

Noushed Ahmed

RESEARCH ON DATA ANALYTICS AND USER TRACKING

RESEARCH ON DATA ANALYTICS AND USER TRACKING

Noushed Ahmed
Bachelor's Thesis
Spring 2018
Degree Programme in
Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Internet Services

Author: Noushed Ahmed

Title of the bachelor's thesis: Research on Data Analytics and User Tracking

Supervisor: Veijo Väisänen

Term and year of completion: Spring 2018

Number of pages: 68

In this technological era, the software business has been growing rapidly at an enormous rate. For the advanced software development, sales, marketing, customer support, the requirement of tracking data regarding the usage of the software has become more important than ever. It makes possible to get a higher revenue from software business.

The target of this bachelor's thesis was to research data analytics and usage analytics by tracking users and to provide a secured and seamless integration process of an analytical data tracking software into an application. A wide overview of choosing an appropriate third-party user's data tracking library, an analysis of the EU user privacy law in terms of data tracking and usage of generated tracking data were handled in this thesis.

This research and implementation were carried out in cooperation with Solibri Oy which is one of the leading software companies in Finland. The integration of analytics plugin was executed into their unique software, Solibri Model Checker.

This research and implementation report is beneficial for a software company as well as a software development team to establish a user tracking mechanism into a software and to gear up the business by making a secured bridge with their customers in terms of collecting data regarding the behaviour of their usage.

Keywords: Analytical plugin, Revulytics, Software Usage Tracking, EU User Privacy Law: GDPR, Tracking Data Analysis, Use Native Library into Java, Anonymous Tracking.

PREFACE

Solibri Oy is a ruling software company in the world in terms of BIM (Building Information Modelling) Quality assurance and Quality Control. Solibri Model Checker (SMC) and Solibri Model Viewer (SMV) are its bellwether products which work as a guard to ensure the quality of building and construction. It plays a revolutionary role by providing the construction and architectural companies, building owners a simple way to analyse clashing of components by means of severity, to visualize and investigate the building model. It finds out the possible faults in advance to justify the building information constructive models and provides a bunch of data regarding the research on models. This thesis task was completed in cooperation with this company.

I have been working for this company as a Junior Software Developer since May 2017. This company has provided me the opportunity to utilize my educational knowledge in practical world. One of my tasks was to implement user data analytics to SMC.

The entire project was completed during the year 2017-2018 under the supervision of Mr. Veijo Väisänen (bachelor's thesis supervisor), Ms. Kaija Posio (Language instructor), Pasi Paasiala (CTO of client company), Matti Kannala (Supervisor from client company) and Tuomo Paavilainen (Team leader from client company).

I would like to dedicate my bachelor thesis to my family, specially to my sister for supporting me in every situation. To Shejuti, for being my best friend and guiding me in hard times. To my friends, Sharmin, Srabonty and Sheekha from Oulu, Finland, for being beside me. Finally, I am really thankful for the entire Oulu University of Applied Sciences for lightening up my knowledge and Solibri family for believing in me.

Helsinki, 20.04.2018

Noushed Ahmed

CONTENTS

ABSTRACT	3
PREFACE	4
TABLE OF CONTENTS	5
VOCABULARY	7
1 INTRODUCTION	9
2 RESEARCH ON DATA ANALYTICS	11
2.1 Data Analytics	11
2.2 Usage Data Analytics	13
2.2.1 Comparative Research on Usage Analytics Service Providers	13
2.2.2 Selection of Appropriate Analytical plugin for SMC	15
3 RESEARCH ON USER TRACKING	16
3.1 User Tracking	16
3.2 EU User Privacy Laws	17
4 ANALYTICAL PLUGIN INTEGRATION	19
4.1 Tools and Methods	20
4.1.1 Eclipse	20
4.1.2 Xcode	20
4.1.3 SourceTree	20
4.1.4 JIRA	20
4.1.5 Bamboo CI	21
4.1.6 Bitbucket	21
4.1.7 SMC	22
4.1.8 Revulytics	23
4.1.9 Java	24
4.1.10 SWIG	24
4.1.11 Wireshark	25
4.1.12 FindBugs	26
4.1.13 PMD	26
4.1.14 Maven	26

4.1.15 Visual Studio	26
4.2 Design	27
4.3 First Integration Phase	28
4.3.1 Development Environment	28
4.3.2 Version Control	28
4.3.3 Build Automation	28
4.4 Second Integration Phase	28
4.5 Third Integration Phase	32
4.6 Fourth Integration Phase	40
4.7 Fifth Integration Phase	42
4.8 Sixth Integration Phase	44
4.9 Seventh Integration Phase	44
5 USAGE OF REVULYTICS	45
6 ANALYSIS OF DATA DRIVEN BY REVULYTICS	49
7 CONCLUSION	62

VOCABULARY

ABBREVIATIONS	MEANING
AES	Advanced Encryption Standard
API	Application Program Interface
BCF	BIM Collaboration Format
BIM	Building Information Modelling
CDA	Confirmatory Data Analysis
CI	Continuous Integration
CPU	Central Processing Unit
CSV	Comma-separated Values file
DA	Data Analytics
DLL	Dynamic-link Library
DWG	AutoCAD Drawing file extension
EDA	Exploratory Data Analysis
GDPR	General Data Protection Regulation
Git	Version Control System
GUI	Graphical User Interface
HTTPS	Hyper Text Transfer Protocol Secure
IDE	Integrated Development Environment
IFC	Industry Foundation Classes

ABBREVIATIONS	MEANING
iOS	IPhone Operating System
IP	Internet Protocol
JDK	Java Development Kit
JNI	Java Native Library
KPI	Key Performance Indicator
MAC OS	Macintosh Operating System
NPS	Net Promoter Score
OS	Operating System
pcap	Packet Capture
PDF	Portable Document Format
PHP	Hypertext Preprocessor
PMD	Programming Mistake Detector
RTF	Rich Text Format
RUI	Revulytics Usage Intelligence
Scrum	Agile Development Framework
SDK	Software Development Kit
SMC	Solibri Model Checker
SSH	Secure Shell
URI	Uniform Resource Identifier
URL	Universal Resource Locator

1 INTRODUCTION

Nowadays technology is taking over the ownership of the world. A life without the usage of technology is just a nightmare in the current situation. This statement can be proved by the increasing number of software start-up companies every day in the entire world with the aim of generating the business market for high-tech and innovative products which will help the users to simplify their day to day life. Start-up software companies are such companies which have basically zero or minimal operating history with a high hope to generate a great revenue gradually (1, p. 27). From the starting point of Software development in such a start-up company, every step in the software business should be taken carefully to meet the specific tech-based need of the customers. After the first successful release of a software product, it becomes quite important to make a clear bridge with the customers to get an overall insight about the usage of the software.

Although executing a user survey is an effective way in terms of gathering data regarding the user's satisfaction and overall review of the software, still this type of survey is not enough to understand the usability of the product. Thus, a company needs to be in an unknown zone regarding the collection of actual data. This data would enable to understand e.g. the business point, the business market and the flow of software development. This is the reason why the term Data Analytics (DA) was introduced in a software company.

An analysis of data driven by the usage of a specific software makes a clear bridge between a software company and the need of the software for what the users are ready to pay for. By tracking the user's way of using the software, a software company could be able to generate a goal for further development as well as the business process. Not only for the start-up companies, but also for other software companies it is becoming a mandatory part to get the insight regarding the usage of their software.

Software Piracy is one of the worst scenarios in software business. By analysing it seems that six of every seven software users are heading for the pirated copy

of the software (2, p. 29). Software piracy creates a great loss in software business. In this case, it becomes important to keep eyes on software piracy which could be possible by a usage analysis. With a data analytics a software company can also handle the licencing process. It is also possible to see the geographical information of the users and the Internet Protocol (IP) address which helps to identify the users utilizing a pirated copy of the software.

For improving the software development process, it is a basic need to understand the performance of the released features and the ratio of the functionality usage. By researching the performance related data, a software development team could identify the way to optimize the already released as well as the future features to be released. The team could emphasize the functionality which is being used mostly by the users. Again, it is also necessary to find out the worst-case scenario while running a single feature by the users. For all these cases, data analytics can play a vital role.

However, data analytics is related to track user data. Therefore, the user privacy law also needs to be taken into account before implementing data analytics into a software. As the privacy law is gradually being updated time to time, the integration should be in a modular way so that it does not consume much time to rebuild the whole mechanism of gathering user data to comply with the latest law.

The main theme of this bachelor's thesis is to provide a research on data analytics in a software company, to classify different options of third party libraries for integrating data analytics into a software, to analyse the benefits of an analytics plugin into a software, to integrate analytics plugin into a modular, in a secured and seamless way in a software, to analyse the EU user privacy law in terms of tracking data and to figure out a proper way of data analysis driven by tracking for the improvement of all different departments in a software company such as development, sales, marketing and support departments.

2 RESEARCH ON DATA ANALYTICS

Before integrating a data analytics mechanism into a software, it is quite important to learn the theme of data analytics, its use cases, its application and a selection of proper analytics software provider according to the business size. With a proper knowledge of these above-mentioned points, it would be much easier for a software company to start familiarizing itself with the data analytics world. In this section of the thesis, it is described how to provide an overall idea of data analytics, why it is necessary and how it would be beneficial for a software company. It is also analysed how to choose an appropriate data analytics service provider from a bunch of different existing options.

2.1 Data Analytics

Data is the main ingredient of the world digitalization. The term “Data” refers to a qualitative or quantitative value of any object or sets of objects. It can also be described as a series of symbols which are the root of generating information. The data can be classified by structured and unstructured data, where the structured data is holding the directive information of any object and the unstructured data is the primary element for further processing with the aim of collecting information.

Data Analytics (DA) is a mechanism of providing a shape to the unstructured data to discover useful information, to identify a significant pattern of any hypothesis, and to draw a suggestive conclusion. Without data analytics, data is nothing but a useless object. Data can be collected by different internal and external sources. After collecting the raw data, a preparation process is needed to create an input for the data analytics process. With the proper input of cleaned data, the data analytics can be performed to generate valuable information on top of it. There are different types of data analytics such as an exploratory data analysis (EDA) based on a summarization of data set characteristics - emphasising more of the usage of a visual display (3, p. 5), a Confirmatory data analysis (CDA) which quantifies the data using the statistical method of significance or confidence to calculate the probability of occurrence (4, p. 757), a Qualitative and Quantitative

data analysis which basically generates a number and pattern-based analysis (5, p. 13; 6, p. 3), and a predictive data analysis which helps to generate a hypothesis for future (7, p. 1). In a nutshell, data analytics is a combination of procedures such as refining input data, searching patterns among the data, generating a relational connection among the data, identifying a difference between several objects, comparing objects in terms of relations and finally generating the predictable decision. In a life cycle of data, DA is a very important part in terms of generating meaningful and usable information.

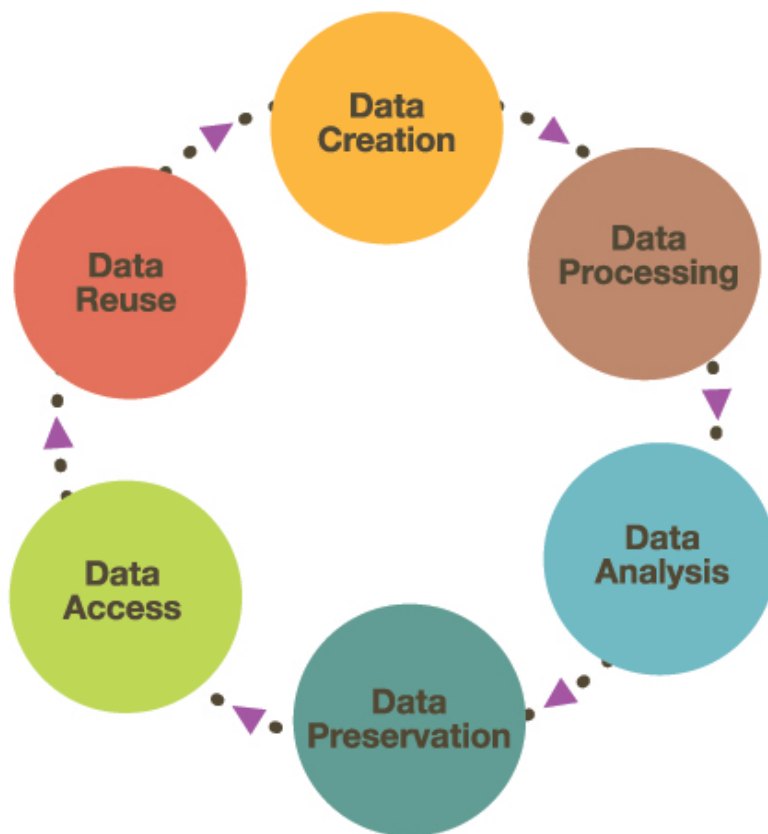


FIGURE 1. Life cycle of research data (8)

Figure 1 provides an overall idea of how a data turns into information after going through a cyclic process of processing, analysis, preservation, accessing and re-using of data. Information can also be used as the data for a further analysis until drawing a suitable conclusion. In each section of this life cycle, different methods

and algorithm are used. However, a better result always relies on a systematic process of data analysis. The method of data analysis should be chosen by the formation of data as mentioned earlier in this chapter.

2.2 Usage Data Analytics

Usage Data Analytics is a predominant type of Data analytics which helps to generate statistical information regarding the usage. With a goal of clear understanding the behaviour of usage, usage data analytics plays a vital role in the current world. It provides a proof of conceptual prediction regarding the software as well as a product usage. To verify the product usability, customer satisfaction and product management, every company should analyse the data driven by usage of the product by customers. It also generates an overall insight regarding the effectiveness of a product, business opportunities with a product and visualize the value proposition of a product. Usage data analytics saves time in product management, ensures a better software development process and also helps to create a strong base in the competitive business world.

2.2.1 Comparative Research on Usage Analytics Service Providers

Currently there exists a huge number of Usage analytics services. In this section a comparative research is described in order to choose an appropriate usage analytics service.

“Cluvio” is an embedded cloud analytics tool which helps small and medium sized software companies to analyse the data using the SQL and R Programming language (9). It also visualizes the data analysis with an attractive graphical user interface to evaluate the usage data in terms of making a business decision and having a sensible knowledge. It also provides an advanced user management and collaboration between the clients. It also follows the European data protection rule by protecting the data inside the EU region. The supported language of this software is English. The pricing module of this software starts from 99€ per month with a single user management and around 2, 000€ per month for a large company with an unlimited user management. However, it also provides a 14 days of free trial usage to check the usability of this software. (10.)

“Coveo” is another cloud-based usage analytics service provider. It analyses the search metrics to provide the business company an insight of what the users are looking for. It has its own mechanism to figure out the most relevant data from the usage analytics to understand the user behaviour in terms of usage (11). It provides the business intelligence solution to companies. Coveo is also enriched with a graphical user interface for a collaborating purpose within a company. To use Coveo’s usage analytics framework, a company needs to count at least 2,000\$ per month (12).

