbeit, die zu erledigen ist
- Die Komplexität der Arbeit
- Jedes Risiko oder jede Unsicherheit im Zusammenhang mit dieser Arbeit

Alle diese drei Faktoren müssen in die Schätzung eines Arbeitspakets einfließen, da alle Faktoren den Aufwand einer Story beeinflussen. Obwohl es sich um unterschiedliche Einflussgrößen handelt, verursachen alle drei Aufwand – und dieser kann in Form einer Zahl – eben „Story Points“ – ausgedrückt werden.

Zusammenfassend kann festgestellt werden, dass das Verständnis darüber, welche Größe ein Story Point repräsentiert, nicht einheitlich ist. Somit kann die gemessene Größe, für welche Story Points verwendet werden, zwischen unterschiedlichen Unternehmen oder unterschiedlichen Teams im gleichen Unternehmen, variieren. Die persönliche Erfahrung hat gezeigt, dass sogar innerhalb eines Teams mitunter abweichende Ansichten darüber herrschen, welche Größe mittels Story Points ausgedrückt wird.

5.1.2 Vorteile der Schätzung mit Story Points

Die relative Schätzung mit Story Points hat den Vorteil, dass das relative Schätzen den meisten Menschen leichter fällt als das absolute Schätzen einer Größe. Sie setzt nicht die genaue Kenntnis der absoluten Größe eines Objekts (einer User Story) voraus, sondern es genügt, unterschiedliche Stories in ein Verhältnis zueinander zu bringen (Fowler, et al., 2013).

Als Vorteil wird auch vielfach angesehen, dass die Schätzung der Größe einer Story von der Schätzung der Umsetzungsdauer entkoppelt wird. Dadurch können viele Einflussfaktoren, wie die Teamgröße, die Erfahrung der Entwickler, die zusätzlichen Aufgaben, die ein Team zu erledigen hat und vieles andere außer Acht gelassen werden. Die relative Größe von User Stories zueinander wird durch diese Faktoren nicht beeinflusst. Wenn sich in weiterer Folge herausstellt, dass die Umsetzungsdauer für einen Story Point höher ist als ursprünglich angenommen, muss die Schätzung nicht verändert werden, sondern lediglich der Faktor für die Umrechnung von Größe (in Story Points) in Dauer (in Stunden oder Tagen) (Cohn, Agile Estimating and Planning, 2006).

Die Schätzung mit Hilfe von Story Points hilft Entwicklungsteams, funktionsübergreifend zu denken. Da für die Umsetzung einer Story eine einzelne Zahl als Schätzung zu nennen ist, verwischen die Grenzen zwischen den unterschiedlichen Disziplinen. Es wird nicht der Aufwand für die Analyse, das Design, die Architektur, die Umsetzung, das Testen und weiteres getrennt geschätzt, sondern der Aufwand für die gesamte Story. Dadurch, dass der gesamte Aufwand in

einer Zahl ausgedrückt wird und die dadurch einhergehende Diskussion im Zuge der Schätzung, wird die Umsetzung der Story positiv beeinflusst (Cohn, Agile Estimating and Planning, 2006). Ähnliche Vorteile beschreibt (Leffingwell, 2011); er weist vor allem auf die positiven Auswirkungen der interdisziplinären Diskussion im Zuge des Prozesses der Schätzung hin. Des Weiteren merkt er an, dass im Zuge der Schätzung und der Einigung auf einen Wert Teammitglieder, die differierende Ansichten über die Größe einer Story haben, die Gründe für den jeweils gewählten Wert erläutern. Durch den Austausch der Meinungen und der Beweggründe für die Wahl einer Schätzung werden die implizierten Annahmen für alle sichtbar und die Schätzung aller Teammitglieder kann mit den neuen Erkenntnissen angepasst werden.

(Leffingwell, 2011) ergänzt, dass durch die Skala der Story Points, und dadurch, dass eine Story eine Größe haben muss, sodass sie innerhalb einer Iteration abgeschlossen werden kann, keine zu großen Stories (20, 40 oder 100 Story Points) zur Umsetzung gelangen. Stories mit solchen estimates sollten geteilt werden, da deren Umsetzung sonst ein zu großes Risiko birgt.

5.1.3 Von Story Points zur Zeitdauer

Um auf Basis der Anzahl von Story Points für ein Backlog Item oder für ein Release eines Produkts die Umsetzungsdauer zu ermitteln, wird in der Literatur das Verwenden der Velocity vorgeschlagen (Cohn, User Stories applied for Agile Software Development , 2004), (Cohn, Agile Estimating and Planning, 2006), (Pichler, 2010), (Rubin, 2012) u.a.

Die Velocity gibt an, welche Anzahl von Story Points ein Team innerhalb eines Sprints abarbeiten kann. Innerhalb eines Teams sollte die Velocity über einige Sprints hinweg wenig variieren, sodass die Velocity der Vergangenheit als Vorhersage für die Velocity in den folgenden Sprints verwendet werden kann.

Wenn ein Team eine Velocity von 30 Story Points per Sprint hat und ein Projekt in der Größe von insgesamt 300 Story Points umsetzen soll, wird das Team hierfür $300/30 = 10$ Sprints benötigen. Bei einer Sprintlänge von 2 Wochen ergibt sich damit eine Umsetzungsdauer von 20 Wochen.

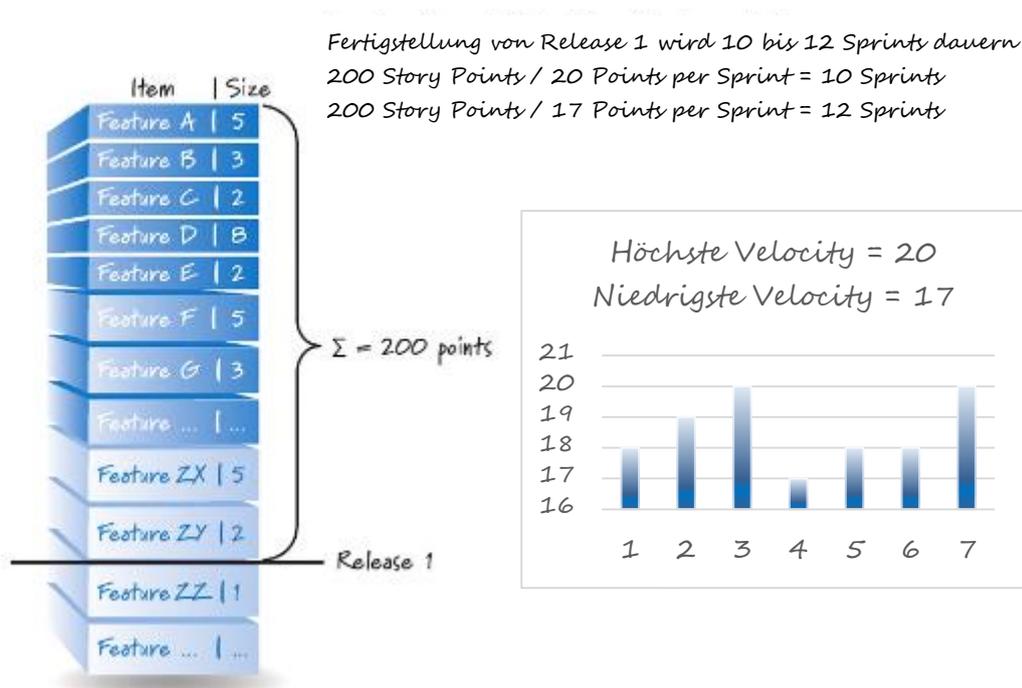


Abbildung 8: Berechnen und verwenden eines Bereichs für die Velocity, nach (Rubin, 2012)

Um Unsicherheiten in der Planung auszudrücken, schlagen (Rubin, 2012) u.a. vor, die Velocity als Bereich („Das Team erreicht üblicher Weise eine Velocity zwischen 28 und 33 Story Points per Sprint“) anzugeben. Ausgehend von diesem Bereich ergibt sich entweder ein Bereich für die Umsetzungsdauer oder, falls diese fixiert ist, ein Bereich, welche User Stories voraussichtlich umgesetzt werden können.

Empfehlungen, wie die Schätzung mit Story Points genauer gemacht werden kann, konnten im Rahmen dieser Arbeit in der Literatur nicht gefunden werden.

5.2 Schätzung mit idealen Tagen

Eine weitere Möglichkeit, die Größe von Stories zu schätzen, sind „ideale Tage“. Diese idealen Tage unterscheiden sich von Kalendertagen bzw. verstrichener Zeit. Wenn ideale Tage angegeben werden, unterliegen dieser Schätzung einige Annahmen:

- Der Entwickler arbeitet ausschließlich an dieser einen Story
- Wenn die Entwicklung beginnt, ist alles für die Umsetzung Notwendige vorhanden
- Der Entwickler wird nicht unterbrochen

Bei der Schätzung in idealen Tagen werden also keine Zeiten für Support, für Meetings, für Telefonanrufe, andere Projekte, Wechsel zwischen verschiedenen Aufgaben, Präsentation oder Anderes berücksichtigt. Durch diese Trennung unterscheiden sich Zeitangaben in idealen Tagen von Kalendertagen. Ideale Tage können also ähnlich wie Story Points als Größenangabe für eine Story verwendet werden. Wenn das Team Kenntnis von der Velocity (ausgedrückt als

Verhältnis zwischen Idealen Tagen und tatsächlichen Tagen innerhalb eines Sprints) des Teams besitzt, kann die Schätzung in idealen Tagen für die Planung der Umsetzungsdauer genutzt werden (Cohn, Agile Estimating and Planning, 2006).

5.2.1 Vorteile der Schätzung mit idealen Tagen

Die Schätzung in idealen Tagen hat im Vergleich zur Schätzung in Story Points laut (Leffingwell, 2011) einige Vorteile:

Die Schätzung auf Basis von Zeiträumen ist für Entwicklungsteams weniger abstrakt als die Schätzung mit Hilfe von Story Points. Schätzen in Zeiträumen ist etwas, das jeder Entwickler kennt. Ähnliches führt auch (Cohn, Agile Estimating and Planning, 2006) an, wenn er erläutert, dass es zu Beginn für das Team einfacher ist, in Idealen Tagen zu schätzen als mit Story Points zu schätzen.

Unabhängig von der verwendeten Methode für die Schätzung, besteht vor allem von Seiten des Managements immer der Anspruch, Terminpläne und Kosten für die Umsetzung von Stories und Projekten zu erhalten. Es ist also jedenfalls notwendig, die Einheit der Schätzung in Zeiträume umzurechnen.

Ideale Tage sind einfacher zu verstehen und zu erklären als Story Points. Es fällt oft auch leichter, Änderungen in der Teamzusammensetzung (Urlaube, Krankenstandstage, weniger oder zusätzliche Teammitglieder) zu berücksichtigen, wenn mit idealen Tagen geschätzt wird.

Dass das Erklären des Konzepts von idealen Tagen innerhalb und außerhalb des Teams einfacher ist, beschreibt auch (Cohn, Agile Estimating and Planning, 2006).

Nicht zuletzt sind ideale Tage, anders als Story Points, normalisiert. Ideale Tage können zwischen unterschiedlichen Teams verglichen werden, während die Größe eines Story Points von Team zu Team variiert. Story Points können daher üblicher Weise nicht zwischen Teams verglichen werden.

Einen Bericht darüber, dass das Schätzen auf Basis von Zeiträumen einfacher ist und zu weniger Missverständnissen führt als das Schätzen mit Story Points, gibt auch (Hohman, 2005). Allerdings räumt er auch ein, dass das Schätzen mit idealen Tagen seiner Erfahrung nach Probleme bei der Planung und Überwachung des Fortschritts mit sich bringen. Diese Probleme würden seiner Beobachtung nach mit Story Points ebenfalls bestehen.

5.3 Nachteile der vorangegangenen Methoden und Schätzen mit tatsächlicher Zeit

(Hohman, 2005) berichtet in seinem Paper *Estimating in Actual Time* von den Erfahrungen seines Teams mit unterschiedlichen Schätzmethoden. Betreffend die existierenden Methoden geht er auf das Schätzen mit idealer Zeit und auf das Schätzen mit Story Points ein.

Bei der Schätzung mit idealen Tagen nennt er als Vorteil, dass durch die verwendete Einheit (Tage) die Ungenauigkeit in der Schätzung ausgedrückt wird und nicht durch zu kleine Zeiteinheiten der Eindruck von nicht vorhandener Genauigkeit der Schätzung erzeugt wird. Durch die Betrachtung eines längeren Zeitraums werden zudem allfällige Ungenauigkeiten bei einzelnen Schätzwerten gemittelt. Probleme stellt er jedoch bei den folgenden Punkten fest:

- Die Vorhersage basierend auf vergangenen Sprints ist nur dann einfach möglich, wenn alle Sprints gleich lang sind, d.h. die gleiche Anzahl von Entwicklern im Team sind, keine Feiertage oder Urlaube zu berücksichtigen sind und kein Entwickler an anderen Projekten arbeitet.
- Der Aufwand für eine größere Anzahl von keinen Storys (z.B. Bugkorrekturen) kann nicht repräsentativ wiedergegeben werden, wenn die kleinste Einheit ein idealer Tag ist, die Umsetzung einer Story aber nur wenige Stunden oder noch kürzer dauert.
- Innerhalb eines Sprints ist eine Aussage darüber, wie weit die Umsetzung eines Tasks vorangeschritten ist, schwierig, wenn keine detaillierten Zeitaufzeichnungen geführt werden. Das begründet er damit, dass innerhalb eines Sprints keine Information darüber vorliegt, ob ein Mitarbeiter bis dahin an dem Task gearbeitet hat oder von anderen Tätigkeiten abgelenkt wurde (eine Multiplikation der Velocity mit den verstrichenen Kalendertagen führt also nicht zu einer validen Aussage über den tatsächlichen Fortschritt).

Die Schätzung mit Story Points adressiert laut Hohman einige Nachteile der Schätzung mit idealen Tagen, vor allem mögliche Verwirrungen zwischen „idealen Tagen“ und „tatsächlichen Tagen“, die beide als Zeiteinheiten in Tagen angegeben werden, aber tatsächlich unterschiedliche Zeiträume darstellen. Andere Probleme bleiben aber bestehen:

- Mit einer einzelnen Zahl, gleichgültig ob diese Zahl Story Points oder ideale Tage ausdrückt, ist es nicht möglich, eine Ungenauigkeit oder die Wahrscheinlichkeit, dass eine Story mehr Aufwand als geschätzt verursacht, auszudrücken.
- Ebenso ist es bei Verwendung von Story Points schwierig, auf den Fertigstellungsgrad eines Tasks oder einer Story innerhalb eines Sprints zu schließen.
- Bei Verwendung von idealen Tagen oder Story Points ist es schwierig, allfällige Überschreitungen des geplanten Aufwands zu ermitteln. Das ist leichter möglich, wenn die tatsächlich investierte Zeit pro Story täglich erfasst wird.

Als Antwort auf diese Probleme hat Hohmans Team die Schätzung in tatsächlicher (Implementierungs-) Zeit eingeführt. Ausgehend von einer Schätzung von z.B. einem idealen Tag (= acht Stunden Implementierungszeit) für eine Story und dem bisher gebuchten Aufwand für diese Story (z.B. vier Stunden) lässt sich leicht ermitteln, dass diese Story zur Hälfte fertiggestellt sein sollte. Allfällige Abweichungen davon sind relativ einfach zu erkennen und erlauben es frühzeitig Steuerungsmaßnahmen zu ergreifen.

Die Planung auf Basis der tatsächlichen Zeit basiert darauf, dass für einen Entwickler die Velocity (also jene Zeit, die er an Projekten arbeitet und nicht durch andere Aktivitäten

abgelenkt ist) ermittelt wird. Für jeden Entwickler wird die Zeit für Abwesenheiten und Urlaube abgezogen und des Weiteren berücksichtigt, in welchem Ausmaß der Entwickler für das betrachtete Projekt (d.h. nicht für andere Projekte) zur Verfügung steht. Somit ergibt sich nach Aufsummierung der Werte für alle Entwickler im Team eine Anzahl von zur Verfügung stehenden Stunden innerhalb einer Iteration.

Einen vergleichbaren Ansatz beschreiben (Coelho & Basu, 2012) für das Ermitteln eines Ausgangswertes für Vorhersagen zu Beginn eines Projekts. Sie nehmen dabei an, dass ein Teammitglied zwischen 55% und 80% seiner Zeit für Projektaktivitäten aufwenden kann. Aus der Sprintlänge, der Anzahl der Teammitglieder und der angenommenen Prozentsätze ergibt sich der verfügbare Stundenpool für Projektentwicklungen im betrachteten Sprint.

5.4 Zusammenfassung

Über die Vor- und Nachteile der Schätzung mit Story Points oder Idealer Zeit gibt es unterschiedliche Ansichten. Manche Autoren bevorzugen das Schätzen mit Story Points, andere bevorzugen das Schätzen mit Idealen Tagen. (Leffingwell, 2011) schlägt eine Kombination der beiden Schätzmethode („A hybrid Model“) vor, um die Vorteile der beiden Methoden zu kombinieren.

Den ersten beiden vorgestellten Schätzmethode ist gemein, dass aus der Schätzung in einem zusätzlichen Schritt die Dauer in tatsächlicher Zeit abgeleitet werden muss. Dazu wird meist das Konzept der „Velocity“ genutzt.

Um die Probleme zu vermeiden, die bei der Nutzung von idealer Zeit oder Story Points für die Schätzung auftreten, hat (Hohman, 2005) als Alternative die Schätzung mit tatsächlicher Zeit vorgestellt.

6 EINFLUSSFAKTOREN AUF DIE GENAUIGKEIT VON SCHÄTZUNGEN

Nach dem Überblick über Methoden für Schätzungen in traditionellen und in agilen Softwareprojekten in den vorangegangenen Kapiteln werden in Folge Einflussfaktoren vorgestellt, die Schätzungen negativ oder positiv beeinflussen.

6.1 Schätzungen mit Story Points und Expertenschätzungen in agilen Projekten

(Usman, Mendes, Weidt, & Britto, 2014) haben in einem systematischen Literatur-Review die Ergebnisse von 25 relevanten Studien, die sich mit der Schätzung oder Vorhersage des Aufwands in agilen Software-Projekten beschäftigen, untersucht und zusammengefasst. Die Ergebnisse des Reviews zeigen, dass die Schätzung mit Story Points zu den zwei am meisten verwendeten Methoden für die Schätzung in agilen Projekten gehört. Die Schätzung mit Story Points wird laut Usman et al. vor allem dann angewendet, wenn Anforderungen in Form von User Stories beschrieben sind, was eine gängige Praxis in agilen Entwicklungsprojekten ist. Die Autoren weisen aber auch darauf hin, dass Studien zur Genauigkeit von Schätzungen fehlen und dass weitere Forschungen notwendig sind: *„The lack of studies measuring prediction accuracy and presenting good accuracy represents a clear gap in this field.“*

Das Review von Usman et al. hat des Weiteren gezeigt, dass die vier am meisten genutzten Methoden für Schätzungen in der agilen Software-Entwicklung alle in einer gewissen Form auf dem Urteil von Experten basieren, sowie dass die meisten eingesetzten Techniken zu keinen guten Ergebnissen hinsichtlich der Genauigkeit der Schätzung geführt haben. Lediglich zwei Studien haben gute MRE-Werte (2 – 21%) bei der Schätzung mittels der Use Case Points Methode ausgewiesen, alle anderen Studien haben Fehler in den Schätzungen von über 25% (für MRE; MMRE; Mean BRE²) berichtet.

Somit kann also festgehalten werden, dass Überlegungen dazu, wie die Genauigkeit von Schätzungen in agilen Software-Projekten (basierend auf Story Points) erhöht werden kann, derzeit weitgehend fehlen, für die Arbeit von Softwareteams und die wissenschaftliche Forschung aber notwendig und interessant sind.

² MRE: Magnitude of Relative Error, (Tatsächlicher Aufwand – Geschätzter Aufwand) / Tatsächlicher Aufwand

MMRE: Mean Magnitude of Relative Error

Mean BRE: Mean Balanced Relative Error: gleicht Über- und Unterschätzung des Aufwands besser aus als MRE (Usman, Mendes, Weidt, & Britto, 2014), (Software development effort estimation, 2016)

6.2 Faktoren, die die Genauigkeit negativ beeinflussen

Wenn über die Verbesserung der Genauigkeit von Schätzungen gesprochen wird, rücken auch jene Faktoren in den Fokus, die die Genauigkeit negativ beeinflussen. Ein wesentlicher Faktor dabei ist die Änderung von Anforderungen.

6.2.1 Änderungen von Anforderungen

Ein Faktor für die Genauigkeit von Schätzungen oder vielmehr eine Gefahr für die Ungenauigkeit von Schätzungen ist der *scope creep*. *Scope creep ist das Hinzufügen von zusätzlicher Funktionalität, die entweder nicht speziell definiert wurde oder zum Zeitpunkt der Schätzung noch nicht identifiziert wurde* (Hill, 2011). *Scope creep* beeinflusst dabei insbesondere den Gesamtaufwand in einem Projekt, kann aber auch Einfluss auf die Schätzung eines einzelnen Arbeitspaketes haben, wenn innerhalb dieses zusätzliche Funktionalität hinzugefügt werden muss.

Gleiches berichtet auch (Hohman, 2005), indem er und sein Team festgestellt haben, dass die häufigsten Ursachen für das Überschreiten des Aufwands das Vergessen von Tätigkeiten, die für die Fertigstellung eines Tasks notwendig waren und mangelnde Kommunikation mit dem Auftraggeber während der Umsetzung waren. Letzteres führte dazu, dass Funktionalität umgesetzt wurde, die in dieser Form nicht vom Auftraggeber gewünscht war und daher umfangreiche Nacharbeiten notwendig wurden.

(Basten, 2012) hingegen führt dazu aus, dass Änderungen von Anforderungen dazu führen müssten, dass die Planung von Projekten aktualisiert wird. Er begründet dies damit, dass sich eine Schätzung und eine Planung immer auf einen bestimmten Projektinhalt bezieht. Wenn sich der Projektinhalt ändert, müsse die Aufwandsschätzung und die Planung aktualisiert werden. In einer unter 52 Projektmanagern durchgeführten Umfrage hat Basten jedoch ermittelt, dass dieses Vorgehen in der Praxis in 50% der Fälle nicht angewendet wird, d.h. dass in der Hälfte der betrachteten Software-Projekte keine Änderung des Projektplans auf Grund von geänderten Anforderungen (Change Requests) vorgenommen wurde.

| Maßnahmen bei Change Requests | Häufigkeit der Nennung |
|---|------------------------|
| Es traten keine Change Requests auf | 12% |
| Es traten Change Requests auf, eine erneute Schätzung wurde nicht durchgeführt | 27% |
| Es traten Change Requests auf, eine erneute Schätzung wurde durchgeführt, der Projektplan wurde aber nicht aktualisiert | 23% |
| Auf Grund von Change Requests wurden Schätzungen und der Projektplan aktualisiert | 38% |

Tabelle 4: Änderungen in Anforderungen und der Umgang damit in der Projektplanung, aus (Basten, 2012)

Zusammengefasst ergibt sich somit, dass eine Änderung der Anforderungen durch eine neuerliche Schätzung und eine Aktualisierung der Planung für ein Software-Entwicklungsprojekt abgebildet werden muss, andernfalls diese Änderung einen negativen Einfluss auf die (wahrgenommene) Genauigkeit der Schätzung hat.

6.2.2 Weitere negative Einflussfaktoren

Wie schon früher in dieser Arbeit beschrieben, nimmt die Genauigkeit der Schätzung mit dem Abstand zwischen Schätzung und Umsetzung ab. Je früher in einem Projekt eine Schätzung durchgeführt wird, desto ungenauer ist sie (vgl. Cone of Uncertainty). (Gloski, 2013) führt in diesem Zusammenhang mehrere Punkte an, die zu Unsicherheiten bei der Schätzung des Product Backlogs führen:

- User Stories sind zu Beginn des Projekts noch nicht klar beschrieben.
- User Stories können später ins Backlog aufgenommen, anders priorisiert oder gelöscht werden.
- Die Umsetzung der User Stories erfolgt möglicher Weise nicht durch die Entwickler, die die Schätzung ausgeführt haben. Dadurch geht die Verantwortung für das Einhalten der Schätzung verloren.
- Wenn mit Story Points geschätzt wird, erfolgt die Ermittlung der Zeitdauer für die Umsetzung mittels der Velocity. Diese kann und wird sich im Verlauf des Projekts jedoch ändern, sei es durch geänderte Erfahrung des Teams, andere eingesetzte Technologien oder eine geänderte Zusammensetzung des Entwicklungsteams.

6.3 Faktoren, die die Genauigkeit positiv beeinflussen

6.3.1 Methoden zur Reduzierung von situationsbedingten und persönlichen Verzerrungen

Wie bereits in Kapitel 3 angesprochen, werden Schätzungen von situationsbedingten Faktoren beeinflusst. Eine rationale Schätzung über den Aufwand für ein Softwareprojekt führt mitunter nicht zu einem gewünschten oder erwarteten Fertigstellungstermin, was z.B. eine Präsentation auf einer Messe unmöglich machen würde, oder zu einem verlorenen Auftrag führen könnte. Wenn diese Einflussfaktoren den Schätzern bekannt sind, kann die Schätzung durch die Erwartung der involvierten Personen beeinflusst bzw. verzerrt werden (McConnell, 2006), (Jorgensen, 2004). Andere Verzerrungen beruhen auf persönlichen Faktoren, z.B. dem Wunsch nach positivem Feedback.

(Jorgensen, 2004) schlägt zur Reduzierung von situationsbedingten und persönlichen Verzerrungen sechs Prinzipien für Schätzungen vor. Die Vorschläge von Jorgensen basieren auf Erkenntnissen aus Beobachtungen in Studien und auf Empfehlungen in der Literatur.

Evaluierung der Genauigkeit von Schätzungen, ohne hohen Druck durch die Evaluierung auszuüben: Jorgensen argumentiert, dass verschiedene Studien zeigen, dass in Situationen, in denen sich die schätzenden Personen persönlich verantwortlich für die Genauigkeit von Schätzungen fühlen, wenn sie finanzielle Anreize für genaue Schätzungen erwarten oder wenn das Ergebnis der Schätzung als sehr wichtig wahrgenommen wird, die Genauigkeit der Schätzung abnimmt. Untermauert wird diese Beobachtung durch andere Studien, die zeigen, dass Schätzungen dann ungenauer werden, wenn diese in hohem Maße evaluiert werden. Die zitierten Studien beziehen sich nicht im speziellen auf die Schätzung von Software-Entwicklungsprojekten, sondern allgemein auf menschliche Beurteilung; Studien aus dem Bereich der Software-Schätzung hingegen zeigen, dass die Evaluierung von Schätzungen zu einer besseren Genauigkeit führt. Laut Jorgensen stellen diese Beobachtungen keinen Widerspruch dar.

Im Bereich der Schätzung von Software-Entwicklungsprojekten weist Jorgensen auf zwei Effekte hin: Die „*self-fulfilling prophecy*“: Wenn von den schätzenden Experten ein hoher Evaluierungs-Druck wahrgenommen wird, kann das zu vorsichtigen Schätzungen führen und danach dazu, dass der Aufwand während der Umsetzung so erhöht wird, dass die Schätzung zutrifft. Ein zweiter, positiver, Effekt ist, dass Evaluierung zu selbstkritischem Denken der beteiligten Personen führt und somit zu besseren Schätzungen.

Durch eine maßvolle Evaluierung der Schätzungen können die positiven Effekte der Evaluierung gefördert und die negativen Effekte minimiert werden. Checklisten und ein definierter Prozess für das Schätzen können das selbstkritische Denken und somit die Genauigkeit von Schätzungen fördern.

Vermeiden von widersprüchlichen Zielen: Der Prozess des Schätzens kann durch andere Ziele als die bestmögliche Genauigkeit beeinflusst werden. Das sind Differenzen zwischen dem gebotenen Aufwand, dem wahrscheinlichsten Aufwand und dem geplanten Aufwand sowie zwischen Realismus und Wunschdenken.

Die Unterscheidung zwischen den ersten drei Werten (gebotener Preis, wahrscheinlicher Aufwand und Planung) wird häufig nicht getroffen, und die Schätzung somit verwischt. Da diese drei Werte aber unterschiedliche Ziele haben – der gebotene Preis muss niedrig genug sein, um einen Auftrag zu erhalten, der geplante Aufwand soll ein erfolgreiches Projekt ermöglichen und der wahrscheinlichste Aufwand soll den realistischsten Wert für den Aufwand widerspiegeln – führt das zu geringerer Genauigkeit der Schätzung.

Ein weiterer belegbarer Einfluss auf die Genauigkeit der Schätzung ist, ob die eigene Arbeit oder die Arbeit von anderen geschätzt wird. Studien aus unterschiedlichsten Bereichen zeigen, dass die eigene Leistung häufig zu gut eingeschätzt wird. Um diesen Effekt zu reduzieren, ist es notwendig, dass nicht die eigene Arbeit geschätzt wird, sondern die von anderen Personen, d.h. dass nicht derjenige, der ein Arbeitspaket erledigen wird, die Schätzung dafür erstellt. Dieser Effekt, ebenso wie der Einfluss des Wunschdenkens ist dann umso höher, je größer das Interesse einer Person am Ausgang des Projekts hat.

Die Auswahl der Person, die eine Schätzung erstellt, kann somit die Genauigkeit einer Schätzung erhöhen. Für die Auswahl sind nicht nur fachliche Kriterien, sondern auch

situationsbedingte Kriterien wie das persönliche Interesse oder das Maß der Beteiligung in einem Projekt zu berücksichtigen.

Anpassung und Prüfung der Schätzungen durch die Schätzenden: Studien, die nicht aus dem Bereich der Software-Entwicklung stammen, legen nahe, dass die Prüfung und Evaluierung der Schätzung durch die ausführenden Personen eine Verbesserung der Genauigkeit bewirken können. Diese Prüfungen haben auch weitere positive Effekte auf die Schätzung:

- Sie führen zu einem analytischeren Prozess für die Schätzung und reduzieren das Risiko, zu einfache Strategien für die Schätzung zu verwenden
- Sie erhöhen die Vertrauenswürdigkeit der Schätzung
- Sie verbessern die Kompensation für fehlende Information (Jorgensen, 2004)

Ein Experiment von Jorgensen mit einer kleinen Gruppe von Software-Experten zeigte, dass die Überprüfung der eigenen Schätzungen zu einer Anpassung derselben führte (im Schnitt eine Erhöhung des Wertes um 10%). In dem Experiment wurden die Experten aufgefordert, nach der Schätzung Gründe anzugeben, die möglicherweise zu einer falschen Schätzung geführt haben könnten. Die 13 Experten listeten zwischen 2 und 8 Gründen auf und passten danach ihre Schätzungen an. Die Streuung der Anpassungen (fünf Personen erhöhten die Schätzung nach der Überprüfung um maximal 25%, vier Personen um mehr als 25% und vier Personen reduzierten die Schätzung) und die geringe Größe (10% Erhöhung im Schnitt) lassen laut Jorgensen keinen großen Effekt der Anpassung und Prüfung erwarten.

Da der Aufwand hierfür jedoch gering ist, empfiehlt Jorgensen trotzdem, diese Praktik anzuwenden.

Vermeiden von irrelevanter und unzuverlässiger Information: Verschiedene Studien belegen, dass irrelevante oder unzuverlässige Informationen Schätzungen von Experten beeinflussen. Dies ist sogar dann der Fall, wenn die Schätzenden wissen, dass Information irrelevant ist. So konnte unter anderem festgestellt werden, dass Schätzungen dadurch beeinflusst werden, dass den Schätzenden die Erwartungen der Kunden hinsichtlich der Kosten für das Projekt bekannt sind.

Es ist nicht ausreichend, dass die schätzenden Experten über irrelevante Informationen informiert werden. Vielmehr muss es vermieden werden, irrelevante oder unzuverlässige Informationen weiterzugeben. Jorgensen empfiehlt sogar, im Zweifelsfall die Schätzung von einem anderen Experten, der nicht durch irrelevantes Wissen vorbelastet ist, durchführen zu lassen.

Verwenden von dokumentierten Daten aus vorangegangenen Entwicklungsaufgaben: Wenn Personen eine Schätzung durch Vergleich erstellen, tendieren sie dazu, den Vergleich mit einem zuletzt ausgeführten, ähnlichen Arbeitspaket zu ziehen. Dabei wird der Aufwand von länger zurückliegenden, ähnlichen Arbeitspaketen außer Acht gelassen. Die Schätzung wird somit in hohem Maße auf Basis der persönlichen Erinnerung erstellt. Durch Verwendung von

historischen Daten kann die möglicherweise große Verzerrung durch die persönliche Erinnerung reduziert werden.

Das Verwenden von historischen Daten hat laut (Jorgensen, 2004) vergleichbare Effekte wie das Nutzen von Modellen, um ungenaue Schätzungen und eine Verzerrung durch persönliche Einflüsse zu vermeiden.

6.3.2 Methoden, um den Schätzprozess zu unterstützen

Laut (Jorgensen, 2004) gibt es viele Möglichkeiten, den Prozess der Expertenschätzung zu unterstützen. Prozessrichtlinien leiten die Arbeit des Schätzenden. Einige Richtlinien für die Verbesserung des Prozesses sind im Folgenden beschrieben.

Kombination von Top-Down und Bottom-Up Schätzungen: Verschiedene Studien belegen laut (Jorgensen, 2004), dass eine Bottom-Up Schätzung keine besseren Ergebnisse bringt als eine Top-Down-Schätzung. Bei der Bottom-Up-Schätzung werden die zu schätzenden Aufgaben nach verschiedenen Strategien zerlegt, während bei einem Top-Down Ansatz meistens mittels Analogie geschätzt wird. Bottom-Up Schätzungen führen öfters zu überoptimistischen Schätzungen, andererseits sind viele Probleme zu komplex, um sie als gesamtes zu verstehen und zu schätzen.

Daher empfiehlt Jorgensen eine unabhängige Kombination der beiden Ansätze, damit die Vorteile beider Strategien vereinigt werden. Einer der Vorteile der Top-Down (vergleichenden) Schätzung ist es, dass sie unempfindlicher gegen vergessene Aktivitäten oder unvorhergesehene Ereignisse ist, während der Bottom Up Ansatz dabei hilft, den Projektinhalt, die Umsetzung und die Planung zu verstehen.

Es ist wichtig, die Schätzungen unabhängig voneinander auszuführen, da es sonst zur gegenseitigen Beeinflussung der Schätzungen kommen kann. Wenn die beiden unabhängig ausgeführten Schätzungen zu differierenden Ergebnissen führen, können die Schätzungen durch Einholen von zusätzlichen Informationen oder der Beiziehung zusätzlicher Experten verbessert werden.

(Leung & Fan, 2002) bestätigen in ihrem Artikel „Software Cost Estimation“ die Vorteile und Risiken des Top-Down Ansatzes. Der Vergleich mit Daten von historischen Projekten (Analogie) bietet – eine hohe Übereinstimmung der Einschränkungen, der Umgebung und der Funktionalität vorausgesetzt – Vorteile für die Ermittlung der gesamten Kosten respektive des gesamten Aufwands eines Softwareprojekts. Wenn diese Übereinstimmung jedoch nicht gegeben ist, ist der Vergleich mit historischen Daten nicht zielführend und muss daher validiert werden.

(Trendowicz & Ross, 2014) schlagen ebenfalls eine Kombination des Top-Down und Bottom-Up Ansatzes („*Consider using top-down estimation strategy for cross-checking bottom-up estimates (and vice versa).*“) vor, da sich beide Methoden durch ihre jeweiligen Vor- und Nachteile gegenseitig ergänzen.

Die Kombination der Schätzung mit dem Bottom-Up Ansatz und dem Top-Down Ansatz hilft, die Vorteile der jeweiligen Methode zu nutzen und die Nachteile aufzuwiegen. Daher empfiehlt auch Jorgensen die Kombination der beiden Methoden.

Verwenden von Checklisten für Schätzungen: Laut (Jorgensen, 2004) besteht wenig Zweifel, dass Checklisten ein wichtiges Mittel sind, um die Schätzung von Experten zu verbessern. Zumindest vier Beobachtungen aus unterschiedlichen Studien bestätigen dies.

| Beobachtung | Nutzen von Checklisten |
|--|---|
| Experten vergessen leicht Aktivitäten und unterschätzen den Aufwand für die Lösung von unerwarteten Ereignissen. | Checkliste helfen, wichtige Variablen und Möglichkeiten zu berücksichtigen. |
| Schätzungen von Experten sind uneinheitlich, insbesondere bei steigender Unsicherheit. | Checklisten helfen die Konsistenz und damit die Genauigkeit von Schätzungen zu erhöhen. |
| Personen neigen dazu, Schätzstrategien, die minimalen Aufwand erfordern, zu bevorzugen. | Checklisten drängen Personen dazu, eine angemessene Schätzstrategie zu nutzen. |
| Personen neigen dazu, nur präsentierte Optionen zu berücksichtigen („Aus den Augen, aus dem Sinn“) | Checklisten helfen, wahrscheinlichere Ergebnisse zu generieren. |

Tabelle 5: Beobachtungen zu Expertenschätzungen und der Nutzen von Checklisten, basierend auf (Jorgensen, 2004)

Eine Empfehlung, die mit dem Verwenden von Checklisten in Zusammenhang steht, gibt auch Mike Cohn in seinem Blog bei Mountain Goat Software: „*Consider Everything in the Definition of Done*“ (Cohn, What are Story Points, 2016). Die Schätzung des Aufwands für ein Arbeitspaket (in Form von Story Points) muss alles enthalten, das notwendig ist, um ein Arbeitspaket im Sinne der „Definition of Done“ fertigzustellen. Wenn in einer „Definition of Done“ festgelegt ist, dass eine Funktion, die in einem Arbeitspaket umgesetzt wird, automatisiert getestet sein muss, muss der Aufwand für die Testerstellung in der Schätzung berücksichtigt werden. Da die „Definition of Done“ eine Auflistung aller Tätigkeiten bis zur Fertigstellung eines Arbeitspakets enthält (Pichler, 2010), kann diese auch als Checkliste verstanden werden.

Checklisten helfen den Experten, wesentliche Punkte für die Schätzung zu berücksichtigen und die Schätzung einem einheitlichen Review-Prozess zu unterziehen.

Kombination von Schätzungen von unterschiedlichen Experten und Methoden: Darüber, dass die Kombination der Schätzungen von unterschiedlichen Experten Vorteile gegenüber einer einzelnen Schätzung bringt, besteht wenig Zweifel. Daher stellt sich nicht die Frage, ob Schätzungen kombiniert werden sollen, sondern wie die Mittelung erfolgen soll (Jorgensen,

2004). Jorgensen zitiert Hogarth's Modell, nach dem der Nutzen der Mittelung von unterschiedlichen Schätzungen von verschiedenen Faktoren abhängt. Diese Faktoren sind: Die Anzahl der Experten, die erwartete Schätzgenauigkeit der Experten, der Grad der Verzerrung zwischen den Experten, die Korrelation zwischen den unterschiedlichen Schätzungen und die Auswirkung der Art der Mittelung.

- Um die Vorteile der Mittelung mehrere Expertenschätzungen zu nutzen, sei eine relativ geringe Anzahl von Experten notwendig. Es handelt sich dabei um 3-5 Experten mit unterschiedlichem Hintergrund.
- Über die Schätzgenauigkeit der Experten sind üblicher Weise keine gesicherten Informationen vorhanden; möglicher Weise aber informelle Informationen, die dem Projektleiter vorliegen. Diese Information kann und soll genutzt werden, um sicherzustellen, dass die Schätzungen eines Experten nicht systematisch in eine Richtung verzerrt sind.
- Um die Vorteile der Mittelung nutzen zu können, ist es wichtig, dass die Schätzungen der Experten nicht stark korrelieren.
- Die Art der Kombination der Schätzungen scheint laut (Jorgensen, 2004) keinen wesentlichen Einfluss auf die Genauigkeit des Ergebnisses zu haben. Während manche Studien zeigen, dass das Mitteln der Schätzungen zu den besten Ergebnissen führen, zeigen andere Studien, dass ein gruppenbasierter Prozess bessere Ergebnisse bringt. Die Evidenz, dass diese Art der Kombination besser performt, ist jedoch nicht stark.

Wichtig für die Verbesserung der Genauigkeit der Schätzungen ist somit grundsätzlich die Kombination von unterschiedlichen Schätzungen; in welcher Art die Kombination erfolgt, ist zweitrangig. Die Schätzungen sollten von Experten mit unterschiedlichem Hintergrund und gutem Schätzvermögen stammen.

6.3.3 Feedback für die Schätzung und Trainingsmöglichkeiten zur Verfügung stellen

(Jorgensen, 2004) stellt fest, dass es schwer ist, die Kompetenz für das Schätzen zu erhöhen, ohne Feedback und Training zu erhalten. Verschiedene Studien belegen, dass es keine Verbindung zwischen der Erfahrung der Experten und der empirischen Genauigkeit der Schätzungen gibt.

Feedback über die Schätzung geben: Um die Genauigkeit der Schätzungen positiv zu beeinflussen, reicht es jedoch nicht aus, Feedback über die Schätzung und den tatsächlichen Aufwand zu geben. Vielmehr ist es notwendig, Feedback darüber zu geben, welche Tätigkeiten und welche Variablen den Aufwand für ein Arbeitspaket beeinflusst haben.

Gleiches führt auch (Basten, 2012) aus. Er beschreibt, dass eine kontinuierliche Überprüfung von Aufwandsschätzungen für eine realistische Einschätzung und das Sammeln von Erfahrungswerten notwendig ist. Es sei jedoch von entscheidender Bedeutung, neben dem geschätzten und tatsächlich benötigten Aufwand auch die Ursachen von Abweichungen zu

identifizieren und zu protokollieren. Nur in diesem Fall können aus Erfahrungswerten Analogien zu vorigen Projekten gezogen werden und somit Schätzungen für künftige Projekte verbessert werden.

Jedoch ist der Einfluss von Feedback nicht so groß, wie man gerne erwarten würde; dennoch empfiehlt Jorgensen, dass Unternehmen ihr Bestes geben sollten, um Feedback über den Schätzprozess zu geben.

Trainingsmöglichkeiten für Schätzer: In Softwareprojekten stehen in aller Regel zu wenige Informationen zur Verfügung, um zuverlässige Rückschlüsse auf die Ursachen von Veränderungen gegenüber der Planung ziehen zu können (Jorgensen, 2004). Jorgensen schlägt daher vor, in Trainings-Sessions abgeschlossene Projekte auf Basis der zum Zeitpunkt der ursprünglichen Schätzung bekannten Informationen zu schätzen. Dieses Vorgehen bringt mehrere Vorteile, darunter die Möglichkeit, sofort Feedback zu geben; die Auswirkung der (Nicht-)Verwendung von Hilfsmitteln für den Schätzprozess wie den Einsatz von Checklisten zu überprüfen; die Ursache für vergessene Aktivitäten zu reviewen; die Tendenz zu übermäßigem Optimismus zu untersuchen und die Validität der eigenen Schätzung zu überprüfen.

Dieser Ansatz kann zwar dazu führen, die Fähigkeiten der Experten im Bereich der Schätzung zu erhöhen, es handelt sich dabei aber nicht um einen Faktor, der direkten Einfluss auf die Genauigkeit der Schätzung eines einzelnen Arbeitspakets hat.

6.3.4 Berücksichtigung der (Un-)Genauigkeit der Schätzung

(Hohman, 2005) schlägt in Anlehnung an DeMarco und Listers „Walzing with Bears“ vor, Schätzungen nicht als einzelne Zahl, sondern als „dreiseitige“ Schätzung mit Werten für early/likely/late zu erfassen. Dadurch sollen überoptimistische Schätzungen vermieden werden, da tatsächlich eine einzelne Schätzung sehr oft jenen Wert darstellt, der im besten Fall erreicht werden kann; d.h. jener Wert ist, für den die Wahrscheinlichkeit der Erreichung nicht mehr Null ist. Wenn jedoch mehrere Werte für eine Schätzung abgegeben werden, kann vermieden werden, dass der „likely“-Wert, also der erwartete Aufwand, die optimistischste Schätzung ist.

(Jorgensen, 2004) weist darauf hin, dass ein einzelner Wert für eine Schätzung oftmals eine Vermischung von unterschiedlichen Werten ist. Da sehr häufig keine Unterscheidung zwischen dem gebotenen Wert, dem geplanten Wert und dem wahrscheinlichsten Wert getroffen wird, werden diese Werte implizit vermischt und somit die Genauigkeit der Schätzung reduziert. Es liegt die Vermutung nahe, dass eine explizite Trennung dieser Werte die Vermischung reduzieren kann; laut Jorgensen ist es jedenfalls gute Praxis, Zielkonflikte in der Schätzung zu vermeiden.

(Jorgensen, 2004) empfiehlt, die Unsicherheit einer Schätzung durch die Angabe eines Vorhersage-Intervalls zu beurteilen. Bei der Angabe von Konfidenz-Intervallen in Experten-Schätzungen besteht allerdings die Gefahr, dass die Genauigkeit zu hoch angegeben, respektive das Konfidenzintervall zu klein angegeben wird. Um dieser Gefahr zu begegnen, schlägt Jorgensen vor, für den Aufwand statt des minimalen und maximalen Werts für einen

vorgegebenen Konfidenzwert (z.B. 90%) anzugeben, einen Konfidenzwert für ein fixes Intervall zu ermitteln:

- Zu Beginn wird der wahrscheinlichste Wert für den Aufwand ermittelt.
- Dann wird der minimale und maximale Aufwand mit einem festen Verhältnis ermittelt, z.B. 50% des erwarteten Aufwands als minimaler Wert, 200% des erwarteten Aufwands als maximaler Wert für den Aufwand.
- Danach wird das Konfidenzlevel ermittelt, d.h. die Wahrscheinlichkeit, dass der tatsächliche Aufwand im angegebenen Intervall liegt.

Dadurch, dass das Intervall für den Aufwand mit festen Werten angegeben wird, wird unter anderem die Gefahr der zu hohen Zuversicht in die eigene Schätzung reduziert; Jorgensen konnte in einer Studie belegen, dass die Erfolgsrate dieses Vorgehens höher ist als bei der traditionellen Art der Angabe der Ungenauigkeit von Schätzungen.

6.3.5 Planning Poker

Beim Planning Poker wird vor der Schätzung die zu schätzende Story dem gesamten Entwicklungsteam vorgestellt. Wenn Unklarheiten über die Story bestehen, werden diese in einer Diskussion geklärt. Danach notiert jeder Entwickler seine Schätzung auf einer Karte und erst wenn alle Entwickler einen Wert notiert haben, decken sie Ihre Schätzung auf. Im Falle einer Übereinstimmung, ist das Schätzen dieser Story abgeschlossen und die nächste Story kann geschätzt werden.

Wenn Uneinigkeit über die Größe einer Story herrscht, begründen die Entwickler Ihre Schätzung. Im optimalen Fall wird nach kurzer Zeit Einigkeit über die Größe einer Story hergestellt. Der Vorteil dieser Art zu schätzen ist, dass alle Entwickler in den Prozess involviert sind, ihre Schätzung begründen müssen, Diskussion angeregt wird und alle Entwickler im Laufe der Zeit Erfahrung gewinnen (Grenning, 2002).

Im Verlauf der Schätzungen entsteht für jeden Entwickler ein Satz von Karten, aus dem dann nur noch die entsprechende Karte gezogen und „ausgespielt“ werden muss. Grenning schlägt als gültige Werte für die Schätzung (in einheitenlosen Zahlen oder Idealen Programmier Tagen) die Reihe 1, 2, 3, 5, 7, 10 und ∞ vor. Ebenso hält er 4, 6, 8, oder 9 für gültige Werte, die durch Kombination von zwei Karten angezeigt werden können. Er weist aber darauf hin, dass durch diese zusätzlichen Werte kein merkbarer Nutzen entstehen wird. (Cohn, Agile Estimating and Planning, 2006) hingegen schlägt die für Story Points gängige Reihe 0, 1, 2, 3, 5, 8, 13, 20, 40 und 100 vor. Cohn schlägt des Weiteren vor, dass bei differierenden Schätzungen nicht alle Beteiligten, sondern lediglich jene, die die höchste und niedrigste Schätzung abgegeben haben, ihre Schätzung erklären. Danach überdenken alle Entwickler ihre Schätzung und „spielen“ in einer weiteren Runde ihre Schätzungen aus. In sehr vielen Fällen wird laut Cohn so schon in der zweiten Runde Konsens über die Größe einer Story erzielt.

Planning Poker vereinigt für die Schätzung mehrere Elemente: Expertenmeinung, Schätzen mittels Analogie und Disaggregation (Usman, Mendes, Weidt, & Britto, 2014).

Wie der Vergleich des Vorgehens beim Planning Poker mit den zuvor vorgestellten Vorschlägen für die Verbesserung von Schätzungen zeigt, führt Planning Poker zur Nutzung einiger Vorschläge aus der Literatur:

Die Schätzungen von mehreren Experten werden kombiniert, das strukturierte Vorgehen hilft dabei, keine Tätigkeiten zur Fertigstellung einer Aufgabe zu vergessen (vgl. den Einsatz von Checklisten) und der Bottom Up- (Disaggregation) und Top Down- (gesamte Aufgabe mit einer Zahl schätzen) Ansatz werden kombiniert.

(Osman & Musa, 2016) zitieren mehrere Quellen, die darauf hindeuten, dass das Schätzen mittels Planning Poker die Genauigkeit der Schätzungen von agilen Teams positiv beeinflusst. Insbesondere hilft Planning Poker dabei, überoptimistische Schätzungen von agilen Teams zu vermeiden. Die Ergebnisse der Schätzungen mittels Planning Poker sind weniger überoptimistisch als mechanisch gemittelte Schätzungen und in einigen Fällen auch genauer.

Entsprechende Ergebnisse konnten (Gandomani, Wei, & Binhamid, 2014) in einer empirischen Studie beobachten. Sie stellten fest, dass in den beiden in der Studie untersuchten Unternehmen die Expertenschätzungen nur in 5 von 32 Fällen zu hoch, in den 27 anderen Fällen jedoch zu niedrig waren. Wenn in den Unternehmen für das Schätzen die Wideband Delphi³-Methode oder Planning Poker (beides Methoden, die auf der Zusammenarbeit von Experten basieren) verwendet wurden, waren 19 Schätzungen zu hoch, 13 zu niedrig. Es zeigte sich bei den untersuchten Schätzungen also eine deutliche geringere Tendenz für zu niedrige Schätzungen.

(Usman, Mendes, Weidt, & Britto, 2014) zitieren eine Studie von N.C. Haugen, die zeigt, dass mit Planning Poker dann genauere Schätzungen als mit einer unstrukturierten Vorgangsweise erreicht werden, wenn das Team Erfahrungen mit ähnlichen Projekten hat. Des Weiteren zeigen lt. Usman et al. andere Studien von Haugen und Molokken-Ostvold, dass mit Schätzmethoden, die auf Konsensbildung innerhalb eines Expertenteams basieren, zu weniger optimistischen und genaueren Schätzungen führen, als die statistische Mittelung von Einzelschätzungen. Auch eine weitere Studie zeigt laut Usman et al., dass Planning Poker die optimistische Verzerrung reduziert und die Genauigkeit verbessert wird, wenn die Methode von erfahrenen Praktikern angewendet wird.

6.3.6 Stufenweise Planung

Bedingt dadurch, dass in der agilen Software-Entwicklung iterativ und inkrementell vorgegangen wird, ist es notwendig, dass auch Schätzungen und Planungen stufenweise durchgeführt werden. Im Allgemeinen erfolgt dies durch eine dreistufige Planung: die Planung

³ Bei der Wideband Delphi Methode (ursprünglich beschrieben in (Boehm, Software Engineering Economics, 1981)) werden in einem sechsstufigen Prozess Schätzungen mehrerer Experten kombiniert, wobei in einem Estimation Meeting die Schätzungen der Experten verglichen und diskutiert werden. Ziel der Diskussion ist das Erreichen eines Konsenses über die Schätzung. Zusätzlich zur Schätzung wird durch die allfällige Varianz der Schätzungen auch die (Un-)genauigkeit der Schätzung wiedergespiegelt (Wieggers, 2000).

für die Version, die Planung für die folgende Iteration (in Scrum „Sprint“) und die tägliche Planung (Usman, Mendes, Weidt, & Britto, 2014), (Steyer, 2010).

