MASTER’S THESIS

ADAPTION OF A WEB ANALYTICS FRAMEWORK FOR COMMERCIAL SINGLE PAGE APPLICATIONS

carried out at

CAMPUS 02 GRAZ
FACHHOCHSCHULE DER WIRTSCHAFT

Course Programme
Information Technologies & Business Informatics

By: Peter Göri
PID: 1610320018

Haselsdorf, 12th December 2017 ....

Peter Göri

STATUTORY DECLARATION

I hereby declare that I have authored this thesis independently, that I have not used other than the declared sources and resources and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....

Peter Göri

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.

....

Peter Göri

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor DI Selver Softic, BSc for his support during the past year. He supported me with his direct and uncomplicated approach and gave me feedback whenever I needed it.

Numerous hours of work have been invested into this thesis and this would not have been possible if my employer, Reval Austria GmbH, had not been so understanding and flexible. I appreciate the ongoing support during my studies. Thank you Patrick, Wolfi, Martin, Gaurav, Chirag and Christoph for being very considerate and always having my back.

Sabrina was a big help in this endeavor. She provided quick and precise quality assurance and made sure I could meet my deadline.

I would also like to express my gratitude towards my family. I appreciate all the support throughout my life and for always being there for me.

Lilly, my dear, you were continuously supporting me during this work. You kept motivating me time after time and did not get tired of it. I appreciate all the times you cheered me up. I'm looking forward to a wonderful life with you.

Peter Göri

Haselsdorf, 12th December 2017

ABSTRACT

Web analytics has been used to track visitor behavior on web sites since the early 1990s with the goal to understand the visitors and to constantly improve the web site. Although it is also possible to track users of single page applications, a special form of web application that behaves like a desktop application, there are some limitations. The aim of this work is to identify gaps in web analytics solutions regarding single page applications. Subsequently the impact of resolving such a gap on the intended long-term usage of a given web analytics system, the so-called *Continuance Intention*, is evaluated. Initially, an integration of an open-source web analytics framework with a commercial single page application is performed. Based on the gained experience and by conducting a literature research, a list of 13 gaps including potential solutions is identified. A group of experts is ranking them according to their impact on the usefulness of the web analytics solution. The resulting top item is the custom metrics feature, which allows to track application-specific metrics by configuring them generically in the web analytics system. It is subsequently developed and evaluated by means of an online survey using an adapted version of the *Expectation-Confirmation Model*. A linear regression model is based on the two resulting latent factors *Perceived Usefulness* and *Continuance Intention*. The statistical analysis of the model shows that a specific improvement of a web analytics solution with the goal to enhance the usefulness of this information system increases the intention to keep using this software in the long-term.

KURZFASSUNG

Web-Analytics Software wird bereits seit den frühen 1990er Jahren eingesetzt, um das Besucherverhalten auf Websites zu verfolgen. Dies geschieht mit dem Ziel, die Besucher besser zu verstehen und die Website kontinuierlich zu verbessern. Es ist grundsätzlich möglich, auch Benutzer von Single-Page-Webanwendungen, einer speziellen Art von Webanwendungen die sich ähnlich wie Desktopanwendungen verhalten, zu beobachten, jedoch ist dies nur eingeschränkt durchführbar. Ziel dieser Arbeit war die Identifikation der funktionalen Verbesserungsmöglichkeiten in Web-Analytics Systemen in Bezug auf Single-Page-Webanwendungen. Weiters wurde eine Modifikation implementiert und die sich daraus ergebende Auswirkung auf die beabsichtigte Langzeitnutzung, die sogenannte *Continuance Intention*, der Web-Analytics Software evaluiert. Im ersten Schritt wurde ein quelloffenes Web-Analytics Framework in einer kommerziellen Single-Page-Webanwendung integriert. Basierend auf der daraus gewonnenen Erfahrung und der durchgeführten Literaturrecherche, wurde eine Liste mit 13 Optimierungsmöglichkeiten inklusive der möglichen Lösungsansätze erstellt. Die Vorschläge wurden von Experten in Hinblick auf die potentielle Nützlichkeitssteigerung der Web-Analytics Software bewertet. Daraus resultierend wurde die *Custom Metrics* Funktionalität implementiert, die es erlaubt, anwendungsspezifische Metriken zu definieren und aufzuzeichnen. Diese Funktion wurde anschließend mittels einer Onlineumfrage, basierend auf einer modifizierten Version des *Expectation-Confirmation Modell,* evaluiert. Dadurch wurden zwei latente Faktoren ermittelt: Die wahrgenommene Nützlichkeit *(Perceived Usefulness)* und die beabsichtigte Langzeitnutzung (*Continuance Intention*). Die lineare Regression zwischen den beiden Faktoren wurde berechnet. Diese statistische Analyse zeigt, dass sich die beabsichtigte Langzeitnutzung signifikant erhöht, wenn eine spezifische Verbesserung mit dem Ziel der Nützlichkeitssteigerung in einer Web-Analytics Software eingeführt wird.

CONTENTS

1 INTRODUCTION

“If you cannot measure it, you cannot improve it.” (Lord Kelvin)

In today’s business world a company cannot succeed in the long run without constantly improving its processes and products. Continuous improvement is one of the major principles in many management models like Lean or Six Sigma (Bhuiyan & Baghel, 2005). The Plan-Do-Check-Act (PDCA) cycle is a simple tool to constantly question the status quo and to implement changes with the goal to improve workflows. The third phase is about evaluating the changes and comparing them against the defined goal(s). However, this requires to measure performance indicators before and after the change, otherwise it is just not possible to determine the impact of it (Moen & Norman, 2006).

Thus, measuring and tracking is a very important activity in today’s business world. This is particularly true for companies with an internet-based revenue model. For example, online retailers need to measure which products were sold and when, and also which advertisement campaign did actually increase sales. Such information can be tracked with the help of web analytics (WA) solutions. These systems are necessary when applying the PDCA cycle to web sites and web applications. This work focuses on challenges faced in WA systems when used with web applications.

1.1 Background

The aim of web analytics is to gather, aggregate and present data about the users of web sites. Such data contains for example, how the users reached the web site, how long and which pages they have viewed, and when and from which page they left the web site. The general idea behind WA is to provide an insight into the users’ behavior (Atkinson, 2007). One of the main objectives is to improve the usability of web sites based on the gathered knowledge. This is especially important for e-commerce platforms, as the ease-of-use for customers has a direct impact on the revenue stream (Hasan, Morris, & Probets, 2009).

An unrelated development relevant for this work is the increasing number of single page applications (SPAs) (Chitkara & McCaffrey, 2016a). This special form of web applications behaves similar to desktop applications, but has the benefit of being device-agnostic and easy to deploy (Mikowski, Powell, & Benson, 2014). As the term SPA already indicates, such an application only consists of one web page, which typically uses JavaScript (JS) to load additional data on-demand from the server and to update certain sections of the page with new content without actually refreshing the whole page. Thus, there is a clear technical difference between SPAs and regular web sites or web applications.

1.2 Motivation

Web analytics frameworks like Google Analytics (GA) or Piwik are historically focused on regular web sites. Although there is some support for analyzing single page applications by tracking user events, the integration with SPAs leaves space for improvement. An initial literature research showed that the scientific community has not put focus on the special considerations and requirements of SPAs regarding web analytics frameworks so far.

However, to improve and enhance commercial web applications it is important to understand how the clients are using the software. It is crucial to know which features are used and which ones are not. This is especially relevant for companies with a subscription-based revenue model, as unsatisfied clients might not renew their subscriptions and this would have an immediate impact on the cash flow (Pugh, 2014). It is relevant to research how to improve the integration of web analytics solutions with (commercial) single page applications as this is a fast-growing industry. The International Data Corporation (IDC) is predicting that the worldwide revenue will almost double from $77 billion in 2017 to $151 billion in 2021 (IDC, 2017b).

The goal of this work is not only to improve the integration of WA software with SPAs, but also to research the impact of such an improvement. To be more precise, the influence on the long-term usage of the WA framework should be analyzed, as regular usage by the users is an important goal for information systems (IS). It is one indicator of sustainable IT investment (DeLone & McLean, 2014).

1.3 Objectives

This subchapter presents the research question (RQ) of this work and the derived hypotheses. Furthermore, a brief overview of the underlying theory is given.

1.3.1 Research Question

The aim of this work is to answer the following research question:

What influence does better support of commercial single page applications have on the continuance intention towards web analytics frameworks?

The *continuance intention* is a construct that describes a user's intention to keep using an information system as defined by Anol Bhattacherjee in the *Expectation-Confirmation Model* (ECM). This latent factor is the main dependent construct in this model (Bhattacherjee, 2001). The evaluation of the research question is based on the ECM, which is explained in more detail in Chapter 2.5.

1.3.2 Hypotheses

The following research hypothesis will be used to answer the RQ.

Research Hypothesis:
Enhancing web analytics frameworks with a focus on modern web applications has an influence on the intention of users to keep using the information system in the long-term.

The directed alternative hypothesis and null hypothesis are derived from this research hypothesis and are used in the statistical hypothesis test.

Alternative Hypothesis (H_1):
Enhancing web analytics frameworks with a focus on modern web applications increases the intention of users to keep using the information system in the long-term.

Null Hypothesis (H_0):
Enhancing web analytics frameworks with a focus on modern web applications has no influence on the intention of users to keep using the information system in the long-term.

1.3.3 Goals

The objective of this master's thesis is to develop two artifacts. The first is a list of identified gaps when using a WA system with an SPA. Every gap is presented with a potential solution to solve the issue. The second artifact of this work is the implementation of one of the suggested improvements.

The list of potential improvements covers the following three areas:

- The integration of a WA framework with a single page application
- The tracking of metrics and goals that are particularly relevant for SPAs
- The problems with existing reports and potentially required new reports regarding web applications

1.4 Approach and Methodology

This subchapter presents the approach followed in order to answer the research question. The three phases of this work and all applied research methods are briefly described.

In the first phase a *literature research* was performed to investigate all topics relevant for this work. The resulting theoretical background is described in Chapter 1. It covers single page applications, commercial web applications and the state of the art in the field of web analytics. Furthermore, concerns regarding digital privacy are discussed. Finally, the theoretical model that is used to evaluate the RQ is described.

The second phase of this work is focused on the practical implementation. It presents both artifacts created as part of this thesis. At first an initial integration of a WA solution with an SPA was performed with the goal to identify gaps in the WA software. Further *literature research* on necessary enhancements with focus on commercial SPAs was conducted. This procedure resulted in a list of gaps and associated potential solutions. This list was evaluated by professionals in the field of software development via a *score voting method* by using paper questionnaire forms. The highest-ranking improvement was subsequently implemented as the second artifact of this work.

The third and final phase covers the evaluation of the developed software artifact. A third *literature research* was conducted to design a survey and therefore to determine how to operationalize the latent variables of the underlying theoretical model. The resulting questionnaire was used to conduct an *online survey* based on items using *Likert scales* and *semantic differential scales*. Furthermore, an *unobtrusive research method* was applied to supplement the results of the survey. The digital footprints taken from the access log file of a web server were used to verify the validity of some questionnaire items. Finally, the survey responses were used to create a *linear regression* model and to perform *statistical calculations*, which were used to answer the research question.

2 BACKGROUND

This chapter presents the theoretical background for this work. It covers single page applications in general as well as the evolution of commercial web applications. Furthermore, a broad overview of web analytics, including a special focus on digital privacy matters in this regard, is given. Finally, the theoretical model used for the evaluation of the research question is explained.

2.1 Single Page Applications

The following chapter describes what single page applications are and how they evolved. Furthermore, benefits of SPAs over traditional web applications and native desktop applications are presented. The last section gives a brief overview of JavaScript frameworks and presents the concept of multi-window *user interfaces* (UIs) in SPAs.

2.1.1 **Development**

Tim Berners-Lee laid the ground work for the success of the World Wide Web (WWW) in 1990 when he implemented the *hypertext transfer protocol*[1] (HTTP) and the *hypertext markup language* (HTML). In this early stage of the internet a web site consisted of multiple static web pages. Navigation was possible through hyperlinks from one page to another leading to a full refresh of the whole page. Users were confronted with long waiting times until the page was loaded and rendered. Thus, delivering a good *user experience* (UX) was not conceivable in these early days. By introducing server-side scripting languages like PHP[2] and *Active Server Pages* it became possible to create dynamic web sites that were generated on the server depending on the input of the user. It allowed making web pages to be more interactive. In 1996 JavaScript was introduced as a client-side scripting language. It enabled dynamic content, as the web page was manipulated directly in the browser as a reaction to user actions. Back then JavaScript was mostly used to modify the UI and to implement business logic (like validations or calculations) on the client-side. It was still a long way from dynamic web pages to single page applications and the Web 2.0 as it is known today. (Doyle & Lopes, 2008; Fink, 2014)

In 2002 Macromedia introduced the term *rich internet application* (RIA). The idea behind such applications was to improve the user experience in the internet. The user interface should be

[1] See https://tools.ietf.org/html/rfc2616

[2] See https://github.com/php/php-langspec/blob/master/spec/00-specification-for-php.md

responsive and the look and feel of the application should be modern. A RIA is displayed in the browser window, but should feel like a native application. Hardware with higher performance and the availability of better and faster internet connections were two developments that made such applications possible. (Noda & Helwig, 2005)

Two different approaches exist for implementing rich internet applications. One approach is to make use of plug-ins installed in the internet browser (e.g. Adobe Flash/Flex, Java Applet, Microsoft (MS) Silverlight) and to embed the application by using these plug-ins within a web site. The second approach is to leverage AJAX (Asynchronous JavaScript and XML) as defined by Garrett (2005). AJAX is not a framework or tool. It is a combination of multiple web standards. The idea is to asynchronously load content from the web server and use the received content to manipulate the current web page on the client-side without reloading the whole page. Thus, less data needs to be transferred between the client and the server and only the updated section of a page needs to be rendered by the browser. (Mesbah & van Deursen, 2007; Noda & Helwig, 2005)

The format commonly used for the payload, it being the data received from the server, was originally XML (Extensible Markup Language), but has been mostly replaced with JSON (JavaScript Object Notation) by now. Using this different serialization format is sometimes called AJAJ (Asynchronous JavaScript and JSON) (Merelo-Guervós, Castillo, Laredo, Mora García, & Prieto, 2008), although, this acronym has not really been adopted by the community. Google Scholar lists 56 search results[3] for AJAJ in scientific papers, whereas Microsoft Academic Search only returns two results[4] based on queries made on September 20th, 2017.

The usage of browser plug-ins for RIA has become deprecated in the past years. The two biggest issues with plug-ins are that they are often not available for all platforms (e.g. no Flash on iOS, no Silverlight on Linux) and that they frequently contain security vulnerabilities. Especially Adobe Flash has become infamous for many critical defects that allowed malicious attacks on client machines (Adobe Systems Inc., 2017). As a result, browsers have already stopped or will stop supporting Flash and other plug-ins in the near future (Laforge, 2017; Microsoft Edge Team, 2017; Smedberg, 2017). In July 2017 Adobe itself announced that Flash will not be provided with further security updates after 2020 (Adobe Corporate Communications, 2017). Using AJAX is the preferred approach nowadays to implement responsive user interfaces on web sites (Fink, 2014). Thus, the remainder of this work will focus on this method.

[3] See https://scholar.google.com/scholar?hl=en&num=20&as_sdt=0,5&q=ajaj+%22Asynchronous+JavaScript+and+JSON%22

[4] See https://academic.microsoft.com/#/search?q=ajaj%20%22Asynchronous%20JavaScript%20and%20JSON%22

2.1.2 **Definition**

A single page application is a special form of a RIA. Instead of enriching the various pages of a web site with dynamic content, there is only a single web page used for the entire web site. For SPAs based on JavaScript (using AJAX) there is only one synchronous HTTP GET request executed from the browser to the server to load the initial web page and the required client-side JavaScript code. After that request, the URL (Uniform Resource Locator) in the address bar of the browser does not change anymore. The loaded page – called shell – consists of multiple segments that can be updated independently from each other. From that point onwards any additional communication with the server is happening asynchronously via AJAX. (Mikowski et al., 2014)

In the late 2000s big companies started applying the AJAX pattern to their web applications to deliver rich internet experiences. Some examples are Google Maps, Google Mail, and Flickr. These popular early adopters made this new approach known to a bigger audience and thus, were crucial in initiating the shift from traditional web applications to SPAs. (Doyle & Lopes, 2008; Farrell & Nezlek, 2007; Fraternali, Rossi, & Sánchez-Figueroa, 2010)

One advantage of SPAs over traditional web applications is, that the web site behaves almost like a native desktop application. Any communication with the server is happening asynchronously and therefore does not necessarily lock the UI for the user. The interaction with the application is fluent, as the whole UI is significantly more responsive. (Farrell & Nezlek, 2007)

Furthermore, SPAs can store the state of the application in the internet browser and thus, can even work without an active internet connection to a certain degree. This is something they have in common with desktop applications. However, a crucial advantage over native applications is that SPAs are immediately available on multiple operating systems without the need to port the source code, as long as an internet browser is available for the targeted platform. (Mikowski et al., 2014)

Accessing the applications via the browser comes with an additional benefit besides portability. SPAs do not require a deployment on the client machine, in contrast to native applications. Besides the internet browser no additional software needs to be installed on the client to access the SPA. Not only is the initial deployment obsolete, but any subsequent upgrades made to the application will take effect without a need to act on the client. This allows for shorter release cycles and an increased number of updates. (Fink, 2014; Mikowski et al., 2014)

In summary, single page applications combine the advantages of native desktop applications like a responsive UI and offline-capability and the benefits of traditional web applications, for instance simple and quick deployment and cross-platform support.

2.1.3 JavaScript Frameworks

At first single page applications were implemented by using plain JavaScript code, which means that all the source code was written from scratch. This approach was only feasible for rather simple applications. As soon as a lot of functionality was required on the client or if features (e.g. drag and drop, asynchronous requests, etc.) needed to be implemented in different ways to be compatible with multiple browsers, it made sense to build the SPA on top of an existing code stack. Therefore, numerous JavaScript libraries and frameworks have been released over the past decades, with the aim to ease and speed up the development of dynamic web sites, web applications and single page applications. Another purpose of such frameworks is to increase and strengthen the security against vulnerabilities in the source code (e.g. to prevent cross-site-scripting). Nowadays, the scope and focus of JS frameworks varies a lot. Some aim to be as lightweight and performant as possible, others try to provide a vast array of features to front-end developers (e.g. abstracting communication with the server, many UI controls, etc.). (Kozlov, 2016; Lennon, 2010; Sviatoslav, 2017)

As of September 2017, some of the most popular JS frameworks are:

- AngularJS[5]
- Backbone.js[6]
- Ember.js[7]
- Ext JS / Sencha Touch[8]
- OpenUI5[9]
- React[10]
- Vue.js[11]

(Fain, Rasputnis, Tartakovsky, & Gamov, 2014; Rambeau, 2017)

[5] See https://angularjs.org/

[6] See http://backbonejs.org/

[7] See https://www.emberjs.com/

[8] See https://www.sencha.com/products/Ext JS/

[9] See http://openui5.org/

[10] See https://reactjs.org/

[11] See https://vuejs.org/

When migrating native desktop applications to an SPA it is desired to preserve the look and feel of the application in the best possible way, in order to allow the trouble-free conversion for users from the original desktop application. Thus, JS frameworks are needed with a special focus on providing rich UI controls for this use case. Furthermore, they need to be able to support having multiple windows opened at the same time and allow resizing, minimizing, and maximizing of these application windows within the SPA. The technical term for this approach is *multiple document interface* (MDI) (James, 2008). Such JS frameworks can even emulate the behavior of a window manager in an operating system (OS) and thus make it possible to implement so called web-based desktops (webtops) (Lawton, 2008). One example for such a rich *Graphical User Interface* (GUI) can be seen in Figure 1. The DiskStation Manager (DSM)[12] software is a web application used to configure and maintain the *network attached storage* (NAS) devices produced by Synology. It is based on the Ext JS framework and imitates an operating system by providing a start menu, a task bar, multiple resizable windows and a user menu in the top right corner.

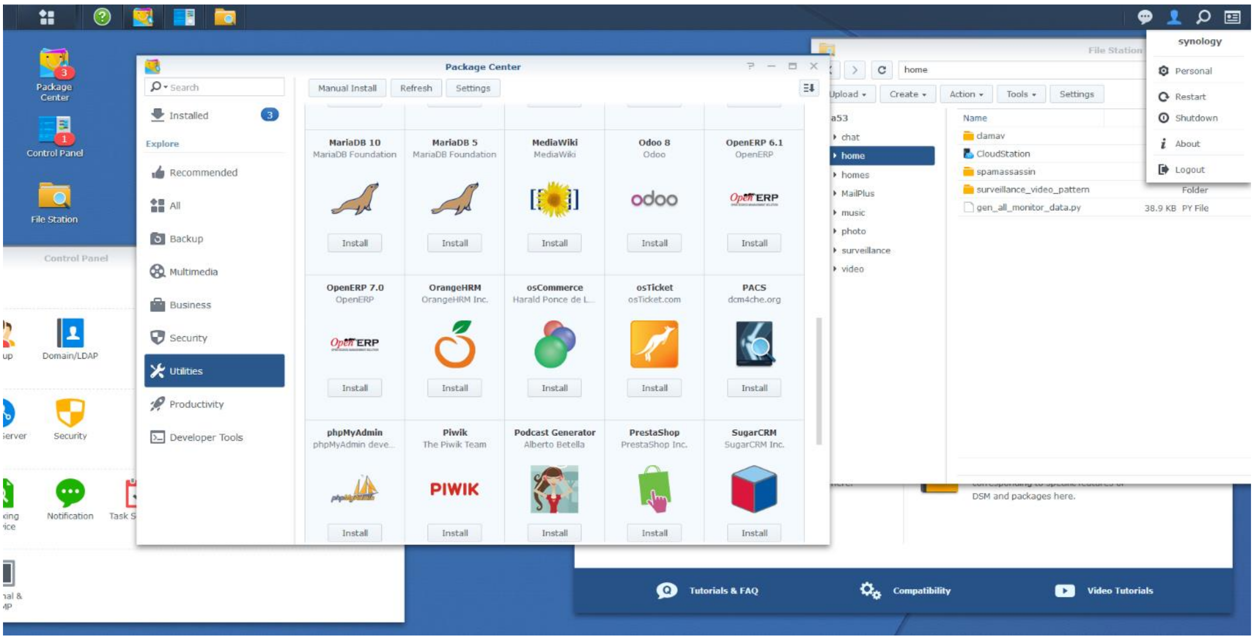


Figure 1: DSM: A web-based desktop for managing a NAS (Synology Inc., 2017)

[12] See https://www.synology.com/en-us/dsm/live_demo

2.2 Commercial Web Applications

Single page applications feel and behave much like native desktop applications, but have big advantages over them in terms of portability and deployment. Thus, independent software vendors (ISVs) have an interest in leveraging these benefits for them by migrating their existing desktop software to SPAs.

At first single page applications were distributed like traditional software and installed on the client's premises, which allowed access to the software from anywhere in the company network. The SPA paradigm gained further popularity with the introduction of a business model called *application service provider* (ASP) in the 1990s, and subsequently led to the rise of the software delivery model *Software-as-a-Service* (SaaS) in the 2000s. More and more ISVs are entering the SaaS market, as the global occurred revenue is expected to increase from $84 billion in 2017 to $160 billion in 2021 (Barrett, 2010; IDC, 2017b; Mangiuc, 2009; Wohl, 2008)

2.2.1 Application Service Provider (ASP)

The current success of the SaaS model is based on two developments that were ongoing in the software industry in the late 1990s. One was the technical progression that made fast and responsive SPAs a reality. The second was the introduction of a new IT business model, which was named *application service provider* by Jostein Eikeland in 1996 (Bianchi, 2000). This Norwegian entrepreneur founded TeleComputing, the first ASP company. This new business model was based on the idea of deploying software not on the client's premises, but in the data center of the provider and to let the client access it via the internet. Subsequently, maintaining and upgrading the software was also taken care of by the ASP. Therefore, clients could focus on their business instead of managing their own IT infrastructure. Main drivers for the rise of ASP were the constant increase of transfer rates of internet connections and the expansion of access to the World Wide Web in general. (Bianchi, 2000; Luit Infotech Private Limited, 2013)

However, in the end most ASP companies did not last, and the business model was not as successful as anticipated. Many declared bankruptcy during the burst of the dot-com bubble. In retrospect Wohl (2008) identified multiple mistakes that were made by most ASP companies. One of the main problems was that they had to invest a lot of money upfront in the necessary data centers and therefore had to predict quite accurately how much resources they would need at a certain point of time. There was no easy way to quickly scale up or down based on demand (yet). A second reason for the failure of ASPs was that the market was not ready for new licensing models like pay-as-you-go or yearly subscriptions back then. Wohl (2008, p. 15) describes it as the problem of an "uneducated market". Furthermore, the providers did not adequately budget for marketing costs to make the ASP approach popular enough amongst potential clients and the market needed more time to develop in general.

2.2.2 Software-as-a-Service (SaaS)

The National Institute of Standards and Technology (NIST) defines SaaS as “the capability provided to the consumer […] to use the provider’s applications running on a cloud infrastructure” (Mell & Grance, 2011, p. 2). The access to the application is usually realized via thin-clients like a web browser, but could also be accomplished via an *application programming interface* (API) for instance in form of a web service: some hosted services do not require a UI front-end as the clients directly integrate them with other software solutions.[13] However, this work focuses only on SaaS solutions that are provided in form of an SPA front-end.

Companies started offering Software-as-as-Service in the 2000s. By some it is considered a successor and by others an evolution of ASP. However, this time around the conditions had changed. On the one hand, the software industry learned from their mistakes by promoting the focus on clients as a *unique selling proposition* (USP) when compared to native desktop applications. On the other hand, the market had time to become “educated”. Which means that enough clients felt that the technology was established and that they acknowledged the value (and price) of this new deployment model. (Wohl, 2008)

One way to measure the readiness of the market is to observe reactions on the stock market related to hosted software. Jeong and Stylianou (2010) conducted an empirical study to determine the impact of public announcements regarding the adoption of hosted software solutions (ASP & SaaS) on the market value of enterprises, which are the service consumers in this scenario. As the result of analyzing press releases between 1999 and 2007 they found a statistically significant positive correlation between the announcements and the shareholder value for large firms, whereas no such effect was found for *small and medium-sized businesses* (SMBs).

One success factor of SaaS was its technical superiority over ASP. For instance, in terms of resource efficiency: Application service providers usually hosted one instance of the provided software per client (called a tenant), whereas SaaS systems are typically multi-tenant capable, which means that one software deployment can serve multiple clients using one database (DB). The data is strictly segregated for each client in the database, although it is processed by the

[13] See https://www.programmableweb.com/category/software-as-a-service/api

same instance of the software. This in turn leads to a lower *total cost of ownership* (TCO) for the SaaS companies, as they need less hardware. (Callender, Marshall, Cardon, & Patel, 2015; Hossain & Shirazi, 2015; Luit Infotech Private Limited, 2013; Mäkilä, Järvi, Rönkkö, & Nissilä, 2010; Wohl, 2008)

Another big difference between ASP and SaaS is that in the former business model the companies often only provided already existing software created by other vendors. They did not develop the systems themselves und thus could not modify them to their needs. Whereas SaaS is in most cases driven by the ISVs themselves. Therefore, the software architecture is designed to be scalable and easily extendable. (Luit Infotech Private Limited, 2013)

In the 2000s the need for upfront investments in hardware became irrelevant. With the ubiquitous availability of “the cloud” there was no need for ISVs to set up an own data center anymore. SaaS providers can rent the required resources based on their demand in form of Infrastructure-as-a-Service (IaaS) or Platform-as-a-service (PaaS). IaaS allows clients to rent hardware like servers (dedicated or virtual machines), storage and network bandwidth based on a service model. PaaS includes the required software stack (operating system, database servers, libraries, and frameworks) on top of IaaS to enable the quick deployment of software in the cloud. These new cloud computing services let the ISVs focus on their core business, the software development. (Hossain & Shirazi, 2015; Wan & Wang, 2014)

In summary, SaaS companies managed to overcome the problems of ASPs and to establish a profitable business model. One popular product in the SaaS market is Microsoft Office 365. This service was launched in 2011 (Motal, 2011). Since then, instead of purchasing the license of an office product, it has been possible to use offline products based on annual subscriptions. MS Office 365 is a bundle of “Office Online” (multiple SPAs) and the traditional office suite, which needs to be installed on the client machine. The subscription includes software updates for the installed version and furthermore upgrades to the newest version as long as the subscription is valid. In July 2017 Microsoft announced that the revenue based on subscriptions exceeded profits made by traditional license sales for the first time (Bright, 2017). (Alzahrani, 2016; Fowley & Pahl, 2016)

Another very popular SaaS company is SalesForce.com Inc. They started their business as a pure ASP company already in 1999 and are one of the few that survived the dot-com bubble burst. They are offering SaaS and PaaS solutions and have built up a vast network of enterprises around their products by allowing integration of third-party add-ons into their system. According to Gartner they ranked number nine amongst global cloud computing companies in 2014 with a revenue of 482 million USD (Chitkara & McCaffrey, 2016b). The top three SaaS companies by revenue in 2014 are Google Inc. Microsoft and ADP (a SaaS-only vendor for human resource software). Further well-known examples for successful SaaS vendors are Atlassian, Survey Monkey, GitHub and Concur. (Chitkara & McCaffrey, 2016a; IDC, 2013; Jeong & Stylianou, 2010; Wohl, 2008)

A survey conducted by PricewaterhouseCoopers (PwC) amongst 32 big software vendors (with an annual revenue ranging from $100 million to $86 billion) in 2015 revealed that already 86% of the participants are offering hosted services (Lo, 2015). Moreover, IDC announced in a press release in August 2017 that they expect the total global public cloud spending in 2017 to be $128 billion (an increase of 25,4% compared to 2016). Furthermore, they predict spending to reach $266 billion by 2021. The article states that SaaS will remain the prevailing cloud computing model with an expected decline from 66% in 2017 to 60% in 2021. They expect higher growth rates for IaaS and PaaS over the next five years. Nevertheless, SaaS and the whole field of cloud computing is a rapidly growing market. (IDC, 2017a, IDC, 2017b)

To summarize, web applications were sold and licensed as software-as-a-product (Mangiuc, 2009) and deployed on the client's premises at first. In the 1990s they were hosted in data centers and accessed via the internet in the form of ASP offerings. Finally, SaaS entered the market in the 2000s with more focus on client needs, new pricing models and by replacing traditional web applications with SPAs. Market research companies are predicting a continued steep growth path for the years to come.

2.3 Web Analytics

One success factor for commercial SPAs is to be aware of how the application is used. For instance, it is crucial to know whether errors occur and if yes, which errors occur, and which are the most used features in the application. In order to achieve this the actions of users need to be logged and analyzed. Web analytics has already been used to accomplish that for web sites for more than two decades and can be leveraged for SPAs as well. This chapter presents a definition of web analytics and describes the two different sources for data. Furthermore, in-house vs. third-party deployment, and concerns regarding the interpretation of WA reports are discussed. Finally, an overview of the terminology used with web analytics is presented.

2.3.1 Definition

“Web analytics is the technology and method for the collection, measurement, analysis and reporting of websites and web applications usage data” (Zheng & Peltsverger, 2015, p. 7674). The goal behind it is to increase the success of a web site. For example, by improving the user experience or by increasing the revenue generated by a web site. The numbers reported by web analytics are used to make data-based decisions.

Clifton (2012) defines web analytics as two different sets of tools. *Offsite tools* provide information regarding the potential audience of a web site, how a web site is doing in comparison with competitors and about current trends on the internet. Whereas *onsite tools* are purely focused on the specific web site itself. They measure how many users are viewing the individual pages, how long they are staying on each page, and much more.

The remainder of this work will focus on onsite tools for applying web analytics to a specific web site.

2.3.2 Data Sources

The initial step of the web analytics process is to gather the relevant data. There are two ways to collect the user data. One approach is to analyze the log files produced by the web server applications in the back-end. The other approach is to embed specific tracking code in the web page itself, which is called page tagging (or web beacons). These techniques and their advantages and disadvantages are presented in the following subchapters.

2.3.2.1 Log Files

The approach to analyze log files produced by the web servers has already been in use since the early 1990s (Kaushik, 2007a). Especially in the days of static HTML pages, every page request let to an entry in the log files of the web server. In combination with the internet protocol (IP) address of the requester it was possible to reconstruct which user was accessing which pages. Special software tools were introduced which analyze the log files and extract the relevant data. (Zheng & Peltsverger, 2015)

A clear benefit of this approach is that the log files are already available on the server. It is very simple to obtain the raw data (Kaushik, 2007a). Thus, already existing historical data can be leveraged as well. Furthermore, the tracking does not interfere with the user’s experience on the web page. Another advantage of analyzing log files is that it is the only technique which can provide information on whether a download was actually completed. This data is only available on the server and cannot be tracked with client-side code (Clifton, 2012).

However, analyzing log files has its limitations. It is not possible to say whether a user was actively viewing the page, or which sections of a page were of special interest. Furthermore, multiple users nowadays often share the same IP address, as they are accessing the web site from a company network or are using a proxy server. Another problem with this approach for collecting the data is the caching of web sites. Proxy servers, internet service providers or the internet browser itself may cache web sites to improve the performance for the end users. This leads to the side effect that the request is not reaching the underlying web server and the gathered data is not representative and thus becomes unreliable. (Kaushik, 2007a) (Clifton, 2012)

2.3.2.2 **Page Tagging**

The second technique to gather data about the user behavior on web sites is to insert a tracking code on the web page itself. In the beginning, this was usually a hidden image that allowed to track when a specific user was requesting a specific page. With the rise of AJAX it became more common to use asynchronous HTTP requests to return the tracking information to a web analytics server. (Zheng & Peltsverger, 2015)

The downsides of page tagging are that it can have a negative impact on the performance of the web page and thus lead to a bad user experience. Furthermore, observed users could become aware of the tracking. This could lead to negative reactions by the users and might cause bad publicity. Moreover, it is possible that users actively take counter measures against the tracking (e.g. by installing privacy add-ons), which will lead to non-representative data. Even if the users are not suppressing the sending of the data by themselves, it could be omitted by a firewall within an institution (Clifton, 2012). Another problem can also be the incompleteness of the data, in case the page tag is not added to all pages of the web site (e.g. because it was forgotten for new pages). Ideally adding of the JavaScript code should happen automatically through a generator or a *Content Management System* (CMS) (Weischedel & Huizingh, 2006).

One of the benefits of page tagging is that it is more powerful than analyzing log files. Very specific events, which only happen on the client, can only be tracked via page tagging. Examples for such events are mouse clicks, input via the keyboard, or starting and pausing of a video. In contrast to log file analysis, the data is captured and available for reporting almost in real-time. The caching of web sites is not a problem for this technique, as the JavaScript code is executed regardless from where the page was loaded. The script approach also provides more flexibility on which data is gathered. Some sections of a web site may be more relevant than others and need more detailed logging. Furthermore, this method is actively enhanced and any ongoing innovation in the field of web analytics is currently applied to page tagging. Most tools for log files analysis are not actively development anymore. (Kaushik, 2007a) (Clifton, 2012)

2.3.3 In-House vs. Third-Party Deployment

Another consideration that needs to be taken into account when introducing web analytics software for a web site is where the gathered data should be stored and thus, who has control over the data. For the first kind of data source – the log file analysis – the raw data is on the web servers. This means the IT department maintains control over the log files. However, when using page tagging this can be either done by deploying the necessary web analytics server in the internal IT infrastructure, or by using the services of a third-party. In that case the request of the JavaScript tag is pointed to the web analytics servers of the third-party. One example is Google Analytics, a SaaS web analytics product. It is offered as a premium and a free version. The gathered data is stored on servers under the control of Google Inc. and the whole reporting and configuration is done via a web interface. (Kaushik, 2007a)

The benefits of using the services of another vendor are that it is easy and quick to set-up. In case of Google Analytics, it comes even free of charge. However, most of the providers only offer paid services. The advantages of an in-house solution are that all the data is stored internally and full control over privacy of the data exists. In case other data sources should be linked with the gathered analytics data (e.g. internal sales numbers) it is easier to set this up and there is no need to send sensitive data to external parties. Furthermore, one might get into a lock-in situation when relying on an application service provider. It might not be possible to retrieve historical data when cancelling the service and even if, it can be a struggle to integrate the existing data with a new solution. (Kaushik, 2007a)

2.3.4 Interpretation

Web analytics measures what is happening on a web site. However, it cannot give insight on reasons for the reported numbers. It can answer questions about “what” and “when” something happened, but not “why” (Weischedel & Huizingh, 2006). Other methodologies or the consultation of experts are required to interpret the numbers (Clifton, 2012). Qualitative data can be used to make sense of the quantitative data. Methodologies for obtaining qualitative information are for example surveys or speak-aloud usability tests (Kaushik, 2007a).

The advantages of WA in comparison to the qualitative methodologies are that data for the whole population is gathered and not just for a selected sample. Furthermore, the tracking itself is discreet and does not influence the behavior of the users (Rogers, Sharp, & Preece, 2011). Guzdial, Santos, Badre, Hudson, and Gray (1994) point out that the advantage of log file analysis is that the data is gathered outside of the laboratory, while users are handling real tasks. Thus, the data is more representative. The same statement is valid for page tagging. By tracking the users unobtrusively, the validity of the data increases and the Hawthorne effect[14] in the data can be avoided. This effect describes a research bias in studies where persons behave differently just because they are aware that they are subject to a study. Ideally both types of data should be gathered to make informed decisions.

Although the whole population is tracked instead of a sample, there is still an incompleteness of data that needs to be considered when interpreting reports based on web analytics data. For determining the data via log files, this stems from requests which never hit the web server, but are served by a proxy or cache instead. In terms of page tagging firewalls or browser add-ons can block the tracking code and lead to incomplete data (Weischedel & Huizingh, 2006). Thus, it is crucial to always be aware of this problem regarding the underlying data when interpreting results. The best approach is to never report absolute numbers. It is better to interpret the results in relative changes between two periods or two segments. (Clifton, 2012)

Another aspect critical for the interpretation of web analytics data is the possibility to apply segmentation on reports. This means that the data is grouped by one or multiple attributes. For example, a report could show all visits to a web site grouped by the information from which original web site – the so-called referrer – the request was coming. Kaushik (2010, p. 88) even states "nothing is more important in analytics than segmentation" as visitors have diverse reasons to use the web site. Furthermore, different reports need to consider different types of users. The benefits of segmentation are that reports become clearer to interpret. Thus, it is easier to derive actions from a report for a specific segment. For example, if the site traffic generated via search engines is declining, applying search engine optimization should be considered. A second example would be cancelling expensive advertisement which only produces little traffic. In general, segmentation puts more information into reports.

2.3.5 Terminology

As various software vendors have created their own applications for web analytics, different terms have been coined in the field of web analytics. In 2007 the Web Analytics Association (WAA)

[14] See https://explorable.com/hawthorne-effect

published a document to unify and generalize the common terms[15]. The following sections are based on these definitions. All relevant terms for this work are presented.

2.3.5.1 **Types**

The WAA (2007) categorized all terms into four different types. Three of them are metric and are so-called measurements: Count, ratio, and key performance indicator (KPI). The fourth type of terms are dimensions. This is in line with the terminology used for data cubes in the field of data mining: Measurements – or also called measures, metrics or facts – have linked various attributes to them – called dimensions (Gray et al., 1997). Thus, a particular measure, such as the number of page views, can be aggregated to a total number per year over all visitors, or filtered down to the number of *page views* for a specific page (e.g. "contact information") made by MS Windows users from Europe in the past week. These examples show the given flexibility when creating reports based on measures and dimensions.