“Pendo” is a product usage data capture software which allows its client to analyze customer usage, feature engagement and in-app messaging module. A company could analyze the data to understand the usage flow, to make changes of their software according to the customer feedback or survey and to define a user segment based on the usage behavior. It has also a mechanism to calculate a Net promoter score (NPS) in order to measure the dissatisfaction of customers. This software is free to use until 100 of active customers per month. For more than 100 customer bases, it is said to contact with the customer service team of Pendo. (13.)

“Keen IO” is a web-based analytics and data visualization platform. It provides an overall insight to a software company regarding the usage analytics, business growth analytics, error found by using the software and real time performance analytics. It has also a graphical interface to manipulate the analysis by the users. The supported language is English. It has monthly subscription-based pricing module which depends upon the number of events that are needed to be analyzed. For around 15,000,000 events, it costs 1,000\$ per month. (14.)

“Analytics Plus” is an analytics tool to generate a data analysis report using a graphical interface. It has built in statistical and mathematical formulas which could be used to analyze data. However, it is also possible to generate custom formulas in terms of analysis. It also provides a reports collaboration mechanism. It supports 10 different languages, such as English, German, French, Spanish (15). To get the professional edition for 2 users, a company needs to count 2,395\$ per year (16).

“Revulytics” is a software usage analytics tool which enables to visualize the real time software usage analytics and direct messaging to the customers in terms of campaigning and survey. It tracks the usage behavior of the users and generate statistical reports based on that. It also has a mechanism of managing the Licensing of a software and churn analysis of product. A company could also gather customize data regarding the software functionalities and performance. To protect the software from piracy, it is a useful tool. It has also a subscription-based pricing module and the price might vary based on the number of events to be tracked, the number of custom events tracking or different features such as a direct message campaigning. The pricing starts from 59\$ per month up to 1,500 installations and 100 basic events tracking (17). For an advanced and ultimate plan, it is advised to contact with Revulytics. (18.)

2.2.2 Selection of Appropriate Analytical plugin for SMC

The main requirement of the supervising company of this bachelor’s thesis was to get a wide insight of how the users are using its software. It was searching for an option to measure the product metrics, duration of feature performing and real time exception collection. This company was also trying to get an overall idea regarding the mostly used and “never used” features so that they could optimize the performance of those specific features for customer satisfaction.

After comparing all the above-mentioned analytics tools, “Revulytics” was chosen to track the users in terms of the usage analytics, as it meets all the requirements of this software company and its product, SMC.

3 RESEARCH ON USER TRACKING

3.1 User Tracking

User tracking is a granular study to collect the usage behaviour data. For a software related business company, there is hardly any way to understand the viewpoint of software usage other than holding a survey or customer society events. Although a survey could present the feelings of the customers regarding the software, it does not give much information as the usage vary from different customers segments. Often a company needs to predict the future development of its software with vague information generated by the survey which does not come out successfully in terms of the better business management. To keep pace with the current growing number of software business, it is becoming more important to analyse the usage more efficiently to generate a high revenue and make a strong position in a competitive situation. User tracking opens a clear path in terms of understanding the software market and future product management. It gives real time data to analyse why a customer is using the software, what are the specific features that are used by the customers, how long they are using a specific feature, what are the system architectures of the users while using the software, which makes a user to avoid the software, and what types of error users are facing while using a particular software. Piracy is also a burning issue for the entire software industry. To protect a software from piracy, user tracking plays a vital role.

However, a software industry could benefit a lot by tracking their users in terms of the data collection. To generate a meaningful KPI (Key Performance Indicator), user tracking could be the first choice. To analyse the business market and to prioritize the sales management, user tracking could help by providing the data for a research. Usability check is a key point of a software which generates the growing number of customers. By analysing the data driven by user tracking, a software company could take necessary steps to improve the usability of the software. To analyse the user experience on a software and to monitor the adoption of the new features, there is hardly any other better way than user tracking. User

tracking not only helps in product management, but it also helps all different sections in a company. A software developer can analyse the data to identify performance related issues, a possible bug analysis and an optimization of performance.

After releasing a product for a business market, user tracking clears the darkness between the software company and its users. By analysing the tracked data, a software company could analyse the needs of a customer. This can be beneficial in terms of setting up the pricing module and expanding the business widely.

3.2 EU User Privacy Laws

To legally proceed with tracking users, a company must follow the user privacy regulation. The purpose of this section is to provide up-to-date information regarding the EU user data protection regulations while starting to track a user for a business purpose. A short research will be described pointing to the facts that are quite necessary to understand for a software company before tracking users.

The GDPR (General Data Protection Regulation) legislation is currently a hot topic in regards of user privacy. The main purpose of this legislation is to unify and strengthen data collection within the entire EU zone (19). This legislation will start ruling from 25th May 2018 by replacing the existing data protection directive (officially Directive 95/46/EC) (20).

The following described regulations should be kept in mind before tracking users,

- GDPR will not only be implied to the inner European companies who are dealing with the user's personal data but also on the outsider companies that will handle the European citizens personal data (21, p. 5).
- As a punishment of not abiding this law, an enterprise has to face 4% of its yearly worldwide turnover and up to 100 million euros in all cases (22, p. 287).
- The business companies could no longer be able to provide the terms and conditions within a big scope. The consent should be logical, intelligible and easily readable enough that it will make sense for the users to understand the whole points (23).

- The consent between the user and data collector companies should be working in both ways. It means that there must be a possibility for the users to withdraw his or her consent whenever they want after providing the consent. While tracking the user, the data collection company should ask for a consent from the customer. They should also provide a way so that the user can uncheck his or her consent and it should automatically stop collecting the user's personal data (See Figure 2 below). Silent consent procedure is not possible anymore. (19.)
- The user must be informed early enough regarding the usage of their personal data collection (19).
- If the collected users' personal data is required to pass to other third-party company for whom the user does not provide consent, the user must also be informed, and a permission needs to be collected before transferring the data (19).
- The terms and conditions, including the permission asking procedure should be concise and transparent enough so that the user can access those whenever they want. In terms of a non-understandable consent, the validity of the consent would be denied. (24, p. 5.)

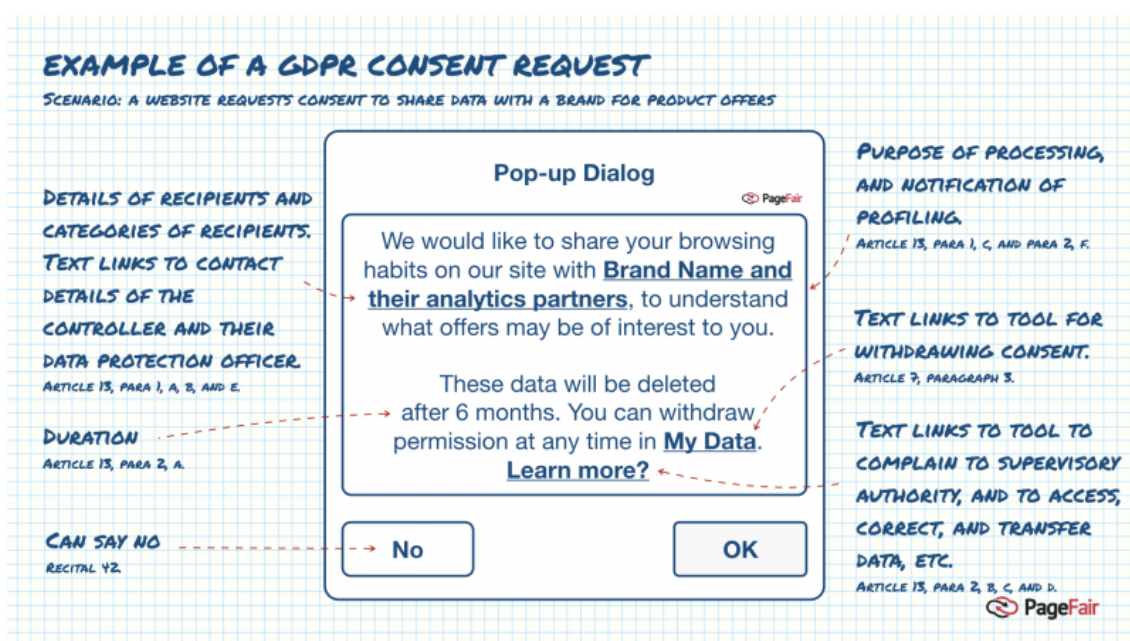


FIGURE 2. An example of GDPR Consent request (25)

4 ANALYTICAL PLUGIN INTEGRATION

In this section of bachelor's thesis, the entire integration development process will be described elaborately. As Revulytics was chosen as the usage analytics service provider by Solibri Oy, the whole integration was done into their software, Solibri Model Checker. The integration phases have been divided into several phases to provide the development flow from the scratch. By following the phases, it is possible to integrate the Revulytics usage tracking mechanism into any software. However, the initial environment setup may vary from company to company upon the selection of different software for maintaining the code base, such as the repository maintaining software, the artefact maintenance and usage of development environment. In this case, these phases will help the reader to be on the right track while integration.

In a nutshell, the integration phase will be described starting by providing the information on different tools and methods used for the integration, setting up the development environment, creating an extension project, generating the Java library from the Third party native C++ Library of Revulytics for Windows and MAC OS, describing the connection procedure from the application to the Revulytics server, developing different tracking methods to gather the user usage data, developing test classes to check the connection of this external plugin project, creating a plugin package of Analytics in the SMC code base, explaining of Maven dependency, Creating an instance of Revulytics and forming of different tracking methods using the Revulytics interface, Error handling in different cases and examples of different types of tracking methods usage in the software codebase.

The example of codes might need to be edited to provide own information. Before implementing, it needs to be checked if the license and other useful data regarding the Revulytics connection have been gathered, so that the usage data could be visible in the Revulytics dashboard. Also, it needs to be checked the license policy in order to use different tools in the business sector.

4.1 Tools and Methods

The usage of different tools and methods is described shortly in the following,

4.1.1 Eclipse

Eclipse is the most widely used Java Integrated Development Environment (IDE) which helps a developer to develop any Java software (26). The coding of entire integration of analytical plugin was mostly executed in the Eclipse platform.

4.1.2 Xcode

Xcode is an IDE for MacOS. It is available through Mac App Store and it is useful to develop applications for e.g. MacOS, iOS (27). Xcode was used to generate the “dylib” Java library from the C++ library of Revulytics. A file with an extension “.dylib” in Mac OS stands for a dynamic library which is used by any application to run necessary methods during the runtime of the application in a need basis (28). Whereas in Windows OS, a file with an extension “.dll” stands for the same purpose as “.dylib”. Xcode has aggregated tools to generate dylib libraries.

4.1.3 SourceTree

SourceTree, a Graphical user interface (GUI) application to perform Git (Version Control System) was used for entire integration.

4.1.4 JIRA

JIRA is a project management software which allows to track issues, bug and roles. It was used in this project work for Scrum maintenance, Issues related with Integration, setting of priorities of the issues and in collaboration with team mates.

4.1.5 Bamboo CI

Bamboo CI (See Figure 3 below) is a server provided by Atlassian company which helps to integrate and deploy the workflow continuously. It helps to create automated builds, and releases by providing the appropriate configuration. In the analytics plugin integration, it was used to make releases of different versions of the Revulytics project so that it could be used in the main code base of the application.

Continuous Integration

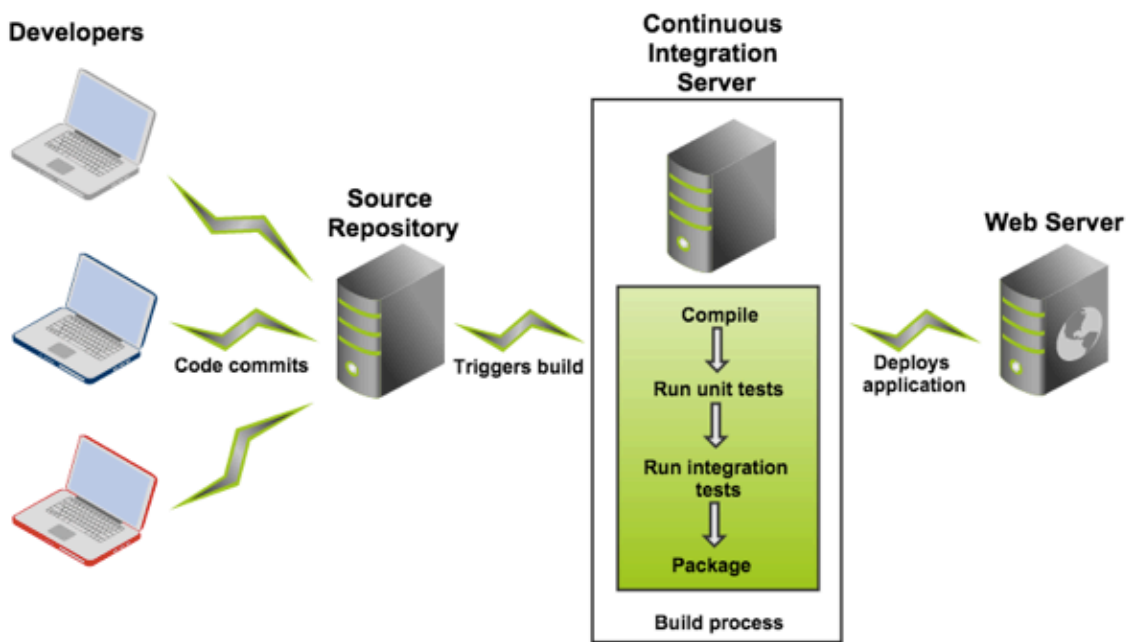


FIGURE 3. Continuous Integration by Bamboo CI (29)

4.1.6 Bitbucket

Bitbucket is another major product of Atlassian which helps to review code, check merge, make a pull request, keep large file storage of Git, code search and Smart mirroring which helps to clone large repositories within a short time while working on a project globally (30). It was mainly used for the code repository storage, code reviewing and merging content into the feature branch.

4.1.7 SMC

Solibri Model Checker (SMC) is the software from the client company, Solibri Oy, of this thesis. The entire Analytical plugin integration was executed on top of this software. This software works as a guard to visualize problems and issues regarding the architectural model before and during the construction. Information regarding the building life cycle could also be generated for a further analysis with the help of SMC. Models can be imported from ArchiCAD and it has also support to import IFC and DWG formats. Different calculations regarding the model such as area, length, component counts and filtering, component classifications and relational view can be performed through SMC. SMC also provides the mechanism of analyse Models, building accessibility by running different rules. With the powerful 3D Engine, it is also possible to walk through the architectural model even before the construction starts. This 3D view can also be used to highlight the searched components from the model (See Figure 4 below) and analyse them according to the need. However, SMC also provides an easy communication mechanism of presentation. Presentation regarding any issue raised while analysing the model with SMC can be exported to different formats such as BCF (BIM Collaboration Format), PDF, RTF and Excel. It also gives the possibility to import presentations. However, a user could also visualize the structural hierarchical tree of any model and maintain the large number of models into SMC according to their own wish. (31.)

