A Usage-based Data Extraction Framework for Cloud-Based Application
An Human-Computer Interaction approach

Manoj Kesavulu, Markus Helfert and Marija Bezbradica
Lero – Irish Software Research Organization
School of Computing, Dublin City University, Dublin, Ireland
{*manoj.kesavulu, markus.helfert, marija.bezbradica*}@*lero.ie*

Keywords: Usage Data, Data Extraction, Application Features, Functionality, Cloud, Software-as-a-Service (SaaS), User Behaviour

Abstract: Features or functionalities provided by cloud-based applications are accessed by users through various interfaces such as web browser, mobile app, and command line interface. Yet for monitoring cloud-based applications, software developers and researchers have focused on web browsers. Software updates are provided for such applications based on the data acquired from the cloud monitoring components but usage data of the cloud application features are difficult to extract in a cloud environment as the usage data is spread across the interfaces on the front-end and the back-end. In this paper, we focus on the usage of the cloud application features from the user perspective and how to extract these data in a cloud environment. We define six criteria for the user-level usage data, analyse the existing usage data extraction techniques and propose a usage data extraction framework adhering to the defined criteria.

1 INTRODUCTION

Software monitoring is a well-matured field where applications post-deployment are monitored for their usage data to understand how the applications are used by end-users (Pachidi et al., 2014). The usage data is generated after the applications are deployed and being used in real-time by the end-users. This usage data deems necessary for software developers and architects to provide updates for the applications. The majority of the research in the software monitoring domain focus on collecting software operational data, event logs, resource usage monitoring in order to identify performance issues, errors and other usability problems (Fabijan et al., 2015).

On the other hand, web usage mining field has seen a lot of development (Gasparetti, 2016; Ghezzi et al., 2014). However, many lessons can be learned when compared to a cloud-based application, the cloud application can also be accessed by a smartphone for example. Hence, usage data also lies in this device and the methods and techniques for analysing web usage by website visitors has significant differences, compared to how mobile applications are used by the user to access the cloud-based applications. The techniques that are used in web usage mining (and other related domains) need to be revised if they are intended for mining usage data in a cloud environment.

Cloud-based software applications deployed over the Internet offer various advantages over traditional software such as reduced time to benefit, scalability, access through various interfaces and so on. One of the main advantage of using cloud applications is the option to an end-user to access the cloud application using multiple devices (interfaces). So, it is critical for any usage monitoring component to consider all these interfaces as usage data sources. In this paper, we focus on usage data extraction in Software-as-a-Service (SaaS) layer of the cloud. We aim to create a framework that can be used to extract the usage data that is generated in the cloud system along with the interfaces used by the end-user to access the cloud system. This will help to understand the features that are important for the user and critical to the system. For example, the user might use a feature very rarely but it might still be critical to him/her such as an online bill payment feature where the user might use it once a month but still is a critical feature for him/her. This cannot be determined by analysing the frequency of usage of the feature by the user. Analysing and understanding the usage data from the users perspective can be used by the software developers and software architects to determine how much development time, development cost to allocate and spend for which features of the cloud application before rolling out new updates. As a part of our future work, we aim to build the usage data extraction artefact and follow the evaluation approach using Design Science Research (Helfert et al., 2012).

The remainder of this paper has the following structure: We provide background on SaaS development lifecycle and usage data in the cloud in Section 2. In Section 3, we discuss the criteria that we have identified and provide justification while analysing the literature based on the criteria. In Section 4, we propose the usage data extraction framework that adheres to the criteria discussed in the previous section. Then in Section 5, we provide the conclusion and show direction for the future work.

2 BACKGROUND

2.1 SaaS Software Development

Traditional IT (Information Technology) aligns resources according to the way the applications are deployed within dedicated infrastructure and data storage to fulfil business requirements. Cloud computing has emerged as a computing paradigm with benefits such as high scalability, reduced IT costs, on demand self-service, pay-as-you-go price models, elasticity in provision computing resources.