Count

A count is the most basic unit of measure. It is just a single number. In most cases it is an integer (e.g. the number of visits is 4,815), but sometimes it can be a decimal number (e.g. the amount earned with a specific product is € 1,623.42) as well.

Ratio

A ratio is the result of a division. It is usually a count divided by a count. Although a ratio could be used in the numerator or denominator of the division as well. Ratios typically have the word "per" in their name. One example of a common ratio is "page views per visit".

KPI

A KPI is often a ratio but can also be a count. The fact that makes a count or ratio a KPI is its relevance to the business. Burby and Brown (2006, p. 3) define a KPI as a number that is "infused with business strategy". This means that it is of significant relevance to the operator of the web site. Thus, there is no specific set of KPIs. The decision of which numbers are the "key" numbers, depends on the strategic goals and the kind of web site.

Clifton (2012) highlights the importance of the definition of KPIs. Only measuring and tracking information is not enough. Once a company has already tracked some data it is important to

[15] See https://www.digitalanalyticsassociation.org/standards

define KPIs based on that data. It helps to break down the huge amount of gathered data into some understandable and processable numbers. Not only can the success of changes to the web site or new marketing campaigns be measured via KPIs, it also allows to monitor the health of the web site: any negative effects can be observed, and counter measures can be taken. For example, a marketing campaign launched by competitors may draw away visitors from the own web site.

Dimension

A dimension is an attribute of a measurement. It describes the measurement in more detail and can be used to aggregate, segment or filter measurements. Dimensions allow for different views and a varying level of granularity in reports (Golfarelli, Maio, & Rizzi, 1998). A report could for example show the average of page views (= aggregation) grouped by the operating system of the visitor (= segmentation) in the year 2016 (= filter).

Every measurement is saved with a reference to the existing dimensions. It references one entry for every dimension and therefore defines the scope of the measurement, i.e. on which page, at which timestamp and for which user something happened. (Kimball & Ross, 2009)

2.3.5.2 Building Block Terms

In 2006 the WAA started working on a standardization of definitions and terms. At first, they described and defined the so called “big three” in web analytics: page view, visit/session, and unique visitor. As most of the web analytics definitions are based on these three terms they decided to focus on them in the beginning. In 2007, they added the definition of a page and segregated the definition of a visitor in three different categories and thus, formed the so-called building block terms.

Page (Dimension)

A page is a unit of content definable by the web analyst. On a web site of the web 1.0 this would be a web page reachable via a specific URL. For example, a specific product description or the cart summary on an e-commerce web site. However, other types of content can also be regarded as a page. An example for SPAs would be to open a dialog – a form contained in a window, to play a specific video on youtube.com, or to download a file. The individually tracked pages can be used to segment and filter measures, as it is possible for all dimensions.

Page View (Count)

The number of times a page has been requested is defined as a page view. As soon as a subsequent page is requested WA frameworks can save the time the user spent on the page as an attribute of the page view. The time needed by the server to generate the page is often also stored as an additional information with every page view. Such information is used to identify problems with the performance of a web site.

Visit/Session (Count)

A visit – also called session – is at least one page view by an individual user. If other page views are tracked within a certain time window by the same user, they are counted as the same visit. Most WA frameworks terminate a session after a configurable time if the user has not taken another action in that period. As most vendors use different time-out periods or other methodologies to identify sessions, this measure varies a lot when using different WA frameworks for tracking the same web site. This fact needs to be considered when interpreting visits and comparing them to numbers of other web sites (e.g. the competitor’s web site) (Clifton, 2012).

Unique Visitor (Count)

The measure *unique visitors* represents the number of individual users which had a least one visit in a certain reporting timeframe. A user with multiple visits in the defined reporting period is only counted once. This count is always connected to a period of time such as days, weeks, months and cannot stand by itself.

There are various ways to identify unique visitors. The most reliable is via the authentication of the user. However, the majority of web sites do not support or require a user login. Thus, most systems use cookies to uniquely identify visitors as an alternative. However, the resulting number can only approximate the unique visitors, as it actually tracks unique browsers. The issue with that solution is that some browsers or devices are shared by multiple people and some people use multiple devices to access the same web site (Kaushik, 2010). In addition, this approach will lead to a certain degree of error in the tracked data, as some users or firewalls might block the cookies. Furthermore, the deletion of cookies by the users or by some security software will lead to a higher reported number of unique visitors as there existed in reality. Another problem that needs to be considered when tracking this number is the inclusion of non-human visitors like search robots and web site crawlers. Although they are generally excluded by the tools, a certain level of unintentionally tracked non-human visitors needs to be expected. All these issues need to be considered when using cookies for tracking individual users. Using authentication to identify unique visitors is the best way to avoid these problems.

It is crucial to point out that this measurement must not be summed up. The sum of all unique visitors per day over one week might not be the same as the unique visitors for the same week: A visitor accessing the web site on Monday and Tuesday would be counted twice in the first approach and once in the second. This is a common error when interpreting this number and must be avoided. (Clifton, 2012)

Three specific types of unique visitors exist, which are described below: new visitor, return visitor, and repeat visitor.

New Visitor (Count)

A new visitor is a unique visitor who accesses the web site for the very first time. This number can, in contrast to the unique visitors, be summed up across multiple periods. For example, the number of new visitors of last week and the week before that will return the same number as a report regarding the new visitors over the past two weeks.

Return Visitor (Count)

A return visitor is a unique visitor that has accessed the web site at least two times, once in the reporting period itself and once sometime before. A visitor can either be counted as new visitor or as return visitor within a specific reporting period.

Repeat Visitor (Count)

A repeat visitor can either be a new visitor or a return visitor. A unique visitor qualifies as repeat visitor if she/he accessed the web site at least twice within the reporting period.

2.3.5.3 Visit Characterization

This chapter covers terms that are closely related to a visit. It presents different kinds of pages and explains what the referrer of a page view is. Furthermore, all special types of referrers are described.

Entry Page (Dimension)

The entry page is the first page accessed during a visit. An entry page report usually lists all unique entry pages and sorts them by the number of total entries via this page. Thus, it gives a clear picture of which pages the visitors see first when reaching the web site.

Exit Page (Dimension)

The exit page is the last page accessed during a visit. It represents the end of a session. The reporting is similar as it is for entry pages. This dimension can point out pages, which fail to keep the visitors engaged. However, there are pages which are expected to be exit pages, like the last page in an e-commerce process. It is appropriate for a visitor to leave the web site after ordering some goods.

Kaushik (2010) criticizes this metric as it indicates pages, which are responsible for driving away visitors, but in fact every visitor must leave the web site at some point. It is not valid to assume that there was a problem with the specific exit page. He suggests using reports based on bounces instead (see Section 2.3.5.4 Content Characterization).

Landing Page (Dimension)

"Landing page" is a marketing term. It describes a stand-alone page serving a specific purpose. It is usually designed for a particular marketing campaign. Such pages are often accessed via a link sent in an email or reached by clicking on an online advertisement. They aim to get the visitor to take a specific action. For example, to register with a web site, to sign-up for a newsletter or to download an e-book. Therefore, landing pages are pages for whom the marketing department has a special interest. The tracking is needed to determine the success rate of campaigns and the different marketing channels.

Ideally landing pages are also top entry pages. However, both terms should not be confused with each other.

Visit Duration (Count)

The time that has passed while accessing the entry and exit page is tracked as the visit duration. For visits with only a single page view typically no visit duration is logged, as the time the user actively was engaged on the exit page cannot be reliably measured via web site requests.

Referrer (Dimension)

The referrer is the URL from which a page view originates. This information is transmitted via a field in the HTTP request header. However, there are cases where the referrer is empty, which is called a "direct navigation". For example, if a link is clicked on in an email client, a browser bookmark is used to navigate to a web site, or the URL is entered directly in the address bar by the user. The referrer of the entry page – a so-called external referrer – is of particular interest, as it reveals, which external source takes the credit for referring the visitor to the tracked web site.

There are five special types of referrers in total: internal, external, search, visit, and original referrer. A referrer can be either an internal or an external referrer. Any source outside of the tracked web site is regarded as external referrer. This distinction is required to track activities on and off the web site. The external referrer can contain additional valuable information in the form of URL parameters. For example, which keywords were searched for when the request originated from a search portal. If search keywords are contained in the URL, it is called a search referrer.

A search referrer can be an external or internal referrer, as it can originate from a search portal like Google Search, or from a locale search function on the tracked web site itself.

The visit referrer is the very first referrer in a session. In other words, it is the referrer to the entry page. Whereas the original referrer is the first referrer in the very first session of a unique visitor. This dimension reveals which sources take credit for bringing visitors to the web site in the first place.

Page Views per Visit (Ratio)

This ratio represents the number of page views divided by the number of visits within a defined reporting period.

2.3.5.4 **Content Characterization**

This chapter presents counts and ratios that are connected to the entry and exit page. They can give an indication on which pages have engaging and non-engaging content.

Page Exit Ratio (Ratio)

This ratio is the number of exits from a page (whenever the page is the exit page in a visit) divided by the total number of page views for this page. A high value of the ratio indicates pages that fail to keep the visitor engaged. Using the ratio instead of the total number for exit pages allows for a meaningful comparison of pages with high and low traffic.

Single Page Visits (Count)

This number represents all visits that only accessed one page. It does not matter whether the page was accessed once or multiple times in the session. This term must not be confused with single page view visits.

Single Page View Visits / Bounces (Count)

In contrast to single page visits this number only counts visits which accessed a single page only once. In other words, these are visits with a single page view. The term “bounce” is also commonly used for this number, as visitors are figuratively bouncing off the web site.

Bounce Rate (Ratio)

The bounce rate can be calculated for a single page, a group of pages or the entire web site. If the focus is only on a single page, it represents the number of times visitors bounced off the page by the number of times the page was an entry page. For groups of pages the ratio uses the number of times a visitor bounced of any page in the group divided by the number of times any page in the group was an entry page. The bounce rate of the entire web site is the percentage of all visits that were single page view visits.

Kaushik (2010) deems the bounce rate to be the best metric of all, as it is easy to understand and communicate to others. Furthermore, it is straightforward to determine actions based on this metric. It clearly identifies pages that need to be improved.

2.3.5.5 **Conversion Metrics**

The following sections explain events in general and how reporting works for them. Furthermore, conversions and related metrics are described.

Event (Dimension / Count)

The term event is very generic. It is used as a dimension – which event happened – and a count – how often the event happened. This is identical to the relation between a page (dimension) and a page view (count). A page view is only one example for an event, there exist many others like a click, entering registration details, or scrolling down to the end of the content. Events are especially relevant for SPAs, as most user interactions do not lead to a new page view and need to be tracked regardless.

Depending on the situation it makes sense to report events in three different ways:

- By event – Each interaction is considered
- By visit – Each visit during which the event happened at least once is considered
- By visitor – Each unique visitor triggering the event at least once is considered

Conversion (Dimension / Count)

A conversion is defined as a target action completed by a visitor. It is a special kind of an event and can be reported the same way as events in general. The term conversion is used for the dimension and the count.

Clifton (2012, p. 9) defines a conversion as “any action or engagement that builds a relationship with your visitors”. For an e-commerce web site this would mean that a customer purchases items. Examples for other web sites are that a user downloads a brochure or enters her/his personal information in a form. It is basically the purpose of the web site and captures the actions visitors are supposed to do before leaving the web site. Clifton determines that defining these goals for a web site is the most important task when creating a web site, as it allows measuring how successful the web site is.

There are two kinds of conversions as defined by the WAA:

- Goal conversions – also known as target, or macro conversions – are key milestones completed by the visitor. For example, submitting an order in an online shop or registering as a new member on a social media platform.
- Micro conversions – also known as step, support, or mini conversions – are the completion of tasks that might lead later to a goal conversion. Examples are putting items in the shopping cart or signing up for a newsletter in an online shop.

Kaushik (2010) uses the two terms micro and macro conversions and defines the latter similar to the WAA as the main business objectives of a web site. For example, an online shop wants to make revenue and thus, it will track how many visitors actually order something. Whereas micro conversions are also actions worth tracking, but are only loosely connected to the macro conversions. They are targets of secondary degree. For example, the registration of a user account or writing a review for a product on an e-commerce platform. However, Lee, Podlaseck, Schonberg, Hoch, and Gomory (2000) uses the term micro conversions in a different meaning. He defines them as the specific steps that lead to a macro conversion. They are integral parts of the macro conversion. In the example of an online shop they would be actions such as looking at a product, adding the product to the shopping cart and the various steps of the checkout process. All single steps of a process that lead to a conversion are also called a conversion funnel (Clifton, 2012). This work will follow Clifton and use the term micro conversions for targets of secondary degree and conversion funnel for all steps in the process that lead to a conversion.

Conversions are of special interest for marketing departments as well, since they are used to track the success rate of marketing campaigns. They can also be used as an additional kind of segmentation for visitors (e.g. reporting of exit pages separately for registered and unregistered users).

Conversions can and should be connected to an economic value. This can either be the revenue that the completion of the target brought for the business or which costs have been saved by the conversion. This task is easier in the field of e-commerce, but should also be applied for non-e-commerce web sites. For example, the economic value of a customer finding an answer to a product related question in the support area of a web site should be measured with the same amount as an employee in a call-center would have cost to help the customer. Connecting conversions to a monetary value can also help to justify improvements to the web site in front stakeholders. (Kaushik, 2010)

Conversion Funnel (Combination)

A conversion funnel – also just called funnel – is a distinct group of steps (page views or events) leading to a conversion. It is sometimes also called a “click stream path”. The most common example for a funnel is the checkout process in an online shop. A funnel report shows the percentage of potential conversions aborted at some step in the process. (Clifton, 2012)

Abandonment Rate (Ratio)

Whenever a visitor enters a conversion funnel, but does not follow the process all the way until to the goal conversion, she/he abandons the conversion. The percentage of visitors entering a step, but not proceeding to the next step, is the abandonment rate for this specific step. This metric is related to the exit page ratio, but has a lot more significance to it, as every lost conversion means a missed chance for making revenue. (Kaushik, 2010)

2.3.5.6 Summary

A summary of all terms covered in the previous chapters can be found below in Table 1. It gives an overview of all terms, the category, and the type they belong to, and provides a brief description of the term.

Term	Category	Type	Description
Page	Building Block	Dimension	A unit of content like an HTML document or a dialog in an SPA.
Page View	Building Block	Count	The number of times a page was requested.
Visit/Session	Building Block	Count	A group of page views by a unique visitor within a short duration.
Unique Visitor	Building Block	Count	A uniquely identifiable human user of the web site.
New Visitor	Building Block	Count	A unique visitor with the first visit in the reporting period.
Return Visitor	Building Block	Count	A unique visitor with a visit in the reporting period and at least one before it.
Repeat Visitor	Building Block	Count	A unique visitor with at least two visits in the reporting period.
Entry Page	Visit Characterizations	Dimension	The first page accessed during a visit.
Exit Page	Visit Characterizations	Dimension	The last page accessed during a visit.
Landing Page	Visit Characterizations	Dimension	A page designed for a particular marketing campaign.
Visit Duration	Visit Characterizations	Count	The time between accessing the entry and the exit page.
Referrer	Visit Characterizations	Dimension	The URL where a page view originates from.
Internal Referrer	Visit Characterizations	Dimension	A referrer that belongs to the tracked web site.
External Referrer	Visit Characterizations	Dimension	A referrer that does not belong to the tracked web site.
Search Referrer	Visit Characterizations	Dimension	A referrer that originated from search results.

Term	Category	Type	Description
Visit Referrer	Visit Characterizations	Dimension	The referrer of the entry page.
Original Referrer	Visit Characterizations	Dimension	The referrer of the entry page in the first visit of a unique visitor.
Page Views per Visit	Visit Characterizations	Ratio	The number of page view entries divided by the number of visits.
Page Exit Ratio	Content Characterizations	Ratio	The number of exit page entries divided by the page views for a specific page.
Single Page Visit	Content Characterizations	Count	The number of visits, which only accessed a single page.
Single Page View Visit / Bounce	Content Characterizations	Count	The number of visits with only a single page view.
Bounce Rate	Content Characterizations	Ratio	The number of bounces divided by the entry page entries for a specific page
Event	Conversion Metrics	Dimension/ Count	Any tracked action by the visitor. For example, a page view, a mouse click or input via the keyboard.
Conversion	Conversion Metrics	Dimension/ Count	A target action completed by the visitor. For example, completing the checkout process in an online shop.
Goal Conversion	Conversion Metrics	Dimension/ Count	A key target action in terms of the business objects of a web site.
Micro Conversion	Conversion Metrics	Dimension/ Count	A subordinate target action that supports the business objective of a web site.
Conversion Funnel	Conversion Metrics	Combination	A sequence of defined actions that lead to a goal conversion.
Abandonment Rate	Conversion Metrics	Ratio	The percentage of visitors that do not take the next step in the funnel.

Table 1: Summary of WA terminology based on definitions by Burby and Brown (2007), Kaushik (2010) and Clifton (2012)

2.4 Digital Privacy

The general goal of web analytics is to draw conclusions from aggregated tracking data. This is often happening in combination with different segments of visitors, e.g. by segmenting reports by the country of origin or by the days since the last visit. The goal of WA is never to track individuals in order to gain insight into their behavior. However, it could be that some gathered information falls under the objective of data privacy regulations.

The following sections cover the considerations that need to be made when operating a web analytics solution in regard to digital privacy. An overview of the most important changes introduced by the general data protection regulation is presented as well. Furthermore, some measures that users can take to protect their digital privacy are covered.

2.4.1 **Current Legal Situation**

There are no legal concerns regarding the gathering and storing of tracking data per se, as long as the information is impersonal or anonymous. It must not be possible to connect the data to a specific person. However, as soon as tracked information is qualified as *personally identifiable information* (PII) various privacy regulations come in play. Many countries have passed laws granting the right of self-determination regarding the personal data to each individual themselves. This privilege often consists of the right to decide who stores which data, the right to receive a report on all already stored personal data and the right to request the deletion of said personal data. (Hassler, 2017)

The following topics in the field of web analytics need special consideration in terms of digital privacy:

- User input
- IP addresses
- User authentication
- Cookies
- Privacy Policy

(Hassler, 2017)

User input

As soon as the input of a user into form fields is tracked and stored, it can be a digital privacy related topic, as the user may enter information qualifying as PII, like her/his name or her/his email address. Hassler (2017) recommends including a disclaimer on any form to which the user needs to agree to before submitting the data, e.g. by ticking a checkbox. The entity collecting the data needs the visitor to actively give her/his consent, which is commonly called to “opt-in”. The opposite is to “opt-out”, meaning that the user is by default part of a process and needs to act to not be part of it anymore (e.g. unsubscribing from an email newsletter). (Hassler, 2017)

IP addresses

Although an IP address seems like an impersonal attribute it may still be classified as a personally identifiable information in some cases. Especially German data protection authorities consider it a PII. Thus, IP addresses are deemed a grey area in this perspective in Europe. The general advice is to either not store them at all, or to anonymize them by only storing a subset of it. (Petrlic & Sorge, 2017)

In general, IP addresses only offer a marginal information gain on visitors to web analysts. Except for one use case: Geolocation, where the IP address is used to determine the visitor’s location. The mapping of the internet protocol address to a physical location is done in such cases via lookup tables. Providers like MaxMind[16] are maintaining mapping information in databases, which allows to link the IP address to countries, states or even cities. However, Huiyan (2017) found out that an anonymization of the IP address by trimming the last octet for IPv4 addresses or trimming the last 80 bits (out of 128 bits in total) of IPv6 addresses led only to a reduction in accuracy on the state level of two to four percent and to a reduction on the country level of one percent. Therefore, the use case of geolocation is no excuse for saving the original IP address together with the tracking information. This should be avoided in order to abide by privacy laws.

User authentication

For any web site that offers registration and authentication features to users it is very likely that some of the information gathered via web analytics has the nature of PII. This fact is especially relevant for commercial web applications, as they need some method to recognize legit users. Service providers have two options when having an authentication system in place. They either need to be very thorough and make sure to not track any PII at all (e.g. by never tracking any

[16] See https://www.maxmind.com/en/home

user input), or they need to get the consent of the user upfront, which is usually done by presenting a privacy policy in the form of an opt-in during the registration process (and before any data has been tracked). (Hassler, 2017)

Cookies

Web analytics frameworks are leveraging cookies to recognize returning visitors to a web site. In 2009 the EU passed a directive[17], commonly called the "EU cookie directive". It regulated that any cookie which does not have a clear functional use for the underlying web site requires the consent of the user. However, the EU missed to indisputably clarify whether the consent must happen via an opt-in or opt-out procedure. Thus, many web sites just display an informative popup, when a user visits for the first time (Petrlic & Sorge, 2017). For instance, the web site nintendo.co.uk displays the text "By using this website, you agree to our use of tracking cookies. You can change your tracking cookie settings here[18]" (Nintendo of Europe GmbH, 2017). Web applications that provide a user authentication do not need such cookies to identify returning visitors. In this case it is advised to turn off the use of cookies in the web analytics solution.

Privacy Policy

Regardless of whether PII or non-PII is actually stored, it is recommended to provide users with an easily accessible and understandable – by using common language – web page that transparently explains what kind of data is gathered, why this is happening and what is done with the tracked information afterwards. (Clifton, 2012)

All topics covered above need to be considered when integrating a web analytics solution with a web application. Furthermore, changes in legislation and representative trials concerning this topic should be monitored as well whenever users are tracked.

2.4.2 General Data Protection Regulation (GDPR)

The current legal situation will change on May 25th, 2018 when the general data protection regulation[19] (GDPR) will come into force. The GDPR will not require any further national legislation, as it will become active law throughout Europe in May 2018.

[17] E-Privacy Directive 2009/136/EG. See http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32009L0136

[18] The word "here" links to http://www.nintendo.co.uk/Legal/Website-Privacy-Policy/Website-Privacy-Policy-637785.html#trackingcookies

[19] Regulation 2016/679. See http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A32016R0679

The major novel points in this regulation are:

- The area of application of European data protection legislation will expand. Any entity (also outside of the EU) will need to abide by the regulation in case it offers products or services within the European union, or if individuals within the EU are tracked by the entity.
- Entities need to be able to prove that any processing of personal data abides by the law. Furthermore, they need to document all measures they have taken to reach this state.
- Entities need to create and maintain a list of potential risks. They need to handle the identified risks regarding data processing accordingly and they are obliged to create a mitigation plan for a potential data breach.
- In case of data breaches the supervisory authority must be notified within 72 hours after detecting the incident. Probably the most significant change by the GDPR is that the regulation allows to impose sanctions after infringements. Fines up to €20 million or four percent of the global yearly turnover of a company - whatever is higher - are possible in case of violations of the GDPR.
- Pseudonymisation is a new term introduced by the law. It describes a process where PII gets transformed in a way so that additional information is required to connect it to an individual. For instance, by encrypting PII it cannot be linked to a person anymore without knowing the private key. However, the resulting data is still considered personal and not anonymous data. The goal of pseudonymisation is to reduce the risk for the data subjects. Applying pseudonymisation will become especially relevant in case of data breaches and may lead to lower fines as the lost data is not as critical.

(Petrlic & Sorge, 2017)

The GDPR covers of course a lot of other topics, although the ones presented above are the most relevant for operators of web analytics solutions.

2.4.3 Countermeasures

Although private data is already protected quite well through legislation, many individuals take extra measures to defend their digital privacy. This section gives a brief overview of different ways of how users can detect if they are being tracked. Furthermore, methods on how to prevent the tracking altogether are presented.

Do Not Track

The term "Do Not Track" (DNT) describes a feature implemented in the HTTP protocol and in modern web browsers with the goal to give individuals control over whether they are being tracked or not. The user can configure the DNT behavior in the browser settings and when accessing a web site, this information is sent to the web server in the form of a flag within the HTTP request. The web site provider is expected to adhere to the flag and to not track the user when receiving such a flag, e.g. to not create any marketing profiles in order to deliver personally customized adds. (Zheng & Peltsverger, 2015)

Anti-Tracking Tools

Privacy Badger is a browser plug-in released by the Electronic Frontier Foundation (EFF), which suppresses tracking code on web sites (Petrlic & Sorge, 2017). When accessing a web page, the tools shows a complete list of all trackers on this page and allows a granular configuration on which trackers are allowed to observe the user and which ones are not.

Ghostery is another privacy plug-in working very similar to Privacy Badger. An additional focus of this plug-in is to inform users about slow trackers, which have a negative impact on the performance of the web page. Thus, this tool aims not only to protect the digital privacy, but also to improve the user experience of web sites.

Saito et al. (2017) also developed a browser extension as part of their research, but they are following a more generic approach. They aim to detect all tracking scripts dynamically by identifying JavaScript code that is listening to certain events (e.g. clicks, keystrokes, and scrolls) and sending a request to a third-party domain when these events get triggered. The tool suppresses such identified trackers. In comparison to Privacy Badger and Ghostery the extension only blocks event tracking and no page tracking, as the authors do not consider page tracking such a big risk for data privacy.

PrivacyScore

PrivacyScore[20] is an online platform that checks web sites throughout the internet for privacy and security issues and ranks them based on the results. The platform allows to create groups of web sites, for instance, a list of all car manufactures or all German universities. The thought behind it is to list all competitors ordered by their score on one web page and to allow an easy comparison for the users of the platform. The operators of PrivacyScore aim to provide an incentive to entities with a bad ranking to resolve identified issues and therefore to go up in the list. The authors state in their paper that they want to encourage entities to switch from Google Analytics to Piwik, as it is considered to be the better choice in terms of protecting digital privacy. (Maass, Wichmann, Pridöhl, & Herrmann, 2017)

[20] See https://privacyscore.org/

2.5 Expectation-Confirmation Model

When introducing a web analytics solution, concerns regarding legal risks are obvious. However, those are not the only considerations that should be made. A WA framework is after all also an information system and any considerations that need to be made when implementing an IS in a company must be adhered to in that case as well. One common concern is how well the users accept the new system. As IT investments are representing an essential financial effort, it is crucial to investigate the initial acceptance and furthermore to maintain and nourish the long-term usage of the system. This work aims to increase the usefulness of web analytics solutions for single page applications and is specifically interested in the long-term perspective. The Expectation-Confirmation Model is regularly used in IS research to explain the continuing usage of IS systems. It is a so-called post-acceptance model, which means that the constructs capture the situation after the initial use of an IS. (Hossain & Quaddus, 2012). This model will be used for the evaluation in this thesis.

The ECM was developed by Anol Bhattacherjee in 2001. He derived it from the *Expectation-Confirmation Theory* (ECT), which describes how the confirmation or disconfirmation of expectations is connected to the initial use of artifacts (Oliver, 1980). The ECT was developed in the field of marketing research and used to explain customer behavior regarding products; whereas the model defined by Bhattacherjee describes the continued use of an artifact with special emphasis on information systems (Bhattacherjee, 2001). Furthermore, the focus is not on the customer (i.e. the person paying for a system) anymore, but on the consumer (i.e. the actual user of a system) (Hossain & Quaddus, 2012).

The ECM consists of four constructs: The main dependent construct is the *IS Continuance Intention* (CI) and the three independent constructs are *Perceived Usefulness* (PU), *Satisfaction* (SF) and *Confirmation* (CF) as illustrated in Figure 2. All four constructs and the associations between them are briefly explained in the following paragraphs.

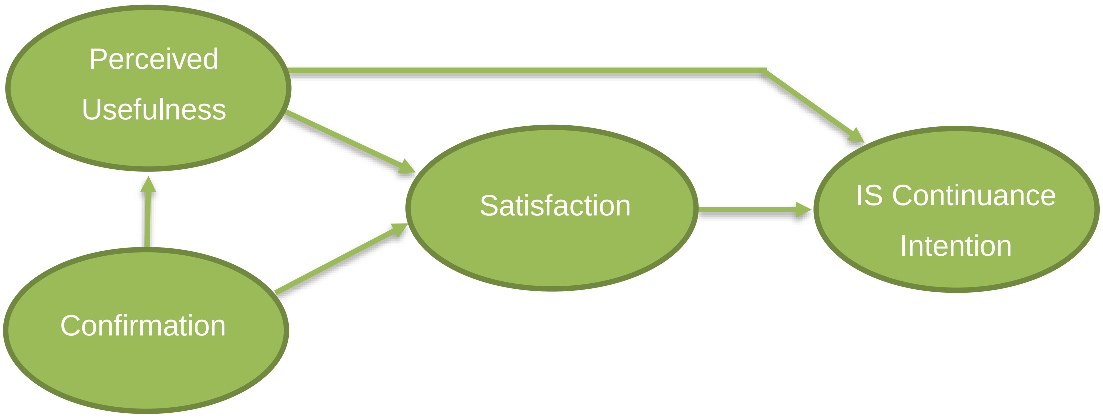


Figure 2: Expectation-Confirmation Model (Bhattacherjee, 2001)

IS Continuance Intention

This construct captures the intention to continue using an already accepted IS. The term "continuance" stands for "continued use" (in contrast to "initial use" or "acceptance"). The *IS Continuance Intention* is positively linked to the users' level of *Satisfaction* with the information system and the ongoing *Perceived Usefulness* of the IS (Bhattacherjee, 2001). This means for instance, if a user is highly satisfied with the IS, it is very likely that she/he will continue to the use the information system.

Satisfaction

This construct covers the satisfaction with using the IS. It also implicitly captures any pre-usage - also known as pre-acceptance - variables. The *Satisfaction* is positively associated with the *Confirmation* of expectations and the continuous *Perceived Usefulness* in which the user beliefs in (Bhattacherjee, 2001). For example, if a user perceives a software as impractical, she/he will be dissatisfied with the software.

Perceived Usefulness

The *Perceived Usefulness* is a post-acceptance construct based on individual beliefs and expectations. It depends on the ongoing use of the IS. This construct is often confused with pre-usage expectations as defined in other theoretical models like the *Technology Acceptance Model* (TAM), but Hossain and Quaddus (2012) emphasize its post-usage nature in the ECM.

The *Perceived Usefulness* is positively associated with the *Confirmation* of expectations (Bhattacherjee, 2001). The link between these two constructs represents the assumption that any initial perceived usefulness (before actually using an IS) will be adjusted once the system has been used and the perception will continue to adapt throughout the ongoing usage of the system

(Bhattacherjee, 2001). For instance, a user with too low expectations will increase the initial perceptions to a more realistic level, whereas very high expectations regarding an IS will be adjusted by the user after a disconfirmation of those expectations.

Confirmation

The fourth construct covers the *Confirmation* (or also disconfirmation) of expectations towards the IS. Bhattacherjee (2001) states that it also implicitly captures pre-acceptance variables similar to the *Satisfaction*. Thus, he defined all four constructs as post-acceptance factors and did not include any explicit pre-usage constructs in his model. The *Confirmation* is positively linked with the *Perceived Usefulness* and the *Satisfaction*. (Bhattacherjee, 2001)

Adaptation of the ECM

Two constructs of this model are required for answering the research question of this work. The *IS Continuance Intention* is used to capture the users' intention to continue using the implemented web analytics solution. However, the *Perceived Usefulness* must be adapted to fit the needs of this thesis. In addition to describing the perceptions regarding the already experienced web analytics system the construct is altered to also measure the alleged value of an enhancement, which aims to improve the support for commercial single page applications in the web analytics solution. Thus, this factor acquires a pre-usage nature. All other constructs are still focused on the unmodified web analytics software. This modification allows to answer whether the perceived usefulness of the enhancement has an influence on the continuous usage intention towards the complete WA system.

The next step after determining how to evaluate the research question is to initially integrate a web analytics solution with an SPA.

3 IMPLEMENTATION

This chapter describes the practical implementation performed as part of this work. The first step was to initially integrate a WA solution with an SPA. This was followed by researching and summarizing identified gaps. The final step was to resolve one of the gaps by enhancing the WA system.

3.1 Initial Integration

The initial integration of a web analytics solution with an SPA is the prerequisite to understand the challenges of such a project and to identify gaps in WA systems regarding web applications. This subchapter briefly presents Reval (the tracked SPA) and Piwik (the web analytics system). It covers the requirements and decisions why this particular WA solution was selected. Finally, the technical integration of the two information systems is presented.

3.1.1 Reval (Commercial SaaS)

Reval Inc. is a financial SaaS solution provider, which has its headquarters in New York City and subsidiaries located in London, Delhi, Graz as well as several other locations. This ISV develops and distributes a cloud-based *treasury and risk management* (TRM) solution in the form of a single page application. The scope of Reval, which is the name of the software, is to support clients in managing cash, liquidity and financial risks and in performing hedge accounting (Reval Inc., 2017c). As the system is distributed as SaaS, clients can get access to the software based on annual subscriptions instead of purchasing a license. Currently about 650 clients have subscribed to Reval (Reval Inc., 2017b).

PwC conducted a pricing survey amongst leading companies in the software industry in 2015. Lo (2015) reports that 84% of all respondents in the survey are already offering subscription-based pricing models to their clients. This pricing strategy comes with the benefit of a steady and predictable yearly cash flow, in contrast to the considerable higher payments via traditional licensing models, which usually happen only once (Fowley & Pahl, 2016). However, the challenge every commercial SaaS provider needs to face, is to not lose existing clients. The goal is to understand the needs and impediments clients are facing, to keep them pleased with the software, which should lead to yearly renewals of the subscription. Thus, one of the KPIs for any ISV with a subscription-based revenue model is the *renewal rate*. It is measured in the percentage of clients that did renew their contract out of all existing clients. The opposite of it is the *churn rate*. (Pugh, 2014)

One factor in ensuring a high renewal rate is to understand how the clients operate the software. It is critical to know which features are used and which ones are not. Furthermore, any usability improvements should be designed by first understanding the status quo. Subsequently it is crucial to confirm that any resulting changes in the software are actually improving the usability. Web analytics is a tool well suited for such tasks. Thus, a WA solution will be integrated with the Reval SPA.

3.1.2 Requirements

This section covers the goals and the functional and technical requirements for the integration of the web analytics system.

The general goal of the implementation is to gain more insight in how the software is used by the clients. It is anticipated that the detailed knowledge will enable the following tasks:

- Identifying and redesigning dialogs with major usability issues
- Improving performance on heavily used and slow dialogs
- Tracking the adoption rate of new features
- Identifying problems in the workflow (e.g. with the order of steps in a process, or based on regularly occurring errors)
- Identifying areas of the application that are hardly used
- Improving the prioritization of the product backlog
- Installing health monitoring to track effects of code and configuration changes (e.g. to detect a significant drop in usage of a specific dialog in a future release)

3.1.2.1 Functional Requirements

The following paragraphs cover the functional requirements for a WA software that should be integrated with the Reval SaaS application.

Visitor attributes

Some minimum functional requirements need to be met when integrating a WA solution with the SPA in terms of tracking dimensions. The software needs to be able to track general visitor attributes. For example, the operating system, the browser (including engine and version), the type of hardware used to access the SPA (e.g. desktop, tablet, smartphone, etc.), the screen resolution used to display the web application and the location and time zone of the visitor. Any of these visitor attributes should also be available in filters in order to apply segmentation on reports.

Tracking dialogs

It must be possible to track which dialog was opened and when and how long it stayed open. Knowing the dialogs that are frequently used would be a reasonable starting point for analysis. However, it should also be possible to report on the path a user takes through the system, for example by visualizing it in a so-called user journey map. Understanding how users reach a dialog and what they do next is an important information and can be leveraged to improve the navigation and to streamline workflows.

Tracking actions

Besides tracking the dialogs, it is required to track certain actions that users are performing within the application. This starts with high level activities such as login, logout or accessing the reference manual, but also entails a more detailed level to truly understand how users are operating the software. Examples for more comprehensive action tracking would be observing any actions regarding grids (e.g. reloading, sorting, filtering, hiding/showing columns, etc.), any button clicks, switching the focus between UI controls and more.

Definition and monitoring of goals and funnels

Finally, it is required to define certain goals and to monitor the micro conversions throughout the software. For instance, a funnel could consist of the most important steps to enter a new loan trade in the system. Subsequently, it should be possible to report how many users abort in any of the defined steps. The hardest part in terms of a web application is to define meaningful goals, as they should ideally be linked to an economic value. For example, whenever a user opens the reference manual or accesses the client community portal this probably would avoid an email or a call to the client support department and thus, should be tracked with a positive monetary amount.

Summary of the functional requirements:

- Must support tracking of general visitor attributes
- Must support tracking of dialogs
- Must support tracking of actions
- Must allow the definition and monitoring of goals and funnels

3.1.2.2 Technical Requirements

This section presents all technical requirements for integrating a WA solution with the Reval SaaS system. Every requirement is presented alongside the reasoning behind it.

Support UI with multiple windows

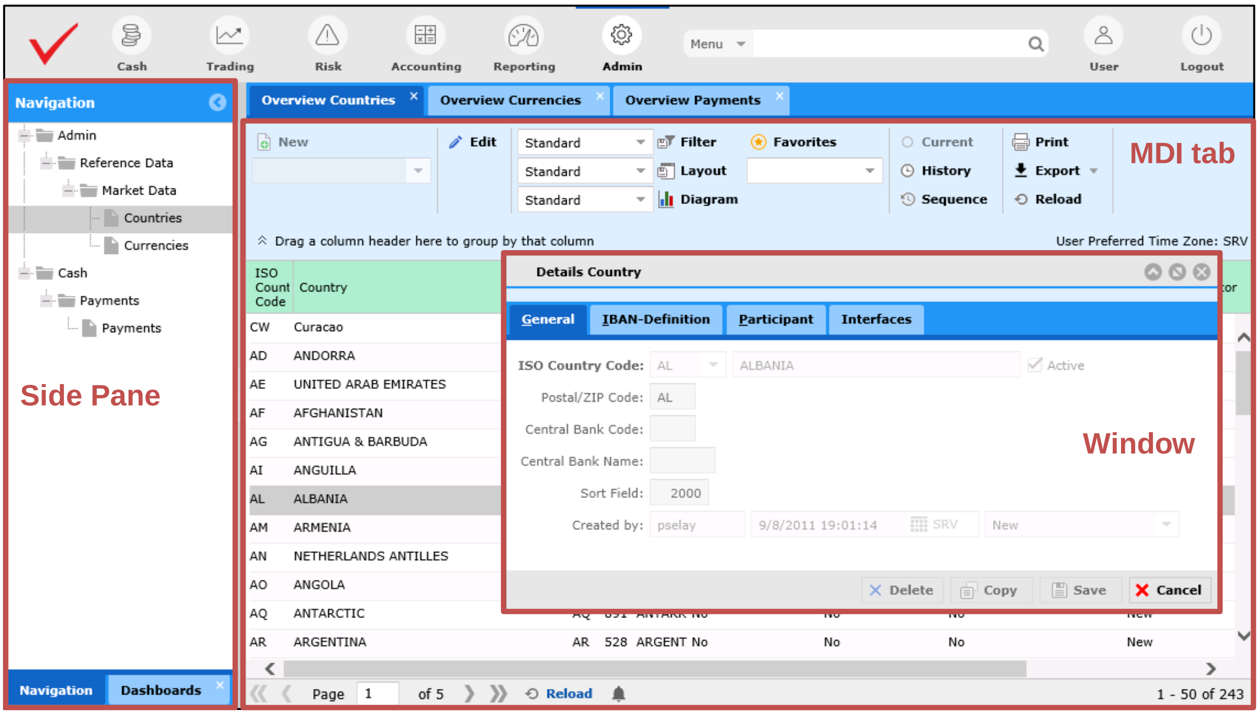
Reval is capable of showing multiple windows at the same time, similar to a native desktop application. Within the software the term *dialog* is used for such a window. It corresponds to the *view component* as defined in the model-view-controller (MVC) architectural pattern. Furthermore, it is equivalent to the more commonly used word *form* (e.g. in Delphi or HTML). Cowell (1996) describes the form as a collection of UI controls. Every form must be encapsulated in a container in order to be displayed. Such containers can be a tab or a window. This description of forms also applies to dialogs in Reval.