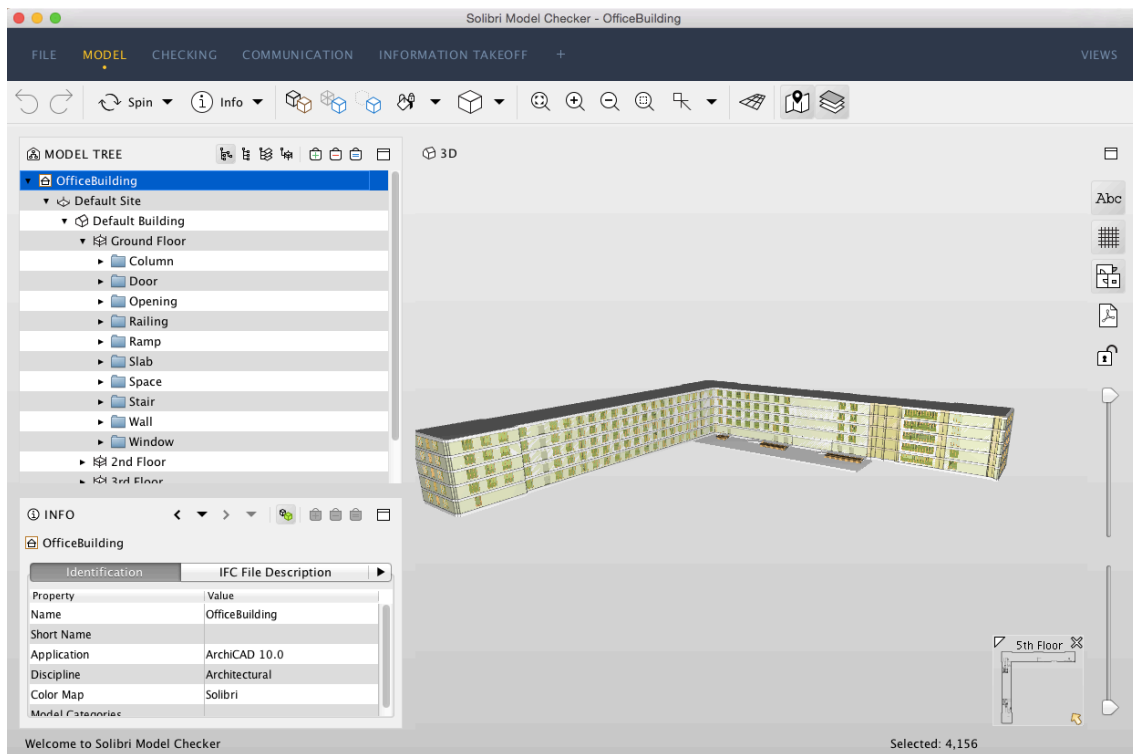


FIGURE 4. A View of SMC Application

The Solibri Model checker application can be found from the following link,

<https://www.solibri.com/products/solibri-model-checker/>

4.1.8 Revulytics

Revulytics is a software usage analytics which helps to track the user's usage of a software and provides analytical data to perform data driven software development. It has also support for license compliance. Revulytics was chosen as an analytical plugin to integrate into the SMC application. This software justifies the collected data and visualizes it into a digital user interface (See Figure 5 below) where it is possible to analyse the data further by applying filters. It has a mechanism to protect the software from piracy and provides a way of real time user survey. It also collects the user data, such as geographical information and system configurations where the software is being used. Revulytics helps software companies to make a decision on top of usage analytics of the software. After releasing a software, a software company loses the visibility regarding the happenings of the software. In this case Revulytics helps to track the installation and

the behaviour of software usage which generates a reliable overview on product management. (32.)

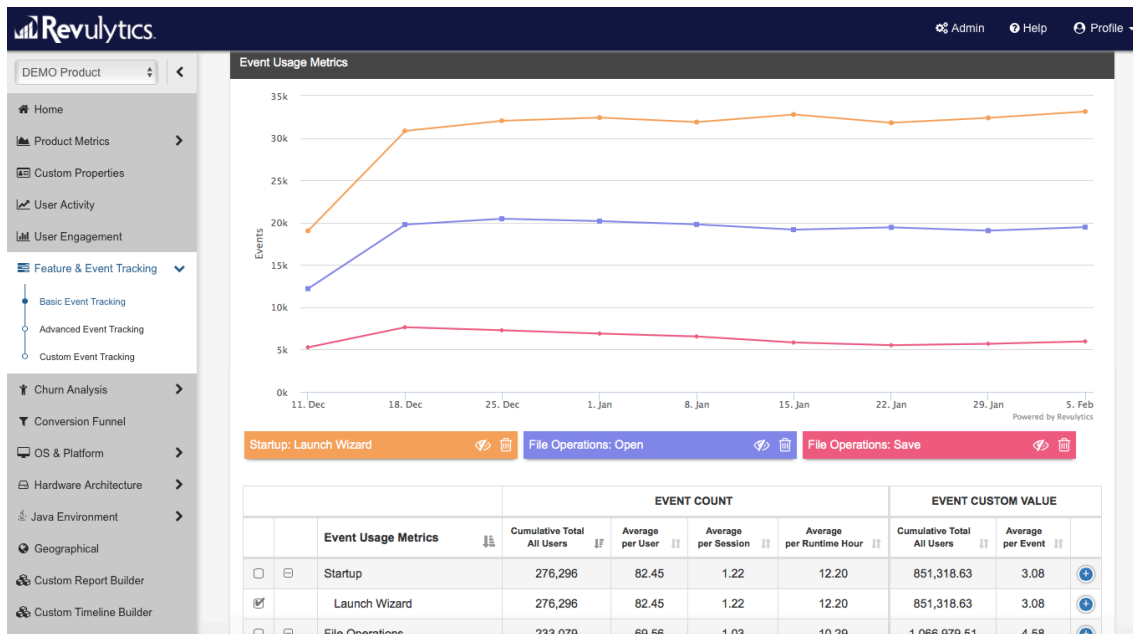


FIGURE 5. A View of Revulytics Dashboard

This service license can be bought from their website,

<https://www.revulytics.com/buy-software-usage-analytics>

4.1.9 Java

Java is a class-based, concurrent and object-oriented programming language (33, p. 1). It was used for the entire integration development of the Analytics plugin.

4.1.10 SWIG

Simplified Wrapper and Interface Generator (SWIG) is a tool for generating native C, C++ libraries into a different targeted library such as Java, C#, JavaScript and other scripting languages (34). It has the mechanism to convert native language-based functions and methods to another language. SWIG creates source code to make a bridge between a native and a target programming language. It is a free open source software tool. (35.)

4.1.11 Wireshark

Wireshark is a free and open source network protocol analyser. It works similarly as tcpdump. But it has a GUI (See Figure 6 below) which makes it user friendly. With Wireshark it is possible to analyse different types of network traffic. It uses pcap (packet capture) to gather the traffic data (36). Through its interface, it is possible to analyse the data in live and from pcap. By using Wireshark, identifying the running application, monitoring the virtual machine traffic and troubleshooting a network problem become much easier. It could also generate the report after the successful analysis. (37.)

In this thesis project, Wireshark - version 2.4.4 was used to analyse the packet sending to the Revulytics server from SMC. Wireshark helped to find out any security leakage between the connections while tracking and the behaviour of Revulytics connection while proxy is in use from SMC, was also verified. To justify the EU user privacy rule, this tool was quite helpful to ensure the tracking traffic.

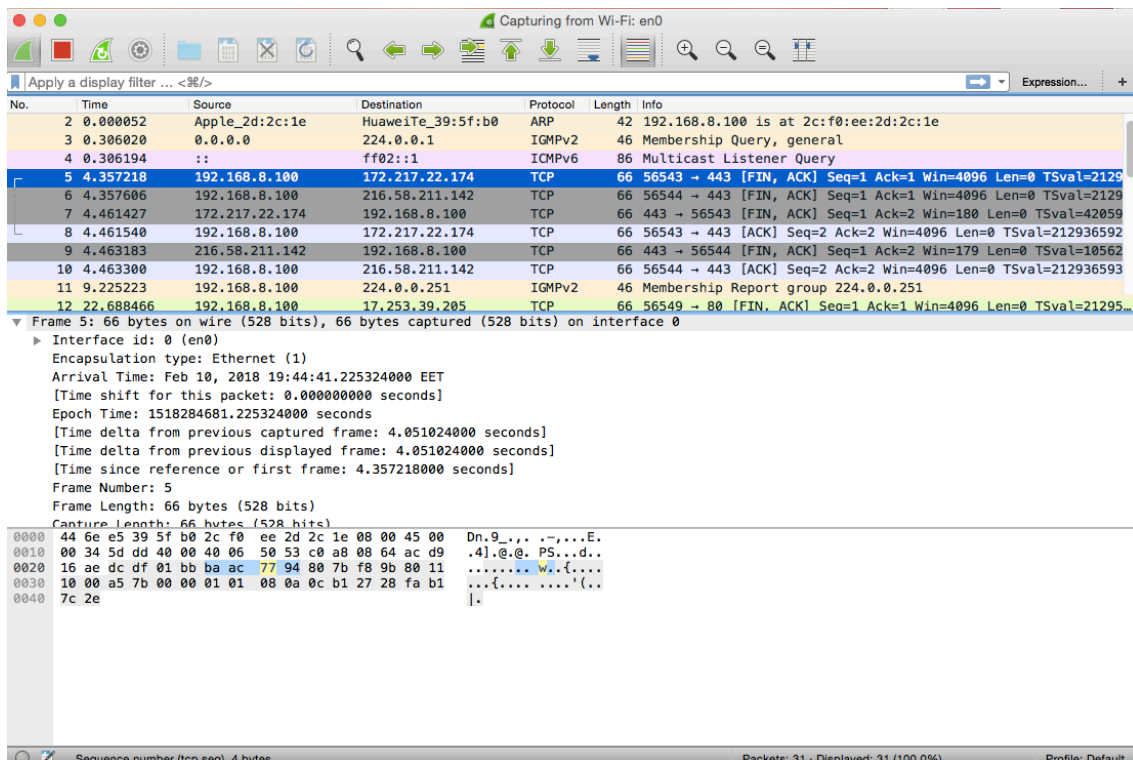


FIGURE 6. A View of capturing WI-FI network using Wireshark

4.1.12 FindBugs

FindBugs is a static code analysing tool to analyse the Java code and to find out the possible bugs from the code. The FindBugs plugin was used in Eclipse for this whole project during the integration of Revulytics.

4.1.13 PMD

Programming Mistake Detector (PMD) is a Source code analyser which keeps tracking every time new code is being saved into a project to find out the possible bugs and inefficient code or habit of Bad Programming (38). The PMD plugin was used in this thesis project. It helped to find out the duplicate code, dead code and possible bugs.

4.1.14 Maven

Maven is a tool for managing and building a Java project. It performs the handling of all the inter related dependencies and process the local builds with all the third-party libraries. It finalizes the deployable JAR, WAR or EAR files (39). The Maven plugin was used in the project to handle the dependencies between Revulytics and SMC. The project was also structured using the Maven plugin in Eclipse.

4.1.15 Visual Studio

Visual studio is IDE from Microsoft. In this project, Visual studio was used to generate a Dynamic link library using C++ and a header class so that the native library could be used in a Java application.

4.2 Design

Figure 7 below shows the entire design of Analytics plugin integration.

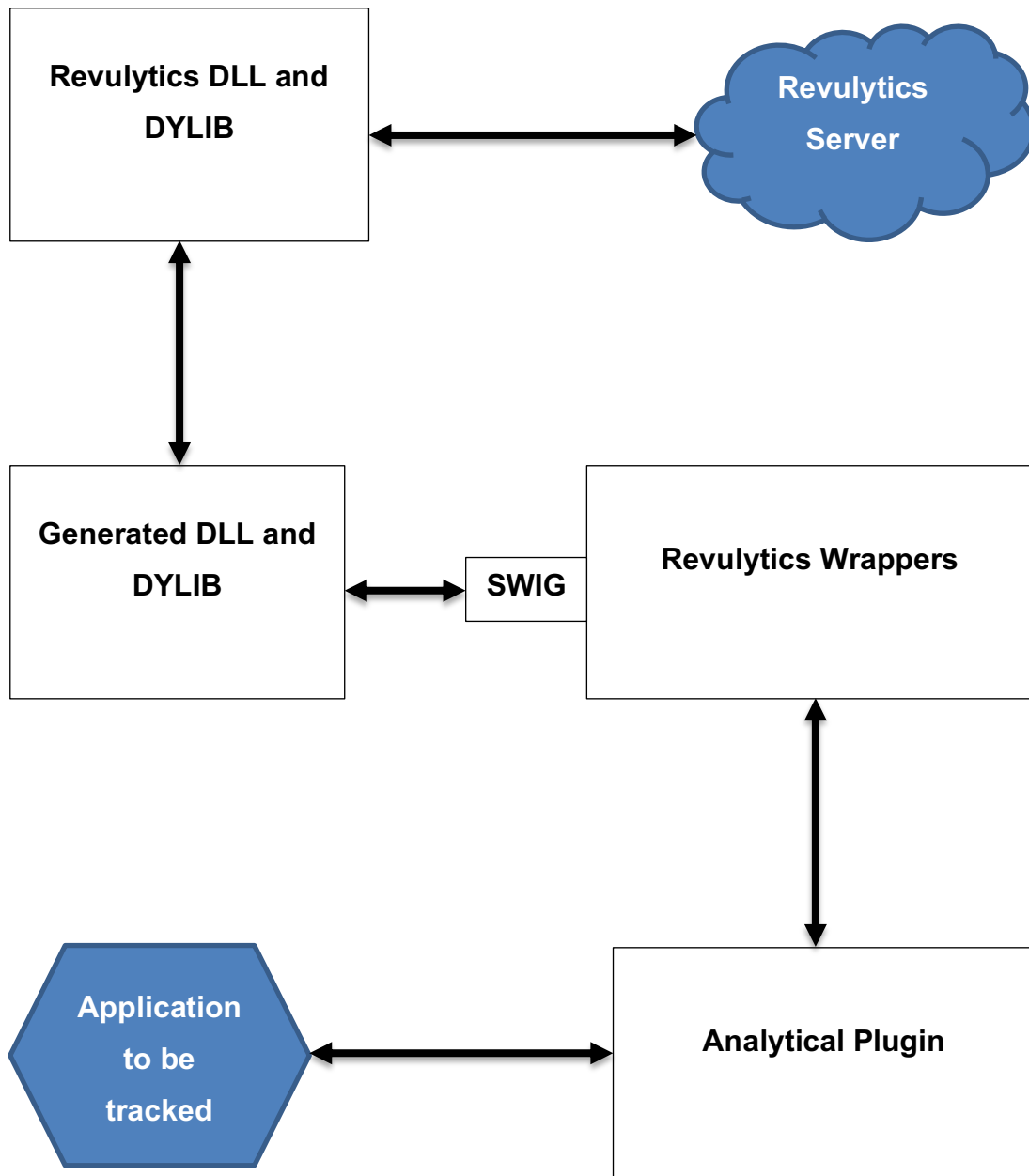


FIGURE 7. Analytics plugin integration design

4.3 First Integration Phase

The Integration phase was started by creating a new project from the scratch for Revulytics. The following steps were followed:

4.3.1 Development Environment

The Programming environment was set up by installing Eclipse including Maven, FindBugs and PMD plugin into it. After the successful creation of the Maven project, the setup procedure was performed in Bitbucket and the newly created project was committed into Bitbucket.