Während das Planen und Schätzen der täglichen Aufgaben und eines Sprints eher genau erfolgen kann, ist das für den Gesamtaufwand eines Projekts weniger leicht möglich. (Gloski, 2013) führt dazu aus, dass eine Schätzung für einen Sprint viele Voraussetzungen erfüllt, die für eine genaue Schätzung erfüllt sein müssen. Dazu zählt, dass Anforderungen klar und verständlich sind, dass die Schätzung von denjenigen Personen ausgeführt wird, die für die Umsetzung verantwortlich sind, dass im Falle differierender Schätzungen der Konsens für eine gemeinsame Schätzung gesucht wird und dass die Anforderungen zeitnah umgesetzt werden. Eine Schätzung des Gesamtaufwands sei, im Gegensatz zur Schätzung für einen Sprint, jedoch nicht mit vergleichbarer Genauigkeit möglich. Gründe dafür sind, dass zu Beginn der Entwicklung der Gesamtinhalt des Projekts noch nicht bekannt ist, sich die Velocity, auf deren Basis eine Planung erfolgt, sich im Laufe des Projekts ändern kann, die Backlog Items noch nicht in der notwendigen Genauigkeit beschrieben sind, der Zeitpunkt für die Umsetzung von Backlog Items noch nicht bekannt ist, die Backlog Items nicht von den Entwicklern umgesetzt werden, die die Schätzung durchgeführt haben.

Somit lassen sich die Ausführungen von (Gloski, 2013) dahingehend interpretieren, dass eine stufenweise Planung dazu führt, die Genauigkeit der Schätzung des Gesamtaufwands zu verbessern.

7 ZUSAMMENFASSUNG TEIL 1

Die Schätzung und Planung von Softwareprojekten ist für jegliches Projekt, unabhängig davon, ob es sich um ein internes oder ein extern verkaufte Projekt handelt, ein wesentlicher Faktor für den Projekterfolg. Dies ist auch unabhängig davon, mit welchem Vorgehensmodell das Projekt umgesetzt wird.

Die Schätzung des Aufwands von einzelnen Arbeitspaketen und des gesamten Projekts ist ein wesentlicher Faktor, um eine zuverlässige Planung zu erstellen.

Die Genauigkeit der Schätzung ist gemäß dem Cone of Uncertainty umso geringer, je früher im Projekt diese Schätzung erstellt wird. Durch das Ausräumen von Unsicherheiten im Projektverlauf und einer genauer werdenden Spezifikation der Anforderungen kann die Ungenauigkeit von Schätzungen im Projektverlauf reduziert werden.

Traditionelle Vorgehensmodelle unterscheiden sich in einigen Aspekten, die signifikante Auswirkung auf die Schätzung und Planung haben, von agilen Vorgehensmodellen. Wesentliche Unterschiede sind im Entwicklungs-Lebenszyklus (linear bei traditionellem Zugang vs. iterativ bei agilem Zugang), im Stil der Entwicklung (vorausschauend vs. adaptiv) und in der Definition der Anforderungen (früh bekannt und stabil vs. entstehend, sich ändernd und unbekannt zu Projektbeginn) zu finden.

Dementsprechend sind auch die Methoden, die für das Erstellen von Schätzungen angewendet werden, in traditionellen und agilen Projekten unterschiedlich. Daher kann nicht davon ausgegangen werden, dass Faktoren, die die Qualität von Schätzungen in traditionellen Projekten verbessern, ebenfalls positive Auswirkungen auf die Schätzung in agilen Projekten haben. Des Weiteren können nicht alle Faktoren, die in traditionellen Projekten wirken, auf agile Projekte angewendet werden.

Für Schätzungen in agilen Vorgehensmodellen gibt es zwar eine Anzahl von unterschiedlichen Methoden, jedoch wenige Empfehlungen oder Studien darüber, wie die Genauigkeit der Schätzung in agilen Projekten verbessert werden kann. In gewissem Maße wird auf genaue Schätzungen und eine Planung, die Projektinhalt, Projektbudget und Projektzeitraum genau vorhersagt, sogar verzichtet, da agilen Projekten inhärent ist, dass sich Anforderungen und damit der Aufwand für die Umsetzung ändern.

Dennoch ist es auch in agilen Projekten insbesondere für das Management wichtig, Inhalt und Termine von Produktfreigaben zu kennen.

Zusammenfassend sind nachfolgend nochmals die Faktoren aufgelistet, welche laut den in dieser Arbeit zitierten Quellen die Genauigkeit von Aufwandsschätzungen in Softwareprojekten positiv beeinflussen.

| <i>Einflussfaktor</i> | <i>Quelle</i> | <i>Teil der weiteren Untersuchung</i> |
|---|--|--|
| <i>Maßvolle Evaluierung</i> | (Jorgensen, 2004) | Ja |
| <i>Vermeiden von widersprüchlichen Zielen; Auswahl der schätzenden Person</i> | (Jorgensen, 2004) | Ja |
| <i>Evaluierung der eigenen Schätzung</i> | (Jorgensen, 2004) | Ja |
| <i>Vermeiden irrelevanter Information</i> | (Jorgensen, 2004) | Ja |
| <i>Verwenden von historischen Daten</i> | (Jorgensen, 2004), (Hill, 2011) | Ja |
| <i>Verwenden von Checklisten</i> | (Jorgensen, 2004) | Ja |
| <i>Berücksichtigung der (Un-)Genauigkeit der Schätzung</i> | (Hohman, 2005), (Jorgensen, 2004) | Ja |
| <i>Kombination von Bottom-Up und Top-Down Schätzung</i> | (Jorgensen, 2004), (Leung & Fan, 2002), (Trendowicz & Ross, 2014) | Ja |
| <i>Kombination von (unabhängigen) Schätzungen mehrerer Experten</i> | (Jorgensen, 2004) | Ja |
| <i>Feedback zur Schätzung geben</i> | (Jorgensen, 2004), (Basten, 2012) | Nein |
| <i>Trainingsmöglichkeiten zur Verfügung stellen</i> | (Jorgensen, 2004) | Nein |
| <i>Schätzmethode (Planning Poker)</i> | (Grenning, 2002), (Usman, Mendes, Weidt, & Britto, 2014), (Osman & Musa, 2016), (Gandomani, Wei, & Binhamid, 2014) | Ja |
| <i>Stufenweise Planung</i> | (Usman, Mendes, Weidt, & Britto, 2014), (Steyer, 2010), (Gloski, 2013) | Ja |

Tabelle 6: Übersicht über die beschriebenen Einflussfaktoren auf die Genauigkeit von Schätzungen. Die letzte Spalte gibt an, ob die genannten Kriterien in dieser Arbeit weiter untersucht werden.

8 FALLSTUDIE

Im ersten Teil der Arbeit wurden, ausgehend von einer Betrachtung über die unterschiedlichen Kategorien von Vorgehensmodellen in der Softwareentwicklung über einen Überblick über wichtige agile Vorgehensmodelle, Erfolgsfaktoren für die Genauigkeit von Schätzungen zusammengefasst. Diese Erkenntnisse aus dem ersten Teil der Arbeit sollen nun im zweiten Teil der Arbeit verifiziert oder falsifiziert werden.

Für diese Fallstudie werden die Erfahrungen der Mitarbeiter sowie die historischen Daten aus dem österreichischen Software-Unternehmen rmDATA GmbH herangezogen.

Die rmDATA GmbH ist ein IT-Unternehmen mit in Summe rund 70 Mitarbeitern, wovon etwa die Hälfte in der Technik (Software-Entwicklung, Support, Betrieb und Produktmanagement) beschäftigt sind. rmDATA ist Software-Entwickler und Dienstleister für Vermessung und Geoinformation. Etwa die Hälfte des Umsatzes wird mit Standard-Software, die zweite Hälfte mit Projektentwicklungen für Kunden unterschiedlicher Größe erwirtschaftet. Im Laufe von nunmehr über 30 Jahren ist rmDATA von einem kleinen, Eigentümergeführten Unternehmen im Südburgenland zu einer Gruppe mit Standorten in drei Ländern gewachsen. Die Unternehmenszentrale und gesamte Software-Entwicklung und Administration ist weiterhin am Hauptsitz des Unternehmens im Südburgenland gebündelt. Die Gruppe wird durch Vertriebsniederlassungen in Südösterreich, in Wien, in München und in der Schweiz ergänzt. In Wien befindet sich des Weiteren der Sitz einer Tochterfirma, welche Dienstleistungen für Datenaufbereitung, -migration und -prüfung anbietet.

Seit rund 15 Jahren arbeitet rmDATA projektorientiert, was insbesondere die Entwicklung von Standard-Produkten und Kunden-Projekten, aber auch die Entwicklung von intern eingesetzter Software sowie Forschungsprojekte betrifft.

Seit einigen Jahren werden alle Entwicklungsprojekte mit agilen Methoden, die sich stark an Scrum orientieren, umgesetzt. Die Einführung von Scrum ist ab 2011 ausgehend von der Arbeitsweise eines großen Kunden zuerst in einzelnen Teams, danach in allen Teams in der Softwareentwicklung vollzogen worden. Im Jahr 2013 erfolgte eine Schulung für alle Product-Owner und Scrum-Master durch einen externen Dienstleister. Unmittelbar nach dieser Qualifizierungsmaßnahme wurde das vorgeschlagene Vorgehen sehr strikt eingehalten. Nach und nach wurde das Vorgehensmodell jedoch je nach Entwicklungsteam angepasst und teilweise auch wieder mit klassischen Methoden kombiniert (siehe auch Interview 4: *„Da muss man jetzt einmal unterscheiden, was heißt „agil“, weil wir zu 99% Kunden-Themen abhandeln und das ein komplett anderer agiler Workflow ist, wie wir es jetzt in anderen Abteilungen verwenden da in der Firma.“*)

Seit einer allgemeinen Konsolidierung des Vorgehensmodells im Jahr 2015 werden die Aufgaben des „klassischen“ Projektmanagers und des „agilen“ Scrum-Masters pro Team von einem Mitarbeiter in Personalunion wahrgenommen.

Die Abteilung Technik ist in die beiden Fachbereiche Vermessung und Geoinformation gegliedert. Im Fachbereich Vermessung sind derzeit 8 Personen in der Software-Entwicklung und zwei Personen im Kundensupport tätig. Die Software-Entwickler arbeiten in 2 Teams.

Im Fachbereich Geoinformation sind 26 Personen, gegliedert in vier Teams, tätig.

8.1 Experteninterviews

Mittels Experteninterviews mit den Projektmanagern/Scrum Mastern wurden die Erfahrungen der Mitarbeiter mit agilen Methoden, mit Schätzungen in agilen Projekten und deren Genauigkeit erhoben. Dazu wurden die Projektmanager von vier Teams interviewt. In zwei Fällen erfolgte das Interview nicht mit dem Projektmanager, sondern mit dem Teamleiter. In diesen beiden Fällen ist der Teamleiter ebenfalls regelmäßig in die Projektabwicklung, Anforderungserhebung und Schätzung involviert und verfügt über gute Erfahrung in diesen Gebieten.

Die Experteninterviews wurden als teilstrukturierte Interviews mit einem vorbereiteten Fragenkatalog durchgeführt. Der Fragenkatalog diente als Leitfaden, die Fragen wurden abhängig vom Verlauf des Interviews abgeändert, in der Reihenfolge variiert oder durch weitere Frage ergänzt. Die Interviews wurden aufgezeichnet und danach transkribiert (siehe Anhang).

Vor Beginn des Interviews wurden den Interviewpartnern die Ziele und in einem Überblick die bisherigen Erkenntnisse der Arbeit präsentiert. Alle Interviewpartner wurden persönlich zum Interview eingeladen und über die Ziele der Arbeit und des Interviews informiert.

Folgende Interviews wurden durchgeführt:

Fachbereich Vermessung

| Team | Projektmanager | Teamleiter |
|---------------|----------------|------------------|
| Vermessung I | Interview 2 | Thomas Marschall |
| Vermessung II | Interview 3 | Thomas Marschall |

Tabelle 7: Teams, Nummer des Interviews des Projektmanagers sowie Teamleiter im Fachbereich Vermessung

Fachbereich Geoinformation

| Team | Projektmanager | Teamleiter |
|--------------------|----------------|-------------|
| Geoinformation I | Interview 1 | Teamleiter |
| Geoinformation II | Projektmanager | Interview 5 |
| Geoinformation III | Interview 4 | Interview 4 |
| Geoinformation IV | Projektmanager | Interview 6 |

Tabelle 8: Teams, Nummer des Interviews des Experten sowie Teamleiter im Fachbereich Geoinformation

Wie sich aus der obigen Aufstellung ergibt, ist der Autor der Arbeit für zwei der Interviewten Experten Teamleiter. In einem Team ist der Projektmanager/Scrum Master gleichzeitig auch Teamleiter und in zwei Teams wurde der Teamleiter an statt des Projektmanagers interviewt.

8.1.1 Umfeld der Interviews

Zu Beginn des Gesprächs erfolgte, soweit dies nicht bereits im Vorfeld bei der persönlichen Einladung der Fall war, eine einleitende Erklärung über das Ziel der Arbeit und das Ziel des Interviews.

Ziel der Arbeit

„Das Ziel der Arbeit ist es, zu verifizieren, dass es in agilen Projekten Erfolgsfaktoren gibt, welche die Genauigkeit von Schätzungen positiv beeinflussen. Im theoretischen Teil der Arbeit wurden einige Faktoren ermittelt, welche die Genauigkeit von Schätzungen in klassischen und agilen Projekten positiv beeinflussen, sowie einige Faktoren, die die Genauigkeit von Schätzungen negativ beeinflussen. Dies sind organisatorischen, methodische und persönliche Faktoren sowie Faktoren, die von außen (z.B. das Management) auf das Projekt einwirken.

Im praktischen Teil der Arbeit werden Interviews mit Experten geführt, um die Methodik der Schätzungen in den Teams bei rmDATA kennen zu lernen, sowie die im Unternehmen bekannten Faktoren für eine Verbesserung der Schätzgenauigkeit. Von Interesse dabei sind auch die unterschiedlichen Hauptaufgaben der Teams. Weitere Erkenntnisse sollen aus einem Fragebogen, der an alle Mitarbeiter der Technik gesendet wird, gewonnen werden. Für Projekte, für die es genügend Daten gibt, erfolgt dann noch ein Vergleich des tatsächlichen und des geschätzten Aufwands.“

Ziel des Interviews

„Ziel des Interviews ist es, die Erfahrung des Projektmanagers kennen zu lernen, die Methodik, wie das Team bei Schätzungen umgeht sowie, ob und welche Methoden eingesetzt werden, um eine hohe Genauigkeit für Schätzungen zu erreichen. Es geht dabei um Erkenntnisse über die eingesetzten Methoden und die Bewertung dieser Methoden.

Kein Ziel des Interviews oder der Arbeit ist es, die involvierten Personen oder deren Leistungen zu beurteilen oder zu bewerten.

Das Interview wird aufgezeichnet und für die Arbeit transkribiert.“

Umfeld für das Interview

Aus organisatorischen Gründen erfolgten alle Interviews in einem Besprechungsraum am Firmensitz. Die Interviews wurden während der Arbeitszeit durchgeführt. Dem Interviewpartner wird das wissenschaftliche Interesse der Arbeit und des Interviews vermittelt. Die Interviewpartner werden in den Vorgesprächen und der Einleitung zum Interview dazu ermuntert, Ihre Erfahrungen ausführlich zu schildern. Die Erfahrungen der Projektleiter sind ein wesentlicher Beitrag zum Erfolg der Arbeit. Die Gesprächspartner werden auch dazu

eingeladen, ihre Fragen zum Interview und der Arbeit zu stellen. Für das Interview wird inklusive Vor- und Nachgespräch eine Dauer von maximal einer Stunde anberaunt.

8.1.2 Die Interviewpartner

Interview 1: Das Interview 1 wurde am 23.03.2017 ab 16:00 am Firmensitz durchgeführt. Interviewpartner 1 ist seit 2009 im Unternehmen und seit 2015 Projektmanager/Scrum Master in einem agilen Team. Das Team von Experte 1 besteht aus sieben Personen.

Interview 2: Das Interview wurde am 24.03.2017 ab 08:00 am Firmensitz durchgeführt. Interviewpartner 2 ist seit 1999 im Unternehmen und seit 2015 Projektmanager/Scrum Master in einem agilen Team. Davor hat er auch als Projektmanager in klassischen Teams gearbeitet. Das Team von Experte 2 besteht derzeit aus drei Personen. Die länger zurückliegenden Projekte wurden in größeren Teams mit bis zu neun Personen umgesetzt.

Interview 3: Das Interview wurde am 29.03.2017 ab 08:00 am Firmensitz durchgeführt. Interviewpartner 3 ist seit 2002 im Unternehmen und war mehrere Jahre als Scrum-Master in einem agilen Team. Im Jahr 2015 war er in einem ausgegliederten Team, dem Technologie- und Plattformteam, tätig. Derzeit übernimmt er die Rolle des Scrum-Masters im Team des Autors, der die Projektmanagement-Agenden in diesem Team wahrnimmt. Das Team von Experte 3 besteht aus fünf Personen.

Interview 4: Das Interview wurde am 30.03.2017 ab 10:30 am Firmensitz durchgeführt. Interviewpartner 4 ist seit 2002 im Unternehmen und seit 2015 Projektmanager/Scrum Master in einem agilen Team. Er ist gleichzeitig Projektmanager und Teamleiter. Das Team von Experte 4 besteht derzeit aus sechs Personen.

Interview 5: Das Interview wurde am 03.04.2017 ab 16:00 am Firmensitz durchgeführt. Interviewpartner 5 ist seit 2012 als freier Dienstnehmer für das Unternehmen tätig und seit 2015 Teamleiter in einem agilen Team. Vor 2012 war er bereits nebenberuflich für das Unternehmen tätig. Er hat Software-Engineering an der TU Wien studiert. Das Team von Experte 5 besteht derzeit aus fünf Personen, bis vor kurzem aus 6 Personen, eine davon teilzeitbeschäftigt.

Interview 6: Das Interview wurde am 05.04.2017 ab 09:30 am Firmensitz durchgeführt. Interviewpartner 5 ist seit 2003 im Unternehmen tätig. Er arbeitet als Teamleiter und Software-Entwickler. Das Team von ihm bestand bis vor kurzem aus fünf Personen und wird künftig auf drei Personen reduziert werden.

8.1.3 Gesprächsleitfaden

Die Interviews wurden an Hand eines vorbereiteten Gesprächsleitfadens geführt. Die Fragen wurden, abhängig vom Verlauf des Gesprächs, teilweise in anderer Reihenfolge oder mit angepassten Formulierungen gestellt. Teilweise wurden zusätzliche Fragen ergänzt oder einzelne Fragen ausgelassen.

Wie lange werden in Ihrem Team agile Methoden für die Projektentwicklung eingesetzt?

Welche Erfahrungen haben Sie mit der Einführung der agilen Methoden gemacht?

Hat es dabei besonders bemerkenswerte Ereignisse gegeben?

Welche agilen Methoden setzt das Team ein?

Mit welcher Methodik (Anm.: Planning Poker, Magic Estimation, ...) führt das Team Schätzungen durch?

Welche Metrik (Anm.: Arbeitstage, Story Points, Ideal Days, ...) verwendet das Team für die Schätzung?

Wenn mit Story Points geschätzt wird: Welche Faktoren fließen in die Schätzung mittels Story Points ein?

Führen Sie ein Controlling der Schätzungen (Vergleich von Schätzung und tatsächlichem Aufwand) durch?

Wissen Sie, wie gut die Qualität der Schätzungen ist oder haben Sie ein Gefühl für die Güte der Schätzungen?

Von welchen Faktoren hängt Ihrer Meinung nach die Genauigkeit von Schätzungen ab?

Wann (Anm.: in welchem zeitlichen Abstand zur Umsetzung) führen Sie die Schätzungen durch?

Von welchen Personen werden die Schätzungen durchgeführt (Anm.: sind das die Personen die auch für die Umsetzung verantwortlich sind)?

Welche Arten von Projekten setzt Ihr Team hauptsächlich um (Anm.: Standardprodukte, Kundenprojekte, Forschungsprojekte, neuartige Technologien, Domänen, Programmiersprachen, ...)?

Gibt es in den Projekten oft einen starken Druck (z.B. von der Unternehmensleitung), einen Zeit- oder Kostenrahmen einzuhalten?

In welcher Form werden die Arbeitspakete, die geschätzt werden müssen, beschrieben (Anm.: knappe oder ausführliche Beschreibung, nur essentielle Informationen oder auch Zusatzinformationen, rein technische Beschreibung oder auch Informationen zum Geschäftswert, ...)?

Erfolgt die Schätzung in Projekten mit unterschiedlichen Ansätzen (Anm.: Top-Down und Bottom-Up)?

Tabelle 9: Die Fragen, welche als Leitfaden für das Interview der befragten Experten dienen.

8.1.4 Die Transkriptionsregeln

Die Transkription erfolgte nach einigen wenigen Regeln unmittelbar nach dem jeweiligen Interview:

- Dialektfärbungen von Wörtern werden in hochdeutscher Form wiedergegeben.

- Dialektausdrücke oder umgangssprachliche Satzstellungen werden in ihrer originalen Form belassen und nach Gehör wiedergegeben.
- Füllwörter wie ah, äh, und ähnliches werden weggelassen.
- Wort- oder Satzteilwiederholungen werden weggelassen, soweit diese nicht auf das Verständnis oder den Sinn des Gesagten einen Einfluss haben.
- Erklärende Anmerkungen, welche Abkürzungen oder inhaltliche Aussagen für den Leser verständlich machen, werden pro Interview eingefügt.

8.1.5 Auswertung der Experteninterviews

Die Auswertung der Experteninterviews erfolgte durch Zusammenfassung und induktive Kategorienbildung wie von (Mayring, 2015) vorgeschlagen.

Als Analyseeinheit diente jeweils ein Interview. Aus jeder Analyseeinheit wurden jene Fundstellen übernommen, die Aussagen hinsichtlich der Genauigkeit von Schätzungen enthalten. Diese wurden in eine knappe, beschreibende Form umgeschrieben (paraphrasiert). Im nächsten Schritt wurde aus diesen Paraphrasen eine Generalisierung abgeleitet, welche den Erfolgsfaktoren für Schätzungen entspricht.

Anders als von (Mayring, 2015) beschrieben, erfolgte hier, auf Grund des vergleichsweise geringen Umfangs des Materials, keine zweistufige Reduktion. Somit wurden mehrfache Nennungen einer Generalisierung innerhalb eines Interviews nicht berücksichtigt. In diesem Fall wurden die zweite oder folgende Fundstellen, die zu gleichen Paraphrasen führen, nicht in die Auswertung aufgenommen und die Generalisierung somit kein weiteres Mal gezählt.

Die folgende Tabelle stellt das Ergebnis der induktiven Kategorienbildung aus den sechs durchgeführten Interviews dar.

| Quelle | Fundstelle | Paraphrase | Generalisierung | Auswirkung |
|----------------------|---|--|---|------------|
| Interview 1, Seite 2 | Dass man dann jetzt mittlerweile weiß, die Stunden sind schon wichtig, drum versuchen wir jetzt auch wieder in Stunden zu schätzen, damit wir die halt besser treffen. | Schätzungen in Stunden führen zu genaueren Schätzungen. | Schätzung in Stunden | positiv |
| Interview 1, Seite 5 | Und am Ende der Woche bekommt man eine Hilfe, dass man weiß, ob man noch viel Zeit hat, um sich auszutoben und etwas zu verbessern, oder ob's halt schon gescheiter ist, wenn man es auf den Boden bringt. | Am Ende der Woche erfolgt ein Controlling des Aufwands. | Laufendes Controlling des Aufwands | positiv |
| Interview 1, Seite 6 | Aber ich glaube, durch das öfter mal hingewiesen werden, schafft man es dann halt leichter, dass man es ein wenig schneller zu einem Ende bringt. | Es erfolgt eine laufende Evaluierung des Umsetzungsgrades durch den Projektmanager. | Evaluieren des aktuellen Umsetzungsgrades | positiv |
| Interview 1, Seite 6 | [...] der es schätzt vielleicht der ist, der es dann auch macht [...] | Die Schätzung erfolgt durch jene Person, die die Umsetzung macht. | Eigene Schätzung | positiv |
| Interview 1, Seite 6 | [...] der es schätzt auch einer ist, der das relativ gut kennt, das Produkt [...] | Die Schätzung erfolgt durch eine Person, die den zu schätzenden Gegenstand gut kennt. | Erfahrung des Schätzenden | positiv |
| Interview 1, Seite 8 | Es ist sicher auch eine Möglichkeit, dass die Zahl weniger wird, weil man im Zuge vom Reden eine alternative Umsetzung findet, oder weil halt der Schätzer draufkommt, dass er etwas verkehrt verstanden hat. | Während der Schätzung werden Unklarheiten oder Missverständnisse ausgeräumt. | Evaluierung der Schätzung durch andere | positiv |
| Interview 1, Seite 9 | [...] was halt da komplett rausfällt ist, sind irgendwie Erfahrungen, die halt viele Personen machen, wie gut ihre Schätzungen halt sind sozusagen. [...] was du bei einer anderen [...] Schätzung wieder besser wissen sollst, geht das eigentlich verloren. | Wenn die Top-Down Schätzung durch Personen vorgenommen wird, die bei der Bottom-Up Schätzung und dem Vergleich nicht dabei sind, geht das Wissen verloren. | Keine Evaluierung der eigenen Schätzung | negativ |
| Interview 2, Seite 2 | Wobei da natürlich sehr viel die Erfahrung mit einspielt. | Die Schätzung wird besser, wenn die schätzende Person viel Erfahrung hat. | Erfahrung des Schätzenden | positiv |
| Interview 2, | Schätzungen mit Story-Points [...] Hat es früher auch gegeben, wobei wir ehrlich gesagt mit | Die Schätzung mit Story-Points hat | Schätzung in Story Points | negativ |

| | | | | |
|-----------------------|--|--|---|---------|
| Seite 2 | dem, in dieser Phase damals eigentlich nur Probleme gehabt haben. | Probleme in der Genauigkeit verursacht. | | |
| Interview 2, Seite 3 | Wo die Schätzung gut passt, da sind keine Entwicklungen im NG-CORE, [...] | Hohe Komplexität des zu schätzenden Gegenstandes führt zu weniger genauer Schätzungen. | Anforderung mit technisch wenig komplexer Umsetzung | positiv |
| Interview 2, Seite 3 | Indirekt schon, ja. Weil doch immer auch die Ansage gekommen ist, ‚das kann ja nicht so lange dauern‘. Und mit diesem Hintergedanken machst du automatisch eine optimistischere Schätzung, obwohl es gar nicht stimmt. | Die Erwartung Externer hinsichtlich des Ergebnisses der Schätzung beeinflusst die Genauigkeit negativ. | Externer Druck, einen Zeit- oder Kostenrahmen einzuhalten | Negativ |
| Interview 2, Seite 3 | Arbeitspakete [...] sehr ausführlich beschrieben | Ausführlich beschriebene Anforderungen führen zu genaueren Schätzungen. | Detaillierte Beschreibung der Anforderung | Positiv |
| Interview 3, Seite 2 | Planning-Poker [...] weil du halt dann wirklich unbeeinflusst bist und wenn dann halt einer eine komplett andere Zahl hinhält, dann denkst du dir vielleicht doch einmal ‚Oha‘. | Planning Poker erhöht durch die Unabhängigkeit der Schätzungen die Genauigkeit der Schätzung. | Eingesetzte Schätzmethode | Positiv |
| Interview 3, Seite 3 | [...] aber wir haben sicher die Meilensteine regelmäßig abgeglichen. | Es erfolgt ein Abgleich des tatsächlichen Aufwands mit dem geschätzten Aufwand auf Basis von Meilensteinen. | Laufendes Controlling des Aufwands | positiv |
| Interview 3, Seite 3 | Ich glaube, bei den kleineren Paketen wird es so sein, dass wir manchmal zu viel annehmen und bei den großen zu wenig. | Abhängig von der Größe des zu schätzenden Arbeitspakets weichen die Schätzungen nach oben oder unten ab. | Größe der zu schätzenden Anforderung | Negativ |
| Interview 3, Seite 4 | [...] haben wir das dann im Endeffekt detailliert, die Arbeitspakete, [...]. Das waren am Anfang auch recht grobe Tasks [...] die müssen wir schneiden [...] Und da ist das dann eigentlich in Gang gekommen. | Im Laufe des Projekts werden die grobgranularen Anforderungen nach und nach detailliert. | Mehrstufige Planung | Positiv |
| Interview 3, Seite 4 | [...] dass die Erfahrung, die man selber hat in Projekten und auch mit der technischen Umsetzung vor allem, dass das eine sehr große Rolle spielt. | Die Erfahrung der Schätzenden in Projekten und hinsichtlich der technischen Umsetzung hat einen großen Einfluss auf die Genauigkeit der Schätzung. | Erfahrung des Schätzenden | positiv |
| Interview 3, Seite 4f | [...] wenn du dann einen Druck hast, irgendwas sagen zu müssen, dann läufst du, glaube ich, | Externer Druck auf den Schätzenden führt zu ungenauen Schätzungen. | Externer Druck, einen Zeit- oder Kostenrahmen | Negativ |

| | Gefahr, dass du halt irgendetwas sagst. | | einzuhalten | |
|----------------------|--|---|--|---------|
| Interview 3, Seite 5 | Dass man gewisse Stories, wenn man also wöchentlich schätzt, dass man die dann schon zwei- oder dreimal gesehen hat zumindest. | Im Zuge des Scrum-Prozesses werden Stories mehrfach in einem Estimation-Meeting behandelt und geschätzt. | Mehrfache Schätzung | Positiv |
| Interview 3, Seite 5 | [...] weil ich in dem Zuge draufgekommen bin, dass in dieser Story ganze Funktionalität nicht bedacht wurde, die noch irgendwo gemacht werden sollte oder muss. | Erfahrungen innerhalb des Projekts führen dazu, dass der Inhalt einer Anforderung genauer verstanden wird. | Umfassende Kenntnis der zu schätzenden Anforderung | positiv |
| Interview 3, Seite 5 | Jetzt haben die das vorher mal, um eine Größenordnung für die Projektdefinition zusammen zu bringen, mal geschätzt. Und jetzt schätzen wir es gerade wieder im Team [...] Jetzt, wo wir eigentlich, die Abschätzung der Arbeitspakete, die da drinnen sind, schon recht weit fortgeschritten sind, zeigt sich halt, dass da so ziemlich das gleiche herauskommen wird. | Schätzungen vor Beginn des Projekts werden Schätzungen auf Basis von Stories, die im Laufe des Projekts durchgeführt werden, gegenübergestellt. | Kombination von Top-Down und Bottom-Up Schätzung | Positiv |
| Interview 3, Seite 7 | Und dann wird halt, wenn genügend Information in Rücksprache mit jedem einzelnen – da wird jeder gefragt, ob er sich das jetzt Schätzen traut. Und dann sagt jeder reihum eine Zahl. | Mehrere Experten führen die Schätzung gemeinsam durch. | Kombination der Schätzung von mehreren Experten | Positiv |
| Interview 4, Seite 3 | Risiko bei der Sache ist natürlich, dass man sich halt stark verschätzen kann auch, weil man die Anforderungen nicht im Detail kennt. | Nicht bekannte Anforderungen führen gegebenenfalls zu einer falschen Schätzung. | Unklare Anforderungen | Negativ |
| Interview 4, Seite 3 | Und dann kommen diese Abgrenzungen, wenn sie für den Auftraggeber OK sind, aber auch in das Angebot hinein. Das hat sich gezeigt, das ist notwendig. | Abgrenzungen zu einer Anforderung müssen, um eine genaue Schätzung erstellen zu können, bei dieser beschrieben werden. | Detaillierte Beschreibung der Anforderung | Positiv |
| Interview 4, Seite 4 | Also dieses Controlling ist sehr wichtig, sonst kannst du es nicht im Blick behalten. | Laufendes Controlling des Aufwands ist für ein erfolgreiches Projekt notwendig. | Laufendes Controlling des Aufwands | Positiv |
| Interview 4, Seite 4 | Ich sage mal, bei Projekten, wo die Abschätzung bis, sage ich mal, 2.000 Stunden oder so geht, glaube ich, liegt man meistens recht gut. | Je größer ein Projekt ist, desto schwerer ist es, eine zutreffende Schätzung zu erstellen. | Größe des Gesamtprojekts | Negativ |
| Interview 4, Seite 5 | Eine Abschätzung, die Haupt Michael zum Beispiel fürs Web-GIS-Professional trifft, kann ich nicht umlegen auf jeden anderen Kollegen. | Bei der Schätzung muss berücksichtigt werden, ob der Schätzende die Umsetzung selbst macht oder nicht. | Eigene Schätzung | Positiv |

| Das geht einfach nicht. | | | | |
|-------------------------|--|---|--|---------|
| Interview 4, Seite 6 | Weil es kann sich in der Zwischenzeit schon so viel getan haben, dass wir da noch etwas Essentielles mitberücksichtigen müssen, was den Aufwand deutlich erhöht. | Wenn zwischen Schätzung und Umsetzung während eines laufenden Projekts viel Zeit vergeht, besteht die Gefahr, dass die Schätzung nicht mehr zutreffend ist. | Zeitliche Ferne von Schätzung und Umsetzung | Negativ |
| Interview 4, Seite 7 | Das sind einfach Erfahrungswerte, die wir über die Jahre gesammelt haben, wo man einfach schon einmal „eingefahren ist“ auf gut Deutsch, wo man dann nachträglich analysiert hat, warum ist das so schief gegangen, und festgestellt hat: Ja, dieses und jenes, und das müssen wir in Zukunft mitberücksichtigen dort. | Die Erfahrungen der Schätzenden in vergangenen Projekten begünstigen genauere Schätzungen. | Erfahrung des Schätzenden | Positiv |
| Interview 4, Seite 8 | Stichwort Checkliste [...] Schätzungen, was habe ich alles zu berücksichtigen, was habe ich nicht vergessen: das, das, das ... | Das Verwenden einer Checkliste hilft dabei, bei der Schätzung alle durchzuführenden Tätigkeiten zu berücksichtigen. | Verwenden von Checklisten | Positiv |
| Interview 5, Seite 3 | Wenn mehrere schätzen, hast du mehrere Vorteile. | Eine Schätzung, die von mehreren Personen gemeinsam durchgeführt wird, begünstigt genauere Schätzungen. | Kombination der Schätzung von mehreren Experten | Positiv |
| Interview 5, Seite 3 | [...] wir machen schon ein Controlling wo wir sagen: geht sich bis zum Ende aus, oder geht sich nicht aus. Wo müssen wir vielleicht „effizienter“ werden, damit es sich dann trotzdem noch ausgeht. | Laufendes Controlling des Aufwands ist für ein erfolgreiches Projekt notwendig. | Laufendes Controlling des Aufwands | Positiv |
| Interview 5, Seite 3 | Also bei uns ist es umso besser, umso besser die Anforderungen beschreiben sind in der Story. Umso genauer können wir schätzen. | Eine genaue Beschreibung einer Anforderung begünstigt eine genaue Schätzung. | Detaillierte Beschreibung der Anforderung | Positiv |
| Interview 5, Seite 3 | Und natürlich, umso mehr Wissende dabei sind, umso genauer wird es natürlich auch. Wenn jetzt lauter Programmierer etwas schätzen, bei einer Applikation, die sie gar nicht kennen, dann wird es natürlich auch ungenau. | Wenn Personen ein Produkt gut kennen, für das sie eine Anforderung schätzen, begünstigt das die Genauigkeit der Schätzung. | Erfahrung des Schätzenden | Positiv |
| Interview 5, Seite 4 | Und sie sich natürlich besser auskennen und natürlich ganz genau wissen, an welchen Schrauben man drehen muss, dass man die | Die genaue Kenntnis eines Produkts führt dazu, dass bei der Schätzung alle durchzuführenden Tätigkeiten | Umfassende Kenntnis der zu schätzenden Anforderung | positiv |

| | | | | |
|----------------------|---|---|---|---------|
| | Anforderung des Kunden erfüllen kann. | berücksichtigt werden können. | | |
| Interview 5, Seite 4 | Das heißt, das war ein Projekt, wo wir sehr viel mit Fremdsystemen zu tun hatten. [...] Und da war die Schätzung komplett daneben, [...] | Mangelnde Kenntnis von betroffenen Komponenten, die von einer Schätzung betroffen sind, kann zu einer ungenauen Schätzung führen. | Mangelnde Kenntnis der zu schätzenden Anforderung | Negativ |
| Interview 5, Seite 5 | Das heißt, wir schätzen jetzt nicht nur rein die Implementierung, sondern auch noch Zusatzaufwendungen für Abstimmungen, für Fremdsysteme, die einen aufhalten, und, und, und. | Wenn außer der Implementierung weitere Tätigkeiten bis zur Fertigstellung einer Anforderung zu erledigen sind, müssen diese in der Schätzung berücksichtigt werden. | Berücksichtigen aller Teilaspekte | Positiv |
| Interview 6, Seite 2 | Je nachdem, wie detailliert die Stories sind, kann man das detaillierter schätzen. | Eine genaue Beschreibung der Anforderungen begünstigt eine genaue Schätzung. | Detaillierte Beschreibung der Anforderung | Positiv |
| Interview 6, Seite 3 | Und ich sage dann, ja, das ist doch ein bisschen wenig, ein bisschen was aufschlagen | Die Evaluierung einer Schätzung durch einen Kollegen hilft dabei, eine ungenaue Schätzung zu vermeiden. | Evaluierung der Schätzung durch andere | Positiv |
| Interview 6, Seite 3 | Also, sie werden treffsicherer, wenn mehrere Leute schätzen. | Die Kombination der Schätzungen von mehreren Personen begünstigt die Genauigkeit der Schätzung. | Kombination der Schätzung von mehreren Experten | Positiv |
| Interview 6, Seite 4 | Also nicht so ein nachträgliches Controlling, sondern ein laufendes Controlling. Das wäre lässig. | Ein laufendes Controlling hilft dabei, zu hohem Aufwand für die Umsetzung einer Anforderung hintanzuhalten. | Laufendes Controlling des Aufwands | Positiv |
| Interview 6, Seite 4 | Umso weniger Tätigkeiten berücksichtigt werden müssen, bei der Schätzung, umso genauer wird es. [...] Das heißt, wenn der Entwickler, der wirklich nur seine Tätigkeiten schätzen muss wie Entwicklung, Test, [...] | Schätzungen werden genauer, wenn Entwickler nur eigene Tätigkeiten bei der Schätzung berücksichtigen müssen. | Keine Schätzung der Arbeit von anderen | Positiv |
| Interview 6, Seite 6 | Die, die das schätzen, stellen sich irgendwas vor; man unterschätzt das, weil der sich nicht reinversetzen kann in den anderen. Das vielleicht anders verstanden hat. | Nicht ausreichend beschriebene Anforderungen führen dazu, dass das Verständnis des Schätzenden von den gewünschten Anforderungen des Kunden abweichen kann. | Unklare Anforderungen | Negativ |

Tabelle 10: Übersicht über die Fundstellen, die zugeordneten Paraphrasen, die Generalisierung und die Art der Auswirkung auf die Genauigkeit von Schätzungen.

Auf Basis der obigen Tabelle wurden die gefundenen Generalisierungen, welche die Genauigkeit von Schätzungen positiv oder negativ beeinflussen, aus den Experteninterviews extrahiert und zusammengefasst. Die Anzahl den Nennungen der jeweiligen Generalisierung (=Erfolgsfaktor) ist ebenfalls angegeben. Die Sortierung erfolgte nach Anzahl der Nennungen.

In der ersten Tabelle sind jene Faktoren aufgelistet, welche laut Meinung der Experten positive Auswirkungen auf die Genauigkeit von Schätzungen in agilen Software-Entwicklungsprojekten haben.

| Erfolgsfaktor | Auswirkung | Interview 1 | Interview 2 | Interview 3 | Interview 4 | Interview 5 | Interview 6 | Anzahl Nennungen |
|---|------------|-------------|-------------|-------------|-------------|-------------|-------------|------------------|
| Erfahrung des Schätzenden | positiv | X | X | X | X | X | | 5 |
| Laufendes Controlling des Aufwands | positiv | X | | X | X | X | | 4 |
| Detaillierte Beschreibung der Anforderung | positiv | | X | | X | X | X | 4 |
| Kombination der Schätzung von mehreren Experten | positiv | | | X | | X | X | 3 |
| Eigene Schätzung | positiv | X | | | X | | | 2 |
| Evaluierung der Schätzung durch andere | positiv | X | | | | | X | 2 |
| Umfassende Kenntnis der zu schätzenden Anforderung | positiv | | | X | | X | | 2 |
| Schätzung in Stunden | positiv | X | | | | | | 1 |
| Evaluieren des aktuellen Umsetzungsgrades | positiv | X | | | | | | 1 |
| Anforderung mit technisch wenig komplexer Umsetzung | positiv | | X | | | | | 1 |
| Eingesetzte Schätzmethode | positiv | | | X | | | | 1 |
| Mehrstufige Planung | positiv | | | X | | | | 1 |
| Mehrfache Schätzung | positiv | | | X | | | | 1 |
| Kombination von Top-Down und Bottom-Up Schätzung | positiv | | | X | | | | 1 |
| Verwenden von Checklisten | positiv | | | | X | | | 1 |
| Berücksichtigen aller Teilaspekte | positiv | | | | | X | | 1 |
| Keine Schätzung der Arbeit von anderen | positiv | | | | | | X | 1 |

Tabelle 11: Von den Experten genannte Faktoren, welche die Genauigkeit von Schätzungen positiv beeinflussen.

In der zweiten Tabelle sind jene Faktoren aufgelistet, welche laut Meinung der Experten negative Auswirkungen auf die Genauigkeit von Schätzungen in agilen Software-Entwicklungsprojekten haben.

| (Negativer) Erfolgsfaktor | Auswirkung | Interview 1 | Interview 2 | Interview 3 | Interview 4 | Interview 5 | Interview 6 | Anzahl Nennungen |
|---|------------|-------------|-------------|-------------|-------------|-------------|-------------|------------------|
| Externer Druck, einen Zeit- oder Kostenrahmen einzuhalten | negativ | | X | X | | | | 2 |
| Unklare Anforderungen | negativ | | | | X | | X | 2 |
| Schätzung in Story Points | negativ | | X | | | | | 1 |
| Keine Evaluierung der eigenen Schätzung | negativ | X | | | | | | 1 |
| Größe der zu schätzenden Anforderung | negativ | | | X | | | | 1 |
| Größe des Gesamtprojekts | negativ | | | | X | | | 1 |
| Zeitliche Ferne von Schätzung und Umsetzung | negativ | | | | X | | | 1 |
| Mangelnde Kenntnis der zu schätzenden Anforderung | negativ | | | | | X | | 1 |

Tabelle 12: Von den Experten genannte Faktoren, welche die Genauigkeit von Schätzungen negativ beeinflussen.

Die Erfolgsfaktoren sind auch in der zweiten Tabelle nach der Anzahl der Häufigkeit der Nennung gereiht. Sieben Faktoren, welche die Genauigkeit von Schätzung positiv beeinflussen, wurden von mehreren Experten genannt. Zwei Faktoren, welche die Genauigkeit von Schätzung negativ beeinflussen, wurden von jeweils zwei Experten genannt. Alle weiteren, positiven wie negativen, Erfolgsfaktoren wurden in den Experteninterviews mit den Projektmanagern und Teamleitern in den sechs Entwicklungsteam des betrachteten Unternehmens nur einmal genannt.

Erfahrung des Schätzenden

Mit fünf Nennungen am häufigsten genannt wurde die Erfahrung des Schätzenden. Dabei handelt es sich um die Erfahrung, die der Schätzende mit der Domäne ⁴, dem Projektumfeld ⁵ oder dem zu entwickelnden Produkt ⁶ gesammelt hat. Einmal wurde auch jene Erfahrung genannt, welche der Schätzende im Zuge des laufenden Projekts gesammelt hat. Dadurch, dass er Erfahrungen mit ähnlichen Stories im laufenden Projekt gesammelt habe, könne er in weiterer Folge den Aufwand für nachfolgend umzusetzende Stories genauer schätzen.

⁴ Interview 5: Also, dadurch, dass wir jetzt die Erfahrung gewonnen haben, mit SAP ist es sehr, sehr schwer.

⁵ Interview 4: Das sind einfach Erfahrungswerte, die wir über die Jahre gesammelt haben, wo man einfach schon einmal „eingefahren ist“ auf gut Deutsch, wo man dann nachträglich analysiert hat, warum ist das so schief gegangen, und festgestellt hat: Ja, dieses und jenes, und das müssen wir in Zukunft mitberücksichtigen dort.

⁶ Interview 3: [...] dass die Erfahrung, die man selber hat in Projekten und auch mit der technischen Umsetzung vor allem, dass das eine sehr große Rolle spielt.

Laufendes Controlling des Aufwands

Als positiven Einflussfaktor darauf, wie gut die Schätzung und der tatsächliche Aufwand übereinstimmen, wurde viermal das laufende Controlling des Aufwands genannt. Wie sich aus den Interviews ergibt, beeinflusst dieser Faktor aber tatsächlich nicht die Genauigkeit der Schätzung vor Beginn der Umsetzung, sondern beeinflusst den Aufwand, welcher in die Umsetzung einer Anforderung investiert wird ⁷. Durch das Controlling kann verhindert werden, dass der Aufwand die ursprüngliche Schätzung übersteigt.

Das wiederum bedingt, dass der geschätzte Aufwand in Stunden bekannt ist. Die **Schätzung in Stunden** wurde in einem Interview als positiver Einflussfaktor genannt, wohingegen die **Schätzung in Story Points** von einem anderen Experten als negativer Einflussfaktor genannt wurde ⁸. In diesem Zusammenhang hat sich gezeigt, dass in den Teams aller befragter Experten tatsächlich in Stunden oder Tagen geschätzt wird. Selbst wenn die verwendete Einheit „Story Points“ ist, ist einem Story Point im Verständnis der beteiligten Teammitglieder immer eine Zeitspanne hinterlegt ⁹.

Detaillierte Beschreibung der Anforderung

In vier Interviews wurde die detaillierte Beschreibung der Anforderung als positiver Einflussfaktor auf die Genauigkeit der Schätzung genannt. Demgegenüber wurden unklare Anforderungen weitere 2 Mal als negativ für die Genauigkeit der Schätzung bezeichnet. Der Detaillierungsgrad bezieht sich dabei auf die Anforderung selbst, also was ist umzusetzen, aber auch auf die Abgrenzungen ¹⁰, wodurch definiert wird, was im Rahmen der Umsetzung einer Anforderung nicht gemacht werden muss. Von einem Experten besonders betont wurde zusätzlich ¹¹, dass Implikationen in Anforderungen sehr Häufig zu Problemen in der Genauigkeit der Schätzung führen, da sich die Implikationen dessen, der die Anforderung stellt und dessen, der die Umsetzung durchführt, unterscheiden könnten. Dadurch ergäben sich Unterschiedliche Auffassungen über den Inhalt einer User-Story und in weiterer Folge darüber, wann die Umsetzung abgeschlossen sei.

Kombination der Schätzung von mehreren Experten

Die Kombination der Schätzungen von mehreren Experten wurden in drei Interviews als positive Faktoren für die Genauigkeit der Schätzung genannt ¹². Bei der Schätzung von mehreren

⁷ Interview 5: Nein, wir machen schon ein Controlling wo wir sagen: geht sich bis zum Ende aus, oder geht sich nicht aus. Wo müssen wir vielleicht „effizienter“ werden, damit es sich dann trotzdem noch ausgeht.

⁸ Interview 2: Schätzungen mit Story Points [...] Hat es früher auch gegeben, wobei wir ehrlich gesagt mit dem, in dieser Phase damals eigentlich nur Probleme gehabt haben. Niemand hat dann einen Endtermin sagen können, niemand hat wirklich den Stundenaufwand sagen können [...]

⁹ Interview 4: Bei uns ist es so, dass ein Story Point ein Tag ist. Angebotstechnisch sind es 10 Stunden allerdings.

¹⁰ Interview 4: Und dann kommen diese Abgrenzungen, wenn sie für den Auftraggeber OK sind, aber auch in das Angebot hinein. Das hat sich gezeigt, das ist notwendig.

¹¹ Interview 4: Weil jemand, der sich bei dem Thema top auskennt, neigt dazu, bei den Beschreibungen etwas wegzulassen. Die ‚eh klar sind‘. Und das, was ‚eh klar ist‘, ist das, was immer das Problem macht.

¹² Interview 6: Also, sie werden treffsicherer, wenn mehrere Leute schätzen.

Experten könnten mehrere Meinungen berücksichtigt werden, unterschiedliche Schätzungen diskutiert und so differierende Annahmen ausgeräumt werden. Dieser positive Einflussfaktor hänge wiederum auch mit dem vorigen Punkt, der Detaillierten Beschreibung der Anforderungen sowie den erfassten Abgrenzungen zusammen. Diese kämen durch die Diskussion mehrerer Experten zu Tage und könnten somit in der Schätzung berücksichtigt werden¹³, was diese genauer mache.

Eigene Schätzung

Zweimal als positiv für die Genauigkeit der Schätzung genannt wurde die eigene Schätzung. Das bezieht sich darauf, dass die Schätzung dann genauer werde, wenn Personen ihre eigene Arbeit schätzen, da dann die eigene Erfahrung und Expertise in der Schätzung berücksichtigt werden könne, was bei der Schätzung von Arbeiten, die durch einen anderen erfolgt, nicht der Fall sei¹⁴. Im zweiten Fall wurde keine nähere Begründung für den positiven Einfluss der eigenen Schätzung auf die Genauigkeit genannt.

Evaluierung der Schätzung durch andere

In zwei Fällen wurde die Evaluierung der Schätzung durch andere als positiver Faktor für die Genauigkeit der Schätzung genannt. Durch diese Evaluierung sei es möglich, eventuell vergessene Aspekte, die den Aufwand für die Umsetzung einer User Story beeinflussen, zu berücksichtigen¹⁵. Ebenfalls könnten durch die Evaluierung der Schätzung durch andere, sofern es unterschiedliche Meinungen über den geschätzten Aufwand gibt, Unklarheiten oder Missverständnisse hinsichtlich der Anforderung vorab ausgeräumt¹⁶ werden und dadurch die Schätzung genauer gemacht werden.

Umfassende Kenntnis der zu schätzenden Anforderung

Ebenfalls zweimal wurde die umfassende Kenntnis der zu schätzenden Anforderung als positiver Faktor für die Genauigkeit der Schätzung genannt. Dieser bezieht sich vor allem auf das Produkt, in welchem die Anforderung umzusetzen ist, in geringerem Maße auch darauf, was der Kunde für Erwartungen mit der Formulierung der Anforderung ausdrücken möchte. Die umfassende Kenntnis des Produkts sei deshalb hilfreich, weil dadurch notwendige Änderungen in allen Programmteilen bereits vorab bekannt seien und sich im Zuge der Umsetzung keine zusätzlichen Arbeitspakete ergäben¹⁷.

Dem gegenüber wurde einmal die **mangelnde Kenntnis der umzusetzenden Anforderung** als negativ für die Genauigkeit der Schätzung genannt. Diese mangelnde Kenntnis führe dazu,

¹³ Interview 6: Und dann redet man nochmal drüber. Es wird dann tendenziell weniger, weil gewisse Sachen ausgeräumt werden, Unklarheiten. Und gewisse Ausschluss-Kriterien definiert werden.

¹⁴ Interview 4: Eine Abschätzung, die Haupt Michael zum Beispiel fürs Web-GIS-Professional trifft, kann ich nicht umlegen auf jeden anderen Kollegen. Das geht einfach nicht.

¹⁵ Interview 6: Und ich sage dann, ja, das ist doch ein bisschen wenig, ein bisschen was aufschlagen

¹⁶ Interview 1: Es ist sicher auch eine Möglichkeit, dass die Zahl weniger wird, weil man im Zuge vom Reden eine alternative Umsetzung findet, oder weil halt der Schätzer draufkommt, dass er etwas verkehrt verstanden hat.