However, there exists some ambiguity about the term dialog. In JavaScript and Delphi this expression is only used for very simple modal windows, with only one or no UI control at all, like the *input box* (Morris, 1994). Though, in Microsoft .NET the term dialog has a different meaning. It describes a standard window, like the print dialog, or the file chooser dialog (MacDonald, 2006). Regardless of the existing ambiguity, this work will continue to use the term dialog based on the definition by Cowell (1996), as this is the commonly used word within Reval Inc.

In Reval a dialog can be put in one of three different containers: a window, an MDI tab, or a side pane. The UI of Reval uses concepts such as side panes or MDI tabs, as they can be found in integrated development environments (IDEs) like Eclipse or MS Visual Studio. In Reval there is one center pane and one collapsible side pane on the left. The center pane is used to display the MDI tabs. Every MDI tab can have multiple windows hovering above it. These windows are movable and resizable. The focus can be switched between windows of the same MDI tab and between different MDI tabs. Figure 3 shows a screenshot of the Reval UI. It contains the “Navigation” side pane, the window “Details Country” hovering above the selected MDI tab “Overview Countries”, and two other unselected MDI tabs.

The front-end of Reval has been implemented based on the Ext JS framework[21], which supports such UI concepts by nature. As a technical requirement, the WA solution must support the tracking of multiple simultaneously open dialogs in the application.

[21] An example for another native-application like UI: https://www.sencha.com/blog/ext-js-customer-spotlight-greentree-international/



ISO Count Code	Country						
CW	Curacao						
AD	ANDORRA						
AE	UNITED ARAB EMIRATES						
AF	AFGHANISTAN						
AG	ANTIGUA & BARBUDA						
AI	ANGUILLA						
AL	ALBANIA						
AM	ARMENIA						
AN	NETHERLANDS ANTILLES						
AO	ANGOLA						
AQ	ANTARCTIC						
AR	ARGENTINA	AR	528	ARGENT No	No	No	New

ISO Country Code: AL ALBANIA ☑ Active

Postal/ZIP Code: AL

Central Bank Code:

Central Bank Name:

Sort Field: 2000

Created by: pselay 9/8/2011 19:01:14 SRV New

Figure 3: A screenshot of Reval displaying the three different dialog containers (Reval Inc., 2017a)

Data Source: Page Tagging

As presented in Chapter 2.3.2 there are two different data sources available for obtaining tracking data. The required information can either be extracted from the log files, which get produced by the web server; or by embedding a so-called page tag into the client-side code of the web application. The selected WA solution must support the latter data source in the form of a JavaScript tracker, as there are many actions happening only within the browser in Reval. The web server does not know for instance if a user switches the focus between MDI tabs. Thus, the tracking of user actions needs to be done on the client and not on the server. The server log files are not detailed enough to meet the requirements.

Page Tracking & Event Tracking

Historically, WA frameworks only focused on tracking pages and page views. With the introduction of RIAs it became necessary to also support the tracking of the client-side-only interaction with the web application. Nowadays this is done via event tracking. Atterer, Wnuk, and Schmidt (2006) were amongst the first to research possibilities to track such client-side actions via JavaScript in 2006 and subsequently implemented a tool to generically track any mouse and keyboard interaction on a web site. On October 16, 2007 (about a year later) Google Analytics introduced *Event Logging* (Kaushik, 2007b), which allows the tracking of such actions via specific JavaScript code in the form of a non-generic approach. By today this has become a standard feature in web analytics (Kaushik, 2010).

The Reval SPA will use virtual page views, as the browser does not actually load a new web page, to track individual dialogs in the system. Event tracking will be leveraged for any actions the user performs. This way two out of four functional requirements are covered.

On-premises deployment

As another requirement it is crucial that Reval Inc. keeps control over any logged data. Sending tracking data to a third party would raise legal questions and probably involve adjustments of client contracts. A second and even more important consideration is the opinion of the clients. Potential sensitive data that would be transferred to an external WA provider might lead to concerns amongst clients and increased discontent. Thus, a technical solution is required that can be hosted on the premises.

Source code open for modifications

One limitation in current web analytics frameworks is their focus on tracking web sites and e-commerce solutions. It requires special effort to integrate event tracking with SPAs. Therefore, it might be necessary to adapt the WA system to the needs of the web application. The underlying source code must be available for modification. A *free and open-source software* (FOSS) should be selected for the integration.

Summary of the technical requirements:

- Must support tracking of multiple simultaneously open dialogs
- Must use (client-side) page tagging as data source
- Must support page and event tracking
- Must be available for on-premises hosting
- Must be FOSS

3.1.3 Selection

There exist many WA solutions and every year new (mostly SaaS) products are entering the market. The indisputable industry leader is Google Analytics with a market share of 83.5% as of November 3rd, 2017 (W3Techs, 2017b). However, as GA is neither released under an open-source license nor can it be hosted on the premises, it is no option for the integration with Reval based on the requirements defined above. Table 2 shows a list of all FOSS, which have released a stable version after January 1st, 2016, and which support tracking via JavaScript. All pure log-file based systems were excluded from the list.

Name	Latest stable release	Latest release date	Relative popularity	Relative activity	# of web sites
Countly[22]	17.09	21.09.2017	8.4	9.6	121[23]
Open Web Analytics[24]	1.6.0	28.03.2016	6.1	3.9	10,786
Piwik[25]	3.2.0	12.10.2017	9.6	9.4	1,022,680
Snowplow[26]	92	11.09.2017	8.8	8.9	844,339

Table 2: Overview of all FOSS that match the defined criteria (BuiltWith.com, 2017; libhunt.com, 2017)

These four systems (Countly, Open Web Analytics, Piwik and Snowplow) were compared against each other based on different measures. The web site *Awesome Self Hosted* is a directory for FOSS and ranks the covered analytics software[27] against each other in two categories: relative popularity (based on the number of stars and watchers on GitHub) and relative activity (based on the number of forks and commits on GitHub). Piwik reached the highest popularity with 9.6 and the highest activity with 9.4 out of these four projects. It did not get the highest score of 10 as *Awesome Self Hosted* lists any software related to analytics in the same category and the number one entry in this category is Elasticsearch[28], which is a search engine and not a web analytics software. (libhunt.com, 2017)

The web portal BuildWith.com[29] browses the entire internet and analyzes which technology is used on each web site. Therefore, they provide numbers of how many out of all 374,463,632 web sites in the WWW use one of the four investigated WA solutions. Piwik has again the highest number with 1,022,680 deployments as of November, 2017 (BuiltWith.com, 2017). W3Techs.com[30] provides a similar service and lists Piwik as the only open-source solution in the top ten with a market share of 2.1% (W3Techs, 2017b).

Piwik meets almost all functional and technical requirements. Only the support for funnels is not part of the software and would need to be acquired as a premium plug-in[31]. Based on the analysis

[22] See https://github.com/Countly/countly-server

[23] There might be a problem with detecting the usage of Countly, as this reported value seems unrealistically low.

[24] See https://github.com/padams/Open-Web-Analytics

[25] See https://github.com/piwik

[26] See https://github.com/snowplow/snowplow

[27] See https://selfhosted.libhunt.com/categories/1647-analytics

[28] See https://www.elastic.co/products/elasticsearch

[29] See https://trends.builtwith.com/analytics

[30] See https://w3techs.com/technologies/overview/traffic_analysis/all

[31] See https://plugins.piwik.org/Funnels

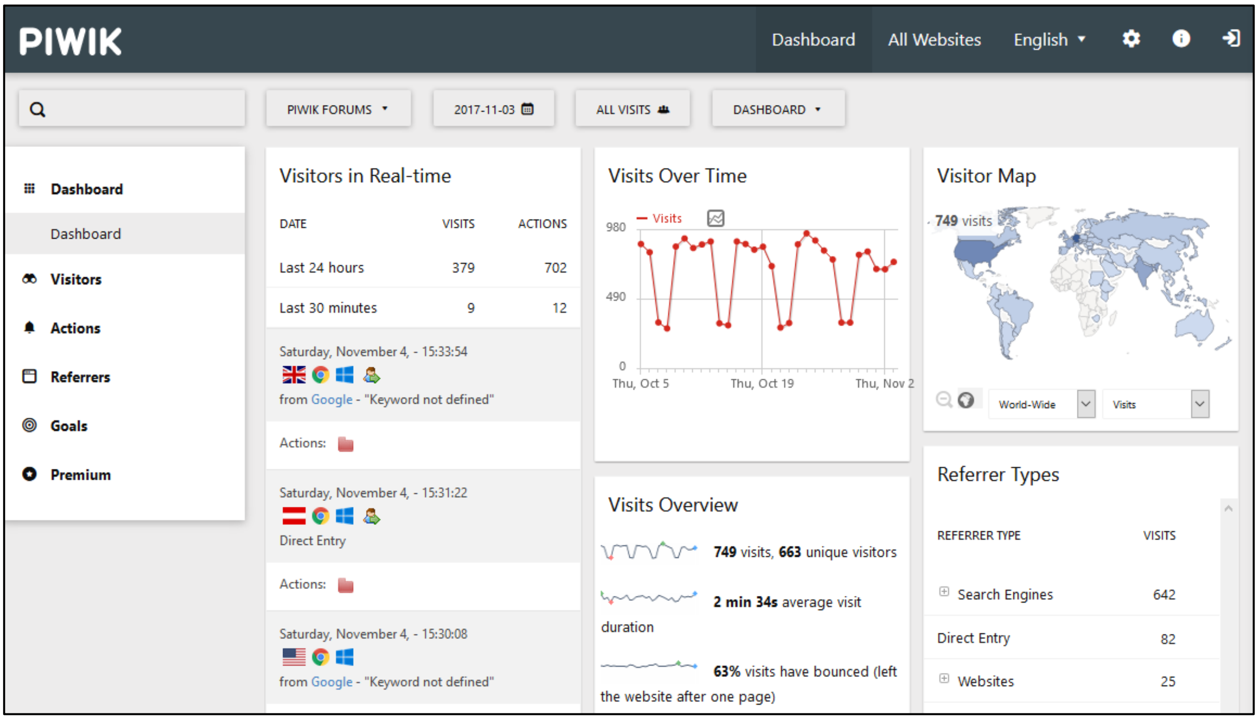
above Piwik is the favorable choice amongst the four solutions. It will be used for the integration with the TRM application.

3.1.4 Piwik (Web Analytics)

In 2003 Matthieu Aubry started working on *phpMyVisites*, a web analytics program written in PHP. Four years later many fellow developers had joined the project and they decided to start with a new web analytics software from scratch. They aimed to leverage the experience they had gained so far and to design the new software with an improved architecture, i.e. with a special focus on extensibility and an easy access to data via a powerful API. The name of the new project was Piwik. (Piwik Core Team, 2008)

The back-end of Piwik is also implemented in PHP and uses a MySQL/MariaDB database. It can be deployed on many different web server systems. The official recommendation is to use Microsoft IIS, nginx or Apache. The front-end of Piwik has been completely rewritten with the release of version 3.0 and uses now AngularJS.

Since its launch in 2007 the software has been constantly improved. Some of the outstanding features are real-time reporting, advanced privacy settings, reporting on the complete set of tracked data (no sampling), configurable dashboards via widgets and a wide degree of extensibility in general through the use of plug-ins (Miller, 2012). Figure 4 is a screenshot of such a dashboard taken from https://demo.piwi.org. The dashboard shows the information about visitors in real-time and general statistics about visitors that accessed the Piwik forum on November 3rd, 2017.



Visitors in Real-time

DATE	VISITS	ACTIONS
Last 24 hours	379	702
Last 30 minutes	9	12

Saturday, November 4, - 15:33:54
from Google - "Keyword not defined"
Actions:

Saturday, November 4, - 15:31:22
Direct Entry
Actions:

Saturday, November 4, - 15:30:08
from Google - "Keyword not defined"

Visits Overview

749 visits, **663** unique visitors

2 min 34s average visit duration

63% visits have bounced (left the website after one page)

Referrer Types

REFERRER TYPE	VISITS
Search Engines	642
Direct Entry	82
Websites	25

Figure 4: A screenshot of the dashboard for the Piwik forums on https://demo.piwik.org from November 3rd, 2017. (Piwik.org, 2017a)

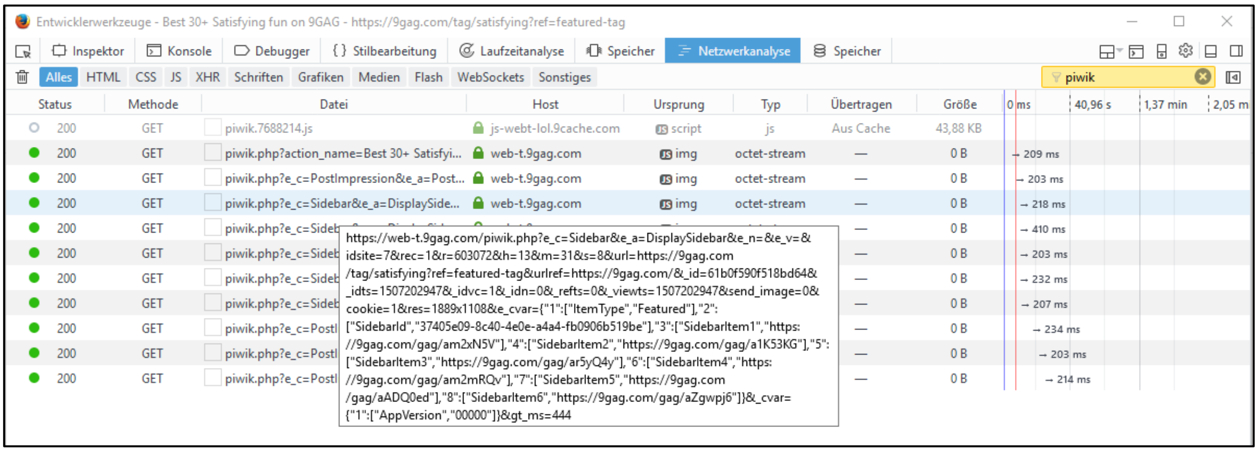
As already stated in the previous chapter Piwik is quite actively developed. The project receives about 20 commits per week in average[32] and releases a new version roughly every month[33]: Version 3.2.0 was released on October 12th, 2017. The documentation[34] is very detailed and continuously maintained.

W3Techs (2017b) reports Piwik to be the sixth most used web analytics tool globally with a market share of 2.1%. When segmenting the reported numbers by the server location it shows that Piwik is even the market leader in Germany with a market share of 36.3% (W3Techs, 2017c). This is most likely due to stricter privacy regulations in the European Union and in Germany in particular. A popular web site outside of Germany using Piwik is 9GAG (W3Techs, 2017a). Figure 5 shows captured network traffic between a Firefox web browser and 9GAG's Piwik tracking server. This screenshot also documents how the integration of a web site and the tracking server can look like: Certain events in the browser trigger an HTTP GET request to the tracking servers to load an image file. All the tracking information is send in the form of URL parameters to the server as part of that request.

[32] See https://github.com/piwik/piwik/graphs/commit-activity

[33] See https://piwik.org/changelog/

[34] See https://piwik.org/docs/#analytics-features



Entwicklerwerkzeuge - Best 30+ Satisfying fun on 9GAG - https://9gag.com/tag/satisfying?ref=featured-tag

Status	Methode	Datei	Host	Ursprung	Typ	Übertragen	Größe	0 ms 40,96 s 1,37 min 2,05 m
○ 200	GET	piwik.7688214.js	js-webt-lol.9cache.com	script	js	Aus Cache	43,88 KB	
● 200	GET	piwik.php?action_name=Best 30+ Satisfyi...	web-t.9gag.com	img	octet-stream	—	0 B	→ 209 ms
● 200	GET	piwik.php?e_c=PostImpression&e_a=Post...	web-t.9gag.com	img	octet-stream	—	0 B	→ 203 ms
● 200	GET	piwik.php?e_c=Sidebar&e_a=DisplaySide...	web-t.9gag.com	img	octet-stream	—	0 B	→ 218 ms
● 200	GET	piwik.php?e_c=Sideb				—	0 B	→ 410 ms
● 200	GET	piwik.php?e_c=Sideb				—	0 B	→ 203 ms
● 200	GET	piwik.php?e_c=Sideb				—	0 B	→ 232 ms
● 200	GET	piwik.php?e_c=Sideb				—	0 B	→ 207 ms
● 200	GET	piwik.php?e_c=Postl				—	0 B	→ 234 ms
● 200	GET	piwik.php?e_c=Postl				—	0 B	→ 203 ms
● 200	GET	piwik.php?e_c=Postl				—	0 B	→ 214 ms

```
https://web-t.9gag.com/piwik.php?e_c=Sidebar&e_a=DisplaySidebar&e_n=&e_v=&
idsite=7&rec=1&r=603072&h=13&m=31&s=8&url=https://9gag.com
/tag/satisfying?ref=featured-tag&urlref=https://9gag.com/&_id=61b0f590f518bd64&
_idts=1507202947&_idvc=1&_idn=0&_refts=0&_viewts=1507202947&send_image=0&
cookie=1&res=1889x1108&e_cvar={"1":["ItemType","Featured"],"2":
["SidebarId","37405e09-8c40-4e0e-a4a4-fb0906b519be"],"3":["SidebarItem1","https:
//9gag.com/gag/am2xN5V"],"4":["SidebarItem2","https://9gag.com/gag/a1K53KG"],"5":
["SidebarItem3","https://9gag.com/gag/ar5yQ4y"],"6":["SidebarItem4","https:
//9gag.com/gag/am2mRQv"],"7":["SidebarItem5","https://9gag.com
/gag/aADQ0ed"],"8":["SidebarItem6","https://9gag.com/gag/aZgwpj6"]}&_cvar=
{"1":["AppVersion","00000"]}&gt_ms=444
```

Figure 5: Network traffic between a Firefox web browser and 9GAG's Piwik tracking server (Mozilla Foundation & Mozilla Corporation, 2017)

3.1.5 Integration

The integration of Piwik in a common web site which uses regular hyperlinks to navigate to a new page is quite simple. One just needs to add a JavaScript snippet – the tracking code – to every HTML page that is delivered to the client. This is usually done via a page generator in a *Customer Relationship Management* (CRM) software or in the back-end software (written in PHP, ASP.NET, Node.JS, etc.) on the server. As soon as the browser renders the received web page and executes the embedded JS, a request is sent to the tracking server. However, the integration with SPAs is more complex.

3.1.5.1 Single Page Applications

An extensive research showed, that although there exist various frameworks for developing SPAs (e.g. Angular JS, vue.js, Ext JS, etc.), in most cases there is no way to simply enable tracking in the SPA through a feature of the underlying framework. This is in line with Holmes (2016), who states that it can be tricky to integrate web analytics in single page applications. He recommends to carefully consider whether to develop a new application as an SPA or a traditional web application. Holmes acknowledges that there are already extensions available for Angular JS, which can ease the integration process. AngularJS in contrast to Ext JS is following a *single document interface* (SDI) approach, i.e. it shows only one dialog at a time. It uses URL hashes to identify the current state (i.e. dialog) of the application. This concept makes the integration simpler. However, Reval allows for displaying multiple dialogs simultaneously. There is no generic approach in the Ext JS framework to integrate web analytics in a web application. Virtual page views and events need to be tracked explicitly via JavaScript code[35] [36] [37] [38]. Such an integration needs to be done specifically for each SPA.

3.1.5.2 Reval

This section describes the initial integration of Piwik with Reval. It was performed based on version 3.0.4[39], which was released on May 17th, 2017. The following paragraphs cover a high-level architecture as well as considerations made regarding the tracking of users. The actual integration via JavaScript with the front-end is explained and all features that get tracked as part of the initial integration are presented. Finally, the rollout process at Reval Inc. is described.

Architecture

An important decision in the overall architecture was that the Reval application server does not directly communicate with the Piwik tracking server. All the communication goes through the client, as can be seen in Figure 6. After successfully completing an AJAX request to the application server, the client sends a tracking request to the tracking server. This server is running Piwik. It persists the information in a database on a separate server. Therefore, the application servers are not impacted by integrating the tracking functionality in the application.

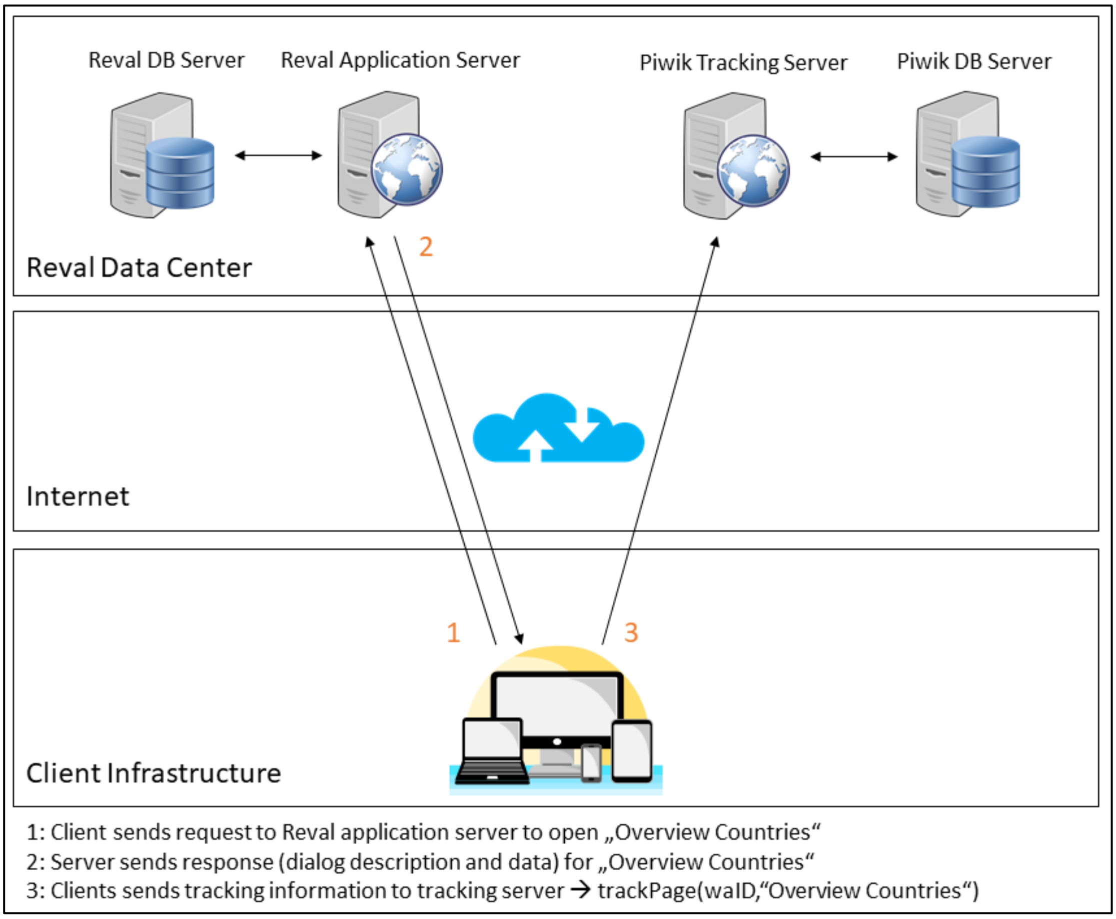
[35] See https://developers.google.com/analytics/devguides/collection/analyticsjs/single-page-applications

[36] See https://developers.google.com/analytics/devguides/collection/analyticsjs/events

[37] See https://piwik.org/blog/2017/02/how-to-track-single-page-websites-using-piwik-analytics/

[38] See https://developer.piwik.org/guides/tracking-javascript-guide#javascript-tracker-features

[39] See https://github.com/piwik/piwik/releases/tag/3.0.4



1: Client sends request to Reval application server to open „Overview Countries“
2: Server sends response (dialog description and data) for „Overview Countries“
3: Clients sends tracking information to tracking server → trackPage(waID,"Overview Countries")

Figure 6: Architecture of the integration of Piwik with Reval

User tracking

Within Europe multiple laws containing strict regulations on personally identifiable information exist. Thus, it must not be possible to identify the individual user based on the tracked data[40]. However, on the other hand all sessions of the same user must be linked together in order to be able to answer questions like "How much percent of all users log in at least once per week?". The userID[41] feature provided by Piwik will be leveraged to connect multiple visits of the same user to each other. The expected userID can generally be any unique identifier such as a username or an email address, but that would link the tracked data to an individual. In order to comply with privacy regulations Reval Inc. introduced the artificial *waUserID*. It is a pseudonymisation of the real user ID. The server generates it after a successful login and sends it to the client for tracking purposes.

[40] See https://piwik.pro/blog/pii-personal-data/

[41] See https://piwik.org/docs/user-id/

JavaScript integration

The Piwik JavaScript tracker was specifically integrated with the Reval web application. However, no tracking is happening on pages without an active session (e.g. the login page, the logout confirmation page). During the login process it is determined whether or not the tracking is enabled for the user and only then the web analytics specific information (`waURL`, `waUserID`) is send to the browser. The Piwik server (defined by the `waURL`) hosts a JavaScript library (piwik.js) to integrate tracking on the client-side. The first step after the login is to load this tracking script from the server asynchronously and then use the exposed API[42] to track page views and events. Listing 1 shows the JS code that sets up the tracking on the client. The URL to the Piwik tracking server, the `userID` and the `siteID` is configured before loading the script asynchronously. The `siteID` is required to distinguish multiple web sites using the same tracking servers, which is not required for Reval. Listing 2 presents the function that is used to track an event via the API.

```
var _paq = _paq || []; // this array is only used for the initial
activation, it gets replaced with an object
(function() {
    _paq.push(['setTrackerUrl', RevalUser.waURL + '/piwik.php']);
    _paq.push(['setSiteId', '1']);
    _paq.push(['setUserId', RevalUser.waUserID]);
    var d=document, g=d.createElement('script'),
          s=d.getElementsByTagName('script')[0];
    g.type = 'text/javascript';
    g.async = true;
    g.defer = true;
    g.src = RevalUser.waURL + '/piwik.js';
    s.parentNode.insertBefore(g,s);
})();
```

Listing 1: JavaScript code to load the tracking script asynchronously (Piwik.org, 2017a; Reval Inc., 2017a)

```
function piwik_trackEvent(category, action, name, value) {
    _paq.push(['trackEvent', category, action, name, value]);
}
```

Listing 2: JavaScript code to track an event via the JS API (Piwik.org, 2017a; Reval Inc., 2017a)

[42] See http://developer.piwik.org/api-reference/tracking-javascript

<u>Functional Features</u>

The following functional features of the application are tracked as part of the initial integration: Dialogs are generically tracked as virtual page views, either when opened (including the response time of the application server) or when an already open dialog receives the focus. Various types of events are tracked as well:

- All button clicks (generically)
- Printing of a dialog
- Pressing the help button (on a particular dialog)
- Displaying of error, message and warning popups (on a particular dialog)

3.1.6 Rollout

A deployment on the production environment in the Reval data center is out of scope for this work. Reval Inc. is only releasing two major upgrades per year. The first is happening every April and the other one every October. As intensive security and performance testing needs to be executed for this new component, the web analytics enhancement could not be considered for the October release in 2017. The rollout on the productive environment is scheduled for April 2018. Thus, this thesis does not focus on gathering client information, as it is not necessary for answering the research question.

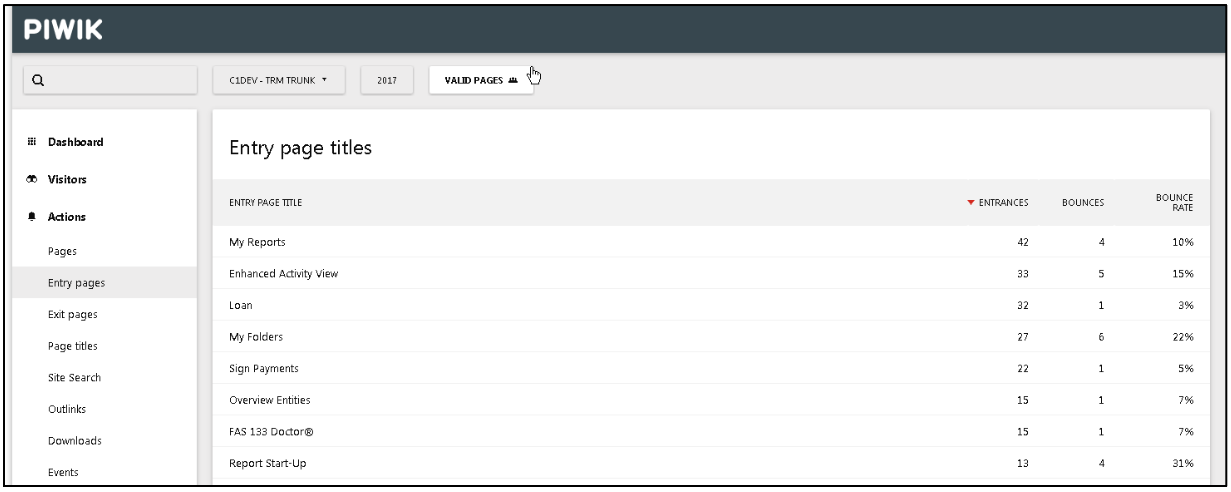
However, the tracking of users has been enabled on an internal test deployment in August 2017 to gain more insight into the reporting capabilities of Piwik. The whole product development division (Product Management, Software Engineering and Quality Assurance) has actively used this environment to continuously perform manual testing of new features in the application. The gathered information was used to evaluate the reports and to identify gaps in Piwik regarding SPAs.

A crucial step during the internal rollout was to inform the entire product development division about the integration of web analytics and to demonstrate the reporting capabilities of Piwik in general. The briefing was held in September 2017 and covered three objectives:

- Inform employees about the project, the current status and the roadmap
- Demonstrate the Piwik front-end and show them how to use the tool
- Ask the staff for suggestions on how to improve the integration

All participants have been provided with credentials to the test deployment and were encouraged to review the system. This was important for allowing to answer the research question, as employees need to know the unmodified system to make a proper comparison with the enhanced version.

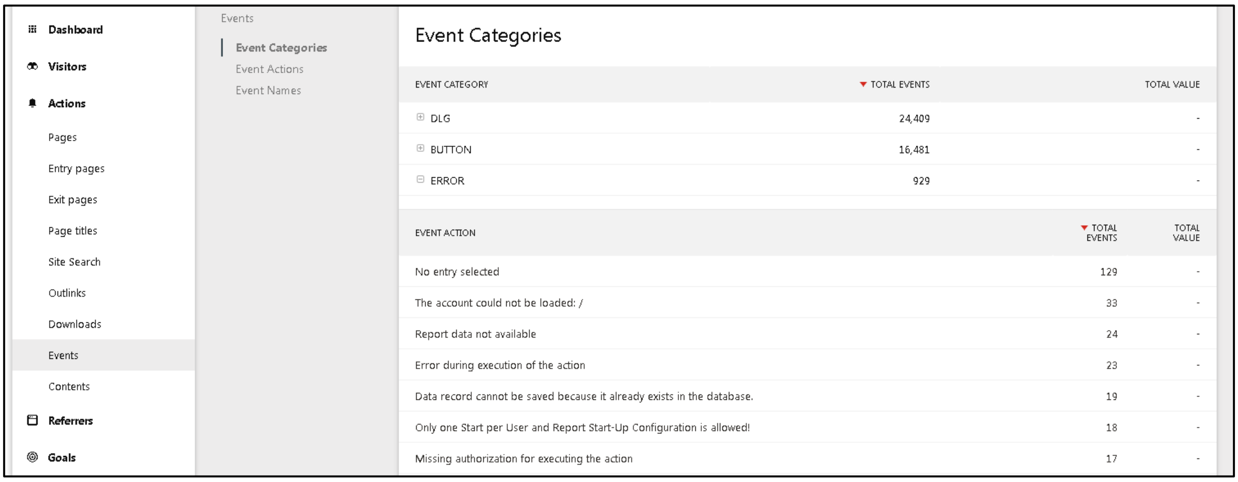
The following two screenshots were taken of the Piwik reporting front-end. Both reports are based on data gathered on the internal test deployment. The “Entry Page Titles” report depicted in Figure 7 presents the top entry pages based on the page title. In the case of Reval these are the first dialogs opened after a successful login. Figure 8 shows the top error messages in the “Event Categories” report that occurred on the test deployment.



Entry page titles

ENTRY PAGE TITLE	ENTRANCES	BOUNCES	BOUNCE RATE
My Reports	42	4	10%
Enhanced Activity View	33	5	15%
Loan	32	1	3%
My Folders	27	6	22%
Sign Payments	22	1	5%
Overview Entities	15	1	7%
FAS 133 Doctor®	15	1	7%
Report Start-Up	13	4	31%

Figure 7: The “Entry page titles” report shows the dialogs opened most often directly after the login. (Piwik.org, 2017a)



Event Categories

EVENT CATEGORY	TOTAL EVENTS	TOTAL VALUE
⊞ DLG	24,409	-
⊞ BUTTON	16,481	-
⊟ ERROR	929	-

EVENT ACTION	TOTAL EVENTS	TOTAL VALUE
No entry selected	129	-
The account could not be loaded: /	33	-
Report data not available	24	-
Error during execution of the action	23	-
Data record cannot be saved because it already exists in the database.	19	-
Only one Start per User and Report Start-Up Configuration is allowed!	18	-
Missing authorization for executing the action	17	-

Figure 8: The “Event Categories” report showing the most common error messages. (Piwik.org, 2017a)

3.2 Suggested Enhancements

This chapter puts a focus on the special considerations and requirements of a single page application regarding web analytics frameworks. All gaps presented below were identified based on literature research and first-hand experience gained during the initial integration of Piwik with

Reval. Every description of the missing functionality is followed by a recommendation on how to resolve the gap. Three different categories of suggested enhancements were defined: *Integration gaps* are deficits in functionality that impact the integration of an SPA with a WA (or specifically the Piwik framework) itself. *Metrics gaps* cover measurements which currently cannot be tracked, but would be important for a web application. This also includes reporting on these new metrics, whereas the *reporting gaps* do not entail new metrics, but point out missing reporting capabilities, which are required especially for single page applications.

3.2.1 **Integration Gaps**

The following chapter describes issues that exist when integrating Piwik with a single page application. The problems are explained, and potential solutions are presented.

3.2.1.1 **Generic Tracking**

While researching how to integrate a WA framework with an SPA based on Sencha Ext JS, it became obvious that there is not much documentation available in this area. The community has been requesting built-in support for WA frameworks in Ext JS now and then[43] [44], but so far nothing has been implemented in this regard. The same conclusion was drawn when specifically researching how to integrate Piwik with Ext JS by searching in the Piwik forum[45]. This search query produced only two results, and both were reports about a specific problem and not focused on how to do the integration in general. The literature research showed that the situation is similar for other JS frameworks such as Backbone.js and vue.js; although there are plug-ins available[46] for Angular[47], React[48] and Ember.js[49] [50], which provide a generic tracking of page views. However, this only works when implementing a GUI following an SDI approach, where the hash fragment in the URL is representing the current state of the web application. Specific events (e.g. a button click, ticking a checkbox, etc.) within a dialog still require manually adding JS code to the UI controls, which takes care of the event tracking. The research showed that no generic approach for MDI based JS frameworks exists so far.

[43] See https://www.sencha.com/forum/showthread.php?113042-Google-Analytics-Use-in-Sencha-Touch

[44] See https://www.sencha.com/forum/showthread.php?191308-Integrating-Google-Analytics

[45] See https://forum.piwik.org/search?q=extjs

[46] See https://piwik.org/integrate

[47] See http://angulartics.github.io/

[48] See http://www.npmjs.com/package/piwik-react-router

[49] See http://ember-insights.github.io/

[50] See https://emberobserver.com/categories/analytics

The suggested enhancement to resolve this gap is to enhance JS frameworks in a way to automatically track a set of certain events like the “onClick” event on button controls, or the “onChange” event on a radio button group. This would reduce or even eliminate the need for developers, who implement an SPA on top of the framework, to integrate the web analytics framework themselves. So specifically, the Ext JS framework itself (not an SPA based on it) should be integrated with the Piwik web analytics framework to generically track page views and events. The technical approach could either be to modify the source code of the UI controls in the framework directly, or to develop a separate module in JavaScript that is weaved into the framework by applying aspect-oriented programming techniques.[51] [52]

3.2.1.2 Configuration of Generic Tracking

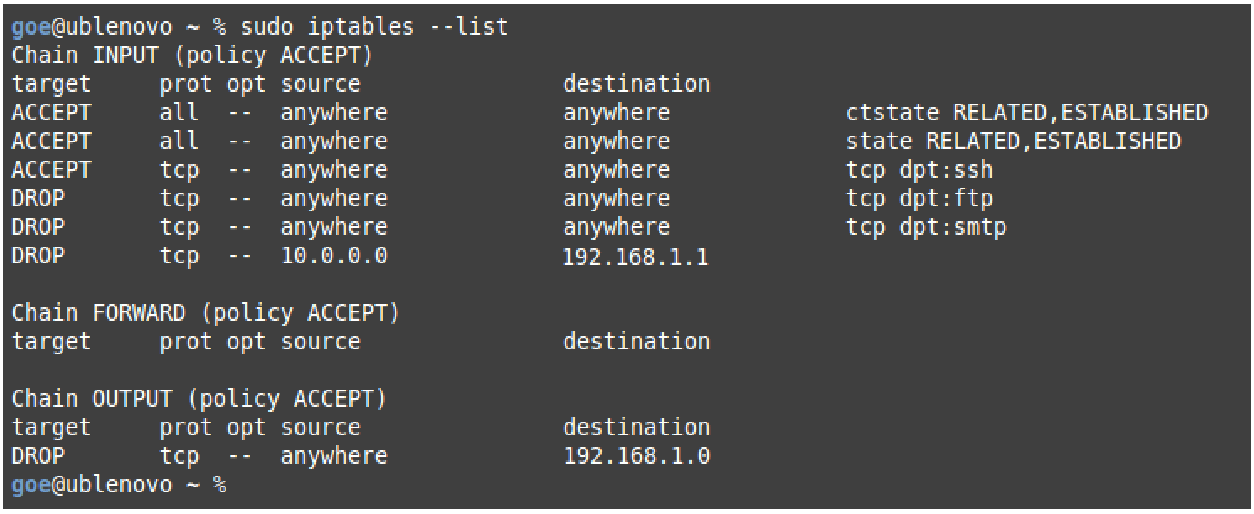
In case an SPA is tracking events that happen in the GUI generically a lot of data is tracked and may actually not be needed. (The generic tracking could either by done due to the enhancement to the JS framework described above, or be based on a generic tracking implementation within the SPA itself.) It might be desired to track only some dialogs on a very detailed level (e.g. a complex dialog, or the five top used dialogs in the system), while other sections (e.g. reference data) only require simple page view tracking and no event tracking at all. Less tracked data would be beneficial by reducing the hardware costs for the tracking servers and by increasing the performance of the database servers when running queries and reports.

