LoadSplunker

Dashboard

LUND UNIVERSITY
Campus Helsingborg

LTH School of Engineering at Campus Helsingborg
Department of Computer Science

Bachelor thesis:
Justin Rees
Martin Szuter

LTH School of Engineering
Lund University
Box 882
SE-251 08 Helsingborg
Sweden

LTH Ingenjörshögskolan vid Campus Helsingborg
Lunds universitet
Box 882
251 08 Helsingborg

Printed in Sweden
Lunds universitet
Lund 2016

Abstract

At the IT department of IKEA, test specialists who are responsible for software testing are currently using a suite from Hewlett Packard called HP Application Lifecycle Management that includes an analysis tool for test result evaluation. During the implementation of this thesis work, the real world demand for a replacement of the analysis tool became apparent, as the interface of the analysis tool is regarded as being unnecessarily complicated to use, yet the test reports it generates are too simplistic and aesthetically unappealing. Furthermore, these reports are unduly static and lacks in interactivity if amendments are needed further down the line. In turn, this often leads to misunderstandings between test specialists and stakeholders, creating a feedback loop where a report is submitted, sent back for clarification, reworked and finally re-submitted. Because of this, test reporting becomes a very time consuming process and test specialists often find themselves spending more time writing reports than actually performing tests. Employees find this frustrating as well as ungratifying as the reports are perceived as being of lower quality than expected. The authors witnessed employees regularly having to copy and paste test result data into excel spreadsheets and bounce manually written test reports back and forth between stakeholders for weeks at a time, with their hands tied during the downtime. Previous students from Lund University have developed a substitute for the *HP Analysis Tool* called "LoadSplunker" for their thesis work. LoadSplunker utilises an analytical tool called "Splunk" and has been configured to extract the raw test result information used by *HP Analysis Tool* and produce XML-files containing the test results. However, the previous students did not develop a suitable dashboard in Splunk to present the test results in. Additionally, the previous solution depended on Windows, and a requirement from IKEA was that LoadSplunker should run on Linux. This thesis work reviewed and modified the previous solution to run within the Linux sphere. After conducting a survey and gathering requirements from IKEA, three configurable Splunk dashboards were implemented, with the aim of presenting the test results in a manner that is both easy to read and understand. The three dashboards developed are each categorized and aimed at stakeholders with varying technical proficiency and give an overview of selected test runs. Replacing the Analysis Tool with Splunk could eliminate almost all of the downtime associated with the test reporting process and the authors feel this thesis assignment has successfully implemented a proof-of-concept with LoadSplunker. While there is potential for test reports being completely replaced with Splunk for recurring or regularly scheduled tests, test reports can probably never be fully automated for large projects or new test runs. For large projects, focus should instead be shifted towards creating individual dashboards in Splunk on a per-project basis.

Keywords: IKEA, HP, ALM, Performance Center, LoadRunner, Analysis, Splunk, LoadSplunker, Dashboard.

Sammanfattning

Testspecialister vid IT avdelningen på IKEA, som ansvarar för all testning utav mjukvara, använder sig för tillfället av ett verktyg utvecklat av Hewlett Packard som kallas *HP Application Lifecycle Management*. Det här analysverktyget används för att utvärdera erhållna testresultat. Ofta visar sig dock analysverktyget vara onödigt komplicerat att använda samtidigt som rapporterna det genererar blir för otydliga och innehåller för lite information för det eftersträvande syftet. Rapporterna blir dessutom alldeles för statiska och saknar interaktivitet, som hade underlättat om förbättringar eller ändringar behöver införas efterhand. Detta leder ofta till missuppfattningar mellan test specialister och intressenter, vilket ofta medför en rapportrundgång där en rapport skickas in, skickas tillbaka för att förtydligas, omarbetas och slutligen skickas in igen. På grund av detta brukar rapporteringen av testresultat bli onödigt tidskrävande och test specialister känner att de spenderar mer tid på att skriva rapporter än att faktiskt utföra test. De anställda tycker att detta är frustrerande och lönlöst då rapporterna uppfattas vara av lägre kvalité än förväntat. Tidigare studenter från Lunds Universitet har under ett examensarbete lyckats utveckla ett substitut för det nuvarande använda analysverktyget *HP Analysis Tool* som de kallade för *LoadSplunker. LoadSplunker* utnyttjar ett externt analysverktyg som kallas “Splunk” och har konfigurerats för att samla in det råa test resultatet *HP Analysis Tool* använder och utifrån det generera XML-filer innehållande test datan. Dock så hann inte de föregående studenterna med att skapa en passande översiktspanel(dashboard) för att kunna presentera resultaten i. Den tidigare lösningen var även utvecklad för Windows, och ett krav från IKEA var att LoadSplunker måste kunna köra på Linux. Under detta examensarbete utvärderades och vidareutvecklades de förra studenternas lösning så att den kunde köra på Linux. En enkät skrevs och skickades ut till anställda på IKEA samt att projektkrav samlades in från intressenterna. Detta ledde till implementeringen av tre stycken konfigurerbara Splunk-översiktspaneler, i syfte att kunna presentera testresultat på ett sätt som är både enkelt att läsa och förstå. De tre utvecklade panelerna är var och en kategoriserade och inriktade mot målgrupper med varierande teknisk färdighet, och ger en överblick över valda test körningar. Genom att ersätta HP Analysis med Splunk kan i stort sett all dödtid associerad med rapportering av tester likvideras, och författarna känner att ett lyckat konceptbevis har implementerats i detta examensarbete.
Medan där finns potential för en fullständig ersättning av test rapporter med Splunk för upprepande eller schemalagda testkörningar, så kan förmodligen inte stora projekt eller helt nya testkörningar bli fullt automatiserade på grund av det stora omfånget. För större projekt bör fokus istället ligga på att skapa individuella dashboards i Splunk för varje projekt i syfte att fungera som en ersättning till manuellt skrivna rapporter.