Cloud computing architecture defines three distinct services layers as shown in Figure 1:

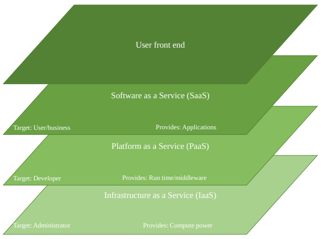
User front end
Software as a Service (SaaS)
Target: User/business
Provides: Applications
Platform as a Service (PaaS)
Target: Developer
Provides: Run time/middleware
Infrastructure as a Service (IaaS)
Target: Administrator
Provides: Compute power

Figure 1: Cloud Service Layers [Source: (Pallis, 2010)]

The three distinct layers of the cloud are as follows:

- Infrastructure as a Service (IaaS) offers computing resources, both physical and virtual, for processing and storage.
- Platform as a Service (PaaS) offers development environment for software developers to write their applications on a particular platform without worrying about the underlying hardware infrastructure.
- Software as a Service (SaaS) offers software applications that can be accessed and used by the end-users.

The focus of our research is to understand what features and functionalities are important to the end-user by analyzing the end-user usage of the applications deployed in a cloud environment. We consider SaaS layer of the cloud where the user uses various interfaces to access the cloud-based applications. The widely-adopted definition for SaaS cloud model is provided by National Institute of Standards and Technology (NIST) as The capability provided to the consumer is to use the providers applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface (Mell and Grance, 2011). In other words, SaaS applications are deployed on cloud infrastructure and are provided to end-users as a service over the Internet, the end-users can access these applications using various interfaces such as web browsers, mobile applications and command line interfaces. These applications are often monitored by Application Performance Monitoring (APM) tools to understand how much of the underlying resources are used by the application, errors, bugs, usability issues and to identify outdated services where architectural refactoring can be applied while migrations applications to the cloud (Kesavulu et al., 2017). These data will be analysed by the application developers to fix the errors and bugs, improve the application and rollout updates. This constitutes the software development cycle in a SaaS environment as shown in Figure 2. The software vendors are also interested in their customer behaviour to understand how end-users use the application. For this purpose, user behaviour knowledge is collected from analysing the users interaction with the web-browser while accessing the application in the form of click-streams (Pachidi et al., 2014; Wang et al., 2016).

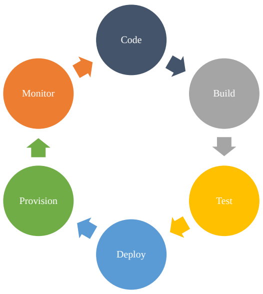

Figure 2: SaaS Software Development Lifecycle

2.2 Usage Data in Cloud

The rise of cloud computing and SaaS has eased the process of monitoring application usage as the applications are deployed on the cloud environment and provisioned to the end-user over the Internet as services. The cloud provider (vendor) provides APM tools (for example, CloudWatch (http://aws.amazon.com/cloudwatch/) in Amazon Web Services) to monitor the status of the deployed applications, the amount of resources used by the applications based on the agreement between cloud vendor and the application provider called Service Level Agreement (SLA). The application developers can also use various third-party monitoring tools such as New Relic (https://newrelic.com/), Binadox (https://www.binadox.com/salesforce-saas-monitoring/) and so on. But these tools mainly focus on monitoring application oriented usage such as measuring the number of users logged-in to the application, identifying rare logins, cloud resource usage, idle times, license types etc. Log files are analysed to derive models (Petruch et al., 2012). Understanding usage data of an application has various uses such as to personalise the application according to the end-users preferences (Yang et al., 2017), profiling users for security (Al-Bayati et al., 2016), improvement in marketing of software products (Bucklin and Sismeiro, 2009) and to analyse the performance of the application in the deployed environment for maintenance purposes (Petruch et al., 2012; Zaidman, 2010).