The goal of this enhancement would be that no change on the software is required anymore to add or remove tracking capabilities. Instead a rule-set can be configured to define which actions should be tracked and which should be discarded. This would be similar to the list of inclusion and exclusion rules in the configuration of a firewall or file backup software as shown in Figure 9. The configuration itself should be maintained in the WA framework in such a generic way for this to be applicable for any Piwik deployment and not just for the integration with Reval. When configuring the rules, the system should make suggestions based on the already tracked data (e.g. show a list of all existing event categories when using it in a rule). This is already the case for the configuration of segments. The rules should always be checked within the WA framework itself before handling a tracking request. Additionally, it would be beneficial to make this decision as early as possible. If feasible from a performance perspective, the client-side code of Piwik should be enhanced as well to pay respect to the rules. The idea is to load the rules from the WA

[51] See https://github.com/antivanov/jsAspect

[52] See https://github.com/mgechev/aspect.js

server when creating the tracker and to make the decision already within the browser and never contact the WA server for disabled actions.



```
goe@ublenovo ~ % sudo iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
DROP       tcp  --  anywhere             anywhere             tcp dpt:ftp
DROP       tcp  --  anywhere             anywhere             tcp dpt:smtp
DROP       tcp  --  10.0.0.0             192.168.1.1

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             192.168.1.0
goe@ublenovo ~ %
```

Figure 9: Example for a rule-set in the iptables firewall software (Pennington et al., 2014; Russel, 2014)

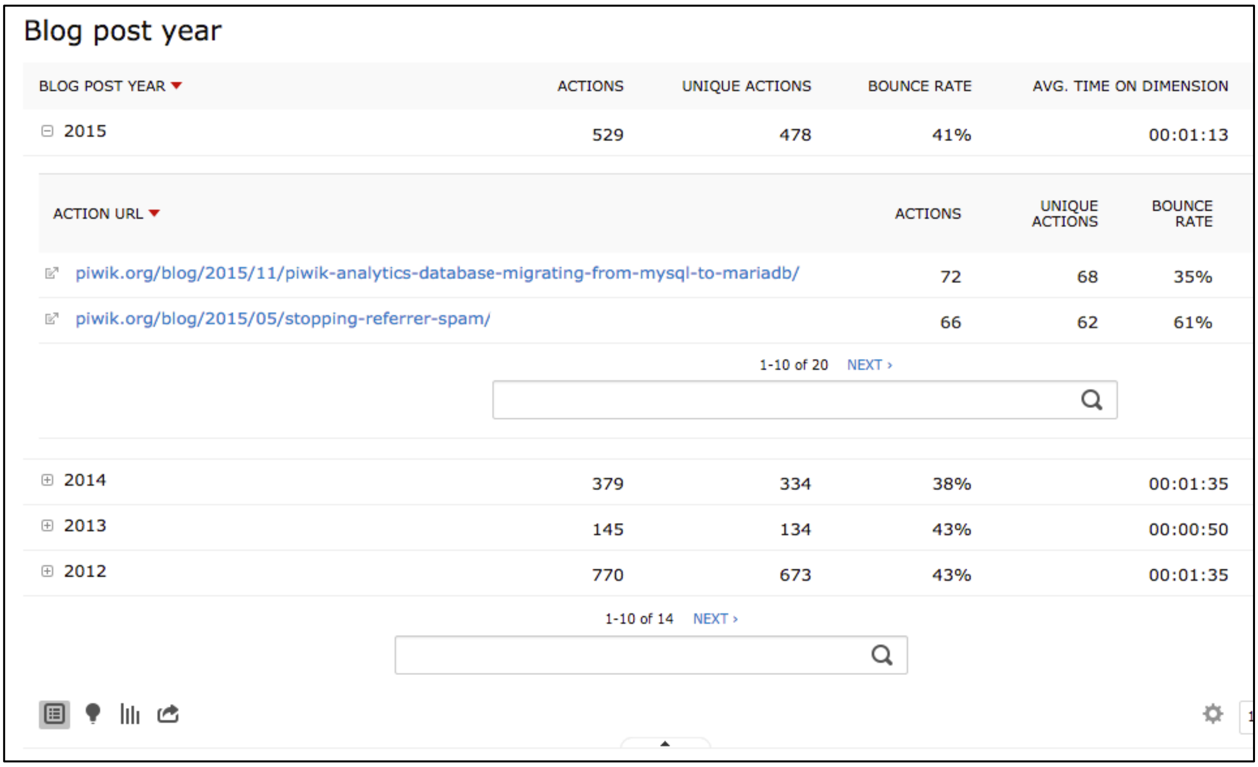
This enhancement would allow a quicker reaction to new analytics requirements, in comparison to making changes in the source code of the SPA for every new tracking necessity. This is especially relevant for software solutions with longer release cycles like six months or a year (e.g. critical systems, which are undergoing extensive security testing before an update).

3.2.1.3 **Configuration of Custom Dimension Values**

Custom dimensions[53] (CD) is a generic feature, which allows to meet the individual tracking needs of any web site or web application. It is possible to record additional attributes in the form of key-value pairs in the scope of a visit or an action (a page view or an event). Piwik supports five dimensions in every scope by default, and it is possible to enhance this number by applying changes to the underlying database schema[54]. The WA framework adds a new report for every defined dimension in the navigation menu and also supports segmentation based on the custom dimension. An example would be to track the calendar year of the creation date for every entry in a blog as a new dimension. A report based on such a dimension can be seen in Figure 10. Examples in regard to an SPA in the scope of a visit would be the used software version or the configured language. An additional meaningful attribute for a page view would be the module or section the page belongs to.

[53] See https://piwik.org/docs/custom-dimensions/

[54] See https://piwik.org/faq/how-to/faq_21121/



Blog post year

BLOG POST YEAR ▼	ACTIONS	UNIQUE ACTIONS	BOUNCE RATE	AVG. TIME ON DIMENSION
⊟ 2015	529	478	41%	00:01:13

ACTION URL ▼	ACTIONS	UNIQUE ACTIONS	BOUNCE RATE
piwik.org/blog/2015/11/piwik-analytics-database-migrating-from-mysql-to-mariadb/	72	68	35%
piwik.org/blog/2015/05/stopping-referrer-spam/	66	62	61%

1-10 of 20 NEXT ›

BLOG POST YEAR ▼	ACTIONS	UNIQUE ACTIONS	BOUNCE RATE	AVG. TIME ON DIMENSION
⊞ 2014	379	334	38%	00:01:35
⊞ 2013	145	134	43%	00:00:50
⊞ 2012	770	673	43%	00:01:35

1-10 of 14 NEXT ›

Figure 10: Report for the custom dimension "Blog post year" (Piwik.org, 2017a)

However, segmentation based on application specific criteria is not good enough for SaaS providers. They also need to segment their clients and users. Clients could be for example segmented by the sales region or their importance (a key account client with licenses for 100 users vs. a small of-the-shelf client) to the business. Whereas the users could be for instance segmented by their role (administrator, back office[55], front office, etc.) or their experience level (novice, advanced, expert). Though, the determination of the values for these types of custom dimensions can be challenging. They may either be evaluated based on an algorithm or must be configured explicitly for every client and user.

Therefore, it is suggested to enhance the underlying SPA to load a list of available custom dimensions from the Piwik server and allow the definition of the value for a specific client or users in the application. This configuration dialog must only be available to the service provider. It should be possible to either specify the value explicitly for every record, or to work with a rule-set (like regular expressions or SQL statements) in order to be as flexible as possible. Such an

[55] See https://www.allaboutcareers.com/careers-advice/looking-for-a-job/front-office-back-office-what-is-the-difference

enhancement would make it possible to control the application-specific segmentation directly from within the application itself, where all the required information is already available.

However, it must not be possible to assign a specific value only to a handful of users or even to a single user. This approach would be in violation with the protection of digital data privacy. A segmentation must never allow to draw conclusions on the behavior of a single user. Thus, the implementation of this enhancement needs to ensure that only custom dimensions for user segmentation are enabled if every resulting segment is big enough.

A nice-to-have would be to allow the complete configuration of custom dimensions from within the SPA. In that case Piwik would need to expose web service methods to create, modify and delete custom dimensions. Furthermore, it would be helpful if an enumeration of valid values for a specific dimension and a default value could be configured in Piwik.

3.2.1.4 **Session Handling**

After the initial integration of Piwik into Reval it turned out that visits are currently not always identified correctly. In general, Piwik has multiple mechanisms to determine whether a tracked request is coming from a known visitor or a new one:[56]

- A user ID (e.g. a user name or an email address) can be set via an API to identify visitors. However, this requires that visitors authenticate themselves before the tracking begins.
- Otherwise first or third-party cookies are used to identify a known visitor.
- As a fallback, a so called "fingerprint" is calculated based on the system configuration (operation system, browser, installed plug-ins, etc.) and the IP address. Piwik uses this config ID to match an incoming request to a known visitor if no userID is set and no cookies are available.

Once a request has been matched with a known visitor the next step is to determine whether the request belongs to an existing visit, or whether to create a new visit in the database. This is currently done based on a timeout window. If no action was sent within a defined period, a new visit gets created. The default setting for the timeout is 30 minutes.[57]

[56] See https://piwik.org/faq/general/faq_21418/

[57] See https://piwik.org/faq/general/faq_36/

The underlying application also typically creates a session for every active user. Due to the way Piwik determines a visit, the sessions in the application do not match up with the tracked visits. Especially if both systems use a different setting for the session timeout. Over the past years multiple requests were made by the community to solve this issue.[58] [59] [60] [61] [62] [63] One issue at hand is, that if a user manually logs out and logs in within 30 minutes it is tracked as one visit, although the underlying SPA creates two sessions. Another problem with the current approach appears whenever a user accesses the same application via different devices (e.g. from a smartphone and a PC simultaneously). In such a case, actions from both devices get merged into one visit in the WA platform. The resulting reports are not correct anymore, as for instance the sequence of page views gets mixed up.

Therefore, it is suggested to modify the tracking API. It should accept an external visit ID as part of the tracking request. If the external visit ID is set, the decision whether a new visit is happening should be purely based on this ID and all other mechanisms should be disabled. Such an enhancement would improve the integration of Piwik with web applications.

3.2.2 **Metrics Gaps**

The following section presents four gaps in terms of metrics in the Piwik WA framework. A suggestion of how to resolve the gaps is provided as well. One focus in this chapter is on missing application-specific measurements. This is especially important as the metrics might align with the strategic goals of the application, in which case they would be deemed KPIs. Furthermore, suggestions on how to improve the configuration of goals to better meet the needs of SPAs are examined as well.

3.2.2.1 **Custom Metrics**

Piwik already supports tracking the generation time of pages with each page view. Based on that the average generation time is shown in page related reports. However, it would also be interesting to track the size of the payload for page views. The payload is another indicator for performance issues. It can help to put the generation time in the right perspective. For instance,

[58] See https://forum.piwik.org/t/vorhandene-session-id-zusammenhaengender-requests/21613

[59] See https://forum.piwik.org/t/usernamen-und-sessions/8819/3

[60]See https://forum.piwik.org/t/send-session-information-along-with-user-id-to-get-session-analytics/24896

[61] See https://www.quora.com/unanswered/How-do-I-get-the-browser-session-ID-SID-to-the-PIWIK-database

[62] See https://github.com/piwik/piwik/issues/10112

[63] See https://github.com/piwik/piwik-sdk-ios/issues/148

a generation time of three seconds for a huge payload is to be expected, whereas the same period for a small payload might indicate a problem worth investigating. Furthermore, it should be possible to track the generation time and the payload size for events as well. For example, when clicking the reload button for a grid it should be tracked how long the request to the server took and how much data was transported in order to refresh the grid.

In general, the tracking of additional attributes is already doable with the help of custom dimensions (see Section 3.2.1.3). However, custom dimensions are expecting string values and apply segmentation directly based on the received values (e.g. software version, user role). Adding new metrics is already possible via creating new plug-ins (e.g. the social media plug-in), but a generic solution would be required.[64] [65] Piwik needs to be enhanced to support custom metrics to better satisfy the needs of SPAs. Goggle Analytics does already support it.[66] [67]

The following approach should be taken into consideration to resolve this gap. A new plug-in for "custom metrics" (similar to the "custom dimensions" plug-in) should be introduced to support the tracking of application-specific metric values. The raw values should be stored with every event, but for reporting purposes it would be necessary to create groups. This means to convert the original metric scale to an ordinal scale by applying data binning[68]. The dialog for the configuration of the custom metrics needs to allow the definition of buckets. Furthermore, it should be possible to modify them at any time. New reports and segments should be added automatically for every new custom metric (as is the case for custom dimensions).

3.2.2.2 **Error Tracking**

Informing a user about problems via an error message is a common approach in applications. This is clearly something that needs to be tracked in an SPA. It is already possible to track error popups either as page views for the "error" dialog or as events for the "error" category. The latter approach even allows to link the tracked error directly to a specific dialog. However, it would be helpful for data analysts to track more attributes with every error and to run more advanced reports on it. Adding more attributes to an error message like the type (exception, critical, validation, warning, message) or the internal error ID would already be doable via custom dimensions. Although, it would not be very efficient to use two slots in the action scope only for error events.

[64] See https://forum.piwik.org/t/custom-track-metrics-and-reports-is-possible/4250

[65] See https://forum.piwik.org/t/custom-metrics/16626

[66] See https://developers.google.com/analytics/devguides/collection/analyticsjs/custom-dims-mets

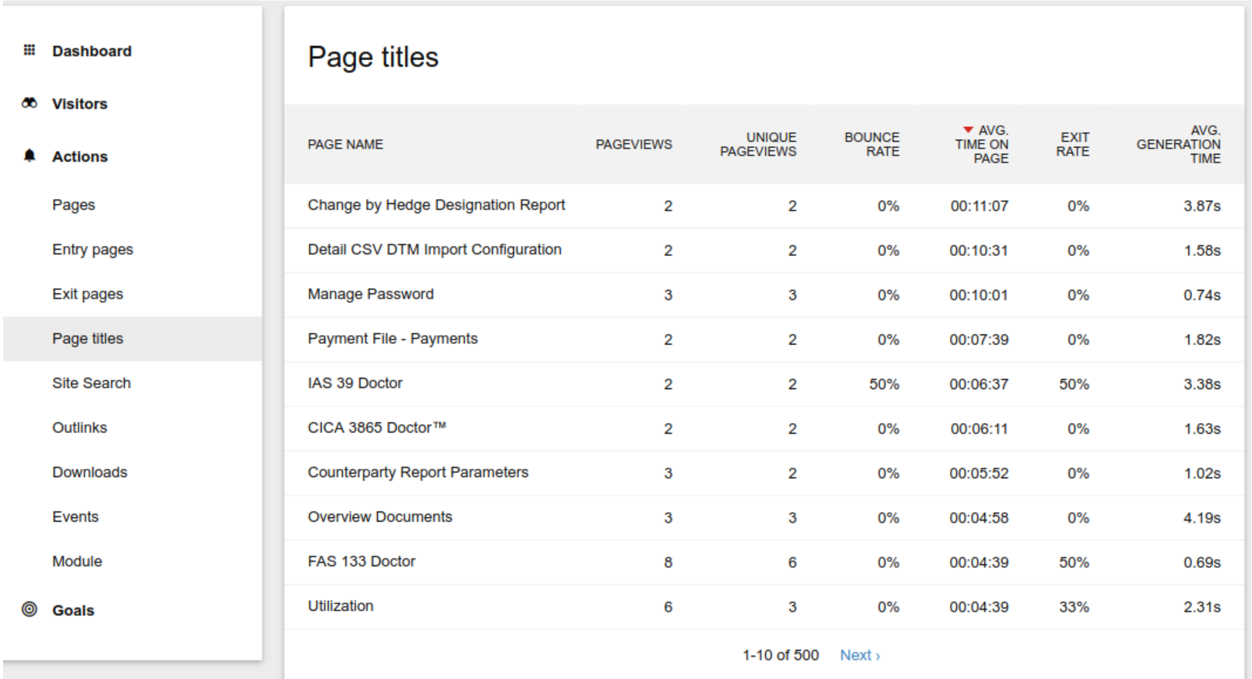
[67] See https://support.google.com/analytics/answer/2709828

[68] See https://www.ibm.com/support/knowledgecenter/en/SSLVMB_24.0.0/spss/base/idh_webhelp_scatter_options_palette.html

They would be empty for any other event and all page views. Furthermore, it would make sense to have special reports and metrics based on the tracked errors. One measurement for instance that would make sense is the ratio between the number of tracked errors and the number of page views per error type. Such a report would put high numbers of errors in the right perspective. Maybe the errors were all caused by a heavy user who was logged in for several hours and most of the errors were just validation errors, which are to be expected. It would also be helpful to track whether an error message was followed by the abandonment of the current task (e.g. entering a new trade), or whether the user was able to resolve the problem and complete the intended task.

3.2.2.3 Dialog Tracking

In an SPA that offers a GUI with a multiple document interface – where multiple dialogs contained in a window or tab can be open at the same time – there is no strict sequence of pages (or dialogs) like on a web site. The approach taken in the initial integration of Piwik with Reval was to track the opening of a new dialog as a new page view. However, the original dialog could still be open and in case of using window containers, both dialogs could even be visible at the same time. Thus, the already available metric “AVG. TIME ON PAGE” (see Figure 11) in all page related reports can just give a rough estimate on how long the user was probably focused on the specific dialog. It would also be good to know how long the dialog was actually open.



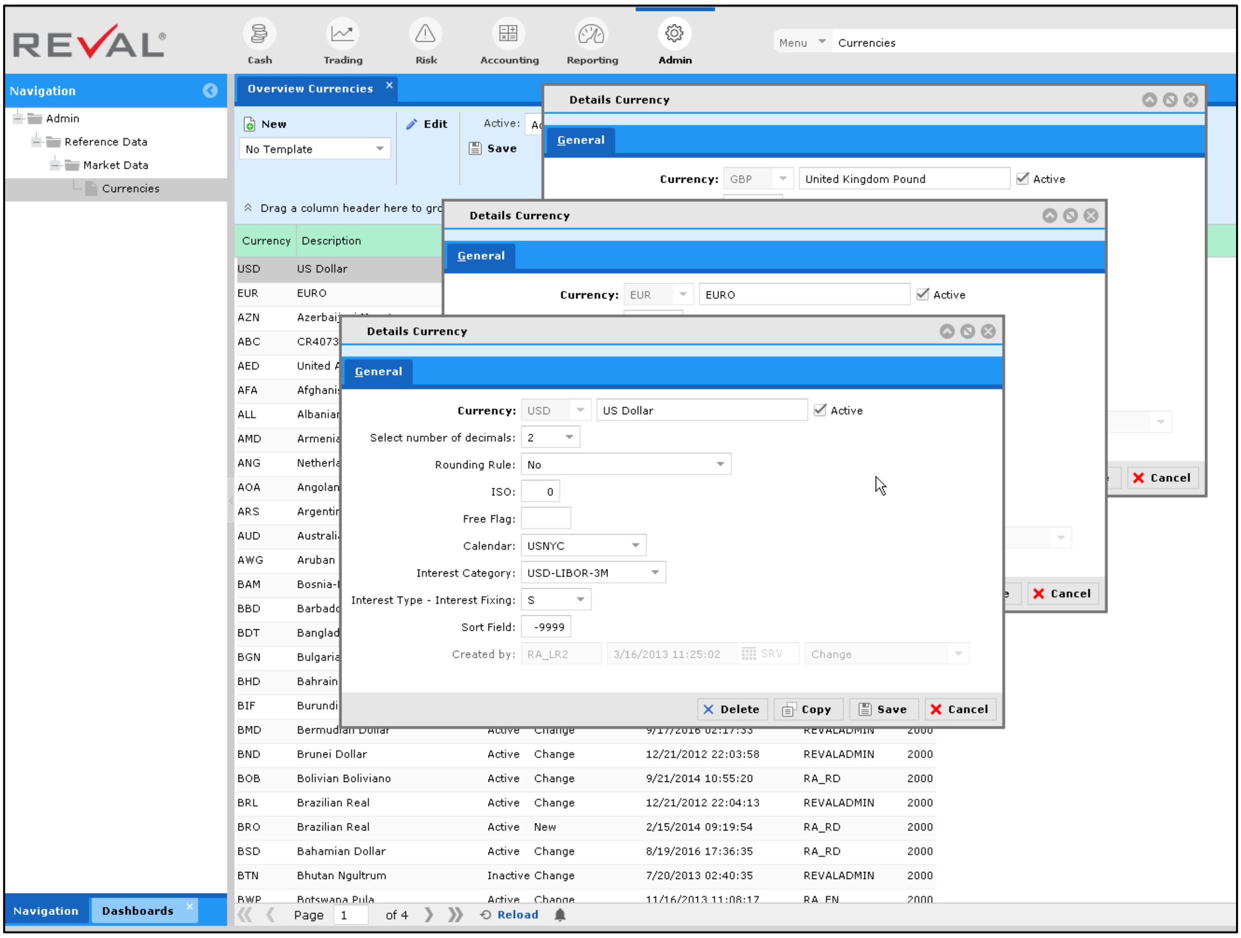
Page titles

PAGE NAME	PAGEVIEWS	UNIQUE PAGEVIEWS	BOUNCE RATE	▼ AVG. TIME ON PAGE	EXIT RATE	AVG. GENERATION TIME
Change by Hedge Designation Report	2	2	0%	00:11:07	0%	3.87s
Detail CSV DTM Import Configuration	2	2	0%	00:10:31	0%	1.58s
Manage Password	3	3	0%	00:10:01	0%	0.74s
Payment File - Payments	2	2	0%	00:07:39	0%	1.82s
IAS 39 Doctor	2	2	50%	00:06:37	50%	3.38s
CICA 3865 Doctor™	2	2	0%	00:06:11	0%	1.63s
Counterparty Report Parameters	3	2	0%	00:05:52	0%	1.02s
Overview Documents	3	3	0%	00:04:58	0%	4.19s
FAS 133 Doctor	8	6	0%	00:04:39	50%	0.69s
Utilization	6	3	0%	00:04:39	33%	2.31s

1-10 of 500 Next ›

Figure 11: Metric “Avg. time on page” in the “Page Titles” report (Piwik.org, 2017a)

A new mechanism would be needed in Piwik where tracking a new “page view” does not automatically end the succeeding page view. Instead special “openDialog” and “closeDialog” tracking actions should be added to the API to better support SPAs. Furthermore, a unique dialog-instance ID would be needed to differentiate multiple simultaneous instances of the same dialog from each other. This is necessary for example when opening multiple currency records in the Reval application as can be seen in Figure 12. New metrics and reports based on dialogs should be added as well. Such a report could show the three following measurements: The total time the dialog was open, the number of dialog views and the number of unique dialog views (in reference to page views and unique page views) should be calculated.



Details Currency

Currency: GBP United Kingdom Pound ☑ Active

Details Currency

Currency: EUR EURO ☑ Active

Details Currency

Currency: USD US Dollar ☑ Active

Select number of decimals: 2

Rounding Rule: No

ISO: 0

Free Flag:

Calendar: USNYC

Interest Category: USD-LIBOR-3M

Interest Type - Interest Fixing: S

Sort Field: -9999

Created by: RA_LR2 3/16/2013 11:25:02 SRV Change

Delete Copy Save Cancel

Currency	Description					
USD	US Dollar					
EUR	EURO					
AZN	Azerbaij					
ABC	CR4073					
AED	United A					
AFA	Afghanis					
ALL	Albaniar					
AMD	Armenia					
ANG	Netherla					
AOA	Angolan					
ARS	Argentir					
AUD	Australi					
AWG	Aruban					
BAM	Bosnia-I					
BBD	Barbado					
BDT	Banglad					
BGN	Bulgaria					
BHD	Bahrain					
BIF	Burundi					
BMD	Bermudian Dollar	Active	Change	9/17/2016 02:17:33	REVALADMIN	2000
BND	Brunei Dollar	Active	Change	12/21/2012 22:03:58	REVALADMIN	2000
BOB	Bolivian Boliviano	Active	Change	9/21/2014 10:55:20	RA_RD	2000
BRL	Brazilian Real	Active	Change	12/21/2012 22:04:13	REVALADMIN	2000
BRO	Brazilian Real	Active	New	2/15/2014 09:19:54	RA_RD	2000
BSD	Bahamian Dollar	Active	Change	8/19/2016 17:36:35	RA_RD	2000
BTN	Bhutan Ngultrum	Inactive	Change	7/20/2013 02:40:35	REVALADMIN	2000
BWP	Botswana Pula	Active	Change	11/16/2013 11:08:17	RA_EN	2000

Page 1 of 4 Reload

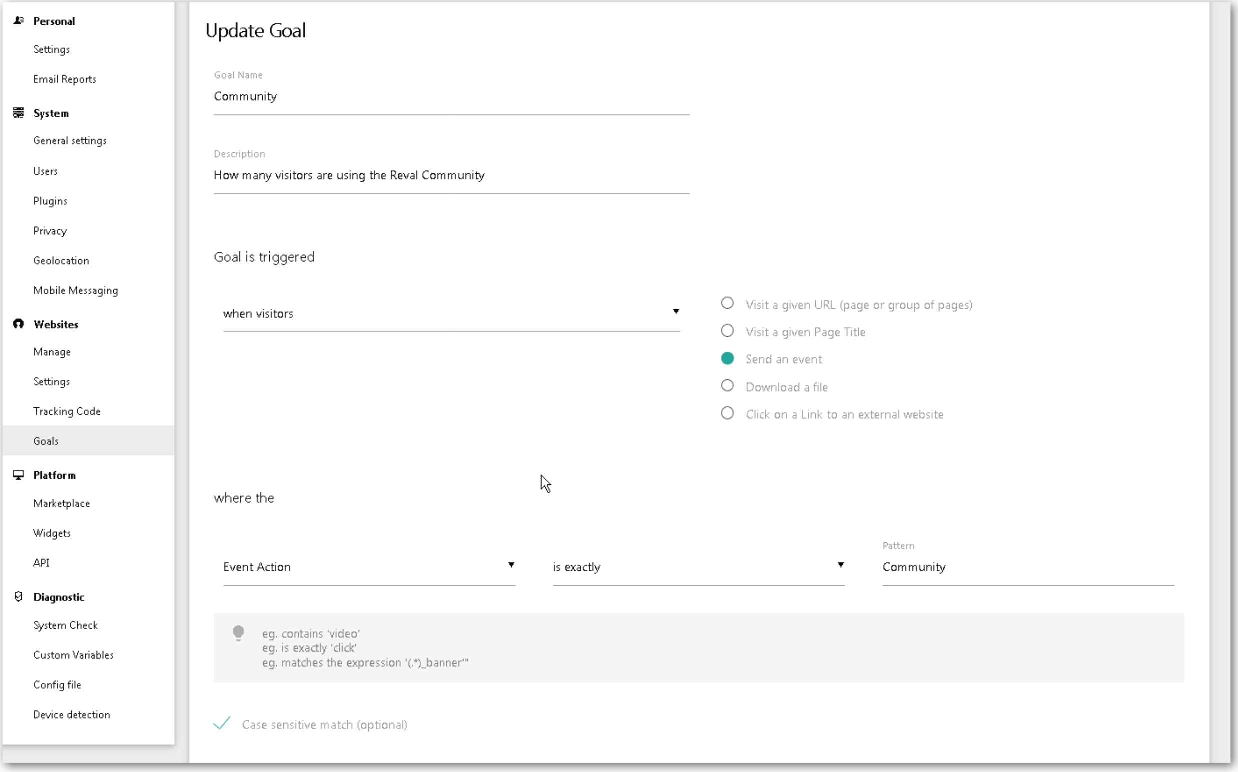
Figure 12: Opening multiple currency records at the same time in Reval (Reval Inc., 2017a)

3.2.2.4 Advanced Goal Configuration

Piwik currently supports five different ways to track the completion of a goal as can be seen in Figure 13:

- Access of a page or a group of pages via a given URL
- Access of a page via a given page title
- Triggering of an event

- Download of a file
- Click on an outgoing link



Update Goal

Goal Name
Community

Description
How many visitors are using the Reval Community

Goal is triggered

when visitors ▾

○ Visit a given URL (page or group of pages)
○ Visit a given Page Title
● Send an event
○ Download a file
○ Click on a Link to an external website

where the

Event Action ▾ | is exactly ▾ | Pattern: Community

eg. contains 'video'
eg. is exactly 'click'
eg. matches the expression '(.*)_banner'"

✓ Case sensitive match (optional)

Figure 13: Configuring a goal in Piwik (Piwik.org, 2017a)

However, a lot of the actions tracked in an SPA are events (interactions with buttons, checkboxes, dropdowns, grids, etc.) and not page views. It is crucial to allow the combination of multiple rules that need to be matched to trigger a goal. For instance, the usage of the fictive release highlight “garbage collector” should be tracked. It is supposed to be enabled via a checkbox in the “User Configuration” dialog. In this case, the goal should only be triggered if the following four conditions are met:

- Event Category equals “UI_CHECKBOX”
- Event Name equals “User Configuration”
- Event Action equals “Garbage Collector”
- Event Value equals “Ticked”

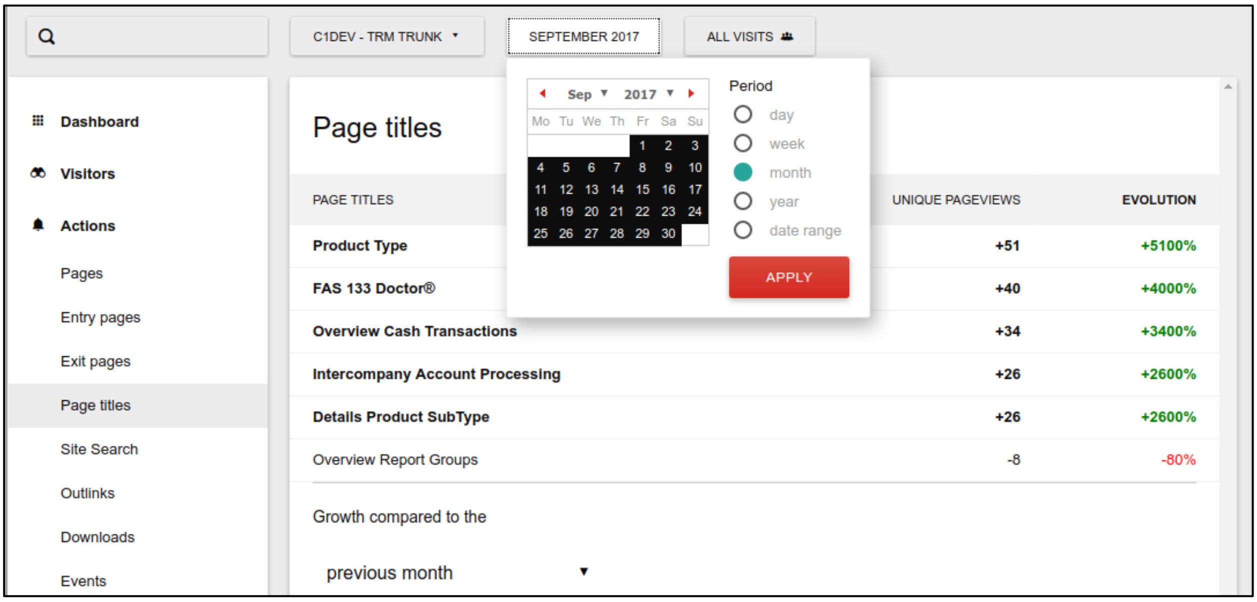
The goal configuration capabilities of Piwik need to be enhanced to be more flexible and meet the needs of tracking an SPA. Multiple rules should be combinable similar to the definition of a segment.

3.2.3 Reporting Gaps

This section covers gaps in the reporting capabilities of the Piwik WA framework in regard to single page applications. Three missing reports with special considerations for SPAs were identified. Furthermore, two gaps in existing reports were found. In addition to describing each lack of functionality a suggestion for a solution is presented as well.

3.2.3.1 Named Date Ranges

Piwik also provides, besides the table-based default report view, other presentations of data such as different chart types and the insights report. This report allows for the comparison of data across two periods of time. It helps to find changes in the user behavior or in the traffic in general. Figure 14 shows the six pages that experienced an enormous gain or loss – called the "movers" – in number of unique page views between August and September 2017. Although it is already possible to define a custom date range[69] in the period selection control in the top, it is not possible to save the data range under a name, or compare two freely configured date ranges with each other. However, this feature is required for using the insights view with data coming from an SPA. Custom defined data ranges are for instance needed for the period during which a specific software version was deployed. Such ranges would be created for the life span of every release, so that the tracked data of two releases can easily be compared to each other.



Page titles

Period
○ day
○ week
● month
○ year
○ date range

PAGE TITLES	UNIQUE PAGEVIEWS	EVOLUTION
Product Type	+51	+5100%
FAS 133 Doctor®	+40	+4000%
Overview Cash Transactions	+34	+3400%
Intercompany Account Processing	+26	+2600%
Details Product SubType	+26	+2600%
Overview Report Groups	-8	-80%

Growth compared to the

previous month ▼

Figure 14: Insights view on "Page Titles" report (Piwik.org, 2017a)

[69] See https://developer.piwik.org/api-reference/Piwik/Period/Range

3.2.3.2 **Click Path Report**

A click path report visualizes the route taken through a web site. It is sometimes also called user flow report or user journey map.[70] The benefits of such a report are that it is a straightforward way to see which common paths exist on a web site and how many visitors are taking each route. It is much easier to comprehend compared to a purely table-based report. Piwik already offers such a feature in the form of a premium plug-in. However, besides the fact that it is not freely available, like the rest of the application, it also has some gaps. The underlying data is prepared in a way that an index is stored with every page view throughout the visit, which is called the interaction number. The click path always starts with entry pages (the first interactions) and shows how the users progress through the web sites starting from there. It is not possible to define a specific page of interest and show the flow to and from this page.

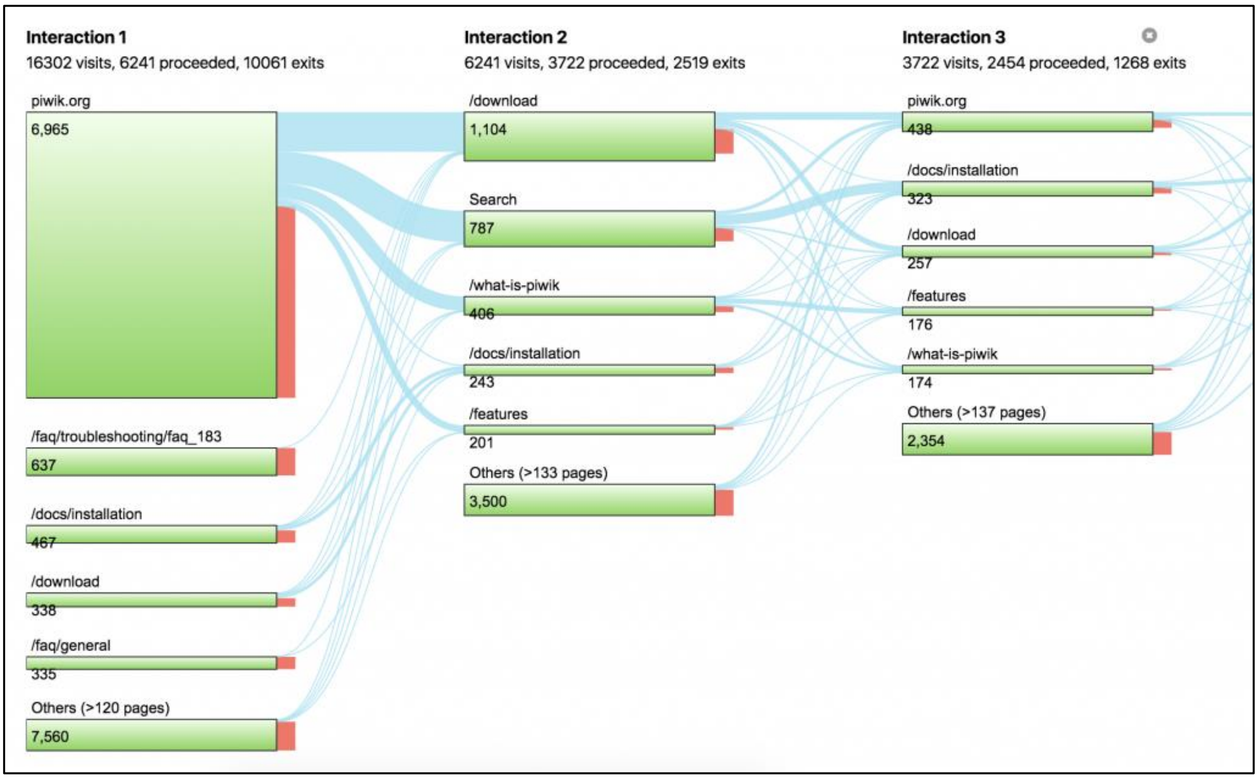
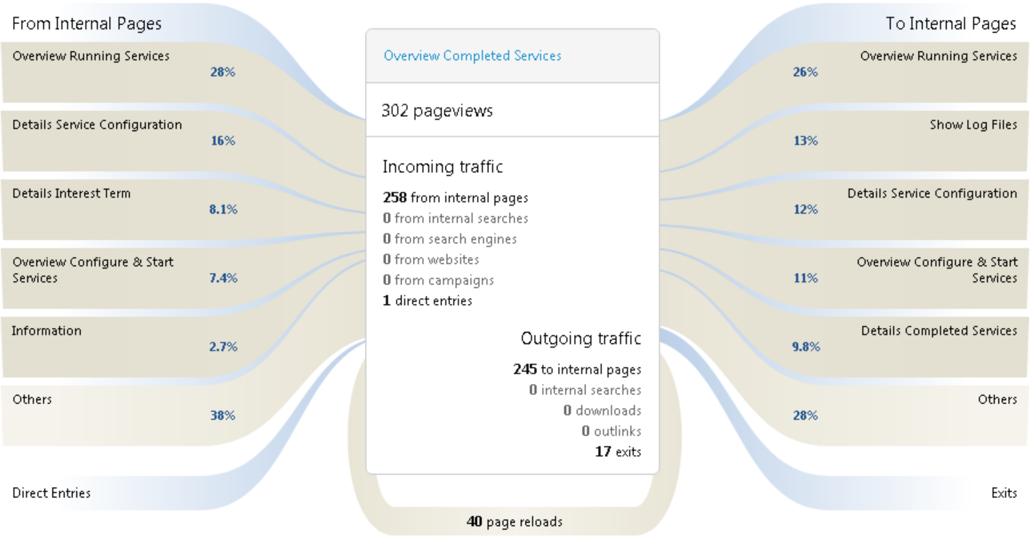


Figure 15: Click path report in Piwik (Piwik.org, 2017b)

Piwik offers another related report as a free plug-in. The transition report visualizes all page views that happened before and after the defined page (see Figure 16). This is basically a user journey map with the limitation to see only one level in each direction. The desired report should work

[70] See https://support.google.com/analytics/answer/2519986?hl=en

similar, but support the configuration of how many steps it should visualize in each direction. A Sankey diagram, as can be seen in Figure 17, could be used to visualize the information appropriately. The possibility to follow the main routes through the software is critical for an SPA in order to identify problems with the user flow.