¹⁷ Interview 5: Und sie sich natürlich besser auskennen und natürlich ganz genau wissen, an welchen Schrauben man drehen muss, dass man die Anforderung des Kunden erfüllen kann.

dass nicht alle notwendigen Tätigkeiten für die Umsetzung einer Anforderung vorab bekannt seien und daher in der Schätzung nicht berücksichtigt werden könnten¹⁸.

Alle weiteren positiven Einflussfaktoren wurden lediglich durch einen der Interviewten Experten genannt.

Externer Druck, einen Zeit- oder Kostenrahmen einzuhalten

In zwei Interviews wurde externer Druck, einen Zeit- oder Kostenrahmen einzuhalten, als die Genauigkeit von Schätzungen negativ beeinflussend genannt. In einem Fall wurde dies als Faktor dafür genannt, dass durch diesen Druck eine zu optimistische Schätzung genannt wird, obwohl diese nicht zutreffend sei. Im zweiten Fall wurde angegeben, dass durch den externen Druck die Schätzung nicht mit der notwendigen Sorgfalt durchgeführt werde und diese daher nicht die erwartete hohe Genauigkeit erreichen könne¹⁹.

Unklare Anforderungen

In zwei Fällen wurden unklare Anforderungen als negativer Einflussfaktor auf die Genauigkeit von Schätzungen genannt. Unklare Anforderungen hätten zur Folge, dass sich die Erwartungen oder Implikationen des Auftraggebers und des Schätzenden möglicher Weise so stark unterscheiden, dass dies einen Einfluss auf den Umsetzungsaufwand habe²⁰.

8.2 Befragung der Mitarbeiter der Abteilung Technik

Ergänzend zu den Interviews der Experten im Unternehmen rmDATA wurde ein Fragebogen erstellt und an alle Mitarbeiter der Abteilung Technik, die auch in Software-Entwicklungs-Projekten tätig sind, versendet. Die Umfrage wurde an 34 Personen, unter diesen auch jene sechs Personen, die im Rahmen der Experteninterviews befragt wurden, übermittelt. 27 Personen aus dem Unternehmen haben den Fragebogen ausgefüllt. Da es sich bei allen Fragen um Pflichtantworten handelte, wurden alle Fragen von allen Teilnehmern beantwortet.

8.2.1 Erfahrung in Projekten, Methodik für die Schätzung und Evaluierung

Knapp 60% der Teilnehmer haben bereits in mehr als 10 agilen Projekten mitgearbeitet, weitere rund 35% an mindestens sechs agilen Projekten:

¹⁸ Interview 5: Das heißt, das war ein Projekt, wo wir sehr viel mit Fremdsystemen zu tun hatten. [...] Und da war die Schätzung komplett daneben, [...]

¹⁹ Interview 3: [...] wenn du dann einen Druck hast, irgendwas sagen zu müssen, dann läufst du, glaube ich, Gefahr, dass du halt irgendetwas sagst.

²⁰ Interview 6: Die, die das schätzen, stellen sich irgendwas vor; man unterschätzt das, weil der sich nicht reinversetzen kann in den anderen. Das vielleicht anders verstanden hat.

In wie vielen agilen Projekten haben Sie bereits mitgearbeitet? (27 Antworten)

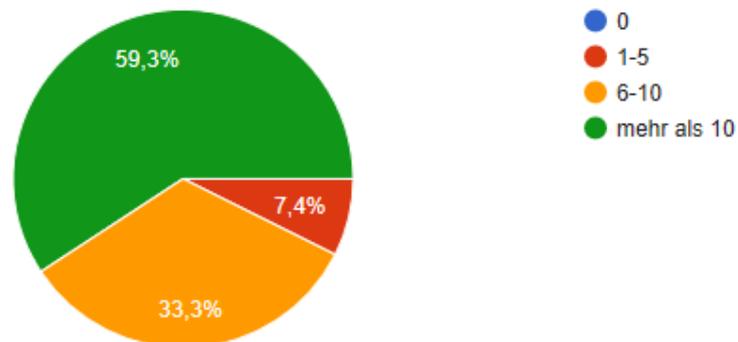


Abbildung 9: Anzahl der Projekte, in denen die Befragten mitgearbeitet haben. Quelle: eigene Erhebung mittels Google Formulare

Entsprechend der Personalsituation der eingeladenen Personen sind über 2/3 aller Teilnehmer überwiegend als Entwickler/Team Member tätig, knapp 20% als Projektleiter bzw. Scrum Master, 7% bzw. 2 der antwortenden Personen als Product Owner und 4% oder eine Person als Auftraggeber-Vertreter.

In welcher Rolle waren Sie in diesen Projekten (überwiegend) tätig? (27 Antworten)

(27 Antworten)

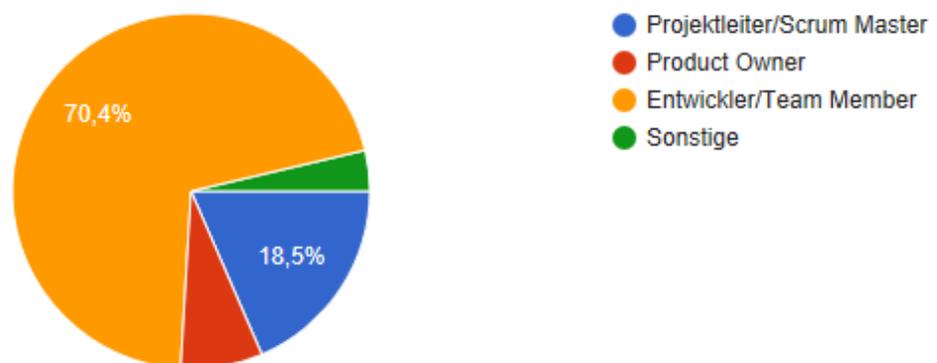


Abbildung 10: Funktion der Befragten. Quelle: eigene Erhebung mittels Google Formulare

In knapp 2/3 aller Fälle wurde die Schätzung mittels Planning Poker durchgeführt, in über ¼ der Fälle mit Magic Estimation. In einem Fall erfolgten die Schätzungen mit beiden Methoden, einmal erfolgte keine Angabe zum eingesetzten Schätzverfahren („Weiß nicht“).

Mit welcher Methode haben Sie in den agilen Projekten Schätzungen durchgeführt?

(27 Antworten)

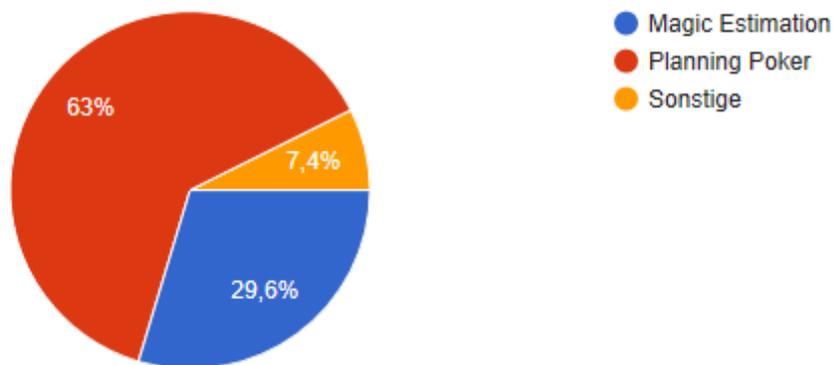


Abbildung 11: Eingesetzte Methode für Schätzungen. Quelle: eigene Erhebung mittels Google Formulare

Magic Estimation ist ein Verfahren für das Durchführen von Schätzungen, welches von Boris Gloger, basierend auf einer Idee von Lowell Lindstrom, entwickelt wurde (Opelt, Gloger, Pfarl, & Mittermayr, 2013).

Magic Estimation hat laut (Opelt, Gloger, Pfarl, & Mittermayr, 2013) die Vorteile, dass es schnell und auch in großen Teams und mit einer großen Anzahl von Stories durchgeführt werden kann. Das Verfahren beruht darauf, dass jeder der teilnehmenden Schätzer eine (für alle ungefähr gleiche große) Anzahl von Stories einem Wert von Story-Points zuordnet. Die geschieht dadurch, dass die jeweilige Story (ausgedruckt auf Papier) zu einem Punktwert gelegt oder an die Wand geheftet wird. Dies machen alle Personen gleichzeitig für ihre Stories, ohne miteinander zu sprechen.

Wenn dies abgeschlossen ist, hat jeder Teilnehmer die Möglichkeit, die Bewertung für jede beliebige Story zu ändern, indem er die Story zu einem anderen Story-Point-Wert legt. Stories, die mehrmals bewegt werden, werden durch den Product-Owner markiert. Diese Stories werden im Anschluss diskutiert, alle anderen sind nach den beiden Runden fertig geschätzt und die Schätzung wird in das Backlog übernommen.

Zwei Drittel aller Befragten haben angegeben, dass in ihrem Team die Größe von Backlog Items mit Story Points gemessen wird. Im restlichen Drittel der Fälle wird entweder mit Arbeitstagen oder tatsächlichen Stunden (22%) geschätzt.

Die Angaben zur Schätzung in Story Points sind jedoch auf Grund der Aussagen der Projektmanager in den Experten-Interviews dahingehend zu relativieren, dass in allen Teams, die Story Points als Einheit verwenden, diese immer eine zeitliche Größe (ein Tag, 8 Stunden oder abhängig vom Projekt ein variabler Wert zwischen sechs und acht Stunden) darstellen.

Mit welcher Metrik wurde die Größe von Backlog Items in den Projekten (überwiegend) gemessen?

(27 Antworten)

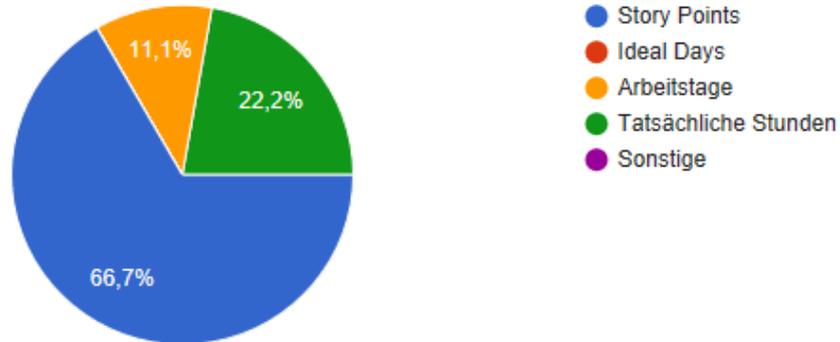


Abbildung 12: Metrik, mit der die Größe von Backlog-Items gemessen wird. Quelle: eigene Erhebung mittels Google Formulare

Eine Evaluierung des Aufwands pro Backlog-Item oder für das Gesamtprojekt erfolgte laut den Aussagen von 22% bis 37% der Befragten überhaupt nicht, in weniger als der Hälfte der Fälle teilweise und in 18% der Fälle pro Backlog-Item, in knapp 30% der Fälle durchgehend für das Gesamtprojekt:

Wurde in diesen Projekten die Genauigkeit der Schätzungen pro Backlog-Item (Vergleich Schätzung vs. tatsächlicher Aufwand) evaluiert?

(27 Antworten)

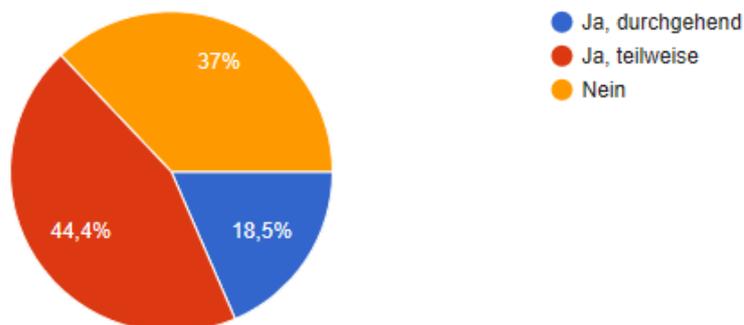


Abbildung 13: Art der Evaluierung der Genauigkeit von Schätzungen pro Backlog-Item. Quelle: eigene Erhebung mittels Google Formulare

Erfolgte jeweils eine Evaluierung für das Gesamtprojekt? (27 Antworten)

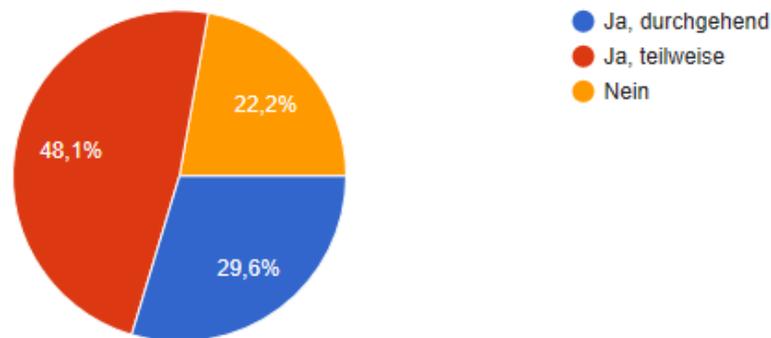


Abbildung 14: Art der Evaluierung der Genauigkeit von Schätzungen für das Gesamtprojekt. Quelle: eigene Erhebung mittels Google Formulare

8.2.2 Meinung der Befragten zu positiven Einflussfaktoren

Im zweiten Teil der Befragung wurden die Teilnehmer gebeten zu beurteilen, wie sehr die folgenden Faktoren die Genauigkeit von Schätzungen positiv beeinflussen. Die Beurteilung sollte auf einer sechsteiligen Skala von 1 (trifft absolut zu) bis 6 (trifft keineswegs zu) erfolgen.

- Evaluierung der Schätzung (durch Dritte)
- Evaluierung der eigenen Schätzung und neuerliche Schätzung
- Verwenden von historischen Daten
- Dreiseitige Schätzung, d.h. Schätzung nicht als einzelner Wert, sondern mit Werten für „early/likely/late“
- Verwenden von Checklisten (z.B. zum Berücksichtigen aller Punkte aus der Definition of Done)
- Kombination von Bottom-Up und Top-Down Schätzungen
- Die eingesetzte Schätzmethode
- Trainingsmöglichkeiten für die schätzenden Personen
- Stufenweise Planung (Kombination aus Planung der Version, des kommenden Sprints und der täglichen Planung)

| | 1 ... Trifft absolut zu | 2 | 3 | 4 | 5 | 6 ... trifft keineswegs zu | Eingesetzt |
|--|-------------------------|----|----|---|---|----------------------------|------------|
| Verwenden von historischen Daten | 2 | 8 | 11 | 3 | 1 | 2 | 16 |
| Evaluierung der eigenen Schätzung und neuerliche Schätzung | 5 | 10 | 7 | 2 | 1 | 2 | 14 |
| Evaluierung der Schätzung (durch Dritte) | 3 | 7 | 8 | 3 | 5 | 1 | 11 |
| Verwenden von Checklisten | 10 | 7 | 3 | 1 | 5 | 1 | 7 |
| Dreiseitige Schätzung, d.h. Schätzung mit Werten für „early/likely/late“ | 3 | 7 | 10 | 2 | 5 | 0 | 5 |
| Stufenweise Planung (Planung der Version, des Sprints und täglichen Planung) | 2 | 7 | 11 | 3 | 2 | 2 | 6 |
| Kombination von Bottom-Up und Top-Down Schätzungen | 1 | 4 | 15 | 4 | 3 | 0 | 4 |
| Trainingsmöglichkeiten für die schätzenden Personen | 3 | 7 | 5 | 5 | 5 | 2 | 1 |
| Die eingesetzte Schätzmethode | 2 | 8 | 10 | 4 | 2 | 1 | |

Tabelle 13: Die abgefragten Erfolgsfaktoren, sortiert nach der Häufigkeit, wie oft sie eingesetzt werden. Quelle: Eigene Erhebung

Am häufigsten werden von den befragten Personen historische Daten, die eigene Evaluierung und die Evaluierung durch Dritte für die Erhöhung der Schätzgenauigkeit eingesetzt. Die Spalten „1“ bis „6“ geben an, wie sehr die befragten Personen annehmen, dass die genannten Erfolgsfaktoren die Genauigkeit von Schätzungen positiv beeinflussen.

In der folgenden Tabelle sind die Erfolgsfaktoren danach gereiht, wie viele der befragten Personen der Meinung sind, dass diese die Genauigkeit von Schätzungen absolut („1“) oder stark („2“) positiv beeinflussen.

| | 1 ... Trifft absolut zu | 2 | 3 | 4 | 5 | 6 ... trifft keineswegs zu | Eingesetzt |
|--|-------------------------|----|----|---|---|----------------------------|------------|
| Verwenden von Checklisten | 10 | 7 | 3 | 1 | 5 | 1 | 7 |
| Evaluierung der eigenen Schätzung und neuerliche Schätzung | 5 | 10 | 7 | 2 | 1 | 2 | 14 |
| Evaluierung der Schätzung (durch Dritte) | 3 | 7 | 8 | 3 | 5 | 1 | 11 |
| Dreiseitige Schätzung, d.h. Schätzung mit Werten für „early/likely/late“ | 3 | 7 | 10 | 2 | 5 | 0 | 5 |
| Trainingsmöglichkeiten für die schätzenden Personen | 3 | 7 | 5 | 5 | 5 | 2 | 1 |
| Verwenden von historischen Daten | 2 | 8 | 11 | 3 | 1 | 2 | 16 |
| Die eingesetzte Schätzmethode | 2 | 8 | 10 | 4 | 2 | 1 | |
| Stufenweise Planung (Planung der Version, des Sprints und täglichen Planung) | 2 | 7 | 11 | 3 | 2 | 2 | 6 |
| Kombination von Bottom-Up und Top-Down Schätzungen | 1 | 4 | 15 | 4 | 3 | 0 | 4 |

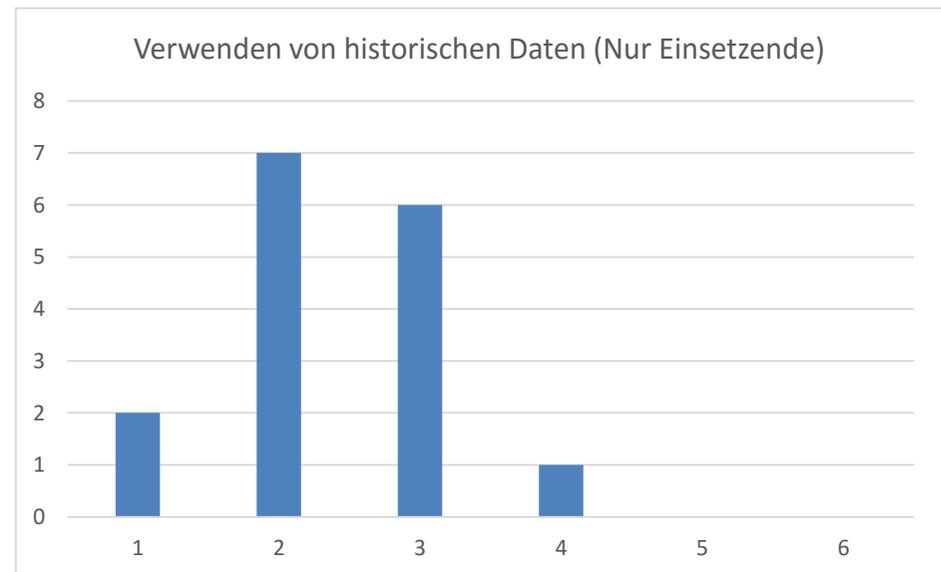
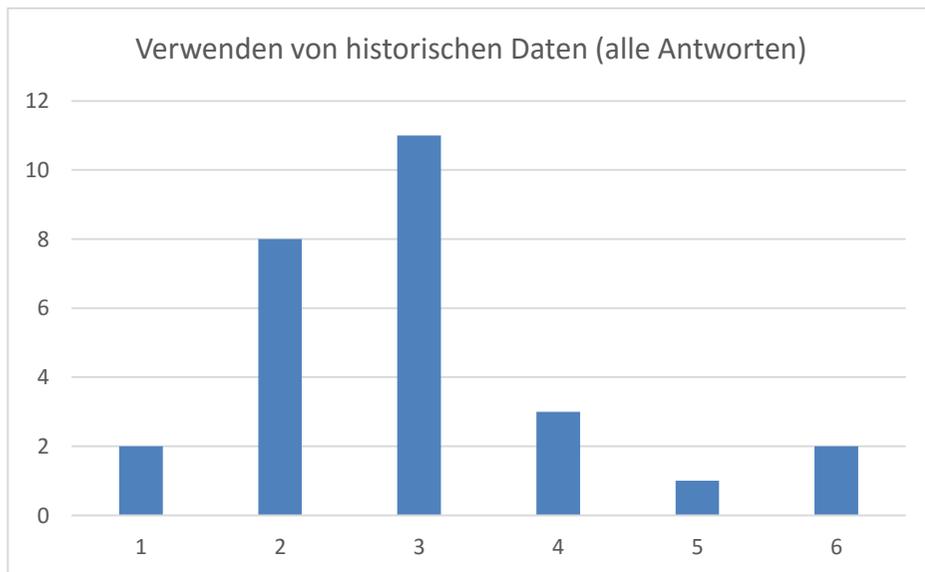
Tabelle 14: Die abgefragten Erfolgsfaktoren, sortiert nach der Häufigkeit, der Einstufung 1 und 2. Quelle: Eigene Erhebung

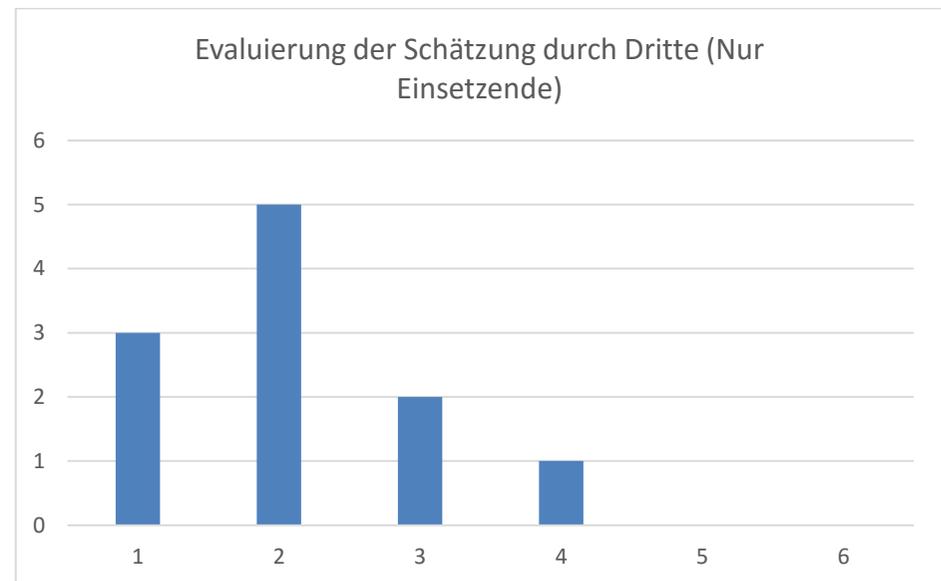
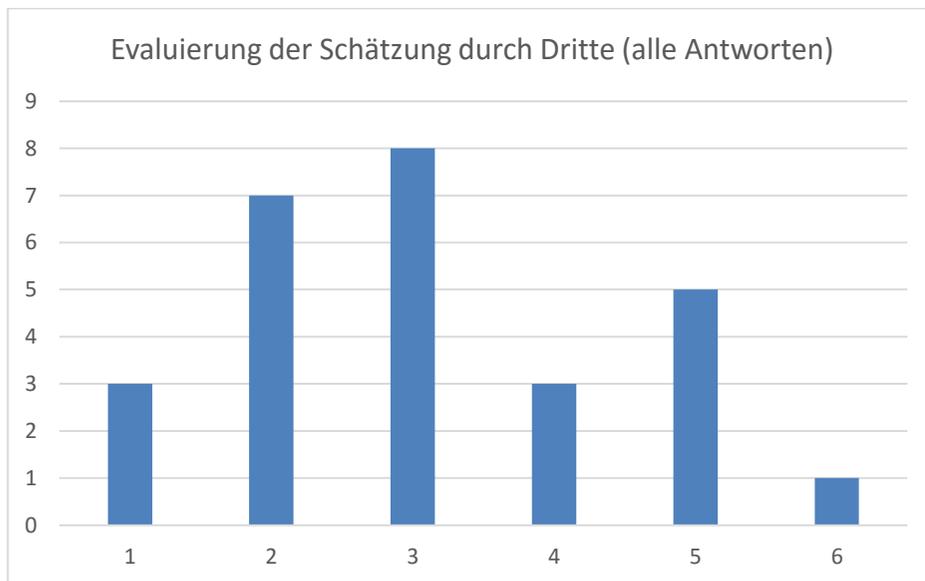
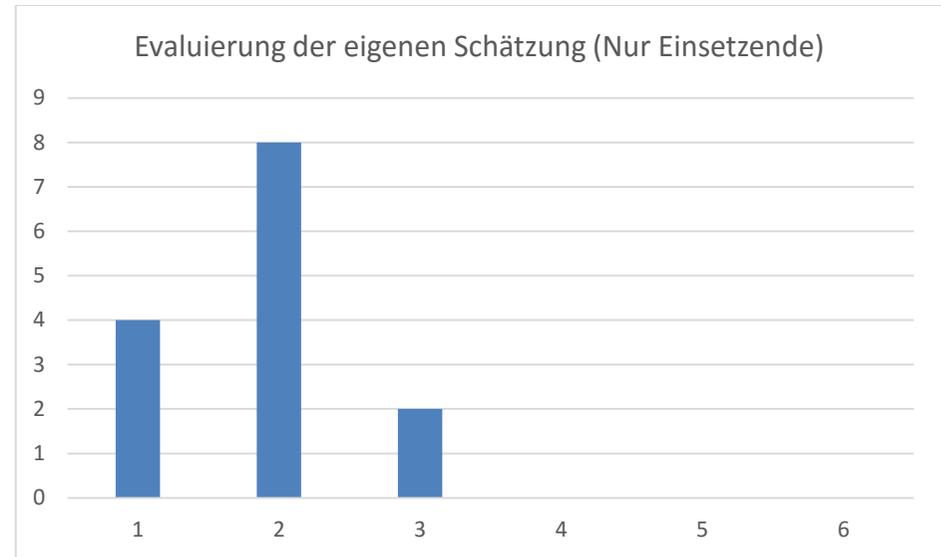
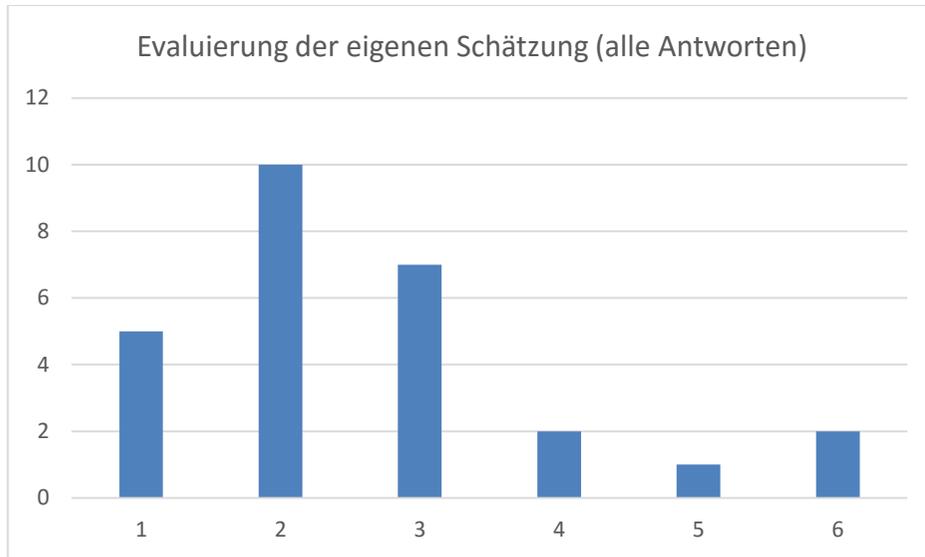
17 der befragten Personen sind der Meinung, dass das Verwenden von Checklisten die Genauigkeit erhöht, 15 der Personen, dass die Evaluierung der eigenen Schätzung und neuerliche Schätzung die Genauigkeit erhöht. Nur 5 Personen sind mit der Einstufung „1“ oder „2“ der Meinung, dass die Kombination von Bottom-Up und Top-Down Schätzungen die Genauigkeit von Schätzungen erhöht. Für diesen Erfolgsfaktor wurde aber auch nur von 4 Personen angegeben, dass er eingesetzt wird.

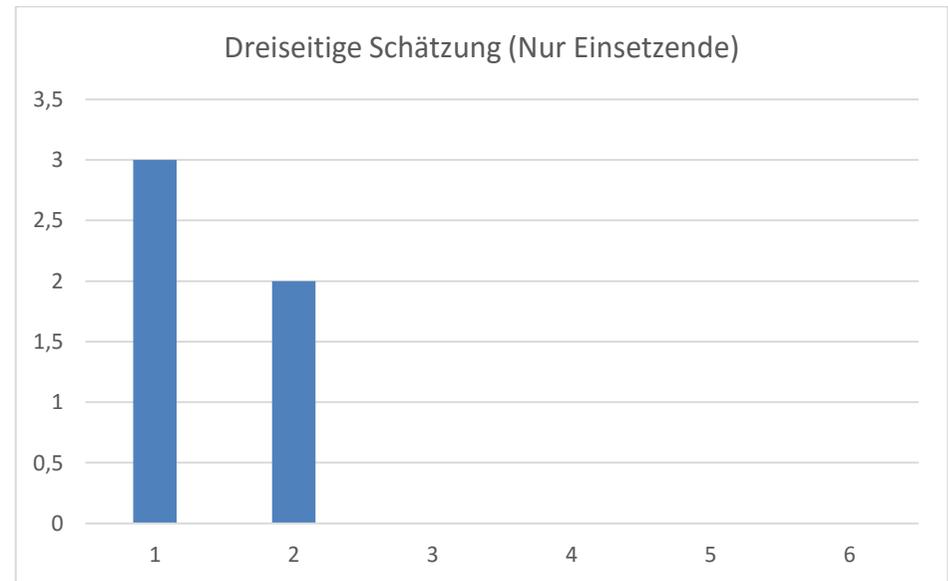
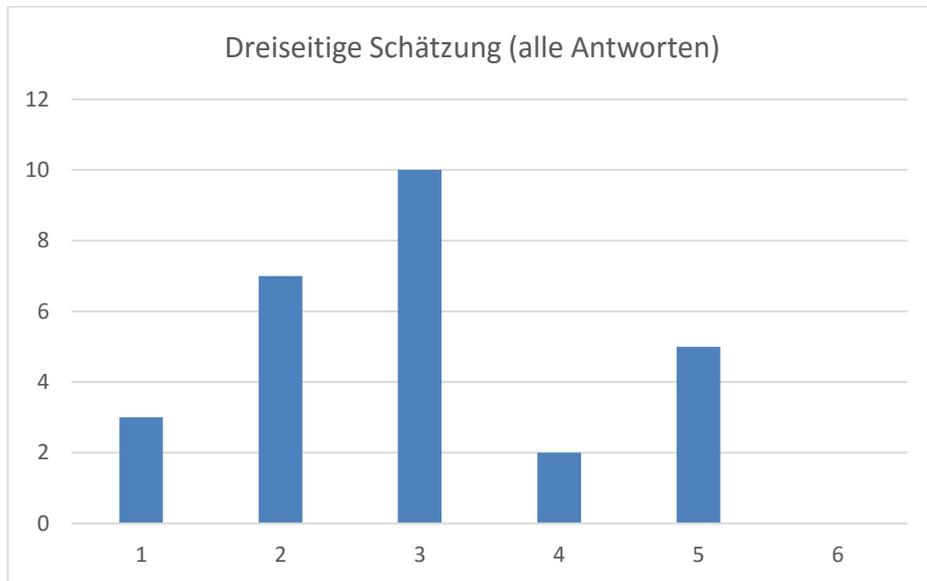
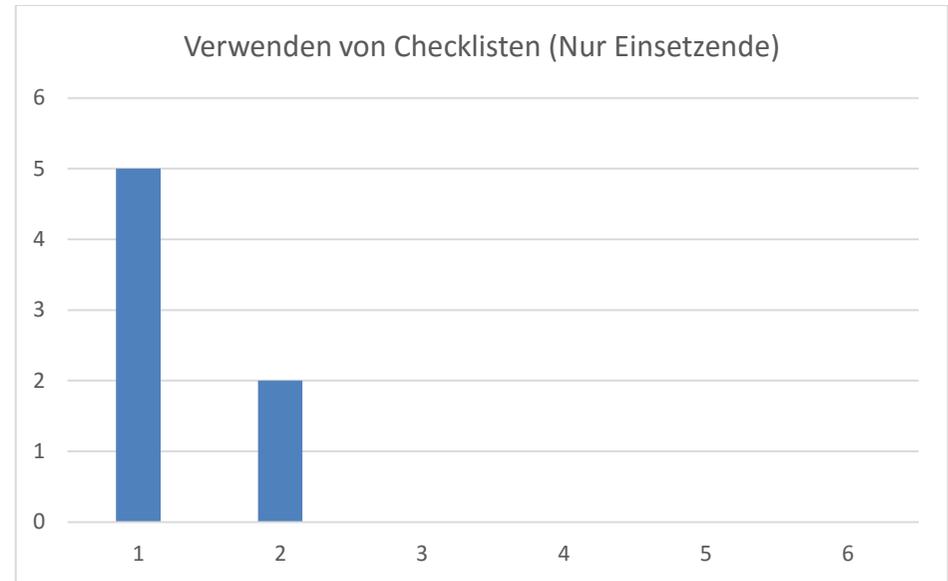
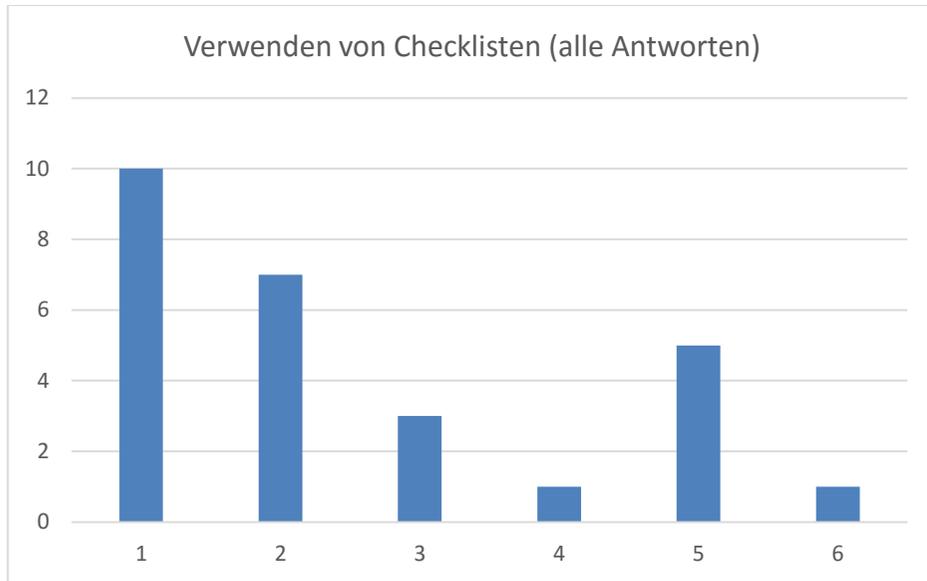
Wenn man die Antworten aller Befragten betrachtet, ergibt sich hinsichtlich der Bewertung, wie sehr die abgefragten Erfolgsfaktoren die Genauigkeit von Schätzungen positiv beeinflussen, ein eher uneinheitliches Bild. Mit Ausnahme von zwei Fällen wurden für alle Erfolgsfaktoren alle Bewertungen von 1 bis 6 abgegeben, in den angesprochenen 2 Fällen fehlte lediglich die Bewertung „6“ („trifft keineswegs zu“).

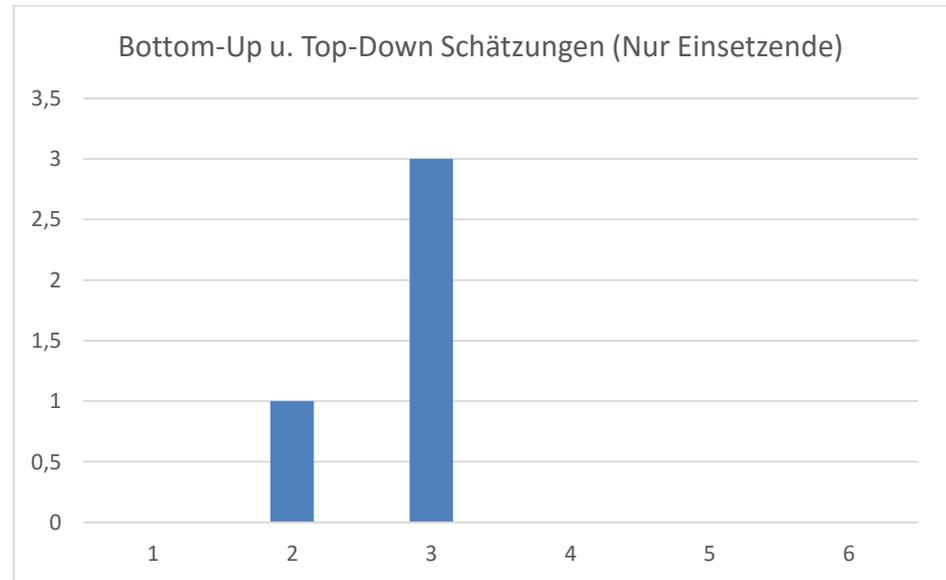
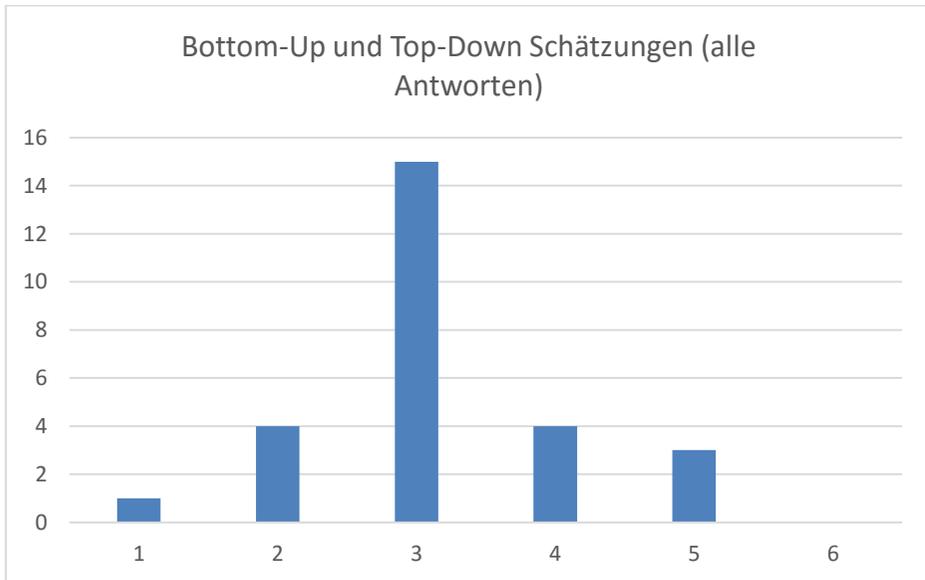
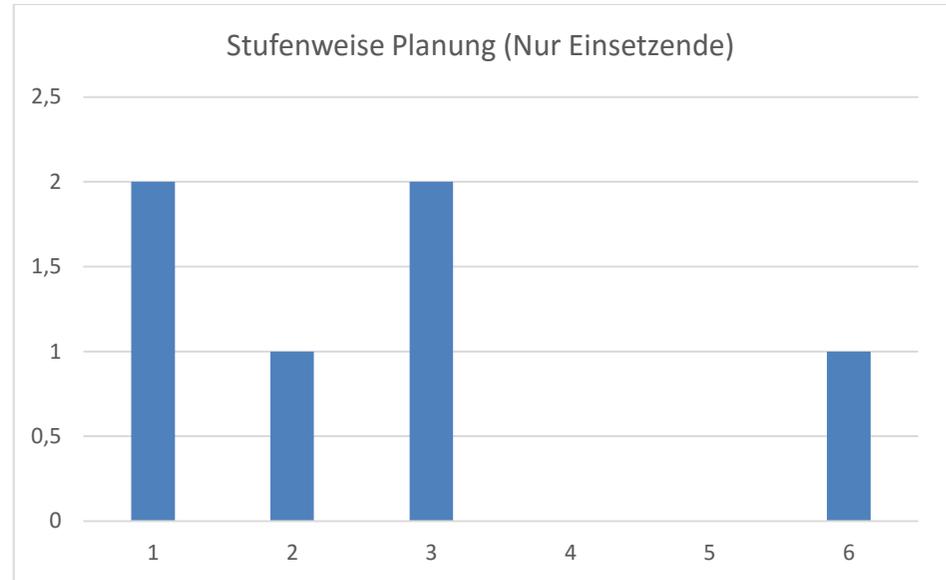
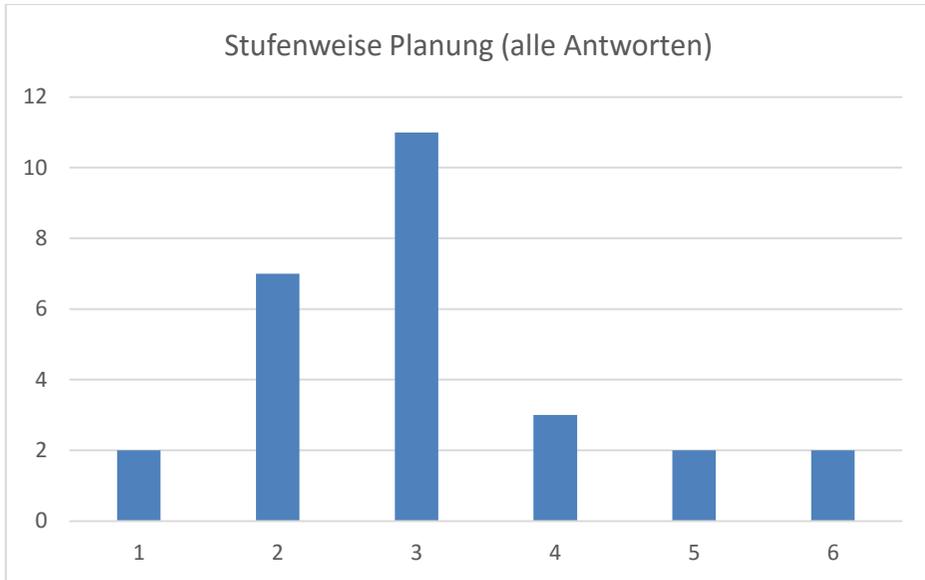
Ein anderes Bild zeigt sich, wenn nur die Antworten jener Teilnehmer berücksichtigt werden, die den jeweils abgefragten Erfolgsfaktor tatsächlich einsetzen. In diesem Fall sind fast alle Teilnehmer der Meinung, dass der jeweilige Erfolgsfaktor die Genauigkeit von Schätzungen positiv beeinflusst („1“- „3“). Der Erfolgsfaktor „Trainingsmöglichkeiten“ wurde in der folgenden vergleichenden Darstellung der Grafiken nicht berücksichtigt, da diese nur von einem Teilnehmer als eingesetzt angegeben wurde.

Abbildung 15: Linke Spalte: die Bewertung aller Teilnehmer, wie stark die genannten Erfolgsfaktoren die Genauigkeit von Schätzungen positiv beeinflussen. Rechte Spalte: die gleiche Bewertung, jedoch nur unter Berücksichtigung der Teilnehmer, die die jeweilige Methode einsetzen. Quelle: eigene Erhebung









8.2.3 Meinung der Befragten zu negativen Einflussfaktoren

Hinsichtlich negativer Einflussfaktoren wurden die Teilnehmer der Befragung zu zwei Faktoren um ihre Meinung gebeten. Auch hier wurde eine Bewertung mittels einer Skala von 1 (trifft absolut zu) bis 6 (trifft keineswegs zu) zugelassen:

- Widersprüchliche Ziele, z.B. Genauigkeit der Schätzung vs. Vorgaben vom Management für den Aufwand
- Zur Verfügung stellen von ergänzenden Informationen, die für die eigentliche Schätzung des Backlog Items nicht notwendig sind

19 der befragten Personen sind der Meinung („1“ oder „2“), dass *Widersprüchliche Ziele, z.B. Genauigkeit der Schätzung vs. Vorgaben vom Management für den Aufwand* einen negativen Einfluss auf die Genauigkeit von Schätzungen haben.

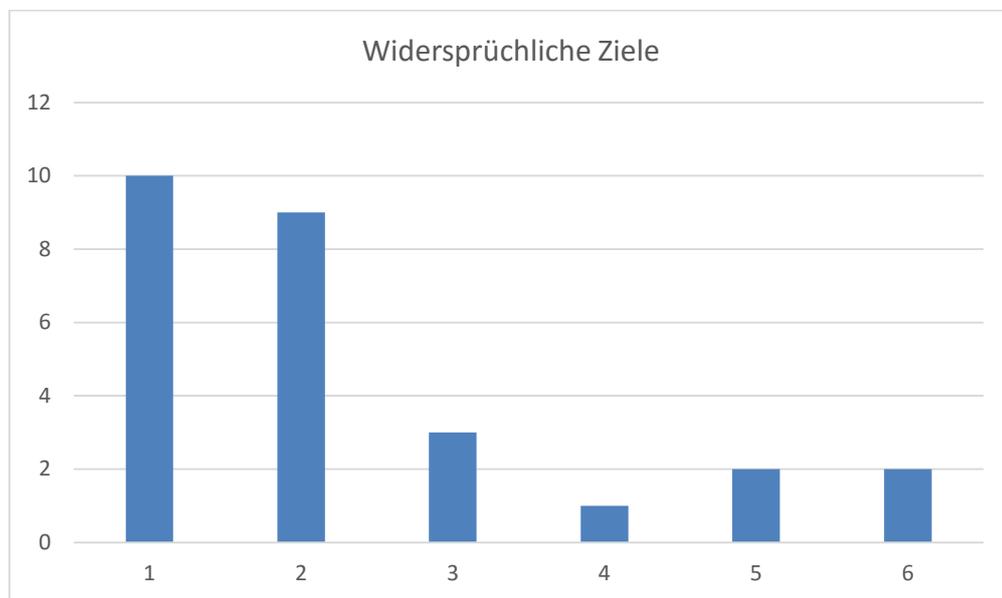


Abbildung 16: Bewertung aller Teilnehmer, wie stark widersprüchliche Ziele die Genauigkeit von Schätzungen negativ beeinflussen.

Zu diesem Einflussfaktor gaben zwei Drittel bzw. 18 Teilnehmer an, dass dieser in Projekten, an denen sie beteiligt waren, einen negativen Einfluss auf die Genauigkeit von Schätzungen hatte.

Nur 5 der befragten Personen sind der Meinung („1“ oder „2“), dass das *Zur Verfügung stellen von ergänzenden Informationen, die für die eigentliche Schätzung des Backlog Items nicht notwendig sind* einen negativen Einfluss auf die Genauigkeit von Schätzungen hat, während 11 der Befragten meinen, dass dies keinen Einfluss („5“ oder „6“) hat.

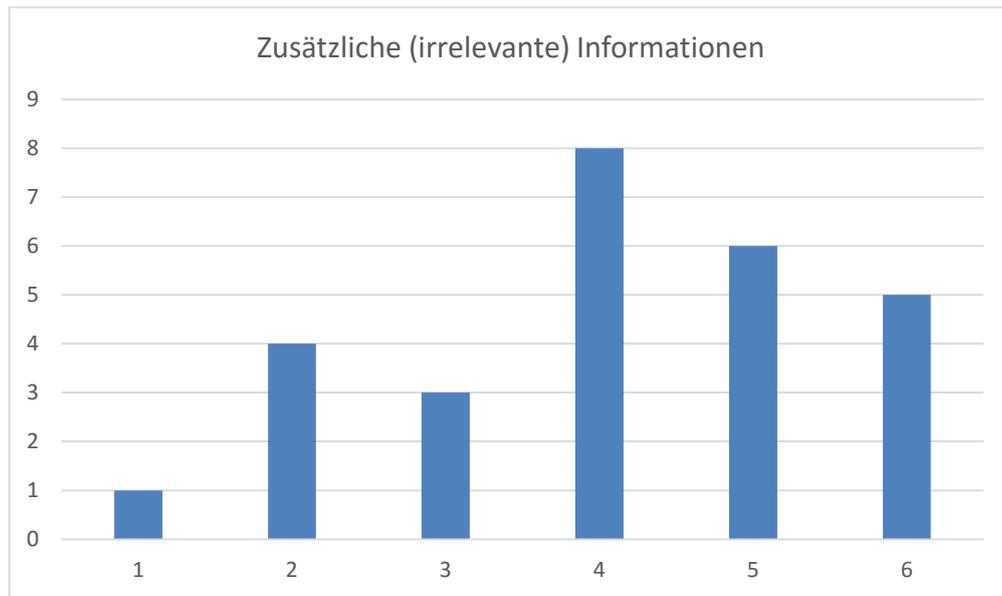


Abbildung 17: Bewertung aller Teilnehmer, wie stark zusätzliche (irrelevante) Informationen die Genauigkeit von Schätzungen negativ beeinflussen.

Neben den explizit abgefragten Faktoren wurden von den Teilnehmern 6 weitere Faktoren genannt, welche die Genauigkeit von Schätzungen negativ beeinflussen:

- Unzureichende Ausformulierung der User Story
- Ungenaue bzw. unzureichende Anforderungen
- Zu große oder zu abstrakte/epische Stories
- Unklare Definition of Done und falsche Erwartungen/Ableitungen der Schätzungsmetrik
- Richtungsänderungen im Projekt
- Mehrere Product Owner für eine Story

8.3 Auswertung der historischen Daten von mehreren Teams im Unternehmen rmDATA

Im Unternehmen rmDATA werden seit der Einführung von agilen Methoden für die Umsetzung von Softwareprojekten die Anforderungen in Form von Backlog-Items in einem Software-System erfasst. Nicht in allen, jedoch in einem großen Teil der Projekte, wird für diese Backlog-Items vor der Umsetzung eine Schätzung erfasst. Die Bewertung der Größe eines Backlog-Items erfolgt dabei je nach Team mittels Story-Points oder in Form von Stunden.

Des Weiteren wird seit etwa 2014, wiederum abhängig vom jeweiligen Team und auch vom Projekt, die Leistungszeiterfassung pro Backlog-Item durchgeführt. Dadurch kann in den jeweiligen Projekten eine Zuordnung des tatsächlichen Aufwands pro Backlog-Item durchgeführt werden. In Kombination mit der beim Backlog-Item erfassten Schätzung bieten diese Daten die Möglichkeit eines Vergleichs des geschätzten und des tatsächlichen Aufwands.

Dieser Vergleich wurde für 16 Projekte unterschiedlicher Größe durchgeführt.

| Projekt | Anzahl Mitglieder im Entwicklungsteam | Gesamtumfang in Stunden | Anzahl ausgewerteter Backlog-Items |
|------------|---------------------------------------|-------------------------|------------------------------------|
| Projekt 9 | 4 | 3560 | 153 |
| Projekt 7 | 8 | 3619 | 47 |
| Projekt 1 | 12 | 2610 | 40 |
| Projekt 3 | 6 | 1928 | 26 |
| Projekt 4 | 8 | 1567 | 106 |
| Projekt 2 | 6 | 1554 | 76 |
| Projekt 5 | 3 | 1160 | 36 |
| Projekt 16 | 9 | 665 | 20 |
| Projekt 14 | 4 | 651 | 19 |
| Projekt 8 | 3 | 498 | 23 |
| Projekt 10 | 3 | 444 | 8 |
| Projekt 12 | 4 | 402 | 9 |
| Projekt 13 | 3 | 366 | 14 |
| Projekt 6 | 4 | 355 | 19 |
| Projekt 15 | 5 | 262 | 6 |
| Projekt 11 | 3 | 182 | 6 |

Tabelle 15: Übersicht über die ausgewerteten Projekte, Anzahl der Teammitglieder, Gesamtaufwand und Anzahl der ausgewerteten Backlog-Items, sortiert nach Projektgröße. Quelle: eigene Erhebung

In der Auswertung berücksichtigt sind jeweils nur jene Backlog-Items, für die es eine Schätzung und einen zuordenbaren tatsächlichen Aufwand gibt. In sämtlichen Projekten ist nur ein Teil des Gesamtaufwands in Form von Backlog-Items angefallen. Projektmanagement-Tätigkeiten, interne Besprechungen, teilweise auch Tätigkeitsbereiche wie Test und Stabilisierung sind nicht als Backlog-Item abgebildet und daher in der Auswertung nicht enthalten. Somit kann im

Folgenden nur eine Aussage über die Genauigkeit der Schätzung der einzelnen Backlog-Items, nicht jedoch über den Gesamtaufwand in einem Projekt gemacht werden.