Nyckelord: IKEA, HP, ALM, Performance Center, LoadRunner, Analysis, Splunk, LoadSplunker, Dashboard.

Foreword and Acknowledgements

This thesis work is written for our Bachelor's degree in Computer Science and Engineering at Lund University. The thesis was made possible by the IT department of IKEA, whose employees have made our work a most pleasant experience.

We would like to personally thank Jacek Goralski at IKEA IT for his support and guidance throughout this thesis work. We would also like to extend our gratitude to Magnus Johansson for taking time out of his busy IKEA work schedule to teach us the basics of the Splunk syntax and sharing his expertise of the software with us. Finally, we would also like to thank our academic supervisor Christian Nyberg for his feedback and guidance.

List of contents

1 Introduction

IKEA is one of the largest companies in the world with over 9000 servers handling over 250 million cash transactions and 1.2 billion website hits each year [1]. In order to manage this, a large amount of hardware and software is required, and everything has to be tested before being deployed to the live environment. This in turn means that, in order to test, one first has to simulate the real life conditions that occur in a live environment. One of the tools IKEA uses to simulate this much traffic is HP Performance Center [2], see figure 1.

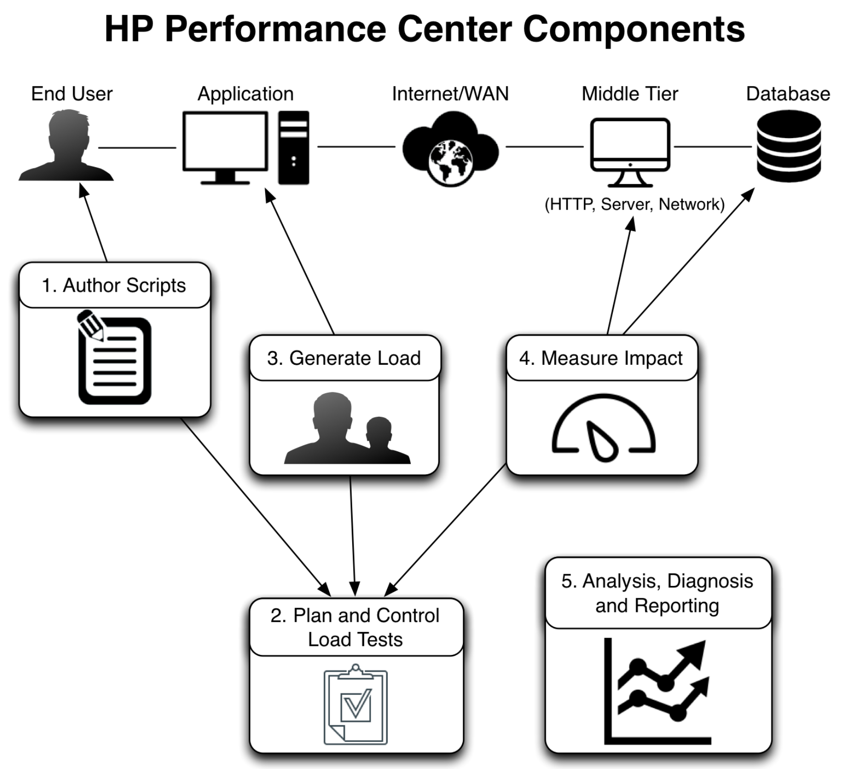


Figure 1. Components of the HP Performance Center tool [31].

Performance Center is a a software with a web-based administration tool, that simulates user activity by generating information between applications or by simulating interactions such as mouse movements, keypresses or user navigation and storing these interactions as scripts. Performance Center can generate scripts via recording the test specialists actions, for example script A: “Browse product, add to cart, checkout, payment transaction”. Script A can then be configured to run for example 1000 times within 5 minutes. Performance Center can also create scripts simply by observation, such as logging HTTP requests between a web browser and a server. This is only one small example of what Performance Center can be used for. Performance Center is the prefered tool for software testing at IKEA and they have their own in-house test center, with most of the software development being outsourced overseas. This complicates communication between the different departments and puts a high demand on the quality of test reporting. After every test run, a report is written manually by the in-house test specialists and sent to the various stakeholders. Writing these reports can be very time consuming and monotonous. Test specialists have expressed a need for these reports to be generated automatically and be made dynamic so that information can more easily be added and modified.