4.3.2 Version Control

The BitBucket Git repository provides URL in the SSH and HTTPS format. This URL was used to commit the new project from Eclipse and also to clone the repository in SourceTree so that further commits, a local code review, source code management and a pull request could be handled more sophisticatedly.

4.3.3 Build Automation

As soon as the first push was done to a Bitbucket server, A configuration plan was generated using Bamboo CI so that it could automatically build the whole project after each merge of the codes to the master branch. And several releases were held using Bamboo CI after each successful version of the Revulytics project.

4.4 Second Integration Phase

During the second phase of the integration, Java wrapper classes were created from the Revulytics native C++ SDK. Using SWIG, JNI (Java Native Interface) Library was generated for Windows and MAC OS. An overall procedure will be described below.

- First of all, the relevant SDK was downloaded from the Revulytics Development site (Date of retrieval 11.02.2018),
<https://devzone.revulytics.com>

- For this project, C++ SDK, the version 5.1.0, for Windows and MAC OS were used because during 2017 Revulytics did not have support for SDK written in Java. On 23rd of January 2018 they provided SDK in the Java language.
- The Revulytics SDK folder for MAC OS contains a dylib library for MAC OS and it will be used to generate a dynamic library on top of Java wrapper classes. This generated library will work as a middle man to connect wrapper classes with the Revulytics SDK dynamic library.
- Revulytics SDK also contains ruiSDKC, ruiSDKDefines and ruiSDKOBJC headers containing C function declarations. These files were used to generate Java wrapper classes. First of all, a pre-processed file with the “revulytics.i” extension was created. The header files were included into this pre-processed file as well as a “cpointer.i” module to generate pointers to primitive datatypes. A pointer function was also included to manipulate pointer types. The codes of “revulytics.i” file looks like below,

```

/* File: revulytics.i */
%module revulytics
%{
#include "ruiSDKC.h"
#include "ruiSDKDefines.h"
%}

#include cpointer.i

/* Create some functions for working with "int *" */
%pointer_functions(int, intp);

/* Let's just grab the original header file here */
#include "ruiSDKDefines.h"
#include "ruiSDKC.h"
%ignore RUIReachOutHandler;

```

- The “RuiReachOutHandler” function stands for an extra feature which the client’s license does not support. For this reason, it was ignored in this file. The reach out handler is used to deliver a message or a survey directly to the customer’s desktop. For Windows and MAC OS, the creation of this “revulytics.i” file was done similarly in their containment folder.
- After creating the pointer file, SWIG was used to generate the Java and the C++ wrapper. From the MAC OS terminal, SWIG can be installed by the command “brew install swig”. In Windows OS, SWIG was downloaded from the following link, (Date of retrieval 11.02.2018), <http://www.swig.org/download.html>
- In MAC OS, Terminal was used to point to the directory where the Revulytics SDK files including the “revulytics.i” file had existed. To generate the C++ and Java classes, a “swig -c++ -java -package revulytics.wrapper revulytics.i” command was used from the terminal. Generated “revulytics_wrap.cxx” will be used to generate a dylib linker Library. In Windows OS, a “PATH-OF-YOUR-DOWNLOADED-SWIG-FOLDER\swig’ -c++ -java -package revulytics.wrapper revulytics.i” command will generate the same Java wrappers and C++ file.
- A new Library project was created in Xcode to generate a dylib linker Library for MAC OS. librui_5.1.0.x64.dylib, revulytics_wrap.cxx, ruiSDKC.h, ruiSDKDefines.h, ruiSDKOBJC.h – these files were added to the project which will lead to generate the library
- In the Build Settings section, Header search paths from the system JDK path were added into the “Search Paths” section. Those are,

```
/Library/Java/JavaVirtualMachines/jdk1.7.0_80.jdk/Contents/Home/include/darwin
```

```
/Library/Java/JavaVirtualMachines/jdk1.7.0_80.jdk/Contents/Home/include
```

- Into the Library Search Path, the main dylib library of Revulytics was added as “/usr/local/lib”. For MAC OS, this dylib library was needed to

shift to the system library path so that Revulytics could make a connection from the User's system to the Revulytics server.

- From the Build Phases panel, the Revulytics dylib library was added in the "Link Binary with Libraries" section, and the status was provided as "Optional". The Build Configuration was set to "Release" from the Product -> Scheme -> "Edit Scheme..." option so that we could generate the linker Library as a release product.
- By building and running the project, the linker library was created. It can be found by pressing File -> "Project Settings...". There is a default path for the derived data already mentioned which is similar as "/YOUR-XCODE-PATH/DerivedData". The generated library (librevulytics.dylib) was collected and pasted to the Revulytics SDK package for further use in Development. For Windows OS, the following link will be helpful to generate the dll library (Date of retrieval 11.02.2018), <https://www.linkedin.com/pulse/step-java-native-programming-roshan-gerard-bolonna>

After the successful generation of linker library both in Windows and MAC OS, the preparation of the extension project (The project- Revulytics, which was created in the 1st integration phase and added to Bitbucket) was performed and it will be described in the third integration phase.

4.5 Third Integration Phase

In the Revulytics module, a sub project called “revulytics-wrapper” was created. In this sub project, all the generated java wrapper classes (in the 2nd integration phase using SWIG) were moved. Other than the generated classes, “AnalyticsConnection.java” (interface), “RevulyticsConnection.java” (Connection related class which implements the interface), “RevulyticsConnectionType.java” (Enumeration class for different connection protocols), “RevulyticsException.java” (exception handling class), “RevulyticsParams.java” (All the parameters and their getter and setter related class), “Utils.java” (Utility class) were created. The manipulations of these classes will be described gradually.

The C++ API Documentation for MAC OS can be found in the following link, (Date of retrieval 14.02.2018),

<https://devzone.revulytics.com/docs/v5/Cpp/mac/>

And the C++ API Documentation for Windows OS can be found in the following link, (Date of retrieval 14.02.2018)

<https://devzone.revulytics.com/docs/v5/Cpp/windows/>

Using these two documentations, the Revulytics configuration was set up in the Java environment. It is advised to read through the documentation beforehand or while integrating any specific task, as the later integration phases would only emphasize on the important things related with the integration. This integration phase was started by manipulating “RevulyticsConnection.java” file and all the other classes were manipulated as per needed. It is advised not to edit the generated classes as long as there is any specific reason or better understanding of what is needed to be done.

The following steps could be followed to prepare the Revulytics project in a modular way,

It was described earlier that the “RevulyticsConnection.java” class implements the “AnalyticsConnection.java” interface. First of all, a checker for the Operating

system was integrated into the “Utils.java” class so that the specific library can be loaded before connecting to the Revulytics.

To show the exception messages in different occurrences and in a modular way while preparing Revulytics connections, The Java Exception class has been extended by the “RevulyticsException.java” class.

While providing data for the Revulytics connections and to the tracking method, int values are needed to be converted into different pointers such as 32-bit, unassigned. SWIG has generated java classes to manipulate the pointer into primitive data types as “cpointer.i” and pointer function were included while generating the Java classes. In the “Utils.java” class, those conversions were performed.

To configure the Revulytics Connection, it is needed to specify the communication protocol. Revulytics supports 2 different types of protocols. One is with AES-128 encryption mode and another one is without encryption. With encryption, there are two different modes. Those are

- RUI_PROTOCOL_HTTP_PLUS_ENCRYPTION
- RUI_PROTOCOL_HTTPS_WITH_FALLBACK”

With these two modes, a 128-bit (32 hex characters) hex-encoded string (aesKeyHex) must be supplied. (40.)

The non-encrypted protocol mode should be supplied with an empty aesKeyHex (40). That is,

- RUI_PROTOCOL_HTTPS

For further usage of these 3 types of protocols, an enumeration class “RevulyticsConnectionType.java” was created.

In the “RevulyticsConnection.java”, a new method of getting the protocol pointer was introduced to provide the exact value after converting into the int 32-bit pointer.

While setting up the configuration such as connection, proxy setup, product data setting, starting of SDK, starting of new session to track or disconnecting the communication, the result got from the Revulytics server was collected and according to that, the connection is interrupted if the a server sends a bad request. Different response code for various requests can be found in the “ruiSDK-Defines.h” file provided into the Revulytics package. Two different methods for checking the result were introduced. One of them is to check if the server responses with the allowed value mentioned in the API. If not, the SDK will be stopped in terms of bad connections.

```
private SWIGTYPE_p_RUIInstance ruiInstance;

private void checkResult(SWIGTYPE_p_int32_t result, String errorMsg,
boolean stopSDK, Integer allowedValue) throws RevulyticsException
{
    int intValue = Utils.int32pValue(result);
    if (intValue != revulyticsConstants.RUI_OK &&
        (allowedValue == null ||
         intValue != allowedValue.intValue()))
    {
        if (stopSDK) {
            revulytics.ruiStopSDK(ruiInstance,
                Utils.newInt32p(
                    revulyticsConstants.RUI_SDK_STOP_SYNC_INDEFINITE_WAIT));
        }
        revulytics.ruiDestroyInstance(ruiInstance);
        throw new RevulyticsException(errorMsg + ": "
            + Utils.int32pValue(result));
    }
}

private void checkResult(SWIGTYPE_p_int32_t result, String errorMsg,
boolean stopSDK) throws RevulyticsException
{
    checkResult(result, errorMsg, stopSDK, null);
}
```

For configuring a connection, different parameters were supplied through the Revulytics methods. From the API documentation, the information regarding those parameters could be gathered. Some parameters such as a product id and the Revulytics Server URL, were collected from Revulytics when the license was bought. The “RevulyticsParams.java” class will handle those parameters to get and set the values for a further use.

Revulytics is built to support the communication of data in a plug-in type environment. For this reason, during each connection from a client to the server, an instance is needed to be initiated and this should also be destroyed after the session of tracing has been accomplished. (40.)

For the Revulytics configuration, it is required to initialize the Revulytics Usage Intelligence (RUI) after the instance has been initiated. This should be done before calling any other function. This configuration method requires the following parameters (40),

- Pointer to the RUI instance which has been initiated.
- Configuration file path from the user’s system which will be used by the Revulytics to SDK instance configuration. This file should exist in the system and it should also have the write right access by the Revulytics
- 10 Digit Product ID which has been collected from Revulytics
- A maximum 8 characters containing non-empty space application name of the “to be tracked” product.
- Server URL which has been collected from the Revulytics.
- Protocol which has been described while creating the “RevulyticsConnectionType.java” class.
- 32-bit hexadecimal characters AES key.
- A “multiSessionEnabled” Boolean true or false which will indicate the enabling of multi session by the users or automatically by Revulytics.
- A “reachOutOnAutoSync” Boolean true or false which will indicate ReachOut™ should be requested through auto sync of SDK or not. The ReachOut™ handler is an intelligence to send a direct message to the user’s platform through the product.

An example of an overridden connect method into “RevulyticsConnection.java” is below,

```
protected boolean connected = false;

@Override
public void connect(RevulyticsParams params) throws RevulyticsException
{
    if (!connected) {
        SWIGTYPE_p_int32_t result = revulytics.ruiCreateConfig(
            ruiInstance,
            params.getConfigFilePath(),
            params.getProductID(),
            params.getAppName(),
            params.getServerURL(),
            getProtocolPointer(params.getProtocol()),
            params.getAesKeyHex(),
            params.isMultiSessionEnabled(),
            params.isReachOutOnAutoSync());
        if (Utils.int32pValue(result) != revulyticsConstants.RUI_OK) {
            throw new RevulyticsException(
                "Revulytics configuration failed");
        }
    }
}
```

To communicate with the Revulytics server through the HTTPS proxy, the “ruiSetProxy” function was used after the configuration. This function was utilized by providing the host, port, username and password credentials (if an authenticated proxy). The connection was also tested by the “ruiTestConnection” function from Revulytics as following,

Before starting the SDK connection, the data regarding the “To be tracked” product must be provided to the server with the “ruiSetProductData” function. It takes the following parameters (41),

- RUI Instance
- A maximum 128 characters of Product edition information
- A maximum 128 characters of Product language information
- A maximum 128 characters of Product version information
- A maximum 128 characters of Product build number information

While it is required to clear old product information, a “null” value could be provided to clear the old information from the client System. RUI SDK was started followed by setting the product data.

While creating the connection, it was noticed that the starting of SDK requires a bit time before creating the session. If an appropriate time was not provided to the server to start the SDK, the server was sending the bad request to the client. In this regard, a method was created to make the thread in sleep mode for 100 milliseconds.

The explicit session was generated by calling the “ruiStartSession” function. In this case, a specific session id is required by the Revulytics server which has to be followed by certain rules as below (42),

- A 10-64 characters Session id
- No empty space in this id except space character

Within the session, the license data regarding the user’s licensing relation with the “To be tracked” product was also sent to the Revulytics server so that it could keep track on the trial, purchased or newly purchased license users. The “set-LicenseData” function takes the following parameters (43),

- RUI Instance
- License key type, one of the key from the list mentioned in SDK.
- Pointer to indicate if the key is expired
- Pointer to indicate if the key is activated
- Pointer to indicate if the key is blacklisted
- Pointer to indicate if the key is whitelisted
- Session id which has been provided while starting the session.

In this project, the user's license type has been tracked as "Evaluation" which is similar as a trial user, "Purchased" which is similar as a purchased license user and if other than those two options, the license has been tracked as "Unknown".

Starting SDK, Session and Instance should come with a pair with Disconnecting SDK, Session and Instance after the user has been tracked.

To send different types of data to the Revulytics server in case of tracking according to the SDK, different tracking methods were introduced in the "AnalyticsConnection.java" interface. These tracking methods were implemented in the "RevulyticsConnection.java" class to use the tracking functions provided by the Revulytics SDK. An example of a tracking method is below,

```
@Override
public void trackEvent(String eventName, String category)
throws RevulyticsException {
    if (connected) {
        SWIGTYPE_p_int32_t result = revulytics.ruiTrackEvent(
            ruiInstance,
            category,
            eventName,
            params.getSessionID());
        if (Utils.int32pValue(result) != revulyticsConstants.RUI_OK) {
            throw new RevulyticsException("Revulytics event tracking failed:");
        }
    } else {
        throw new RevulyticsException("Tracking failed. Not connected.");
    }
}
```

In the method above the "ruiTrackEvent" function was used from RUI SDK, with the following parameters,

- RUI Instance.
- Event category name. It is optional. Empty String can also be provided.
- Event name.

- Session ID, the same session id is needed to be used within which the track event is called (44).

Revulytics has a tracking mechanism of runtime exceptions. This option could provide the specific method name, exception message as well as stack trace. (45.)