From Internal Pages

Overview Running Services 28%

Details Service Configuration 16%

Details Interest Term 8.1%

Overview Configure & Start Services 7.4%

Information 2.7%

Others 38%

Direct Entries

Overview Completed Services

302 pageviews

Incoming traffic

258 from internal pages
0 from internal searches
0 from search engines
0 from websites
0 from campaigns
1 direct entries

Outgoing traffic

245 to internal pages
0 internal searches
0 downloads
0 outlinks
17 exits

40 page reloads

To Internal Pages

26% Overview Running Services

13% Show Log Files

12% Details Service Configuration

11% Overview Configure & Start Services

9.8% Details Completed Services

28% Others

Exits

Figure 16: The transition report for the dialog "Completed Services" (Piwik.org, 2017a)

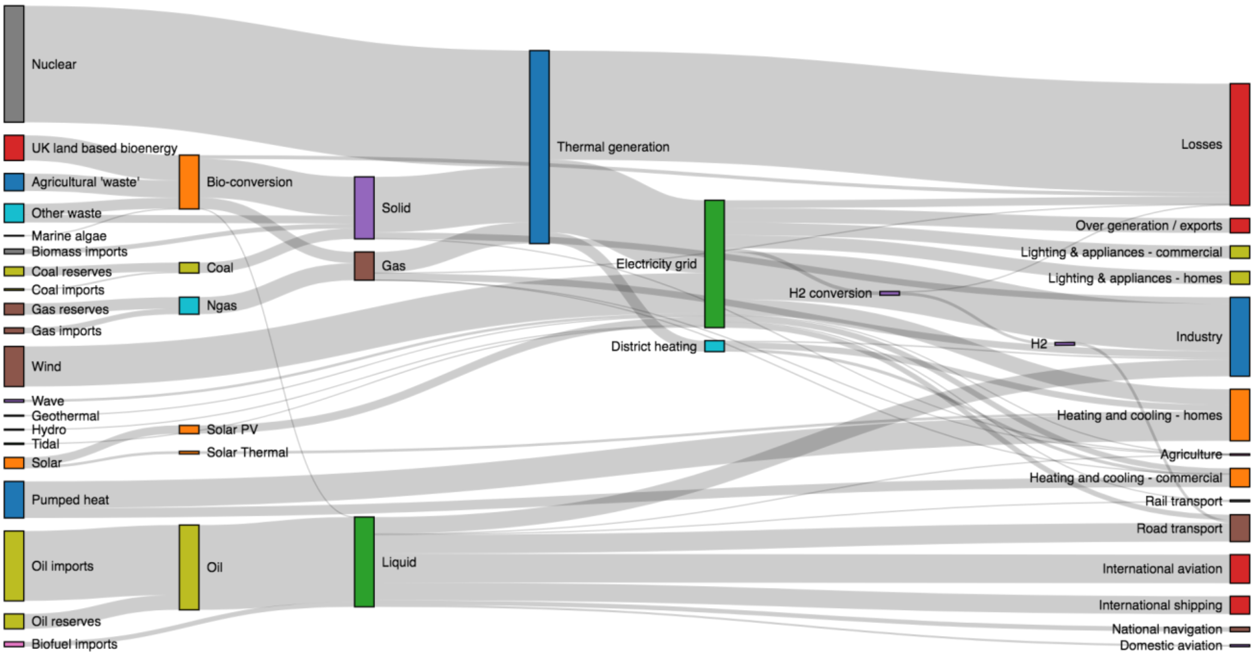
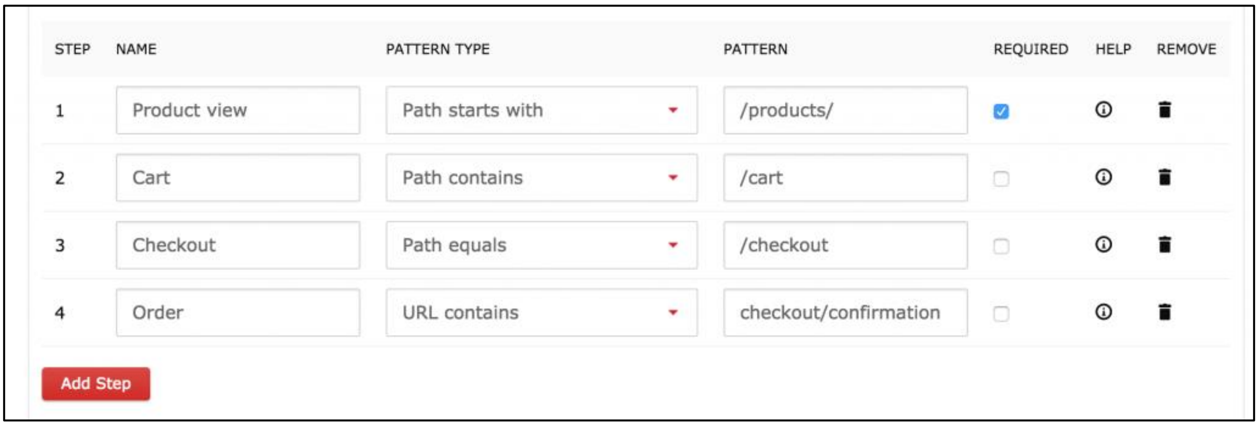


Figure 17: An example for a Sankey diagram (Bostock, 2017)

3.2.3.3 Funnel Report

The funnel report is another report that is only available as part of a premium plug-in.[71] It allows to define multiple steps that can happen before triggering a goal and thus, gives more insight in where visitors are abandoning the predefined path to a conversion. Funnels should be created based on the understandings gained from the click path report: The two or three most common pages in the path to a goal should be configured as the individual steps of the funnel (Kamiński, 2017). The defined funnel can be used as a base line for future improvements to the web site or the application. For instance, in case the web designer tries to raise the completion rate by redesigning the user interface, the funnel report can be used to verify whether an actual improvement happened and in which step it occurred.



STEP	NAME	PATTERN TYPE	PATTERN	REQUIRED	HELP	REMOVE
1	Product view	Path starts with	/products/	☑	ⓘ	🗑
2	Cart	Path contains	/cart	□	ⓘ	🗑
3	Checkout	Path equals	/checkout	□	ⓘ	🗑
4	Order	URL contains	checkout/confirmation	□	ⓘ	🗑

Figure 18: Configuration of a funnel in Piwik (Piwik.org, 2017a)

The funnel plug-in provided by the Piwik team does support optional steps, but is missing the functionality to define alternative steps. This would be helpful for an SPA for which multiple ways often exist to reach the same goal. It would be beneficial to learn how many users are taking each of the possible routes, or whether the abandonment rate varies for the different paths.

One example for a funnel in a single page application is to track the creation of a new record (e.g. a trade). The first step in the funnel is to open the dialog for entering a new record. The goal – the final step – is that the user saves the record successfully. However, there is no specific order in which a user navigates through the dialog or in which she/he is accessing the UI controls. Thus, a beneficial enhancement for SPAs would be to just be able to define a set of steps without the need to settle on a specific order. A new report similar to the user journey should be available to

[71] See https://piwik.org/docs/funnels/

visualize the various possible routes to the goal and how many users were taking each path within the dialog. Although this suggested report has parallels with the click path report described in the previous chapter, it differs in terms of the underlying data. This report would be based on events instead of page views and only focus on a single dialog. Furthermore, it would not show all events that can be triggered within a dialog, which can be quite many, but only those that were defined in the funnel. Thus, it is to be categorized as a special version of a funnel report.

3.2.3.4 **Overlay Report for Dialogs**

Piwik provides a feature that opens the underlying web page in a new browser window and shows the tracked metrics directly on top of the page. This functionality is called "Overlay" (or "Visitors Click Map" or "Click Density Map" in other WA frameworks) and it is accessible from the "Pages" report.[72] Figure 19 shows the original landing page of www.virtual-drums.com, which is one of the web sites tracked by the Piwik team for demonstration purposes[73]. The overlay feature on top of it can be seen in Figure 20. Every tracked link, which leads to a new page view, is marked with a balloon that shows the percentage of clicks that led to the respective successor page view. Three links are marked with 50% as all of them lead to the landing page itself. The additional frame on the left side allows to change the selected period and gives an overview of the main metrics.

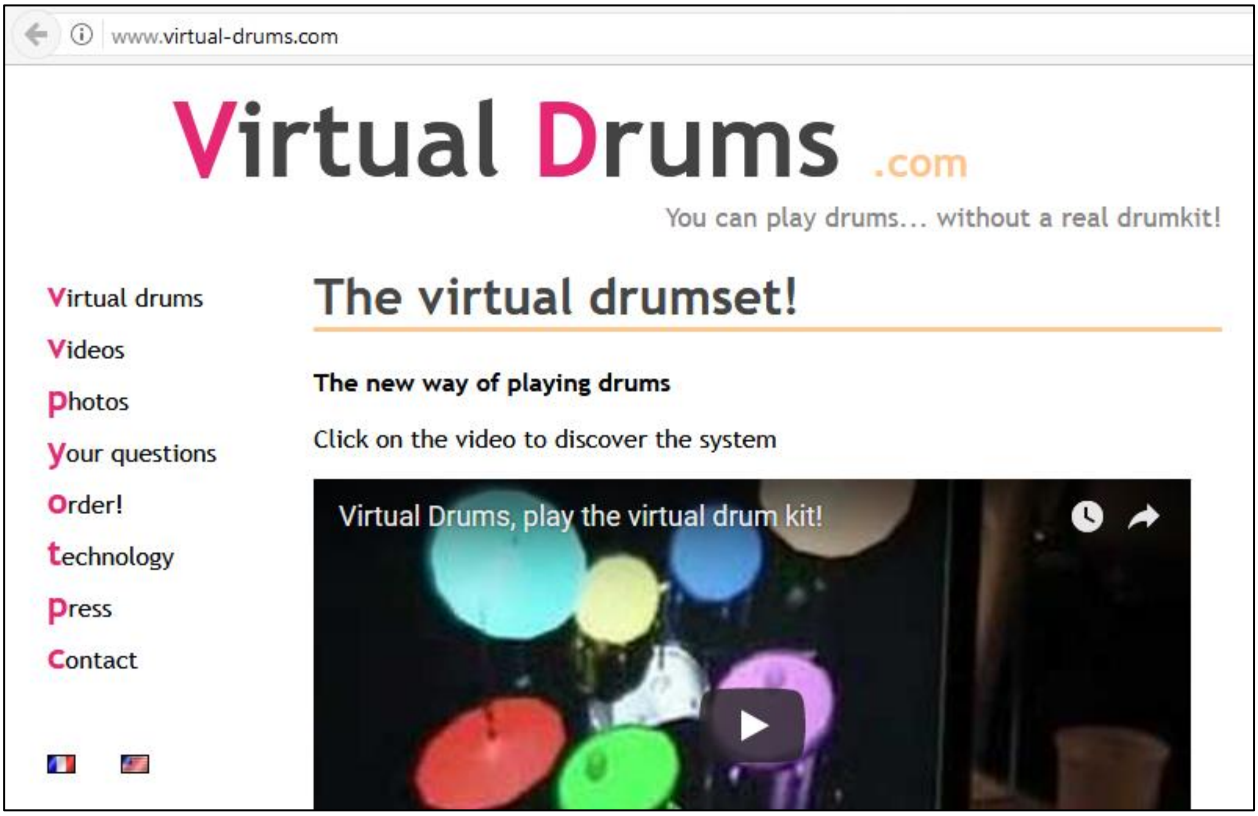
This overlay functionality can currently only be used for pages that are accessible via a URL. Dialogs in a multi-window SPA are not supported. A single page application would need to be enhanced so that the layout of any dialog can be shown via a unique URL. This URL would need to work without a valid session and must not show any user data. However, if there are optional UI controls defined in the layout of the dialog (e.g. a dropdown that only shows up if the user has a special license granted), such fields would need to be displayed in this view. It is required that the dialog would be rendered with all available UI controls in this special mode. Furthermore, the overlay plug-in would need to be enhanced to work with buttons instead of hyperlinks.

Another improvement required for the needs of an SPA is the possibility to visualize tracked events in the overlay view. Most tracking within a dialog is done by tracking events (e.g. how many visitors used a specific dropdown or radio button control). The SPA and the WA framework would need to be enhanced to send an internal ID with every event (potentially as custom dimension) that uniquely identifies an UI control. This ID must stay the same across different

[72] See https://piwik.org/docs/page-overlay/

[73] See https://demo.piwik.org

software versions. The overlay view should support switching between visualizing page views and events.



www.virtual-drums.com

Virtual Drums .com

You can play drums... without a real drumkit!

Virtual drums
Videos
Photos
Your questions
Order!
technology
Press
Contact

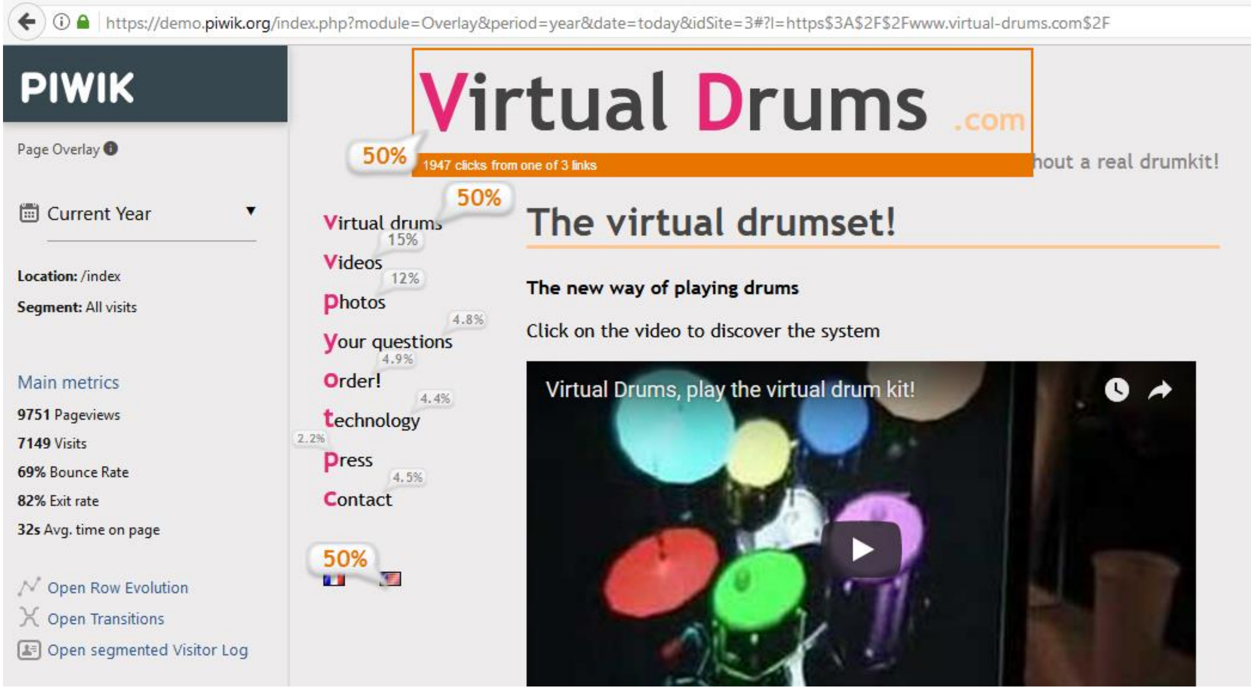
The virtual drumset!

The new way of playing drums

Click on the video to discover the system

Virtual Drums, play the virtual drum kit!

Figure 19: The unmodified landing page of virtualdrums.com



https://demo.piwik.org/index.php?module=Overlay&period=year&date=today&idSite=3#?l=https$3A$2F$2Fwww.virtual-drums.com$2F

PIWIK

Page Overlay

Current Year

Location: /index

Segment: All visits

Main metrics

9751 Pageviews

7149 Visits

69% Bounce Rate

82% Exit rate

32s Avg. time on page

Open Row Evolution

Open Transitions

Open segmented Visitor Log

Virtual Drums .com

1947 clicks from one of 3 links

hout a real drumkit!

Virtual drums
Videos
Photos
Your questions
Order!
technology
Press
Contact

The virtual drumset!

The new way of playing drums

Click on the video to discover the system

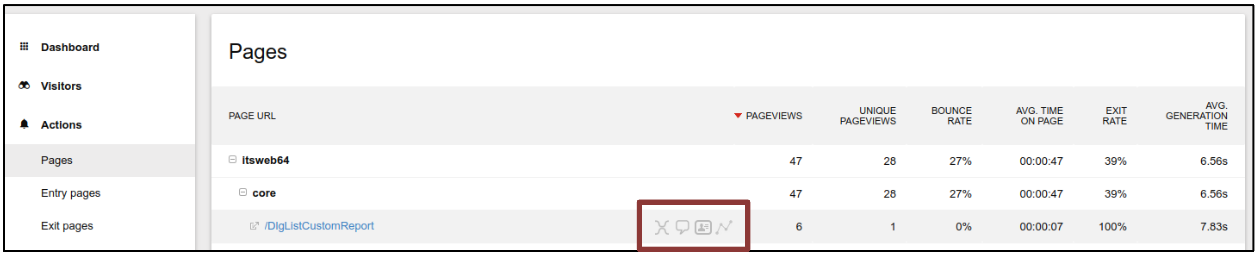
Virtual Drums, play the virtual drum kit!

Figure 20: The landing page of virtualdrums.com with analytics information on top of it (Piwik.org, 2017a)

3.2.3.5 **Enhance Page Titles Report**

Although the “Pages” report and the “Page Titles” report are based on the same data, they do not offer the same functionality as can be seen in Figure 21 and Figure 22 below. When reporting on page titles, it is not possible to navigate directly to the underlying page on the tracked web site. Furthermore, one cannot access the overlay feature, where Piwik opens the underlying page in a new browser window and presents the analytics data in a layer on top of the original web page. The third limitation in functionality is a missing button to open the visitor log segmented by the chosen page. This popup window shows only visitors which have accessed the selected page.

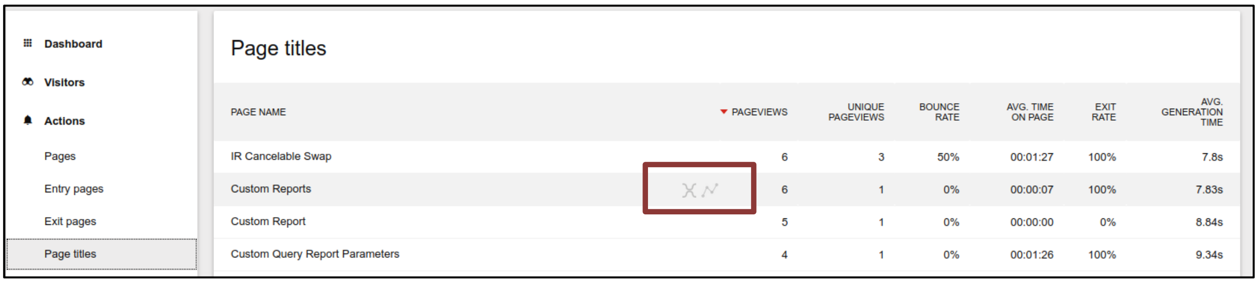
A multi-window SPA does not use URLs. Instead it uses unique IDs for tracking the dialogs as pages. As these IDs are not meaningful to business analysts and stakeholders, the “Page Titles” report is much more convenient for them. Thus, it needs to offer the same functionality as the “Pages” report.



Pages

PAGE URL	▼ PAGEVIEWS	UNIQUE PAGEVIEWS	BOUNCE RATE	AVG. TIME ON PAGE	EXIT RATE	AVG. GENERATION TIME
⊟ itsweb64	47	28	27%	00:00:47	39%	6.56s
⊟ core	47	28	27%	00:00:47	39%	6.56s
/DlgListCustomReport	6	1	0%	00:00:07	100%	7.83s

Figure 21: Available actions on a row in the pages report (Piwik.org, 2017a)



Page titles

PAGE NAME	▼ PAGEVIEWS	UNIQUE PAGEVIEWS	BOUNCE RATE	AVG. TIME ON PAGE	EXIT RATE	AVG. GENERATION TIME
IR Cancelable Swap	6	3	50%	00:01:27	100%	7.8s
Custom Reports	6	1	0%	00:00:07	100%	7.83s
Custom Report	5	1	0%	00:00:00	0%	8.84s
Custom Query Report Parameters	4	1	0%	00:01:26	100%	9.34s

Figure 22: Available actions on a row in the page titles report (Piwik.org, 2017a)

3.2.4 Summary

A list of all identified gaps can be seen in Table 3. The summary shows the category, the title, a brief description of the suggested enhancement and in which scope the enhancement would need to be made.

Category	Gap	Description	Scope
Integration	Generic Tracking	Enhance the Ext JS framework to support page view and event tracking generically.	Ext JS
Integration	Configuration of Generic Tracking	Support enabling and disabling of the tracking for certain pages or events via a configuration dialog.	WA
Integration	Configuration of Custom Dimension Values	Allow determination of user- and client-specific attributes via a configuration dialog.	SPA
Integration	Session Handling	External sessions should match up with visits.	SPA, WA
Metrics	Custom Metrics	Support the definition of application-specific measurements.	WA
Metrics	Error Tracking	Tracking of errors with additional attributes and providing new reports based on it.	WA
Metrics	Dialog Tracking	Tracking of dialog views (in reference to page views) and providing new reports based on it.	WA
Metrics	Advanced Goal Configuration	Allow the combination of multiple rules to trigger a goal.	WA
Reporting	Named Date Ranges	Support of saving and selecting of custom data ranges under a given name.	WA
Reporting	Click Path Report	Provide a report that shows all page views before and after a defined page.	WA
Reporting	Funnel Report	Allow the configuration of funnels with alternative steps or an undefined sequence of steps.	WA
Reporting	Overlay Report for Dialogs	Enhance the SPA to show a read-only view of any dialog. Modify the overlay report to visualize events.	WA, SPA
Reporting	Enhance Page Titles Report	The “Page Titles” report should offer the same functionality as the “Pages” report.	WA

Table 3: Summary of gaps in WA frameworks in regard to SPAs

3.3 Implemented Enhancement

At first this chapter covers which of the 13 suggested enhancements was chosen to be implemented as part of this thesis. Subsequently the various phases in the implementation process are described: After analyzing the status quo, design decisions were made, and the artifact was implemented based on them. The last two phases deal with the integration of the artifact with the SPA and the internal rollout to the product development division at Reval Inc. with the goal to allow answering the research question.

3.3.1 Selection

In order to determine which of the suggested enhancements should be implemented as part of this thesis, employees at Reval Inc. were asked to score the proposed enhancements. Therefore, 13 members of the product development division were invited to a meeting. They received a document describing all enhancements (Chapter 2.4 of this work) upfront. Those 13 employees were selected as they are members of two special working groups at Reval Inc.: One group has a special focus on the architecture of the application and the other group deals with user experience and usability topics. Both groups have an explicit interest in integrating web analytics with Reval and are professionals in the field of developing web applications. Nine out of the 13 invitees joined the meeting, where all gaps and the suggested enhancements were briefly presented and any open questions were answered.

At the end of the meeting, every participant received a questionnaire (in form of a printout) and was asked to score each enhancement based on the impact the modification would have on the usefulness of the web analytics solution. The score voting[74] method was used to evaluate the suggestions: The participants were asked to score each item independently with a value out of -2, -1, 0, 1 and 2. The attendees were instructed to give zero points for any items, for which they had neither a positive nor a negative preference. Based on the nine responses the possible range of the score for each item lies between -18 and 18 points. The result of the voting can be seen in the bar chart in Figure 23. The enhancement with supposedly the most positive impact on the usefulness of the web analytics solution is “Custom Metrics”, closely followed by “Generic Tracking”. Two other suggestions (“Error Tracking” and “Dialog Tracking”) received very high scores as well. It was unexpected that three proposed enhancements received a negative score in total, which would indicate that they have an overall undesirable impact on the usefulness of the WA system. Although, it could very well be that the participants did not want to imply that a

[74] See https://electology.org/score-voting

suggestion will reduce the usefulness, but just made use of the full scale to express their prioritization. The scanned voting forms and a summary of the votes can be found in Appendix A.

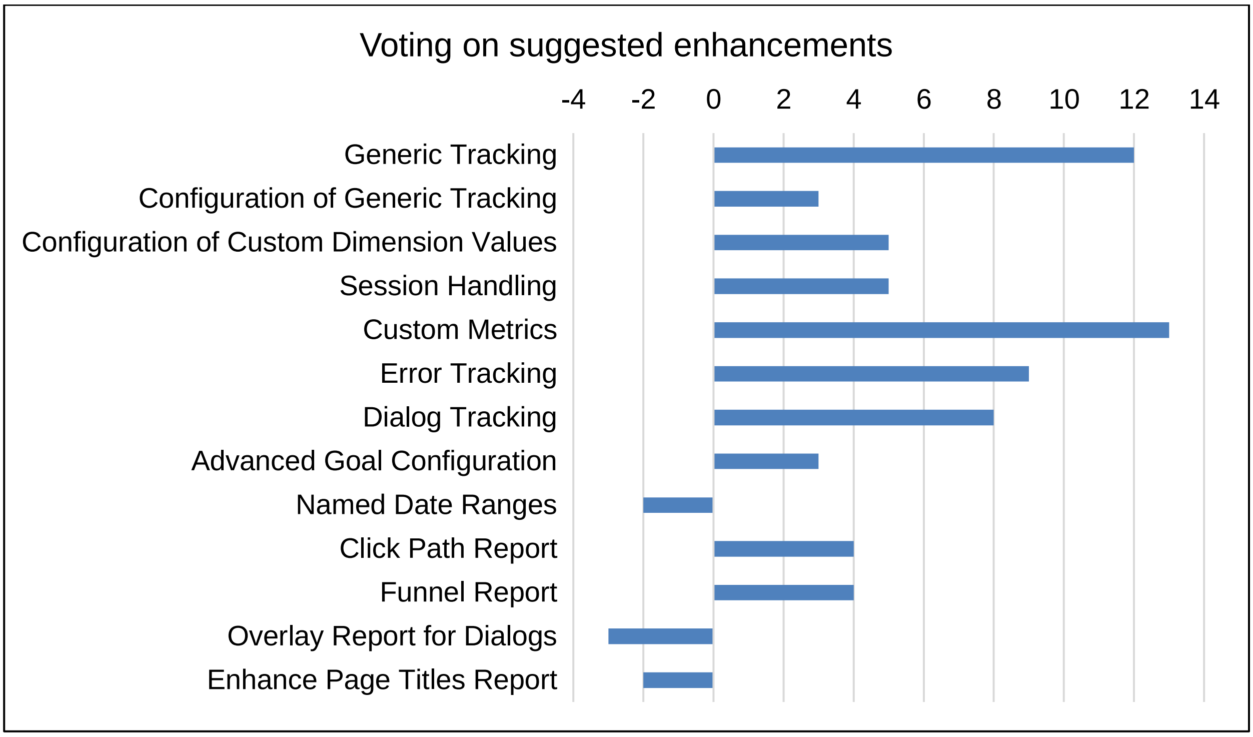


Figure 23: Bar chart showing the results of the voting on which enhancement should be implemented

Based on the results presented above the suggested enhanced “Custom Metrics” (see Section 3.2.2.1) was implemented as the second artifact of this work.

3.3.2 **Analysis**

The first step after the selection process was to analyze the already existing Custom Dimensions (CD) plug-in[75]. The Custom Metrics (CM) plug-in was supposed to be very similar to this existing extension and thus, it was a good starting point to get to know the Piwik code base and to learn about the possibilities and features of a plug-in. By analyzing and debugging the source code the following individual functions of the CD plug-in were identified:

- **Database Changes**: When installing the plug-in, it creates a new table in the DB schema and adds columns in the logging tables. These logging tables are used to store the raw information gathered by the tracker. There exist three such tables in the Piwik schema: One table each for the visit, the action and the conversion scope.

[75] See http://plugins.piwik.org/CustomDimensions

- **Reference Data (Administration)**: The CDs can be configured in the admin section of the Piwik front-end and are stored in a separate table. The dimensions can be setup independently for the visit and the action (e.g. pageview, event, download, search, etc.) scope. The visit scope configuration also applies for conversions.
- **Console Command**: The Piwik command line tool can be used to add or remove slots for custom dimensions by altering the logging tables.
- **API**: One central API allows access to the custom dimension configuration. The same methods in the API are used by the front-end and can be accessed also by any third-party (e.g. via HTTP).
- **Tracking**: The JavaScript tracker exposes functions to set, get and clear custom dimensions on the client, which are transferred to the server with every tracking request. The PHP back-end receives and handles the tracking requests. The back-end part of the plug-in is responsible for storing the received dimension values in the correct columns of the logging tables.
- **Aggregating**: The raw tracking data in the logging tables is aggregated by a so-called archiver process, which loads the data via custom select statements from the logging tables, aggregates the values and stores them in a format optimized for reporting. This format is called a *record* and can be used by one or many reports.
- **Reporting**: The plug-in adds a report and a menu entry for each configured custom dimension. A report consists of a reference to the underlying record and defines the UI build on top of it (e.g. which columns should be shown).

A plug-in is multifaceted as it consists of all the components listed above. Furthermore, it leverages various programming languages and frameworks, e.g. PHP, JavaScript, AngularJS, LESS[76] (for UI styles), Markdown[77] (for documentation), Twig[78] (for UI templates), Symfony[79] (for console commands). Therefore, creating a complete plug-in from scratch is very complex. It was decided to extend the existing CD plug-in instead. Another reason for enhancing the CD plug-in as an alternative to creating a new one was that there are many synergies between custom metrics and custom dimensions. It is easier to keep the artifact created as part of this thesis compatible with upcoming releases of Piwik if it becomes part of an already existing and continuously maintained plug-in, like the CD plug-in.

[76] See http://lesscss.org/

[77] See http://spec.commonmark.org/

[78] See https://twig.symfony.com/

[79] See https://symfony.com/components/Console

One part of the initial analysis was to get to know the Piwik code base and to decide on the general implementation approach. The second part was to investigate other WA solutions and to learn whether and how they support custom metrics. Three analytics solutions with such a feature could be identified: Kony MobileFabric[80] ,Google Analytics[81] [82] [83], and Snowplow[84].

Kony MobileFabric[85] is a back-end service for smartphone apps, which provides a centralized service for identity management, data access and tracking of user interactions. Custom metrics must be configured in this software upfront. Such a configuration consists of the name, a unique index (needed for the API), the data type (string, Boolean, long, double, date or timestamp) and a default aggregation method (sum, average, highest, lowest, count and distinct count) for numeric data types (Kony, Inc., 2017). The data type defines in which format the values are expected via the API, how they are stored in the database and in which format they are displayed in the UI. This analytics software is focused on tracking mobile applications and not web sites and thus provides a big set of data types. This fact highlights the need for custom metrics in Piwik, when using it for web applications. MobileFabric allows to save metrics in three different scopes: session (i.e. visit), request (to the back-end), event (i.e. action; they get cached on the client if there is no connection to the back-end).

Google Analytics, which in contrast to MobileFabric focuses on tracking web sites, provides only three data types (integer, time and currency) in the CM settings. It is not possible to define an aggregation mode, but one can define a name and a unique index. GA allows the tracking of metrics in four different scopes: hit (i.e. action), product (used for tracking everything related to one product on an e-commerce platform), session (i.e. visit), and user (metrics that span over multiple sessions). (Google Inc., 2017)

Snowplow handles custom metrics differently compared to the other two systems. Application-specific information can be attached to tracking requests as a so-called *custom context*, which is a JSON object. The structure of the object is described in a JSON schema definition. This schema needs to be defined and uploaded upfront to the tracking server, in contrast to configuring the custom metric via an admin interface in MobileFabric and GA (Tomilenko, 2017). Although this

[80] See http://docs.kony.com/7_x_PDFs/mobilefabric/custom_metrics_reports.pdf

[81] See https://developers.google.com/analytics/devguides/collection/analyticsjs/custom-dims-mets

[82] See https://developers.google.com/analytics/devguides/collection/analyticsjs/field-reference#customs

[83] See https://support.google.com/analytics/answer/2709829

[84] See https://github.com/snowplow/snowplow/wiki/Custom-contexts

[85] See http://docs.kony.com/tutorials/MobileFabric/

approach seems very flexible, it makes the integration of the custom data into reports harder, as the application does not know whether a field in a custom context is a metric or a dimension. No further information could be found on how to integrate this data into reports. The assumption is that specific code needs to be written to handle and extract the data on the server. Thus, the procedure used in Snowplow will not be considered for the artifact of this work.

The approach used for custom metrics in Piwik is on one hand loosely based on the corresponding implementation in GA and MobileFabric and on the other hand on the already existing Custom Dimensions plug-in for Piwik.

3.3.3 Design

Based on the analysis certain design decisions were made for the modification of the CD plug-in. Custom metrics should be supported - analogously to custom dimensions – in the visits and the actions scope. For now, two data types (integer and float) should be configurable, but the design should already consider the introduction of more data types in the future.

Configuration

Analogously to the plug-in-specific table that is used to store the configuration of custom dimensions, a new table should be added to the database schema for the custom metric settings. One configuration record should contain the name, the data type and a flag for each supported aggregation method. These flags should control whether or not the result of the aggregation is added as a new column to reports. A new menu entry in the admin section should be added to access the configuration screen.

Tracking

During the installation the CD plug-in adds five string columns to each logging table by default for tracking the dimensions. These slots can then be assigned by creating a configuration. However, following the same approach for the metrics would mean to add five new columns in every logging table for each supported data type. Therefore, a lot of columns would be added during the installation, although they might not be needed after all. An additional concern regards existing deployments. When adding columns to logging tables, which already contain a lot of data, the schema changes can take very long and potentially impact the ongoing tracking. Therefore, it was decided to not add columns for metrics during the installation of the CD plug-in. They can be added via a command line tool as individually required.

Reports

The custom metrics should be added as new columns to all existing CD reports as defined in the configuration. Furthermore, one new report per scope should be added to display the aggregated metrics columns in the UI based on all the data, without a segmentation applied to it. The columns in these two new reports should be the same as for the dimension-specific reports.

All the design decisions above were applied when enhancing the CD plug-in to support custom metrics.

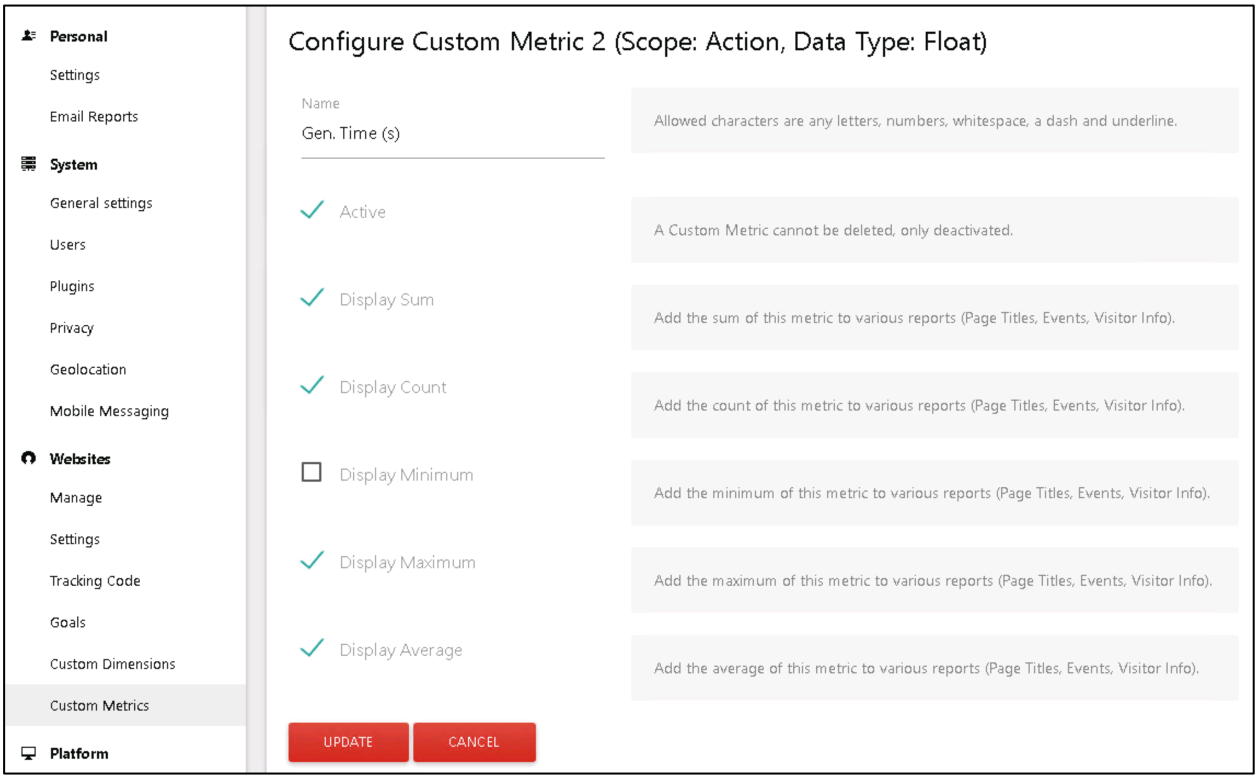
3.3.4 Implementation

This section describes the individual steps during the implementation of the artifact. Version 3.1.1 of the CD plug-in[86] was used as a basis for the modifications. The first step was to develop the UI for the configuration of custom metrics and subsequently to refactor the database access layer by introducing a common base class (`ConfigTable.php`) for both types of configurations. The specific code was implemented in the two sub-classes (`ConfigTableMetric.php` and `ConfigTableDimension.php`). A screenshot of the configuration UI can be seen in Figure 24.

The next step was to rewrite the entire code that populates the plug-in-specific columns in the logging tables (`LogTable.php`) in order to support user-defined metrics for all available data types. Furthermore, the JavaScript library needed to be enhanced to allow the tracking of custom metrics. It was modified to accept attributes in the tracking request starting with “metric” followed by a numeric ID. Such attributes are parsed on the client and afterwards sent to the Piwik server. The back-end uses the metric configuration to determine the specific column of the appropriate logging table to store the value based on the received ID.

The subsequent phase, after storing the values in the logging table, was to prepare the underlying data used for reporting in so-called records. This was done by enhancing the plug-in-specific code (`Archiver.php`) that selects the raw values from the logging tables, aggregates them and stores the data in a binary format in the database. The final step was to enhance the plug-in-specific reports (`GetCustomDimension.php`): All enabled aggregations per metric are added as new columns to each custom dimension report.

[86] See https://github.com/piwik/plugin-CustomDimensions/releases/tag/3.1.1



Configure Custom Metric 2 (Scope: Action, Data Type: Float)

Name: Gen. Time (s)

Allowed characters are any letters, numbers, whitespace, a dash and underline.

✓ Active

A Custom Metric cannot be deleted, only deactivated.

✓ Display Sum

Add the sum of this metric to various reports (Page Titles, Events, Visitor Info).

✓ Display Count

Add the count of this metric to various reports (Page Titles, Events, Visitor Info).

□ Display Minimum

Add the minimum of this metric to various reports (Page Titles, Events, Visitor Info).

✓ Display Maximum

Add the maximum of this metric to various reports (Page Titles, Events, Visitor Info).

✓ Display Average

Add the average of this metric to various reports (Page Titles, Events, Visitor Info).

Figure 24: Screenshot of the detail configuration of a custom metric (Piwik.org, 2017a)

The whole implementation process was accompanied by a time intensive deeper analysis of the Piwik code base and by studying the extensive online documentation[87]. In addition, the individual steps presented above were more effort than initially expected. Especially the tracking of metrics and the aggregation of the values for reporting was not as straightforward as assumed. The total effort starting from analyzing the existing CD plug-in and the Piwik code base to integrating it with Reval was about 80 hours. Therefore, the originally planned support for putting tracked metrics in configurable buckets and thus, creating dimensions out of metrics was not implemented.

The source code created as part of this thesis has not yet been published, but it is planned to share the modification with the community in the form of a pull request to the GitHub project.