8.3.1 Eingesetzte Metriken für die Schätzung der Größe von Backlog-Items

In den untersuchten Projekten wurden drei Arten von Metriken für die Größe von Backlog-Items eingesetzt. Diese sind im Folgenden, ebenso wie die jeweilige Vorgehensweise für die Auswertung der Schätzgenauigkeit, beschrieben:

Story Points: In zwei Teams wurde in der Vergangenheit die Größe von Backlog-Items mittels Story Points als relatives Maß geschätzt. Das heißt, dass ein Story Point keine konkrete konstante Größe hat, sondern einen abstrakten Wert darstellt (siehe Kapitel 5.1 Schätzung mit Story Points).

Damit in diesen Projekten ein Vergleich zwischen der Schätzung (in Story Points) und dem tatsächlichen Aufwand (in Stunden) erfolgen kann, ist es notwendig, einen Story Point in einen Stundenwert für dieses Projekt umzurechnen. Dies erfolgte derart, dass die Quadratsumme aller Abweichungen (zwischen umgerechneter Schätzung und tatsächlichem Aufwand) minimal wird. Dadurch wird jener Wert für die Umrechnung von Story Points in Stunden gefunden, der in Summe die kleinste Differenz bewirkt und somit die beste Annäherung darstellt.

Für die Auswertung der Schätzgenauigkeit in diesen Projekten erfolgte dann die Umrechnung der Story Points mit dem wie zuvor beschrieben gefundenen Wert und anschließende Differenzbildung.

Story Points als Zeiteinheit: In mehreren Teams ist es üblich, dass zwar in Story Points geschätzt wird, diese aber als Synonym für einen festen Stundenwert verstanden werden. Dies sind je nach Team 6 oder 8 Stunden. Für die Auswertung der Schätzgenauigkeit in diesen Projekten erfolgte eine einfache Umrechnung mit dem entsprechenden Faktor und anschließende Differenzbildung.

Tatsächliche Zeit: In zwei Teams wird aktuell in tatsächlicher Zeit geschätzt. Dies ist jener Stundenwert, der für das Umsetzen eines Backlog Items tatsächlich benötigt wird (d.h. Ablenkungen durch andere Tätigkeiten werden bei dieser Schätzung nicht berücksichtigt). Dabei werden aber nicht alle beliebigen Stundenwerte verwendet, sondern eine Skala mit folgenden konkreten Werten:

1, 2, 4, 8, 12, 16, 24, 40

Das Verwenden dieser Skala hat zwei Vorteile: Je größer der Aufwand für die Umsetzung eines Backlog Items ist, desto unsicherer ist im Allgemeinen die Schätzung. Diese Unsicherheit wird durch die größeren werdenden Abstände zwischen den Werten wiedergegeben. Andererseits ersparen die konkreten Werte allzu lange Diskussionen über die „richtige“ Zahl für die Schätzung. Die Erfahrung hat gezeigt, dass unter den Schätzenden Experten sehr schnell Konsens darüber hergestellt werden kann, ob ein Backlog Item in 4 Stunden abgeschlossen werden kann, oder ob doch 8 Stunden benötigt werden. Ob der Aufwand bei 6, 6 ½ oder 7 Stunden liegt, würde aber wesentlich längere Diskussionen auslösen.

Für die Auswertung der Schätzgenauigkeit in diesen Projekten erfolgte eine einfache Differenzbildung zwischen Schätzung und tatsächlichem Aufwand.

Diese Auswertung der historischen Daten für Projekte hat gezeigt, dass es für die meisten Einflussfaktoren auf die Schätzgenauigkeit nicht möglich ist, deren konkrete Auswirkung auf die Genauigkeit von Schätzungen mit den vorliegenden Daten zu belegen. Der Grund dafür ist einerseits in dem Umstand zu suchen, dass die betrachteten Einflussfaktoren niemals isoliert in bestimmten Projekten gewirkt haben.

Andererseits haben bestimmte Faktoren, wie zum Beispiel die Kombination der Schätzung von mehreren Experten in allen Projekten gewirkt, da in allen Teams und allen Projekten die Schätzungen von mehreren Personen durchgeführt wurden.

8.3.2 Auswirkungen von Erfolgsfaktoren in den untersuchten Projekten

Kombination von Schätzungen mehrerer Experten

Wie von (Jorgensen, 2004) angeführt, erhöht die Kombination von Schätzungen unterschiedlicher Experten die Genauigkeit von Schätzungen. Um diesen Vorteil zu nutzen, ist die Mittelung von bereits 3-5 Expertenschätzungen ausreichend, wenn diese Schätzungen nicht zu stark korrelieren. Diese geringe Korrelation ist auf Grund der eingesetzten Schätzverfahren in allen Teams gegeben, da die Schätzungen jeweils unabhängig abgegeben und bei Differenzen danach diskutiert werden. Die Art der Kombination der Schätzungen ist zweitrangig, weshalb durch diese keine wesentliche Auswirkung auf die Schätzungen zu erwarten ist (siehe Kapitel 6.3.2 *Methoden, um den Schätzprozess zu unterstützen*).

Beispielhaft werden nachfolgend die Auswertungen der Schätzungen von zwei unterschiedlichen Projekten dargestellt; in einem Projekt wurden die Schätzungen von 3 Personen, im zweiten Projekt die Schätzungen von 8 Personen kombiniert.

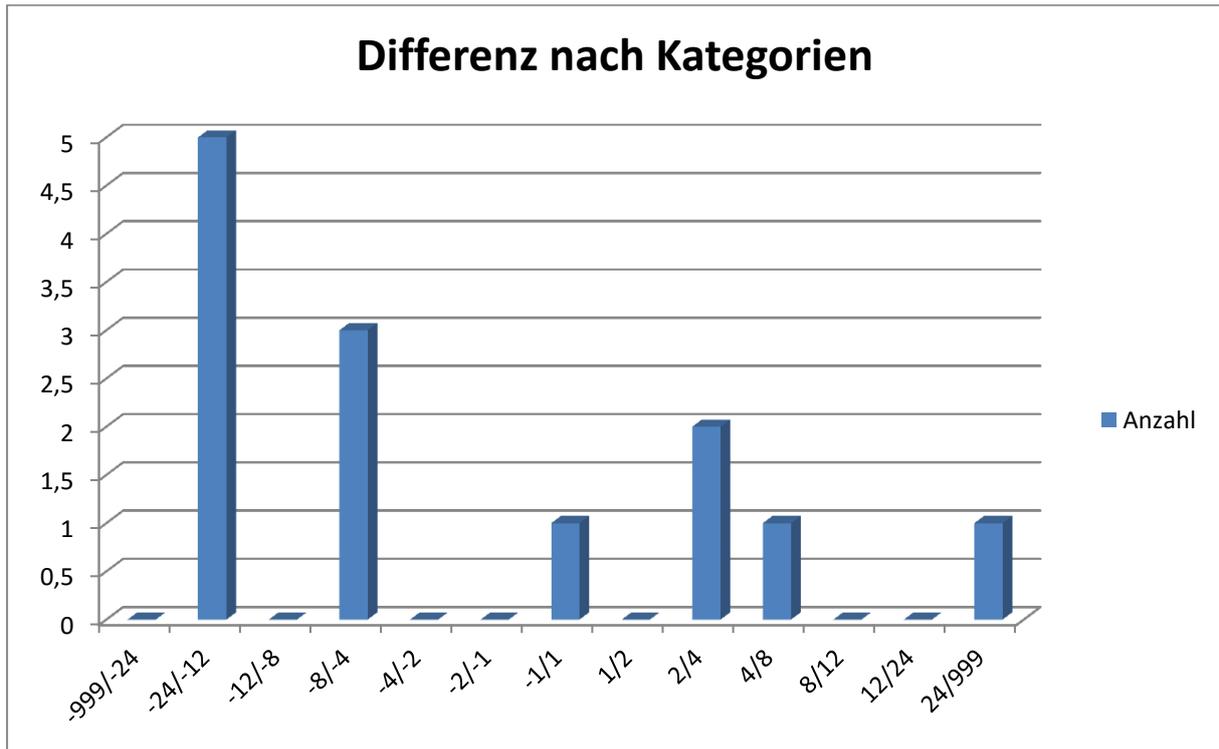


Abbildung 18: Die Auswertung von Projekt 13 mit einem Gesamtumfang von 366 Stunden und 3 Teammitgliedern.

Die Abbildung zeigt eine Darstellung der Differenz von tatsächlichem Aufwand und Schätzung. Die Differenzen sind nach der Größe der Abweichung kategorisiert, wobei die Kategorien die Größe der Abweichung von Schätzung und tatsächlichem Aufwand widerspiegeln:

| | |
|----------|---|
| -999/-24 | tatsächlicher Aufwand war zwischen 999h und 24h geringer als geschätzter Aufwand |
| -24/-12 | tatsächlicher Aufwand war zwischen 24h und 12h geringer als geschätzter Aufwand |
| ... | |
| -1/1 | tatsächlicher Aufwand war zwischen 1h geringer und 1h höher als geschätzter Aufwand |
| ... | |
| 12/24 | tatsächlicher Aufwand war zwischen 24h und 24h höher als geschätzter Aufwand |
| 24/999 | tatsächlicher Aufwand war zwischen 24h und 999h höher als geschätzter Aufwand |

Das bedeutet, links in der Abbildung finden sich User Stories, für die die Schätzung zu hoch war, rechts jene, für die die Schätzung zu niedrig war.

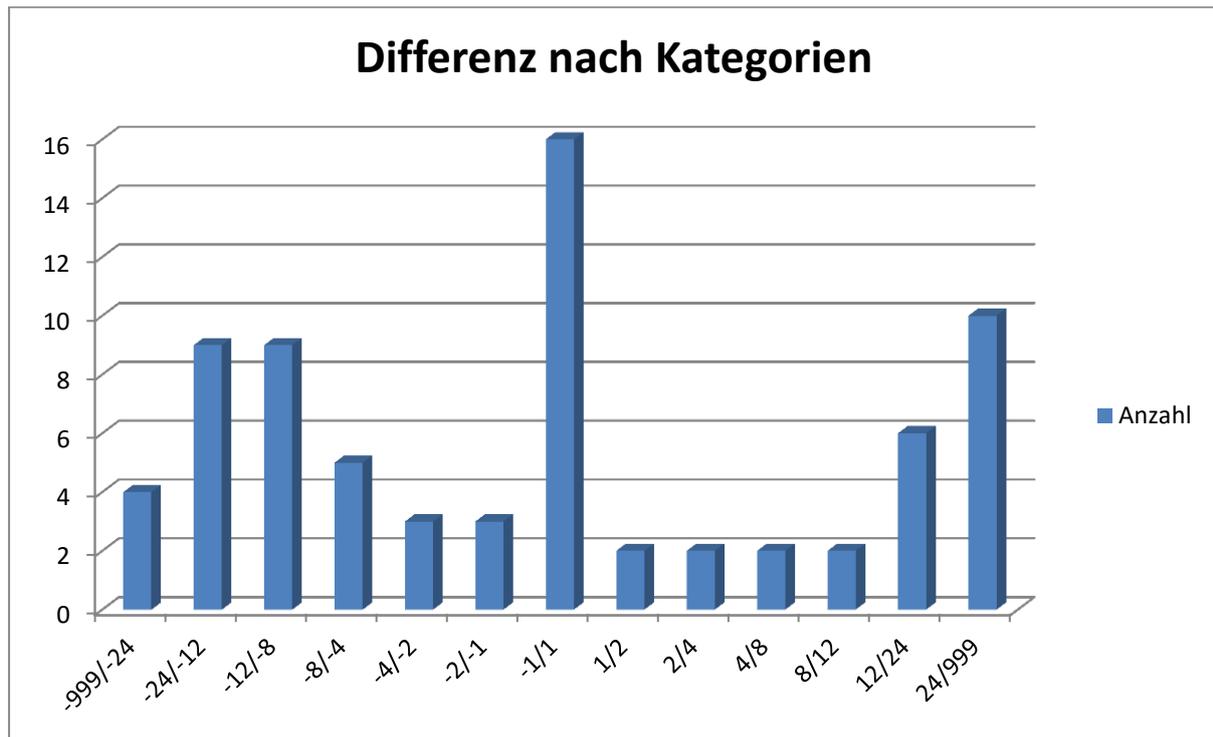


Abbildung 19: Die Auswertung von Projekt 7 mit einem Gesamtumfang von 3619 Stunden und 8 Teammitgliedern.

Abbildung 19 zeigt zwar, dass in dem Projekt mit 8 Teammitgliedern die größte Anzahl (16) der User Stories auf +/- eine Stunde genau geschätzt wurde, und aus Abbildung 18 geht hervor, dass in dem Projekt mit 3 Teammitgliedern die höchste Anzahl (5) von User Stories um 12-24 Stunden zu hoch geschätzt wurde. Die Schätzung in Projekt 13 war im Mittel um 87% zu niedrig, der Median lag bei 33% zu hoch, die Schätzung in Projekt 7 im Mittel auf 0% genau, der Median lag bei 24% zu hoch. Die Standardabweichung der Schätzungen bei Projekt 7 beträgt 74%, bei Projekt 13 400%. Sämtliche Indikatoren für die Genauigkeit der Schätzung (Anzahl der Stories mit geringer Abweichung, Mittelwert der Abweichung und Standardabweichung des Mittelwerts, Median der Abweichung) sind somit bei Projekt 7 mit der größeren Anzahl der Teammitglieder besser.²¹

Dieser Einzelvergleich kann aber lediglich als Indiz, jedoch nicht als Beleg herangezogen werden, dass die Kombination einer größeren Anzahl von Expertenschätzungen die Schätzgenauigkeit erhöht.

Verwenden von Checklisten für Schätzungen

Interviewpartner 4 hat angegeben, dass für die Schätzungen in seinem Team Checklisten verwendet werden. Wiederum ist es (Jorgensen, 2004), der anführt, dass das Verwenden von

²¹ Bei der Standardabweichung in Projekt 13 ist zu berücksichtigen, dass der hohe Wert durch einen Ausreißer mit 1.467% Differenz entsteht. Ohne diesen Ausreißer läge die Standardabweichung in Projekt 13 bei 33%.

Checklisten die Genauigkeit von Schätzungen erhöht (siehe Kapitel 6.3.2 *Methoden, um den Schätzprozess zu unterstützen*).

In einem der untersuchten Projekte des Teams von Experte 4 ist es wiederum so, dass die meiste Anzahl der User Stories auf +/- eine Stunde genau geschätzt wurden, in einem weiteren Projekt des gleichen Teams wurde aber keine der Stories mit einer geringeren Abweichung als 4 Stunden geschätzt:

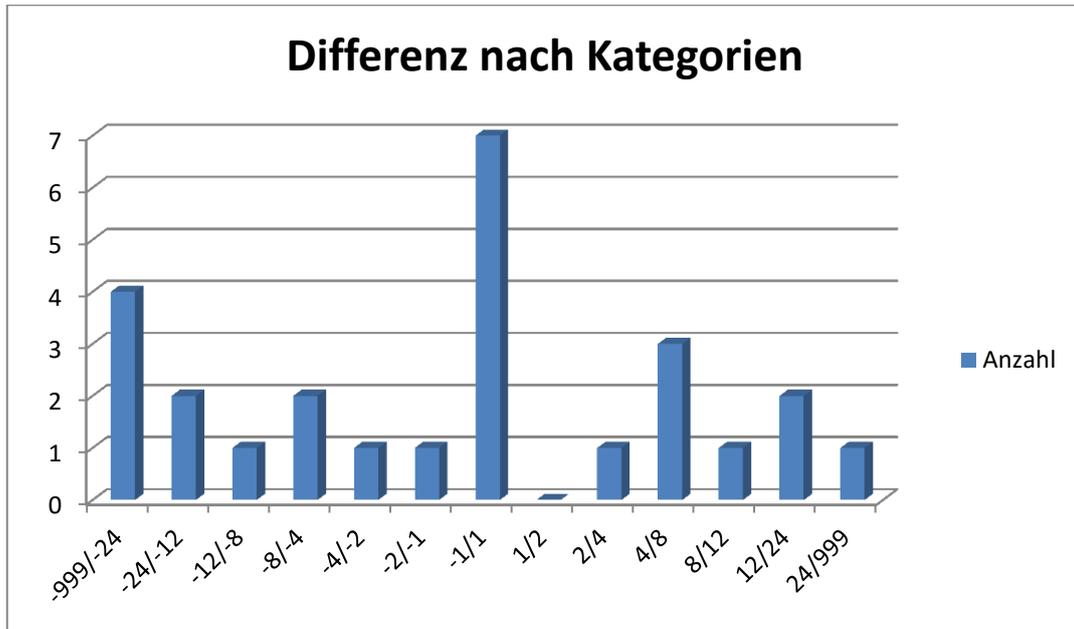


Abbildung 20: Darstellung der Differenzen in einem der Projekte, in denen Checklisten eingesetzt werden (Projekt 15) Quelle: eigene Erhebung.

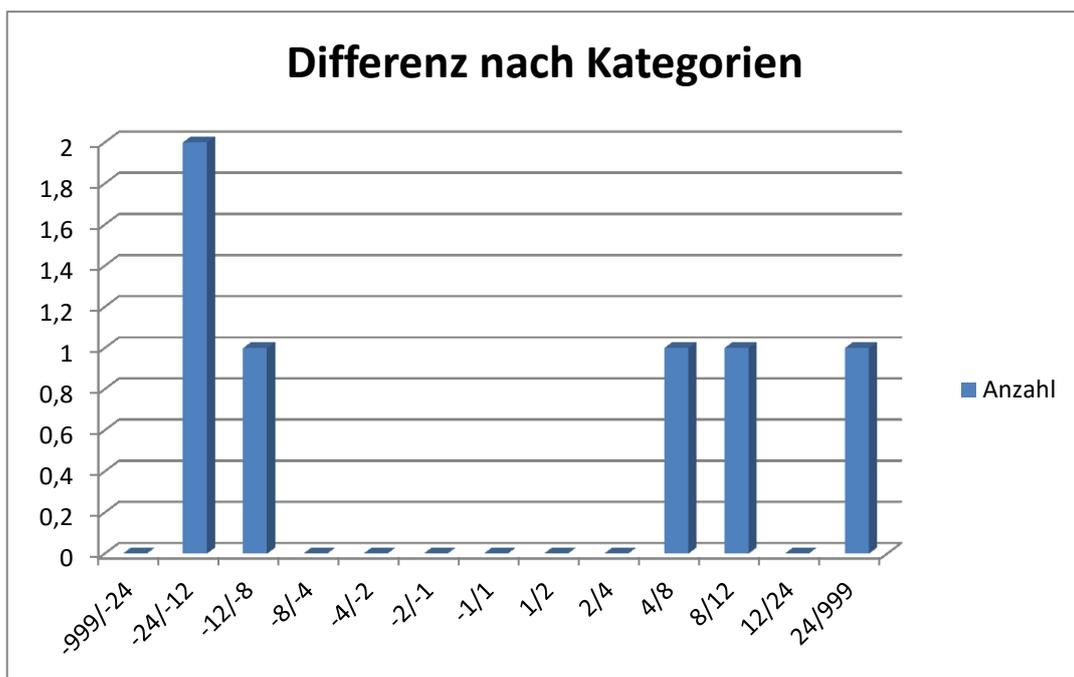


Abbildung 21: Darstellung der Differenzen in einem der Projekte, in denen Checklisten eingesetzt werden (Projekt 16) Quelle: eigene Erhebung.

In Projekt 15 war die Schätzung im Mittel um 8% zu hoch, der Median lag bei 13% zu hoch; in Projekt 16 war die Schätzung im Mittel um 3% zu hoch, der Median lag bei 15% zu hoch.

Details zu den Auswertungen der obigen Projekte finden sich im Anhang. In beiden Projekten wurden Checklisten für die Schätzung eingesetzt. Die Genauigkeit der Schätzungen stellt sich in den Projekten konträr dar, was die Anzahl der Stories mit geringer Abweichung betrifft. Die mittlere Abweichung der Schätzungen und der Medianwert der Abweichungen wiederum sind in diesen beiden Projekten, im Vergleich zu Projekten, in denen keine Checklisten verwendet werden, niedrig.

Die Schätzungen in einem dritten Projekt des gleichen Teams, von dem Checklisten bei Schätzungen verwendet werden, sind mit Abweichungen, die im Mittel um 29% zu hoch und deren Median bei 31% zu hoch liegt, nicht signifikant genauer als Projekte von anderen Teams, die keine Checklisten verwenden.

Die Standardabweichungen der Schätzungen sind mit 34%, 46% und 72% in diesen Projekten im Vergleich zu allen anderen Projekten gering.

Schätzung mit Story Points

Von einem der Experten (Interview 1) wurde angegeben, dass das Schätzen in Stunden die Genauigkeit von Schätzungen positiv beeinflusst, während von einem zweiten Experten (Interview 2) angegeben wurde, dass das Schätzen mit Story Points einen negativen Einfluss auf die Genauigkeit von Schätzungen habe (*„Schätzungen mit Story-Points [...] Hat es früher auch gegeben, wobei wir ehrlich gesagt mit dem, in dieser Phase damals eigentlich nur Probleme gehabt haben.“*).

Diese Aussagen lassen sich durch die ausgewerteten Daten stützen. In jenen Projekten, in denen mit tatsächlichen Stunden geschätzt wurde, nimmt die Anzahl der User Stories pro Kategorie ab, je größer die Abweichung ist, die durch die jeweilige Kategorie repräsentiert wird. In den Projekten, in denen mit „echten“ Story Points geschätzt wurde, sind die Abweichungen dagegen größer bzw. gleichmäßiger über alle Kategorien verteilt. Ein gutes Beispiel für diese Beobachtungen zeigen die folgenden zwei Projekte, die im gleichen Zeitraum und durch teilweise die gleichen Personen geschätzt und umgesetzt worden sind. Die Rahmenbedingungen für die beiden dargestellten Projekte sind somit vergleichbar, jedoch wurde einmal mit Story Points als abstrakte Größe und einmal mit tatsächlichen Stunden geschätzt.

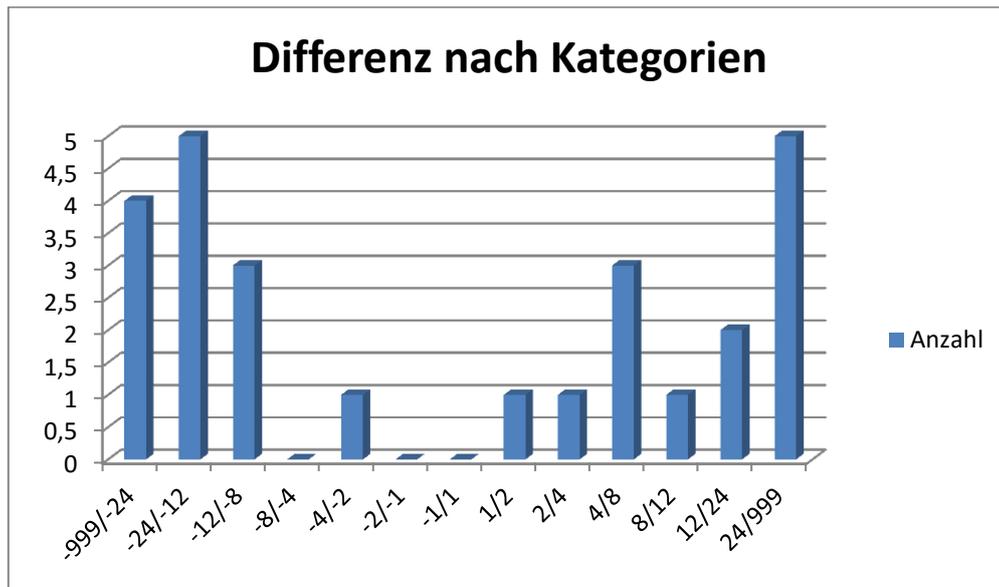


Abbildung 22: Die Auswertung von Projekt 3, in dem mit Story Points geschätzt wurde. Quelle: eigene Erhebung.

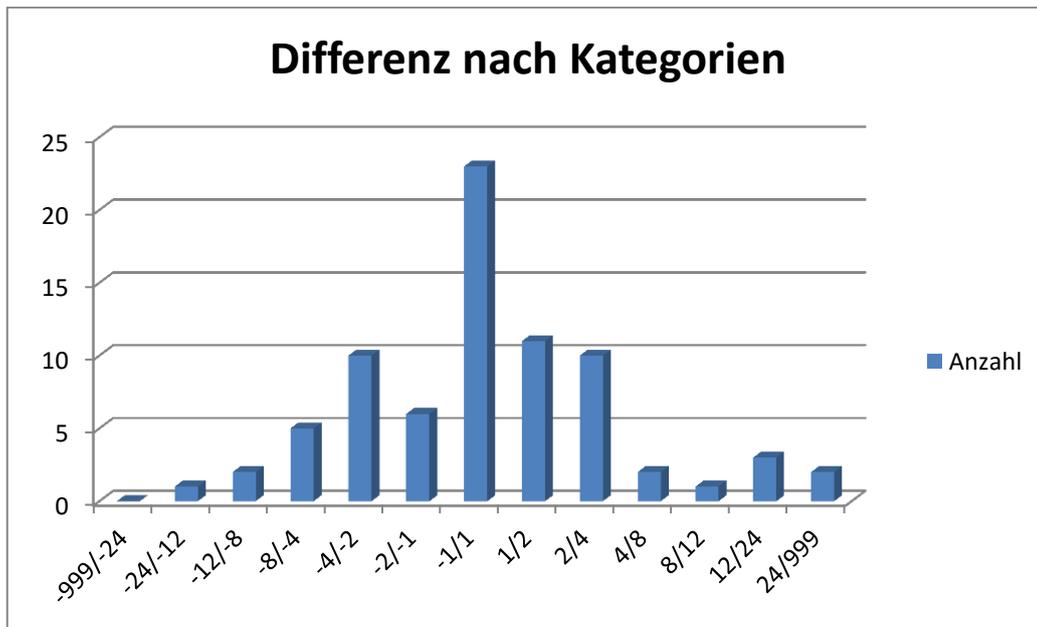


Abbildung 23: Auswertung von Projekt 4, in dem mit tatsächlichen Stunden geschätzt wurde. Quelle: eigene Erhebung

Die beiden weiteren Kriterien für die Genauigkeit von Schätzungen in Projekten, die Mittlere Abweichung und der Medianwert für die Höhe der Abweichungen unterscheiden sich in den Projekten mit Story-Points und den anderen Projekten dagegen nicht signifikant. Die Standardabweichung ist in Projekten, die mit Story Points geschätzt wurden (im Mittel 78%) sogar niedriger als in den Projekten, die auf Basis von idealen Stunden oder Story Points mit Stundenwerten (im Mittel 88%, wenn man Projekt 13 nicht berücksichtigt) geschätzt wurden.

Für das Vorhandensein weiterer Erfolgsfaktoren konnten in den vorliegenden Daten keine ausreichenden Belege bzw. keine signifikanten Unterschiede innerhalb der Projekte belegt

werden. Daher können für weitere Erfolgsfaktoren keine Schlüsse aus den vorliegenden Daten gezogen werden.

9 FAZIT UND AUSBLICK

In der vorliegenden Arbeit wurden im ersten Teil durch Literaturrecherche Erfolgsfaktoren ermittelt, welche die Genauigkeit von Schätzungen in Software-Entwicklungsprojekten positiv beeinflussen. Hier wurden insgesamt 13 Faktoren gefunden, die laut den zitierten Quellen einen positiven Einfluss auf die Genauigkeit von Schätzungen haben. Für alle dieser gefundenen Faktoren konnte in den zitierten Quellen ein positiver Einfluss auf die Genauigkeit von Schätzungen belegt werden, wenn auch in unterschiedlichem Maße. So ist zum Beispiel der Effekt der Evaluierung der eigenen Schätzung gering, aber sehr wohl vorhanden. Der Einsatz von Checklisten als Mittel, um die Genauigkeit von Schätzungen zu erhöhen, ist dagegen durch mehrfache Quellen und unterschiedliche Studien belegt.

Des Weiteren wurden 5 Faktoren gefunden, die die Genauigkeit von Schätzungen negativ beeinflussen. In den zitierten Quellen wurde der Einfluss dieser Faktoren auf die Schätzung bestätigt.

Ausgehend von den in der Literatur beschriebenen Erfolgsfaktoren für Schätzungen in agilen Software-Entwicklungsprojekten wurden mehrere Experten in einem mittelständischen Software-Unternehmen zu ihren Erfahrungen mit Schätzungen in agilen Entwicklungsprojekten befragt. Die Auswertung der Experten-Interviews hat einige der in der Literatur beschriebenen Erfolgsfaktoren bestätigt, sowie weitere Faktoren gezeigt, welche die Genauigkeit von Schätzungen positiv beeinflussen: Das *Verwenden von Checklisten*, die *Evaluierung der Schätzungen* und die *Kombination der Schätzung von unterschiedlichen Experten* wurden von einem oder mehreren Interviewpartnern als positive Erfolgsfaktoren genannt.

Aus der Literaturrecherche nicht ermittelte, aber von der Mehrzahl der Experten in den Interviews genannte Erfolgsfaktoren sind die *Erfahrung des Schätzenden*, das *Laufende Controlling des Aufwands* sowie die *detaillierte Beschreibung der Anforderung*. Aus den Experteninterviews wurden insgesamt sieben Erfolgsfaktoren ermittelt, die von mehr als einem der befragten Experten als positiv für die Schätzgenauigkeit in agilen Projekten genannt wurden. Somit bestätigt und ergänzt die Auswertung der Interviews die Erkenntnis aus der Literaturrecherche, dass es Faktoren gibt, welche die Genauigkeit von Schätzungen in Projekten positiv beeinflusst.

Neben den Faktoren, die einen positiven Einfluss auf die Schätzgenauigkeit haben, wurden auch einige Faktoren gefunden, welche die Genauigkeit von Schätzungen negativ beeinflussen, wobei diese teilweise das Gegenstück zu den genannten positiven Einflussfaktoren sind, wie beispielsweise *unklare Anforderungen* oder *keine Evaluierung der eigenen Schätzung*.

Auf Basis der Ergebnisse der Experteninterviews kann die Arbeitshypothese dieser Arbeit, dass der Einsatz von positiven Erfolgsfaktoren in agilen Software-Entwicklungsprojekten die

Schätzgenauigkeit erhöht, bejahend beantwortet werden. Diese Erkenntnis wird auch durch die Auswertung der Befragung aller Mitarbeiter im Unternehmen untermauert, da bei allen neun der abgefragten Erfolgsfaktoren mehr als 2/3 der befragten Personen angegeben haben, dass sie der Meinung sind, diese Faktoren hätten einen positiven Einfluss auf die Genauigkeit von Schätzungen.

Der Ansatz, die Arbeitshypothese der Arbeit durch die Auswertung von historischen Daten in einem mittelständischen Software-Unternehmen zu bestätigen, bzw. die Nullhypothese zu verwerfen, hat zu differenzierende Ergebnisse geliefert.

Um die Genauigkeit von Schätzungen in der Praxis zu dokumentieren und diese zu verifizieren, wurden die Daten von 16 Software-Entwicklungsprojekten von fünf unterschiedlichen Teams in einem Software-Unternehmen analysiert. Die Ergebnisse dieser Analyse wurden in Form von Projektsteckbriefen mit Fakten und Grafiken aufbereitet.

Für drei der Erfolgsfaktoren, welche aus der Literatur und den Interviews der Experten extrahiert wurden, wurde der Einfluss derselben auf die Projekte, für die die Daten ausgewertet wurden, überprüft. Für die drei untersuchten Erfolgsfaktoren konnte an Hand der historischen Daten kein signifikanter Einfluss auf die Genauigkeit der Schätzungen belegt werden, somit hat dieser Teil der Untersuchung die Nullhypothese bestätigt.

Eine Evaluierung von weiteren Erfolgsfaktoren war an Hand der historischen Daten nicht möglich: Die untersuchten Projekte wurden jeweils von mehreren Faktoren beeinflusst, manche Faktoren haben auf mehrere oder alle Projekte gewirkt, oder das Vorhandensein von anderen Faktoren konnte in den untersuchten Projekten nicht belegt werden.

9.1 Fazit

Somit kann festgehalten werden, dass innerhalb des untersuchten Unternehmens bei den Projektmanagern und den Teammitgliedern Konsens darüber herrscht, dass es Erfolgsfaktoren gibt, welche die Genauigkeit von Schätzungen in agilen Software-Projekten positiv beeinflussen. Diese Faktoren decken sich teilweise mit den in der Literatur beschriebenen Erfolgsfaktoren. Manche der laut Literatur vorhandenen Erfolgsfaktoren wurden nicht genannt bzw. im Unternehmen nicht eingesetzt und konnten daher auch nicht beurteilt werden. Die Auswertung der historischen Daten des untersuchten Unternehmens hat nicht erlaubt, die Nullhypothese zu verwerfen und damit die Arbeitshypothese zu bestätigen. Somit ist auf Basis der in der vorliegenden Arbeit herangezogenen historischen Daten nicht zu bestätigen, dass es Erfolgsfaktoren in agilen Software-Entwicklungsprojekten gibt, welche die Genauigkeit von Schätzungen signifikant positiv beeinflussen.

In Einzelfällen haben die untersuchten Daten aber Indizien dafür geliefert, dass die analysierten Faktoren einen positiven Einfluss auf die Genauigkeit der Schätzungen hatten.

9.2 Ausblick

Um die in dieser Arbeit zusammengefassten und aus den Experten-Interviews extrahierten Erfolgsfaktoren für das Erhöhen der Genauigkeit von Schätzungen in agilen Software-Entwicklungsprojekten an Hand der Daten von Projekten zu evaluieren, bietet es sich an, einzelne Erfolgsfaktoren isoliert in durchzuführenden Projekten anzuwenden. Das durchgängige Anwenden einzelner positiver Faktoren und Ausschließen anderer (negativer) Faktoren in einem Software-Projekt sollte sodann eine Bewertung erlauben, ob der jeweilige Erfolgsfaktor einen messbaren Einfluss auf die Genauigkeit von Schätzungen hat. Diese Art der Evaluierung konnte in der vorliegenden Arbeit nicht durchgeführt werden, da der notwendige Beobachtungszeitraum und die Anzahl der Projekte nicht verfügbar waren.

ANHANG A - Projektsteckbriefe

Anhang A enthält einen Überblick über die untersuchten Projekte in Form von so genannten Projektsteckbriefen. Die Steckbriefe enthalten für jedes Projekt die Anzahl der Projektteam-Mitglieder und den Projektmanager.

Es ist angegeben, mit welcher Metrik geschätzt wurde, wie treffend die Schätzung im Mittel war, sowie der Median der Abweichungen der Schätzungen.

Für jedes Projekt ist angegeben, wie hoch der Gesamtumfang des Projekts war und wie viele Backlog-Items mit Schätzung und zuordenbarem Aufwand ausgewertet wurden.

Zuletzt ist für jedes Projekt eine graphische Darstellung der Abweichung der Schätzungen enthalten. Diese stellt dar, wie viele der Schätzungen um den jeweiligen Kategoriewert zu hoch oder zu niedrig waren.

Projekt 1

Anzahl der Projektteammitglieder: 12, davon 8 Entwickler

Projektmanager: Experte 2 (Interview 2)

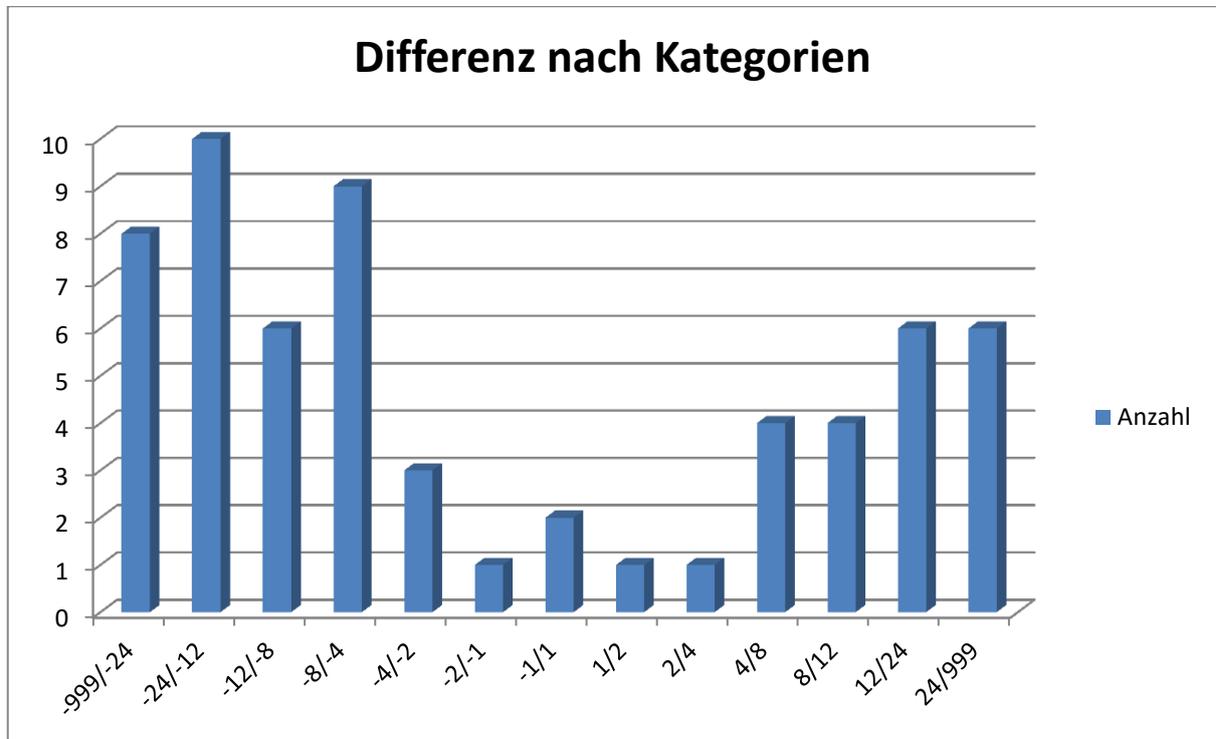
| Mitarbeiter | Mitarbeiter 1 | Mitarbeiter 2 | Mitarbeiter 3 | Experte 2 | Mitarbeiter 4 | Marschall Thomas | Mitarbeiter 5 | Experte 1 | Experte 3 | Mitarbeiter 6 | Mitarbeiter 7 | Mitarbeiter 8 | Gesamtsumme |
|---------------|---------------|---------------|---------------|-----------|---------------|------------------|---------------|-----------|-----------|---------------|---------------|---------------|-------------|
| Summe | 603:55 | 602:13 | 478:02 | 359:10 | 197:55 | 119:39 | 94:48 | 48:42 | 38:29 | 33:30 | 23:39 | 10:13 | 2610:15 |
| Anteil (in %) | 23,1 | 23,1 | 18,3 | 13,8 | 7,6 | 4,6 | 3,6 | 1,9 | 1,5 | 1,3 | 0,9 | 0,4 | 100 |

Schätzgröße: Story Points

Schätzung im Mittel um 3% zu hoch (92% StdDev), Median 38% zu hoch

Gesamtumfang: 2610 Stunden

40 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Schätzung zu hoch

Schätzung zu niedrig

Projekt 2

Anzahl der Projektteammitglieder: 8, davon 6 Entwickler

Projektmanager: Experte 2 (Interview 2)

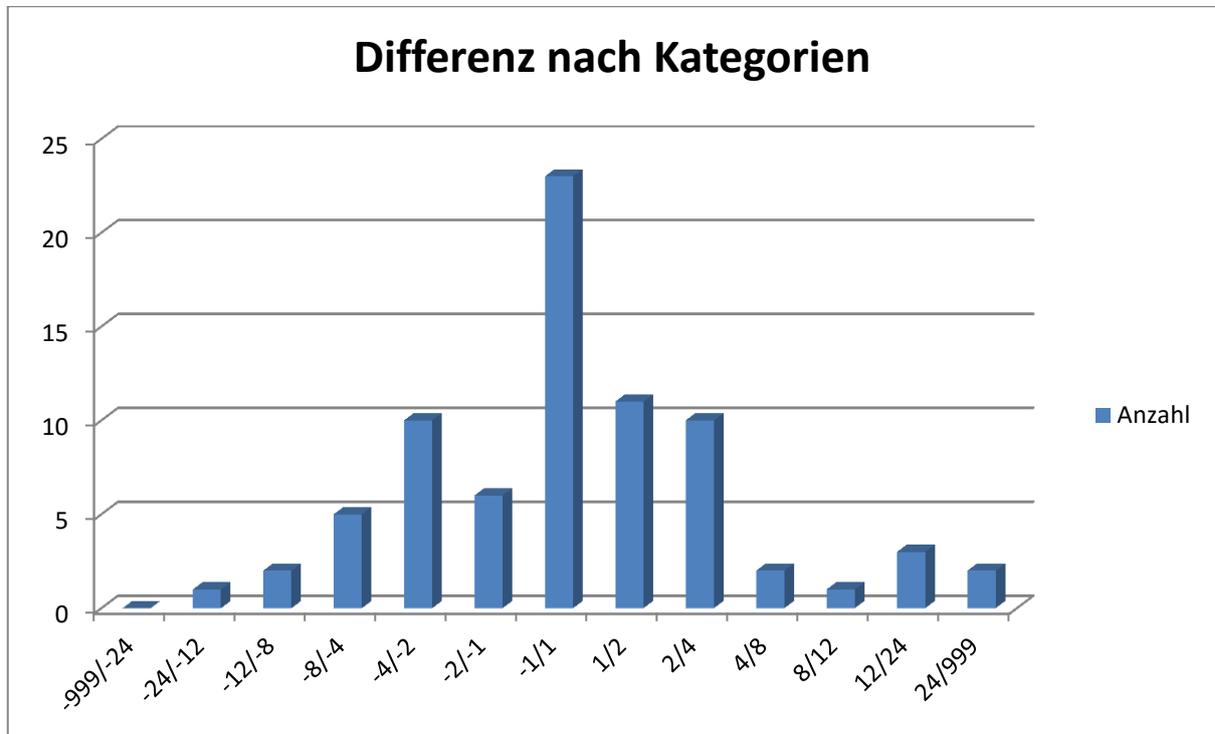
| Mitarbeiter | Mitarbeiter 1 | Experte 2 | Mitarbeiter 7 | Mitarbeiter 9 | Mitarbeiter 2 | Mitarbeiter 6 | Mitarbeiter 4 | Marschall Thomas | Gesamtsumme |
|---------------|---------------|-----------|---------------|---------------|---------------|---------------|---------------|------------------|-------------|
| Summe | 283:31 | 278:31 | 256:39 | 243:41 | 222:05 | 113:44 | 105:15 | 50:53 | 1554:19 |
| Anteil (in %) | 18,2 | 17,9 | 16,5 | 15,7 | 14,3 | 7,3 | 6,8 | 3,3 | 100 |

Schätzgröße: Ideale Stunden

Schätzung im Mittel um 28% zu niedrig (131% StdDev), Median 11% zu niedrig

Gesamtumfang: 1554 Stunden

76 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Schätzung zu hoch

Schätzung zu niedrig

Projekt 3

Anzahl der Projektteam-Mitglieder: 8, davon 6 Entwickler

Projektmanager: Thomas Marschall

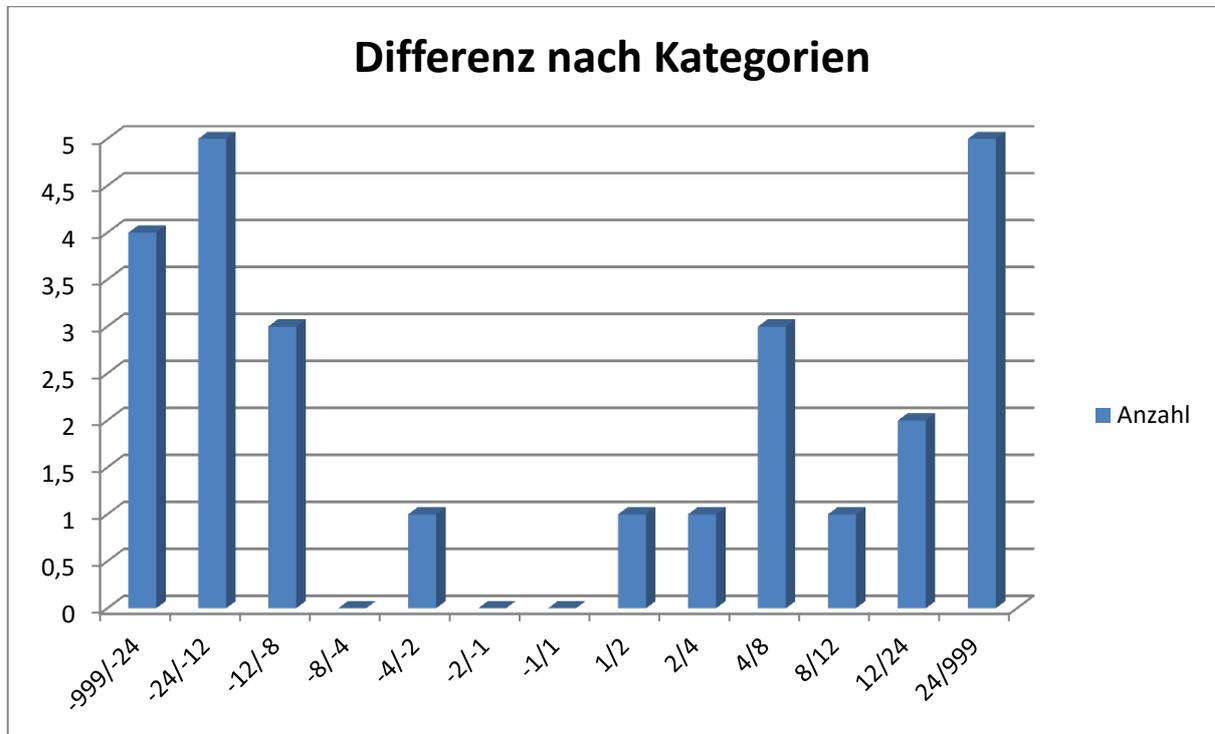
| Mitarbeiter | Mitarbeiter 2 | Mitarbeiter 7 | Marschall Thomas | Mitarbeiter 1 | Experte 2 | Mitarbeiter 6 | Mitarbeiter 9 | Mitarbeiter 4 | Gesamtsumme |
|---------------|---------------|---------------|------------------|---------------|-----------|---------------|---------------|---------------|-------------|
| Summe | 634:14 | 493:57 | 247:39 | 241:04 | 176:09 | 81:03 | 31:55 | 22:38 | 1928:39 |
| Anteil (in %) | 32,9 | 25,6 | 12,8 | 12,5 | 9,1 | 4,2 | 1,7 | 1,2 | 100 |

Schätzgröße: Story Points

Schätzung im Mittel um 0% zu niedrig (81% StdDev), Median 0%

Gesamtumfang: 1928 Stunden

26 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Schätzung zu hoch

Schätzung zu niedrig

Projekt 4

Anzahl der Projektteam-Mitglieder: 10, davon 8 Entwickler

Projektmanager: Experte 2 (Interview 2)

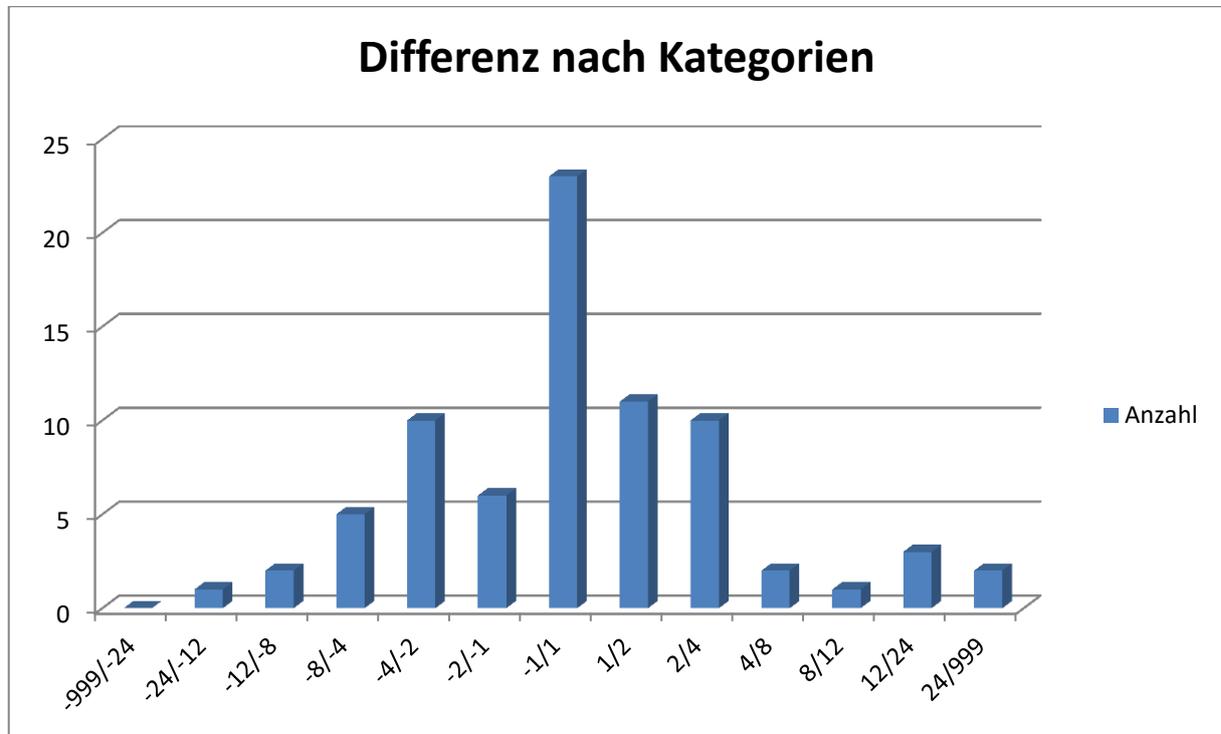
| Mitarbeiter | Mitarbeiter 7 | Experte 2 | Mitarbeiter 6 | Mitarbeiter 4 | Experte 1 | Mitarbeiter 10 | Mitarbeiter 9 | Experte 3 | Mitarbeiter 1 | Marschall Thomas | Gesamtsumme |
|---------------|---------------|-----------|---------------|---------------|-----------|----------------|---------------|-----------|---------------|------------------|-------------|
| Summe | 637:48 | 448:31 | 210:41 | 157:39 | 31:10 | 21:30 | 18:30 | 17:00 | 13:01 | 11:49 | 1567:39 |
| Anteil (in %) | 40,7 | 28,6 | 13,4 | 10,1 | 2,0 | 1,4 | 1,2 | 1,1 | 0,8 | 0,8 | 100 |