To set up this feature, the following example could be useful,

```
@Override
public void trackEvent(String className, String methodName,
                      String exceptionMessage, String stackTrace)
                      throws RevulyticsException {
    if (connected) {
        RUIExceptionEvent exceptionEvent = new RUIExceptionEvent();
        exceptionEvent.setClassName(className);
        exceptionEvent.setMethodName(methodName);
        exceptionEvent.setExceptionMessage(exceptionMessage);
        exceptionEvent.setStackTrace(stackTrace);
        SWIGTYPE_p_int32_t result = revulytics.ruiTrackException(
            ruiInstance,
            exceptionEvent,
            params.getSessionID());

        if (Utils.int32pValue(result) != revulyticsConstants.RUI_OK) {
            throw new RevulyticsException(
                "Revulytics exception tracking failed.");
        }
    } else {
        throw new RevulyticsException(
            "Exception Tracking failed. Not connected.");
    }
}
```

In this integration phase, all the necessary coding in the “Revulytics” side has been completed. Now using the Bamboo CI, an artifact release of this project has

been created so that an external project could be used in the main product code-base.

4.6 Fourth Integration Phase

In this integration phase, a plugin “AnalyticalPlugin.java” class was added to the product code base from where Revulytics had been called to track events.

First of all, a Maven dependency was added in the “pom.xml” file of the package where the Revulytics will be used through the “AnalyticalPlugin.java” class. This class extends the Default plugin super class from the main code base which handles firing different events, starting and destroying the instance of the plugin. To set all the required parameters in regards of connecting to the Revulytics server, a new method “getCurrentParameters()” was added which will return “RevulyticsParams” from the Revulytics package. For generating random session id, the “SecuredRandom()” Java module was used. In the main code base, the connection was set up while starting the application. To start up the Analytics plugin, the following method was introduced,

```
public void start() {
    if ("CHECK-IF-USER-GIVES-PERMISSION-TO-TRACK"){
        // if user gives permission
        try {
            RevulyticsParams params = getCurrentParameters();
            if ("CHECK-IF-SOCKS-PROXY") {
                // Only if not a SOCKS proxy
                connection = new RevulyticsConnection(params);
            }
        } catch (Throwable e) {
            LOG.error("Analytics plugin not started: "
                + e.getMessage(), e);
        }
    }
}
```


As per GDPR, it is required to ask permission from the users if they want to be tracked. For this reason, only if they provide permission, the connection will be made to the Revulytics server. However, the Revulytics server does not support data communication through the SOCKS proxy protocol. The only supportive protocol is HTTPS. In this case, it is also verified if the user is using the app with the proxy, and if the proxy is the HTTPS type, then the connection will be initiated.

All the tracking methods which were introduced in the Revulytics external project, were also created in this “AnalyticsPlugin.java” class to make a bridge between the main code base and the external project. An example of an exception tracking method is below,

```
public void track(String className, String methodName, Throwable er){
    if (connection != null) {
        try {
            String stackTrace = getStackTrace(er);
            String exceptionMessage = (er.getMessage() != null) ?
                er.getMessage() : er.getClass().getSimpleName();
            connection.trackEvent(className, methodName,
                exceptionMessage, stackTrace);
        } catch (Throwable e) {
            LOG.error("Tracking failed: " + e.getMessage(), e);
            disconnect();
        }
    }
}

private String getStackTrace(Throwable er) {
    ByteArrayOutputStream byteStream = new ByteArrayOutputStream();
    PrintStream printStream = new PrintStream(byteStream, false);
    er.printStackTrace(printStream);
    String stackTrace = byteStream.toString();
    return stackTrace;
}
```

4.7 Fifth Integration Phase

As per GDPR, user tracking must be made in a way that the tracking driven data will be anonymous. There should not be any way to know exactly which user's data particular data is. In this case, a user should be informed through the application that the application would like to track the usage anonymously if the user gives permission. It is also needed to notify the user why the application provider is interested in tracking the usage, for example for improving the application. The user must also be provided a way to start and stop tracking whenever they want, for example a checkbox which would lead to track the user in terms of selection. An example of such an environment was integrated as below,

```
private static final String VALUE_NO = "no";
private static final String VALUE_YES = "yes";
private static final String VALUE_ASK = "ask";
private final Preferences prefs = Preferences.userNodeForPackage(
    AnalyticsPlugin.class);
private BooleanSetting allowTrackingSetting = null;
private boolean askOptIn() {
    String userTrackingKey = getUserTrackingKey();
    String trackingValue = prefs.get(userTrackingKey, VALUE_ASK);
    if (VALUE_ASK.equals(trackingValue)) {
        String prompt = getString("AnalyticsPlugin.ALLOW_TRACKING");
        ResourceBundle res = getResources();
        Icon icon = getIcon(res.getString("AnalyticsPlugin.SurveyIcon"));
        Object[] options = {
            res.getString("AskOptInAction.YES"),
            res.getString("AskOptInAction.NO"),
            res.getString("AskOptInAction.LATER") };
        Object initialValue = options[0]; // Yes is the default
        JLabel label = new JLabel(prompt);
        label.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
        int chosen = JOptionPane.showOptionDialog(ApplicationFrame, label,
            res.getString("AnalyticsPlugin.
                ALLOW_TRACKING_TITLE"),
            JOptionPane.YES_NO_OPTION,
            JOptionPane.QUESTION_MESSAGE,
```

```

        icon, options, initialValue);
    switch (chosen) {
    case 0: { // YES
        trackingValue = VALUE_YES;
        break;
    }
    case 1: { // NO
        trackingValue = VALUE_NO;
        break;
    }
    case 2: { // LATER
        trackingValue = VALUE_ASK;
        break;
    }
    }
    prefs.put(userTrackingKey, trackingValue);
}

boolean isTracking = VALUE_YES.equals(trackingValue);
if (allowTrackingSetting != null) {
    allowTrackingSetting.setValue(isTracking);
}
return isTracking;
}

private String getUserHash() {
    String userData = "GET-ANY-DATA";
    if (userData != null) {
        byte[] hash = "GENERATE-SHA-HASH-WITH-USER-DATA";
        return Base64.encodeBase64String(hash);
    }
    return "unknown";
}

private String getUserTrackingKey() {
    return "tracking-" + getUserHash();
}
}

```

The noticeable part of the above code snippet is that the tracking id has been generated by any of the unique user data and SHA-Hash and also the string is base-64 encoded. In this way, the tracking id has been anonymized. The default value for asking the permission is “Yes”. The user could also choose “No” or “Ask later” options. In terms of “Ask later”, the same pop up message for asking the permission could be shown to the user within a fixed amount of time.

4.8 Sixth Integration Phase

However, by calling these different tracking methods from this Analytics plugin, various cases could be tracked such as how long a particular feature is taking to perform, how many times a feature is being used. With an id value pair as a parameter, it is possible to collect particular data while performing any specific task. If all the actions from an application have their single super abstract action, in this case by applying the tracking method into that abstract action would be a much cleaner way to track. In this case, the repetition of applying the tracking method for each of the action could be avoided. With an appropriate category name, it would be easier to visualize all the similar events under a same hood.

4.9 Seventh Integration Phase

JUnit tests should be created to check the connection and disconnection mechanism. It is also a good practise to check the crucial methods through the JUnit test such as all the conversion related methods, checking the proxy configuration and the behaviour of the Revulytics connection in terms of different proxy and user’s permission.

5 USAGE OF REVULYTICS

In this chapter, the usage of the Revulytics site will be described so that a Revulytics client could handle the necessary functionalities regarding the usage tracking.

As a licensed client, it is needed to log into the Revulytics dashboard from the following link, (Date of retrieval 17.02.2018)

<https://analytics.revulytics.com/accounts/login/>

After successfully logging into the Revulytics dashboard, a welcome page will appear to provide the information regarding the product id (which has been used while connecting to the Revulytics server), the subscription plan and also the SDK documentations and Demo for the developers and the client. On the right top corner of the page, there is an option called “Admin”. By clicking it, it will bring the user to the admin site where the licensing, tracking events organization, user’s permissions, property management, Billing and new product registration are existed. In the “Product Settings” sections, there exists an option, “License Key Registry” (See Figure 8 below). On this page, the tracking data of the license can be found to see the number of trial or purchased users, the expiration date and the number of installs of the software. It is also possible to get the data regarding the license type such as, if a user’s license has activated, blacklisted or white-listed. There is also an option to filter the license related data (See Figure 9 below). By using this filter, it is possible to analyse the required data regarding the License.

License Key	License Type	Activated	Blacklisted	Expired	Whitelisted	Added Date	Install Count	Install Quota	Over Quota Count
2c5bc269-ac2c-465c-8f07-25fa4930ce13	Purchased	✗	✗	✗	✗	24 Jul 2014 08:18	1	2	-1
Expiry Date 17 May 2017									
38f19c54-e130-46a6-9537-57e68b9dc39b	Purchased	✗	✗	✗	✗	24 Jul 2014 08:18	1	2	-1
Expiry Date 17 Mar 2017									
6c9b094b-52a2-4f4e-a76d-9c82fe25a5fc	Purchased	✗	✗	✗	✗	24 Jul 2014 08:18	2	2	0
Expiry Date 05 Apr 2017									
9dddde50-06aa-40e6-afe2-712387b4761	Purchased	✗	✗	✗	✗	24 Jul 2014 08:18	1	2	-1
Expiry Date 17 May 2017									
41daf750-ebf6-4883-bc99-83e413f02b3a	Freeware	✓	✗	✗	✓	24 Jul 2014 08:18	2	2	0

FIGURE 8. An example of license key registry of demo product.

License Key	License Type	Activated	Blacklisted	Expired	Whitelisted	Added Date	Install Count	Over Quota Count	
211d01fd-afe2-4490-a160-91b6a6f6f2b4	Freeware	✓	✗	✗	✓	24 Jul 2014 08:18	2	0	
Expiry Date 19 Nov 2017									

FIGURE 9. An example of filtering license key registry.

In the Admin section, there is also another important section under the Product Settings, “Tracked Events Whitelist Management” (See Figure 10 below). This section is used to select the events to be tracked and visualized in the dashboard.

There is an option to automatically populate the table with the newly events which has been used by the user, and if it is still not being added in the list. In this table, it is possible to select the events to bring those for a further analysis into the dashboard. There are two options to be chosen. A Selection as a basic event and a selection of an Advanced event. The basic event will be tracked e.g. a counter of usage, duration of per usage and cumulative total users. This event does not support any filtering or deeper analysis. However, the advanced events provide a way to filter the data and analysis per tracking event data and comparing between different data cycles and it also provides the information regarding the features which have been used mostly and which have not been used ever. Exporting the list of events into CSV and an Excel format is also possible through this page. The number of the selection of Basic and Advanced events depends upon the License agreement with Revulytics.

The screenshot displays the 'Tracked Events Whitelist Management' interface for 'DEMO Product (2376158762)'. It includes a sidebar with navigation options, a top navigation bar with 'Admin', 'Help', and 'Profile' links, and a main content area. The main area features a 'Tracked Events Whitelist Management' title, a notification box, an 'Auto-Whitelist' section with a 'Disabled' toggle, and a table of events. The table has columns for 'Category', 'Event Name', 'Date First Seen', 'Basic', and 'Advanced'. Below the table are pagination controls and a search bar.

Category	Event Name	Date First Seen	Basic	Advanced
File Operations	Clone	01 Feb 2011 00:00 (UTC)	✓	✗
File Operations	Copy	01 Feb 2011 00:00 (UTC)	✓	✗
File Operations	Open	01 Feb 2011 00:00 (UTC)	✓	✓
File Operations	Save	01 Feb 2011 00:00 (UTC)	✓	✓
Install Wizard	Step 1	01 Feb 2011 00:00 (UTC)	✓	✗
Install Wizard	Step 2	01 Feb 2011 00:00 (UTC)	✓	✗
Install Wizard	Step 3	01 Feb 2011 00:00 (UTC)	✓	✗

FIGURE 10. An example of Tracked events whitelist management.

However, it is advisable not to unselect the event from the whitelist, as all the past-data collected with that event name will be removed as soon as an event has been unselected from this list.

While integrating the analytical plugin, a configuration path has been provided. This path has been used by the Revulytics server to identify the user. If different users are using a software in the same environment without any manipulation of configuration file path from the software provider, the users will be acknowledged as a single user. (46.)

The default firewall setup should not create any problem while connecting to the Revulytics server. However, if the user has a strict firewall setup, in this case the connection between the Revulytics server and the user's system would be aborted if the firewall has a mechanism to filter unknown HTTPS access. (47.)

Revulytics Usage Intelligence has a mechanism to cache the tracking data into the user's system as a "ruilog.log" file. In terms of poor network or if for some reason the connection between the application and the Revulytics server could not be established for a number of days, all tracking data will be regained by the server when the next successful sync will occur. (48.)

RUI has been designed to be light weighted. While initializing the SDK from the application, it will immediately sync for the first time. After this first sync, the synchronization between the client and the server will be performed after every 20 minutes. The data will be sent to the server as compressed and immediately after the successful sync, the data will be visible in the Revulytics dashboard. The synchronization will also be performed while the stopping of the SDK will be performed from the client application. (49.)

In case of clearing History data, it is advisable to contact the support team by creating a ticket from the webpage. For example, while integrating the Revulytics into the application, there might be a lot of duplicate events if the data for the same event has been sent through different event name or the category name and the event name have been changed often while developing. It is possible to remove that data either by removing it from the Tracking events whitelist management section or by submitting a ticket to the customer support and providing the list which is also needed to be removed from the event whitelist to keep the list clean and up-to-date.

6 ANALYSIS OF DATA DRIVEN BY REVULYTICS

In this chapter, an overall analysis will be described to provide a general idea how the data driven by Revulytics could be gathered for an investigation in terms of product management and how the data could be beneficial for different sectors of a software company.

After logging into the Revulytics site, a dashboard becomes visible from the left side of the page. For this analysis, the data from the demo product was used to maintain the confidential policy of the supervising company.

From the Revulytics dashboard, the first analysis of tracked data could be found from the “Product Metrics” section. This section provides a graphical analysis regarding the product performance summary such as how many new installations of the software have been occurred, how many have been lost and how many currently active installations there are. It also provides a statistical analysis of the distribution of product versions, editions, build numbers and languages (See Figure 11 below). A Graphical pie chart has also been used in this section to provide the information regarding the license status, such as the number of activated, whitelisted, trial or blacklisted users. It also analyses the number of different license types activated on different date of a time frame. This section has a filter mechanism which can be used to generate appropriate data for any proof of concept. The following benefits could be gathered by these data,

- The number of maximum used versions of a software can be helpful to provide more emphasis on the development of that particular version. A comparison of the number could also be helpful to understand which features make the difference between the old and new or any particular version of the software usage more lucrative to the users.
- This metrics is also helpful for the sales or marketing team to recommend the user by showing the number of usage. In this case the new customer will be attracted to use the software by seeing the number of customers.
- The language distribution analysis (See Figure 12 below) will help the product management to figure out which language version of the software

needs to be improved, the localization of it as it has less customers than others.