3.3.5 Integration

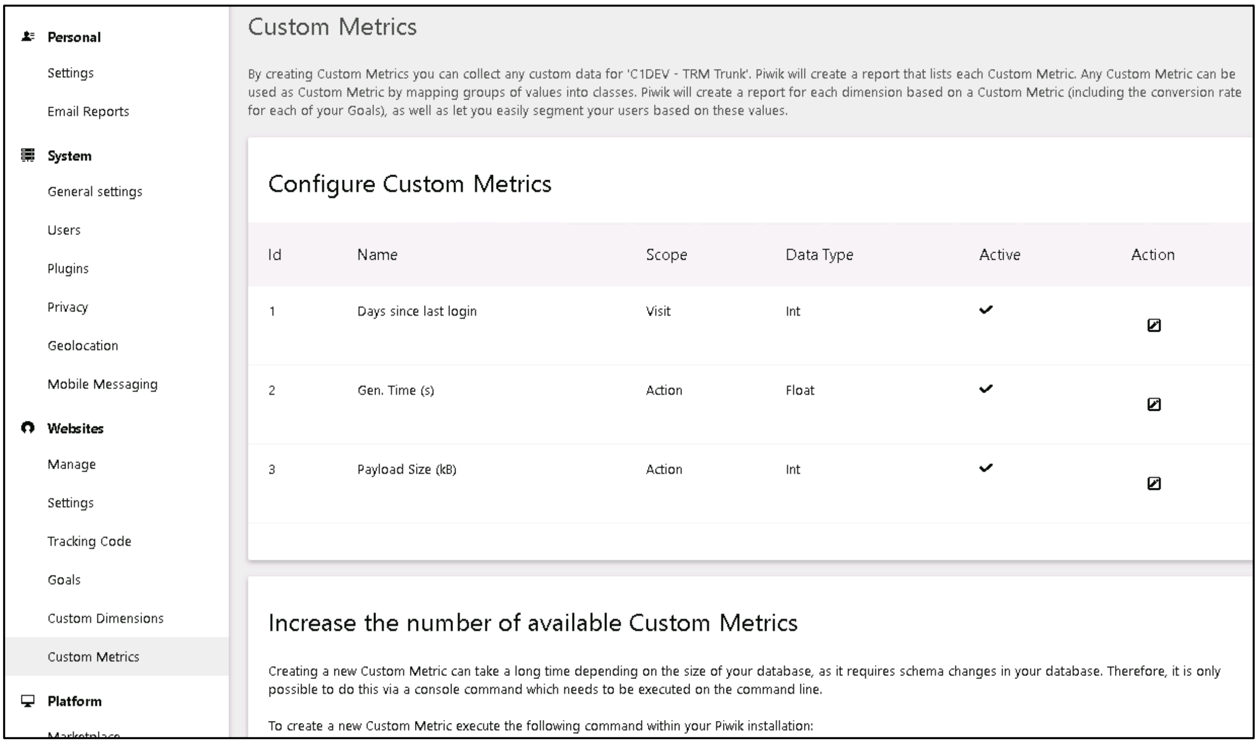
The next phase after enhancing the CD plug-in with support for custom metrics was to integrate this new feature with Reval. The following custom metrics should be tracked:

- The days since the last login (visit scope)

[87] See https://developer.piwik.org/guides/how-piwik-works

- The size of the payload in the response (measured in kilobytes; action scope)
- The duration the server needed for the response (measured in seconds; action scope)

Figure 25 shows a screenshot of the configuration overview in Piwik for these three custom metrics. Figure 26 displays how the same configuration is stored in the Piwik database schema.



Custom Metrics

By creating Custom Metrics you can collect any custom data for 'C1DEV - TRM Trunk'. Piwik will create a report that lists each Custom Metric. Any Custom Metric can be used as Custom Metric by mapping groups of values into classes. Piwik will create a report for each dimension based on a Custom Metric (including the conversion rate for each of your Goals), as well as let you easily segment your users based on these values.

Configure Custom Metrics

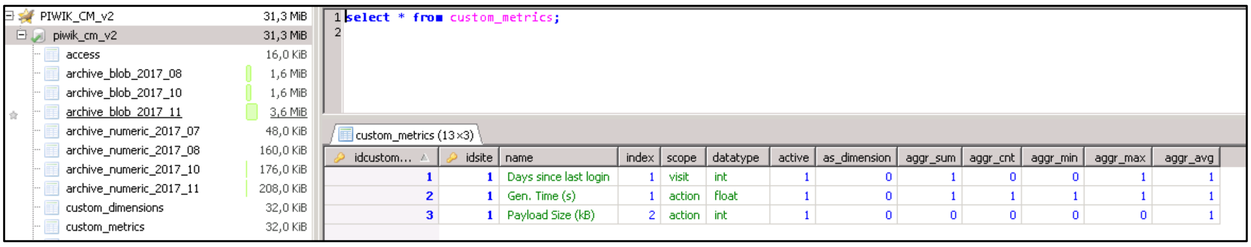
Id	Name	Scope	Data Type	Active	Action
1	Days since last login	Visit	Int	✓	✎
2	Gen. Time (s)	Action	Float	✓	✎
3	Payload Size (kB)	Action	Int	✓	✎

Increase the number of available Custom Metrics

Creating a new Custom Metric can take a long time depending on the size of your database, as it requires schema changes in your database. Therefore, it is only possible to do this via a console command which needs to be executed on the command line.

To create a new Custom Metric execute the following command within your Piwik installation:

Figure 25: A screenshot of the CM configuration overview in Piwik (Piwik.org, 2017a)



PIWIK_CM_v2	31,3 MiB
piwik_cm_v2	31,3 MiB
access	16,0 KiB
archive_blob_2017_08	1,6 MiB
archive_blob_2017_10	1,6 MiB
archive_blob_2017_11	3,6 MiB
archive_numeric_2017_07	48,0 KiB
archive_numeric_2017_08	160,0 KiB
archive_numeric_2017_10	176,0 KiB
archive_numeric_2017_11	208,0 KiB
custom_dimensions	32,0 KiB
custom_metrics	32,0 KiB

```
select * from custom_metrics;
```

custom_metrics (13×3)

idcustom...	idsite	name	index	scope	datatype	active	as_dimension	aggr_sum	aggr_cnt	aggr_min	aggr_max	aggr_avg
1	1	Days since last login	1	visit	int	1	0	1	0	0	1	1
2	1	Gen. Time (s)	1	action	float	1	0	1	1	1	1	1
3	1	Payload Size (kB)	2	action	int	1	0	0	0	0	0	1

Figure 26: A screenshot showing the content of the table custom_metrics in the Piwik DB schema (Becker, 2017)

Besides configuring the application-specific metrics in Piwik, the SPA needed to be enhanced to provide the according values. The metrics for the visit scope are set statically in the tracker object after loading the JS library during the login process, whereas the action metrics are attached individually to each tracking request. Listing 3 shows the modified function used for tracking events. In contrast to the original implementation (see Listing 2) it now supports sending application-specific information in the form of custom dimensions and metrics.

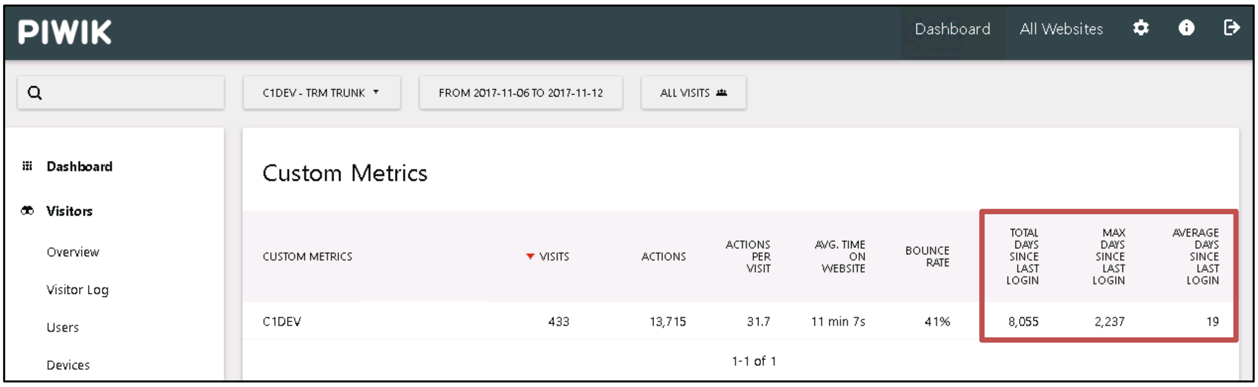
```
function piwik_trackEvent(category, action, name, value,
                          genTime, payLoadSize, module) {
    // send action specific custom fields as part of the request
    var customData = { };
    if (module) {
        customData.dimension4 = module;
    }
    if (genTime) {
        // convert from milliseconds to seconds
        customData.metric2 = genTime / 1000.0;
    }
    if (payLoadSize) {
        customData.metric3 = payLoadSize;
    }
    _paq.push(['trackEvent', category, action, name, value,
        customData]);
}
```

Listing 3: JavaScript code to track custom metrics and dimensions as part of an event via the JS API (Piwik.org, 2017a; Reval Inc., 2017a)

3.3.6 Rollout

The enhanced version of the CD plug-in was rolled out on the internal test environment at Reval Inc. in October 2017. After gathering usage data (including custom metrics) for two weeks, the plug-in was presented to the product development division. The custom metric specific enhancements were demonstrated by first showing the configuration capabilities and then presenting the resulting reports. Any open questions were answered subsequently. Figure 27 shows a screenshot of the report that contains the application-specific measurements created for the visit scope. The configurable columns are highlighted with a red frame. Figure 28 presents the aggregations “sum”, “count”, “max” and “average” for custom metrics in the report based on actions.

The next step was to evaluate the enhancement. Therefore, a link to an online survey was sent out via email to all members of the product development division right after the presentation. The content and results of this questionnaire are covered in the next chapter.

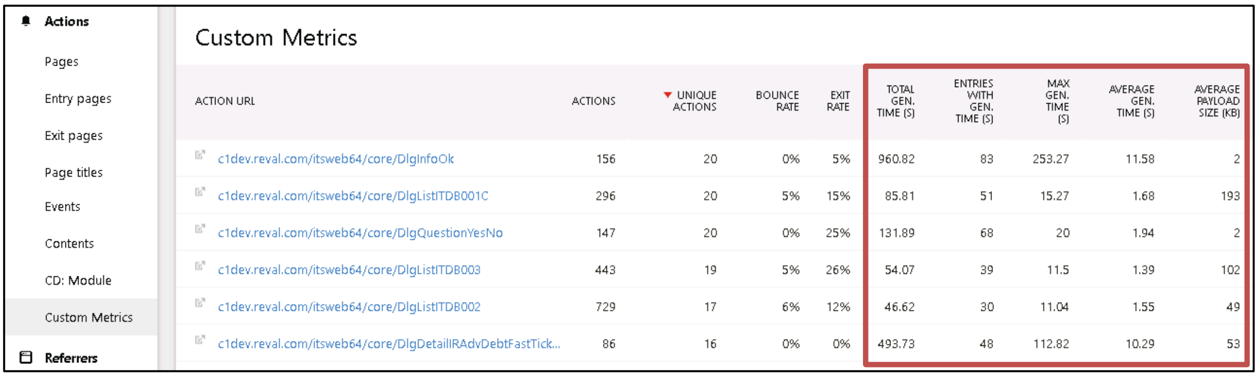


Custom Metrics

CUSTOM METRICS	VISITS	ACTIONS	ACTIONS PER VISIT	AVG. TIME ON WEBSITE	BOUNCE RATE	TOTAL DAYS SINCE LAST LOGIN	MAX DAYS SINCE LAST LOGIN	AVERAGE DAYS SINCE LAST LOGIN
C1DEV	433	13,715	31.7	11 min 7s	41%	8,055	2,237	19

1-1 of 1

Figure 27: A screenshot of the custom metrics report for the visit scope (Piwik.org, 2017a)



Custom Metrics

ACTION URL	ACTIONS	UNIQUE ACTIONS	BOUNCE RATE	EXIT RATE	TOTAL GEN. TIME (S)	ENTRIES WITH GEN. TIME (S)	MAX GEN. TIME (S)	AVERAGE GEN. TIME (S)	AVERAGE PAYLOAD SIZE (KB)
c1dev.reval.com/itsweb64/core/DlgInfoOk	156	20	0%	5%	960.82	83	253.27	11.58	2
c1dev.reval.com/itsweb64/core/DlgListITDB001C	296	20	5%	15%	85.81	51	15.27	1.68	193
c1dev.reval.com/itsweb64/core/DlgQuestionYesNo	147	20	0%	25%	131.89	68	20	1.94	2
c1dev.reval.com/itsweb64/core/DlgListITDB003	443	19	5%	26%	54.07	39	11.5	1.39	102
c1dev.reval.com/itsweb64/core/DlgListITDB002	729	17	6%	12%	46.62	30	11.04	1.55	49
c1dev.reval.com/itsweb64/core/DlgDetailIRAdvDebtFastTick...	86	16	0%	0%	493.73	48	112.82	10.29	53

Figure 28: A screenshot of the custom metrics report for the action scope (Piwik.org, 2017a)

4 SURVEY

This chapter describes how the implemented artifact was evaluated. The first part covers the design of the survey and the operationalization of the constructs. The second part presents the results of the survey and the statistical evaluation that was made based on the gathered responses.

4.1 Design

This subchapter describes the survey design. The evaluation of the enhanced CD plug-in is based on an adapted version of the ECM as described at the end of Chapter 2.5. The first step is to operationalize the four constructs of the adapted model. The survey items to measure the four latent variables are based on the questionnaire originally used by Bhattacherjee (2001). They were compared with items used by Kang, Hong, and Lee (2009) and then adjusted to the context of this work.

In contrast to the items utilized by Bhattacherjee (2001) the statements used for measuring the *Perceived Usefulness* in this work do not include a particular activity, as web analytics can aid various tasks (e.g. writing requirements, improving performance, restructuring of the application, etc.). It has been left to the interpretation of the participants in which way the artifact will be useful to them. Furthermore, the nature of this construct was changed from a post-acceptance to a pre-acceptance variable by addressing the implemented Custom Metrics enhancement instead of Piwik. Subsequently, the tense of the sentences was changed from present continuous to future, as the participants did not have the possibility to use the enhancement before answering the questionnaire. A specific example displaying these changes is item PU1: Bhattacherjee (2001) used the statement "Using OBD increases my productivity in managing personal finances", whereas the item modified for this work reads "Using the Custom Metrics functionality in Piwik will increase my productivity".

A few additional minor modifications have been made to the original statements: The term "service level" in item CF2 was replaced with the more generic word "functionality" as the original sentence did not make sense for Piwik. Thus, the resulting statement is "The functionality provided by Piwik was better than I expected". Another performed alteration had to do with concerns regarding the understandability of the term "contented". Thus, it has been replaced with the adjective "happy" in item SF3.

Two different types of scales are used in the questionnaire. The *Satisfaction* gets evaluated via four items based on a seven-point semantic differential scale. The other three constructs are measured via seven-point Likert scales anchored between “strongly agree” and “strongly disagree”.

Similar to the study conducted by Bhattacherjee (2001) two additional variables, which are not part of the ECM, have been added to determine the prior use of the information system: The participants were asked how often they had already accessed the test installation of Piwik. They were allowed to freely answer the question via a text input field. In contrast to that, the total duration of the prior IS usage could only be answered by selecting one of multiple predefined answers. This was done deliberately as it is supposedly easier to remember the number of usages than the complete duration of the usage. Furthermore, four demographic questions regarding age, sex, department and the participation in trainings were added to the survey to allow a comparison between groups. The complete list of items can be found in Appendix B.

4.2 Results

The link to the online survey was sent out to all 65 members of the product development division right after the presentation of the Custom Metrics feature. The survey was available for 10 days and received 32 responses during that time. This paragraph presents an overview of the distribution of the respondents for each of the six dimensions after converting the metric items age and times into classes: Six out of 32 participants were female and the most responses came from the product management (37.5%) and software engineering (46,9%) departments within the product development division. The majority of the participants was between 25 and 44 years old (81,3%). The Piwik-specific dimensions show that most responders used the IS between one and five times (75.0%). A good number of them invested between 10 and 20 minutes to evaluate the WA solution (37.5%). Both dimensions are visualized in the two bar charts in Figure 29 and Figure 30.

A more comprehensive visualization of the distribution of dimensions can be seen in two ParallelSet[88] charts in Figure 32 and Figure 31. These charts present a hierarchic flow of participants from the top level to the lower levels. The first chart in Figure 32 shows for example in which department the male respondents worked and subsequently the age distribution within the departments. The complete demographic data can be found in Appendix B.

[88] Both charts were rendered with the open-source tool ParallelSets. See https://eagereyes.org/parallel-sets

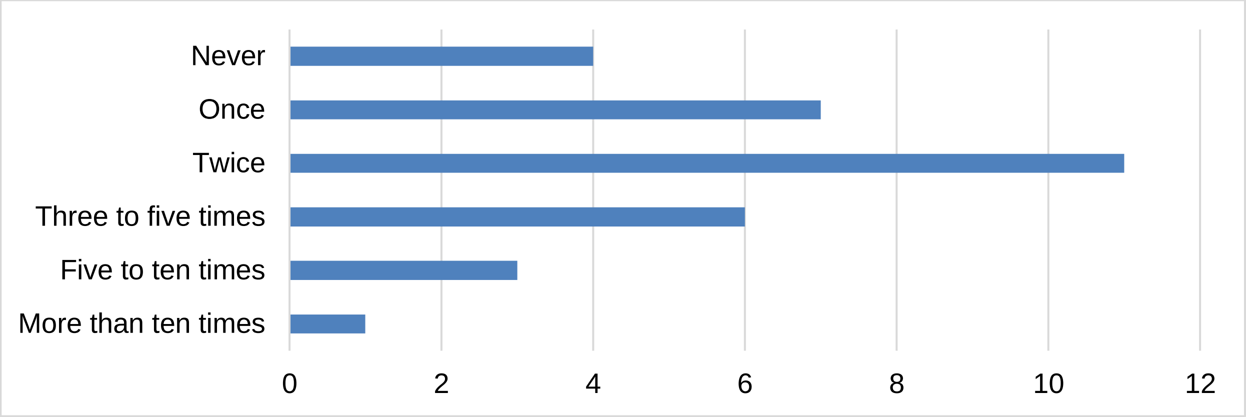


Figure 29: Bar chart showing how often the respondents used Piwik

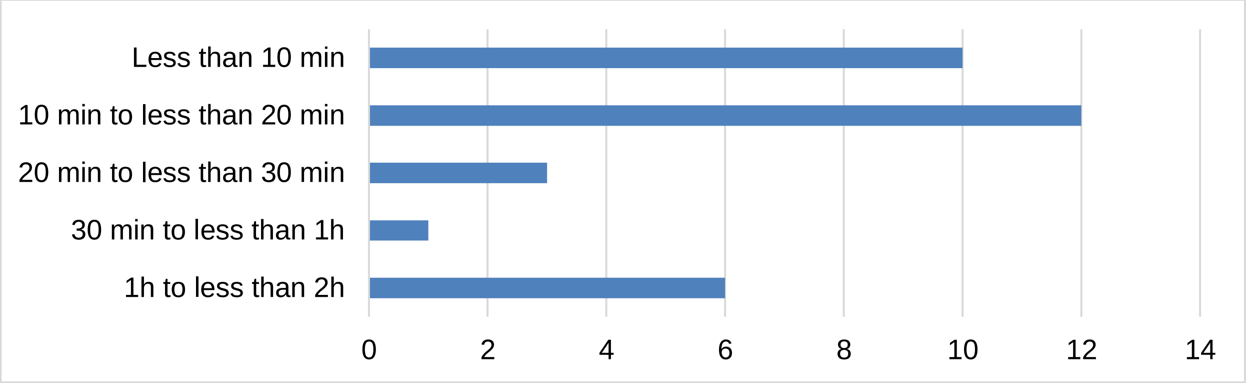


Figure 30: Bar chart visualizing for how long in total the participants used the test deployment of Piwik

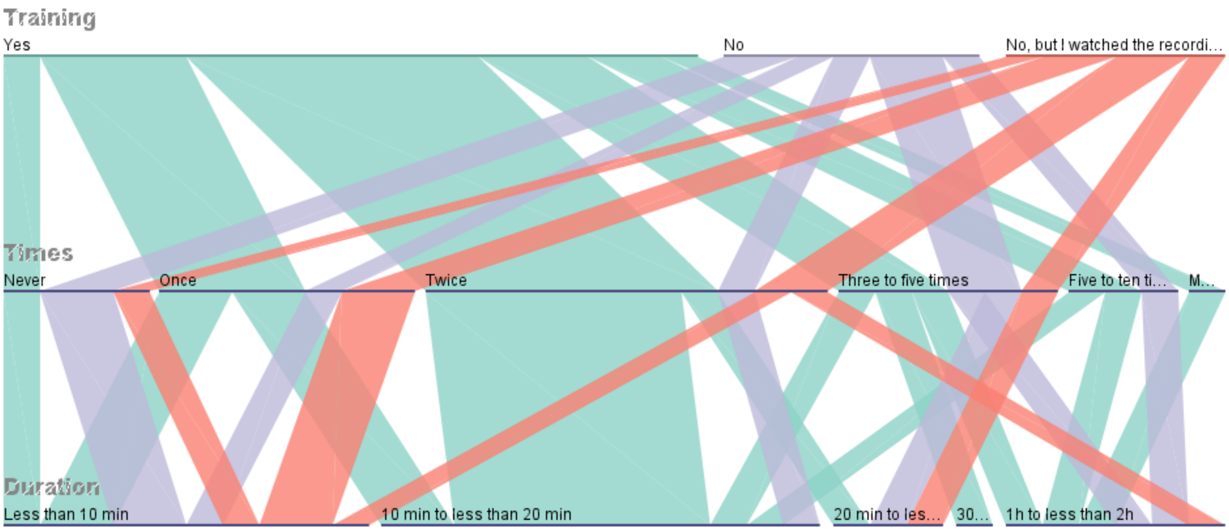


Figure 31: Distribution of respondents in the dimensions training, times, and duration

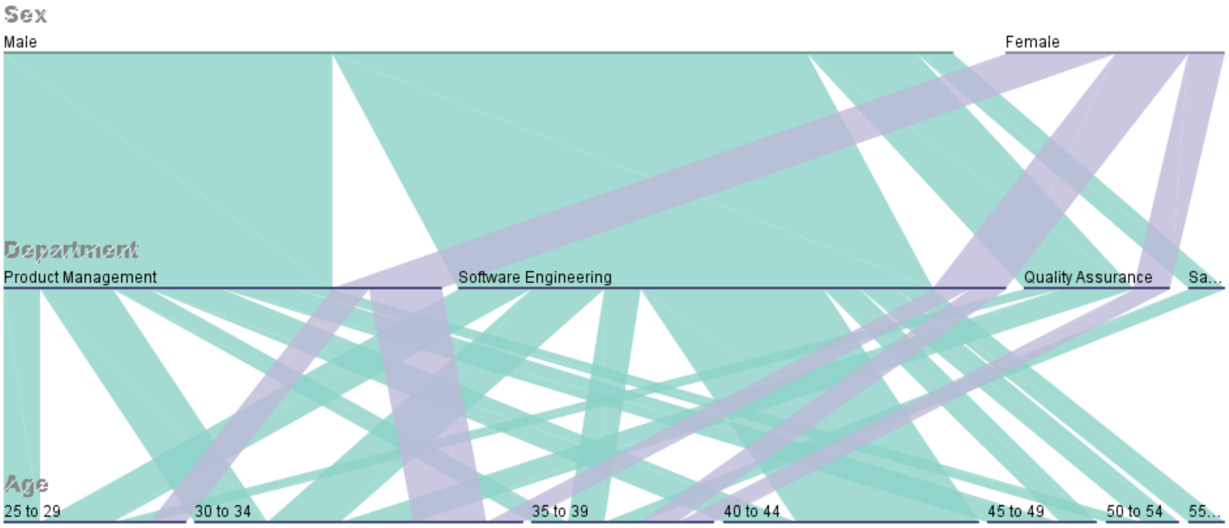


Figure 32: Distribution of respondents in the dimensions sex, department, and age

An unobtrusive research method was used to verify the validity of some of the answers: The access log file of the apache web server, which is used to host the internal Piwik deployment at Reval Inc., was imported into a separate Piwik installation[89]. Thus, the usage of the evaluated tool was analyzed with the help of the very same software. A SQL query was used to extract the number of visits and the total visit duration per IP address directly from the underlying DB schema. The complete comparison of the survey results against the SQL query response can be found in Appendix C. It shows that 28 respondents stated to have used Piwik at least once and exactly 28 unique visitors accessed the test installation. The real number of visits and the total duration time per visitor found in the log file are higher than the values stated in the survey results. An explanation for that could be that the participants underestimated their actual IS usage or that it was overstated by the software (e.g. a visitor using two different web browsers simultaneously leads to two visits in the WA system and a doubled usage time per unique visitor).

The results were evaluated with the open-source software “R”[90]: The first step was to analyze the responses and to form factors out of the items. As four respondents reported that they have never used Piwik so far, these observations were excluded from the analysis right away. The factor analysis of the remaining data revealed problems with the two independent latent factors *Perceived Usefulness* and *Satisfaction*. The loading factors show that all eight corresponding items represent only one common factor instead of the expected two factors. Therefore, all these items were combined into the latent factor *Perceived Usefulness*, as this construct is required for

[89] See https://piwik.org/docs/log-analytics-tool-how-to/

[90] See https://www.r-project.org/

answering the research question. The resulting model can be seen in Figure 33. Furthermore, the reverse coded item CI3 did not correlate with the two other items for the dependent variable *Continuance Intention*. Thus, it was excluded from the analysis and only CI1 and CI2 were used to form the latent factor.

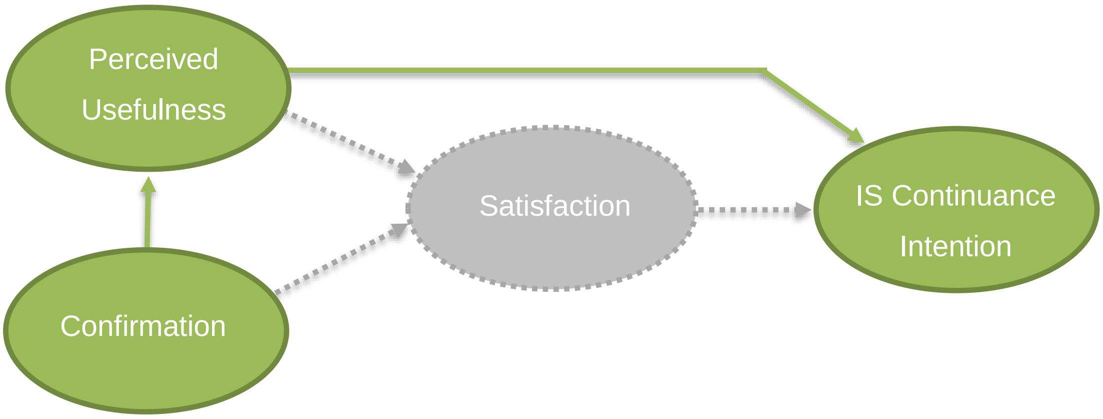


Figure 33: Adaption of the Expectation-Confirmation Model as used in the statistical analysis (Bhattacherjee, 2001)

Based on the three resulting factors two linear regressions were calculated. The first linear model (`m.ecm.pu`) evaluates the influence of the *Confirmation* on the *Perceived Usefulness*. The resulting coefficient has a value of 0.6092 (z-scaled) with a p-value of 0.000581 for the independent factor CF. The second model (`m.ecm.ci`) analyzes the impact of the *Perceived Usefulness* on the *Continuance Intention*. The resulting coefficient of the factor PU is 0.5119 (p=0.00149). The complete results for both models as created by R can be seen in Appendix B. These results will be interpreted in the next chapter.

5 CONCLUSION

The goal of this thesis was to identify gaps in web analytics solutions regarding single page applications and subsequently to evaluate the impact of resolving such a gap on the intended long-term usage of a given WA system by users. Therefore, an initial integration of an open-source WA framework with a commercial SPA was performed. Based on the gained experience and by conducting a literature research a list of 13 gaps was identified. Every gap was described and accompanied with a suggested solution. A group of professionals in the development of web applications ranked them by their impact on the usefulness of the WA solution. The resulting top item was the custom metrics feature. This enhancement was subsequently developed and integrated with the SPA. It allows to track application-specific metrics by configuring them generically in the WA system. These custom metrics can then be aggregated and used in reports within the WA solution.

The second artifact was evaluated via an online survey based on an adapted version of the Expectation-Confirmation Model. The replies were analyzed and the resulting 28 valid responses were the source for forming three latent factors, which were in turn used in two linear regression models. The first model `m.ecm.pu` shows that the CF explains the PU very well, as the probability that the correlation happened randomly is only 0.05% (which corresponds to a very high significance level of $p < 0.001$). The resulting coefficient of the regression (0.6092) means that the higher the confirmation of expectations regarding Piwik is, the higher the perceived usefulness of the Custom Metrics feature is. The second model `m.ecm.ci` evaluates the relation between PU and CI. The analysis identified a very probable association between the two variables ($p=0.00149$). The resulting coefficient is reported as 0.5119. It can be interpreted as: the higher the perceived usefulness of Custom Metrics in Piwik is, the more the intention to continue using the IS increases. The second linear regression is relevant for testing the initially stated hypotheses:

Alternative Hypothesis (H_1):

Enhancing web analytics frameworks with a focus on modern web applications increases the intention of users to keep using the information system in the long-term.

Null Hypothesis (H_0):

Enhancing web analytics frameworks with a focus on modern web applications has no influence on the intention of users to keep using the information system in the long-term.

The calculated p-value of 0.00149 is well beyond the applied significance level (*α=0.01)*, which means that it is very unlikely that the observed values are just a matter of coincidence. Therefore, the null hypothesis is rejected in favor of the alternative hypothesis. Subsequently this allows to answer the research question of this work:

What influence does better support of commercial single page applications have on the continuance intention towards web analytics frameworks?

The statistical analysis of the survey results shows that a specific improvement of a WA solution with the goal to increase the usefulness of this IS in fact **increases the intention** to keep using this software in the long-term.

6 DISCUSSION

This chapter takes a critical look at the decisions made in this work. Furthermore, the applied methods are discussed and potential alternatives are presented.

A big impediment for this work was that the WA solution was only hosted on an internal test system. It turned out that the availability of a web analytics system was less appealing to the employees as expected, as it did not track actual clients and thus, did not provide new insights into the behavior of clients. Just the possibility to try out the IS and to get to know the features and reports was only compelling 28 employees to access the installation. However, the release cycles at Reval Inc. last six months and thus, tracking of actual clients could not be incorporated in this work.

While determining the methodology of this thesis multiple approaches have been considered. It would have been an option to measure the intention of using the WA system in the long-term at different points in time. For instance, once before introducing the Custom Metrics feature and another time after an appropriate evaluation period. However, no suitable way could be determined to connect the two responses of the same individual to each other without asking them to give up their anonymity. Another alternative would have been to conduct an experiment with two groups: One group would get access to the original version of the WA system and the other would evaluate the enhanced version. However, this approach was not taken, as it would have been difficult to ensure that the experiment group on one hand is aware of the additional feature, and on the other hand does not get biased by too much training regarding the enhancement.

Observing the usage of the WA system was considered as an unobtrusive method for this work. This would have been accomplished by tracking the users on the internal test installation of Piwik with another separate deployment by making use of a special plug-in for that[91]. However, as "tracking" is in general a sensitive topic, it seemed too risky that employees would realize that they are observed and that this would subsequently create resentment against "web analytics" in general. Therefore, in the end only the log files produced by the web server were used to unobtrusively get an insight into the usage of the internal test installation.

[91] See https://plugins.piwik.org/AnonymousPiwikUsageMeasurement

The low usage numbers by the employees made it necessary to adjust the evaluation process. The initial plan for the developed artifact was to let the employees test it for two weeks. However, a deliberate decision was made against it, as it was unlikely that many would try it out. Instead it seemed more reasonable to send out the survey right after the presentation, as the impressions were still fresh at this point of time. Therefore, the survey items corresponding to the *Perceived Usefulness* were modified to measure expectations instead of experience.

One surprising development was that the survey results did not allow to use the items for *Perceived Usefulness* and *Confirmation* as separate factors. No other studies could be found that faced similar difficulties when applying the Expectation-Confirmation Model. This could have been an unintended side effect of modifying the questions to evaluate the PU.

As this work applies statistical inference based on a sample, this presumably led to the introduction of bias into the results of this work. For instance, it seems that the respondents were experiencing a recall bias when answering how often and how long they were using the IS in total, as the alternative method of analyzing log files revealed that the gathered responses were in general to low.

The practical part of this work focused on integrating a WA framework (Piwik) with a single page application (Reval). It would have been interesting to integrate another WA solution and to compare the insights gathered from both integrations. Furthermore, other gaps might have been identified when using a different commercial SPA as underlying IS. However, it was already quite an effort to get to know one web analytics system, it would have exceeded the volume of this work to include more WA frameworks.

7 OUTLOOK

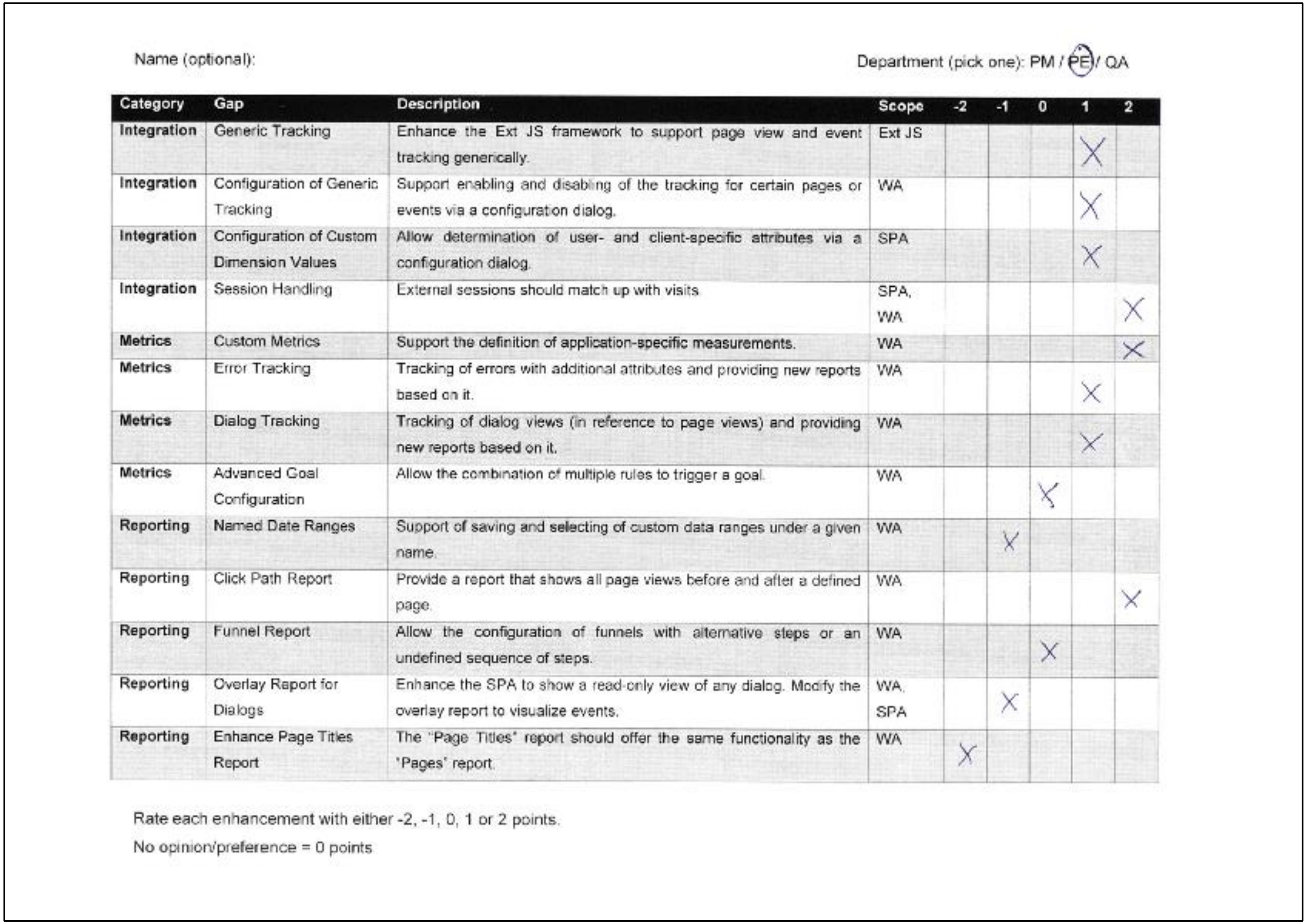
Future research in the area of integration of web analytics systems with single page applications should build upon this work. It would be desirable to implement more of the suggested enhancements and to evaluate how they impact the usage of the WA system. Furthermore, this study should be repeated with a WA deployment that actually already contains client data in order to verify the results of this work under such conditions. It would be interesting to analyze which effect the availability of tracked client behavior has on the evaluated constructs.

Further research could be based on this work once a reasonable amount of usage data has been tracked. With enough data available, it would be possible to apply machine learning algorithms. The goal could for example be to use clustering algorithms to identify usage patterns or unapparent workflows. The insight gained through such mechanisms could be used to in turn improve the underlying application.

Another idea for a web analytics based research topic would be to automatically identify users that are having troubles operating the web application. This could be done by establishing a baseline behavior and identify users deviating from the regular behavior. For example, a user that takes very long to fill out a form could have troubles understanding the UI. Again, machine learning could be applied in such a scenario to draw to the correct conclusions.

A final idea for a follow up research would be to investigate the integration of A/B testing into single page applications. The idea behind A/B testing is to deliver two different versions of a UI to the users of web sites and then decide based on statistical inference whether or not one of the two options is significantly better than the other. Integrating something like that into web applications would result in technical challenges that need to be identified and overcome.