Schätzgröße: Ideale Stunden

Schätzung im Mittel um 13% zu niedrig (93% StdDev), Median 7% zu hoch

Gesamtumfang: 1567 Stunden

106 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Schätzung zu hoch

Schätzung zu niedrig

Projekt 5

Anzahl der Projektteam-Mitglieder: 5, davon 3 Entwickler

Projektmanager: Mitarbeiter 12

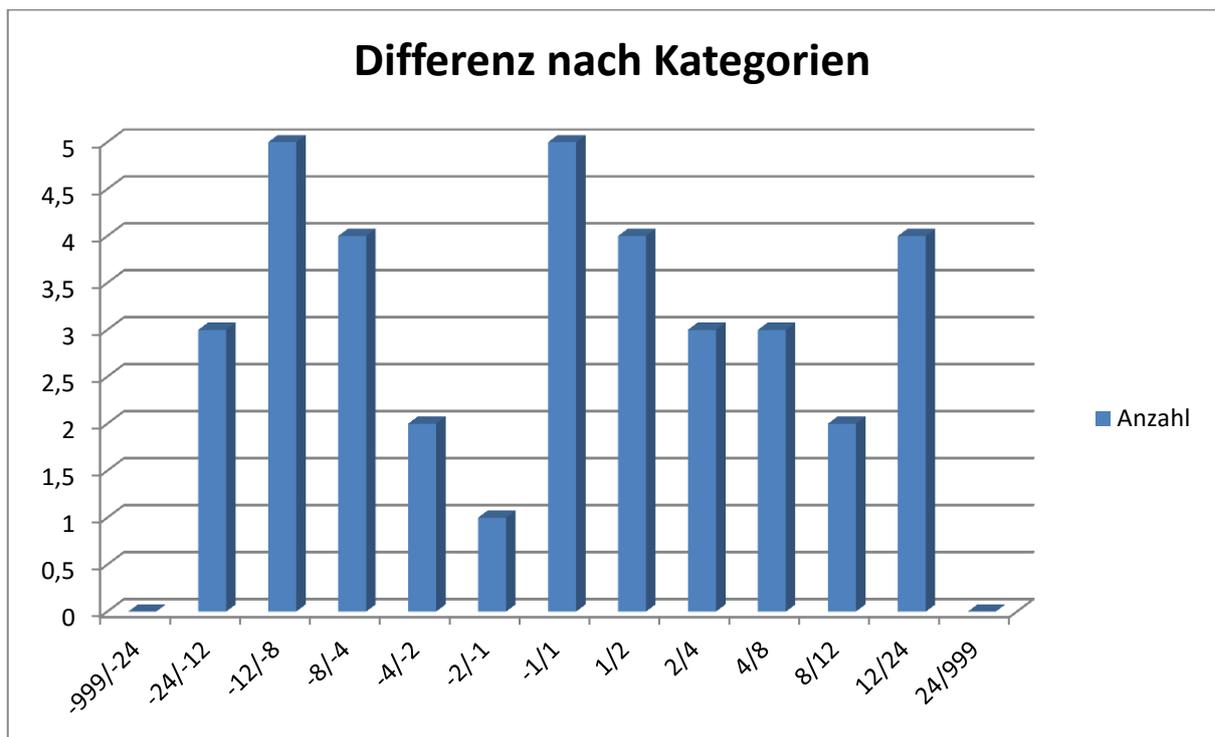
| Mitarbeiter | Experte 3 | Mitarbeiter 11 | Mitarbeiter 12 | Mitarbeiter 13 | Mitarbeiter 14 | Gesamtsumme |
|----------------------|---------------|----------------|----------------|----------------|----------------|----------------|
| Summe | 503:03 | 450:02 | 132:38 | 59:58 | 15:09 | 1160:50 |
| Anteil (in %) | 43,34 | 38,77 | 11,43 | 5,17 | 1,31 | 100,00 |

Schätzgröße: Story Points

Gesamtumfang: 1160 Stunden

Schätzung im Mittel um 0% zu niedrig (62% StdDev), Median 5% zu niedrig

36 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Schätzung zu hoch

Schätzung zu niedrig

Projekt 6

Anzahl der Projektteam-Mitglieder: 8, davon 4 Entwickler (> 10h)

Projektmanager: Mitarbeiter 2

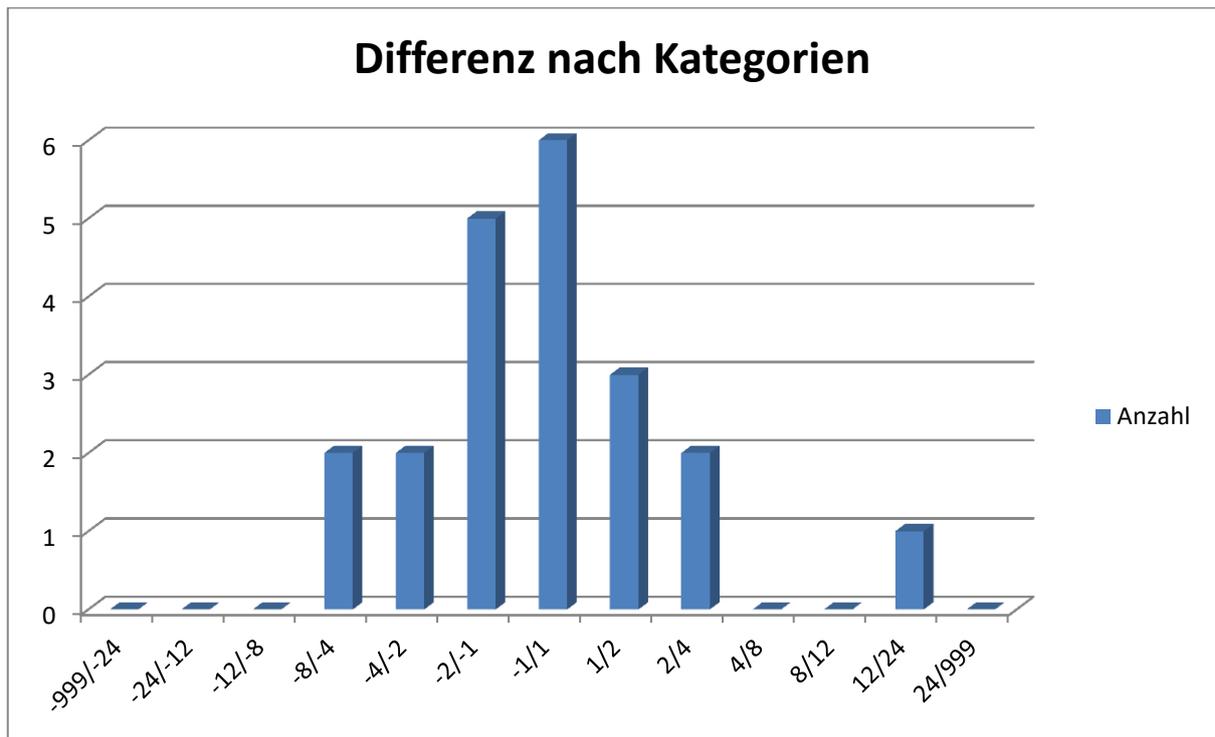
| Mitarbeiter | Experte 3 | Mitarbeiter 11 | Mitarbeiter 12 | Mitarbeiter 4 | Mitarbeiter 2 | Mitarbeiter 1 | Marschall Thomas | Mitarbeiter 14 | Gesamtsumme |
|----------------------|-----------|----------------|----------------|---------------|---------------|---------------|------------------|----------------|-------------|
| Summe | 148:02 | 106:15 | 41:53 | 28:23 | 23:30 | 3:35 | 2:51 | 1:10 | 355:39 |
| Anteil (in %) | 41,62 | 29,87 | 11,78 | 7,98 | 6,61 | 1,01 | 0,80 | 0,33 | 100,00 |

Schätzgröße: Story Points

Schätzung im Mittel um 0% zu niedrig (86% StdDev), Median 9% zu hoch

Gesamtumfang: 355 Stunden

19 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Schätzung zu hoch

Schätzung zu niedrig

Projekt 7 DBE GeoDesktop

Anzahl der Projektteam-Mitglieder: 15, davon 8 Entwickler (>15 h)

Projektmanager: Experte 1 (Interview 1)

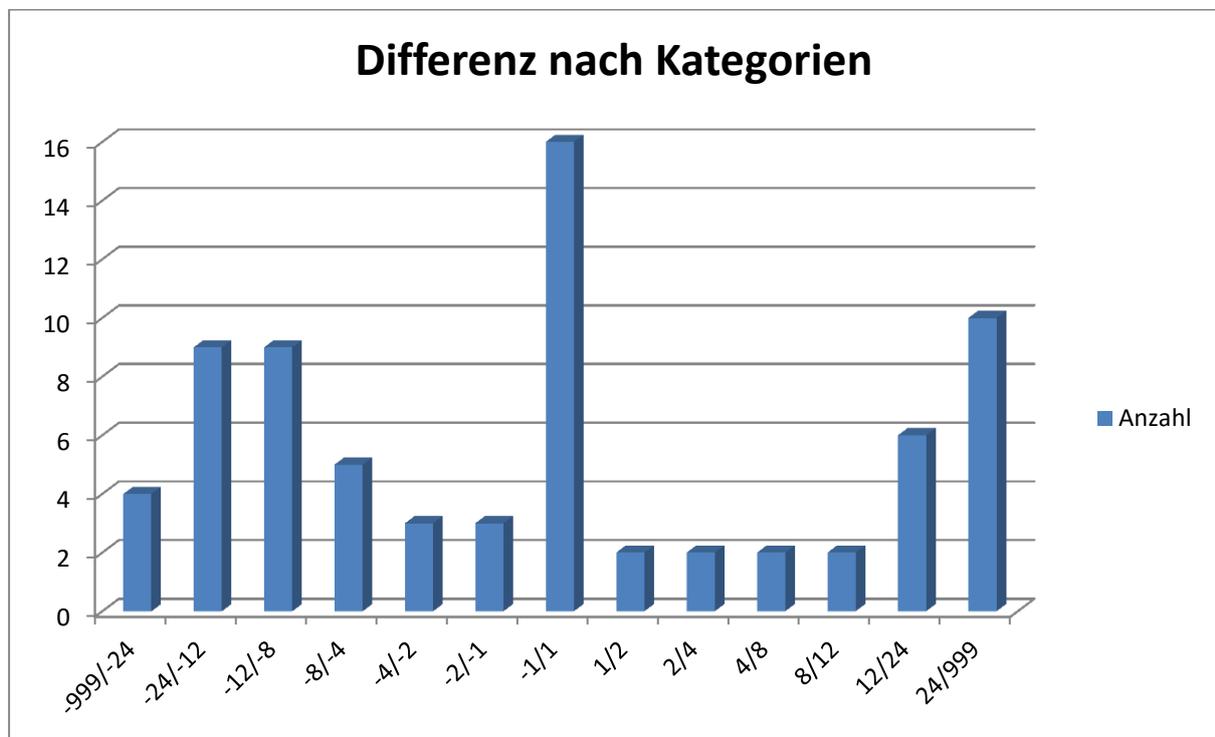
| Mitarbeiter | Experte 1 | Mitarbeiter 15 | Mitarbeiter 16 | Mitarbeiter 17 | Mitarbeiter 18 | Mitarbeiter 19 | Mitarbeiter 20 | Experte 2 | Mitarbeiter 21 | Mitarbeiter 13 | Mitarbeiter 22 | Mitarbeiter 4 | Mitarbeiter 23 | Mitarbeiter 1 | Mitarbeiter 6 | Gesamtsumme |
|----------------------|-----------|----------------|----------------|----------------|----------------|----------------|----------------|-----------|----------------|----------------|----------------|---------------|----------------|---------------|---------------|-------------|
| Summe | 1087 | 505 | 480 | 402 | 302 | 228 | 186 | 145 | 134 | 66 | 27 | 20 | 14 | 12 | 4 | 3619 |
| Anteil (in %) | 30,0 | 14,0 | 13,3 | 11,1 | 8,4 | 6,3 | 5,2 | 4,0 | 3,7 | 1,8 | 0,8 | 0,6 | 0,4 | 0,4 | 0,1 | 100,0 |

Schätzgröße: Story Points

Schätzung im Mittel um 0% zu niedrig (74% StdDev), Median 24% zu hoch

Gesamtumfang: 3619 Stunden

47 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Schätzung zu hoch

Schätzung zu niedrig

Projekt 8

Anzahl der Projektteam-Mitglieder: 4, davon 3 Entwickler

Projektmanager: Thomas Marschall

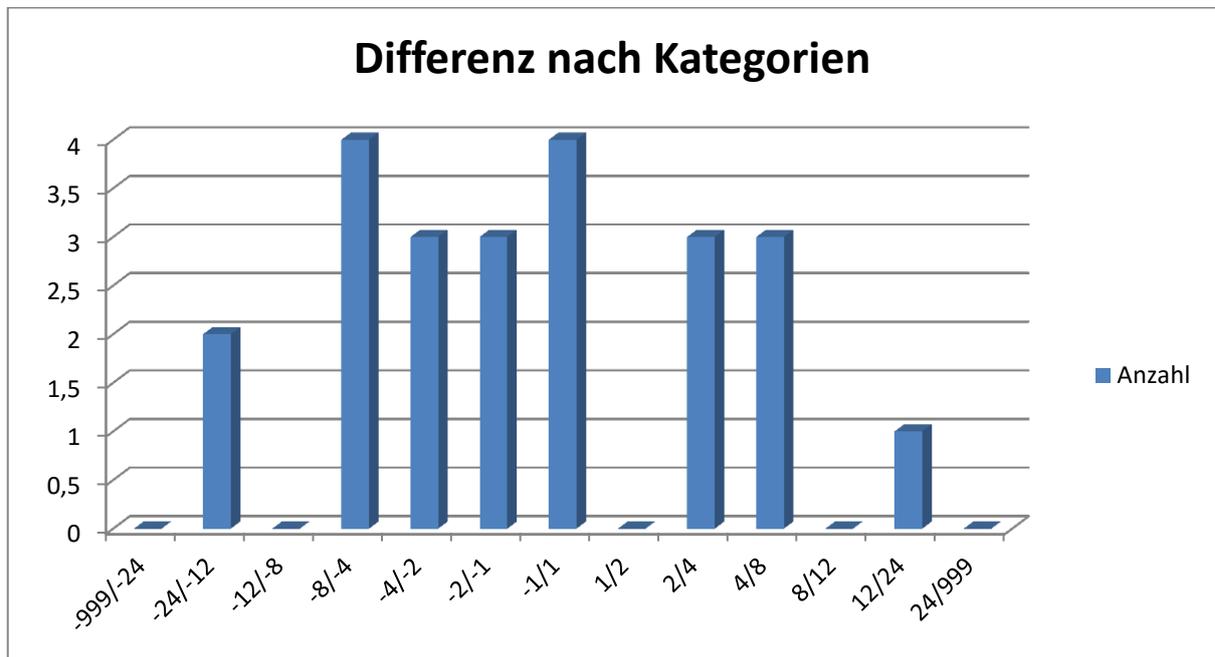
| Mitarbeiter | Mitarbeiter 24 | Experte 3 | Mitarbeiter 1 | Marschall Thomas | Gesamtsumme |
|----------------------|----------------|---------------|---------------|------------------|---------------|
| Summe | 211:34 | 139:28 | 101:26 | 45:43 | 498:11 |
| Anteil (in %) | 42,5 | 28,0 | 20,4 | 9,2 | 100,0 |

Schätzgröße: Story Points

Schätzung im Mittel um 0% zu niedrig (70% StdDev), Median 25% zu hoch

Gesamtumfang:498 Stunden

23 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Schätzung zu hoch

Schätzung zu niedrig

Projekt 9

Anzahl der Projektteam-Mitglieder: 8, davon 4 Entwickler im gesamten Projektzeitraum

Projektmanager: Thomas Marschall

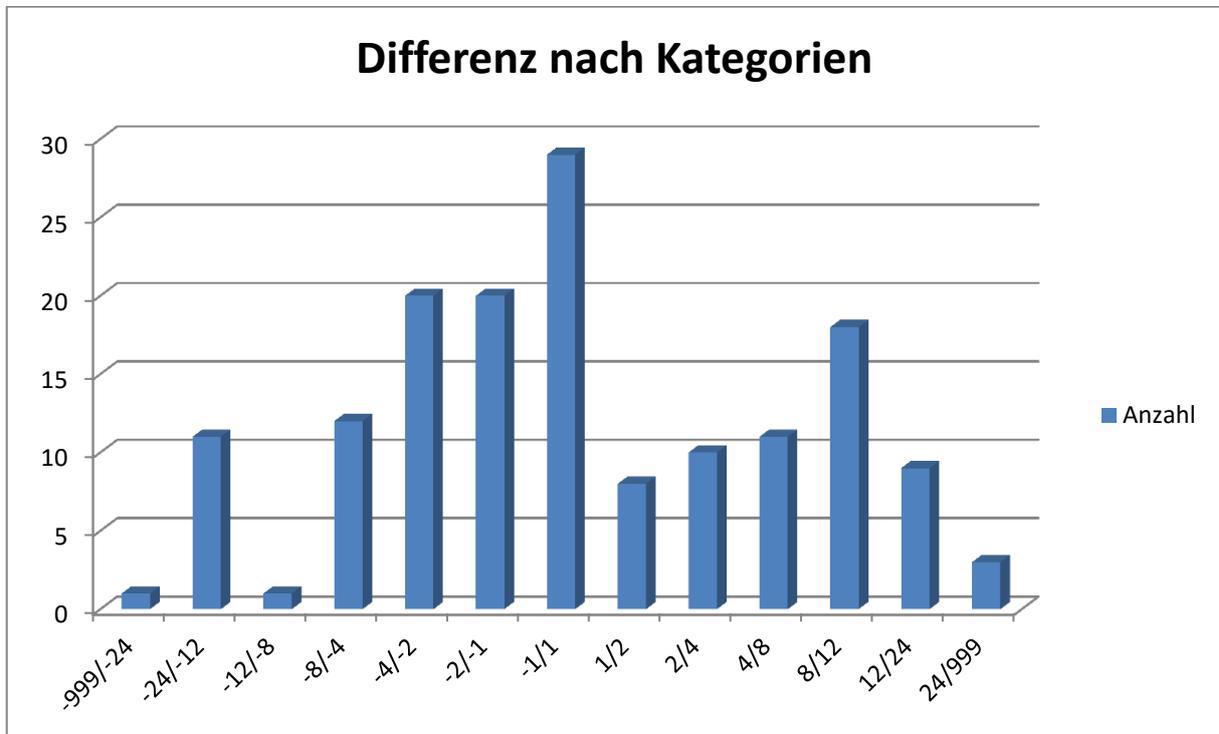
| Mitarbeiter | Mitarbeiter 10 | Experte 3 | Mitarbeiter 9 | Mitarbeiter 1 | Marschall Thomas | Mitarbeiter 2 | Mitarbeiter 7 | Experte 2 | Gesamtsumme |
|---------------|----------------|-----------|---------------|---------------|------------------|---------------|---------------|-----------|-------------|
| Summe | 1035:07 | 922:21 | 688:30 | 574:41 | 251:02 | 55:05 | 19:06 | 13:42 | 3559:34 |
| Anteil (in %) | 29,1 | 25,9 | 19,3 | 16,1 | 7,1 | 1,5 | 0,5 | 0,4 | 100 |

Schätzgröße: Ideale Stunden

Schätzung im Mittel um 36% zu niedrig (163% StdDev), Median 10% zu hoch

Gesamtumfang: 3560 h

153 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Schätzung zu hoch

Schätzung zu niedrig

Projekt 10

Anzahl der Projektteam-Mitglieder: 3

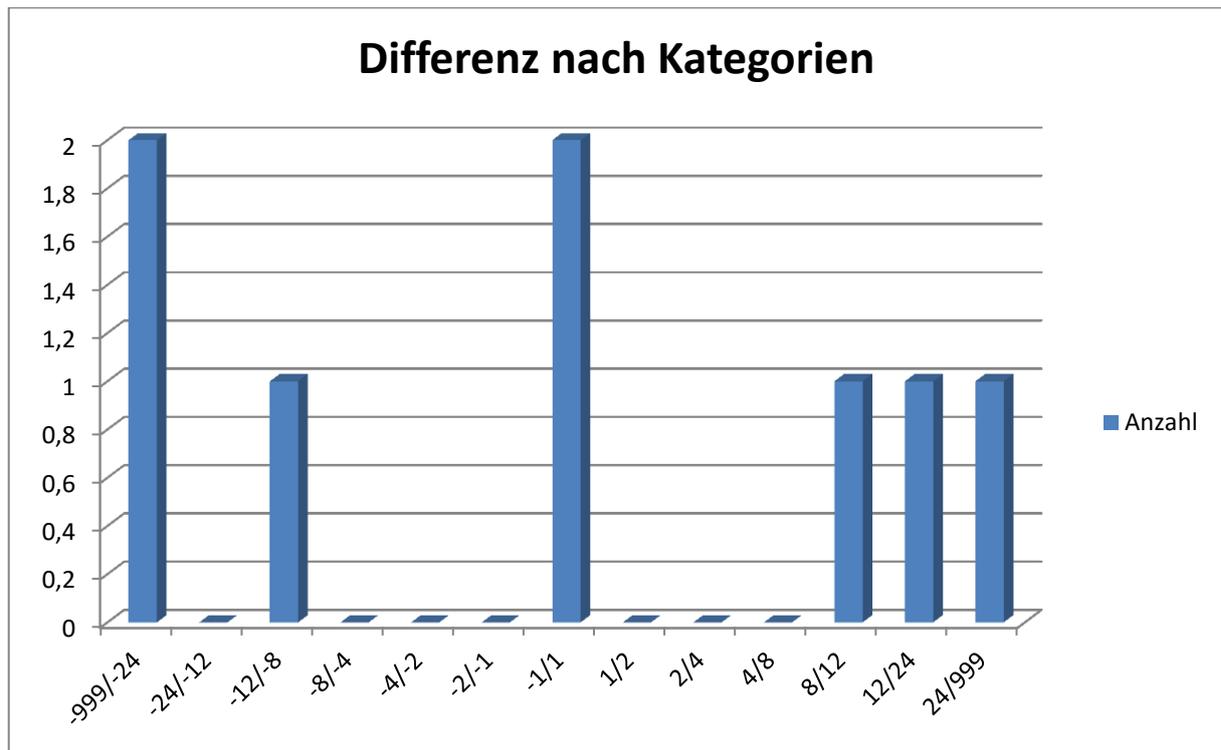
Projektmanager: Experte 5 (Interview 5)

Schätzgröße: Story Points (entsprechend 6 Stunden)

Schätzung im Mittel um 6% zu niedrig (67% StdDev), Median 3% zu hoch

Gesamtumfang: 444 h

8 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Schätzung zu hoch

Schätzung zu niedrig

Projekt 11

Anzahl der Projektteam-Mitglieder: 3

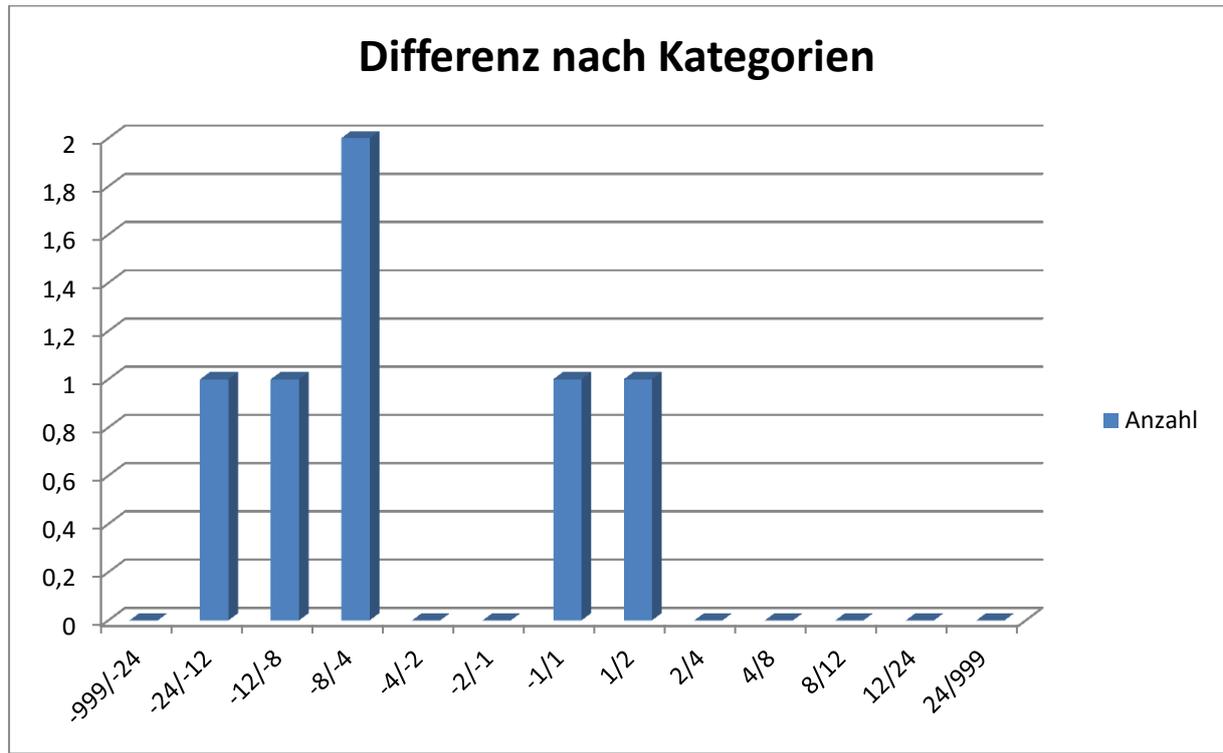
Projektmanager: Experte 5 (Interview 5)

Schätzgröße: Story Points (entsprechend 6 Stunden)

Schätzung im Mittel um 22% zu hoch (33% StdDev), Median 21% zu hoch

Gesamtumfang: 182 h

6 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Schätzung zu hoch

Schätzung zu niedrig

Projekt 12

Anzahl der Projektteam-Mitglieder: 4 (> 8h)

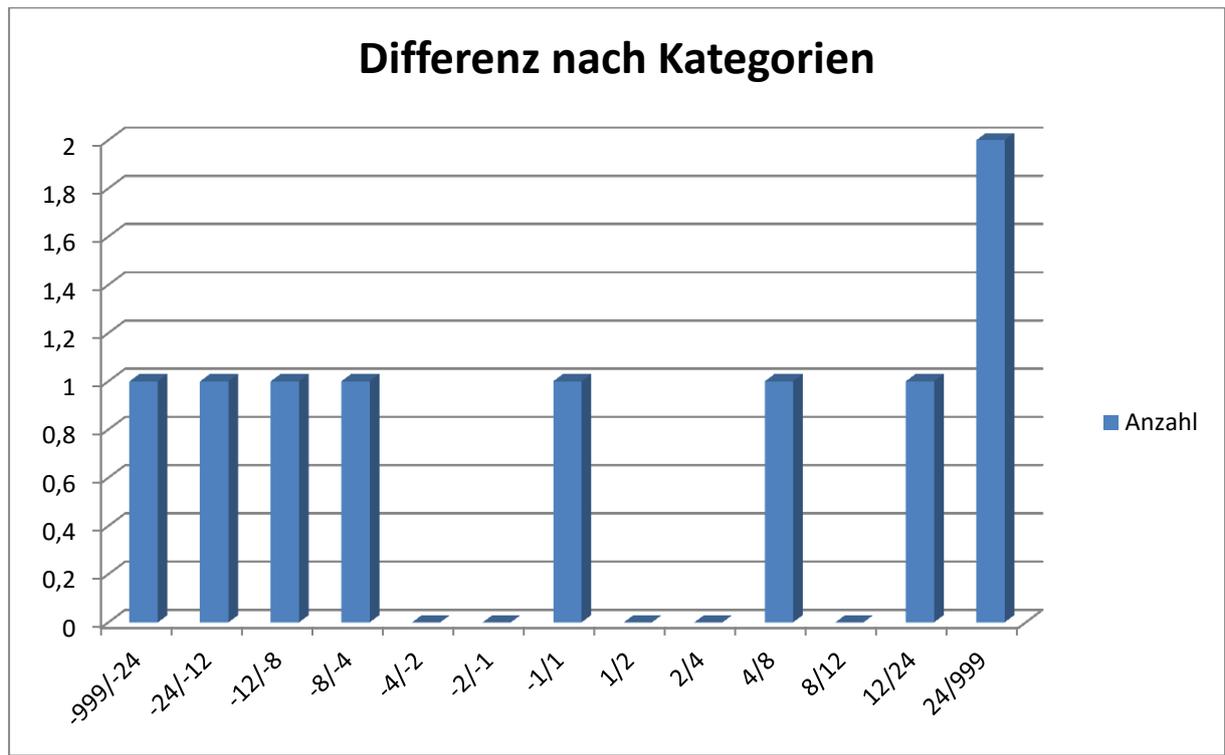
Projektmanager: Experte 5 (Interview 5)

Schätzgröße: Story Points (entsprechend 6 Stunden)

Schätzung im Mittel um 38% zu niedrig (150% StdDev), Median 3% zu hoch

Gesamtumfang: 402 h

9 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Schätzung zu hoch

Schätzung zu niedrig

Projekt 13

Anzahl der Projektteam-Mitglieder: 3 (> 15 h)

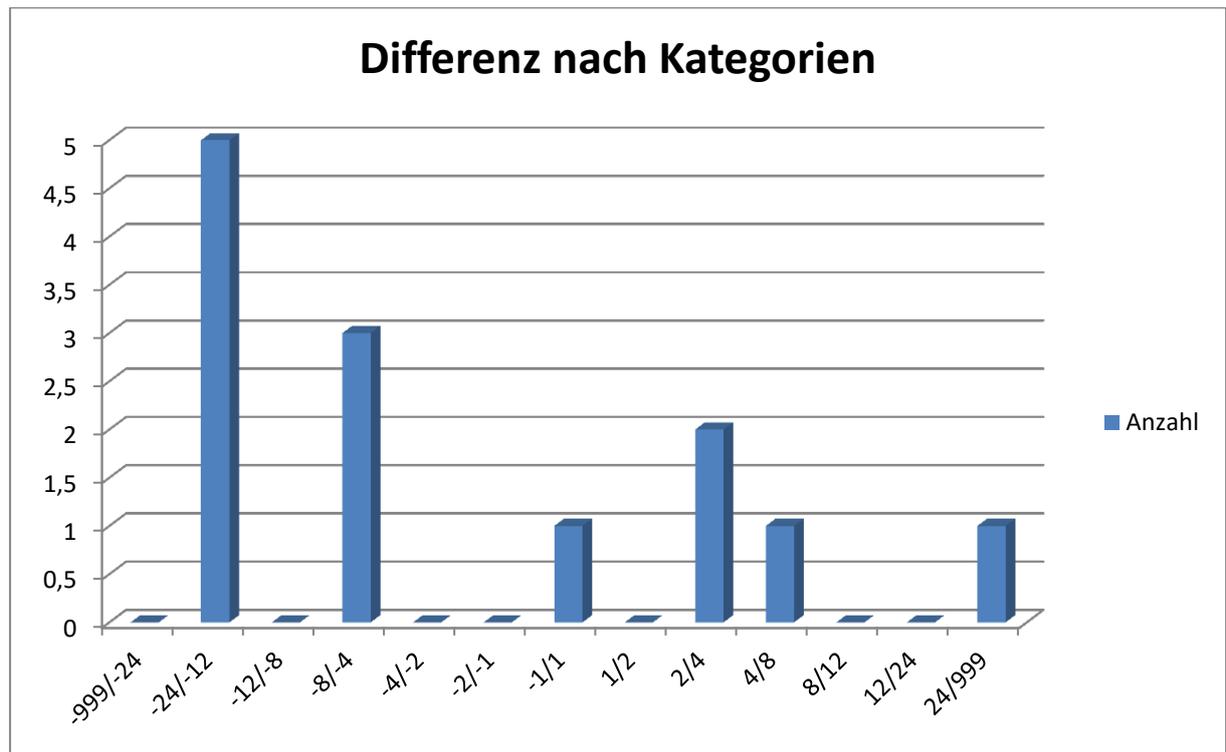
Projektmanager: Experte 5 (Interview 5)

Schätzgröße: Story Points (entsprechend 6 Stunden)

Schätzung im Mittel um 87% zu niedrig (400% StdDev), Median 33% zu hoch

Gesamtumfang: 366 h

14 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Schätzung zu hoch

Schätzung zu niedrig

Projekt 14

Anzahl der Projektteam-Mitglieder: 4 (> 8h)

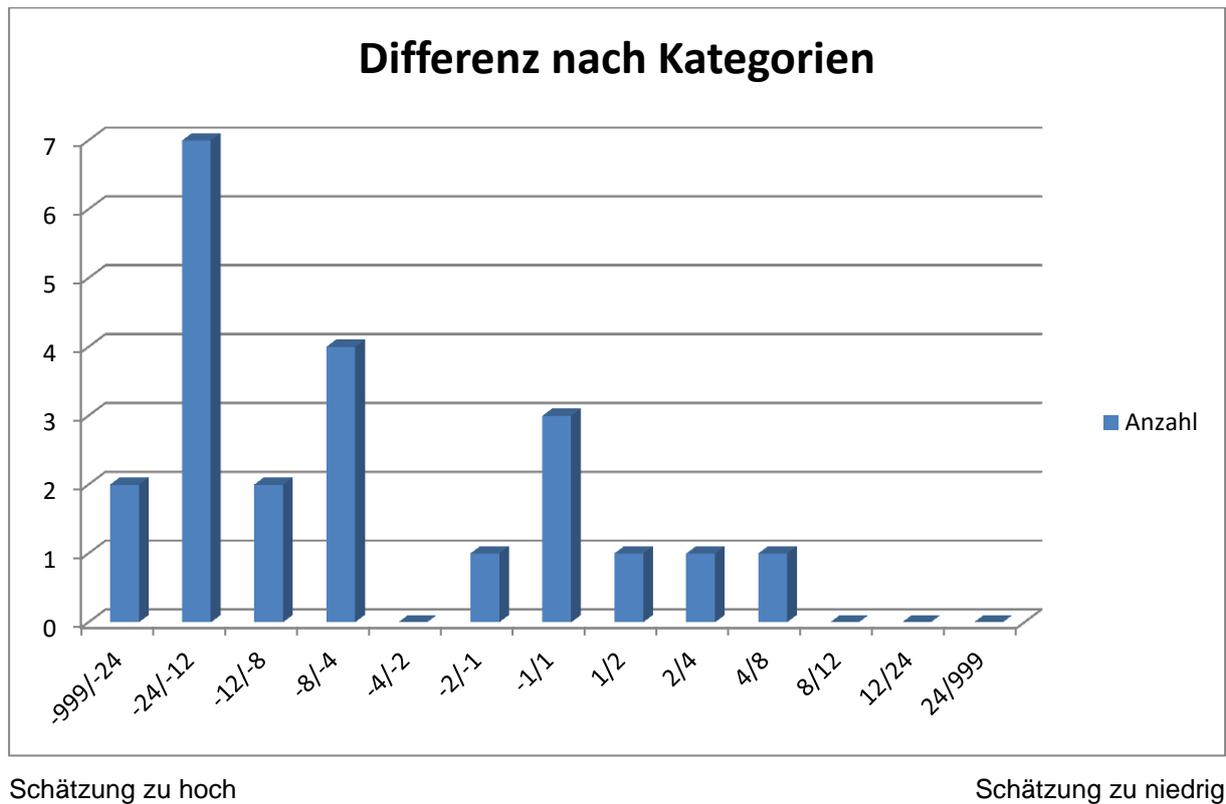
Projektmanager: Experte 4 (Interview 4)

Schätzgröße: Story Points (entsprechend 8 Stunden)

Schätzung im Mittel um 29% zu hoch (34% StdDev), Median 31% zu hoch

Gesamtumfang: 651 h

19 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Projekt 15

Anzahl der Projektteam-Mitglieder: 5 (> 8h)

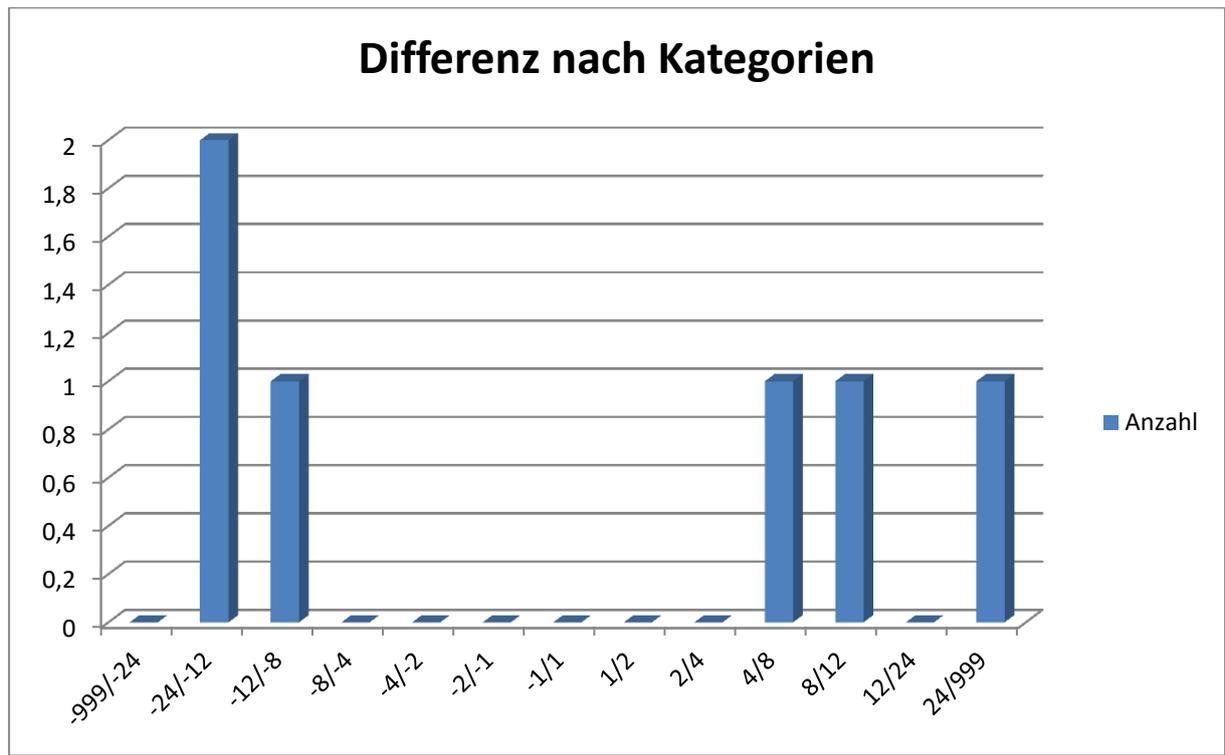
Projektmanager: Experte 4 (Interview 4)

Schätzgröße: Story Points (entsprechend 8 Stunden)

Schätzung im Mittel um 8% zu hoch (46% StdDev), Median 13% zu hoch

Gesamtumfang: 262 h

6 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Schätzung zu hoch

Schätzung zu niedrig

Projekt 16

Anzahl der Projektteam-Mitglieder: 9 (> 8h)

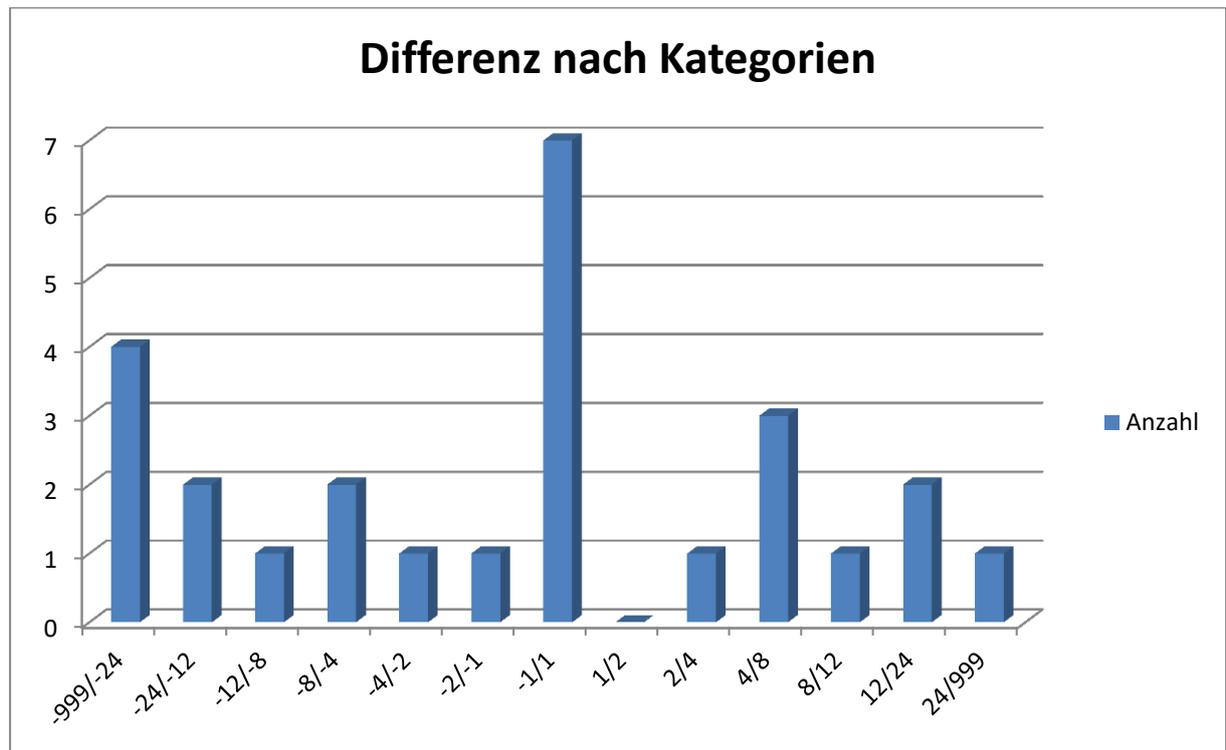
Projektmanager: Experte 4 (Interview 4)

Schätzgröße: Story Points (entsprechend 8 Stunden)

Schätzung im Mittel um 3% zu hoch (72% StdDev), Median 15% zu hoch

Gesamtumfang: 665 h

20 Backlog-Items mit Schätzung und zuordenbarem Aufwand



Schätzung zu hoch

Schätzung zu niedrig

Übersicht über die wesentlichen Kenndaten der Projekte aus den Projektsteckbriefen:

| | Schätzung mit Story Points | Schätzung mit Story Points entspr. Stunden | Schätzung mit Idealen Stunden | Anzahl Teammitglieder | Checklisten verwendet | Mittelwert Abweichung der Schätzungen | Standardabweichung | Median |
|-------------------|----------------------------|--|-------------------------------|-----------------------|-----------------------|---------------------------------------|--------------------|--------|
| Projekt 1 | X | | | 8 | | 3% | 92% | 38% |
| Projekt 2 | | | X | 6 | | -28% | 131% | -11% |
| Projekt 3 | X | | | 6 | | 0% | 81% | 0% |
| Projekt 4 | | | X | 8 | | -13% | 93% | 7% |
| Projekt 5 | X | | | 3 | | 0% | 62% | -5% |
| Projekt 6 | X | | | 4 | | 0% | 86% | 9% |
| Projekt 7 | X | | | 8 | | 0% | 74% | 24% |
| Projekt 8 | X | | | 3 | | 0% | 70% | 25% |
| Projekt 9 | | | X | 4 | | -36% | 163% | 10% |
| Projekt 10 | | X | | 3 | | -6% | 67% | 3% |
| Projekt 11 | | X | | 3 | | 22% | 33% | 21% |
| Projekt 12 | | X | | 4 | | -38% | 150% | 3% |
| Projekt 13 | | X | | 3 | | -87% | 400% | 33% |
| Projekt 14 | | X | | 4 | X | 29% | 34% | 31% |
| Projekt 15 | | X | | 5 | X | 8% | 46% | 13% |
| Projekt 16 | | X | | 9 | X | 3% | 72% | 15% |

ANHANG B - Transkriptionen der Experten-Interviews

Anhang B enthält die Transkriptionen der Interviews mit 6 Projektmanagern/Scrum Master bzw. Teamleitern zum Thema Agile Schätzungen und Erfolgsfaktoren für die Genauigkeit von Schätzungen in agilen Software-Entwicklungsprojekten.

Die Transkription erfolgte nach folgenden Regeln:

- Dialektfärbungen von Wörtern werden in hochdeutscher Form wiedergegeben.
- Dialektausdrücke oder umgangssprachliche Satzstellungen werden in ihrer originalen Form belassen und nach Gehör wiedergegeben.
- Füllwörter wie ah, äh, und ähnliches werden weggelassen.
- Wort- oder Satzteilwiederholungen werden weggelassen, soweit diese nicht auf das Verständnis oder den Sinn des Gesagten einen Einfluss haben.
- Erklärende Anmerkungen, welche Abkürzungen oder inhaltliche Aussagen für den Leser verständlich machen, werden pro Interview eingefügt.

INTERVIEW EXPERTE 1

Das Interview wurde am 23.03.2017 ab 16:00 in Pinkafeld durchgeführt. Experte 1 ist seit 2009 im Unternehmen und seit 2015 Projektmanager in einem agilen Team. Das Team von Experte 1 besteht aus sieben Personen.

Transkription

Am Anfang würde ich gerne herausfinden, wie deine Erfahrung in Projekten ist, was für agile Methoden eingesetzt werden – die Arbeit des Teams, wie das bis jetzt abläuft, und dann in weiterer Folge noch ein bisschen näher auf die verschiedenen Schätzmethode und die eingesetzten Dinge kommen. Da wäre die erste Frage, wie lange bei dir im Team agile Methoden für die Projektentwicklung eingesetzt werden.

Das kann ich nicht genau sagen, aber ich hätte jetzt zwei Jahre gesagt. Länger wird es nicht sein. Also zwei, zweieinhalb Jahre, als wir da begonnen haben, in dieser Firma.

Aber du selber machst das schon länger, oder? Wenn ich da zum Beispiel an GeoDiscoverer denke.

Du weißt das wahrscheinlich eh besser – die Zeit, als GeoDiscoverer, war's die 2.0er?

Ich glaube, ja. Da bist du ja so quasi als Wissender zu uns ins Team gekommen, das war, glaube ich, das erste agile Projekt, das wir gemeinsam gemacht haben. (ANMERKUNG: DAS ANGESPROCHENE PROJEKT WURDE VON SEPTEMBER 2011 BIS MAI 2012, ALSO RUND FÜNF JAHRE VOR DEM ZEITPUNKT DES INTERVIEWS, DURCHGEFÜHRT.)

Vorher war bei uns nicht ganz viel, sondern ein oder zwei GeoDesktop-Projekte – drum musst du wissen, wann diese Version GeoDiscoverer 2.0 war.

Genau kann ich es jetzt auch nicht sagen – das kann ich dann ja noch ergänzen, wann das war.

Ok, aber jetzt durchgehend arbeiten mit deinem Team und mit agilen Methoden so ungefähr seit 2 ½ Jahren, würdest du sagen.

Ich hätte es jetzt auf dieses – ein Projekt vor dem GeoDiscoverer, so hätte ich es formuliert. Ich weiß, die Zeit vergeht manchmal sehr schnell, aber, bevor GeoDiscoverer, bevor ich für dich gearbeitet habe, da haben wir gerade damit begonnen. Sozusagen. Und seitdem sagen wir, dass wir agil arbeiten und in der Arbeitsweise hat sich dann schon was geändert, aber seitdem gibt es dieses Scrum-artige Vorgehen und seitdem gibt es dieses Daily Scrum und Estimation und Story-Points und Backlog. Und wenn du das Jahr herausfinden kannst, dann weißt du es.

OK. Ja, das werde ich herausfinden.

Kannst du dich noch so an die Einführung der agilen Methoden bei uns erinnern, wie das so war, an irgendwelche besonderen Ereignisse, Erlebnisse? War das schwierig zum Einführen, war das einfach? Hat's Widerstand gegeben, hat's besondere Erkenntnisse, hat's aha-Erlebnisse gegeben?

Widerstand hat's keinen gegeben. Es war glaube ich immer nur eine – wie soll man sagen – in bisschen unterschiedliche Meinungen bezüglich der Wichtigkeit von Schätzungen und so. Weil halt manche Leute dann glauben, wenn's agil ist, dann ist jetzt diese Zahl nicht mehr so wichtig und dann gibt's keine Stunden mehr, die relevant sind, und dann ist das alles so mehr abstrakt quasi. Und dahingehend – weiß ich nicht – war dann ein bisschen eine Abstimmung zwischen den Leuten notwendig, dass man von einem Extrem – Stunden sind gar nicht wichtig – und es geht um die Weiterentwicklung des Produktes und fertig ist es nur, wenn es tatsächlich fertig ist und es gibt keinen Halbfertig-Stand und wenn wir nicht die ganze User-Story umgesetzt haben, dann ist's nicht halb fertig sondern immer noch ganz offen. Diese eine sehr extreme Sicht gegenüber der anderen extremen Sicht – die Stunden sind das wichtigste sozusagen und die sind auf null zu treffen, das war irgendwie so ein Lerneffekt, bis man halt so ein bisschen die Mitte gefunden hat. **Dass man dann jetzt mittlerweile weiß, die Stunden sind schon wichtig, drum versuchen wir jetzt auch wieder in Stunden zu schätzen, damit wir die halt besser treffen.** Aber sie sind jetzt auch nicht so wichtig, dass man das jetzt auf null irgendwie – dass man jetzt schon auf ein Monat oder ein Jahr, wenn wir das Projekt jetzt erfinden, wissen müssen die genauen Stunden, die es dann sein werden, und es muss auf null und so ausgehen. Also das war so ein Lerneffekt.

Also ein Lerneffekt von einem Extrem ins andere und im Endeffekt dann einen Mittelweg zu finden.

Ja, wahrscheinlich so die Leute, die es halt eingeführt haben, die waren halt auf der einen Seite des Lagers – sozusagen – zu finden. Und drum muss man sich selbst im Laufe der Zeit beim Arbeiten halt den eigenen Weg da so aneignen quasi und halt diese beiden Seiten unter einen Hut bringen.

Du hast eh vorher schon angesprochen: „Das Scrum-Artige Vorgehen“ – ist das eigentlich das, was ihr einsetzt?

Wir haben ja diese Schulung gemacht bzw. bin ich ja auch mit dem Hannes bei dieser OOP-Messe gewesen. Und da sind ja auch diese agilen Thematiken immer ein großes Thema und da heißt es dann immer „Scrum – but“. Also es gibt praktisch nur Scrum, wenn man alles macht, was im Scrum drinnen ist, aber viele Firmen machen eigentlich ein „Scrum – but“. Also ich glaube, wir machen auch dieses „Scrum – but“. Weil eigentlich das was wir von dieser Vorgehensweise machen sind halt diese Meetings quasi, oder. Die kennen wir alle, aber auch wenn wir bei dieser Schulung vielleicht mehr, genau detailliert gehört haben, wie diese Scrum eigentlich, was das noch haben möchte, machen wir eigentlich hauptsächlich diese Meetings. Alles andere geht halt trotzdem in die Projekt-Richtung, projektbasiertes Arbeiten. Da kümmert man sich eigentlich nicht besonders detailliert drum, was das Scrum eigentlich haben möchte.

Die Meetings, das sind das Daily Scrum, Estimation Meeting, ...

Ja genau. Sprint Planning und Review.

Eine Sprint-Retrospektive macht ihr auch regelmäßig?

Die gibt's jetzt nicht mehr. Ursprünglich haben wir sie eigentlich regelmäßig nach jedem Sprint-Review gemacht, aber es hat sich auch gezeigt, dass eigentlich – weiß ich nicht – die Mitarbeit dort eher, ja, mangelhaft ist. Drum, wenn man dann halt immer die Leute dazu stoßen muss, dass sie halt überhaupt etwas sagen, oder nach diesem Muster etwas sagen: „Was war gut, was wollen wir ändern und so“, dann hat es sich halt nicht durchgesetzt, sozusagen, behaupte ich. Weiß ich nicht. Wir haben es eh relativ lang mit sanftem Druck versucht. Die Leute dazu bewegen, dass sie da halt mitmachen, aber, weiß ich nicht, ich habe eher gelernt dass das nicht funktioniert und drum machen wir es halt nicht weil es ist sonst diese

Runde, da wird jeder benötigt, unter dieser Überschrift etwas zu sagen – aber seit wir es abgeschafft haben, hat es auch keiner nachgefragt.

Wie viele Leute seid ihr im Team?

Jetzt gerade, 1, 2, 3, 4, 5, 6.

Inklusive dir?

Sieben.

Ja – dann vielleicht ein bisschen in Richtung Schätzung. Wie habt ihr die Arbeitspakete? Habt ihr User-Stories, die geschätzt werden und mit welcher Methodik wird bei euch geschätzt? Wie läuft so eine Schätzung bei euch ab?