- A license distribution analysis (See Figure 13 below) will help the management team as well as the sales department to analyse the number of trial users who have been shifted to purchase the license. The flow of purchased license will also provide an idea of yearly sales prediction and using this prediction it is possible to generate a business plan.
- To protect the software from piracy, this tracked data will be helpful to take necessary steps.
- By analysing the number of users in different language version, it would be possible to make a decision to increase the customer support members for that specific language.



FIGURE 11. An example of statistical analysis of tracked data regarding product version and edition

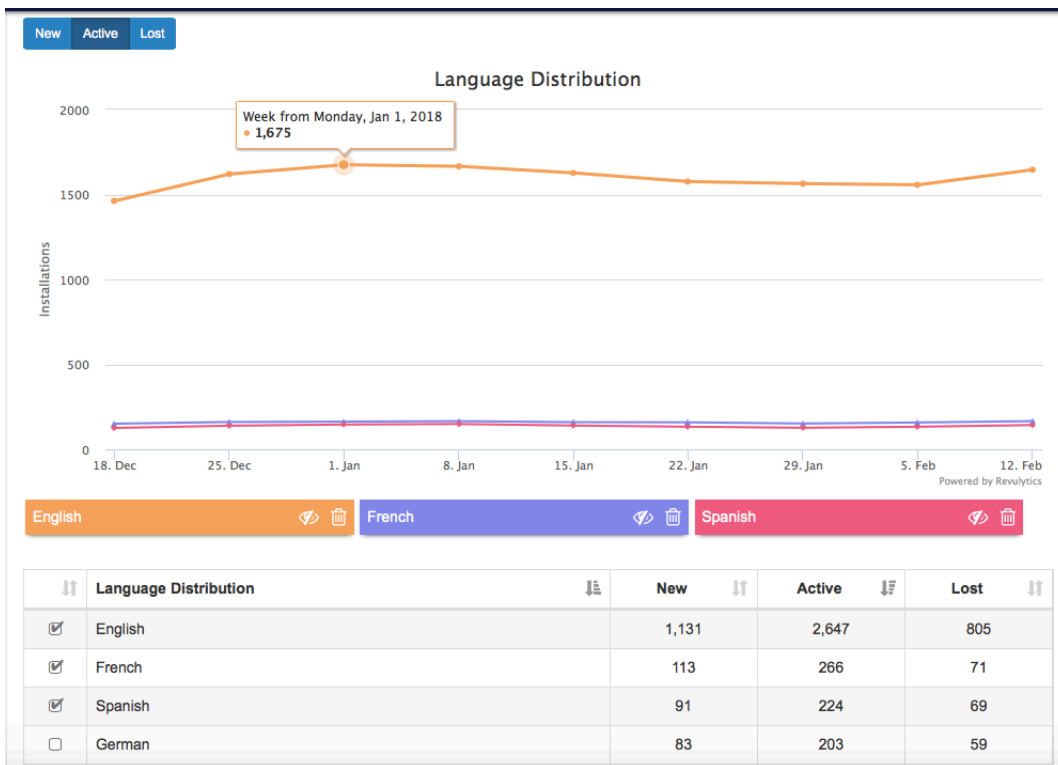


FIGURE 12. An example of statistical analysis of tracked data regarding Language distribution

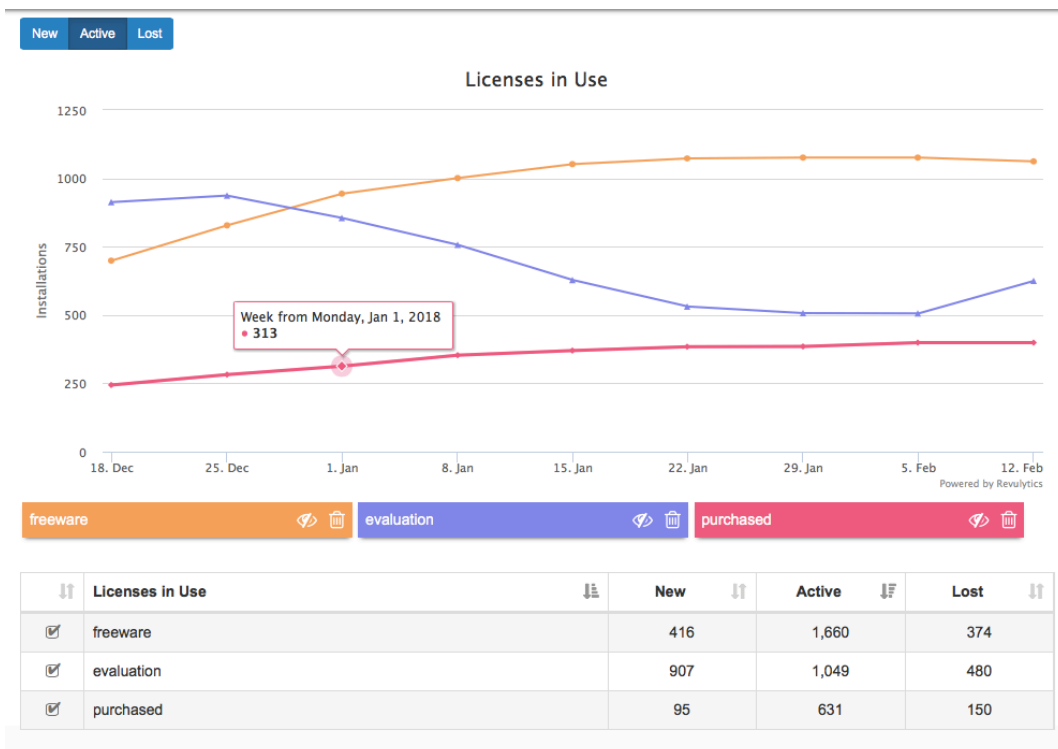


FIGURE 13. An example of license in use statistics

Under the section of Feature & Event Tracking, there are three different statistical analyses of data driven by tracking. First of all, there exists the Basic event tracking analysis (See Figure 14 below) which provides a detailed information regarding every function or action which has been tracked. For example, if a tracking method is used within a specific button in the software, it will generate the statistical analysis of how many times the button has been pressed by how many users, the average number of a particular action usage, and also how many times in a single session the same button has been used.

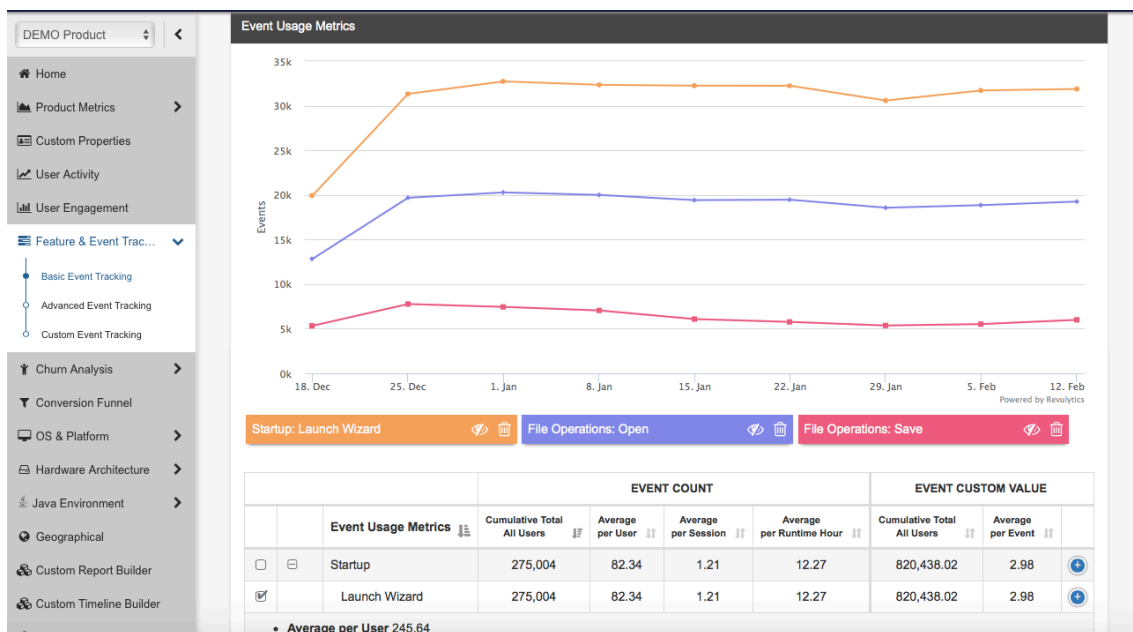


FIGURE 14. An example of statistical analysis of basic event tracking driven data

This basic event tracking data will help by following,

- A concept could be generated regarding which feature has been mostly used by the users.
- The development team could put emphasize on the optimization of those vastly used features.
- The “Never used” features could be identified, and it is possible to analyze how the features could be improved to make them more attractive to the users.

- The Marketing department could generate an idea regarding the features which are needed to be advertised to the users so that the users get noticed about those features.
- The Sales team could make a pricing decision by analyzing the mostly used features and analyzing the dependency of the users on those features which will lead to a higher revenue.

The advanced event tracking section (See Figure 15 below) provides deeper knowledge regarding the usage. In this section the filter and segmentation can be used to examine the data.

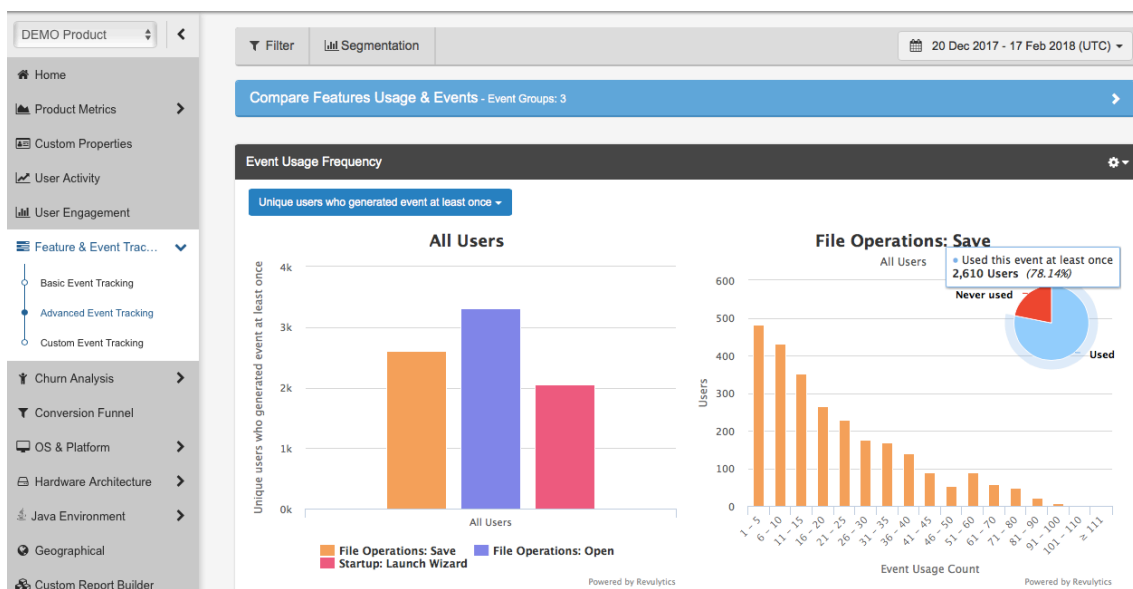


FIGURE 15. An example of statistical analysis of Advanced event tracking driven data

Custom Event Tracking (See Figure 16 below) is the most useful section. By custom event tracking, it is possible to get the duration of a particular feature usage. This event tracking could track the entire setup of the user’s application. As for the user privacy law it is not possible to detect the users, but this event tracking opens a path to analyze the data for understanding the use case of the software for a particular group of users. The beneficial points of this event tracking are the following,

- The Exact duration of a particular feature usage could help the development team to analyze the negative factors of a feature. This will lead to optimize that particular feature if it has been mostly used.
- If a feature is mostly used and it still takes a longer time than expected to perform that particular action, the development team could pin point the reason by getting the data related with users' settings while using the software.
- A proof of concept could be generated by the support team regarding the system requirements to use a specific feature more efficiently. For example, if a heavy feature can be performed within a short moment by having a specific Operating system and platform, this information could be utilized to inform the users regarding the minimum requirement of the system specification while using any feature as well as the entire software.

Custom Event Preview
✕

Event Details	
Timestamp	2018-02-16 14:51:54
Client ID	2958A7C5911ABF77
Event Category	File Operations
Event Name	Open
Event Data	testfl.doc

Product Details	
Version	2
Build	719
Edition	Premium
Language	English

Geo Location	
Country	US
US State	IL

Custom Properties	
⚠ No Custom Properties	

OS & Platform	
OS Type	MS Win Vista
OS Architecture	64-bit
OS Language	English
.NET Versions	2.0;3.5 SP1;3.0 SP2;4...

Hardware Architecture	
Computer Type	Desktop
RAM	1024
CPU Type	Intel Pentium
CPU Cores	2
GPU	Intel HD Graphics 4600
Screen Resolution	1920x1080
Number of Monitors	1

Licensing Details	
License Type	purchased
Key Whitelisted	✕
Key Blacklisted	✕
Key Activated	✕
Key Expired	✕

1 of 25
✕ Close

FIGURE 16. An example of custom event tracking data

A Churn statistical analysis will provide a vast knowledge of how long the users used the software after installation and at which point the users were stopped using the software. It also provides an analysis of which features were used by the lost users and what was the last event used by the lost users. By this churn analysis, it is possible to identify the cases which made a user to stop using the software, which version of the software has the most number of user loss, users from which country have the biggest number of demotivation in terms of using the software or which operating system that particular group of lost users were using etc. This analysis certainly pin points the bad remarks of a feature, the less marketing of a particular place, the less supports of some particular operating systems while using the software (See Figure 17, 18, and 19). This information will help the product management to improve their product to make it more attractive for the users.