APPENDIX A - Voting forms for suggested enhancements



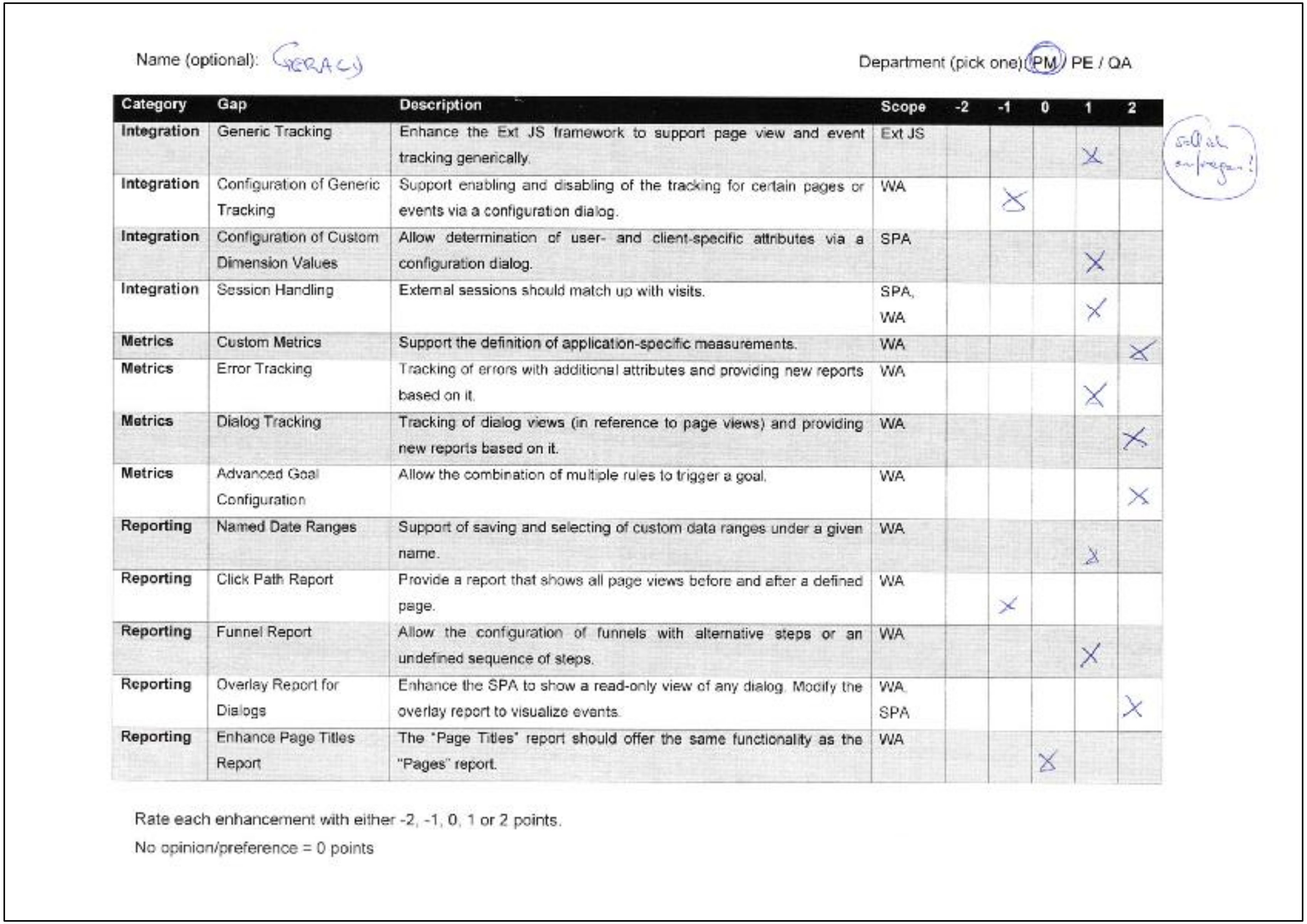
Name (optional):

Department (pick one): PM / [✓ PE] / QA

Category	Gap	Description	Scope	-2	-1	0	1	2
Integration	Generic Tracking	Enhance the Ext JS framework to support page view and event tracking generically.	Ext JS				X	
Integration	Configuration of Generic Tracking	Support enabling and disabling of the tracking for certain pages or events via a configuration dialog.	WA				X	
Integration	Configuration of Custom Dimension Values	Allow determination of user- and client-specific attributes via a configuration dialog.	SPA				X	
Integration	Session Handling	External sessions should match up with visits.	SPA, WA					X
Metrics	Custom Metrics	Support the definition of application-specific measurements.	WA					X
Metrics	Error Tracking	Tracking of errors with additional attributes and providing new reports based on it.	WA				X	
Metrics	Dialog Tracking	Tracking of dialog views (in reference to page views) and providing new reports based on it.	WA				X	
Metrics	Advanced Goal Configuration	Allow the combination of multiple rules to trigger a goal.	WA			X		
Reporting	Named Date Ranges	Support of saving and selecting of custom data ranges under a given name.	WA		X			
Reporting	Click Path Report	Provide a report that shows all page views before and after a defined page.	WA					X
Reporting	Funnel Report	Allow the configuration of funnels with alternative steps or an undefined sequence of steps.	WA			X		
Reporting	Overlay Report for Dialogs	Enhance the SPA to show a read-only view of any dialog. Modify the overlay report to visualize events.	WA, SPA		X			
Reporting	Enhance Page Titles Report	The "Page Titles" report should offer the same functionality as the 'Pages' report.	WA	X				

Rate each enhancement with either -2, -1, 0, 1 or 2 points.

No opinion/preference = 0 points



Name (optional): GERACY

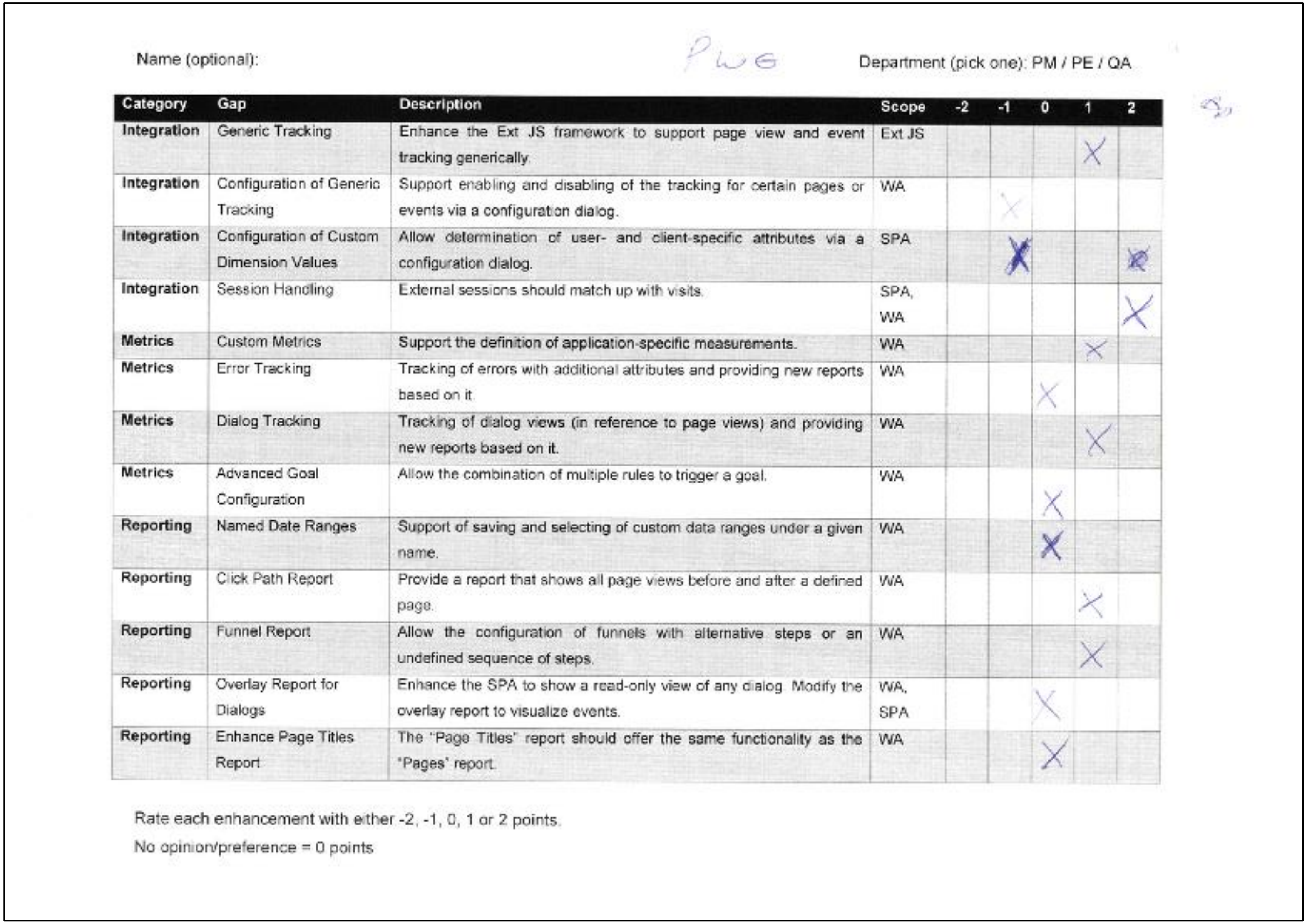
Department (pick one): [✓ PM] / PE / QA

Category	Gap	Description	Scope	-2	-1	0	1	2
Integration	Generic Tracking	Enhance the Ext JS framework to support page view and event tracking generically.	Ext JS				X	
Integration	Configuration of Generic Tracking	Support enabling and disabling of the tracking for certain pages or events via a configuration dialog.	WA		X			
Integration	Configuration of Custom Dimension Values	Allow determination of user- and client-specific attributes via a configuration dialog.	SPA				X	
Integration	Session Handling	External sessions should match up with visits.	SPA, WA				X	
Metrics	Custom Metrics	Support the definition of application-specific measurements.	WA					X
Metrics	Error Tracking	Tracking of errors with additional attributes and providing new reports based on it.	WA				X	
Metrics	Dialog Tracking	Tracking of dialog views (in reference to page views) and providing new reports based on it.	WA					X
Metrics	Advanced Goal Configuration	Allow the combination of multiple rules to trigger a goal.	WA					X
Reporting	Named Date Ranges	Support of saving and selecting of custom data ranges under a given name.	WA				X	
Reporting	Click Path Report	Provide a report that shows all page views before and after a defined page.	WA		X			
Reporting	Funnel Report	Allow the configuration of funnels with alternative steps or an undefined sequence of steps.	WA				X	
Reporting	Overlay Report for Dialogs	Enhance the SPA to show a read-only view of any dialog. Modify the overlay report to visualize events.	WA, SPA					X
Reporting	Enhance Page Titles Report	The "Page Titles" report should offer the same functionality as the "Pages" report.	WA			X		

[illegible]

Rate each enhancement with either -2, -1, 0, 1 or 2 points.

No opinion/preference = 0 points



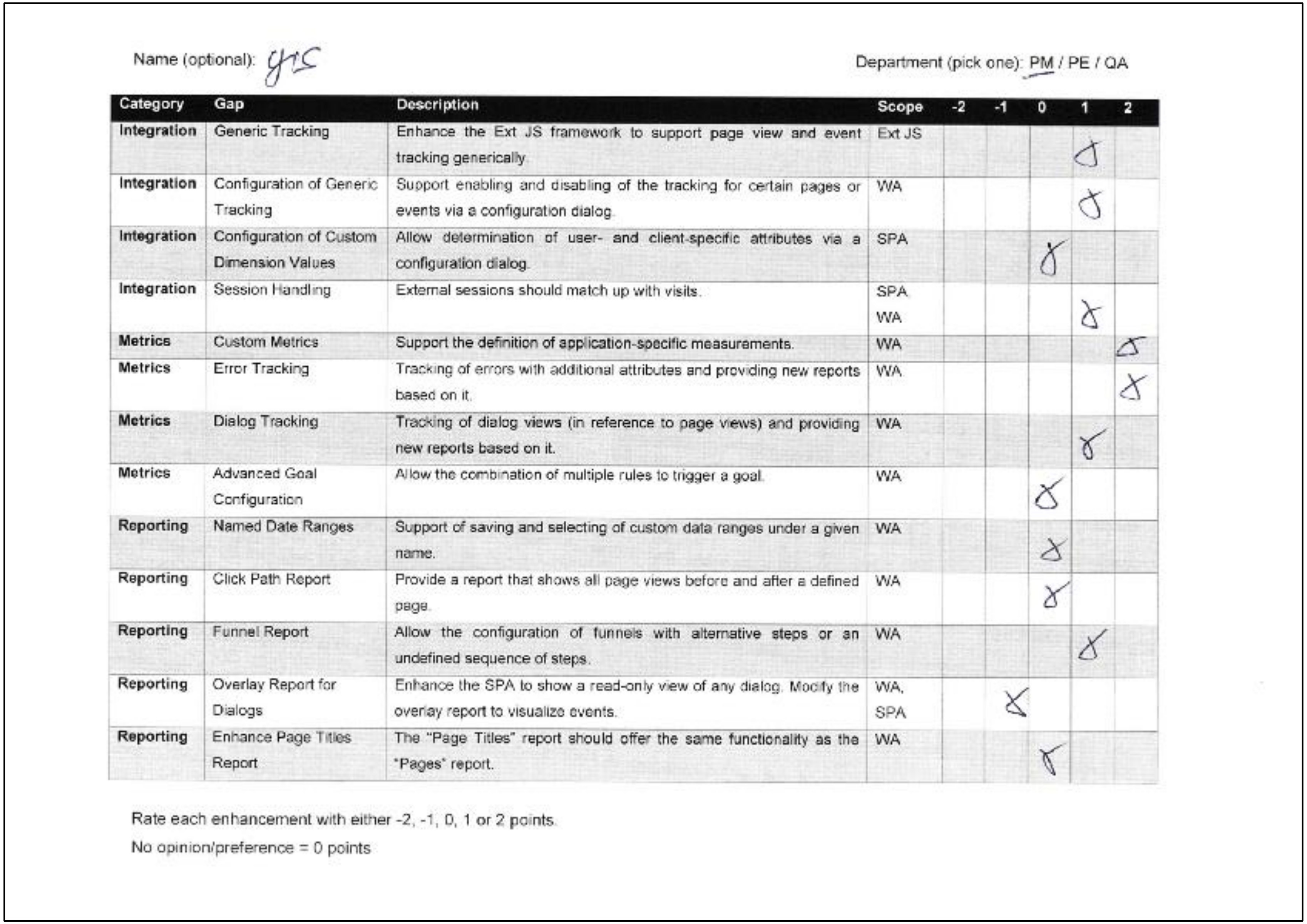
Name (optional): PWG

Department (pick one): PM / PE / QA

Category	Gap	Description	Scope	-2	-1	0	1	2
Integration	Generic Tracking	Enhance the Ext JS framework to support page view and event tracking generically.	Ext JS				X	
Integration	Configuration of Generic Tracking	Support enabling and disabling of the tracking for certain pages or events via a configuration dialog.	WA		X			
Integration	Configuration of Custom Dimension Values	Allow determination of user- and client-specific attributes via a configuration dialog.	SPA		X			X
Integration	Session Handling	External sessions should match up with visits.	SPA, WA					X
Metrics	Custom Metrics	Support the definition of application-specific measurements.	WA				X	
Metrics	Error Tracking	Tracking of errors with additional attributes and providing new reports based on it.	WA			X		
Metrics	Dialog Tracking	Tracking of dialog views (in reference to page views) and providing new reports based on it.	WA				X	
Metrics	Advanced Goal Configuration	Allow the combination of multiple rules to trigger a goal.	WA			X		
Reporting	Named Date Ranges	Support of saving and selecting of custom data ranges under a given name.	WA			X		
Reporting	Click Path Report	Provide a report that shows all page views before and after a defined page.	WA				X	
Reporting	Funnel Report	Allow the configuration of funnels with alternative steps or an undefined sequence of steps.	WA				X	
Reporting	Overlay Report for Dialogs	Enhance the SPA to show a read-only view of any dialog. Modify the overlay report to visualize events.	WA, SPA			X		
Reporting	Enhance Page Titles Report	The "Page Titles" report should offer the same functionality as the "Pages" report.	WA			X		

Rate each enhancement with either -2, -1, 0, 1 or 2 points.

No opinion/preference = 0 points



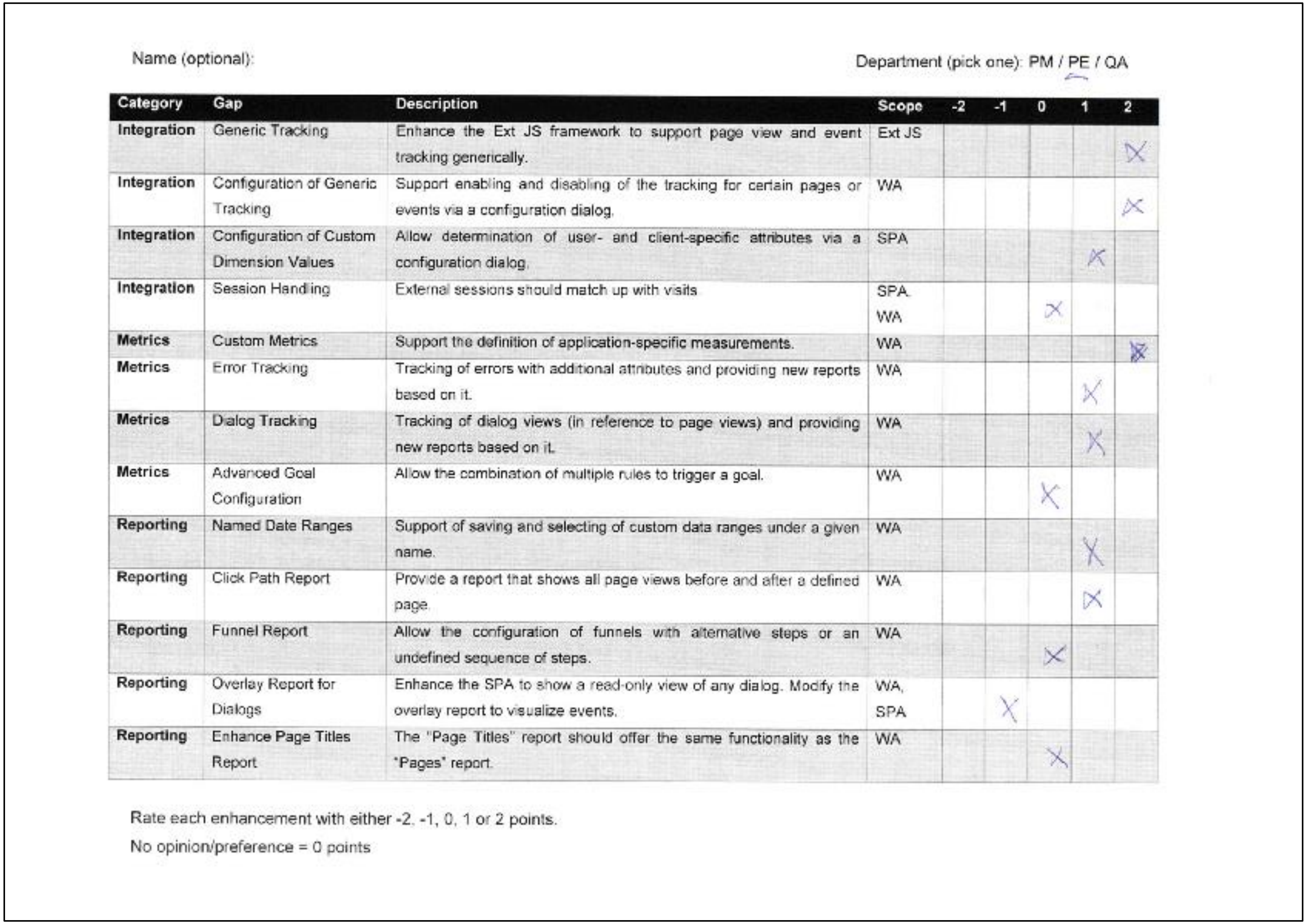
Name (optional): yis

Department (pick one): [✓ PM] / PE / QA

Category	Gap	Description	Scope	-2	-1	0	1	2
Integration	Generic Tracking	Enhance the Ext JS framework to support page view and event tracking generically.	Ext JS				X	
Integration	Configuration of Generic Tracking	Support enabling and disabling of the tracking for certain pages or events via a configuration dialog.	WA				X	
Integration	Configuration of Custom Dimension Values	Allow determination of user- and client-specific attributes via a configuration dialog.	SPA			X		
Integration	Session Handling	External sessions should match up with visits.	SPA WA				X	
Metrics	Custom Metrics	Support the definition of application-specific measurements.	WA					X
Metrics	Error Tracking	Tracking of errors with additional attributes and providing new reports based on it.	WA					X
Metrics	Dialog Tracking	Tracking of dialog views (in reference to page views) and providing new reports based on it.	WA				X	
Metrics	Advanced Goal Configuration	Allow the combination of multiple rules to trigger a goal.	WA			X		
Reporting	Named Date Ranges	Support of saving and selecting of custom data ranges under a given name.	WA			X		
Reporting	Click Path Report	Provide a report that shows all page views before and after a defined page.	WA			X		
Reporting	Funnel Report	Allow the configuration of funnels with alternative steps or an undefined sequence of steps.	WA				X	
Reporting	Overlay Report for Dialogs	Enhance the SPA to show a read-only view of any dialog. Modify the overlay report to visualize events.	WA, SPA		X			
Reporting	Enhance Page Titles Report	The "Page Titles" report should offer the same functionality as the "Pages" report.	WA			X		

Rate each enhancement with either -2, -1, 0, 1 or 2 points.

No opinion/preference = 0 points



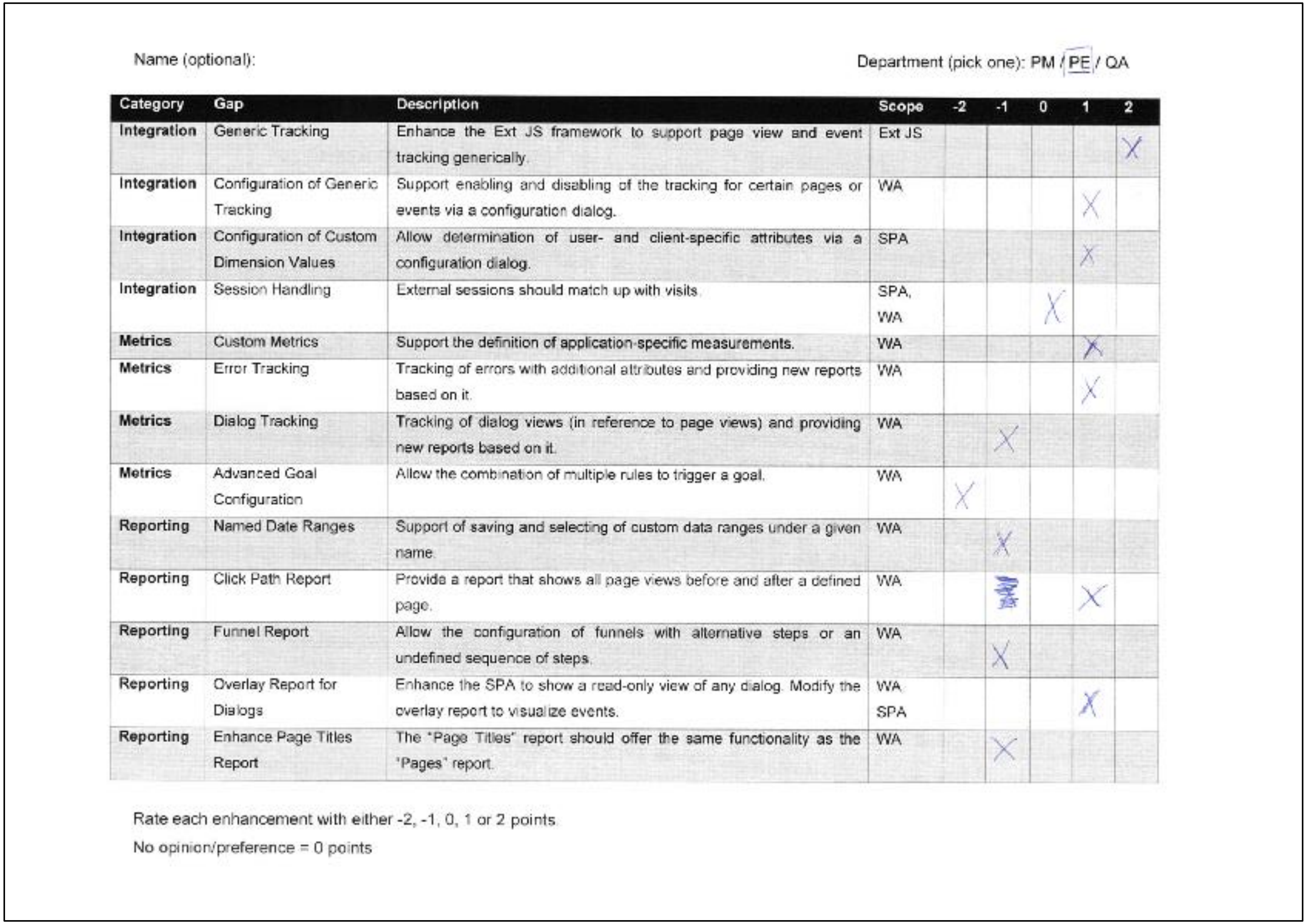
Name (optional):

Department (pick one): PM / [✓ PE] / QA

Category	Gap	Description	Scope	-2	-1	0	1	2
Integration	Generic Tracking	Enhance the Ext JS framework to support page view and event tracking generically.	Ext JS					X
Integration	Configuration of Generic Tracking	Support enabling and disabling of the tracking for certain pages or events via a configuration dialog.	WA					X
Integration	Configuration of Custom Dimension Values	Allow determination of user- and client-specific attributes via a configuration dialog.	SPA				X	
Integration	Session Handling	External sessions should match up with visits	SPA WA			X		
Metrics	Custom Metrics	Support the definition of application-specific measurements.	WA					X
Metrics	Error Tracking	Tracking of errors with additional attributes and providing new reports based on it.	WA				X	
Metrics	Dialog Tracking	Tracking of dialog views (in reference to page views) and providing new reports based on it.	WA				X	
Metrics	Advanced Goal Configuration	Allow the combination of multiple rules to trigger a goal.	WA			X		
Reporting	Named Date Ranges	Support of saving and selecting of custom data ranges under a given name.	WA				X	
Reporting	Click Path Report	Provide a report that shows all page views before and after a defined page.	WA				X	
Reporting	Funnel Report	Allow the configuration of funnels with alternative steps or an undefined sequence of steps.	WA			X		
Reporting	Overlay Report for Dialogs	Enhance the SPA to show a read-only view of any dialog. Modify the overlay report to visualize events.	WA, SPA		X			
Reporting	Enhance Page Titles Report	The "Page Titles" report should offer the same functionality as the "Pages" report.	WA			X		

Rate each enhancement with either -2, -1, 0, 1 or 2 points.

No opinion/preference = 0 points



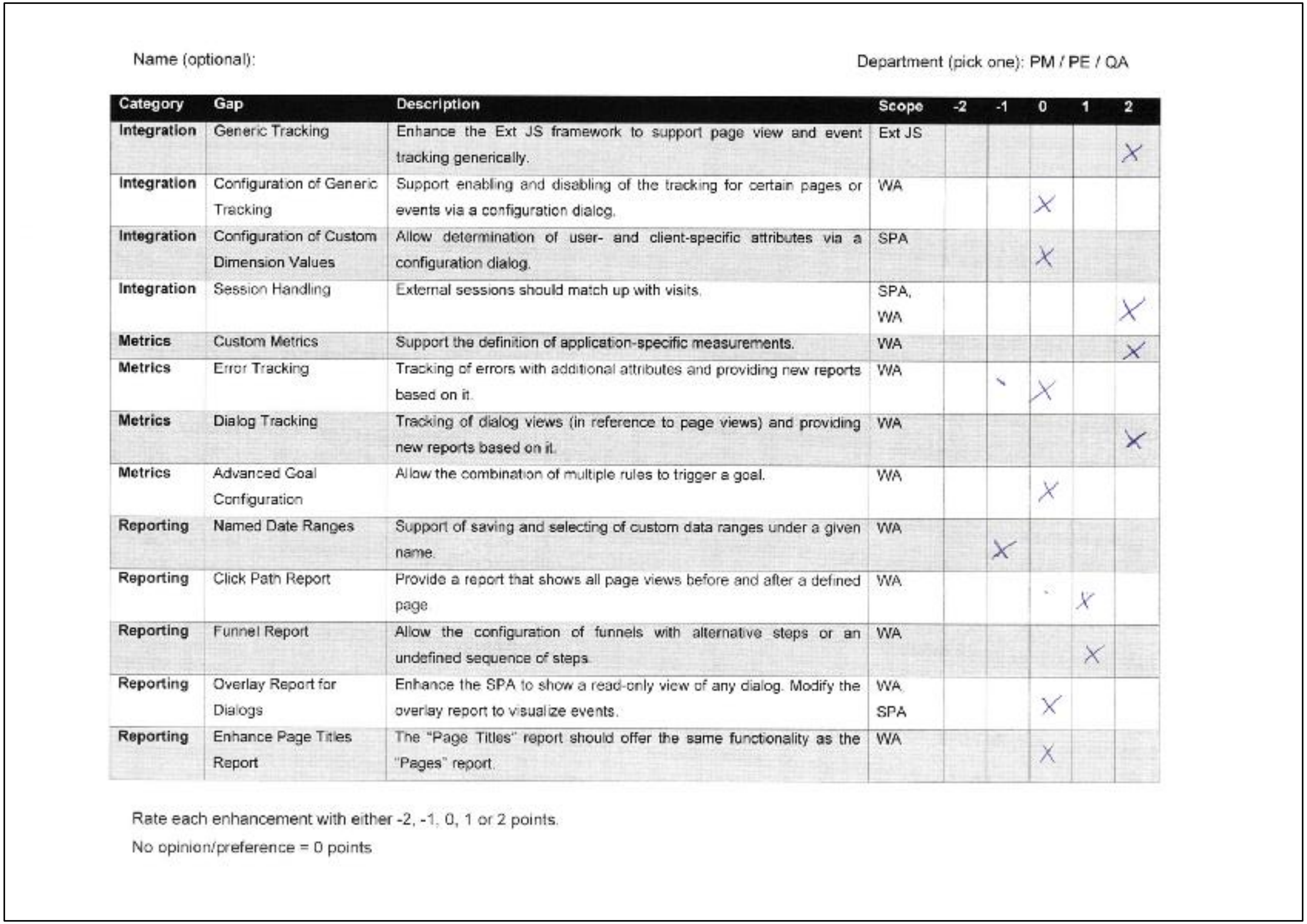
Name (optional):

Department (pick one): PM / [✓ PE] / QA

Category	Gap	Description	Scope	-2	-1	0	1	2
Integration	Generic Tracking	Enhance the Ext JS framework to support page view and event tracking generically.	Ext JS					X
Integration	Configuration of Generic Tracking	Support enabling and disabling of the tracking for certain pages or events via a configuration dialog.	WA				X	
Integration	Configuration of Custom Dimension Values	Allow determination of user- and client-specific attributes via a configuration dialog.	SPA				X	
Integration	Session Handling	External sessions should match up with visits.	SPA, WA			X		
Metrics	Custom Metrics	Support the definition of application-specific measurements.	WA				X	
Metrics	Error Tracking	Tracking of errors with additional attributes and providing new reports based on it.	WA				X	
Metrics	Dialog Tracking	Tracking of dialog views (in reference to page views) and providing new reports based on it.	WA		X			
Metrics	Advanced Goal Configuration	Allow the combination of multiple rules to trigger a goal.	WA	X				
Reporting	Named Date Ranges	Support of saving and selecting of custom data ranges under a given name.	WA		X			
Reporting	Click Path Report	Provide a report that shows all page views before and after a defined page.	WA		~~X~~		X	
Reporting	Funnel Report	Allow the configuration of funnels with alternative steps or an undefined sequence of steps.	WA		X			
Reporting	Overlay Report for Dialogs	Enhance the SPA to show a read-only view of any dialog. Modify the overlay report to visualize events.	WA SPA				X	
Reporting	Enhance Page Titles Report	The "Page Titles" report should offer the same functionality as the "Pages" report.	WA		X			

Rate each enhancement with either -2, -1, 0, 1 or 2 points.

No opinion/preference = 0 points



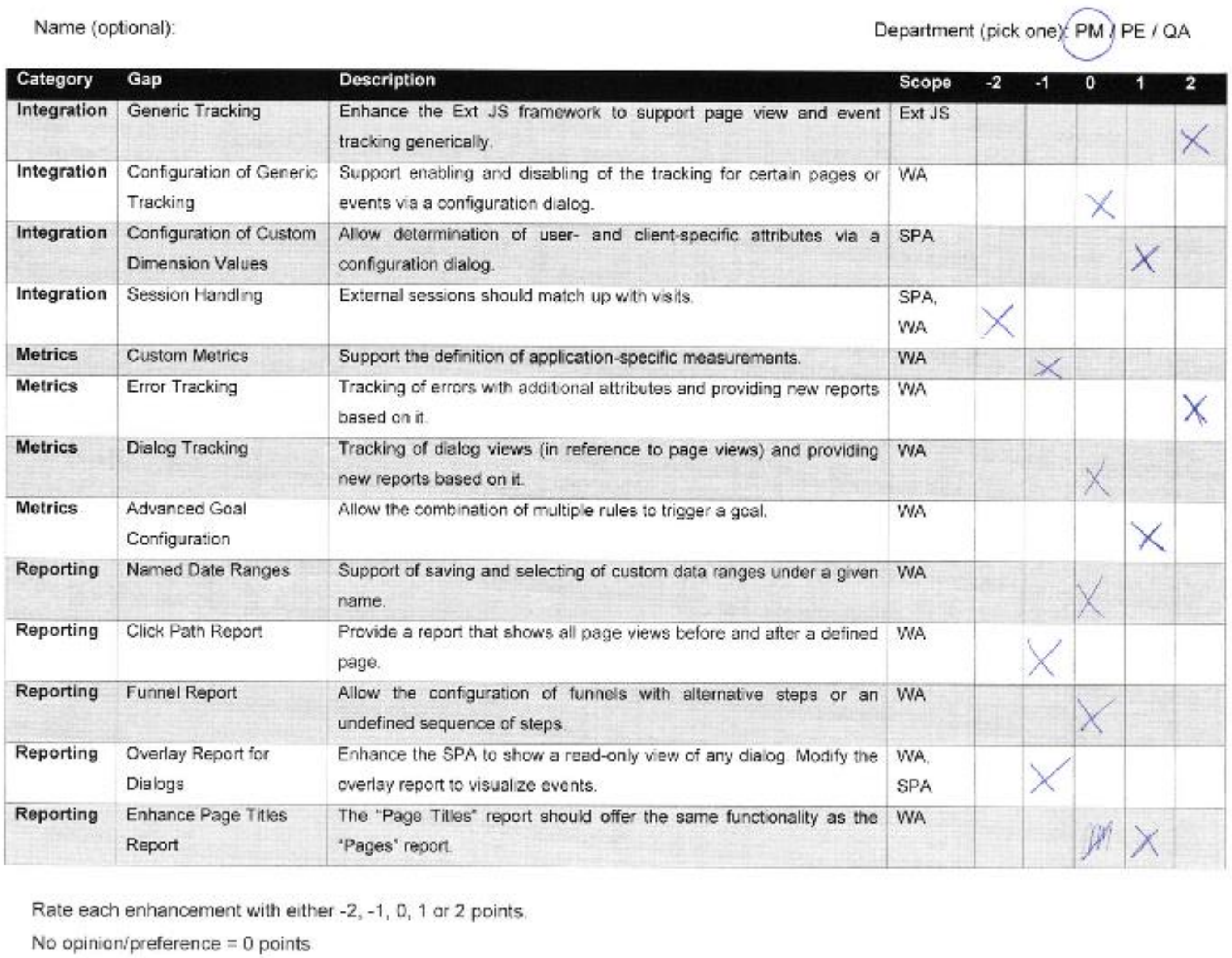
Name (optional):

Department (pick one): PM / PE / QA

Category	Gap	Description	Scope	-2	-1	0	1	2
Integration	Generic Tracking	Enhance the Ext JS framework to support page view and event tracking generically.	Ext JS					X
Integration	Configuration of Generic Tracking	Support enabling and disabling of the tracking for certain pages or events via a configuration dialog.	WA			X		
Integration	Configuration of Custom Dimension Values	Allow determination of user- and client-specific attributes via a configuration dialog.	SPA			X		
Integration	Session Handling	External sessions should match up with visits.	SPA, WA					X
Metrics	Custom Metrics	Support the definition of application-specific measurements.	WA					X
Metrics	Error Tracking	Tracking of errors with additional attributes and providing new reports based on it.	WA			X		
Metrics	Dialog Tracking	Tracking of dialog views (in reference to page views) and providing new reports based on it.	WA					X
Metrics	Advanced Goal Configuration	Allow the combination of multiple rules to trigger a goal.	WA			X		
Reporting	Named Date Ranges	Support of saving and selecting of custom data ranges under a given name.	WA		X			
Reporting	Click Path Report	Provide a report that shows all page views before and after a defined page	WA				X	
Reporting	Funnel Report	Allow the configuration of funnels with alternative steps or an undefined sequence of steps.	WA				X	
Reporting	Overlay Report for Dialogs	Enhance the SPA to show a read-only view of any dialog. Modify the overlay report to visualize events.	WA, SPA			X		
Reporting	Enhance Page Titles Report	The "Page Titles" report should offer the same functionality as the "Pages" report.	WA			X		

Rate each enhancement with either -2, -1, 0, 1 or 2 points.

No opinion/preference = 0 points



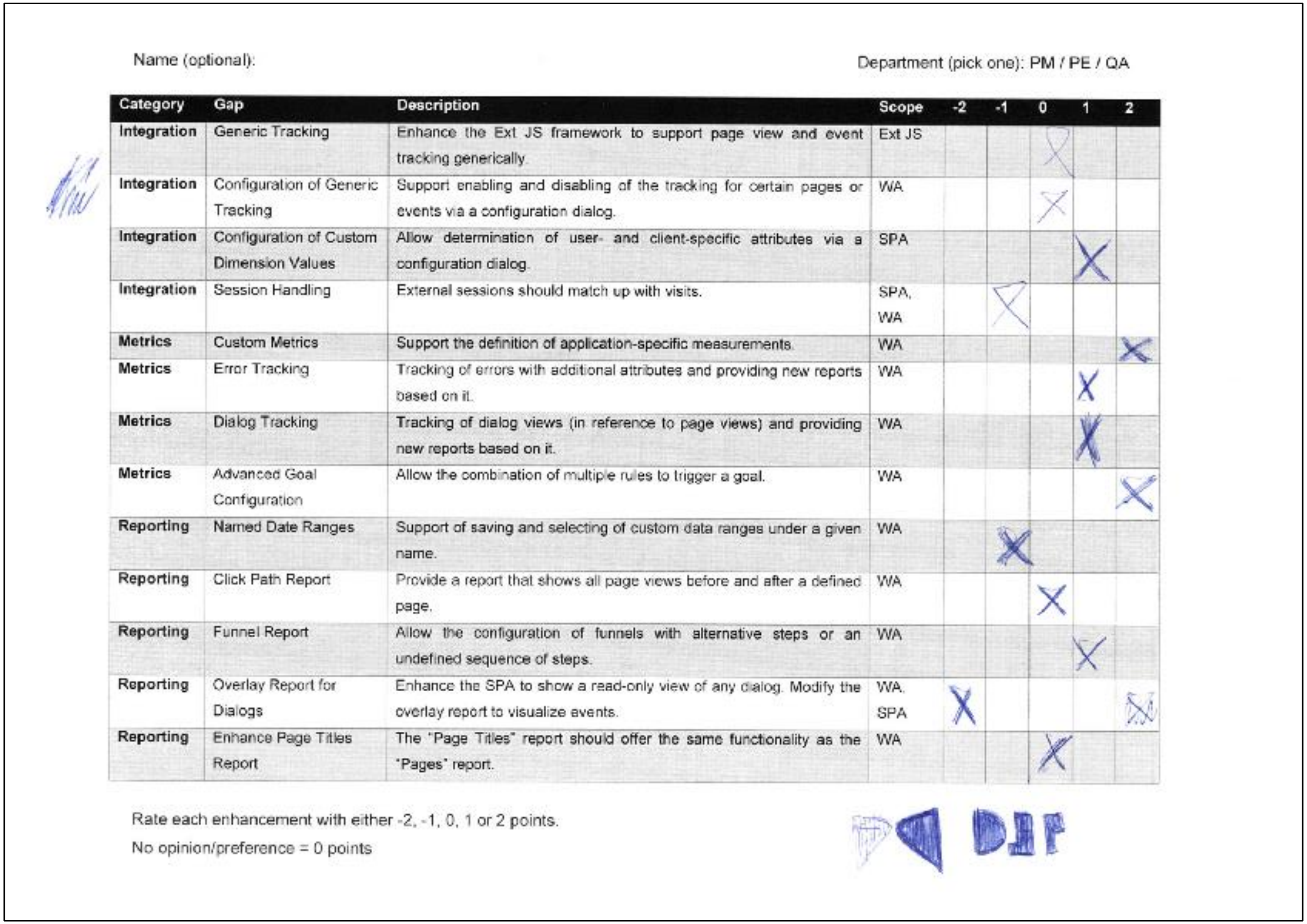
Name (optional):

Department (pick one): [✓ PM] / PE / QA

Category	Gap	Description	Scope	-2	-1	0	1	2
Integration	Generic Tracking	Enhance the Ext JS framework to support page view and event tracking generically.	Ext JS					X
Integration	Configuration of Generic Tracking	Support enabling and disabling of the tracking for certain pages or events via a configuration dialog.	WA			X		
Integration	Configuration of Custom Dimension Values	Allow determination of user- and client-specific attributes via a configuration dialog.	SPA				X	
Integration	Session Handling	External sessions should match up with visits.	SPA, WA	X				
Metrics	Custom Metrics	Support the definition of application-specific measurements.	WA		X			
Metrics	Error Tracking	Tracking of errors with additional attributes and providing new reports based on it.	WA					X
Metrics	Dialog Tracking	Tracking of dialog views (in reference to page views) and providing new reports based on it.	WA			X		
Metrics	Advanced Goal Configuration	Allow the combination of multiple rules to trigger a goal.	WA				X	
Reporting	Named Date Ranges	Support of saving and selecting of custom data ranges under a given name.	WA			X		
Reporting	Click Path Report	Provide a report that shows all page views before and after a defined page.	WA		X			
Reporting	Funnel Report	Allow the configuration of funnels with alternative steps or an undefined sequence of steps.	WA			X		
Reporting	Overlay Report for Dialogs	Enhance the SPA to show a read-only view of any dialog. Modify the overlay report to visualize events.	WA, SPA		X			
Reporting	Enhance Page Titles Report	The "Page Titles" report should offer the same functionality as the "Pages" report.	WA				X	

Rate each enhancement with either -2, -1, 0, 1 or 2 points.