Ich glaube, es gibt zwei Formen von Schätzungen. Die erste Schätzung passiert mal zu Anfang des Projektes, weil irgendein Verkäufer oder so, oder weil für die GBL (ANMERKUNG: GESCHÄFTSBEREICHSLEITUNG) irgendjemand Zahlen in Stunden haben möchte für das Projekt gesamt. Und dann passiert die Schätzung eigentlich eher im sehr kleinen Rahmen, dass halt ich mit dem – ich als Lead Developer sagen wir halt mal – nicht als Projektmanager, sondern als Senior-Entwickler mit dem Produktmanager, mit dem Volker, eher so grob, größere Arbeitspakete sozusagen, halt mit dem Volker mir anschau und dann so Arbeitspakete, die so über eine Woche oder über zwei Wochen von der Stundenanzahl gehen halt, in dieser Granularität, diese Arbeitspakete schätzen. Das heißt, bei den letzten Projekten haben wir es halt eigentlich so immer gemacht, Das sind dann die groben Pakete, die auch in die Projektdefinition hineinkommen; das ist ein Teil von dem Projekt, der auch für den Kunden etwas angreifbares ist, was ja wahrscheinlich, wenn ich es sehr feingranular mache, nicht so der Fall ist. Und ja, das ist dann eigentlich auch das, was wir dann treffen sollen; das ist das, was ich controlle. In LZE (ANMERKUNG: LEISTUNGSZEITERFASSUNG) und so. Und, ja, alles andere, die zweite Art der Schätzung, da stellt sich dann natürlich die Frage, wie relevant die jetzt noch ist in Bezug auf das Projekt einhalten sozusagen. Die zweite Art der Schätzung findet dann im größeren Rahmen, im Rahmen der Estimation-Meetings, statt. Da schätzt dann eigentlich das ganze Team. Und schon immer noch auf Stundenbasis. Also da nennen wir es dann Story-Points und ein Story-Point ist ein Tag. Also – und das ist auch jedem bewusst, dass ein Story-Point ein Tag ist und drum schätzen wir hier auch in Stunden.

Und ist der Story-Point dann ein Arbeitstag oder ist das ...

Das ist ein Arbeitstag, ein Ein-Personen-Tag

Oder heißt das, „Ich arbeite acht Stunden an dieser Story“

Das heißt, wenn ich um acht Uhr in die Firma komme und wenn ich das dann verlasse um 17:00 Uhr, dann habe ich's gerichtet.

Aber möglicher Weise habe ich das zwischendurch Fragen von anderen Kollegen bearbeitet, musste einen Supportfall lösen, und hatte möglicherweis noch irgendein Meeting dazwischen.

Ja, das ist sicher nicht herausgerechnet. Also – ich frage mich gerade selber, wie wichtig mir diese Schätzung dann im Rahmen des Estimation Meeting ist. Weil die kommt ja eigentlich viel später dran. Also die Leute, die das dann vielleicht umsetzen, weil ich nicht alle in die Hand nehme sozusagen. Die bekommen das viel später gezeigt, die können das selber schätzen. Also die ganze Runde einigt sich dann auf diese Zahl, aber im Sinne des Projektmanagements ist mir das jetzt nicht ganz so wichtig, weil ich musste ja vorher schon diese Zahl finden. Und das was ich controlle, ist auch die Zahl, die ich vorher gefunden habe. Zuerst ist es für die GBL, das muss wer beantragen, die kauft 500 Stunden. Und erst

später dann, wenn das Team zum Zug kommt, dann könnten die draufkommen, dass insgesamt 1500 Stunden herauskommen. Ich meine, das passiert nicht, aber.

Gibt es da üblicher Weise irgendeinen Vergleich zwischen der anfänglichen Schätzung und der Summe der Schätzungen der Stories?

Die zwei Schätzungen stellen wir eigentlich nicht gegenüber. Eigentlich auch weil ich jetzt – weiß ich nicht – weil ich es jetzt irgendwie schwierig finden würde, wenn zuerst die Leute nicht gefragt werden und dann werden sie gefragt und sagen halt etwas Anderes. Was ist dann richtig? Arbeiten sie gut, wenn sie meine Zahl treffen? Aber eigentlich sehe ich dieses Estimation Meeting immer noch so, dass das den Hauptzweck hat, dass die Leute halt sehen, was im nächsten Sprint auf sie zukommt, damit sie die Stories kennen lernen. Und die Schätzung in diesem Zuge ist eigentlich wahrscheinlich trotzdem eine Bewertung der Komplexität. Dass man sieht, wie viel Arbeit dass man sich jetzt in diesem Sprint zutrauen kann. Das ist eigentlich der Hauptgrund für diese Schätzung. Drum habe ich auch noch nicht die Notwendigkeit gesehen, dass ich das auf die andere Zahl abbilde. Die ist eh schon vorher gekauft worden, sozusagen. Da wäre es zu spät, herauszufinden, ob die jetzt zusammenstimmt.

Ok. Ich denke, zwei wichtige Punkte hast du eh schon gesagt, wie ihr Story Points verwendet: Also eigentlich im Sinne von ein Story-Point ist ein Arbeitstag, und es hat auch mit Komplexität zu tun. Sind das so die Faktoren, die ihr dort berücksichtigt, bei der Schätzung. Um die Anzahl in Story-Points zu bewerten.

Die Komplexität würde ich jetzt ausklammern, also weil ... also ursprünglich war es die Komplexität. Naja – vielleicht ist es jetzt auch noch die Komplexität. Ein Story Point ist niemals der Wert für das Produkt. Das gibt es ja auch. Bringt das Produkt doppelt so viel voran wie das andere. Nein, in Wirklichkeit ist es eine Bewertung der Umsetzungszeit, sozusagen. Und jetzt halt auf kleinere Zahlen heruntergebrochen. Das wäre wahrscheinlich genauso gut, wenn wir dort 8 Stunden hinschreiben würden statt eins. Das sind halt Story-Points geblieben, weil in der früheren Phase wir das nicht auf Stunden umgerechnet hatten und diese Fibonacci-Reihe da verwendet hatten. Aber in der Praxis hat sich halt gezeigt, dass es trotzdem wichtig ist, diese Stunden praktisch, für die Gremien über uns das in Stunden zu schätzen und abzuliefern. Und drum, seitdem diese Wichtigkeit da zu allen durchgedrungen ist, macht es eigentlich keinen Sinn, dass wir das Fibonacci ansetzen. Weil, was hilft es mir, wenn es „dreizehn“ ist, oder 21. Wenn's 21 ist, dann ist es „viel“, dann muss man etwas tun. Das sind halt so ...

Ok, das heißt, ihr verwendet wirklich durchgehend 1, 2, 3, 4, 5 und so weiter.

Genau, ja. Wir verwenden eigentlich kein – das habe ich heute gesehen im Backlog, ein 0,5. Ja, ich meine, so wie ich, da kann ich wahrscheinlich nicht für alle sprechen, aber so wie ich bei diesen Schätzungen denke, ist dass das ein Vielfaches von Tagen ist, oder vielleicht von halben Tagen, wenn es ganz wenig ist. Aber ich glaube nicht, dass ich es von Haus aus so gut schätzen kann, dass ich es auf zwei Stunden treffe. Drum ist es mir auch nicht wichtig, dass da irgendwo hinein zu multiplizieren, damit sich das eh ausgeht. In Wirklichkeit ist das dann, wie gesagt, bei der ersten Schätzung, die jetzt weniger Laute vor der Genehmigung des Projektes machen, dort kann man eh genau die Stunden schätzen, die man halt glaubt. Und selbst dort denke ich in Tagen, sozusagen. Und im Rahmen des Estimation-Meetings ist dann eigentlich wichtig, dass man weiß, wie viel in diesen Sprint aufnehmen kann und dann sind die 2 Stunden und 3 ½ Stunden und 7 ½ Stunden nicht so wichtig, dann ist das auch mehr größenordnungsmäßig.

Ok. Das heißt also zwischen den Schätzungen der einzelnen Stories und dem tatsächlichen Aufwand gibt es kein Controlling oder keine Überprüfung, ob diese Schätzungen zutreffen?

Zwischen den Stories eigentlich nicht, weil die Story halt dann – es gibt mehrere Stories, die ein Arbeitspaket ausmachen sozusagen. Und ich kontrolliere dann nicht, ob die vier Stories, die zu dem Arbeitspaket Datenmigration TIGAS gehören, ob die zusammengenommen jetzt im Plan sind, weil die Projektdefinition, die Stunden der Projektdefinition und die Arbeitspakete, so wie sie in der Projektdefinition genannt sind und mit Zahlen versehen sind, nachdem ich ja diese treffen muss, kontrolliere ich diese größeren Arbeitspakete. Indem die Leute halt bei mir nicht auf eine User-Story buchen im LZE, sondern indem ich die Namen dieser Arbeitspakete im LZE vorgebe, wenn er dann, damit er sich halt leichter tut, die Migration TIGAS in 5 User-Stories herunterbricht, damit er sich leichter tut, darf er das eh machen, aber er muss es so buchen, mit einem Titel, damit ich das als Migration TIGAS wiederfinde.

Das heißt, die Arbeitspakete werden vom Team möglicher Weise in mehrere Stories aufgeteilt, und verglichen wird dann aber immer der Gesamtaufwand pro Arbeitspaket, also der geleistete Aufwand mit dem ursprünglich angenommenen Aufwand.

Das machen wir eigentlich – wenn ich nicht schlampig bin, dann machen wir es wochenweise, also dass halt im Rahmen oder nach dem Daily Scrum, am Freitag ich halt eine Liste in der Hand habe, wo ich für die Projekte, die jetzt gerade im Laufen sind, eigentlich, die Zahlen pro Arbeitspaket in der Hand habe, wieviel war geplant und wo stehen wir gerade. Das habe ich ihnen auch gesagt, das könnte man eigentlich als Zettel vor jedem seinen Kopf hinpicken, damit er das jeden Tag, jede Stunden vor Augen hat. Aber das soll es ja auch nicht sein. Also ich sehe das eher als Hilfestellung für die Mitarbeiter, damit wir am Ende der Woche halt wissen, wo wir stehen. Da soll schon jeder die Freiheit haben, dass er –wie soll man sagen - sich bei der kreativen EDV-Tätigkeit schon in seinem Rahmen entfalten kann sozusagen. Aber die Stunden sind halt trotzdem wichtig. **Und am Ende der Woche bekommt man eine Hilfe, dass man weiß, ob man noch viel Zeit hat, um sich auszutoben und etwas zu verbessern, oder ob's halt schon gescheitert ist, wenn man es auf den Boden bringt.**

Und diese Arbeitspakete innerhalb von einem Projekt, ist das dann eine Produkt-Version, oder können das auch unterschiedliche Dinge – weil du jetzt zum Beispiel TIGAS angesprochen hast – wäre das ein Projekt, wo alles für den Kunden TIGAS ist oder sind das auch unterschiedliche ...

Die Arbeitspakete, auf die jetzt gebucht wird? Nein. Was ist ein Arbeitspaket?

Gibt es da – in so einem Projekt, gibt es da Arbeitspakete für die TIGAS, ein Arbeitspaket für das Standard-Produkt GeoDesktop und ein Arbeitspaket für einen anderen Kunden?

Ich hätte gesagt, ein Arbeitspaket spielt sich in einem Projekt ab. Und dann gibt es halt die Arbeitspakete für das Kundenprojekt TIGAS, für Inventory-Manager, dort heißen die Arbeitspakete Installation des Standard-Produkts, Datenmigration für die TIGAS, Erweiterung des Standard-Produktes um eine Grundbuch-Online-Abfrage und Durchführung der Benutzerschulung. Projektmanagement und Betreuung, das ist auch immer so ein Arbeitspaket, damit ich das wohin buchen kann. Aber so wie ich es jetzt aufgezählt habe, das ist alles was für ein Projekt, in dem Fall für ein Kundenprojekt, zu leisten ist. Und so wie ich es jetzt aufgezählt habe, ist es schon sehr grob sozusagen, weil diese Erweiterung um eine Online-Abfrage, das sind vielleicht zwei Wochen, oder so. Aber da schneide ich diese Arbeitspakete eigentlich so, dass ich mir denke, das sagt auch dem Kunden etwas, oder es sagt auf jeden Fall der Person in der GBL etwas. Das ist wahrscheinlich zu grob, als dass es im Backlog, als der, der es

implementiert, haben möchte, aber die Zielgruppe für das, das kommt alles aus der Projektdefinition, das kommt alles vielleicht aus einem Angebot, das ist alles für Leute, die in der GBL sitzen und auch für diese Leute controlle ich es, drum richte ich mich nach dem, was deren Granularität, deren Sicht ist. Und das muss man so ein bisschen abbilden auf das Backlog sozusagen. Ursprünglich haben wir es so gemacht, dass die Leute schon mit den Titeln der User-Stories buchen, wieso weiß ich nicht, das war halt mal so. Dann habe ich einen großen Aufwand, dass ich fünf User-Stories zusammensuche und wieder ... Dann musst du immer im Excel jonglieren und hast eigentlich keinen Mehrwert sozusagen. Der Entwickler ist ja zufrieden damit, wenn er es irgendwo hin buchen kann.

Das heißt also, es gibt ein wöchentliches Controlling – einen Vergleich der Schätzungen und tatsächlichem Aufwand auf Basis von diesen Arbeitspaketen.

Genau.

Und wie ist da so die Güte dieser Schätzungen? Passt das üblicher Weise oder gibt es da mehr Überschreitungen oder mehr Unterschreitungen, und im Mittelwert passt es dann wieder?

Ich glaube, im Mittelwert pass es sogar. Also es gibt sehr wohl Überschreitungen, die ich dann halt versuche einzufangen, am Wochenende sozusagen, wenn das halt gemacht wird. Es gibt aber auch Unterschreitungen. Das letzte Mal waren eigentlich mehr Unterschreitungen, Unterschreitungen, die halt massive Unterschreitungen waren. Da haben wir eigentlich zu viel geschätzt sozusagen. Würde ich jetzt sagen. Die Überschreitungen, die waren eigentlich kleiner, weil man die halt versucht einzufangen mit diesem wöchentlichen drauf schauen und Leute fragen und vielleicht Alternativen suchen, ob es vielleicht etwas einfacher gemacht wird oder so. Dann sagt man den Leuten halt, so war es gedacht, wie lange brauchst du noch. Man will das auch nicht unbedingt stoppen sozusagen. Das haben wir noch nie gemacht. Die Reißleine ziehen sozusagen. **Aber ich glaube, durch das öfter mal hingewiesen werden, schafft man es dann halt leichter, dass man es ein wenig, schneller zu einem Ende bringt.**

Kannst du sagen, welche Faktoren dazu führen, dass Schätzungen genau sind? Weil – was ich jetzt verstanden habe, das war je eher, es gibt eine Schätzung und danach muss ich mich im Zuge der Umsetzung danach strecken, dass ich diese Schätzung treffe. Entweder war ich schneller, dann ist es wahrscheinlich eh eine angenehme Situation, oder ich laufe Gefahr, länger zu brauchen und dann wird an der Umsetzung – möglicherweise was weggelassen oder etwas einfacher gelöst ...

Ja schon. Also möglicher Weise, wie du sagst, wenn man jetzt sieht; wenn wir das wöchentlich bewerten und man sieht, man ist drüber, und man sieht, andere Stories sind aber schon fertig und sind halt drunter geblieben, dann wird man das halt eher noch weiterführen sozusagen. Oder sonst muss man halt trotzdem das abkürzen. Aber das hat ja nichts mit der Genauigkeit, mit dem möglichst gut schätzen was zu tun. Das hat etwas damit zu tun, wenn ich sehe, dass es nicht gepasst hat, wenn ich mich schon verschätzt habe, dass ich es irgendwie versuche einzufangen. Aber wenn du bei der letzten Frage gefragt hast, welche Faktoren das beeinflussen, dass die Schätzung vorab schon möglichst genau ist ... mmh ... dann kann ich wahrscheinlich nur sagen, dass der, **der es schätzt vielleicht der ist, der es dann auch macht**, und dass der, **der es schätzt auch einer ist, der das relativ gut kennt, das Produkt**. Dahingehend trägt die ganze Methodik auch ein Problem in sich. Weil die, die schätzen, sind eigentlich sehr wenige Leute, drum ist die Wahrscheinlichkeit, dass der, der es macht, einer war, der da dabei war.

Das heißt, die Schätzungen, die dann kontrolliert werden, die sind relativ weit weg von der tatsächlichen Umsetzung.

Zeitlich, meinst du?

Ja.

Ja, würde ich schon sagen.

Weil vor Projektbeginn, oder oft auch vor Projektbeauftragung, wird offensichtlich geschätzt, dann vergeht einige Zeit ...

Das ist sicher auch bei den Vermessern so, ihr müsst ja auch zuerst einmal einen Projektantrag mit einer Stundenzahl genehmigen lassen. Oder der Verkäufer muss auch sein Kundenprojekt, wo er dem Kunden sagt, was es kosten wird, mit einer bestimmten Zahl, einer Stundenzahl verkaufen.

Das ist möglicher Weise bei uns in vielen Fällen etwas anders, weil wir oft interne Projekte haben. Standardprodukte. Und da eigentlich ein Stundenrahmen definiert wird an Hand des Umsetzungszeitraumes und danach wird das dann mit möglichen Arbeitspaketen aufgefüllt.

Gerade jetzt sind wir wieder dabei, so einen Stundenrahmen, der so artig übrig bleibt bis zum nächsten Release mit Tätigkeiten für das Standardprodukt zu füllen. Da wird es aber auch so sein, dass wir, bevor das noch ein Implementierer sieht, das eingefüllt haben mit einer Schätzung im kleinen Rahmen, was dann auch von den anderen irgendwie zu treffen sein soll. Also das alles was wir tun, das nenne ich Expertenschätzung so zu sagen. Ein paar Leute, die schon länger da sind, schätzen das nach bestem Wissen und Gewissen und der andere bekommt es dann halt mitgeteilt. Aber eigentlich nicht, damit der dann schätzen kann, was es wirklich ist, sondern damit er es halt kennen lernt. Und weiß, was da zu tun ist sozusagen.

Das heißt – was wir auch schon angesprochen haben – die Arten von Projekten, die ihr umsetzt, sind oft Kundenprojekte, sind auch Standard-Produkte. Und die sonstigen Rahmenbedingungen, sind das meistens bekannte Dinge, was Technologien betrifft, was Domäne betrifft, oder gibt es da auch neuartige Sachen, oder unbekannte Dinge?

Großteils sind es eigentlich bekannte Dinge, hätte ich gesagt. Also, dass in den letzten Projekten, dass da was dabei war, das so sehr unbekannt ist, dass man es jetzt vorab eigentlich gar nicht einschätzen kann, ob das 100 oder 1000 Stunden sind, das ist eigentlich noch nie gewesen.

In den Projekten, gibt es da einen starken Druck von außen, von der Unternehmensleitung oder vom Abteilungsleiter, gewisse Zeit oder einen Kostenrahmen einzuhalten?

Natürlich! Alles, was eine Überschreitung ist, ist schlecht. Also ja, sehe ich schon.

Bei Kundenprojekten ist es wahrscheinlich mehr, oder macht das keinen Unterschied?

Ich glaube, das macht nicht wirklich einen Unterschied. Also so, wie es dann zu mir kommt als Projektleiter, ist es schlecht, wenn die Zahl höher ist. Auch wenn es ein Standard-Produkt ist, dann ist es schlecht, wenn die Zahl höher ist, und es ist sogar schlecht, wenn ein Teil vom Projekt mehr war, obwohl es insgesamt praktisch auf Null ausgegangen ist. Das ist halt so. Irgendwie ist die Argumentation, dass man ja 500 Stunden gekauft hat, als man als GBL sich dafür entschieden hat, das zu machen oder nicht zu machen. Oder das zu machen an Stelle von etwas anderem, sozusagen, und dann will man halt auch, dass das auch gemacht wird um diese Stunden. Ich sehe das immer als Projektleiter so, dass man es zuerst nach bestem Wissen und Gewissen schätzt. Und dass es trotzdem – weil das halt eine komplexe Tätigkeit ist, dass man das trotzdem noch überschreiten kann, dass das nichts Schlimmes ist, aber dass

man es halt früh genug kommuniziert nach oben. Wenn es sich abzeichnet, damit einer, der ja ein Manager ist, draufschauen kann und das mal sieht und dann entscheiden kann als Manager, ob das jetzt eh passt, oder ob er halt das abbrechen will. Hat noch niemand abgebrochen. Das wäre wahrscheinlich ein Fehler, das nicht zu kommunizieren, weil dann ... ich weiß nicht ... wenn man es dem Management sagt, dann ist der unter Zugzwang, etwas zu entscheiden.

Und gibt es auch öfter quasi vorab schon Vorgaben, oder Erwartungen, wie viel Aufwand ein Projekt sein kann. Also wenn ihr eine Schätzung abgibt, gibt es dann irgendwelche Rückmeldungen „das kann doch nicht so viel Aufwand sein, oder der Kunde wäre nur bereit, so und so viel zu bezahlen, das muss doch schneller gehen“, oder gibt es das eher nicht?

Eher nicht. Also gar nicht würde ich nicht sagen, das gibt es schon; der Kunde ist aber bereit, das gibt es gar nicht, aber es gibt es sehr wohl, dass diese Zahl hinterfragt wird. Dass man halt glaubt, dass das nicht so viel sein kann. Im Endeffekt glaube ich auch, dass das im Zuge der Erstellung dieser Projektdefinition, halt ist, wenn sie in der ersten Variante geschrieben ist, dann redet man sich das halt aus, sozusagen.

Und werden dann Schätzungen auch angepasst, oder bleiben die üblicher Weise so, wie von den Experten geschätzt?

Nein. Also das kommt darauf an, was beim Ausreden herauskommt, sozusagen. Also durchaus möglich, wenn der das sieht, was der Senior-Developer geschätzt hat, und wenn der sieht, dass der das auch so verstanden hat, wie es von der Anforderung gedacht war, dass die Zahl halt so hoch ist. Dann gibt es wahrscheinlich auch die Möglichkeit, dass das dann halt herausfällt, weil es das nicht Wert ist quasi. Vielleicht eher in einem Standard-Projekt. **Es ist sicher auch eine Möglichkeit, dass die Zahl weniger wird, weil man im Zuge vom Reden eine alternative Umsetzung findet oder weil halt der Schätzer draufkommt, dass er etwas verkehrt verstanden hat.** Alle Varianten sind möglich. Die übliche ist eigentlich, dass die Zahl gekauft wird.

In welcher Form sind die Arbeitspakete, die geschätzt werden müssen, üblicher Weise beschrieben. Kurz haben wir ja schon drüber geredet – grobe Anforderungen – ist dann zum Zeitpunkt der Schätzung klar, was alles zu tun ist, oder ist das oft eher noch unsicher, was da denn alles drinnen stecken kann?

Das hat jetzt sehr viele Formen, diese Beschreibung von solchen Sachen. Also ich denke jetzt wieder hauptsächlich auf diese Vorab-Schätzung, die Zahl, die ich mal anbieten muss, die ich dann treffen muss. Da kommt es eigentlich – da muss ich wieder an die letzten Projekte, Kundenprojekte, denken – da kommt es eigentlich aus einem Angebot heraus. Und das hat der Verkäufer geschrieben. Wenn es der Strobl (ANMERKUNG: EIN MITARBEITER IM VERTRIEB, DER EINE TECHNISCHE AUSBILDUNG HAT UND VOR LÄNGERER ZEIT ALS SOFTWARE-ENTWICKLER GEARBEITET HAT.) geschrieben hat, dann weiß er genau, was er abgrenzen und so muss, und dass er das halt relativ genau beschreiben muss für den Kunden. Und drum ist das dann eigentlich ziemlich allumfassend. Also das wird dann eigentlich auch zwischen Verkäufer und Produktmanager und mir durchgegangen, diese Beschreibungen. Und dann ist eigentlich für jeden klar, was drinnen ist und was nicht drinnen ist. Und eigentlich bei der zweiten Schätzung probiere ich das dann halt auch so hinüber zu bringen und wenn es drei Stories geworden sind aus einem Arbeitspaket, dann ist es halt auch so, aber dann probiert man das halt auch pro User-Story so alles mit zu sagen, was da jetzt hineingehört. Oder wann das jetzt aus ist. Wenn das zum Beispiel ein Teil

von dem, die User-Story nur ein Teil von dem Arbeitspaket ist, dann wird da auch irgendwie zusammengefasst, was da am Ende von dem herauskommt und warum wir es brauchen, weil es der Teil ist von dem ... jetzt habe ich das falsche Vokabel gebraucht ... weil diese User Story jetzt der Teil ist, damit dieses Arbeitspaket dann mal einen gewissen Stand hat am Ende des Sprints.

Ich glaube, dann haben wir so die wesentlichen Punkte durch. Die letzte Frage wäre noch, ob mit unterschiedlichen Ansätzen geschätzt wird, und da haben wir ja auch schon drüber gesprochen, dass es üblicher Weise schon gemacht wird, habe ich verstanden. Es gibt einerseits diese Top-Down Schätzung am Anfang vom Projekt, oder eigentlich vor dem Projekt und dann gibt es noch eine Schätzung Bottom-Up auf Arbeitspaket-Ebene. Wo aber üblicher Weise kein Abgleich erfolgt dazwischen. Das habe ich jetzt verstanden.

Ja, genau. Also als Schusssatz vielleicht. Ich weiß nicht, ob mir das jetzt im Zuge dieses Interviews aufgefallen oder eingefallen ist. Was man mit dieser Variante der Schätzung, dass man halt die Vorab-Schätzung wichtiger bewertet, als das was man während der Umsetzung dann schätzt und trifft, **was halt da komplett rausfällt ist, sind irgendwie Erfahrungen, die halt viele Personen machen, wie gut ihre Schätzungen halt sind sozusagen.** Das fließt eigentlich nirgends zurück. Das fließt vielleicht wieder bei deren Schätzung auf Basis von anderen User-Stories wieder in die Bewertung dieser anderen User-Stories ein, aber eigentlich für das, **was du bei einer anderen Vorab-Schätzung wieder besser wissen sollst, geht das eigentlich verloren.** Nicht 100%ig verloren, weil dann der Erfahrungsschatz des Senior-Developers halt auch größer wird und er halt auch dann dort mitbekommt, wo er sich verschätzt hat, aber ein Erfahrungsschatz von vielen Personen wird da nicht 1:1 zurückgeführt.

Dann sage ich herzlichen Dank!

INTERVIEW EXPERTE 2

Das Interview wurde am 24.03.2017 ab 08:00 in Pinkafeld durchgeführt. Experte 2 ist seit 1999 im Unternehmen und seit 2015 Projektmanager/Scrum Master in einem agilen Team. Davor hat er auch als Projektmanager in klassischen Teams gearbeitet. Das Team von Experte 2 besteht derzeit aus drei Personen. Die länger zurück liegenden Projekte wurden in größeren Teams mit bis zu neun Personen umgesetzt.

Transkription

Anfangen würde ich gerne mit deinen Erfahrungen mit agilen Methoden in der Projektentwicklung und dann gerne ein bisschen auf die Methodik für Schätzungen zu sprechen kommen und dann auch noch auf die Faktoren, wie Schätzungen möglicher Weise genauer werden können.

Die erste Frage dazu wäre, wie lange bei dir im Team agile Methoden für die Projektentwicklung eingesetzt werden, wenn du das ungefähr sagen kannst, seit welchem Zeitraum diese Methoden eingesetzt werden.

Also, im Prinzip bin ich bei den agilen Methoden seit den GeoMapper-Projekten – welche Versionsnummer, weiß ich jetzt nicht. Alles was vorher war, haben wir noch in unserem alten Schema gehabt, also bis 2011, 2012 wird das gewesen sein. Und ab da haben wir begonnen im GeoMapper-Umfeld, also mit diesen Projekten.

Kannst du dich noch daran erinnern, wie das war mit der Einführung von agilen Methoden, irgendwelche bemerkenswerten Ereignisse oder Probleme, Widerstände – wie da die Erfahrungen waren?

Also ich glaube, speziell in der Technik hat man natürlich versucht, nachdem die Arbeitsanleitung herausgekommen ist und wir diese Scrum-Schulung gehabt haben, das natürlich so bestmöglich umzusetzen, nur mit den anderen Abteilungen, sprich auch Terminen und Inhalten und so weiter hat das nicht so funktioniert. Da waren am Anfang schon noch Diskrepanzen.

Und innerhalb der Technik, in den Teams, war das problemlos, oder hat es da etwas gegeben?

Also es hat sich niemand quergelegt. Ich bin halt immer der größte Skeptiker gewesen davon, aber ich habe natürlich auch versucht, dass ich da mitmache. Ich bin auch kein Einzelkämpfer, sondern im Wohle des Teams. Und speziell am Anfang war ich ja auch nur ein Teammitglied bei diesen Projekten, bei den ersten Projekten.

Stimmt, ja genau. Bei uns weiß ich es – die agilen Methoden sind im wesentlichen Scrum. Das ist glaube ich das, was bei uns, bei euch im Team eingesetzt wird?

Ja, also das auf alle Fälle. Wobei wir eigentlich aktuell viele Vorteile, sagen wir einmal so, aus dem klassischen Projektablauf, aus dem Wasserfallmodell, wieder heranziehen. Das ist jetzt so ein bisschen Scrum mit Klassik, glaube ich, trotzdem kombiniert wieder.

Inwieweit, was sind da die Sachen aus dem klassischen Projektmanagement?

Die Abschätzungen in Stunden, ist, glaube ich, der größte Unterschied zum Scrum. Vom Scrum holen wir uns halt die Zeitspannen heraus – die Timeboxes.

Dein Team besteht zur Zeit aus zwei Mitarbeitern und einem Product-Owner?

Ganau, ja – naja, Product-Owner sind es mehr. Also jetzt im aktuellen Projekt – ja.

Gut, dann kommen wir vielleicht schon mal zu den Schätzungen: Mit welcher Methodik werden bei euch im Team aktuell die Schätzungen durchgeführt?

Mittels Estimation-Meeting mit dem Product-Owner, und aber auf Basis von Stunden. Wir versuchen bestmöglich natürlich die Unit-Tests mit einfließen zu lassen, wobei dort ist noch ein bisschen Aufholbedarf.

Und die Schätzung erfolgt so, dass jeder der ... in dem Fall ... beide Teammitglieder, einfach sagen, was sie denken, wie groß der Aufwand ist?

Genau. **Wobei da natürlich sehr viel die Erfahrung mit einspielt.** Wenn einer die Erfahrung nicht hat, kann er sich der Stimme enthalten. Weil es dann eh keinen Sinn macht.

*Also derzeit wird mit Stunden geschätzt, **Schätzungen mit Story-Points** hat es aber auch gegeben?*

Hat es früher auch gegeben, wobei wir ehrlich gesagt mit dem, in dieser Phase damals eigentlich nur Probleme gehabt haben. Niemand hat dann einen Endtermin sagen können, niemand hat wirklich den Stundenaufwand sagen können, was natürlich mit dem Endtermin zusammenspielt.

Als mit Story-Points geschätzt worden ist, hat es da ein gemeinsames Verständnis gegeben, was ein Story-Point ist?

Glaube ich nicht.

Das war einfach eine abstrakte Zahl? Oder hat es da etwas gegeben, was das denn sein könnte?

Einige haben sich sehr an die abstrakte Zahl gehalten, also das eine Paket ist mehr oder weniger komplex als das andere. Andere haben es intern in Stunden umgerechnet, und da war der Faktor so ungefähr 6 damals, 6,5. Und einige haben das ... ‚Könnte 20 Stunden dauern, sagen wir 3‘

*Werden aktuell im Team,... wird da ein **Controlling** der Schätzungen durchgeführt, d.h. gibt es da einen Vergleich zwischen der Schätzung und dem tatsächlichen Aufwand?*

Also **ich, für mich mache es schon**, aber ich tu's dann nicht so direkt präsentieren.

Erfolgt das auf Basis von Stories, oder für den Gesamtaufwand, oder ...

Auf Basis von Stories.

Hast du da ein Gefühl dafür oder auch eine tatsächliche Auswertung, wie gut diese Schätzungen passen?

Also, ich habe die letzten drei Projekte hergenommen, durch unsere Buchungen im LZE kann ich das sehr genau zuweisen. Da haben wir von 237 Stories, hätten einen Soll-Stundewert gehabt von 1130, ist waren dann 1330. Und von diesen 237 Stories waren 119 also schneller umgesetzt als geplant und 117 haben länger gebraucht als geplant. Wobei man sagen muss, was länger dauert, dauert dann meist viel länger. Also wenn es schneller fertig ist, dann ist es meist nur so ein bisschen, und wenn es länger dauert, artet es dann halt aus auch gleich. Also nicht nur im 2-Stunden-Bereich, sondern auch im 20-Stunden Bereich.

Kannst du sagen, wovon das abhängt, ob Stories gut geschätzt werden oder nicht? Gibt es irgendwelche Erfahrungen, was das positiv beeinflusst, dass die Schätzung gut passt?

Wo die Schätzung gut passt, da sind **keine Entwicklungen im NG-CORE**, da sind wir in unserem Team sehr schlecht aufgestellt, was die Erfahrung betrifft. Und wenn dort Änderungen sind, das dauert dann immer länger. Wenn irgendein kleiner Befehl zu machen ist, das ist für uns eigentlich kein Problem.

Das heißt, das ist eigentlich die Erfahrung der betroffenen Entwickler mit dem umzusetzenden Arbeitspaket?

Genau, ja.

In welchem zeitlichen Abstand zur Umsetzung werden die Schätzungen normaler Weise durchgeführt? Ist das unmittelbar vor der Umsetzung, ist das lange davor, wie ist das normaler Weise?

Also, wir haben sicher viele Stories in unserem Backlog, die wir schon ewig lange abgeschätzt haben, aber der Großteil, was wir umsetzen, wird kurz vorher abgeschätzt. Also wirklich im 2-, 3-Wochen-Bereich. Stories, die länger abgeschätzt sind, kommen nur spärlich dann hinzu.

Und die Schätzung wird auch von den Personen durchgeführt, die dann die Umsetzung machen?

Ja.

Welche Arten von Projekten sind das, die hauptsächlich umgesetzt werden? Sind das Standard-Produkte, Kundenprojekte, gibt es da neuartige Technologien, andere Programmiersprachen? Kann man da ein bisschen was drüber sagen?

In den letzten 2 bis 3 Jahren haben wir nahezu ausschließlich an Standard-Produktentwicklung gearbeitet, und einige kleine Kundenprojekte, die aber auch nur aufsetzen auf das Standard-Produkt. Also da war jetzt nicht technologisch, Programmiersprachen-mäßig, nichts Herausforderndes.

Und gibt es in diesen Projekten von außen, zum Beispiel von der Unternehmensleitung, von Vertriebskollegen, oder vielleicht auch aus Kundensicht irgendwie einen starken Druck, Zeit- und Kostenrahmen einzuhalten? Oder werden Schätzungen, so wie sie sind, zur Kenntnis genommen und gutgeheißen?

Also früher war der Einfluss sicher stärker, das hat aber in den letzten zwei Jahren schon abgenommen. Was auch, glaube ich, positiv fürs Projektteam ist.

Und hat das deiner Meinung nach irgendeinen Einfluss gehabt auf die Qualität und die Genauigkeit von Schätzungen?

Indirekt schon, ja. Weil doch immer auch die Ansage gekommen ist, ‚das kann ja nicht so lange dauern‘. Und mit diesem Hintergedanken machst du automatisch eine optimistischere Schätzung, obwohl es gar nicht stimmt.

*Die Schätzung erfolgt üblicher Weise auf Basis von den Arbeitspaketen, oder den User-Stories. Wie sind diese **Arbeitspakete** beschrieben? Ist das ausführlich oder umfassend beschrieben, oder nur in Stichworten, oder wie ist das üblicher Weise?*

Also wenn ich GeoMapper als unter Leitprojekt, Leitprodukt hernehme, sind die schon **sehr ausführlich beschrieben**. Wobei natürlich, je nach Stress des Product-Owners, der Product-Ownerin, das natürlich abnimmt. Und ich glaube, umso länger, umso älter, umso langlebiger dieses Produkt wird, umso unübersichtlicher wird es. Weil eben schon so viel drinnen ist, dass man dann auch gar nicht mehr alles

mitberücksichtigen kann. Was dann auch gar nicht Absicht ist beim Beschreiben der Stories, aber da kommt man dann erst beim Testen drauf oder beim Umsetzen.

Gibt es dann zum Zeitpunkt der Schätzung, gibt es da dann ein gemeinsames Verständnis, was alles zu tun ist, oder ändert sich das dann zwischen Schätzung und Umsetzung?

Nein, also zwischen Schätzung und Umsetzung ändert es sich nicht. Was da natürlich mit hineinspielt, ist die Erfahrung des Mitarbeiters. Also dienstjüngere Mitarbeiter, da ist die Diskrepanz schon ein bisschen größer. Wobei wir eh versuchen im Estimation-Meeting das alles zu beschreiben, bestmöglich.

Werden die Schätzungen mit unterschiedlichen Ansätzen durchgeführt, also zum Beispiel Top-Down und Bottom-Up. Also gibt es eine Gesamtschätzung, die dann mit der Schätzung der einzelnen Stories verglichen wird, oder gibt es das eigentlich nicht?

Haben wir eigentlich nicht. Wir haben zwar einige Schätzungen, die wir für ein Angebot brauchen, die geht der Product-Owner mit dem Projektmanager einmal durch, da wird eine große Zahl gesagt. Aber was dann tatsächlich aus den Stories herausfällt, tun wir mit dem nicht mehr vergleichen.

Ja, danke, das war es dann eigentlich schon an meinen Fragen. Dann sage ich dankeschön.

INTERVIEW EXPERTE 3

Das Interview wurde am 29.03.2017 ab 08:00 in Pinkafeld durchgeführt. Experte 3 ist seit 2002 im Unternehmen und war mehrere Jahre als Scrum-Master in einem agilen Team. Im Jahr 2015 war er in einem ausgegliederten Team, dem Technologie- und Plattformteam, tätig. Derzeit übernimmt er die Rolle des Scrum-Masters im Team des Autors, der die Projektmanagement-Agenden in diesem Team wahrnimmt. Das Team von Experte 3 besteht aus fünf Personen.

Transkription

Anfangen möchte ich mit ein paar Fragen zum Hintergrund, zu deinen bisherigen Erfahrungen mit agilen Methoden und Schätzungen. Die erste Frage dazu wäre, wie lange bei dir im Team agile Methoden für die Projektentwicklung schon eingesetzt werden.

Ja, eigentlich betreiben wir das schon seit einigen Jahren, ich glaube, das erste Projekt wo wir das so richtig gemacht haben, wo ich das zumindest als Teammitglied, noch nicht als Scrum-Master so mitmachen, so handhaben durfte, war GeoDiscoverer 1.0 für die Vermessungsverordnung (ANMERKUNG: GEMEINT IST TATSÄCHLICH DIE VERSION 1.2 DES PRODUKTS, WELCHE ZUM FREIGABETERMIN MAI 2012 UMGESETZT WORDEN IST). Das war sicher vor der Zeit, wo ich diese Scrum-Master-Schulung oder Weiterbildung gehabt habe. Seither sind alle Geo*-Projekte; also ist das in allen NG-Standardprodukten (ANMERKUNG: ALLE PRODUKTNAMEN VON PRODUKTEN, DIE AUF DER GEMEINSAMEN PLATTFORM „NG“ (NEXT GENERATION) ENTWICKELT WERDEN, BEGINNEN MIT „GEO“) im Einsatz gewesen (abgesehen vom TUP-Team (ANMERKUNG: DAS TECHNOLOGIE- UND PLATTFORMTEAM) halt, diese Forschung zum Scripting). Da bin ich halt immer bei agilen Projekten Teil davon gewesen. Also schon mehrere Jahre im Endeffekt.

Welche Erfahrungen hat es gegeben bei der Einführung von agilen Methoden? Kannst du da ein bisschen was dazu sagen?

Also ich glaube neu war sicher vor allem, oder eine große Umstellung war sicher vor allem, dass man eben nicht ein dickes Pflichtenheft hat von Anfang an, wo natürlich auch immer viel Aufwand hineingegangen ist, das alles niederzuschreiben. Und die Kommunikation ist sicher angekurbelt worden durch die andere, die agile Vorgehensweise. Dass man eben diese Regeln quasi hat, dass bestimmte Besprechungen regelmäßig sind, so wie das Daily Scrum, das empfinde ich schon als hilfreich, dass es gewisse Regeln gibt, nach denen man sich richten kann. Dass es immer zur gleichen Zeit, nach dem gleichen Ablauf passiert. Das sind sozusagen die Fixpunkte. Das andere kann man ja wirklich dann variieren. Gezeigt hat sich, glaube ich, dass vor dem Zeitpunkt, wo aus den verschiedenen Teams die Leute die Schulung genossen haben, dass das sehr unterschiedlich gehandhabt worden ist, glaube ich, in den Teams. Bei GeoDiscoverer haben wir das, glaube ich, recht gut gemacht. Am Anfang war das recht rigoros, dass man sich versteift auf gewisse Sachen. Sozusagen, das ist das Modell und das müssen wir befolgen, weil sonst ist es kein Scrum. Weil es halt außen drauf steht auf dem Prozess, dass

es Scrum ist und so funktioniert. Seither haben wir glaube ich den Prozess ein bisschen abgewandelt und unseren Gegebenheiten angepasst. Was glaube ich schon der Fall ist, dass wir gewisse Sachen ein bisschen aufgeweicht haben. ‚Im Sprint-Planning, da mag ich nichts mehr schätzen.‘ Wobei man halt aufpassen muss, dass man sich selbst die eigenen Prinzipien nicht verrät sozusagen, weil irgendwie möchte man ja schon am Ende des Sprints wissen, was man da kriegt und nicht komplett alles umstricken zwischendrin. Weil auf gewisse Sachen richtet sich das Team halt aus, wenn man das quasi mal geschätzt und den Sprint angefüllt hat. Das hat schon einen Sinn, dass es vorgesehen ist, dass es sich nicht verändert. Ich glaube, grundsätzlich halten wir uns schon dran, aber in anderen Teams glaube ich schon nach wie vor, dass es anders ist.

Mit „Halten wir uns schon dran“ meinst du, an die Vorgaben, die es für Scrum gibt?

Ja. Also ich glaube schon, das ist noch irgendwie der Leitfaden und könnte man ... was weiß ich, öfter, wenn es halt nicht so läuft, wird rmDATA vor 20 Jahren hergenommen. Also, als Extremist könnte man das auch als ultra-agil bezeichnen. Wenn halt alles, was gerade reklamiert wird, sofort umgesetzt wird. Im Gegensatz zu einem dicken Pflichtenheft schreiben, das halt ultra-starr ist und am Ende ist es dann für uns gut gelaufen, wenn alles was da drinnen steht „richtig“ unter Anführungszeichen umgesetzt ist. Aber der Kunde, der will es dann erst anders haben, aber wir sind halt glücklich, wenn das was da drinnen steht, was wir vor 1 ½ Jahren geschrieben haben, gut ist. Da haben wir uns schon in die richtige Richtung bewegt.

Du hast vorher angesprochen das TUP-Team, das Technologie- und Plattformteam. Inwieweit war da das Vorgehen anders, als es in deinem Standard-Team ist?

Ich glaube, da haben wir vorher schon in den Teams – jetzt muss ich mich erinnern versuchen – da haben wir dann begonnen in Stunden zu schätzen im nächsten GeoMapper-Projekt, das war nach dem TUP. Vorher haben wir in den Standard-Produkten halt auch noch nach Story-Points geschätzt. Ja, was war anders? Also ich glaube, nachdem das Team kleiner war – da waren wir ja nur zu dritt – war das alles noch agiler sage ich einmal. Also der Hannes hat da recht auf die Einhaltung der Time-Boxes geschaut und dass wir nicht allzu viel Zeit sozusagen verschwenden bei den Schätzungen, sondern so artig ‚Sag, was es kostet‘. Und da haben wir auch so Sachen wie dieses Scrum-Poker so richtig betrieben, also mit den Karten quasi, nur halt mit einer App (einer Windows-Phone App).

Planning-Poker?

Planning-Poker, genau. Das macht natürlich auch jedes Team anders und das haben wir zuerst auch so gemacht in GeoMapper-Projekten – oder nein, da haben wir Magic-Estimation gemacht. Also das war halt dann gewissermaßen eine andere Erfahrung, **weil du halt dann wirklich unbeeinflusst bist und wenn dann halt einer eine komplett andere Zahl hinhält, dann denkst du dir vielleicht doch einmal ‚Oha‘**. Also im Endeffekt geht es ja nur darum, die Diskussion zu einem Thema, das scheinbar nicht eindeutig zuordenbar ist, anzukurbeln. Das war halt dort ein anderer Zugang sozusagen.

Ok. Also im Technologie- und Plattformteam habt ihr das Planning-Poker verwendet für die Schätzungen. In dem Team, wo du jetzt wieder bist, wie wird da geschätzt?

Da schätzen wir nach Stunden. Wobei wenn man – ich weiß nicht – ich komme ja in dem Fall von einem Hintergrund, wo man eben noch nicht agil gearbeitet haben und das ist dann immer – ich war auch kurz Projektleiter bei ein paar Projekten. Da muss man dann immer einen Projektantrag mit Stunden abgeben und irgendwie glaube ich dann schon, dass man das dann nie aus dem Kopf heraus kriegt. Also wenn es

auch immer heißt, nach Komplexität soll man die Story-Points vergeben. Nur irgendwie haben wir im TUP-Team dann auch, also der Peter Kremnitzer war da drin, von Telekom-Seite, da haben sie auch immer alles mit Stunden, die haben nie Story-Points gehabt, glaube ich. Das war halt dann irgendwie schon mehr oder weniger im Hinterkopf oder sogar im Vordergrund: Was könnte das jetzt in Stunden kosten und dann hat man wahrscheinlich bewusst oder unterbewusst umgerechnet in wieviel – ich habe jetzt – nach einem Sprint weiß ich so zirka, was ein Story-Point in Stunden ist und dann habe ich das halt wieder umgerechnet. Und dann hat man dann erst nach Stunden, glaube ich, geschätzt. Und das halt im Kopf umgerechnet. (Wir gehören ja noch zu einer Generation, die das kann.)

Das heißt, deiner Meinung nach, sind in diese Schätzung mit Story-Points, ist doch hauptsächlich der Aufwand eingeflossen, um zu sagen, wie viele Story-Points eine Story kostet.

Ich glaube schon, ja.

In dem Team, wo Du jetzt aktuell bist, wie viele Leute sind da drinnen?

Fünf

In dem Team, gibt es da ein Controlling der Schätzung, das heißt einen Vergleich zwischen dem tatsächlichen Aufwand und dem geschätzten Aufwand.

Es gibt in diesem Tool, das wir verwenden, gibt es mittlerweile – das war auch nicht immer – gibt es eine technische Möglichkeit pro Arbeitspaket die Stunden zu sehen, die drauf gebucht wurden und kann man dann mit einem anderen Tool – dem Jira – gegenüberstellen. Ich bin mir jetzt nicht sicher, ob das regelmäßig wer macht, **aber wir haben sicher die Meilensteine regelmäßig abgeglichen**. Ob das jetzt quasi, ob die Stories, die einem Meilenstein zugeordnet sind, ob die datumsmäßige, sage ich einmal, ob dieser Meilenstein erreicht wurde, ob er erreichbar ist. Wenn er nicht erreicht wurde, um wieviel und wir dann den nächsten noch erreichen können. Also von dieser Seite ja. Aber nicht laufend irgendwie pro ... wir haben nicht bei jedem Daily Scrum das im Detail durchgesprochen, wie weit dass ein bestimmter Task ist und wieviel dass man noch zu brauchen glaubt, das zieht sich nicht durch, das haben wir nicht regelmäßig gemacht. Nur wenn es halt auffällt – was weiß ich – ein Task ist schon sehr lange auf „Doing“. Dann fragt man schon mal nach, ‚Was ist mit dem jetzt los‘? Wird da noch dran gearbeitet oder ist man vielleicht abgelenkt und das ist eh nicht in Arbeit sozusagen oder braucht man halt wirklich um das länger.

Weißt du, wie gut die Qualität der Schätzungen ist oder hast du ein Gefühl dafür, dass man sagen könnte ‚Wir treffen mit unseren Schätzungen den tatsächlichen Aufwand gut oder weniger gut‘?

In diesem Projekt, oder ...

Allgemein in unserem Team.

Ob ich es weiß, oder was mein Gefühl ist?

Ob du es weißt, oder, falls du es nicht weißt, ob du ein Gefühl dafür hast, ob du sagen kannst ...

Wissen? Nein. Aber ich glaube, wir treffen es im Allgemeinen recht gut. **Ich glaube, bei den kleineren Paketen wird es so sein, dass wir manchmal zu viel annehmen und bei den großen zu wenig**. Aber ich glaube, in Summe gleicht es sich aus. Was ich auf Grund der Tatsache, dass wir die Meilensteine alle ziemlich gut getroffen haben, annehme. Auf Grund dessen habe ich sicher nicht jede einzelne Arbeitspaket-Schätzung angeschaut. Weils im Großen sozusagen eh passt.

Die Meilensteine sind eine Zusammenfassung von größeren Themen, wo es auch wiederum eine Schätzung in Stunden gegeben hat?

Ja. Also jetzt am Beispiel des Projekts GeoDesigner. Ich weiß jetzt nicht mehr genau, wie das gegangen ist, ob wir da schon einen Termin vorgegeben hatten. Ich glaube schon. Bis dann und dann können wir uns vorstellen, das freizugeben. Dann haben wir es zu zweit, glaube ich, haben wir es geschätzt. Also wieviel Stunden sind es bis dorthin, mit wieviel Mann, haben versucht es aufzugliedern und mal zu schauen, wie kann man es aufteilen, welche Meilensteine. Das ist so passiert, dass man versucht die Themen zu gliedern. Was weiß ich, wir machen zuerst einmal die Import-Funktionen, damit ich einmal Daten da habe, damit ich das Produkt verwenden kann. Dann machen wir das und das. Also, wir haben schon versucht, dass man jeweils, wenn ein Meilenstein fertig ist, mit dem Produkt wieder was anfangen kann, sage ich mal, ohne es freizugeben. Aber dass ein bestimmtes Kapitel abgeschlossen ist, so wie die Übernahme von CAD-Elemente, die wir als kritisch gesehen haben, weil das auch für den Anwender sehr wichtig ist. Drum haben wir es auch recht weit vorne gemacht, so weit vorne als möglich, und, ja zu diesen Meilensteinen haben wir dann geschaut: Von der Summe der Stunden, wie viel davon könnten wir für das brauchen und geht sich das dann aus. Das war definitiv Top-Down. Da haben wir noch nicht gewusst, was jetzt für den Meilenstein für Arbeitspakete sind. Aber die Schätzung war halt quasi anteilmäßig. Wie viel Anteil an dem ganzen ist jetzt diesem Meilenstein oder diesem Thema, das bis zu diesem Meilenstein abgeschlossen sein soll, einzuräumen. Und nach diesen Gesichtspunkten haben wir das dann quasi aufgezogen. Zu zweit. Also zwei von fünf haben das gemacht.

Und danach hat es dann – vor Umsetzung der einzelnen Meilensteine – hat es Schätzungen auf Arbeitspaket-Ebene gegeben?

Mh-hm (ANMERKUNG: BEJAHEND)

Und das ist dann auch verglichen worden?

Ja, schon. Wir haben dann quasi zu jedem Meilenstein oder zu jedem Thema, **haben wir das dann im Endeffekt detailliert, die Arbeitspakete**, oder die Tasks geschrieben. **Das waren am Anfang auch recht grobe Tasks** und dann haben wir im Team, in Estimation-Meetings, sind wir nach und nach draufgekommen, da wissen wir zu wenig, **die müssen wir schneiden**, die Tasks. Und da ist das dann eigentlich in Gang gekommen. Also wir haben am Anfang ja auch außerordentliche Estimation-Meeting gemacht. Also grundsätzlich haben wir jeden Freitagvormittag. Und dann haben wir halt am drauffolgenden Montag oder am Mittwoch nochmal, damit wir da halt schnell quasi zu einem Gefühl im Team für diese Tasks halt kommt und dass halt jeder ein Gefühl für den Aufwand oder halt für die Schätzung entwickelt.