From the literature exploration, we see that the idea of monitoring user behaviour is to understand how users interact with the application and this is mainly done through analysing the clickstreams (Pachidi et al., 2014; Wang et al., 2016; Banerjee and Ghosh, 2001; Bucklin and Sismeiro, 2009). The authors (Cito et al., 2015) provide a high-level taxonomy of types of operation data:

1. Monitoring data (Operational application metadata, Collected from state-of-the-art APM tools)
   (a) Performance data service response times, database query times
   (b) Load data incoming request rate, server utilization
   (c) Costs data hourly cloud virtual machine costs, data transfer costs per 10,000 page views
   (d) User behaviour data - clickstreams
2. Production data
   (a) Data produced by SaaS application itself-placed orders, customer information

Consideration of user behaviour data only through clickstreams is mainly under the assumption that the end-user has access to the SaaS application only through a web-browser. Other interfaces such as mobile apps and command line interfaces are also used to access the application and these interfaces should be considered as sources for the extraction of the usage data from a cloud-based application.

3 USER-LEVEL USAGE DATA CRITERIA

The user-level usage data in the context of this paper is the usage data generated because of user interaction with the SaaS application using which we can determine which features are critical and important for the end-user. Such data in the cloud is spread across various interfaces such as Web browser, mobile apps and command line interfaces on the front-end and server and database on the back-end. In order to classify the usage data, we refer the work from (Pachidi et al., 2014) where the authors classify the usage data in a web-based system into six categories: (1) who is using the application; (2) Where the application being hosted; (3) What the end user does; (4) when the user performs the operation; (5) how long it takes to complete the operation; (6) other operation details such as errors, background tasks and number of records loaded. We add another category to this classification called user behaviour that contains clickstreams (from web-browser), view and focus (from mobile app) as a result of Human-Computer Interaction (HCI) between the end-user and the interfaces used to access the cloud-based application. The reason we refer to this classification is that SaaS applications are provisioned to the end-user over the Web and the rationale to add the new category (user behaviour) is because we focus on the usage data from the user perspective and the SaaS applications can be accessed through a mobile application in addition to a web-browser. We provide improved classification of the usage data as shown in Table 1.

Some research is based on understanding user behaviour of smartphone users, where the users are grouped based on their smartphone application usage behaviour (Zhao et al., 2016). The authors consider the recently used apps in the users smartphones and categorise the users into various types. But, the usage data is collected from an external source that is a Telecom company. These data is treated as biased and hence it is not reliable and not real-time. Though the authors provide a comprehensive explanation and detailed analysis steps on how the usage

Table 1: Usage Data Classification [Adapted from (Pachidi et al., 2014)]

1) Who is using the application
   a) User ID
   b) IP address
2) Where the application is being hosted
   a) Web server
   b) Database
3) What the end user does
   a) Application
   b) Page
   c) Method
   d) Function
   e) Button that is accessed
   f) Action that is performed
4) When the user performs the operation
   a) Date and time
   b) Session ID
5) How long it takes to complete the operation
   a) Duration
   b) Query duration
6) Other operation details
   a) Errors
   b) Background tasks
   c) Number of records loaded
7) User behaviour
   a) Clickstream
   b) View
   c) Focus

data was analysed to determine the user behaviour, since the usage data do not adhere to the criteria. It is uncertain whether their method applies to wide range of users globally. The authors in (Cito et al., 2015) aim at integrating runtime monitoring data from production deployments of the software into the development tools to enable tighter feedback loops. The authors call this notion as Feedback-Driven Development (FDD). The authors argue that all the necessary data required is readily available in a cloud environment through built-in cloud monitoring APIs or through external APM solutions. Here, the authors purely rely on built-in or external tools to collect the data. But these tools consider only clickstreams for analysing the user behaviour. Since the end-users access the cloud applications through mobile apps and command line interfaces in addition to a web-browser, the usage data collected here may be treated as incomplete.

Considering the characteristics of the usage data to be extracted from a cloud system to understand the user behaviour, we propose six criteria for the usage data as follows: (1) Real-time – the usage data should be extracted while the user is interacting with the application; (2) Up-to-date – the usage data should contain the recent data; (3) Complete – usage data that is extracted should have no missing data; (4) Correct – the data extraction component should only collect relevant data i.e., only that data should be extracted that could represent the application features that are critical for the user; (5) Available – the usage data should be available to extract by the usage data extraction component; (6) Reliable – the data should be obtained from a reliable source i.e., unbiased.