No opinion/preference = 0 points



Name (optional):

Department (pick one): PM / PE / QA

Category	Gap	Description	Scope	-2	-1	0	1	2
Integration	Generic Tracking	Enhance the Ext JS framework to support page view and event tracking generically.	Ext JS			X		
Integration	Configuration of Generic Tracking	Support enabling and disabling of the tracking for certain pages or events via a configuration dialog.	WA			X		
Integration	Configuration of Custom Dimension Values	Allow determination of user- and client-specific attributes via a configuration dialog.	SPA				X	
Integration	Session Handling	External sessions should match up with visits.	SPA, WA		X			
Metrics	Custom Metrics	Support the definition of application-specific measurements.	WA					X
Metrics	Error Tracking	Tracking of errors with additional attributes and providing new reports based on it.	WA				X	
Metrics	Dialog Tracking	Tracking of dialog views (in reference to page views) and providing new reports based on it.	WA				X	
Metrics	Advanced Goal Configuration	Allow the combination of multiple rules to trigger a goal.	WA					X
Reporting	Named Date Ranges	Support of saving and selecting of custom data ranges under a given name.	WA		X			
Reporting	Click Path Report	Provide a report that shows all page views before and after a defined page.	WA			X		
Reporting	Funnel Report	Allow the configuration of funnels with alternative steps or an undefined sequence of steps.	WA				X	
Reporting	Overlay Report for Dialogs	Enhance the SPA to show a read-only view of any dialog. Modify the overlay report to visualize events.	WA, SPA	X				
Reporting	Enhance Page Titles Report	The "Page Titles" report should offer the same functionality as the "Pages" report.	WA			X		

Rate each enhancement with either -2, -1, 0, 1 or 2 points.

No opinion/preference = 0 points

Table 4 shows a summary of the votes for each suggested enhancement. The table also contains the calculated result and the consequential rank.

Category	Gap	-2	-1	0	1	2	Result	Rank
Integration	Generic Tracking			1	4	4	12	**2**
Integration	Configuration of Generic Tracking		2	3	3	1	3	9
Integration	Configuration of Custom Dimension Values		1	2	6		5	5
Integration	Session Handling	1	1	2	2	3	5	5
Metrics	Custom Metrics		1		2	6	13	**1**
Metrics	Error Tracking			2	5	2	9	**3**
Metrics	Dialog Tracking		1	1	5	2	8	**4**
Metrics	Advanced Goal Configuration	1		5	1	2	3	9
Reporting	Named Date Ranges		4	3	2		-2	11
Reporting	Click Path Report		2	2	4	1	4	7
Reporting	Funnel Report		1	3	5		4	7
Reporting	Overlay Report for Dialogs	1	4	2	1	1	-3	13
Reporting	Enhance Page Titles Report	1	1	6	1		-2	11

Table 4: Summary of votes for the suggested enhancements

APPENDIX B - Survey

Survey Items

Items marked with an asterisk were not used in the evaluation after performing a factor analysis.

Variable	Code	Measure	Loading
Confirmation	CF1	My experience with using Piwik was better than I expected.	0.887
	CF2	The functionality provided by Piwik was better than I expected.	0.582
	CF3	Overall, most of my expectations from using Piwik were confirmed.	0.828
Perceived Usefulness	PU1	Using the Custom Metrics functionality in Piwik will increase my productivity.	0.506
	PU2	Using the Custom Metrics functionality in Piwik will improve my efficiency.	0.443
	PU3	Using the Custom Metrics functionality in Piwik will enhance my effectiveness.	0.446
	PU4	Overall, the Custom Metrics functionality in Piwik is useful.	0.622
Satisfaction		How do you feel about your overall experience of using Piwik so far?	
	SF1	Very satisfied (1) / Very dissatisfied (7)	0.702
	SF2	Very pleased (1) / Very displeased (7)	1.034
	SF3	Very happy (1) / Very frustrated (7)	1.004
	SF4	Absolutely delighted (1) / Absolutely terrible (7)	0.658
Continuance Intention	CI1	I intend to continue using Piwik rather than to discontinue its use.	0.764
	CI2	My intentions are to continue using Piwik rather than to use any alternative means.	0.997
	CI3*	If I could, I would like to discontinue my use of Piwik.	0.234*

Table 5: Statements used for measuring the constructs (Bhattacherjee, 2001)

Code	Question
AGE	What is your age?
SEX	What is your gender?
DEP	To which department do you belong?
TRA	Have you been part of a training session in September?
TIM	How many times have you used Piwik for C1DEV?
DUR	How long have you been using Piwik for C1DEV in total?

Table 6: Demographic questions used in the questionnaire

Responses

ID	CF1	PU1	CI1	CF2	PU2	CI2	CF3	PU3	CI3	PU4	SF1	SF2	SF3	SF4
1	3	2	2	3	2	2	3	2	4	2	2	2	2	3
2	4	3	2	4	4	2	3	4	4	2	4	4	4	4
3	6	2	1	2	2	1	4	1	5	1	2	2	2	2
4	6	6	1	5	4	1	2	2	5	3	4	4	4	4
5	3	4	2	2	3	2	2	2	5	2	2	3	3	2
6	3	1	1	3	1	1	1	2	5	1	3	3	3	3
7	2	4	1	2	4	3	3	1	5	1	3	2	2	2
8	2	4	1	3	3	4	3	4	4	2	2	4	4	2
9	3	3	3	3	3	2	3	3	4	2	3	4	4	3
10	2	2	1	2	2	1	2	2	4	1	2	2	2	2
11	2	3	2	2	2	2	2	2	1	2	2	2	2	2
12	2	2	2	2	2	2	2	2	4	2	2	2	2	2
13	3	4	4	2	3	4	3	2	4	2	3	4	3	3
14	1	2	3	1	2	2	2	1	6	1	2	2	3	2
15	6	5	4	5	5	4	7	4	5	2	5	4	4	4
16	3	1	2	2	2	2	1	3	5	2	2	2	1	2
17	2	2	3	2	4	3	1	3	4	3	4	4	3	3
18	4	2	4	3	1	3	4	2	4	2	3	3	2	3
19	2	2	4	3	3	4	2	4	5	3	4	3	3	4
20	1	2	2	1	1	2	1	2	7	1	1	2	2	1
21	3	5	3	2	4	4	3	5	4	3	3	3	3	3
22	4	5	4	5	6	5	4	3	4	4	3	4	3	4
23	6	6	6	5	7	5	5	6	3	5	4	3	4	4
24	2	2	3	1	1	3	3	2	5	2	3	3	3	3
25	3	4	2	4	4	2	1	3	2	3	4	4	4	4
26	2	2	3	1	2	2	1	1	5	1	2	3	3	3
27	2	3	3	3	2	3	1	1	4	2	4	4	3	3
28	3	4	4	2	3	4	2	4	4	3	4	4	4	4
29	4	2	3	3	1	3	2	2	3	2	3	3	3	4
30	2	2	3	3	2	3	1	2	5	2	2	3	3	2
31	5	4	4	4	5	4	4	5	4	4	4	3	3	4
32	2	2	2	2	1	1	1	1	6	1	3	2	3	3

Table 7: Unprocessed survey results for all items

ID	AGE	Age	Sex	Department	TIM	Times	Duration	Training
1	41	40 to 44	Male	Product Management	3	Three to five times	30 min to less than 1h	Yes
2	33	30 to 34	Male	Software Engineering	0	Never	Less than 10 min	No
3	40	40 to 44	Male	Software Engineering	2	Twice	10 min to less than 20 min	No
4	36	35 to 39	Female	Quality Assurance	5	Three to five times	1h to less than 2h	No
5	45	45 to 49	Male	Product Management	5	Three to five times	1h to less than 2h	Yes
6	37	35 to 39	Male	Sales	0	Never	Less than 10 min	No, but I watched the recording afterwards.
7	29	25 to 29	Male	Software Engineering	1	Once	Less than 10 min	No
8	33	30 to 34	Male	Product Management	7	Five to ten times	10 min to less than 20 min	Yes
9	32	30 to 34	Female	Software Engineering	2	Twice	20 min to less than 30 min	Yes
10	38	35 to 39	Male	Software Engineering	30	More than ten times	1h to less than 2h	Yes
11	47	45 to 49	Male	Product Management	2	Twice	1h to less than 2h	No, but I watched the recording afterwards.
12	34	30 to 34	Female	Product Management	0	Never	Less than 10 min	Yes
13	26	25 to 29	Male	Software Engineering	2	Twice	10 min to less than 20 min	Yes
14	53	50 to 54	Male	Software Engineering	5	Three to five times	20 min to less than 30 min	No
15	34	30 to 34	Male	Product Management	1	Once	Less than 10 min	Yes
16	33	30 to 34	Male	Software Engineering	6	Five to ten times	1h to less than 2h	No
17	29	25 to 29	Female	Product Management	2	Twice	10 min to less than 20 min	Yes
18	43	40 to 44	Male	Software Engineering	4	Three to five times	20 min to less than 30 min	No, but I watched the recording afterwards.

ID	AGE	Age	Sex	Department	TIM	Times	Duration	Training
19	40	40 to 44	Male	Software Engineering	2	Twice	Less than 10 min	No, but I watched the recording afterwards.
20	29	25 to 29	Male	Product Management	10	Five to ten times	1h to less than 2h	Yes
21	54	50 to 54	Male	Product Management	2	Twice	10 min to less than 20 min	Yes
22	32	30 to 34	Male	Quality Assurance	1	Once	Less than 10 min	Yes
23	56	55 to 59	Male	Software Engineering	0	Never	Less than 10 min	No
24	32	30 to 34	Female	Product Management	1	Once	10 min to less than 20 min	Yes
25	33	30 to 34	Male	Quality Assurance	1	Once	Less than 10 min	No, but I watched the recording afterwards.
26	38	35 to 39	Male	Product Management	1	Once	10 min to less than 20 min	Yes
27	35	35 to 39	Female	Software Engineering	2	Twice	10 min to less than 20 min	Yes
28	41	40 to 44	Male	Software Engineering	2	Twice	10 min to less than 20 min	Yes
29	28	25 to 29	Male	Quality Assurance	3	Three to five times	10 min to less than 20 min	Yes
30	47	45 to 49	Male	Software Engineering	2	Twice	10 min to less than 20 min	Yes
31	44	40 to 44	Male	Software Engineering	2	Twice	10 min to less than 20 min	Yes
32	42	40 to 44	Male	Product Management	1	Once	Less than 10 min	No, but I watched the recording afterwards.

Table 8: Unprocessed survey results for all demographic questions

Item	Category	Frequency	Percent	Cum. Percent
Age	25 to 29	5	15,6%	15,6%
	30 to 34	9	28,1%	43,8%
	35 to 39	5	15,6%	59,4%
	40 to 44	7	21,9%	81,3%
	45 to 49	3	9,4%	90,6%
	50 to 54	2	6,3%	96,9%
	55 to 59	1	3,1%	100,0%
Sex	Female	6	18,8%	18,8%
	Male	26	81,3%	100,0%
Department	Product Management	12	37,5%	37,5%
	Quality Assurance	4	12,5%	50,0%
	Sales	1	3,1%	53,1%
	Software Engineering	15	46,9%	100,0%
Training	No	7	21,9%	21,9%
	No, but I watched the recording afterwards.	6	18,8%	40,6%
	Yes	19	59,4%	100,0%
Times	Never	4	12,5%	12,5%
	Once	7	21,9%	34,4%
	Twice	11	34,4%	68,8%
	Three to five times	6	18,8%	87,5%
	Five to ten times	3	9,4%	96,9%
	More than ten times	1	3,1%	100,0%
Duration	Less than 10 min	10	31,3%	31,3%
	10 min to less than 20 min	12	37,5%	68,8%
	20 min to less than 30 min	3	9,4%	78,1%
	30 min to less than 1h	1	3,1%	81,3%
	1h to less than 2h	6	18,8%	100,0%

Table 9: Summary of demographic data of respondents

Linear Regression Models

```
lm(formula = perceivedUsefulness ~ confirmation, data = factors)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.98272 -0.44784 -0.08607  0.52686  1.41759 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -2.965e-16  1.527e-01   0.000 1.000000    
confirmation  6.092e-01  1.555e-01   3.917 0.000581 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8082 on 26 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.3711,	Adjusted R-squared:  0.3469 
F-statistic: 15.34 on 1 and 26 DF,  p-value: 0.0005807
```

Listing 4: Summary of the linear regression between confirmation and perceived usefulness (R Core Team, 2017)

```
lm(formula = continuanceIntention ~ perceivedUsefulness, data = 
factors)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.2132 -0.3934  0.1893  0.4726  1.0593 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)   
(Intercept)         0.005985   0.141519   0.042  0.96659   
perceivedUsefulness 0.511900   0.144116   3.552  0.00149 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7488 on 26 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.3267,	Adjusted R-squared:  0.3008 
F-statistic: 12.62 on 1 and 26 DF,  p-value: 0.001486
```

Listing 5: Summary of the regression between perceived usefulness and continuance intention (R Core Team, 2017)

APPENDIX C - Log File Analysis

Listing 6 shows the SQL statements used to extract the number of visits and the total duration time from the Piwik installation, which was populated with the content of the log files from the internal Piwik test deployment. The IP address of the author was excluded from the results. The IP addresses were retracted as they represent internal sensitive data.

```
-- ordered by times of usage
select inet_ntoa(conv(hex(location_ip),16,10)) as ip,
    count(*) as cnt,
    round(sum(visit_total_time)/60) as totalDuration
 from log_visit
 where location_ip is not null and
    inet_ntoa(conv(hex(location_ip),16,10)) not in ('0.0.0.1',
    @ownIpAddr)
 group by location_ip
 order by 2 desc;

-- ordered by total usage in minutes
select inet_ntoa(conv(hex(location_ip),16,10)) as ip,
    count(*) as cnt,
    round(sum(visit_total_time)/60) as totalDuration
 from log_visit
 where location_ip is not null and
    inet_ntoa(conv(hex(location_ip),16,10)) not in ('0.0.0.1',
    @ownIpAddr)
 group by location_ip
 order by 3 desc;
```

Listing 6: SQL statements used to extract usage data from the Piwik DB

Table 10 shows the answers of the respondents for the two demographic questions regarding the number of times Piwik has been used and total duration time of that usage. The 32 responses are aligned with the data extracted from the Apache log file of the internal Piwik test installation. This was done by sorting both data set by the “times” column.

ID	Times	Duration	LOG FILE Times	LOG FILE Duration
1	3	30 min to less than 1h	4	180
2	0	Less than 10 min		
3	2	10 min to less than 20 min	2	0
4	5	1h to less than 2h	11	204
5	5	1h to less than 2h	5	2
6	0	Less than 10 min		
7	1	Less than 10 min	1	18
8	7	10 min to less than 20 min	23	16820
9	2	20 min to less than 30 min	3	6
10	30	1h to less than 2h	89	18089
11	2	1h to less than 2h	3	14
12	0	Less than 10 min		
13	2	10 min to less than 20 min	2	8
14	5	20 min to less than 30 min	5	122
15	1	Less than 10 min	1	7
16	6	1h to less than 2h	14	9904
17	2	10 min to less than 20 min	2	3
18	4	20 min to less than 30 min	4	9
19	2	Less than 10 min	1	8
20	10	1h to less than 2h	69	3362
21	2	10 min to less than 20 min	1	15
22	1	Less than 10 min	1	27
23	0	Less than 10 min		
24	1	10 min to less than 20 min	1	4
25	1	Less than 10 min	1	0
26	1	10 min to less than 20 min	1	4
27	2	10 min to less than 20 min	1	2
28	2	10 min to less than 20 min	1	1
29	3	10 min to less than 20 min	4	29
30	2	10 min to less than 20 min	1	0
31	2	10 min to less than 20 min	1	28
32	1	Less than 10 min	1	13

Table 10: Survey results compared to web analytics report based on the log file

GLOSSARY

AJAJ	Asynchronous JavaScript and JSON
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
ASP	Application Service Provider
CD	Custom Dimensions
CF	Confirmation (ECM)
CI	Continuance Intention (ECM)
CM	Custom Metrics
CMS	Content Management System
CRM	Customer Relationship Management
DB	Database
DSM	DiskStation Manager
ECM	Expectation-Confirmation Model
ECT	Expectation-Confirmation Theory
EFF	Electronic Frontier Foundation
FOSS	Free and Open-Source Software
GA	Google Analytics
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure-as-a-Service
IDC	International Data Corporation
IP	Internet Protocol
IS	Information System
ISV	Independent Software Vendor
JS	JavaScript
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
MDI	Multiple Document Interface
MS	Microsoft
MVC	Model-View-Controller
NAS	Network Attached Storage
PaaS	Platform-as-a-Service
PDCA	Plan-Do-Check-Act

PHP	PHP: Hypertext Preprocessor
PII	Personally Identifiable Information
PU	Perceived Usefulness (ECM)
PwC	PricewaterhouseCoopers
RIA	Rich Internet Application
SaaS	Software-as-a-Service
SDI	Single Document Interface
SF	Satisfaction (ECM)
SMB	Small and Medium-Sized Businesses
SPA	Single Page Application
TCO	Total Cost of Ownership
TRM	Treasury and Risk Management
UI	User Interface
URL	Uniform Resource Locator
USP	Unique Selling Proposition
UX	User Experience
WA	Web Analytics
WAA	Web Analytics Association
WWW	World Wide Web
XML	Extensible Markup Language

LIST OF FIGURES

LIST OF TABLES

LISTINGS

REFERENCES

Adobe Corporate Communications. (2017). Flash & The Future of Interactive Content. Retrieved August 27, 2017, from https://blogs.adobe.com/conversations/2017/07/adobe-flash-update.html.

Adobe Systems Inc. (2017). Security updates for Adobe Flash Player. Retrieved August 27, 2017, from https://helpx.adobe.com/security/products/flash-player.html.

Alzahrani, H. (2016). A Brief Survey of Cloud Computing. In *Global Journal of Computer Science and Technology* (Vol. 16). USA: Global Journals Inc.

Atkinson, E. (2007). *Web analytics and think aloud studies in web evaluation: Understanding user experience*: London University.

Atterer, R., Wnuk, M., & Schmidt, A. (2006). Knowing the user's every move. In L. Carr, D. de Roure, A. Iyengar, C. Goble, & M. Dahlin (Eds.), *Proceedings of the 15th international conference on World Wide Web* (p. 203). New York, NY: ACM. https://doi.org/10.1145/1135777.1135811

Barrett, L. (2010). SaaS Market Growing by Leaps and Bounds: Gartner. Retrieved September 04, 2017, from http://www.datamation.com/entdev/article.php/3895101/SaaS-Market-Growing-by-Leaps-and-Bounds-Gartner.htm.

Becker, A. (2017). HeidiSQL [Software].

Bhattacherjee, A. (2001). Understanding Information Systems Continuance: An Expectation-Confirmation Model. *MIS Quarterly*, *25*(3). https://doi.org/10.2307/3250921

Bhuiyan, N., & Baghel, A. (2005). An overview of continuous improvement: From the past to the present. *Management Decision*, *43*(5), 761–771. https://doi.org/10.1108/00251740510597761

Bianchi, A. (2000). Upstarts: ASPs. Retrieved September 20, 2017, from https://www.inc.com/magazine/20000401/18093.html.

Bostock, M. (2017). A possible scenario of UK energy production and consumption in 2050. Retrieved October 05, 2017, from https://github.com/d3/d3-sankey/commits/master/img/energy.png.

Bright, P. (2017). Microsoft 4Q17: Office 365 revenue surpasses traditional licenses. Retrieved October 08, 2017, from https://arstechnica.com/information-technology/2017/07/microsoft-4q17-office-365-revenue-surpasses-traditional-licenses/.

BuiltWith.com. (2017). Analytics technologies Web Usage Statistics. Retrieved November 03, 2017, from https://trends.builtwith.com/analytics.

Burby, J., & Brown, A. (2006). *Web Analytics “Big Three” Definitions - Version 1.0*: Web Analytics Association. Retrieved from https://www.digitalanalyticsassociation.org/standards

Burby, J., & Brown, A. (2007). *Web Analytics Definitions - Version 4.0*: Web Analytics Association. Retrieved from https://www.digitalanalyticsassociation.org/standards

Callender, C., Marshall, B., Cardon, P., & Patel, N. (2015). Obstacles to the adoption of cloud computing: Best practicies in technology and communication. In *Issues in Information Systems* (16:2, pp. 133–139).

Chitkara, R., & McCaffrey, M. (2016a). Global 100 Software Leaders. Retrieved September 02, 2017, from https://www.pwc.com/gx/en/industries/technology/publications/global-100-software-leaders/the-big-picture.html#4.

Chitkara, R., & McCaffrey, M. (2016b). SaaS revenues of Top 50 software companies. Retrieved September 02, 2017, from https://www.pwc.com/gx/en/industries/technology/publications/global-100-software-leaders/explore-the-data.html.

Clifton, B. (2012). *Advanced web metrics with Google Analytics* (3rd ed.). Indianapolis: John Wiley & Sons.

Cowell, J. (1996). *Essential Delphi 2.0 Fast: How to Develop Applications in Delphi 2.0*. *Essential Series*. London: Springer London.

DeLone, W. H., & McLean, E. R. (2014). The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. *Journal of Management Information Systems*, *19*(4), 9–30. https://doi.org/10.1080/07421222.2003.11045748

Doyle, B., & Lopes, C. V. (2008). *Survey of Technologies for Web Application Development*. Retrieved from http://arxiv.org/pdf/0801.2618v1

Fain, Y., Rasputnis, V., Tartakovsky, A., & Gamov, V. (2014). *Enterprise web development: Building HTML5 applications: from desktop to mobile* (1. ed.). Beijing: O’Reilly.

Farrell, J., & Nezlek, G. S. (2007). Rich Internet Applications: The Next Stage of Application Development. In V. Lužar-Stiffler (Ed.), *29th International Conference on Information Technology Interfaces, 2007. ITI 2007 ; 25 - 28 June 2007, Cavtat/Dubrovnik, Croatia* (pp. 413–418). Zagreb: SRCE. https://doi.org/10.1109/ITI.2007.4283806

Fink, G. (2014). *Pro Single Page Application Development: Using Backbone*. Dordrecht: Springer.

Fowley, F., & Pahl, C. (2016). Cloud Migration Architecture and Pricing - Mapping a Licensing Business Model for Software Vendors to a SaaS Business Model. In *European Conference on Service-Oriented and Cloud Computing ESOCC - CloudWays’2016 Workshop.* Springer.

Fraternali, P., Rossi, G., & Sánchez-Figueroa, F. (2010). Rich Internet Applications. *IEEE Internet Computing*, *14*(3), 9–12. https://doi.org/10.1109/MIC.2010.76

Garrett, J. J. (2005). Ajax: A new approach to web applications. Retrieved June 13, 2017, from http://web.archive.org/web/20080702075113/http://www.adaptivepath.com/ideas/essays/archives/000385.php.

Golfarelli, M., Maio, D., & Rizzi, S. (1998). The Dimensional Fact Model: A conceptual model for data warehouses. *International Journal of Cooperative Information Systems*, *07*(02n03), 215–247. https://doi.org/10.1142/S0218843098000118

Google Inc. (2017). Custom dimensions & metrics: Include non-standard data in your reports. Retrieved November 18, 2017, from https://support.google.com/analytics/answer/2709828?hl=en.

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M.,. . . Pirahesh, H. (1997). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Mining and Knowledge Discovery*, *1*(1), 29–53. https://doi.org/10.1023/A:1009726021843

Guzdial, M., Santos, P., Badre, A., Hudson, S., & Gray, M. (1994). *Analyzing and visualizing log files: A computational science of usability*. Atlanta, Ga.: Graphics, Visualization & Usability Center, Georgia Institute of Technology.

Hasan, L., Morris, A., & Probets, S. (2009). Using Google Analytics to Evaluate the Usability of E-Commerce Sites. In M. Kurosu (Ed.), *Lecture Notes in Computer Science: Vol. 5619. Human centered design. First international conference, HCD 2009, held as part of HCI International 2009, San Diego, CA, USA, July 19 - 24, 2009 ; proceedings* (pp. 697–706). Berlin: Springer.

Hassler, M. (2017). *Digital und Web Analytics: Metriken auswerten, Besucherverhalten verstehen, Website optimieren*. *mitp Business*: mitp Verlag.

Holmes, S. (2016). *Getting MEAN With Mongo, Express, Angular, and Node*. Shelter Island, NY: Manning Publications. Retrieved from http://proquest.tech.safaribooksonline.de/9781617292033

Hossain, A., & Shirazi, F. (2015). Cloud Computing: A Multi-tenant Case Study. In M. Kurosu (Ed.), *Lecture notes in computer science Information systems and applications, incl. Internet/web, and HCI: Vol. 9171. Human-computer interaction. 17th international conference, HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015; proceedings* (Vol. 9171, pp. 178–189). Cham: Springer. https://doi.org/10.1007/978-3-319-21006-3_18

Hossain, M. A., & Quaddus, M. (2012). Expectation–Confirmation Theory in Information System Research: A Review and Analysis. In Y. K. Dwivedi, M. R. Wade, & S. L. Schneberger (Eds.), *Integrated Series in Information Systems: Vol. 28. Information systems theory.*

Explaining and predicting our digital society, Vol. 1 (Vol. 28, pp. 441–469). New York, NY: Springer Science+Business Media LLC. https://doi.org/10.1007/978-1-4419-6108-2_21

Huiyan. (2017). Anonymize IP Geolocation Accuracy Impact Assessment. Retrieved November 07, 2017, from https://www.conversionworks.co.uk/blog/2017/05/19/anonymize-ip-geo-impact-test/.

IDC. (2013). The Coming of the 3rd Platform and What This Means for Software Business Models. Retrieved September 03, 2017, from https://www.flexerasoftware.com/producer/resources/research/research-idm-idc-thirdplatform.html.

IDC. (2017a). Public Cloud Services Spending in Central and Eastern Europe to Almost Double Between 2017 and 2021, According to IDC. Retrieved September 03, 2017, from http://www.idc.com/getdoc.jsp?containerId=prCEMA42967317.

IDC. (2017b). Worldwide Public Cloud Services Spending Forecast to Reach $266 Billion in 2021, According to IDC. Retrieved September 03, 2017, from http://www.idc.com/getdoc.jsp?containerId=prUS42889917.

James, K. L. (2008). *Software Engineering*: Phi Learning.

Jeong, B.-K., & Stylianou, A. C. (2010). Market reaction to application service provider (ASP) adoption: An empirical investigation. *Information & Management*, *47*(3), 176–187. https://doi.org/10.1016/j.im.2010.01.007

Kamiński, S. (2017). Why use the Click Path feature before creating your funnel? Retrieved October 05, 2017, from https://help.piwik.pro/user-guides/funnels/why-use-the-click-path-feature-before-creating-your-funnel.

Kang, Y. S., Hong, S., & Lee, H. (2009). Exploring continued online service usage behavior: The roles of self-image congruity and regret. *Computers in Human Behavior*, *25*(1), 111–122. https://doi.org/10.1016/j.chb.2008.07.009

Kaushik, A. (2007a). *Web analytics: An hour a day*. *Serious skills*. Indianapolis, Indiana: Wiley.

Kaushik, A. (2007b). Kick Butt With Internal Site Search. Retrieved November 03, 2017, from https://www.kaushik.net/avinash/kick-butt-with-internal-site-search-analytics/.

Kaushik, A. (2010). *Web analytics 2.0: The art of online accountability & science of customer centricity*. *Serious skills*. Indianapolis, Indiana: Wiley.

Kimball, R., & Ross, M. (2009). *The data warehouse toolkit: The complete guide to dimensional modeling* (2. ed.). New York, NY: Wiley.

Kony, Inc. (2017). Custom Reporting: Metrics, Reports, and Dashboard Guide. Retrieved November 18, 2017, from http://docs.kony.com/7_x_PDFs/mobilefabric/custom_metrics_reports.pdf.

Kozlov, S. (2016). Comparison of JS Frameworks: Angular.js vs React.js vs Ember.js. Retrieved October 02, 2017, from https://www.romexsoft.com/blog/js-frameworks-comparison/.

Laforge, A. (2017). Saying goodbye to Flash in Chrome. Retrieved August 27, 2017, from https://www.blog.google/products/chrome/saying-goodbye-flash-chrome/.

Lawton, G. (2008). Moving the OS to the Web. *Computer*, *41*(3), 16–19. https://doi.org/10.1109/MC.2008.94

Lee, J., Podlaseck, M., Schonberg, E., Hoch, R., & Gomory, S. (2000). Understanding Merchandising Effectiveness of Online Stores. *Electronic Markets*, *10*(1), 20–28. https://doi.org/10.1080/10196780050033944

Lennon, J. (2010). Compare JavaScript frameworks: An overview of the frameworks that greatly enhance JavaScript development. Retrieved October 02, 2017, from https://www.ibm.com/developerworks/library/wa-jsframeworks/index.html.

libhunt.com. (2017). Self Hosted Analytics. Retrieved November 04, 2017, from https://selfhosted.libhunt.com/categories/1647-analytics.

Lo, J. (2015). Pricing benchmark results. Retrieved September 02, 2017, from http://www.pwc.com/us/en/technology/publications/software-pricing-benchmark-results.html.

Luit Infotech Private Limited. (2013). Difference between the ASP model and the SaaS model. Retrieved October 07, 2017, from http://www.luitinfotech.com/kc/saas-asp-difference.pdf.

Maass, M., Wichmann, P., Pridöhl, H., & Herrmann, D. (2017). PrivacyScore: Improving Privacy and Security via Crowd-Sourced Benchmarks of Websites. In E. Schweighofer, H. Leitold, A. Mitrakas, & K. Rannenberg (Eds.), *Lecture Notes in Computer Science: Vol. 10518. Privacy Technologies and Policy. 5th Annual Privacy Forum, APF 2017, Vienna, Austria, June 7-8, 2017, Revised Selected Papers* (Vol. 10518, pp. 178–191). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-67280-9_10

MacDonald, M. (2006). *Pro.NET 2.0 Windows Forms and custom controls in C#. The expert's voice in.NET*. Berkeley Calif.: Apress. Retrieved from http://dx.doi.org/10.1007/978-1-4302-0110-6

Mäkilä, T., Järvi, A., Rönkkö, M., & Nissilä, J. (2010). How to Define Software-as-a-Service – An Empirical Study of Finnish SaaS Providers. In P. Tyrväinen (Ed.), *Lecture Notes in Business Information Processing: Vol. 51. Software Business. First International Conference, ICSOB 2010, Jyväskylä, Finland, June 21-23, 2010. Proceedings* (Vol. 51, pp. 115–124). Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg. https://doi.org/10.1007/978-3-642-13633-7_10

Mangiuc, D. M. (2009). Software: From product to service: The evolution of a model. *Annales Universitatis Apulensis Series Oeconomica*, *1*(11), 88–97. Retrieved from http://EconPapers.repec.org/RePEc:alu:journl:v:1:y:2009:i:11:p:8

Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing*. *NIST special publication: 800-145*. Gaithersburg, MD: Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology.

Merelo-Guervós, J. J., Castillo, P. A., Laredo, J.L.J., Mora García, A., & Prieto, A. (2008). Asynchronous distributed genetic algorithms with Javascript and JSON. In *IEEE Congress on Evolutionary Computation, 2008* (pp. 1372–1379). Piscataway, NJ: IEEE. https://doi.org/10.1109/CEC.2008.4630973

Mesbah, A., & van Deursen, A. (2007). Migrating Multi-page Web Applications to Single-page AJAX Interfaces. *Proceedings of the 11th European Conference on Software Maintenance and Reengineering (CSMR'07), IEEE Computer Society*.

Microsoft Edge Team. (2017). The End of an Era – Next Steps for Adobe Flash. Retrieved August 27, 2017, from https://blogs.windows.com/msedgedev/2017/07/25/flash-on-windows-timeline/.

Mikowski, M. S., Powell, J. C., & Benson, G. D. (2014). *Single page web applications: Javascript end-to-end*. Shelter Island, NY: Manning.

Miller, S. A. (2012). *Piwik web analytics essentials: A complete guide to tracking visitors on your web sites, e-commerce shopping carts, and apps using Piwik web analytics*. *Community experience distilled*. Birmingham, UK: Packt Pub.

Moen, R., & Norman, C. (2006). Evolution of the PDCA Cycle. Retrieved November 28, 2017, from http://www.uoc.cw/financesite/images/stories/NA01_Moen_Norman_fullpaper.pdf.

Morris, S. (1994). *Object-Oriented Programming under Windows*: Butterworth-Heinemann Ltd.

Motal, J. (2011). Microsoft Office 365 Launching June 28. Retrieved October 08, 2017, from https://www.pcmag.com/article2/0,2817,2386447,00.asp.

Mozilla Foundation, & Mozilla Corporation. (2017). Firefox [Software].

Nintendo of Europe GmbH. (2017). Nintendo UK's official site. Retrieved November 16, 2017, from http://www.nintendo.co.uk/.

Noda, T., & Helwig, S. (2005). Rich internet applications: Technical Comparison and Case Studies of AJAX, Flash, and Java based RIA. *UW E-Business-Consortium Opinion Papers*.

Oliver, R. L. (1980). A Cognitive Model of the Antecedents and Consequences of Satisfaction Decisions. *Journal of Marketing Research*, *17*(4), 460. https://doi.org/10.2307/3150499

Pennington, H., Suárez-Alvarez, M., Pastore, G. de, Persch, C., Perberos, & MATE developers. (2014). MATE-Terminal [Software].

Petrlic, R., & Sorge, C. (2017). *Datenschutz: Einführung in technischen Datenschutz, Datenschutzrecht und angewandte Kryptographie*. Wiesbaden: Springer Vieweg. Retrieved from http://dx.doi.org/10.1007/978-3-658-16839-1

Piwik Core Team. (2008). Hello the web analytics world! Retrieved November 04, 2017, from https://piwik.org/blog/2008/01/hello-the-web-analytics-world/.

Piwik.org. (2017a). Piwik [Software].

Piwik.org. (2017b). Users Flow. Retrieved October 05, 2017, from https://piwik.org/docs/users-flow/.

Pugh, P. (2014). The Journey to SaaS Profitability. Retrieved September 02, 2017, from https://www.pwc.com/us/en/technology/publications/saas-profitability.html.

R Core Team. (2017). R [Software].

Rambeau, M. (2017). Best of JavaScript: UI Framework. Retrieved October 02, 2017, from https://bestof.js.org/tags/framework.

Reval Inc. (2017a). Reval [Software].

Reval Inc. (2017b). Reval Client Community. Retrieved November 02, 2017, from https://www.reval.com/clients/.

Reval Inc. (2017c). Treasury Software for Corporates. Retrieved November 02, 2017, from https://www.reval.com/corporates/.

Rogers, Y., Sharp, H., & Preece, J. (2011). *Interaction design: Beyond human-computer interaction* (3nd ed.). Chichester: Wiley.

Russel, R. (2014). iptables [Software].

Saito, T., Hosoya, R., Yasuda, K., Takahashi, K., Tsunoda, Y., Tanabe, K.,. . . Yashiro, S. (2017). Proposal and Implementation of a Countermeasure Against Event Tracking on the Web. In L. Barolli & T. Enokido (Eds.), *Advances in Intelligent Systems and Computing: v.612. Innovative Mobile and Internet Services in Ubiquitous Computing. Proceedings of the 11th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2017)* (pp. 435–444). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-61542-4_41

Smedberg, B. (2017). Plugin Roadmap for Firefox. Retrieved August 27, 2017, from https://developer.mozilla.org/en-US/docs/Plugins/Roadmap.

Sviatoslav, A. (2017). The Best JS Frameworks for Front End. Retrieved October 02, 2017, from https://rubygarage.org/blog/best-javascript-frameworks-for-front-end.

Synology Inc. (2017). DiskStation Manager [Software].

Tomilenko, I. (2017). Custom contexts. Retrieved November 19, 2017, from https://github.com/snowplow/snowplow/wiki/Custom-contexts.

W3Techs. (2017a). Usage Statistics and Market Share of Piwik for Websites, November 2017. Retrieved November 03, 2017, from https://w3techs.com/technologies/details/ta-piwik/all/all.

W3Techs. (2017b). Usage Statistics and Market Share of Traffic Analysis Tools for Websites, November 2017. Retrieved November 03, 2017, from https://w3techs.com/technologies/overview/traffic_analysis/all.

W3Techs. (2017c). Usage Survey of Server Locations broken down by Traffic Analysis Tools. Retrieved November 03, 2017, from https://w3techs.com/technologies/cross/server_location/traffic_analysis.

Wan, Z., & Wang, P. (2014). A Survey and Taxonomy of Cloud Migration. In *2014 International Conference on Service Sciences (ICSS). 22 - 23 May 2014, Wuxi, Jiangsu, China* (pp. 175–180). Piscataway, NJ: IEEE. https://doi.org/10.1109/ICSS.2014.46

Weischedel, B., & Huizingh, E. K. R. E. (2006). Website optimization with web metrics. In M. S. Fox (Ed.), *Proceedings of the 8th international conference on electronic commerce. The new e-commerce innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet* (p. 463). New York, NY: ACM. https://doi.org/10.1145/1151454.1151525

Wohl, A. D. (2008). *Succeeding at SaaS: Computing in the cloud*. Narberth: Wohl Assoc.

Zheng, G., & Peltsverger, S. (2015). Web Analytics. In M. Khosrow-Pour (Ed.), *Encyclopedia of information science and technology* (3rd ed., pp. 7674–7683). Hershey, Pa.: Information Science Reference.