Hast du eine Idee, oder eine Vorstellung davon, von welchen Faktoren die Genauigkeit von Schätzungen abhängt. Gibt es da Erfahrungen: ‚Wann schätzen wir gut, wann schätzen wir schlecht‘?

Erfahrungen, glaube ich, sind, **dass die Erfahrung, die man selber hat in Projekten und auch mit der technischen Umsetzung vor allem, dass das eine sehr große Rolle spielt**. Wenn einer neu im Team ist, dann tut er sich da sicher schwer, wenn er die Materie nicht kennt, sozusagen. Also ich glaube dass, egal was man schätzt, das immer wieder vergleicht mit etwas, das man schon kennt. Das ist sicher ein positiver Faktor. Was du schon angesprochen hast, oder was da vorkommt, glaube ich, **wenn du dann einen Druck hast, irgendwas sagen zu müssen, dann läufst du, glaube ich, Gefahr, dass du halt**

irgendetwas sagst. Etwas, das halt nicht überlegt ist. Dann ist es halt entweder zu viel, weil du halt mehr auf der sicheren Seite sein willst, bevor du was Falsches sagst. Oder es ist halt zu wenig, weil du nicht alles in Betracht gezogen hast, was da halt dranhängt. Weil quasi das zu schätzende zu grob umrissen ist, sozusagen.

Wann werden üblicher Weise die Schätzungen durchgeführt, in welchem Abstand, in welchem zeitlichen Abstand zur Umsetzung?

Zur Umsetzung – also wie viel vor der Umsetzung, meinst du?

Genau.

Naja, ich glaube, das variiert. Wir haben es definitiv schon gehabt, weil das Entwickeln gerade so klass ist, dass man die Stories vielleicht etwas zu sehr hat reifen lassen oder liegen hat lassen im Backlog. Aber ich glaube es hat sich gezeigt, dass wenn du es schon 2 oder 3mal, also so 2 bis drei Wochen würde ich jetzt im Durchschnitt schätzen. Dass man gewisse Stories, wenn man also wöchentlich schätzt, dass man die dann schon zwei- oder dreimal gesehen hat zumindest. Und von dem her glaube ich, dass das halt so die Norm ist, dass man das drei Wochen vor der Umsetzung ...

Hat das einen Einfluss auf die Genauigkeit der Schätzung? Entweder der zeitliche Abstand oder wenn man das öfters durchspricht, wird die Schätzung dann besser?

Ja, ich glaube schon. Oft ist es sicher so, wenn man von außen draufschaut, da denkt man, das kennen wir eh schon, das brauchen wir nicht anschauen. Aber während der Umsetzung der anderen Tasks, kommt man schon öfters auf Sachen drauf, von denen dann diese Stories, die da noch im Backlog sind, profitieren können, wenn man die noch nachschärfen kann. Was im Endeffekt wieder auf Erfahrung, also auf Erfahrung, die man gerade gemacht hat, hinausläuft. Dann passiert das, glaube ich auch, dass man halt draufkommt, ‚Öha, ich hatte gerade diese Problem, da kenne ich eine Story dazu‘, die suche ich mir heraus und da mache ich ein Fähnchen drauf und damit wir im Team nochmal drüber reden. Ob das Problem, das ich gerade hatte, ob das quasi die Zukunft beeinflusst, ob das quasi aufwendiger geworden ist. Weil es ein technisches Problem war oder **weil ich in dem Zuge draufgekommen bin, dass in dieser Story ganze Funktionalität nicht bedacht wurde, die noch irgendwo gemacht werden sollte oder muss.** Also das beeinflusst das schon.

Die Personen, die die Schätzungen durchführen, das sind üblicher Weise auch die Personen, die dann tatsächlich die Umsetzung machen?

Ja.

Also das gibt es eher nicht, dass gewisse Personen schätzen und das sind dann andere, die die Umsetzung machen.

‚Nicht‘ möchte ich nicht sagen. Es gibt schon Fälle, was weiß ich, ein Projekt ist schon länger in Planung und wird jetzt quasi frisch im Backlog erfasst. Es gibt aber schon irgendwo eine Projektdefinition mit Meilensteinen zumindest oder vielleicht auch schon mit Arbeitspaketen, die schon wer anderer geschätzt hat. Das kommt schon vor. Bei GeoDiscoverer-Erweiterungen. Das wurde – wenn ich das noch sagen darf da dazu, also zu dieser Frage – da sind wir kürzlich draufgekommen, das hat halt ein anderes Team geschätzt. Die Truppe, die die Experten für die 1.0-er-Version damals waren. Das hat halt quasi den Zuständigkeitsbereich gewechselt, also jetzt ist wer anderer zuständig, aber die Experten sind jetzt woanders. **Jetzt haben die das vorher mal, um eine Größenordnung für die Projektdefinition zusammen zu bringen, mal geschätzt. Und jetzt schätzen wir es gerade wieder im Team,** das das

umsetzen soll und kommen, glaube ich, zu dem Schluss, dass da so ziemlich das gleiche herauskommen wird. Wobei einer der, im Team, die das umsetzen sollen, drinnen sitzt, bei der alten Schätzung dabei war oder diese zumindest kennt sozusagen. Wobei wir aber jetzt nicht pro Arbeitspaket beeinflusst waren, was der gesagt hat. **Jetzt, wo wir eigentlich, die Abschätzung der Arbeitspakete, die da drinnen sind, schon recht weit fortgeschritten sind, zeigt sich halt, dass da so ziemlich das gleiche herauskommen wird.**

Gibt es in den Projekten, wo du tätig bist, oft einen starken Druck von außen, einen Zeit- oder Kostenrahmen einzuhalten?

Ja. Also im momentan umgesetzten Projekt, das kurz vor der Freigabe ist nicht. Also es gibt schon einen Zeitrahmen, aber die Schätzung, die haben wir recht gut getroffen. Da hat sich dann gezeigt, dass dann vor dem Zeitpunkt, wo jetzt die im ersten Schritt abgesteckte Funktionalität fertig geworden wäre, ist man draufgekommen, man braucht noch ein zusätzliches Feature, ein Umfangreichen – mit dem Drucken – das haben wir dann halt mit hinein genommen. Das war jetzt aber nicht der Druck, glaube ich. Das GeoDesigner-Projekt ist dahingehend recht pflegeleicht, sage ich mal. Vor allem bei 1.0-er Versionen ist es regelmäßig anders. Da gibt es eben auch einen bestimmten Termin. Der Termin ist eigentlich als fix zu betrachten. Was dann hineinkommt, ist dann halt immer kritisch. Ich glaube, da hatten wir schon immer die Angst, wenn wir das hergeben, das können wir so nicht hergeben. Und ja, jetzt so Timebox-bezogen haben wir das nicht gemacht, wir haben gesagt, der Termin und der Umfang muss gegeben sein. Dann entsteht natürlich der Druck und ich glaube schon, dass dann die Schätzungen der Pakete, die da noch drinnen sein sollen, dort schon auch noch einmal beeinflusst war davon. Unter solchen Umständen zu schätzen, wenn man schon einen Stress hat, ist sicher nicht leicht. Ich glaube schon – ich weiß jetzt nicht, was die Studie ergibt – aber, **dass diese dann spät und unter Druck geschätzten Arbeitspakete sich dann üblicher Weise, dass die dann mehr Aufwand bei der Umsetzung benötigen und quasi das Ganze dann nochmal verzerren.** Also da hast dann doppelt gelitten.

Die Arbeitspakete, die geschätzt werden müssen, wie gut sind die üblicher Weise beschrieben? Sind die ausführlich beschrieben oder nur in wenigen Stichworten. Das heißt, ist für das Team, das die Schätzung durchführt, üblicher Weise klar, was alles da drinnen steckt, in dieser Story?

Wenn sie zur Schätzung kommen, meinst du?

Ja.

Das ist glaube ich recht unterschiedlich, wer das schreibt. Wenn das ein Techniker schreibt, dann werden da vielleicht schon so Hinweise auf die Umsetzung transportiert, wo ein anderer Techniker, der das dann schätzt, sich – hoffentlich – orientieren kann. Weil, ich glaube, das ist dann die Intention, dass derjenige der das hineinschreibt. Wenn's der Produktmanager oder ein Produktverantwortlicher schreibt, dann ist das glaube ich eher, wie man eine User-Story wirklich verfassen sollte halt, geschrieben. Was man damit machen möchte, nämlich. Also die Wahrheit ist dann wahrscheinlich irgendwo in der Mitte. Aber, glaube ich, im Großen und Ganzen zeigt sich, dass, wenn sie zur Schätzung kommen, die meisten Stories schon verwendbar sind. Also im Zuge des Schätzens werden dann schon Fragen beantwortet, aber ich glaube, das was ... der rote Faden ist immer vorhanden, dass man dann auch die richtigen Fragen stellen kann.

Eine Frage noch einmal zurück zur Art, wie geschätzt wird: Wie läuft das jetzt aktuell ab, wie die einzelnen Personen ihre Schätzungen abgeben?

Die Aufgabe, ich sage einmal Task, wird von einer Person, das ist üblicher Weise der Produktverantwortliche oder halt der Product-Owner, vorgestellt am Beamer. Also es schauen alle auf diesen Schirm sozusagen. Also derjenige stellt die Story eben vor. Dann fragt normaler Weise der Scrum-Master, ob das Team jetzt eben genug Informationen hat, oder ob wir noch Informationen brauchen oder... Oft ist der der, der die erste Frage stellt, was halt noch unklar sein könnte. Diese Frage- und Antwortrunde wird halt, also kommt dann in Gang und wird dann fortgesetzt, bis nach Auffassung des Teams alles klar ist. Da werden manchmal auch im Team unter den Technikern, jetzt möchte ich nicht sagen Umsetzungsdetails, aber sagen wir mal, die Marschrichtung für die Umsetzung diskutiert, aber nicht sehr ausschweifend, sage ich mal. **Und dann wird halt, wenn genügend Information in Rücksprache mit jedem einzelnen – da wird jeder gefragt, ob er sich das jetzt Schätzen traut. Und dann sagt jeder reihum eine Zahl.** Jetzt quasi nicht alle zugleich, sondern halt reihum. Und wenn da grob was anderes herauskommt, zum Beispiel der eine sagt zwei, der andere sagt 16, dann wird es halt nochmal diskutiert, warum da ein Unterschied herauskommt. Ich glaube, wenn zwei und 8, das ist vielleicht ein Schritt, der noch ohne Diskussion abgeht, und dann wird halt die Mitte genommen. Üblicher Weise nehmen wir dann aber glaube ich das, was die Mehrheit sagt und einigen uns dann halt auf diese Mehrheit. Ob es dann das Kleinere oder das Größere ist, also was die Mehrheit ist – so halt.

Gut. Dann bedanke ich mich recht herzlich.

INTERVIEW EXPERTE 4

Das Interview wurde am 30.03.2017 ab 10:30 in Pinkafeld durchgeführt. Experte 4 ist seit 2002 im Unternehmen und seit 2015 Projektmanager/Scrum Master in einem agilen Team. Experte 4 ist gleichzeitig Projektmanager und Teamleiter. Das Team von Experte 4 besteht derzeit aus sechs Personen.

Transkription

Beginnen möchte ich mit dem Allgemeinen, deiner Erfahrung mit agiler Projektentwicklung und Schätzungen. Die erste Frage dazu wäre, wie lange bei euch im Team agile Methoden für die Projektentwicklung bereits eingesetzt werden.

Da muss man jetzt einmal unterscheiden, was heißt „agil“, weil wir zu 99% A1-Themen abhandeln und das ein komplett anderer agiler Workflow ist, wie wir es jetzt in anderen Abteilungen verwenden da in der Firma. Das heißt, es gibt einen Prozess von der A1, das ist eigentlich ein agiles Fixpreis-Projekt. Das heißt, du hast den Projek-Scope schon vorgegeben am Anfang, der wird dann geschätzt, als wie wenn es ein Wasserfall-Projekt wäre. Weil genau auf das gibt es das Angebot, weil den Einkauf bei der A1 interessiert „agil“ nicht. Und dann, wenn das alles durch ist, kommt dieses Projekt nach einem gewissen Procedere in den A1 agilen Prozess hinein. Der hat Sprints, der hat Story-Points und so weiter und so fort. Allerdings ist er nicht das, was man unter agil versteht. Weil du dich immer in einem Kostenrahmen und einem Anforderungsrahmen bewegen musst. Schwierigkeit bei dem ganzen ist, dass man nie vergisst, dass man gewisse Anforderungen hat und was man umsetzen muss und agil dort nicht heißt, ich kann jetzt mit 2000 Stunden einfach irgendwas machen. Weil at the end muss das Produkt herausfallen. Oder diese Erweiterung des Produkts mit dem und dem. Das heißt, man versucht bei der A1 ein Wasserfall-Modell mit einem agilen Ansatz zu vermischen. Ist recht schwierig und stellt sich vor einigen Herausforderungen, weil natürlich auch der Auftraggeber sagt ‚Na, es ist ja agil, da können wir ja ...‘. Und da muss man immer aufpassen, dass man nicht abdriftet, sage ich jetzt einmal, und irgendetwas implementiert unabsichtlich, das gar nicht bestellt war. Und somit dann aus dem Aufwand hinausfällt.

Kommen wir vielleicht nochmal zurück zur ursprünglichen Frage: Wie lange macht ihr das schon in diesem System?

Kann ich jetzt gar nicht genau sagen, aber es sind schon einige Jahre. Ich kann es nicht sagen, aber fünf Jahre sind es sicher schon. Hätte ich mal gesagt.

Und diese Vorgehensweise ist eigentlich über die A1 zu euch ins Team gekommen?

Genau, das kommt von der A1.

Und wie war da bei der Einführung die Erfahrung?

Bei dem Teil war ich gar nicht dabei. Nur, es ist so, dass die Faustregel gilt, wenn es der Auftraggeber in dem Fall vorgibt, dann ist es so. Ganz am Anfang hat es einen motivierten Scrum-Master gegeben von der A1, der was da wirklich unabhängig war. Eine unabhängige Instanz, so wie es ja auch sein soll. Dem das Produkt an sich gleichgültig war – der hat keine persönlichen Interessen verfolgt, sagen wir mal so.

Und da hat es recht gut funktioniert miteinander alles, auch die Einführung. Und jetzt hat der sich vertschüsselt, sag ich jetzt einmal, und hat den ganzen Haufen sich selbst überlassen. Was ein bisschen schwierig ist, jetzt hängt das Ganze ein wenig bei uns, dass wir das Ganze auf Schiene halten. Aber durch das, dass die Prozesse schon eingespielt sind, läuft es ganz gut. Und von den Teammitgliedern her, sag ich einmal – es gibt Vorgaben, und hat es eigentlich keine großen Diskussionen gegeben. Ich meine, Ideen kann man immer zur Verbesserung – muss man schauen, was man umsetzen kann und was nicht.

Das heißt, du kennst es in deinem Team nur so, wie es jetzt ist, dieses System?

Genau.

Du hast es schon ein bisschen angesprochen, es ist offensichtlich Scrum mit einem Korsett außen herum – kann man das sagen, dass der agile Teil Scrum ist? Oder ist es was anderes?

Das ist schwierig zu sagen. Ich würde schon sagen, dass wir das schon nach Scrum verfolgen, das Ganze. Das Team arbeitet danach. Die kontrollierende Instanz muss halt immer beide Seiten vereinbaren, irgendwie. Das heißt, man muss immer genau das Angebot abklären und schauen und alles Drum und Dran. Aber in Wirklichkeit haben wir das bei uns ja auch, weil es gibt ja auch eine Projektdefinition, wo gewissermaßen der Rahmen vorgegeben ist, was das Produkt können soll, was die Erweiterung können soll. Und ich glaube, es ist dann nicht so unähnlich, sage ich jetzt einmal. Die Arbeitsweise. Weil du wirst genauso drauf schauen müssen, dass das Produkt ‚at the end‘ von diesen Sprints sage ich jetzt einmal, genau das kann, was eigentlich das Unternehmen will, und der Aufwand nicht überschritten ist unterm Strich. Also ich glaube jetzt, dass es recht ähnlich ist, das Ganze.

Kannst du ein Bisschen was drüber sagen, wie die Schätzungen bei euch ablaufen, nach welcher Methodik wird das gemacht und mit welcher Metrik wird das gemacht?

Das schaut so aus, dass – prinzipiell, wenn es richtig läuft, das Ganze – sollte es so sein, dass der technische Verantwortliche bei der A1 – ab und zu ist das auch jemand von uns – je nachdem, wie es im Projekt vereinbart wurde. Da gibt es also eine dezidierte Person, dass, die, sage ich jetzt einmal, eine User Story erfasst. Der legt die an, da gibt es eine Beschreibung dazu. Je nach Fachwissen des jeweiligen Kollegen ist die, sage ich jetzt einmal, in einem gewissen Genauigkeitsgrad. Dann kommt diese Story zu uns, zum Schätzen. Da gibt es jetzt mehrere Methodiken. Die bevorzugte Variante von mir ist, dass sich das Team, oder die Kollegen, die für das spezialisiert sind. Wir schauen nicht, dass das ganze Team das macht, dass alle – weil das können wir aufwandstechnisch nicht rechtfertigen, das geht nicht, sondern wirklich die, die es betrifft, dass die sich zusammensetzen, sich das anschauen, durchdenken, überlegen, Fragen notieren. Die gleich hineinschreiben – es sei denn, sie sind nicht kundentauglich, das gibt es natürlich auch – und dass man dann im Anschluss das mit dem Kunden auch abklärt, oder das einfach weitergibt, der das abstimmt. Und dann, nachdem alle Fragen geklärt sind, zu einer Zahl kommt. Story Points werden bei uns geschätzt. Bei uns ist es so, dass ein Story Point ein Tag ist. Angebotstechnisch sind es 10 Stunden allerdings. Als kleine Feinheit. Das heißt nicht, dass der Kollege 10 Stunden für die Implementierung brauchen kann. Sondern diese anderen zwei Stunden sind prozentuell der Anteil, der was für Deployment und Tests und Kundenbetreuung draufgeht. Das wird dann so, das ist erfahrungsgemäß hineingerechnet.

Und die Schätzung selber?

Macht das Team.

Sagt da einfach jeder, was er glaubt, was der Aufwand ist, oder gibt es da Planning Poker, oder ...

Genau. Also wenn es richtig abläuft, sitzen die Kollegen, die das betrifft, in der Runde. Das sind so zwei aufwärts, einer allein macht es kaum. Und sagt dann, ja ich glaube, dass es 4 sind. Bist du dir sicher? Ich glaube, dass es fünf sind? Warum? Ja, deswegen. Ja, hast recht, es sind doch 5. Zum Beispiel. Also so, wie du sagst Planning Poker. Also jeder sagt, was er glaubt, und dann wird diskutiert, warum. Optimalfall wäre natürlich, dass da der Kunde auch dabei ist. Die wünschen sich das auch oft, dass da der Auftraggeber dabei ist und die kennen das auch und da kann man auch offen über das reden. Das ist der Optimal-Weg, ja.

Das heißt, die Story Points sind de fakto der Aufwand.

Genau.

Nicht so wie in Scrum klassisch die Komplexität oder etwas in der Art, sondern es ist eine andere Einheit für Stunden.

Genau, es ist eine andere Einheit für Stunden, das ist richtig. Und auch, wenn wir auf Basis dessen ein Angebot abgeben, wenn zum Beispiel eine Einzelbestellung für eine Story erfolgt, dann wird das in Wirklichkeit mal 10 gerechnet und daraus fallen dann – mal Euro – fällt das Angebot heraus. Das der Kundenbetreuer dann abgibt.

Und ihr führt immer zuerst die Schätzung auf Basis von Stories durch und danach wird ein Angebot erstellt?

Der Optimalfall wäre es, ja. Der andere Weg ist, wir bekommen Anforderungen, das ist teilweise per Mail, teilweise per Word. Daraus überlegen wir uns, was das sein kann, füllen daraus ein rMDATA-Abschätzungs-Dokument aus, mit richtigen Stunden. Von, bis, erwartet und so weiter. Und auf Basis von dessen werden dann Story Points zurückgerechnet und ein Angebot erstellt. **Risiko bei der Sache ist natürlich, dass man sich halt stark verschätzen kann auch, weil man die Anforderungen nicht im Detail kennt.** Wissen wir eh, wie das abläuft, kennen wir eh. Aus allen Ecken der Firma.

Wenn so ein Fall auftritt, dass es zuerst dieses Angebot gibt, werden dann später nochmal Stories draus gemacht ...

Ja.

... und dann wird es, so wie du es vorher beschrieben hast, ...

Ja.

... geschätzt und dann ist im Optimalfall die Summe der Schätzungen der Stories kleiner oder gleich ...

Genau, das wäre der Optimalfall. Genau, das ist richtig. Das passiert. Also man hat zum Beispiel eine Anforderung von der A1. Wir möchten das, das, das haben. Es gibt noch keine Stories, es gibt noch gar nichts. Es gibt vielleicht nur etwas Zusammengeschriebenes als Doku, aber keine einzige User Story. Deshalb habe ich ja vorher gesagt ‚Optimalfall‘. Den gibt es leider sehr selten. Dann reden wir noch mit ihnen, dann gibt es einen Workshop, einen kleinen, damit man sich mal ein Bild davon schaffen kann, offene Fragen sammeln. Und dann versuchen wir das Ganze in eine Abschätzung zu gießen, und tun aber auch schon für den Vertrieb Abgrenzungen definieren. Wo wir sagen, wir treffen für die Abschätzung diese und jene Annahme. **Und dann kommen diese Abgrenzungen, wenn sie für den Auftraggeber OK sind, aber auch in das Angebot hinein. Das hat sich gezeigt, das ist notwendig.** Wenn das Ganze dann durch ist, wie du gesagt hast, werden Stories erstellt mit den Beschreibungen und im Optimalfall decken sie sich und wenn nicht, dann haben wir ein Problem. Natürlich, ja. Weil die Abschätzung kann natürlich beim detaillierten Ausarbeiten später dann auch natürlich zu wenig gewesen

sein. Diese Fälle gibt es immer wieder. Da kann man nur drauf hoffen, dass man sich bei anderen Dingen positiv verschätzt hat und sich das wieder ausgleicht. Das ist recht schwierig, das Thema.

Das heißt, man schaut sich das schon an, wie ist die Summe der Stories im Vergleich zum Angebot.

Das mache ich immer wieder. Das ist auch für den Statusbericht sehr relevant natürlich. Wie der basiert ja auf Stunden. Und genau dort passiert das. Da habe ich dann die Aufschlüsselung der Abschätzung, umgerechnet auf Story Points oder Stunden. Das hole ich mir aus dem LZE heraus, weil gebucht werden darf nur auf User Stories. Das impliziert, dass es Stories geben muss. Und somit kann ich dann genau bewerten, ob wir gut liegen oder ob wir schlecht liegen pro geschätztem Punkt.

Das heißt, es gibt also auch ein laufendes Controlling.

Genau, das mache ich laufend.

Also den Vergleich zwischen Schätzung und tatsächlichem Aufwand.

Ja, genau. So wie du das gemacht hast, in dem Beispiel, das du mir gezeigt hast (ANMERKUNG: DER PROJEKTSTECKBRIEF EINES PROJEKTS). Das mache ich laufend, um quasi rechtzeitig schon zu warnen, oder Gegenmaßnahmen einzuleiten. Oder zu sagen, das kann sich nicht ausgehen, wir liegen komplett falsch, oder es passt, oder was auch immer. **Also dieses Controlling ist sehr wichtig, sonst kannst du es nicht im Blick behalten.**

Und wie sind da die Erfahrungen im Hinblick auf die Genauigkeit der Schätzungen?

Das hängt immer davon ab, wie gut schon die Anforderungen zu uns gekommen sind. Also, sollte es wirklich der Fall sein, dass schon User Stories zu uns kommen, das sind zum Beispiel so Kleinbestellungen. Das sind zum Beispiel: ‚Ja, wir haben 3 User Stories, die sind eine kleine Erweiterung für dieses oder jenes Produkt‘. Und da redet man vorher, macht sich das aus, dann ist die Abschätzung meiner Meinung nach sehr gut. Immer. Da ist eigentlich auch keine Diskussion mehr, das ganze wird bestellt, die Bestellung ist da, wir beginnen zu arbeiten, lassen das abnehmen, die Sache ist fertig. Vielleicht gibt es noch eine kleine Abrundung, weil man etwas vergessen hat. Aber das hält sich in Grenzen und da sind wir eigentlich, sage ich einmal, liegen wir ganz gut. Bei Punkten, die wir wirklich nur sehr vage hereinkriegen und wo wir dann aus strategischen Gründen rechtzeitig eine Abschätzung abgeben müssen, für das Angebot, damit das Ganze dann ins Rollen kommt, bis es zu einer Bestellung kommt, sage ich einmal, habe ich eher gemischte Gefühle, um das so zu sagen. Da finde ich, ist es eigentlich mehr ein Glücksspiel, ob das wirklich zusammen passt, oder nicht dann. Weil jetzt so bei riesigen Projekten, die reinkommen, wie zum Beispiel die KOPL-Detailplanung – wird jeder schon gehört haben – mit über 7.000 Stunden, da ist die Abschätzung immer wieder mal angepasst worden, descoped, was dazugegeben, was weg. Und at the end ist es da sehr schwer, den Überblick zu behalten, ob man das jetzt wirklich alles eingehalten hat, oder nicht. Und da muss man echt sagen, hängt es auch sehr vom Auftraggeber ab, dass er sagt, ‚Ja, OK, wir haben uns da jetzt in einem sehr agilen Rahmen bewegt, zum Beispiel, in diesem Projekt. Und deshalb fordere ich diesen oder jenen Punkt auch nicht ein, weil ich sehe das ein, dass das quasi als Stundenpool mutiert ist und ich jetzt einfach wirklich nach Scrum arbeite. Weil ich habe gesagt, da habe ich 7.000 Stunden und da arbeite ich, bis es aus ist, quasi‘. Wenn der natürlich alles einfordert, was wirklich mal gesprochen wurde, dann wird es schwierig, natürlich. **Ich sage mal, bei Projekten, wo die Abschätzung bis, sage ich mal, 2.000 Stunden oder so geht, glaube ich, liegt man meistens recht gut.** Alles, wo dann die Abschätzung, die rauskommt, deutlich größer 2.000 Stunden aufwärts geht, muss man dann deutlich hinterfragen, ob man nicht noch mehr in eine Konzeption vorher investiert. Sprich die Erstellung von User Stories und einer genaueren Definition, um die Abschätzung zu verfeinern. Weil da muss ich echt sagen, dass das Risiko sehr hoch ist.

Das heißt, da geht es eigentlich um die Schätzungen des Gesamt-Projekts. Und wenn man die Schätzungen der User Stories anschaut, gibt es da Erfahrungen, wann die Schätzungen gut sind oder wann sie weniger gut sind? Zu dem Zeitpunkt, wo es schon User Stories gibt – die kriegen dann eine Abschätzung und danach erfolgt die Umsetzung. Gibt es da irgendwelche ...

Ja. Das A & O, habe ich festgestellt, ist immer die Beschreibung. Die muss so, sein, dass wenn es sich derjenige, der es umsetzt, der Techniker durchliest, ohne viel Fragen anfangen kann, ohne zu implementieren. Und wenn sich derjenige, der das bestellt hat, sich das durchliest, sage ich jetzt einmal, den nicht technischen Teil, auch sofort weiß, aha, was habe ich mir nachher zu erwarten. Wenn die beiden Fälle gegeben sind, ist die Abschätzung im Normalfall auch sehr exakt treffend. Dazu gehört auch, dass bei der Abschätzung immer berücksichtigt wird, implementiert es jemand, der nicht der Top-Profi auf dem Produkt ist. Ein Beispiel: **Eine Abschätzung, die Haupt Michael zum Beispiel fürs Web-GIS-Professional trifft, kann ich nicht umlegen auf jeden anderen Kollegen. Das geht einfach nicht.** Das ist jetzt nichts gegen die anderen Kollegen, sondern das liegt einfach daran, dass diese Domäne in diesem Detailwissen kann nur jemand, der seit 15 Jahren nichts anderes macht. Deswegen muss ich immer sagen, zu demjenigen Kollegen, ‚Hast du berücksichtigt bei deiner Schätzung, dass du das höchstwahrscheinlich nicht selbst implementierst?‘ Das ist zu beachten. Also das ist mal der erste Faktor. Wird es wer implementieren, der nicht top tief in diesem Produkt drinnen ist. Das muss man immer berücksichtigen. Ist zwar vielleicht ein bisschen ungut dem Kunden gegenüber, weil der erwartet sich natürlich, dass das jeder gleich gut kann, aber die Realität ist einfach so, das ist eine Andere. Dazu ist es auch noch notwendig, man muss auch immer darauf schauen, als Projektmanager, dass, wenn diese Stories definiert werden, sie wirklich so geschrieben werden, dass es auch in Nicht-Profi versteht, das Ganze. Der nicht 100%ig in dem Thema drinnen ist. Weil jemand, der sich bei dem Thema top auskennt, neigt dazu, bei den Beschreibungen etwas wegzulassen. Die ‚eh klar sind‘. Und das, was ‚eh klar ist‘, ist das, was immer das Problem macht. Das macht immer das Problem. Immer wenn man sich denkt ‚eh klar‘, muss man sofort schalten und das aufschreiben. Weil das ist immer – da fällt man immer hinein. Das sind die wichtigsten Themen. Ja. Du hast eh schon herausgehört, bei uns hängt das alles von der Genauigkeit der Dokumentation ab. Erst wenn die perfekt ist, und auch richtig abgegrenzt ist, das kommt auch noch dazu. Abgrenzungen sind ganz wichtig. Dass immer drinnen steht, was kann ich mir nach dieser Umsetzung nicht erwarten. Das ist nämlich oft auch ein Problem. Weil es gibt doch oft vom Kunden dann die Aussage ‚Ja, aber ich habe mir gedacht, das geht dann auch‘. Und das führt dann auch zu Diskussionen. Diskussionen kosten auch Zeit, und wenn dann leider entschieden wird zusammen mit dem Vertriebe sage ich jetzt einmal, ‚Ja, das hätten wir vielleicht wissen müssen‘, dann kann es sein, dass man wieder zu einer Überschreitung kommt. Weil man dann das auch noch nachliefern muss.

Ja das heißt ... Wann diese Schätzungen durchgeführt werden, haben wir, glaube ich, auch schon ein bisschen herausgehört. Es gibt einerseits die grobe Schätzung, die möglicher Weise für ein Angebot benötigt wird. Und die Schätzung von den Stories, ist das dann relativ knapp vor der Umsetzung, oder ist das manchmal auch weiter weg. Die Schätzung und Umsetzung?

Da gibt es zwei Zugänge zu dem Ganzen. Das eine ist, wenn es sich um ein größeres Projekt handelt, wo es vorher noch keine User Stories gegeben hat, dann erfolgt die Schätzung der Stories, bevor es zum Sprint Planning kommt. Das heißt, im Zeitraum vier Wochen, bevor der Sprint beginnt. Bei uns ist ein Sprint circa 4 Wochen lang. Da werden dann die Stories definiert, fertiggestellt und geschätzt. Die Schätzung hier, sage ich jetzt einmal, ist eigentlich jetzt, dient zu mehreren Dingen. Das eine ist eine Kontrolle, ob die ursprüngliche Schätzung für das Angebot eh stimmt und zur Ressourcen-Planung. Weil auf Grund der Story Points kann ich meine Ressourcen einteilen. Also für das brauchen wir das. Und der andere Weg ist, wenn es die Stories vorher schon gegeben hat, und sie geschätzt wurden, dann wird da nichts mehr angeschaut. Im Normalfall. Das heißt, wenn so Kleinbestellungen sind, so 3-4 Stories, die

sind vorher schon definiert, sind geschätzt, dann sind die einfach im Sprint drinnen und dann geht man davon aus, dass die fertig sind. Da wird eigentlich kaum mehr darüber gesprochen. Maximal, dass ich mir das nochmal anschau, das Ganze, ob es eh noch schlüssig ist und – ganz wichtig – ob nach der Angebotslegung nichts mehr verändert wurde an der Story. Wir verwenden bei der A1 das System Confluence, für Dokumentation, das hat eine Änderungshistorie, wie ein Subversion, sage ich einmal, und da kann ich sehr wohl nachschauen, ab da nachträglich jemand etwas geändert hat. Hat sich in letzter Zeit herausgestellt, muss man machen. Weil da könnten manche glauben, das ist ein Wunschkonzert dann nachher. Ich habe den Rohbau bestellt, warum sind keine Fenster jetzt drinnen, kein Boden und kein Dach. Genau so kommt es einem ab und zu vor dann. Deswegen muss man da immer schauen, dass da nichts ist. Und deswegen habe ich vorher gemeint, Abgrenzungen sind ganz wichtig auch. Was erwarte ich mir nicht.

Und macht es einen Unterschied für die Genauigkeit, wie lange vorher das geschätzt worden ist?

Wenn so eine Story schon wirklich – wenn man sagt, man hat das vor einem Jahr gemacht, oder ein dreiviertel Jahr ist so die Schwelle, und es ist dann so lange bei der A1 gelegen, oder beim Kunden sage ich jetzt mal, allgemein so lange gelegen, bis man sich dazu entschließt, man möchte es haben, sollte man es sich schon nochmal anschauen. Mit den Kollegen. Und durchdenken, zumindest eine halbe Stunde. **Weil es kann sich in der Zwischenzeit schon so viel getan haben, dass wir da noch etwas Essentielles mitberücksichtigen müssen, was den Aufwand deutlich erhöht.** Das ist richtig, das muss man dann machen. Wenn das nur ein paar Wochen her ist, dann muss man da nichts mehr tun. Das passt dann im Regelfall. Aber wenn es länger liegt, dann muss man es sich anschauen. Und neu bewerten. Kann sein – es hat schon Fälle gegeben, wo man dann gesagt hat, weil wir haben zwischendurch was Neues eingeführt, und das kostet jetzt deutlich mehr. Oder, der andere Fall, ja eigentlich kostet es jetzt nur mehr die Hälfte, weil das andere Projekt hat den Teil schon mitfinanziert. Gibt es auch. Kann beide Richtungen gehen.

Wer die Schätzungen durchführt, das hast du auch schon angesprochen. Das gibt es offensichtlich manchmal, dass Personen schätzen, die das nicht durchführen, und das berücksichtigt ihr dann aber schon vorab explizit.

Ja, das muss man immer. Das muss der Projektmanager, sage ich jetzt einmal, muss das selbst immer hinterfragen, weil derjenige Kollege denkt dann meistens nicht daran, weil für ihn ist es eh klar. Und da muss man sich vorher schon ein bisschen über die Planung Gedanken machen, wird das dann auch der Kollege machen. Es gibt gewisse Dinge, die macht dann auch der, der es geschätzt hat. Und es gibt gewisse Punkte, die wird er ganz sicher nicht machen. Da ist es dann auch sinnvoll, dass dann die Kollegen auch dabei sind, bei der Schätzung einfach. Ist eh nicht außer der Welt, man sitzt eh' im selben Büro. Und dann setzt man sich einfach kurz zusammen für zehn Minuten und macht eine Schätzung, wo man dann auch sagt, ja, wird einen Tag mehr kosten, weil der sich erst einarbeiten muss, oder so.

Das wird dann also explizit berücksichtigt.

Ja, das muss man berücksichtigen.

Ein Faktor, der die Genauigkeit von Schätzungen beeinflusst bzw. beeinflussen kann, ist, wenn es einen starken Druck von außen gibt, einen Zeit- oder Kostenrahmen einzuhalten. Ich habe jetzt verstanden, das kann bei euch dann passieren, wenn man zuerst das Projekt schätzt, danach die Stories schreibt und dann draufkommt, dass die Schätzungen nicht zusammen passen.

Genau, dann habe ich ein zeitliches Problem. Weil man hat vom Aufwand her ausgerechnet, dass es sich abzüglich anteiliger Urlaube und so weiter, bis dort und da ausgeht. Und wenn man jetzt sieht, es geht sich nicht aus, ...

Beeinflusst das dann die Umsetzung oder beeinflusst es auch die Schätzung.

Du meinst, dass absichtlich weniger geschätzt wird, damit es sich ausgeht?

Absichtlich oder unabsichtlich.

Wir vermeiden das, so gut es geht. Weil das ist ja nur sich selbst anlügen. Das ist in der Vergangenheit garantiert passiert, solche Themen. Dass man das im Hinterkopf gehabt hat und dann gesagt hat ... oder dass man mit Überstunden gerechnet hat, was man ja nicht soll, weil die braucht man meistens eh schon für etwas anderes. Das ist ein Problem, das man erkannt hat schon, dass da was passiert und das wird bewusst vermieden. Da wird dann gesagt, ja aber es hat geheißen, bis dort und dort. Ja, aber wenn es sich nicht ausgeht, geht es sich nicht aus. Das muss man dann auch dem Kundenbetreuer entsprechend erklären. Ganz wichtig natürlich auch warum das Ganze. Nicht einfach, es geht sich nicht aus, weil mit dem kann keiner was anfangen, das wirst du eh kennen. Sondern warum geht es sich nicht aus und wieso ist der Aufwand da, und dann geben wir das meistens dem Kundenbetreuer weiter, der es sich dann ausmacht mit dem. Und wenn man es früh genug weiß, ist es meistens auch kein Problem. Aber wie du schon sagst, man kann die Zeit nicht beugen, noch nicht, somit: Wenn man den Aufwand hat, hat man ihn. Wir schauen da in den letzten Monaten, Jahr, verstärkt drauf, dass wirklich immer alles realistisch ist und wenn man es schon bei der Angebotslegung feststellt, es kann sich nicht ausgehen, ja. Dann muss man das rechtzeitig eben sagen. Also das versuchen wir zu vermeiden, so etwas.

Gibt es einen Einfluss auf die Genauigkeit der Schätzung, je nachdem was gemacht wird - gibt es eigentlich unterschiedliche Dinge, die ihr macht, was Technologien betrifft, oder Domänen oder Programmiersprachen, die die Genauigkeit von Schätzungen beeinflussen?

Bei uns trifft das eigentlich eher mehr auf das Produkt an sich zu. Weil wir ja mehrere haben. Wir fallen ja bei der A1 unter die Netzplansysteme hinein. Da gibt es verschiedene Systeme. Und der Aufwand wird auch je nach System etwas anders bemessen. Das hängt jetzt davon ab, welche Prozedere für einen Rollout notwendig sind und so. Und zum Beispiel mein Lieblings-Thema, SARU. Zum Beispiel das Produkt SARU ist auch eines der vielen Netzplansysteme, die wir betreiben. Hier wissen wir zum Beispiel schon, dass der Kundebetriebsaufwand für uns extrem hoch ist. Das müssen wir in der Abschätzung berücksichtigen. Das heißt, Abschätzungen fallen dort höher aus, als bei anderen. Weil, wir wissen schon von vornherein, dass das Drumherum mindestens ein Drittel des Aufwandes für die Implementierung ist. Das heißt, wenn ich dort eine Abschätzung bekomme für – keine Ahnung – der sagt mir x Story Points dauert das, dann kann ich dort schon mal mindestens 40% draufschmeißen, weil ich weiß, 40% ist rein nur bla bla bla dort. Diskussionen und reden, und die wollen einfach so intensiv betreut werden. Und dann müssen sie das aber auch zahlen. Bei anderen Produkten weiß ich, dass das alles easy going ist, da ist dieser Anteil viel niedriger. **Das sind einfach Erfahrungswerte, die wir über die Jahre gesammelt haben, wo man einfach schon einmal „eingefahren ist“ auf gut Deutsch, wo man dann nachträglich analysiert hat, warum ist das so schief gegangen, und festgestellt hat: Ja, dieses und jenes, und das müssen wir in Zukunft mitberücksichtigen dort.**

Und der tatsächliche Aufwand und die ursprüngliche Schätzung, das passt dann aber trotzdem zusammen. Wenn diese Faktoren berücksichtigt sind, dann kann man sagen, das ist eigentlich unabhängig von den Projekten, das passt immer gleich gut zusammen?

Du meinst, der Faktor, den man da aufschlägt? Oder die Schätzung an sich dann?

Die Schätzung an sich. Wenn ich jetzt sage, das was ich alles weiß, habe ich berücksichtigt.

Ja. Dann passt es gut zusammen. Das haben wir jetzt verstärkt in den letzten Schätzungen aufgenommen, diese Themen, weil wir eben dieser single vendor geworden sind, diesen Rollout Prozess, diesen UAT-Prozess. Bei manchen Produkten ist er mehr, bei manchen ist er weniger erfahrungsgemäß. Und seit dem, wenn wir diesen richtig berücksichtigen, und ihn dezidiert noch genauer aufschlüsseln in den Aufwänden, das interne Abschätzungs-Papier bei uns natürlich, treffen wir den Aufwand besser. Weil der hat früher nämlich immer gefehlt. Das sehen wir jetzt zum Beispiel gerade beim großen KOPL-Projekt, dort hat man die Umsetzung geschätzt, so gut man konnte. Das war wirklich nicht genau, aber mehr hatte man nicht, war trotzdem eine große Summe. Und dort – das ist eine ältere Abschätzung – wurde nie berücksichtigt, diesen ganzen Prozess, den man bei der A1 zu befriedigen hat. Und dann ist es auch noch so weit gekommen, dass einen Vor-Ort-UAT der Kunde nach jedem Sprint wollte. Den konnte man ihm auch nicht abschlagen, weil er tatsächlich notwendig war. Man muss aber immer zwei Leute für zwei Tage hinschicken. So, und jetzt rechnen wir das mal kurz auf: 2 Leute, 2 Tage, das sind aber 12 Stunden fast, jeden Tag. Da reden wir schon von 6 Story Points im Schnitt, die fehlen im Sprint. Was aber aufs Projekt gebucht werden muss, bei einer Projektlaufzeit von über einem Jahr, kannst du dir jetzt denken, wie viele Stunden da jetzt auf einmal fehlen, was wir eigentlich implementieren hätten müssen. Also wie das jetzt zusammengehen soll am Schluss, das weiß keiner so genau. Da merkt man schon, wenn man das nicht rein gibt, diese Themen, da hat man ein massives Problem. Je größer das Projekt, umso stärker wird es. Und da reden wir dann nicht von einer 20-Stunden-Überschreitung, sondern da gehen wir dann gleich in die hunderte Stunden hinein.

Das heißt, wenn man Checklisten oder Erfahrungswerte hat, was alles zu berücksichtigen ist für eine Schätzung, seitdem sind Schätzungen genauer.

Genau. Das ist gut, dass du das sagst, das **Stichwort Checkliste**. Wir haben eine für das. Wir haben eine so und so für den ganzen A1-Scrum-Prozess, eine Checkliste, für jeden einzelnen Mitarbeiter. Und da gibt es auch eine für die Abschätzung. Wo man wirklich die Punkte listet – ich habe sie jetzt leider nicht da, ich kann sie dir mal schicken, wenn du magst – und da steht wirklich genau drinnen: **Schätzungen, was habe ich alles zu berücksichtigen, was habe ich nicht vergessen: das, das, das** ... Da stehen auch so Dinge drinnen wie Kundenbetreuung, UAT, Ausrollen, Developer Acceptance Test. Und das muss ich einfach mit schätzen. Und es scheint bei manchen Projekten zwar viel, dass man zum Beispiel sagt das sind fast 35% vom eigentlichen Implementierungs-Aufwand, und dann legt man das kurz dar, und dann heißt es von allen Beteiligten: ‚Ah, ja, das ist wirklich so‘. Wir sind mittlerweile in einem Zeitalter, wo man nicht einfach blind hineinhackt, so wie es rmDATA vor 15 Jahren gemacht hat, sondern es gibt Prozesse und es gibt einen Kundenbetreuungsanteil. Und den muss man einfach mit abschätzen. Wenn man den vergisst, fährt man ein. Ist Fakt.

Ok, dann sage ich mal danke: Dankeschön.

INTERVIEW EXPERTE 5

Das Interview wurde am 03.04.2017 ab 16:00 in Pinkafeld durchgeführt. Experte 5 ist seit 2012 als freier Dienstnehmer für das Unternehmen tätig und seit 2015 Teamleiter in einem agilen Team. Vor 2012 war Experte 5 bereits nebenberuflich für das Unternehmen tätig. Er hat Software-Engineering an der TU Wien studiert. Das Team von ihm besteht derzeit aus fünf Personen, bis vor kurzem aus 5 ½ Personen.

Transkription

Die erste Frage wäre, wie lange bei dir im Team agile Methoden für die Projektentwicklung bereits eingesetzt werden.

Bei uns im Team grundsätzlich seit einem Jahr. Aus dem Grund, weil es unser Team erst seit einem Jahr gibt. Angefangen grundsätzlich damit haben wir schon 2011/2012. Da haben wir die ersten Telekom-Projekte gemacht. Zwar jetzt nicht bei uns im Team, aber generell die Teams, die da jetzt gearbeitet haben. Die hatten das bereits im Einsatz. Damals, muss ich ehrlich gestehen, noch besser wie heute. Eigentlich aus dem einfachen Grund, weil damals gab es bei der A1 noch einen Scrum-Master, den es in dieser Form heutzutage nicht mehr gibt. Das heißt, damals wusste jeder besser um die ganzen Regeln, die es gibt bei Scrum Bescheid, und aus dem Grund war das auch um einiges bekannter, als es jetzt ist. Wie das ganze vor sich geht, was wirklich ein Story-Point ist, wie man schätzt, und so weiter. Der hat das damals wirklich forciert, und das geht mittlerweile etwas verloren.

Wie waren da so die Erfahrungen bei der Einführung der agilen Methoden?

Für mich grundsätzlich war die Einführung relativ einfach, weil ich es aus der Theorie vom Studium her kannte, ich habe ja Software-Engineering studiert, und von dem her war mir die Theorie sehr gut bekannt. Interessant für mich war dann wirklich die Praxis, wie man das dann wirklich umsetzt.

Und von den Teammitgliedern her, hat das auch gut funktioniert? Oder hat es da irgendwelche Umstellungs-Schwierigkeiten gegeben?

Eigentlich nicht, muss ich sagen. Also das ist eigentlich relativ flott gegangen. Ich merke es nur jetzt, dass sich die Leute sehr schwer tun, und auch die ganzen Konzepte dahinter nicht wirklich verstehen, weil es eigentlich keinen Kümmerer mehr gibt. Und das ist das Problem aus meiner Sicht, warum jetzt nicht die Schätzungen so aussehen, wie sie aussehen sollten. Oder nach der Methodik vorgehen, wie man vorgehen sollte.

Du hast es schon angesprochen: Die agilen Methoden. Ursprünglich war es mehr Scrum, offensichtlich. Jetzt ist es weniger Scrum – wie kann man sich das vorstellen?

Wie kann man sich das vorstellen? Früher war es wirklich strikt. Wenn man zum Beispiel gesagt hat, wir schätzen, dann hat es so ein Meeting gegeben, das war das Planning Poker. Da haben wir ein Planning Poker gemacht, so wie man es aus der Theorie kennt. Da hat man sogar mit den Karten teilweise, vor Ort hat man geschätzt. Jeder hat die Karten gehabt, die aufgliedert waren nach den Fibonacci-Zahlen und so wurde geschätzt. Und das wurde eigentlich bei jedem Projekt so durchgezogen, kann man sagen.

Mittlerweile ist das nur teilweise mehr so. Diejenigen Product-Owner, die etwas schätzen lassen und die das kennen, die machen das teilweise auch so, die meisten aber nicht.

Wie wird dann geschätzt?

Also dann wird wirklich ... Also bei uns ist es so, nach dem Prozess, wir haben grundsätzlich eine Grobkosten-Schätzung. Und dort ist es teilweise schon so, dass man es schon sehr detailliert schätzt. Also teilweise ist es gar keine Grobkosten-Schätzung, sondern wirklich schon auf Story-Ebene, auf Anforderungs-Ebene. Und dann braucht man es nur mehr eintragen, was man geschätzt hat, ursprünglich. Vielleicht was noch interessant ist, vom ganzen Prozess her. Momentan ist jetzt einmal so die Theorie: man macht eine Grobkosten-Schätzung. Und da gibt es die sogenannten T-Shirt-Sizes. Und mit denen soll es möglich sein, schon gewisse Größen anzugeben, damit der Kunde sagen kann schon von vornherein: ‚Nein, so viel Budget haben wir nicht, oder haben wir‘. Das heißt, man hat eine T-Shirt Size XS, man hat eine T-Shirt Size S, M, XL, L. Alles steht für von-bis Kosten, Kostenbetrag und da gibt man eine erste Auskunft Daumen mal Pi wo man sich bewegt, sozusagen. Das heißt, wenn man jetzt im Bereich L ist und er sagt, er hat da eh nur ein Budget in Form von einem S-Budget quasi, dann sagt er, ja, lassen wir gleich weg.

Und welche Größenordnungen sind das, diese T-Shirt-Größen?

Ich weiß es jetzt nicht auswendig. XS glaube ich, ist bis 5.000 Euro, S ist bis 10, 15.000. Ich müsste jetzt lügen. Ich weiß es jetzt nicht auswendig, aber die Staffelung kann ich dir geben. (ANMERKUNG: DIE STAFFELUNG IST: XS BIS € 10.000, S BIS € 25.000, M BIS € 50.000, L BIS € 100.000, XL BIS € 250.000). Mit dem wollen wir jetzt einfach mal beginnen, weil wir in der Vergangenheit sehr viel Zeit für Schätzungen vertan haben, wo das dann einfach nicht gekommen ist. Wenn sie gewusst hätten von vornherein, welche Größenordnung das ist, dann hätte man das auch nicht schätzen brauchen, so genau. Aber das ist jetzt die Form, wie man eine Grobkosten-Schätzung machen sollte und dann, von der Theorie her, sollte man es wirklich weiter pro Sprint betrachten, pro Story und in Form von Planning-Poker. Ist größtenteils nicht so. Da macht man es so dann, dass meistens ich mich, mit einem Kollegen, der ein Wissender ist, zusammensetze und wir das einfach schätzen. Man muss wirklich sagen, derzeit ist es wirklich so: Wir machen es nicht so, wie es funktionieren sollte. Das heißt, wir haben ja vorher schon gesagt, eigentlich sollte man ja die Eigenschaft Komplexität schätzen, das tun wir definitiv momentan nicht. Also bei uns ist das in Wahrheit wirklich eine Zeitdauer, so 10 Stunden ist ein Story Point beispielsweise. Weil man sagt – ich sage jetzt eine Hausnummer – 6 Stunden für Implementierung, 2 Stunden für UAT, Vor- und Nachbereitung und 2 Stunden noch einen Puffer oder so etwas. Also das ist de facto, wie wir momentan schätzen. Und das tun wir dann einfach in dem Kreis, in dem sich die Wissenden befinden.

Das heißt, bei der Schätzung ist nicht das gesamte Team dabei, sondern nur ein Teil.

Teilweise. Wie gesagt, wenn der Product Owner derzeit noch dahinter ist, der diese Vorgehensweise kennt, dann machen wir es auch im ganzen Team. Auch wenn wirklich da Leute dabei sind, die das gar nicht kennen. Ich habe vorher gesagt, wir machen es teilweise richtig. Warum teilweise: Weil grundsätzlich nach Stunden sollte man nicht schätzen, das tun wir aber, aber warum teilweise. Wir machen es bei den Schätzungen intuitiv richtig, indem man oft sagt, also ich habe jetzt für die letzte Story 2 Story Points geschätzt und für die schätze ich jetzt 4, weil ich einfach sage, das ist der doppelte Aufwand. Da macht man es dann intuitiv, intuitiv macht man es eigentlich richtig, wie man es machen sollte. Ja, deswegen halbrichtig.

Gibt es da eine Erfahrung, ob es einen Unterschied macht, ob nur ein Teil schätzt, oder ob alle schätzen? Hat das einen Einfluss auf die Genauigkeit?

Es hat auf jeden Fall einen Einfluss auf die Diskussion dann. **Wenn mehrere schätzen, hast du mehrere Vorteile.** Du hast ... Es werden dann wirklich unterschiedlichste Schätzungen herauskommen, und am Ende des Tage muss ja eine gleich Schätzung herauskommen. Und da hast du erstens den Vorteil, es gibt mehrere Wissende, denen das erklärt werden muss: warum, was, wie ist. Und da schätzt man einfach so lange, bis wirklich die gleiche Zahl dort steht.