Now that we have identified the criteria, we aim to analyse the existing usage data extraction and data analysis techniques according to the criteria, data collection procedure and the user interface(s) considered to collect the usage data as shown in Table 2. Since the usage data is spread across front-end and back-end in a cloud environment and the front-end comprises of multiple interfaces for an end-user to access the SaaS application, we consider web-browser, mobile applications and command-line interfaces as the sources of usage data on the client-side. For this analysis, we started selecting the most recent papers on monitoring SaaS application usage. Since the selected papers refer to the older monitoring methods and techniques, we reached a saturation point and consideration of further older papers might not have yielded different results.

As a result of the analysis, we see that completeness of usage data is seldom considered criteria and command line interface has been neglected as a source of usage data in front-end. The majority of the usage data extraction techniques and methods consider web-browser or mobile applications individually but not together. Since the SaaS applications can be used by the end-user through all the three interfaces, they should be considered as the usage data sources to understand the critical features of the application from the end-users perspective.

4 USAGE DATA EXTRACTION FRAMEWORK

This section describes the proposed framework called Usage Data Extraction Framework. It comprises different phases as shown in Figure 4: Data Understanding, Data Classification, Data Sources Identification and Data Collection. The phases that are connected through the straight arrow are sequential, the dashed arrow represents the dependencies or outputs in each phase.

Data Understanding refers to understanding what we need to know from the data we intend to extract

Table 2: Usage Data Extraction Framework Analysis

Papers	Usage Data Criteria						Data Collection Procedure	User interface		
	Complete	Availability	Dynamic	Up-to-date	Reliable	Correct		Web browser	Mobile-App	CLI
(Pachidi et al., 2014)			X	X	X		Code Injection	X		
(Zhao et al., 2016)				X		X	Manual (provided by a company)		X	
(Cito et al., 2015)		X		X	X		Cloud Monitoring Tools	X		
(Junco, 2013)	X		X	X	X	X	3rd Party software	X	X	
(Sarkar et al., 2014)				X	X		Internal Logging System			
(Al-Bayati et al., 2016)		X			X		Manual			
(Xu et al., 2016)		X	X	X	X	X	Restful Interfaces			
(Smit et al., 2013)			X	X	X		Existing cloud monitoring tools			
(Yang et al., 2017)		X			X	X	Application Plugins		X	
(Ghezzi et al., 2014)		X	X		X	X	URL Logging, REST	X		
(Yang et al., 2015)				X		X	Manual (provided by a company)		X	

from the cloud system. The usage data analysis may be useful for the analysis of user behaviour, application performance, software personalisation, recommendation, software development and so on. It is important to understand and decide what do we make of the usage data before continuing to the further phases.

Data Classification refers to grouping the usage data as there exists various types of usage data and of various formats in a cloud environment. In this work, we classify the usage data into seven categories in a SaaS environment as shown in Table 1 in the data classification phase.

Data Sources Identification refers to the identification of the usage data sources. In this paper, we group the usage data sources into front-end and back-end, we emphasise on considering multiple user interfaces that end-users use to access the SaaS applications such as web browser, mobile applications and command-line interface as front-end usage data sources. Identifying these data sources is essential to understand the types and formats of the usage data, how the data is represented, stored and processed.

Data Collection refers to extraction of the usage data according to the classified types and identified formats. The usage data extracted should adhere to the proposed criteria: the data should be collected dynamically, that is, the usage data should the extracted while the user interacts with the SaaS application; complete - the usage data should be extracted from all the identified data sources; available the usage data should be available at identified data sources; up-to-date the usage data should have recent data from all the identified data sources; reliable the data sources and the data extraction techniques and mechanisms should be reliable in nature and could be trusted, that is, the data extraction techniques should not tamper or manipulate the usage data during the extraction process; correct the data extracted should be able to represent the purpose of the data as identified in the Data Understanding phase.