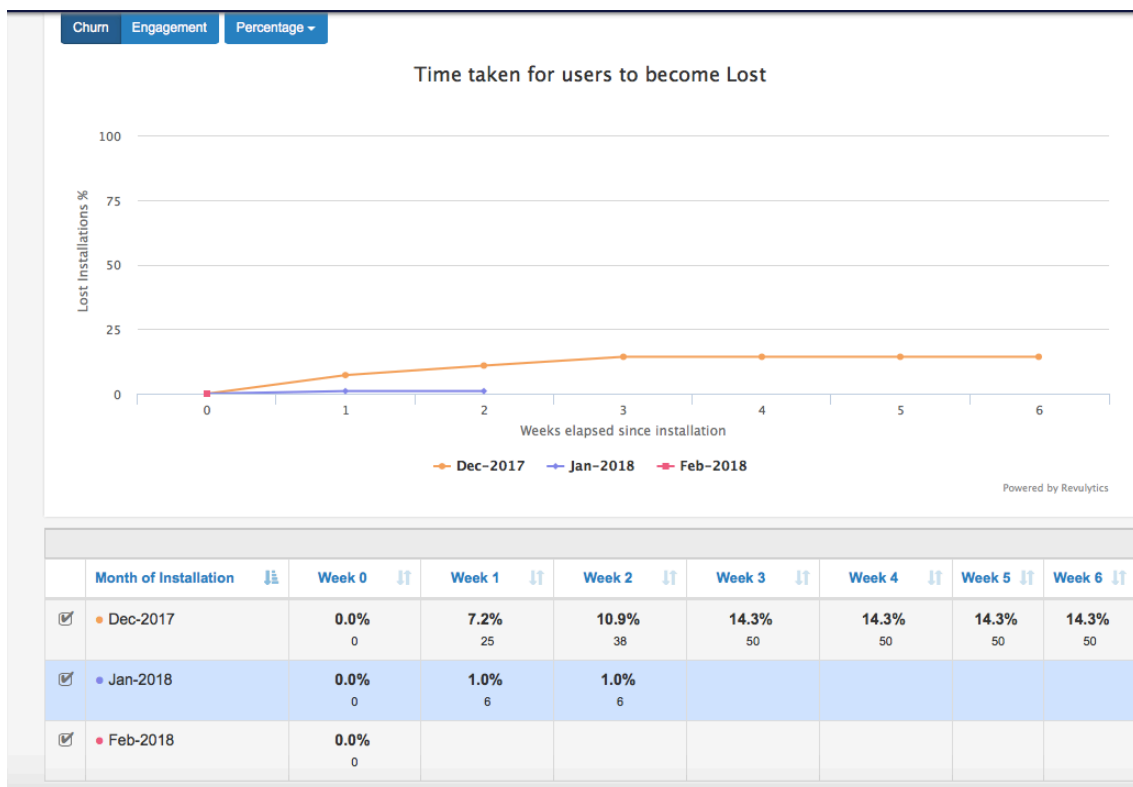


FIGURE 17. An example of statistical analysis of time taken for users to become lost

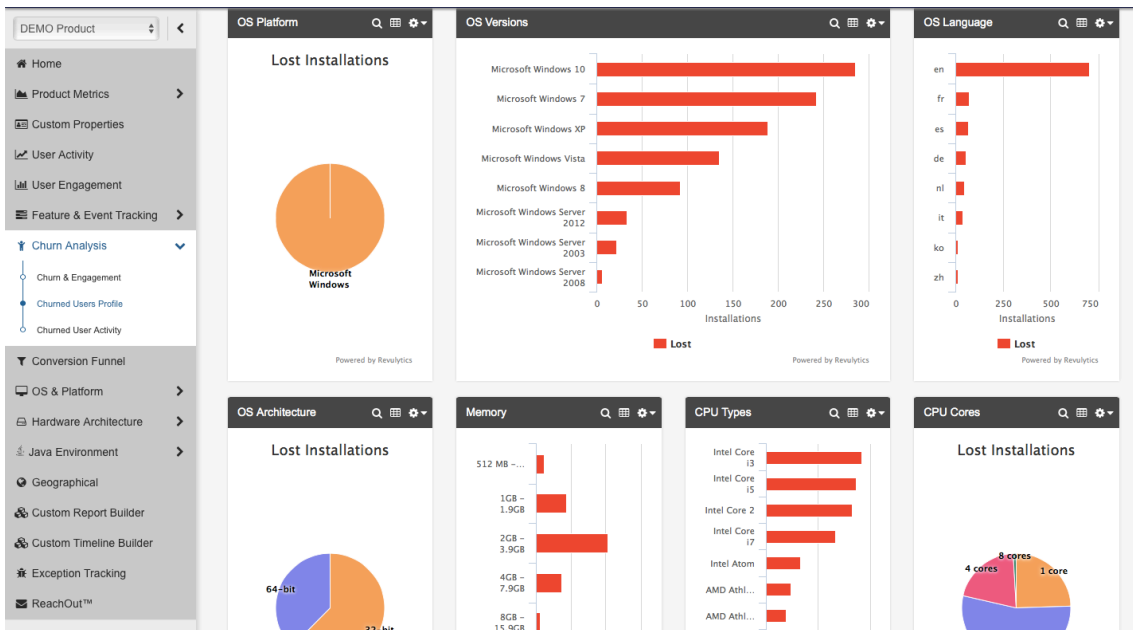


FIGURE 18. An example of statistical analysis of OS architecture of the lost users

Churned User Activity

This report shows you which events are used by Churned users and which of them are never discovered during their lifetime. You may click on the event name to get an Event Usage Frequency distribution, showing you how many times each Churned user used that event before walking away from your product.

Note: This report is only available for Events that are tracked in [Advanced/Filtered](#) mode and NOT for Basic events.

[Hide description...](#)

Events Usage Metrics	EVENT COUNT		EVENT USED AT LEAST ON...		EVENT NEVER USED	
	Cumulative Total	Average per User	User Count	%	User Count	%
Startup: Launch Wizard	91,170	165.46	551	54.88%	453	45.12%
File Operations: Open	90,934	91.30	996	99.20%	8	0.80%
File Operations: Save	32,199	32.26	998	99.40%	6	0.60%
Print Operations: Button Clicked	16,433	31.12	528	52.59%	476	47.41%
Print Operations: Menu Launched	830	22.43	37	3.69%	967	96.31%

FIGURE 19. An example of events used by the lost user before stopped using the software.

The OS & Platform section provides the statistical analysis of the currently used OS Architecture by the users while running the software (See Figure 20 below). This data generates a clear idea of which operating system should be taking care of while developing every single feature. The feature should be optimized in a way that the mostly used architecture could handle the better performance. While testing the “to be released” version of a software, this should also be tested in these different OS architectures to verify.

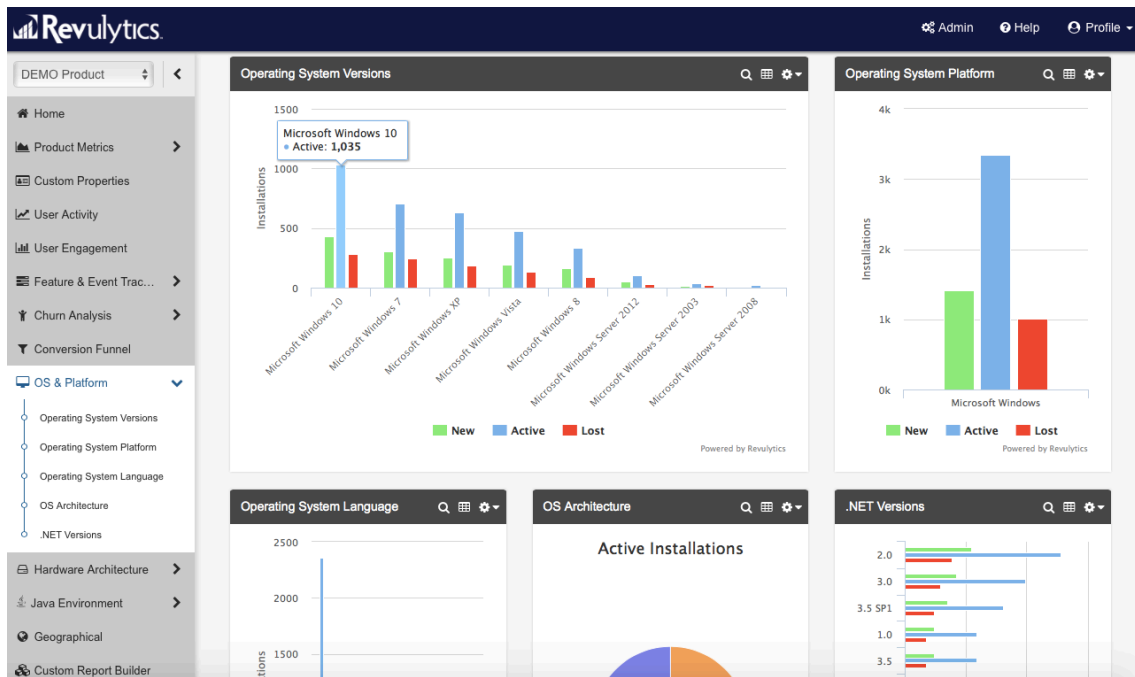


FIGURE 20. An example analysis on OS Architecture

In the Hardware Architecture section, Monitors configuration, resolutions, graphics card specification, Memory allocation, Information regarding CPU types and cores related information have been analyzed in terms of different sections of users such as active, lost or new users (See Figure 21). This analysis would also be a plus point with the OS Architecture analysis. For example, if a software is used for a heavy graphical usage, this analysis would be the hand book to measure the best and worst-case scenarios.

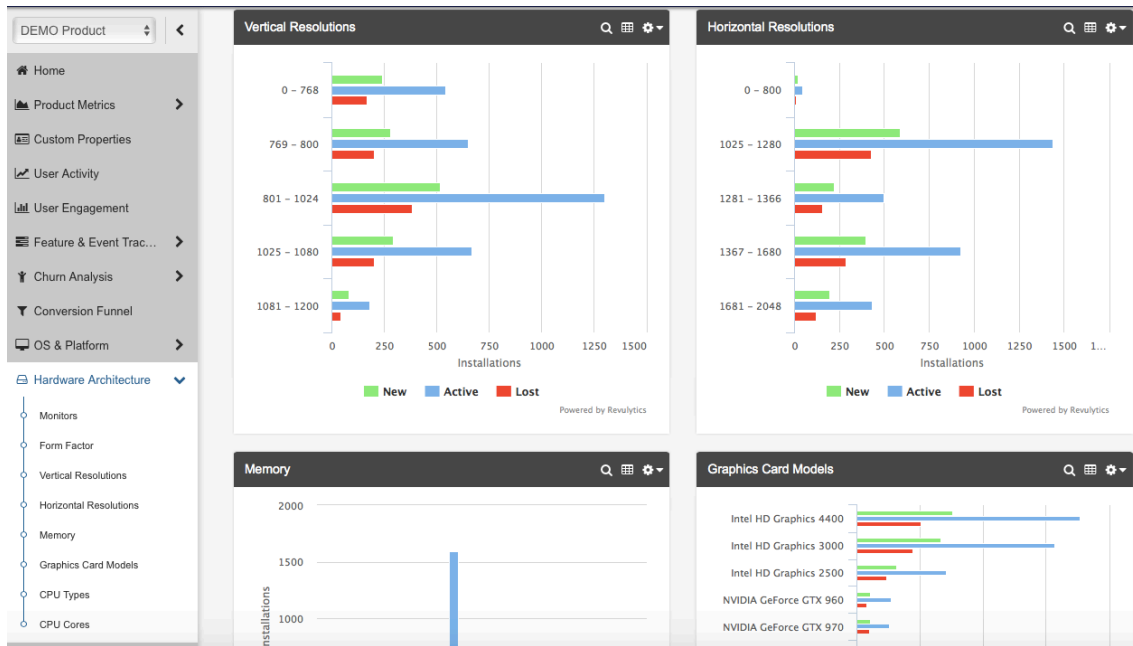


FIGURE 21. An example analysis on Hardware Architecture

Revulytics also generates the statistical analysis in the Java environment, java runtime versions. A Custom report and timeline could also be built through the dashboard for any particular Event.

Another special feature of Revulytics is to track the runtime exception of a software (See Figure 22 and 23). Without usage tracking, it is quite hard to know what sort of exception the users are facing while running a software. The only possible way to know might be the communication between users and the support team if any crashing of the software happens. But it is not a good way as the user might only say what he or she was doing with the software while the crash happened. This information gives a vague idea of a possible exception. Even though a software might have a log receiving mechanism, it is still a time-consuming way to figure out the actual reason behind the exception by debugging and reading through the log. In this case, Revulytics provides the best solution. It catches the run time exception and generates the report in the dashboard. With the stack trace that Revulytics provides, a developer does not need to debug further to

figure out the exception. Within a short moment, a developer gets the clear information regarding where, in which line of the code the exception has happened. To keep a bug free software, this Exception tracking plays a vital role.

The screenshot shows a window titled "Exception Preview" with a close button (X) in the top right corner. The window is divided into several sections:

- Exception Details:**
 - Timestamp: 2018-02-16 14:51:54
 - Client ID: FDB898260D3558C2
 - Exception Class: File
 - Exception Method: getFileList
 - Exception Message: **'NoneType' object has n...**
 - Exception Stack Trace: [View Stack Trace](#)
- Product Details:**
 - Version: 2
 - Build: 719
 - Edition: Premium
 - Language: English
- Geo Location:**
 - Country: US
 - US State: IL
- Custom Properties:**
 - ⚠ No Custom Properties**
- OS & Platform:**
 - OS Type: MS Win Vista
 - OS Architecture: 64-bit
 - OS Language: English
 - .NET Versions: [2.0;3.5 SP1;3.0 SP2;4....](#)
- Hardware Architecture:**
 - Computer Type: Desktop
 - RAM: 1024
 - CPU Type: Intel Pentium
 - CPU Cores: 2
 - GPU: NVIDIA GeForce GTX 9...
 - Screen Resolution: 1920x1080
 - Number of Monitors: 1
- Licensing Details:**
 - License Type: purchased
 - Key Whitelisted: **✖**
 - Key Blacklisted: **✖**
 - Key Activated: **✖**
 - Key Expired: **✖**

At the bottom left, it says "1 of 20". At the bottom right, there is a "Close" button with an X icon.

FIGURE 22. An example of Exception tracking by Revulytics

The screenshot shows a window with a stack trace. The text inside the window is:

```
Traceback (most recent call last):
File "main.py", line 1, in <module>
TypeError: 'NoneType' object has no attribute '__getitem__'
```

At the bottom right of the window, there is an "OK" button.

FIGURE 23. An example stack trace provided with the exception tracking

Another useful feature from Revulytics is “ReachOut™” in-application messaging (See Figure 24). This feature helps a marketing team to send a direct message to the active users if certain criteria set by the marketing team is matched. With this feature, advertising of new features, any interesting information related with the software which will attract the user to extend the subscription, information regarding the upcoming release of the software, providing user up-to-date information regarding the software, could be performed. However, a software company could minimize the expenses for advertising, marketing, observing society events for the users by using this feature.