Gibt es einen Vergleich zwischen dem geschätzten Aufwand und dem tatsächlichen Aufwand?

Können wir im Nachhinein machen, auf Grund dessen, weil wir teilweise in Stunden schätzen. Können wir machen. Wir könnten uns ein paar Projekte heranziehen. Sehr oft ist es aber so, dass man einfach schaut, dass man dort hinkommt.

Es wird aber nicht laufend ein Controlling gemacht?

Nein, **wir machen schon ein Controlling wo wir sagen: geht sich bis zum Ende aus, oder geht sich nicht aus. Wo müssen wir vielleicht „effizienter“ werden, damit es sich dann trotzdem noch ausgeht.**

Das bezieht sich dann aber immer auf ein Gesamtprojekt.

Richtig, das bezieht sich immer auf ein Gesamtprojekt, ja.

Und hast du einen Eindruck, oder eine Vorstellung davon, welche Faktoren für eine bessere oder genauere Schätzung ausschlaggebend sind? Hast du da irgendwelche Erfahrungen: Wenn wir dieses oder jenes machen, dann sind unsere Schätzungen gut, wenn wir etwas anderes machen ...

Wir reden jetzt wirklich von Arbeitspaketen?

Ja.

OK. Ja, das ist ganz klar. **Also bei uns ist es umso besser, umso besser die Anforderungen beschreiben sind in der Story. Umso genauer können wir schätzen.** Das ist einfach so. Es stellt sich schon bei der Schätzung heraus, wenn da teilweise zwei Sätze drinnen stehen, und der Kunde selber nicht weiß, wie das dann im Endeffekt aussehen soll, und wir es einfach wissen, weil wir die bessere Expertise haben, dann wird es meistens nicht hinkommen. Weil dann der Kunde kommt: Und es wäre noch schön, wenn wir das machen würden, oder das machen würden. Das heißt, umso genauer du wirklich die Anforderungen, die Akzeptanzkriterien und so weiter drinnen stehen hast, umso genauer kann man das schätzen. Definitiv. **Und natürlich, umso mehr Wissende dabei sind, umso genauer wird es natürlich auch. Wenn jetzt lauter Programmierer etwas schätzen, bei einer Applikation, die sie gar nicht kennen, dann wird es natürlich auch ungenau.** Hatten wir auch schon.

Was für Arten von Applikationen, was für Arten von Projekten oder Technologien, habt ihr, wo ihr Projekte umsetzt. Ist das immer was Bekanntes, was Ähnliches, was Gleiches. Oder gibt da viele unterschiedliche ...

Seit letztem Jahr sind es sehr viele unterschiedliche. Wir haben einerseits – einerseits machen wir eben Spezialprodukte, das heißt, Standardprodukte erweitern wir oder adaptieren wir auf die Kundenwünsche. Das ist eine Schiene. Die zweite Schiene man hat wirklich – man arbeitet an einer Software oder einer Applikation, die dem Kunden schon gehört. Das ist im Web-Bereich, das ist seit letztem Jahr auch im Fat-Client-Bereich. Sonst fällt mir eh nichts ein. Und wir tun ja nicht nur Applikationen erweitern, sondern wir machen auch Dienstleistungen. Das heißt, wir machen auch Analysen, wir machen Konzepte, und, und, und. Also bei uns kann es auch Konzeptionsprojekte geben, die beauftragt werden, was dann nichts mit Implementierung zu tun hat. Aber die muss man auch schätzen.

Und gibt es da Unterschiede, wie genau Schätzungen sind?

Unterschiede bei was?

Wenn man sagt, das sind Standardprodukte, die erweitert werden, oder es ist eine Kundenapplikation? Gibt es da Erfahrungen, ob das einen Unterschied macht?

Im Regelfall natürlich. Weil gerade wenn man Standard-Produkte erweitert, da hat man natürlich mehr Programmierer, die was schon bei der Standard-Produktentwicklung dabei waren. **Und sie sich natürlich besser auskennen und natürlich ganz genau wissen, an welchen Schrauben man drehen muss, dass man die Anforderung des Kunden erfüllen kann.**

Wann werden üblicher Weise die Schätzungen durchgeführt? Ist das relativ knapp vor der Umsetzung, oder ist das manchmal auch weit weg von der Umsetzung?

Komplett unterschiedlich. Das hängt wirklich von dem Product-Owner ab. Es gibt Product-Owner, bei denen haben wir quasi den richtigen Prozess gemacht. Da haben wir es teilweise davor gemacht, also da haben wir schon für die nächsten drei, vier Sprints geschätzt. Wodurch man natürlich im Sprint mehr Zeit für andere Dinge hat. Gerade bei anderen Applikationen, wo die Verantwortlichen den Prozess und das Agile überhaupt nicht kennen, gerade dort ist es so, dass meistens wirklich kurz vor Sprintbeginn und teilweise sogar im aktuellen Sprint geschätzt wird.

Wie lange ist ein Sprint bei euch?

Unterschiedlich. Aber man kann sagen, zwischen vier und fünf Wochen.

Ok. Das heißt, manchmal erfolgen Schätzungen unmittelbar vor der Umsetzung, und manchmal sind das ein, zwei, drei Monate vorher?

Richtig. Also wirklich komplett durchgemischt. Was man natürlich auch sagen muss: Wir haben ja nicht nur A1 als Kunden. Wir machen ja auch ÖBB-Projekte und viele, viele mehr. Und da schauen wir dann eigentlich, dass wir unsere internen Sprints an die Telekom anpassen, damit wir die gleichen Sprint-Zyklen, die gleichen Iterationen haben. Dass wir und einfach von der Organisation her leichter tun.

Gibt es bei euch immer eine Schätzung des Gesamtkostenrahmens und danach noch eine Schätzung von einzelnen Arbeitspaketen?

Also, wie gesagt. Das ist am Anfang die Grobkosten-Schätzung, auf Basis von der machen wir natürlich das Angebot und am Ende des Tages wäre es natürlich gut, wenn man dort ist. Oder drunter, natürlich.

Wie sind da die Erfahrungen? Geht sich das üblicher Weise aus, oder gibt es dann öfters eine Notwendigkeit, oder einen Druck, diesen Zeit- oder Kostenrahmen einzuhalten?

Den Druck gibt es natürlich, keine Frage. Wir haben aber in der Vergangenheit, die Erfahrung gemacht, dass es sich meistens irgendwie ausgleicht. Wir haben jetzt beispielsweise, die Applikation, die heißt MDF, da haben wir drei Projekte gehabt, da waren wir weit unter dem, was wir geschätzt haben. Das heißt, da sind wir relativ gut ausgestiegen. Wir hatten aber ein Projekt, da sind wir weit drüber. **Das heißt, das war ein Projekt, wo wir sehr viel mit Fremdsystemen zu tun hatten.** SAP ist eh relativ bekannt. SAP und noch einem Drittsystem. **Und da war die Schätzung komplett daneben,** weil man nicht abschätzen kann, wie die Implementierungsqualität von anderen Systemen ist. Und dort war sie extrem schlecht. Das heißt, wir hätten eigentlich schon im September, Oktober, November, irgendwann hätten wir fertig werden sollen; produktiv mit dem kompletten Paket sind wir jetzt noch nicht.

Und wirkt sich das dann auf die Schätzung der einzelnen Arbeitspakete aus?

Natürlich, wirkt sich aus. Also, dadurch dass wir jetzt die Erfahrung gewonnen haben, mit SAP ist es sehr, sehr schwer, immer wenn SAP als Fremdsystem irgendwie beteiligt ist bei zukünftigen Projekten, wird man einen gewissen Puffer noch dazugeben. Das ist genau der Puffer, den ich vorher bei den Stories angesprochen habe. **Das heißt, wir schätzen jetzt nicht nur rein die Implementierung, sondern auch noch Zusatzaufwendungen für Abstimmungen, für Fremdsysteme, die einen aufhalten, und, und, und.**

Und innerhalb der Projekte, wenn man jetzt sieht, es wird knapp, es geht sich wahrscheinlich nicht aus, und man hat dann noch Arbeitspakete, die zu schätzen sind, wirkt sich das dann aus?

Wirkt sich aus. In der Regel ist es so, dass wir dann quasi Überstunden machen müssen. Den geschätzt werden muss es trotzdem, damit man den nächsten Sprint erfolgreich starten kann. Ist natürlich für das Arbeitsklima nicht so angenehm.

Was wir eigentlich schon besprochen haben: Dass die Ausführlichkeit der Beschreibung der Arbeitspakete offensichtlich einen großen Einfluss hat. Also wenn es ausführlich beschrieben ist, dann sind Schätzungen üblicher Weise gut, wenn es knapp beschrieben ist, sind sie schlecht – kann man sagen?

Kann man sagen. Plus, da würde ich jetzt das dazu nehmen, was ich gerade gesagt habe: Wenn Fremdsysteme betroffen sind, muss man das wirklich betrachten, muss man das heranziehen. Da hat die Vergangenheit gezeigt, das kann wirklich das Doppelte von Schätzungen bedeuten. Weil man dann eben bei Tests dabei ist, weil man irgendwelche zusätzlichen Analysen auch für das andere System machen muss, weil da zu wenig Wissen vorhanden ist, und, und.

Wir haben auch schon angesprochen: Es gibt eine Top-Down-Schätzung, also eine Grobkosten-Schätzung am Anfang und dann die Bottom-Up-Schätzung für Arbeitspakete. Aber dann ist es eigentlich schon zu spät, dass man die Schätzung nochmal anpassen würde?

Natürlich.

Üblicher Weise gibt es dann offensichtlich schon ein Angebot ...

Ja.

... und da kommt man dann eh nicht mehr aus.

Ja.

Das kann man nicht verwenden, um die Schätzung noch irgendwie abzugleichen?

Sehr oft ist es halt so, dann muss man dem Ganzen auf den Grund gehen. Warum das jetzt so ist. Wenn man jetzt sagt ... es kommt sehr oft vor, da sind irgendwelche Dinge drinnen, die uns unnötig lange aufgehalten haben, dass Anforderungen nicht genau beschrieben waren, dass da jetzt viel, viel, viel mehr zu implementieren war, da kann man im Nachhinein auch natürlich mit den Kunden sprechen. Dass man sagt, das war ursprünglich in der Grobkostenschätzung nicht enthalten. Wie macht man das: Bestellt man das dazu, bestellt man das extra, lassen wir was anderes weg? Also da gibt es immer wieder Diskussionen. Das ist keine Seltenheit.

Dann bin ich mit meinen Fragen schon durch. Dankeschön.

INTERVIEW EXPERTE 6

Das Interview wurde am 05.04.2017 ab 09:30 in Pinkafeld durchgeführt. Experte 6 ist seit 2003 im Unternehmen tätig. Er arbeitet als Teamleiter und Software-Entwickler. Das Team von ihm bestand bis vor kurzem aus fünf Personen und wird künftig auf drei Personen reduziert werden.

Transkription

Beginnen würde ich gerne mit der Frage, wie lange bei dir im Team agile Methoden für die Projektentwicklung schon eingesetzt werden. Und vielleicht kannst du auch gleich ein bisschen was dazu sagen, wenn ihr das neu eingeführt habt, wie da die Erfahrungen bei der Einführung von agilen Methoden waren.

Grundsätzlich haben wir mit dem angefangen, als ich zu WebGIS gewechselt bin, eben zu den A1-Projekten. Wie lange ist das jetzt her?

Das müsste ungefähr 2011 gewesen sein.

Möglich ja. 2011, 2012, irgend so etwas, ja. Also 5 Jahre. Die Frage war, wie die Einführung war?

Wie die Erfahrungen bei der Einführung waren. Ob es das Probleme oder besondere Ereignisse gegeben hat. Oder ob das eine starke Umstellung zur bisherigen Arbeitsweise war?

Also nach meinem Gefühl nach war es keine wirkliche Umstellung. Du hast deine Arbeitspakete als Entwickler, sage ich mal, und die machst du. Der einzige Unterschied, der merklich ist, ist, dass der Druck steigt. Weil, wenn du ein großes Projekt hast und du arbeitest einfach im Wasserfall dahin, da hast du am Anfang, am Projektanfang keinen Druck. Wenn du jetzt nicht wirklich durchgeplant hast mit irgendwelchen Meilensteinen, machst du deine Sachen, zeigst es her, und am Ende bekommst du dann einen Stress. Ja. In dem Fall, wenn du jetzt die Sprints hast mit vier Wochen zum Beispiel, hast du deine Arbeitspakete und die solltest du fertig bekommen und das heißt, du hast eigentlich einen Dauer-Druck. Zum Durchatmen ist nicht wirklich Zeit. Das ist mein Gefühl, was jetzt „agil“ bedeutet. Das Problem jetzt in meinem Umfeld speziell bei A1 ist jetzt nicht wirklich die agile Software-Entwicklung in dem Sinn, sondern wie sich manche Leute das vorstellen, was „agil“ bedeutet. Dass es unterschiedliche Sichtweisen gibt. Wenn man dabei bleiben würde, was beim Sprint-Planning vereinbart wurde, diese Punkte, und es bei diesen Tätigkeiten bleiben würde, wäre das alles super. Aber bei diesen Tätigkeiten bleibt es nicht. Das Sprint-Planning, dieses Board ... der Sprint, bei der A1 jetzt, spiegelt nicht das wieder, was das Team im Sprint macht. Das ist nur ein Bruchteil davon. Das ist unter anderem ein Problem. Und das ist seit Beginn von A1-Projekten mit der agilen Umsetzung, ist das Gang und Gäbe. Man kann dort einfach nicht alles sichtbar machen, soll manches auch nicht sichtbar machen.

Und diese Tätigkeiten sind für die Projekte, an denen ihr gerade arbeitet, oder sind das andere Tätigkeiten, die da dazukommen?

Grundsätzlich ... Ein großer Punkt ist natürlich Support und Unterstützung und aktuelle Projekte, zukünftige Projekte, Vorbereitung vom nächsten Sprint. Ich sage jetzt einmal, wenn die Product-Owner, die A1-Seite relativ selbständig wäre und das alles machen könnte, wäre es wahrscheinlich kein Problem. Aber das ist eben nicht so, die tun Rosinen-picken und drehen sich das alles so, wie sie selber wollen. Und das führt dann unweigerlich zu Problemen.

Wenn wir jetzt von agiler Software-Entwicklung sprechen, dann ist das hauptsächlich Scrum?

Ja, genau.

Ok, dann vielleicht gleich einmal zu den Schätzungen: Mit welcher Methodik werden bei euch im Team Schätzungen durchgeführt?

Das Thema ist jetzt vor ein, zwei Wochen eskaliert. Wie geschätzt wird. Wir schätzen grundsätzlich ... Also wir erheben jetzt einmal für den Sprint die Ressourcen-Verfügbarkeit, und das ist für mich, für uns, Tage. Ich weiß, der ist zwei Wochen Urlaub, da fallen mal zwei Wochen weg. Und ich habe dann meine Summe von Story-Points, was im Grunde einem Tag entspricht. Die anderen haben dir das sicher auch schon so gesagt. Und beim Schätzen schätzen wir auch so. **Je nachdem, wie detailliert die Stories sind, kann man das detaillierter schätzen.** Aber so wie wir die Stories immer bekommen, und auch Zeitdruck beim Schätzen, wir schätzen nach Tagen: Ja, ich werde drei, vier Tage Arbeit haben damit, mit allem Drum und Dran. Wir lassen dann die diversen Tätigkeiten mit einfließen, so ferne wir gerade beim Schätzen daran denken. Ja. Eskaliert ist das deswegen, weil, wenn wir das jetzt schätzen, betrachten wir auch, sofern möglich, wer das machen wird. Wenn das ein Guru macht, der braucht vielleicht einen Tag. Der Michael zum Beispiel wenn es um irgendeine DB-Geschichte geht. Wenn das jetzt ein Junior macht, muss er ein bisschen mehr veranschlagen. Einerseits für das, was der mehr Zeit braucht, und auch für das, was er Unterstützung braucht von einem Guru. Das eine wäre jetzt ... hätten wir gesagt, ein Story Point. Jetzt habe ich aber nicht die Ressourcen von Gurus – sagen wir vier Story Points. Das ist jetzt natürlich passiert, weil mein Team dezimiert wird. Und da ist natürlich zurückgekommen das Unverständnis: ‚Die Schätzungen sind nicht nachvollziehbar‘. Im Grunde ist es nur um eine Story gegangen – eben statt eins vier. Ja, habe ich natürlich so begründet: ‚Ja, ich habe keine Zeit, Matthias ist nicht verfügbar, Know-How ist nicht da, ich weiß ad hoc nicht, was zu ändern ist, ich müsste auch nachschauen, das wird wer anders machen, und so weiter‘. ‚Ja, das kann es nicht sein. Wir haben uns doch verständigt, dass wir nach Story Points schätzen und nicht nach Zeit, und wir die Komplexität schätzen. Und das sollte Teamübergreifend immer der gleich Wert sein‘. Das heißt, wenn es eine riesige komplexe Story ist, 10. Wenn es was ganz was einfaches ist, dann – theoretisch – eins. Und da hat es Probleme gegeben und das ist immer noch nicht geklärt. Weil die Ansicht von diesem Typen ist, was er sieht, ist immer diese Komplexität, das heißt, das was immer gleich bewertet wird, sind die Stories, das was das Team leisten kann, diese Ressourcenverfügbarkeit, ist dann variabel. Das heißt mein Team, mit sage ich jetzt mal, nur zwei Juniors, schafft jetzt nicht die 40 Story Points, die zwei Gurus schaffen würden, sondern schaffen nur fünf, zum Beispiel. Ja, so viel zu dem Thema. Wir schätzen in Tagen, aber das wird nicht überall so gesehen und es gibt da noch Diskussionsbedarf und Konfliktpotential.

Und wie erfolgt die Schätzung: Macht das einer, machen das zwei, macht das das ganze Team, wird über den Inhalt von Stories gesprochen?

Wenn es Probleme gibt, bzw. sich die A1-Mitarbeiter einbringen wollen, oder das wirklich nach Scrum machen wollen, dann organisieren sie so ein Planning Poker. Da wird das mit dem ganzen ... ist natürlich auch ein zeitliches ... wenn man die Zeit hat. Ich persönlich finde es unnötig, jetzt für eine Story ein Planning Poker zu machen. Wenn man natürlich gleich ein ganzes Projekt, einen ganzen Sprint hat mit zehn Stories, die sind alle fertig – was eigentlich nie sein wird, das kommt immer alles so zierweise – für eine Story finde ich es unnötig. Aber wenn ich jetzt zehn Stories habe, hole ich gerne das Team zusammen, und man setzt sich zusammen mit dem Product Owner und schätzt das. Das haben wir oft gemacht. Ist super, weil gewisse Meinungen und ein gewisses Know-How zusammenkommt und man das alles gleich abklären kann, die Stories gleich ... man gleich Anmerkungen hinschreiben kann und sagt ok, so ist das. Schätzungen – da gibt es ein eigenes Tool im Internet, wo du schätzen kannst am Computer, das wird dann aufgedeckt. Wenn es gewisse Abweichungen gibt, der eine sagt 10, er andere sagt 1, dann muss man das diskutieren, wieso es da Unterschiede gibt. Ist super, wäre natürlich immer toll. Aber wie gesagt, zeitlich ein bisschen problematisch. Wenn wir einen Zeitdruck haben, wenn derjenige sich sicher ist bezüglich seiner Schätzung, schreibt er da gleich selber eine Zahl hinein, sagt das ist so viel. Bzw., wenn ich das als Arbeitspaket habe, dass ich da eine Schätzung auftreiben soll, und ich kann's nicht, ich habe keine Zeit, mach du das, sage mir eine Zahl. **Und ich sage dann, ja, das ist doch ein bisschen wenig, ein bisschen was aufschlagen** für Martin Galanda oder Fink, was die dann brauchen an Betreuung. Kurz gesagt: wischi, waschi.

Macht es deiner Erfahrung nach einen Unterschied, ob das gesamte Team schätzt oder ob einzelne schätzen: Wie zutreffend, wie genau die Schätzung dann ist?

[DENKT LÄNGER NACH] Ich sage einmal: Wenn das ganze Team schätzt, bzw. mit Product Owner, wenn er inkludiert ist, ist die Tendenz Richtung weniger. Also die Schätzungen werden niedriger. Erstens weil der Product Owner dabei ist, auch weil er sagt: ‚Nein, das kann nicht so viel sein‘. Und dann redet man nochmal drüber. Es wird dann tendenziell weniger, weil gewisse Sachen ausgeräumt werden, Unklarheiten. Und gewisse Ausschluss-Kriterien definiert werden: Das brauchen wir nicht und das brauchen wir nicht, ... Was jetzt einer alleine ... ‚Was soll ich mit dem einen Satz, ich muss jetzt sagen, ok, das kann alles sein. Alles und nichts.‘ Und diese Unsicherheit wird einkalkuliert. **Also, sie werden treffsicherer, wenn mehrere Leute schätzen.** Das definitiv. Wobei, wenn jemand alleine schätzt, ist nach oben hin keine Grenze, sage ich einmal. Und das geht sich meistens aus. Also da gibt es selten Fälle, dass man sagt, ok, zu wenig geschätzt. Das ist meistens so, dass man zu viel schätzt. Und das geht sich jetzt für uns aus. Für den Kunden ist das natürlich eher schlecht. Aber durch das, dass wir da eh flexibel und gutmütig sind, sagen wir: ‚Ja, das war alles zu viel. Es ist noch genug da‘. Das kommt dann auf einen Pool – wir streifen das ja nicht ein. Also insofern – er bekommt ja trotzdem seine Leistung des Entwicklers.

Führt ihr ein laufendes Controlling der Schätzungen durch? Das heißt einen Vergleich zwischen der Schätzung und dem tatsächlichen Aufwand.

Nein. Also ich nicht. Ich weiß nicht ob es der Manuel macht (ANMERKUNG: DER TEAMLEITER UND PROJEKTMANAGER EINES ANDEREN TEAMS). Mir fehlen noch die Werkzeuge dazu. Ich habe da auch schon Ideen gehabt für das LZE (ANMERKUNG: DIE LEISTUNGSZEIT-ERFASSUNG, EIN INTERNES TOOL FÜR DAS ERFASSEN DER ARBEITSZEITEN). Wenn ich mein LZE aufmache, dass ich auch gleich sehe mit was das Team gerade beschäftigt ist, wie – so balkenmäßig wenn reserviert sind so 50 Stunden für diese Story,

und wir sind schon bei 90% oder ich bin schon bei 120. **Also nicht so ein nachträgliches Controlling, sondern ein laufendes Controlling. Das wäre lässig.** Weil da kann man relativ schnell schreien, hallo, du bist schon ... Weil der Grundtenor ist bei der A1, wenn ich bei 50% bin von dem, was die Story geschätzt worden ist am Anfang, dann muss man schon schauen. Weil der Entwickler sollte mit 50% eigentlich auskommen. Weil alles andere ist der Overhead. Das heißt, wenn der die ganzen 50 Stunden braucht, und ich habe nicht gerade eine Story, die übermäßig zu viel geschätzt worden ist, dann kann es eng werden. Aber es gibt kein laufendes Controlling im Moment.

Hast du Erfahrungen, was sonst die Genauigkeit von Schätzungen beeinflusst?

Ja, das was ich gestern gesagt habe: **Umso weniger Tätigkeiten berücksichtigt werden müssen, bei der Schätzung, umso genauer wird es.** Nach meiner Ansicht nach. Das heißt, wenn der Entwickler, der wirklich nur seine Tätigkeiten schätzen muss wie Entwicklung, Test, Story weiter schieben, fertig. Und nicht das jetzt mitschätzen muss vom Deployment-Management und Scrum-Management und Projekt-Management, und das ganze Zeug, sind sie jetzt aus Tätigkeits-Sicht genauer. Wäre wahrscheinlich auch zielführender, dass man das in diese Richtung macht. Wäre auch transparenter für den Auftraggeber, wäre auch klar für den Auftraggeber, was da inkludiert ist. Jetzt vom Sprint-Planning her. Würde auch zu weniger Komplikationen und Diskussionen führen. Den Rest müsste man dann halt bei der Beauftragung aufschlagen, eben prozentuell in irgendeiner Form. Oder als eigener Punkt oder sonst wo ersichtlich. Natürlich muss man dann auch, sage ich einmal, auch wirklich ein Controlling machen. Weil dann musst du wirklich punktgenau landen oder drunter sein. Da darf ich dann nicht 40 Stunden für eine drei Story Point Story brauchen. Weil dann ist wirklich knapp kalkuliert.

Ist es bei euch unterschiedlich, wann die Schätzungen durchgeführt werden? Ist das manchmal sehr weit von der Umsetzung weg und manchmal sehr knapp davor? Und macht das dann einen Unterschied, ob das dann genauer ist oder weniger genau ist?

[DENKT LÄNGER NACH] Sagen wir mal so: Wenn ein Projekt geschätzt wird für ein Angebot, das ist natürlich wenn man so etwas ausfüllt (ANMERKUNG: ZEIGT EIN EXCEL-SHEET MIT EINER VORLAGE FÜR SCHÄTZUNGEN, IN DEM ZU BERÜCKSICHTIGENDE PUNKTE RELATIV DETAILLIERT VORGEGEBEN SIND), sind mehrere Leute beteiligt, nicht nur einer. Dann macht man sich auch schon Gedanken drüber. Das ist weit vor der Beauftragung, weit vor der Umsetzung. Das Wissen geht verloren. Das, was man sich da angeeignet hat, was man da besprochen hat, das verschwindet. Das heißt, das was du da schon an Zeitvorsprung herausgeholt hast, das verlierst du. Das heißt, umso weiter das davor ist, meiner Meinung nach, umso schlechter ist das. Weil das einfach verloren geht. Das gleiche gilt jetzt für das Planning Poker oder Sprint Planning. Wenn ich das für den nächsten Sprint am Anfang des vorhergehenden Sprints mache, da vergehen wieder vier, fünf Wochen. Das ist weg. Wir machen es meistens, aus Zeitdruck, kurz vorher. Ein, Zwei Tage vorher oder eben im Laufe des Sprints. Aber wenn da etwas besprochen wird, und das wird nicht festgehalten in den Stories, dann geht das wieder verloren. Also das hat jetzt nichts per se mit der Genauigkeit der Schätzung zu tun, sondern einfach verlorenes Wissen, das ich später wieder aufholen muss. Ja, doppelte und dreifache Arbeit unter Umständen.

Also, ihr habt üblicher Weise Kundenprojekte, die ihr umsetzt. Das heißt, da gibt es vorab eine Schätzung und ein Angebot.

Ja.

Und danach wahrscheinlich pro Story nochmal Schätzungen?

Ja. Genau. Meistens sind es irgendwelche Einzeiler. Das kommt darauf an, wie viele Gedanken sich der A1-Mitarbeiter oder der Zuständige dort schon gemacht hat. Wenn die Anforderungen von unserem Bereich kommen, mit dem wir Großteils beschäftigt sind, sind es meistens Einzeiler. Wenn es von anderen Systemen oder Produkten kommt, wo es ein bisschen besser läuft, also strukturierte, dann sind sie natürlich strukturierter. Wenn wir für solche detaillierte Sachen eine Schätzung machen, ist das natürlich super, da kann man das wirklich genau machen. Wenn man es für solche Einzeiler macht, werden die Schätzungen extrem hoch. Wie war die Frage nochmal?

Ob es eine Schätzung für das Gesamte gibt, und dann die Schätzungen für die einzelnen Stories? Und die Folgefrage wäre dann, ob es einen Abgleich gibt, ob es überprüft wird, ob diese Schätzung im Groben und die Schätzung im Detail, ob das zusammen passt.

Also grundsätzlich, die Schätzung für das Angebot machen zu, ich sage einmal, 70% nicht die Leute, die dann wirklich im Sprint Planning beteiligt sind und dort die Stories schätzen. Die wissen auch meistens nicht, oder nicht mehr, dass es so eine Schätzung gibt. Vor allem auch deswegen, weil diese Schätzungen immer sehr grob sind, meistens noch nicht die Stories beinhalten, meistens eben nur irgendein Word-Dokument oder ein Mail mit irgendwelchen Punkten. Und das schwer zuordenbar ist, welche Story das jetzt war. Also ich kann nicht, wenn da jetzt 5 Story Points steht, kann ich nicht rückverfolgen, jetzt diese Story, das war jetzt da fünf. Also, meistens unterscheidet sich das. Das machen nicht die gleichen Personen und Großteils kommt auch nicht – es ist nicht zuordenbar. Wenn es zuordenbar ist, dann kommt es so ziemlich aufs Gleiche. Ich kann dir das nachher eh zeigen. Ja. Aber es wird kein ... ich tu's nicht abgleichen, ich habe die Zeit nicht dazu. Projektmanager, Projektleiter ... ich glaube, die Jelena schaut sich das schon so ziemlich an, also auf Basis der Punkte im Statusbericht. Also je nachdem, wie granular du den Statusbericht machst, musst du es anschauen. Wenn du dort nur hineinschreibst „Pool“, so viel habe ich noch verfügbar, fertig. Aber wenn du es nach Epics machst – wir haben ja die Epics und die Stories – ja. Also ich mache es nicht. Wenn es der Projektleiter macht, ist es gut. Wäre mir aber noch nicht aufgefallen, dass wir diesbezüglich etwas besprochen hätten.

Habt ihr viele unterschiedliche Arten von Projekten? Das heißt, sind das unterschiedliche Produkte, manchmal neuartige Technologien, unterschiedliche Domänen? Oder sind die Projekte eher gleichartig?

Die A1-Domäne ist riesig. Somit würde ich sagen: unterschiedliche Domänen. Also diese A1-Domäne hat Sub-Domänen, sage ich einmal. Sie ist riesig. Keiner kennt alles. Jedes Projekt ist für sich, eine eigene Schnittstelle, eine eigene Funktion, unterschiedliche Daten, Datenquellen. Also es ist eigentlich immer unterschiedlich. Du lernst immer was Neues, du musst immer was Neues lernen.

Aber von dem her ist es doch wieder ähnlich, weil es immer wieder was Neues ist. Also es ist nicht so, es gibt Projekte, da kennt man schon alles, es gibt Projekte da ist alles neu, sondern es ist insofern ähnlich, dass immer wieder was dazugelernt werden muss?

Ich sage mal, in den meisten Fällen. Natürlich schauen wir, auf Grund – aus Effizienzgründen, dass jetzt wenn wieder so ein Projekt kommt mit dem gleichen Tätigkeitsbereich – das ist aber überall so – na, der hat das schon gemacht, der soll das wieder machen. Ja. Aber auf Grund von Team-Fluktuationen und ... der arbeitet jetzt an dem Projekt und aus terminlichen Gründen und so, musst du es eben wem Anders geben. Kannst es nicht den machen lassen ... Also, es ist selten dasselbe. Also da muss es wirklich grad zeitlich passen, also es macht immer derjenige dasselbe Themengebiet ... Also es wechselt, und es

wechseln auch die Produkte. Wir haben unzählige Produkte. Wenn ich denke an Unified Client von der A1, dann Nero, WebGIS Professional, GIS-Viewer, Datenbank-Middleware. Das macht einen Unterschied, ob ich im asp.NET, im GIS-Viewer, irgendwelche Zeilen, eine Textbox einbaue, oder ob ich in der Datenbank irgendeine Prozedur machen muss. Ich muss, Hausnummer, 30 Technologien vertraut sein im A1 Umfeld. Und das kommt noch erschwerend hinzu. Also ja ...

Macht es einen Unterschied für die Schätzung, in welchem Projekt oder in welchem Produkt man sich bewegt?

Ja. Grundsätzlich, welches Projekt, welche A1-Leute sind beteiligt. Da müssen wir Faktoren einfließen lassen. Der eine sagt geht, da läuft alles super bei dem. Der andere, der macht nichts. Da kann ich keine Unterstützung oder kein Ergebnis erwarten, von dem. Das muss ich einkalkulieren. Ich muss Fremdsystem einkalkulieren. Wenn ich weiß, ich habe mit 3, 4 Fremdsystemen zu tun, speziell dieses eine, dann muss ich einen Fremdsystem-Faktor aufschlagen. Weil das Ganze, alles mit einberechnen, dass ich da eben nicht ins Schlittern komme. Das macht einfach die Erfahrung. Es ist nicht nur das Themengebiet, es ist auch das Projektumfeld mit einzuschätzen, also mit einzukalkulieren. Sprich, Personen und Systeme.

Und wenn das einkalkuliert ist: Kann man dann sagen, die Schätzungen sind überall ähnlich genau?

Ja. Ich sage jetzt einmal, wenn jetzt einer sagt: ‚Ok, diese Schnittstelle ist zu machen‘, der schätzt da 1, 2 Story Points, und das ist SAP – ein Problem – da fragt man schon: ‚Hörst, weißt eh, SAP, weißt‘ eh, was da los ist?‘ Also, wenn da wer drüber liest über die Schätzung, da wird nachgefragt. Also das sieht man. Überhaupt, wenn solche Fremdsysteme dabei sind. Da macht es jetzt nicht einer, der mit dem noch nicht vertraut ist, sondern da sind mehrere Leute dabei. Ich hätte noch nichts davon gehört, dass man da jetzt viel zu wenig geschätzt hätte. Wenn ein Projekt aus den Fugen gerät, sind es mehrere Faktoren. Also jetzt nicht, dass man eine Story unterschätzt hätte. Da hat man dann wirklich das ganze Projekt unterschätzt. Das ist meistens bei den Projekten, die ultra-agil gelebt werden. Wo im Vorfeld nicht klar ist, was in dem Projekt gemacht werden soll. Was Daumen mal Pi Schätzungen und Angebote sind. Und von A1-Seite agil wirklich agil gelebt wird. Ich habe da jetzt auch mal mit dem Martin geredet. Das sind Personen, die schon gedanklich 5 Jahre in der Zukunft sind, aber das jetzt machen und – wie soll ich sagen – sich einfach Schritt für Schritt an dieses 5-Jahres-Ziel da herannähern. Und das jetzt nicht so wirklich im Pakete schnüren können. Also die arbeiten dahin, bis das Geld fort ist, und Anforderung für Anforderung.

Da hat man dann das Problem, dass man zu Beginn des Projekts keine klaren Anforderungen hat?

Ja. Der stellt sich etwas vor, wo er hin will. **Die, die das schätzen, stellen sich irgendwas vor; man unterschätzt das, weil der sich nicht reinversetzen kann in den anderen. Das vielleicht anders verstanden hat.** Da kommt noch die Hinterlistigkeit dazu, weil sie dir was unterschieben wollen. So Zusatzanforderungen und so weiter. Und dann gerät das alles irgendwie außer Ruder, das Ganze. Aber jetzt so, ... dort gäbe es Probleme. Ich sage jetzt einmal, dass das, ... also eigentlich kann es nur bei agilen Projekten so sein. Weil bei einem Wasserfall-Projekt, wenn man jetzt an die Methodik hergeht. Da muss es ein Konzept geben, da muss das alles schon alles vordefiniert sein. Und da ist eine Schätzung, und genau das machen wir. Und dort gibt es die Diskussionen nicht. Da gibt es diese Flexibilität nicht.

Aber wie wir aus Erfahrung gesehen haben, damals, als wir Wasserfall gemacht haben, waren die Projekt auch nicht immer punktgenau und rosig. Das macht keinen Unterschied, meiner Meinung nach ...

Ok, passt. Dann sage ich soweit einmal: Danke schön.

ANHANG C - Das agile Manifest

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

| | | |
|-------------------|----------------|------------------|
| Kent Beck | James Grenning | Robert C. Martin |
| Mike Beedle | Jim Highsmith | Steve Mellor |
| Arie van Bennekum | Andrew Hunt | Ken Schwaber |
| Alistair Cockburn | Ron Jeffries | Jeff Sutherland |
| Ward Cunningham | Jon Kern | Dave Thomas |
| Martin Fowler | Brian Marick | |

ABKÜRZUNGSVERZEICHNIS

| | |
|--------|---|
| BRE | Balanced Relative Error |
| DB | Datenbank |
| GBL | Geschäftsbereichsleitung |
| ISBSG | International Software Benchmarking Standards Group |
| IT | Informationstechnik |
| LZE | Leistungszeiterfassung |
| Max | Maximum |
| Min | Minimum |
| MMRE | Mean Magnitude of Relative Error |
| MRE | Magnitude of Relative Error |
| NG | Next Generation |
| PDR | Project Delivery Rate |
| StdDev | Standard Deviation (Standardabweichung) |
| TUP | Technologie- und Plattformteam |
| XP | Extreme Programming |

GLOSSAR

Backlog

Das agile Produkt-Backlog in Scrum ist eine priorisierte Liste von Merkmalen, die eine kurze Beschreibung der gewünschten Funktionalität für ein Produkt enthält. (Scrum Product Backlog, 2016)..... 1

Function Points

Die funktionale Größe von Software repräsentiert die Größe von funktionellen Benutzeranforderungen. Funktionale Größe wird mittels Function Points ausgedrückt. 20

ISBSG

Die International Software Benchmarking Standards Group ist eine Non-Profit Organisation, die 1997 gegründet wurde. Die Organisation hat sich zum Ziel gesetzt, Unternehmen bei der Planung und dem Management von Software-Projekten zu unterstützen. (International Software Benchmarking Standards Group, 2016) 19

Product Backlog Item

Ein Product Backlog Item ist ein Eintrag im Product Backlog - ein Teil der Arbeit, der klein genug ist, um innerhalb eines Sprints (einer Iteration) umgesetzt zu werden. 25

Project Delivery Rate

Die Project Delivery Rate wird in Stunden pro Function Point ausgedrückt, wobei "Stunden" der Aufwand für die Umsetzung ist. 20

Story Points

Story Points sind ein relatives Maß für die Größe einer User Story. Vielfach werden Story Points aus folgender Menge vergeben

1, 2, 3, 5, 8, 13, 20, 40, 100. Dies ist eine angepasste Fibonacci-Folge, die unter anderem ausdrückt, dass die Genauigkeit einer Schätzung abnimmt, je größer die zu schätzende Story ist. 24

ABBILDUNGSVERZEICHNIS

| | |
|---|----|
| Abbildung 1: Der Cone of Uncertainty bezogen auf Projektmeilensteine, nach (McConnell, 2006) | 8 |
| Abbildung 2: Implementierungsschritte für kleinste Softwareprojekte. Nach (Royce, 1970) | 11 |
| Abbildung 3: Das Vorgehensmodell nach Royce (Wasserfallmodell), wobei Iterationen auf aufeinanderfolgende Schritten beschränkt sein sollen (symbolisiert durch die Pfeile zwischen den Schritten). Nach (Royce, 1970)..... | 11 |
| Abbildung 4: Die Phasen und Workflows (Disciplines) im Verlauf des Unified Process Lifecycles. Nach: (Rational Software Corporation, 2001)..... | 12 |
| Abbildung 5: Arten von Anforderungen in einem Software-Entwicklungs-Projekt; nach: (Hill, 2011) | 21 |
| Abbildung 6: Die Anzahl der Kommunikationswege steigt nicht linear, sondern quadratisch proportional mit der Anzahl der Personen im Team. Übernommen und angepasst aus: (McConnell, 2006)..... | 22 |
| Abbildung 7: Bei der Aufwandsschätzung in agilen Projekten wird aus der geschätzten Größe der Summe der Features die Umsetzungsdauer ermittelt; aus: (Cohn, Agile Estimating and Planning, 2006). | 24 |
| Abbildung 8: Berechnen und verwenden eines Bereichs für die Velocity, nach (Rubin, 2012) | 28 |
| Abbildung 9: Anzahl der Projekte, in denen die Befragten mitgearbeitet haben. Quelle: eigene Erhebung mittels Google Formulare | 62 |
| Abbildung 10: Funktion der Befragten. Quelle: eigene Erhebung mittels Google Formulare | 62 |
| Abbildung 11: Eingesetzte Methode für Schätzungen. Quelle: eigene Erhebung mittels Google Formulare | 63 |
| Abbildung 12: Metrik, mit der die Größe von Backlog-Items gemessen wird. Quelle: eigene Erhebung mittels Google Formulare | 64 |
| Abbildung 13: Art der Evaluierung der Genauigkeit von Schätzungen pro Backlog-Item. Quelle: eigene Erhebung mittels Google Formulare | 64 |
| Abbildung 14: Art der Evaluierung der Genauigkeit von Schätzungen für das Gesamtprojekt. Quelle: eigene Erhebung mittels Google Formulare..... | 65 |
| Abbildung 15: Linke Spalte: die Bewertung aller Teilnehmer, wie stark die genannten Erfolgsfaktoren die Genauigkeit von Schätzungen positiv beeinflussen. Rechte Spalte: die gleiche Bewertung, jedoch nur unter Berücksichtigung der Teilnehmer, die die jeweilige Methode einsetzen. Quelle: eigene Erhebung | 68 |
| Abbildung 16: Bewertung aller Teilnehmer, wie stark widersprüchliche Ziele die Genauigkeit von Schätzungen negativ beeinflussen..... | 72 |
| Abbildung 17: Bewertung aller Teilnehmer, wie stark zusätzliche (irrelevante) Informationen die Genauigkeit von Schätzungen negativ beeinflussen. | 73 |
| Abbildung 18: Die Auswertung von Projekt 13 mit einem Gesamtumfang von 366 Stunden und 3 Teammitgliedern..... | 77 |
| Abbildung 19: Die Auswertung von Projekt 7 mit einem Gesamtumfang von 3619 Stunden und 8 Teammitgliedern..... | 78 |
| Abbildung 20: Darstellung der Differenzen in einem der Projekte, in denen Checklisten eingesetzt werden (Projekt 15) Quelle: eigene Erhebung. | 79 |

Abbildung 21: Darstellung der Differenzen in einem der Projekte, in denen Checklisten eingesetzt werden (Projekt 16) Quelle: eigene Erhebung. 79

Abbildung 22: Die Auswertung von Projekt 3, in dem mit Story Points geschätzt wurde. Quelle: eigene Erhebung. 81

Abbildung 23: Auswertung von Projekt 4, in dem mit tatsächlichen Stunden geschätzt wurde. Quelle: eigene Erhebung 81

TABELLENVERZEICHNIS

| | |
|--|----|
| Tabelle 1: Vergleich von Aspekten in der traditionellen und agilen Software-Entwicklung; Angepasst und gekürzt aus (Moniruzzaman & Hossain, 2013) | 14 |
| Tabelle 2: Beispiele für Objekte, die gezählt werden können, um Schätzungen abzuleiten, aus: (McConnell, 2006) | 19 |
| Tabelle 3: Auswertung der Project Delivery Rate, kategorisiert nach Teamgröße, aus: (Hill, 2011) | 23 |
| Tabelle 4: Änderungen in Anforderungen und der Umgang damit in der Projektplanung, aus (Basten, 2012)..... | 33 |
| Tabelle 5: Beobachtungen zu Expertenschätzungen und der Nutzen von Checklisten, basierend auf (Jorgensen, 2004) | 38 |
| Tabelle 6: Übersicht über die beschriebenen Einflussfaktoren auf die Genauigkeit von Schätzungen. Die letzte Spalte gibt an, ob die genannten Kriterien in dieser Arbeit weiter untersucht werden. | 45 |
| Tabelle 7: Teams, Nummer des Interviews des Projektmanagers sowie Teamleiter im Fachbereich Vermessung | 47 |
| Tabelle 8: Teams, Nummer des Interviews des Experten sowie Teamleiter im Fachbereich Geoinformation | 47 |
| Tabelle 9: Die Fragen, welche als Leitfaden für das Interview der befragten Experten dienten. | 50 |
| Tabelle 10: Übersicht über die Fundstellen, die zugeordneten Paraphrasen, die Generalisierung und die Art der Auswirkung auf die Genauigkeit von Schätzungen. | 56 |
| Tabelle 11: Von den Experten genannte Faktoren, welche die Genauigkeit von Schätzungen positiv beeinflussen. | 57 |
| Tabelle 12: Von den Experten genannte Faktoren, welche die Genauigkeit von Schätzungen negativ beeinflussen. | 58 |
| Tabelle 13: Die abgefragten Erfolgsfaktoren, sortiert nach der Häufigkeit, wie oft sie eingesetzt werden. Quelle: Eigene Erhebung | 66 |
| Tabelle 14: Die abgefragten Erfolgsfaktoren, sortiert nach der Häufigkeit, der Einstufung 1 und 2. Quelle: Eigene Erhebung..... | 67 |
| Tabelle 15: Übersicht über die ausgewerteten Projekte, Anzahl der Teammitglieder, Gesamtaufwand und Anzahl der ausgewerteten Backlog-Items, sortiert nach Projektgröße. Quelle: eigene Erhebung | 74 |

LITERATURVERZEICHNIS

- Angermeier, G. (11. 12 2009). *Magisches Dreieck*. Von ProjektMagazin Das Fachportal für Projektmanagement: <https://www.projektmagazin.de/glossarterm/magisches-dreieck> abgerufen
- Awad, M. (2005). *A comparison between Agile and Traditional SW development methodologies*. University of Western Australia.
- Basten, D. (2012). Es geht genauer! Aufwandsschätzung in Software-Entwicklungsprojekten. *Projekt Magazin*(11).
- Beck, K. (2000). *Extreme Programming explained*. Boston: Addison-Wesley.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). *Manifesto for Agile Software Development*. Von Manifesto for Agile Software Development: <http://www.agilemanifesto.org/iso/de/> abgerufen
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). *Principles behind the Agile Manifesto*. Von Manifesto for Agile Software Development: <http://www.agilemanifesto.org/principles.html> abgerufen
- Bergsmann, J. (2014). *Requirements Engineering für die agile Softwareentwicklung*. Heidelberg: dpunkt.verlag.
- Boehm, B. W. (1981). *Software Engineering Economics*. Englewood Cliffs: Prentice-Hall.
- Boehm, B. W. (Mai 1988). A Spiral Model of Software Development and Enhancement. *IEEE Computer*, S. 61-72.
- Buglione, L., & Abran, A. (2007). Improving Estimations in Agile Projects: Issues and Avenues. In T. Dekkers (Hrsg.), *Proceedings of SMEF 2007*, (S. 265-274). Rom.
- Choudhari, J., & Suman, U. (2012). Story Points Based Effort Estimation Model for Software Maintenance. *Procedia Technology*(4), S. 761-765.
- Cockburn, A. (2004). *Crystal Clear*. Amsterdam: Addison-Wesley.
- Coelho, E., & Basu, A. (August 2012). Effort Estimation in Agile Software Development using Story Points. *International Journal of Applied Information Systems*.
- Cohn, M. (2004). *User Stories applied for Agile Software Development* . Prentice Hall: Addison Wesley.
- Cohn, M. (2006). *Agile Estimating and Planning*. Upper Saddle River: Pearson Education.

- Cohn, M. (23. August 2016). *What are Story Points*. Abgerufen am 10. September 2016 von Mike Cohn's Blog at Mountain Goat Software: <http://www.mountaingoatsoftware.com/blog/what-are-story-points>
- Fowler, M., J.K., W., Beaton, M., Vishwanath, A., Bersano, J., Carroll, I., . . . Li, H. (2013). *How do you estimate on an Agile project?* (T. Works, Hrsg.)
- Gandomani, T. J., Wei, K. T., & Binhamid, A. K. (November 2014). A Case Study Research on Software Cost Estimation Using Experts' Estimates, Wideband Delphi, and Planning Poker Technique. *International Journal of Software Engineering and Its Applications*, S. 173-182.
- Gloski, D. (2013). Gesamtaufwände in agilen Projekten schätzen - darauf müssen Sie achten. *Projekt Magazin*(21).
- Grenning, J. (2002). Planning Poker or How to avoid analysis paralysis while release planning. *Hawthorn Woods: Renaissance Software Consulting*.
- Hill, P. R. (2011). *Practical Software Project Estimation: A Toolkit for Estimating Software Development and Duration*. Mcgraw-Hill Education.
- Hohman, M. M. (2005). Estimating in actual time. *Agile Development Conference (ADC'05), 2005* (S. 132-138). Denver, Colorado: IEEE Computer Society.
- International Requirements Engineering Board (IREB). (1. März 2015). IREB Certified Professional for Requirements Engineering - Foundation Level, Lehrplan.
- International Software Benchmarking Standards Group. (2016). *ISBSG - About*. Abgerufen am 20. August 2016 von ISBSG: <http://isbsg.org/about-isbsg/>
- Jorgensen, M. (2004). A review of studies on expert estimation of software development effort. *The Journal of Systems and Software*, S. 37-60.
- Keaveney, S., & Conboy, K. (2006). Cost estimation in agile development projects. *ECIS 2006 Proceedings, Paper 169*.
- Khatibi, E., & Ibrahim, R. (August 2012). Efficient Indicators to Evaluate the Status of Software Development Effort Estimation inside the Organizations. *International Journal of Managing Information Technology*, S. 23-31.
- Kriegisch, A. (kein Datum). *Scrum-Glossar*. Abgerufen am 23. August 2016 von Scrum-Master.de: http://scrum-master.de/Scrum-Glossar/Story_Point
- Kuhrmann, M. (21. Februar 2013). *Agile Vorgehensmodelle*. Abgerufen am 12. August 2016 von Enzyklopädie der Wirtschaftsinformatik Online-Lexikon: <http://www.enzyklopaedie-der->

wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Vorgehensmodell/Agile-Vorgehensmodelle/index.html

Leffingwell, D. (2011). *Agile Software Requirements*. Boston, MA: Addison Wesley Professional.

Leung, H., & Fan, Z. (2002). Software cost estimation. *Handbook of Software Engineering*, S. 1-14.

Mayring, P. (2015). *Qualitative Inhaltsanalyse, Grundlagen und Techniken*. Weinheim: Beltz Verlag.

McConnell, S. (2006). *Software Estimation - Demystifying the Black Art*. Redmond, Washington: Microsoft Press.

Moniruzzaman, A., & Hossain, S. A. (2013). Comparative Study on Agile Software Development Methodologies. *Global Journal of Computer Science and Technology Software & Data Engineering*.

Opelt, A., Gloger, B., Pfarl, W., & Mittermayr, R. (2013). *Agile Contracts - Creating and Managing Successful Projects with Scrum*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Osman, H. H., & Musa, M. E. (März 2016). A Survey of Agile Software Estimation Methods. *International Journal of Computer Science and Telecommunications*, S. 38-42.

Pichler, R. (2010). *Agile Product Management with Scrum*. Boston, MA: Addison Wesley.

Rational Software Corporation. (2001). *Development Case: Lifecycle Model and Disciplines*. Abgerufen am 10. August 2016 von The Wylie College Software Process Web Site: http://sce.uhcl.edu/helm/rup_school_example/wcsoftwareprocessweb/devcase/overview/dc_life.htm

Royce, W. W. (1970). Managing the development of large software systems. *Proceedings of IEEE WESCON*, (S. 328-338). Los Angeles.

Rubin, K. S. (2012). *Essential Scrum A Practical Guide to the most popular agile process*. Upper Saddle River: Addison-Wesley.

Schwaber, K., & Sutherland, J. (2014). *Der Scrum Guide*. Scrum.org und Scrum Inc.

Scrum Product Backlog. (05. Dezember 2016). Von Mountain Goat Software: <https://www.mountaingoatsoftware.com/agile/scrum/product-backlog> abgerufen

Software development effort estimation. (25. August 2016). Abgerufen am 02. September 2016 von Wikipedia: https://en.wikipedia.org/w/index.php?title=Software_development_effort_estimation&oldid=736164768

Steyer, M. (2010). *Agile Muster und Methoden*. Frankfurt am Main: entwickler.press.

Trendowicz, A., & Ross, J. (2014). *Software Project Effort Estimation Foundations and Best Practice Guidelines for Success*. Springer.

Trudel, S. (2007). Measurement for improving accuracy of estimates: the case study of a small software organisation. *Software Measurement European Forum* (S. 79-91). Rome: Ton Dekkers, Shell Information Technology International B.V.

Usman, M., Mendes, E., Weidt, F., & Britto, R. (2014). Effort estimation in agile software development: a systematic literature review. *Proceedings of the 10th International Conference on Predictive Models in Software Engineering* (S. 82-91). ACM.

VersionOne Inc. (2016). *10th Annual State of Agile Report*. Atlanta.

Wieggers, K. E. (Februar 2000). Stop Promising Miracles. *Software Development*.

Williams, L., & Cockburn, A. (Juni 2003). Agile Software Development: It's about Feedback and Change. *IEEE Computer*, S. 39-43.