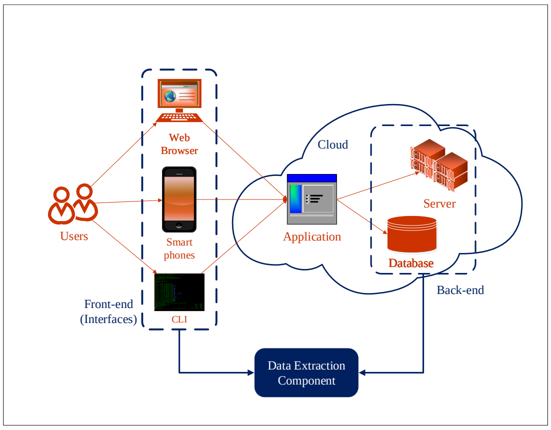

Figure 3: SaaS Usage Data Extraction

The framework we propose to extract usage data from a SaaS application considers web-browser, mobile application and command-line interface as usage data sources in front-end in addition to the back-end sources as shown in Figure 3. Hence, satisfying the completeness criteria; since the sources of the usage data are the front-end and back-end of the SaaS application, the usage data extracted are available and reliable; the usage data is analysed to understand the purpose for the extraction of the usage data in the data understanding phase and classified according to var-

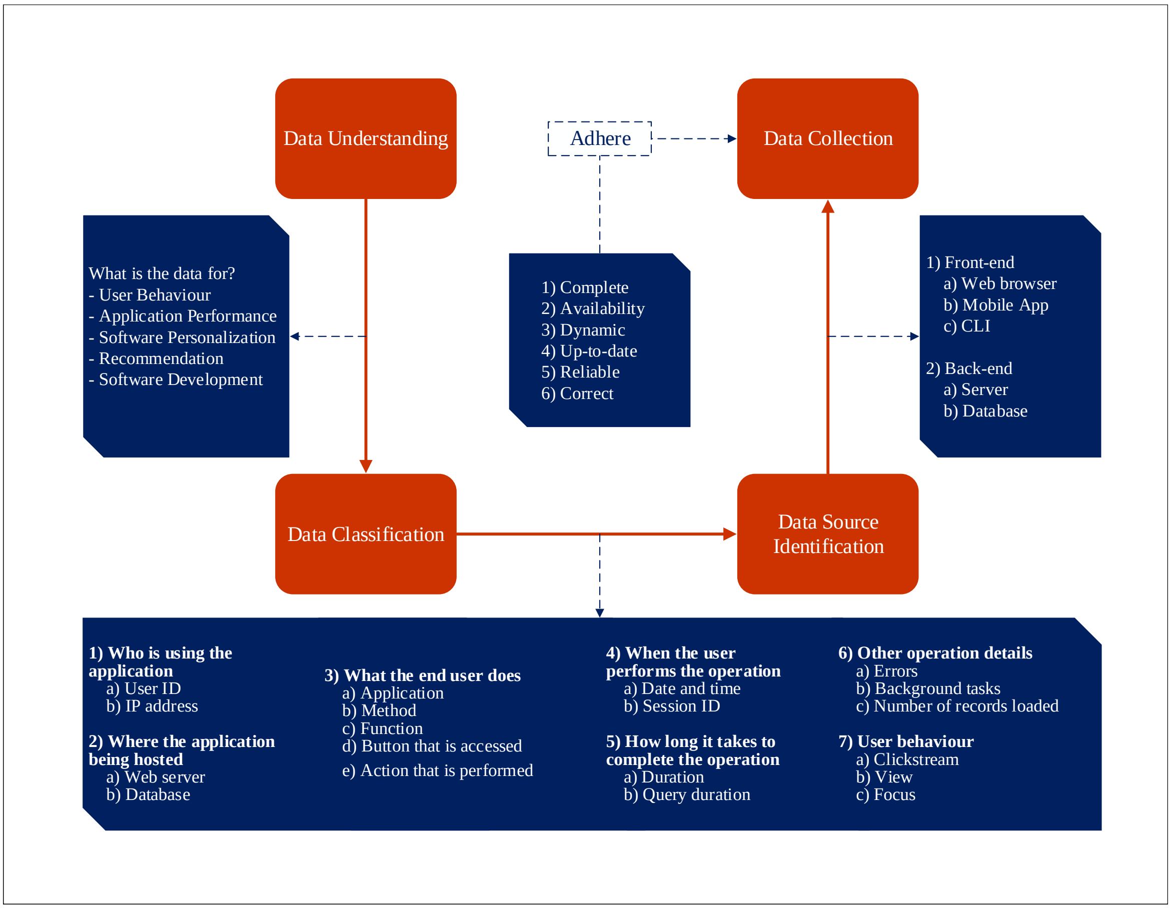

Figure 4: Usage Data Extraction framework

ious categories as shown in Table 1, the usage data satisfies the correctness criteria. The data extraction component will be designed and developed in such a way that it extracts the usage data from the sources while the user interacts with the SaaS application, satisfying dynamic and up-to-date criteria.