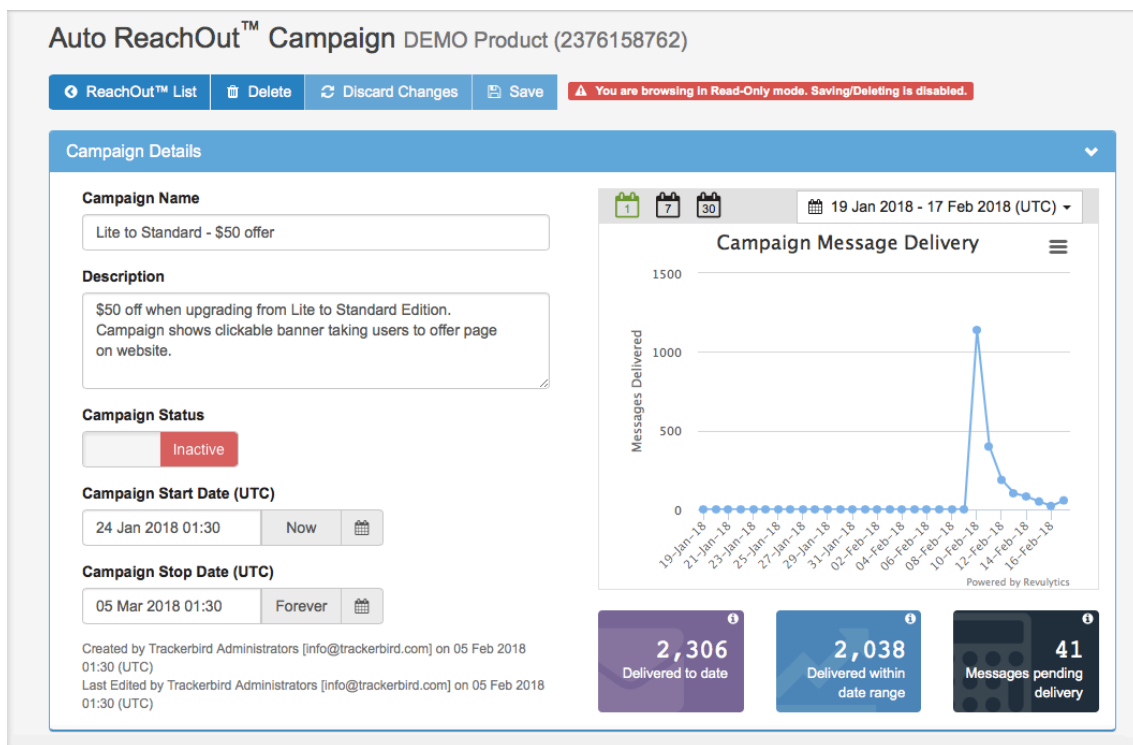


FIGURE 24. An example of ReachOut™ Campaign

A software company does not need to keep anymore tracking on the geographical substitution of their entire number of users. Revulytics provides a graphical visualization of the world map, the number of customers from different countries, the number of new, active or lost customers from different countries. (See Figure 25). This accurate information can be useful for the following cases,

- Prioritizing the marketing procedure in less customer-based countries to try to generate revenue from those countries too.
- A Localization of software could be performed for the countries from where a good number of customers is using the software.
- A Customer support team could be provided for those countries.
- A good marketing report could be generated with this graphical presentation to attract more customers.
- The need of a software in a specific country can be predictable with the growing number of customers. This will lead to make a perfect business plan.

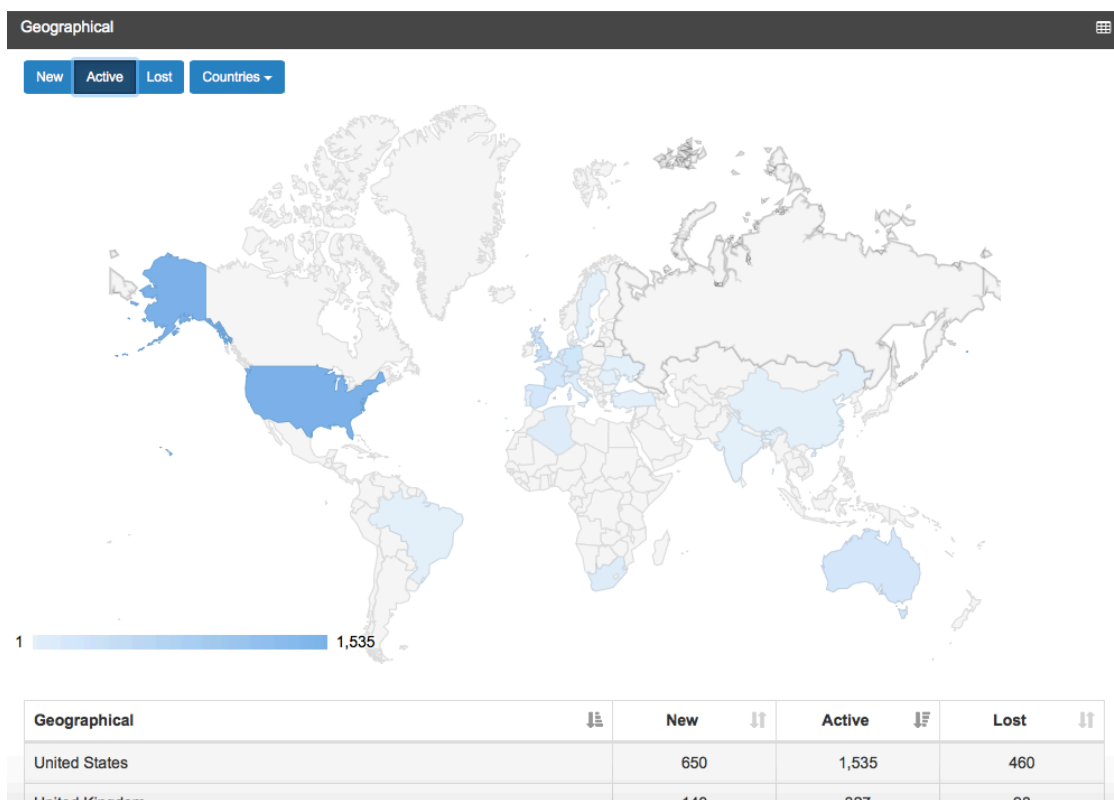


FIGURE 25. An example of geographical presentation of customers by Revulytics.

7 CONCLUSION

In Software businesses, the ultimate satisfaction of the customers plays a vital role in terms of generating a higher revenue and creating a stable position world-wide. Usage tracking works as a relation builder between the customers and the software companies. A concept could turn into a proof of concept by analysing the usage data of the customers. With the help of a tracking usage data software, the usability of an entire software could be proved and this is much more efficient than reading out the mind and usage behaviour of the users.

To plan a prosperous business, to maintain a good product development cycle, to predict the reason of the ups and downs of a business, nothing can beat usage tracking. Moreover, a business company should provide first priority in usage tracking from the beginning of its establishment to understand their customers, their needs as well as the business market and scope.

This bachelor's thesis is not only beneficial for a developer to integrate the analytical usage tracking system in a software, but also for a software business holder including the entire team of a company to leverage the business management. The term "Usage Analytics" has been presented from the scratch to an extent that I believe it could be used as a check book while establishing a software business. In addition, the integration has been performed in cooperation with a software company and by using a widely used software, Solibri Model Checker. This proves the successfulness of this bachelor's thesis work without any confusion.

REFERENCES

1. Giardino, C., Wang, X. and Abrahamsson, P., 2014, June. Why early-stage software startups fail: a behavioral framework. In *International Conference of Software Business* (pp. 27-41). Springer, Cham.
2. Givon, M., Mahajan, V. and Muller, E., 1995. Software piracy: Estimation of lost sales and the impact on software diffusion. *The Journal of Marketing*, pp.29-37.
3. Hartwig, F. and Dearing, B.E., 1979. *Exploratory data analysis*(Vol. 16). Sage.
4. Gelman, A., 2004. Exploratory data analysis for complex models. *Journal of Computational and Graphical Statistics*, 13(4), pp.755-779.
5. Miles, M.B., Huberman, A.M. and Saldana, J., 2013. *Qualitative data analysis*. Sage.
6. Cramer, D., 2003. *Advanced quantitative data analysis*. McGraw-Hill Education (UK).
7. Nyce, C. and CPCU, A., 2007. Predictive analytics white paper. *American Institute for CPCU. Insurance Institute of America*, pp.9-10.
8. Research data management. Malaysia: NTU Libraries blog, Date of retrieval 09.02.2018,
<https://blogs.ntu.edu.sg/lib-datamanagement/data-lifecycle/>
9. Cluvio Pricing, Features, Review & Comparison of Alternatives, Date of retrieval 18.02.2018,
<https://www.getapp.com/business-intelligence-analytics-software/a/cluvio/>

10. Cluvio, Pricing, Date of retrieval 18.02.2018,
https://www.cluvio.com/pricing.html?utm_source=capterra&utm_medium=ppc&utm_campaign=capterra-general&utm_content=pricing
11. Usage Analytics, Coveo, Date of retrieval 18.02.2018,
<https://www.coveo.com/en/platform/usage-analytics>
12. Coveo for Sitecore Editions & Pricing, Coveo, Date of retrieval 18.02.2018,
<https://www.coveo.com/en/platform/usage-analytics>
13. Pendo Pricing, Features, Review & Comparison of Alternatives, Date of retrieval 18.02.2018,
<https://www.getapp.com/business-intelligence-analytics-software/a/pendo/>
14. Keen IO Pricing, Features, Review & Comparison of Alternatives, Date of retrieval 18.02.2018,
<https://www.getapp.com/business-intelligence-analytics-software/a/keen-io/>
15. Analytics Plus Pricing, Features, Review & Comparison of Alternatives, Date of retrieval 18.02.2018,
<https://www.getapp.com/business-intelligence-analytics-software/a/analytics-plus/>
16. Pricing, Analytics Plus, ManageEngine, Date of retrieval 18.02.2018,
<https://www.manageengine.com/analytics-plus/get-quote.html>
17. Pricing, Revulytics, Date of retrieval 18.02.2018,
<https://www.revulytics.com/pricing>
18. Review, Revulytics, Date of retrieval 18.02.2018,
<https://comparisons.financesonline.com/revulytics-usage-intelligence-vs-sap-crystal-reports>

19. Lubowicka K., [UPDATE] How will GDPR Affect Your Web Analytics Tracking?, PIWIK Pro, Blog, Date of retrieval 18.02.2018,
<https://piwik.pro/blog/how-will-gdpr-affect-your-web-analytics-tracking/>
20. General Data Protection Regulation, Wikipedia, Date of retrieval 18.02.2018,
https://en.wikipedia.org/wiki/General_Data_Protection_Regulation
21. Tankard, C., 2016. What the GDPR means for businesses. *Network Security*, 2016(6), pp.5-8.
22. Albrecht, J.P., 2016. How the GDPR Will Change the World. *Eur. Data Prot. L. Rev.*, 2, p.287.
23. Consent, GDPR Key Changes, EU GDPR Portal: Trunomi, Date of retrieval 18.02.2018,
<https://www.eugdpr.org/the-regulation.html>
24. Maldoff, G., 2017. The risk-based approach in the GDPR: interpretation and implications. *IAPP*, Date of retrieval 18.02.2018,
https://iapp.org/media/pdf/resource_center/GDPR_Study_Maldoff.pdf
25. Ryan, J., 2017, August, Here is what GDPR consent dialogues could look like. Will people click yes?, Ireland: PageFair Limited, blog, Date of retrieval 18.02.2018,
<https://pagefair.com/blog/2017/gdpr-consent/>
26. IDEs vs. Build Tools: How Eclipse, IntelliJ IDEA & NetBeans users work with Maven, Ant, SBT & Gradle, ZeroTurnAround, Date of retrieval 10.02.2018,
<https://zeroturnaround.com/rebellabs/ides-vs-build-tools-how-eclipse-intellij-idea-netbeans-users-work-with-maven-ant-sbt-gradle/>
27. Xcode, Wikipedia, Date of retrieval 10.02.2018,
<https://en.wikipedia.org/wiki/Xcode>

28. What is DYLIB File?, Lifewire, Date of retrieval 10.02.2018,
<https://www.lifewire.com/dylib-file-2620908>
29. Understanding the Bamboo CI Server, Bamboo Support: Bamboo 6.3 Documentation: Getting started guide, Date of retrieval 10.02.2018,
<https://confluence.atlassian.com/bamboo/understanding-the-bamboo-ci-server-289277285.html>
30. Smart Mirroring for Bitbucket Cloud, Bamboo Support: Documentation: Repositories: Repository Settings, Date of retrieval 10.02.2018,
<https://confluence.atlassian.com/bitbucket/smart-mirroring-for-bitbucket-cloud-838427434.html>
31. Solibri Model Checker, Product Functionality, Date of retrieval 10.02.2018,
<https://www.solibri.com/products/solibri-model-checker/>
32. Revulytics, Usage Analytics, Date of retrieval 10.02.2018,
<https://www.revulytics.com>
33. Gosling, J., Joy, B., Steele, G.L., Bracha, G. and Buckley, A., 2014. *The Java language specification*. Pearson Education.
34. What is SWIG?, SWIG, Date of retrieval 10.02.2018,
<http://www.swig.org/exec.html>
35. SWIG, Wikipedia, Date of retrieval 10.02.2018,
<https://en.wikipedia.org/wiki/SWIG>
36. Wireshark, Wikipedia, Date of retrieval 10.02.2018,
<https://en.wikipedia.org/wiki/Wireshark>

37. About Wireshark, Wireshark, Date of Retrieval 10.02.2018,
<https://www.wireshark.org>
38. Gurses, L., 2015, October, Improving Code Quality with PMD and Eclipse, EclipseZone, Date of retrieval 10.02.2018,
<https://www.eclipsezone.com//articles/pmd/>
39. Attard A., Why should we use Maven?, Java Creed, Date of retrieval 10.02.2018,
<http://www.javacreed.com/why-should-we-use-maven/>
40. Initializing the configuration, Developer zone, Revulytics, Date of retrieval 16.02.2018,
<https://devzone.revulytics.com/docs/v5/CPP/mac/sdk-configuration.html#initializing-the-configuration>
41. Providing further data, Product Details, Developer zone, Revulytics, Date of retrieval 17.02.2018,
<https://devzone.revulytics.com/docs/v5/CPP/mac/sdk-configuration.html#providing-further-data>
42. Basic SDK Control, Developer zone, Revulytics, Date of retrieval 17.02.2018,
<https://devzone.revulytics.com/docs/v5/CPP/mac/basic-sdk-control.html#c.ruiStartSession>
43. "ruiSetLicenseData", SDK Initialization and Configuration, Developer zone, Revulytics, Date of retrieval 17.02.2018,
<https://devzone.revulytics.com/docs/v5/CPP/mac/sdk-configuration.html#c.ruiSetLicenseData>

44. Feature / Event Tracking, Developer zone, Revulytics, Date of retrieval 17.02.2018,
<https://devzone.revulytics.com/docs/v5/CPP/mac/event-tracking.html>
45. Exception Tracking, Developer zone, Revulytics, Date of retrieval 17.02.2018,
<https://devzone.revulytics.com/docs/v5/CPP/mac/exception-tracking.html>
46. Bonnici, A., customer supporter, Revulytics. Re: How to track-user is using trial version. Support ticket message, Ticket number: 826, 27.09.2017.
47. How is Revulytics Usage Intelligent affected by proxy or firewall environments?, Helpdesk, Revulytics, Date of retrieval 17.02.2018,
<https://helpdesk.revulytics.com/knowledgebase.php?article=57>
48. What is the size threshold for the Revulytics Usage Intelligence log file and can this be changed?, Helpdesk, Revulytics, Date of retrieval 17.02.2018,
<https://helpdesk.revulytics.com/knowledgebase.php?article=58>
49. How many times does the Revulytics Usage Intelligence SDK call home and how much traffic does each call generate?, Helpdesk, Revulytics, Date of retrieval 17.02.2018,
<https://helpdesk.revulytics.com/knowledgebase.php?article=61>