4.1 Implementation Plan

In this section we present a possible implementation of the framework. A SaaS application comprising a server and database at the back-end and web-browser, smart-phone and command-line interface as interfaces to access this application at the front-end will be developed. Different types of users (U1, U2, U3 and so on) are created with different responsibilities. Various features (F1, F2, F3 and so on) of this application will be identified and each type of user is given access to use these features through the interfaces. According to the proposed framework, the first phase is to understand the purpose of extraction of the usage data. For the purpose of this project, we consider *user behaviour, application performance, software personalization and software development*. The next phase is to classify the usage data into one of the seven types as shown in Table 1, the classification of the usage data depends on the purpose identified in previous phase. Once the intended usage data to extract are identified and classified, the next step is to identify the data sources. A SaaS application can be accessed by an end-user through interfaces such as *Web browser, Mobile App and Command-Line Interface* and the application is hosted on a *Server* with the storage provided by a *Database*. These 5 entities can be considered as the usage data sources. The final phase is *Data Collection* phase, the extraction techniques used should adhere to the usage data criteria as discussed in Section 4.

4.2 Implications

The framework has many implications for different stakeholders of a SaaS application. This framework can be used by a software architect to design the usage data extraction or monitoring component for a SaaS based application while designing the application, the architect will first understand the intention or purpose of the usage data that the extraction component should extract. This understanding of the usage data will help in classifying the usage data types and the formats followed by different components of the cloud system. Using this classification, the architect can identify where the usage data resides in the cloud system and then can decide what methods and techniques should be used for the extraction.

Data analyst can use this framework to better understand the nature of the usage data, the source of each data type, how the data is classified and extracted. Understanding the source of usage data and its classification could ease the analysis process. The software developer can use this usage data to understand the critical application features for an end-user, thereby prioritizing the features. This can improve the development time and cost for providing updates for the application.

The identified criteria for the usage data and the proposed usage data extraction framework can help researchers to consider and include all the interfaces used to access the cloud-based applications as usage data sources. This would lead to more replicable studies and results regarding development, instantiation and evaluation of the usage data extraction artefacts.

5 CONCLUSION AND FUTURE WORK

In this paper, we identified criteria for the usage data and analysed usage data extraction techniques according to the identified criteria, extraction procedure, and the considered user interfaces. We proposed usage data extraction framework with four phases: Data Understanding; Data classification – we provide an improved usage data classification; Data Sources Identification – we identified that it is essential to consider mobile applications and command line interfaces as usage data sources in addition to the web-browser and Data collection. As a result of the criteria analysis, the main contribution of the paper is a novel usage data extraction framework as shown in Section 4 which includes an improved usage data classification consisting of multiple interfaces as usage data sources on client-side for the purpose of usage data extraction, this framework includes all the usage data sources on the client-side such as web browser, mobile application and command-line interface. Hence, satisfying the complete criteria of a usage data extraction component.

Our future work aims (i) to consider further the commercial usage data extraction solutions for analysis of the identified criteria, (ii) evaluate the framework using case studies, (iii) further improve the framework to include the usage data storage and analysis procedure (iv) design and development of usage data extraction artefact adhering to the proposed criteria and framework.

ACKNOWLEDGEMENT

This work was supported with the financial support of the Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie)

REFERENCES

Al-Bayati, B., Clarke, N., and Dowland, P. (2016). Adaptive Behavioral Profiling for Identity Verification in Cloud Computing: A Model and Preliminary Analysis. *GSTF Journal on Computing*, 5(1):21–28.

Banerjee, A. and Ghosh, J. (2001). Clickstream clustering using weighted longest common subsequences. *Proc of the Workshop on Web Mining SIAM Conference on Data Mining*, page 3340.

Bucklin, R. E. and Sismeiro, C. (2009). Click Here for Interact Insight: Advances in Clickstream Data Analysis in Marketing. *Journal of Interactive Marketing (Mergent, Inc.)*, 23(1):35–48.

Cito, J., Leitner, P., Gall, H. C., Dadashi, A., Keller, A., and Roth, A. (2015). Runtime Metric Meets Developer: Building Better Cloud Applications Using Feedback. In *2015 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward!)*, pages 14–27, New York, New York, USA. ACM Press.

Fabijan, A., Olsson, H. H., and Bosch, J. (2015). Customer Feedback and Data Collection Techniques in Software R&D: A Literature Review. (210):139–153.

Gasparetti, F. (2016). Modeling user interests from web browsing activities. *Data Mining and Knowledge Discovery*, 31(2):1–46.

Ghezzi, C., Pezzè, M., Sama, M., and Tamburrelli, G. (2014). Mining behavior models from user-intensive web applications. *Proceedings of the 36th International Conference on Software Engineering - ICSE 2014*, pages 277–287.

Helfert, M., Donnellan, B., and Ostrowski, L. (2012). The case for design science utility and quality-Evaluation of design science artifact within the sustainable ICT capability maturity framework. *Systems, Signs & Actions*, 6(1):4666.

Junco, R. (2013). Comparing actual and self-reported measures of Facebook use. *Computers in Human Behavior*, 29(3):626–631.

Kesavulu, M., Bezbradica, M., and Helfert, M. (2017). Generic Refactoring Methodology for Cloud Migration - Position Paper. In *Proceedings of the 7th International Conference on Cloud Computing and Services Science*, pages 692–695, Porto, Portugal. SCITEPRESS - Science and Technology Publications.

Mell, P. and Grance, T. (2011). The NIST definition of cloud computing. *NIST Special Publication*, 145:7.

Pachidi, S., Spruit, M., and Van De Weerd, I. (2014). Understanding users' behavior with software operation data mining. *Computers in Human Behavior*, 30(January):583–594.

Pallis, G. (2010). Cloud computing: The new frontier of internet computing. *IEEE Internet Computing*, 14(5):70–73.

Petruch, K., Tamm, G., and Stantchev, V. (2012). Deriving In-Depth Knowledge from IT-Performance Data Simulations. *International Journal of Knowledge Society Research*, 3(2):13–29.

Sarkar, S., Ganesan, R., Cinque, M., Frattini, F., Russo, S., and Savignano, A. (2014). Mining invariants from SaaS application logs (practical experience report). *Proceedings - 2014 10th European Dependable Computing Conference, EDCC 2014*, pages 50–57.

Smit, M., Simmons, B., and Litoiu, M. (2013). Distributed, application-level monitoring for heterogeneous clouds using stream processing. *Future Generation Computer Systems*, 29(8):2103–2114.

Wang, G., Zhang, X., Tang, S., Zheng, H., and Zhao, B. Y. (2016). Unsupervised Clickstream Clustering for User Behavior Analysis. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*, pages 225–236.

Xu, P., Zhang, Y., and Shuang, K. (2016). Log on Cloud: A SaaS Data Collection, Storage, and Analysis Framework.

Yang, J., Qiao, Y., Zhang, X., He, H., Liu, F., and Cheng, G. (2015). Characterizing user behavior in mobile internet. *IEEE Transactions on Emerging Topics in Computing*, 3(1):95–106.

Yang, J., Wang, H., Lv, Z., Wei, W., Song, H., Erol-Kantarci, M., Kantarci, B., and He, S. (2017). Multimedia recommendation and transmission system based on cloud platform. *Future Generation Computer Systems*, 70:94–103.

Zaidman, A. (2010). Multi-Tenant SaaS Applications : Maintenance Dream or Nightmare ? Position paper. *Iwpse-Evol '10*, pages 88–92.

Zhao, S., Ramos, J., Tao, J., Jiang, Z., Li, S., Wu, Z., Pan, G., and Dey, A. K. (2016). Discovering different kinds of smartphone users through their application usage behaviors. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '16*, pages 498–509.