1.1 Background

This bachelor's thesis work was composed to be a continuation of a thesis work completed by previous students. The previous students attempted to fully replace the HP Analysis tool, which is used to analyse and present the test data generated with Performance Center, with another analytical tool called *Splunk* [3]. Splunk (the product) captures, indexes and correlates real-time data in a searchable repository from which it can generate graphs, reports, alerts, dashboards and visualizations. However, completely replacing the Analysis tool with Splunk proved too large of an undertaking and the previous students delimited their thesis to extracting the raw test result data from Performance Center and uploading it to Splunk. Therefore, some of the goals in the preceding thesis work were only partially implemented but not finalised. For this thesis, the original goal of the previous thesis assignment, that is, to replace the *HP Analysis* tool with Splunk remains. In order for Splunk to be an adequate replacement it is important that all the same information can be displayed in Splunk as in HP Analysis. Therefore, data loss should be minimal or non-existent. The need for replacing the Analysis tool stems from the inefficiency of the current test reporting workflow, see figure 2 and 3.

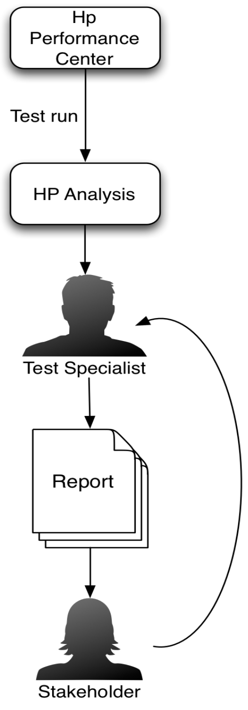


Figure 2:
Current workflow without LoadSplunker/Splunk

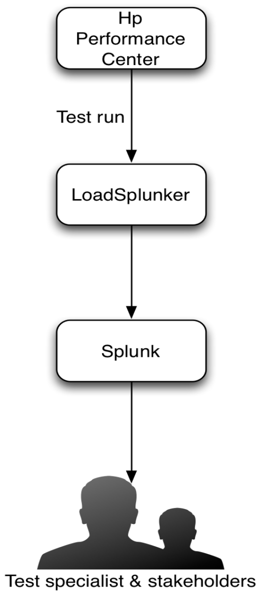


Figure 3:
Expected workflow with LoadSplunker/Splunk

Figure 2 shows the current workflow of the testing and reporting process. While interviewing employees at IKEA it became clear that there is a high risk of a feedback loop for the final report. This is included in the figure.

Figure 3 shows the improved workflow process with the LoadSplunker and Splunk replacement tools. Instead of a circulating report the aim is to have a dynamic dashboard that both test specialists and stakeholders can interact with.

Since this thesis work is a continuation of work completed by previous students, the authors first had to gain insight into what had already been achieved. The previous students main goal was to extract the data from Performance Center and funnel it into Splunk.

To achieve this they first needed to monitor when a test run had been completed in Performance Center so they could trigger their LoadSplunker [4] extraction software. Their solution involved the SiteAdmin COM-based API suite [5], using COM4J [6] to bridge the gap between SiteAdmin and their Java solution. Unfortunately, since their solution utilises SiteAdmin it became reliant on the Windows operating system, whereas IKEA's Splunk installation runs on Linux. This led to Performance Center's test data having to be extracted from IKEA's Windows server dedicated to storing test results to a separate Windows Server running the LoadSplunker software, only to be funneled into a third server which was IKEA's Linux server running Splunk, see figure 4. There were two problematic outcomes that followed as a consequence. Firstly, it led to an unnecessary waste of resources and added delay time. Secondly, this also posed a security risk for IKEA since they now had a *rogue* Windows/LoadSplunker server in their network. *Rogue* meaning it had not undergone the same security screening and does not have the same security features or certificates as required by servers within the network.

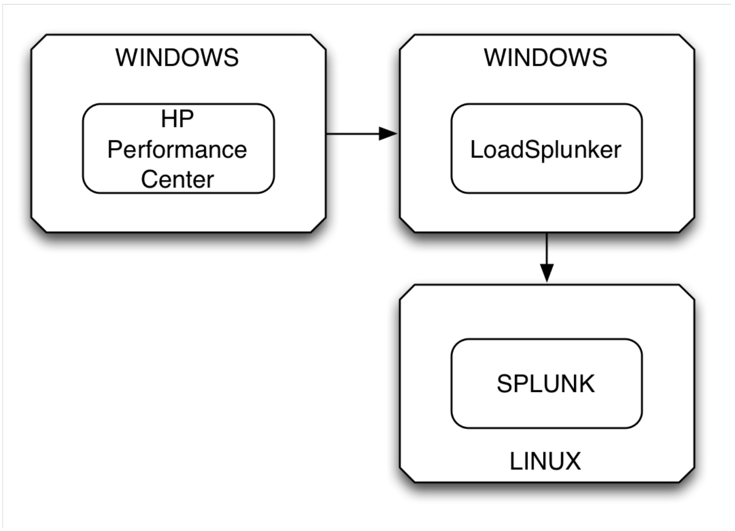


Figure 4. Previous solution with LoadSplunkers Windows dependency

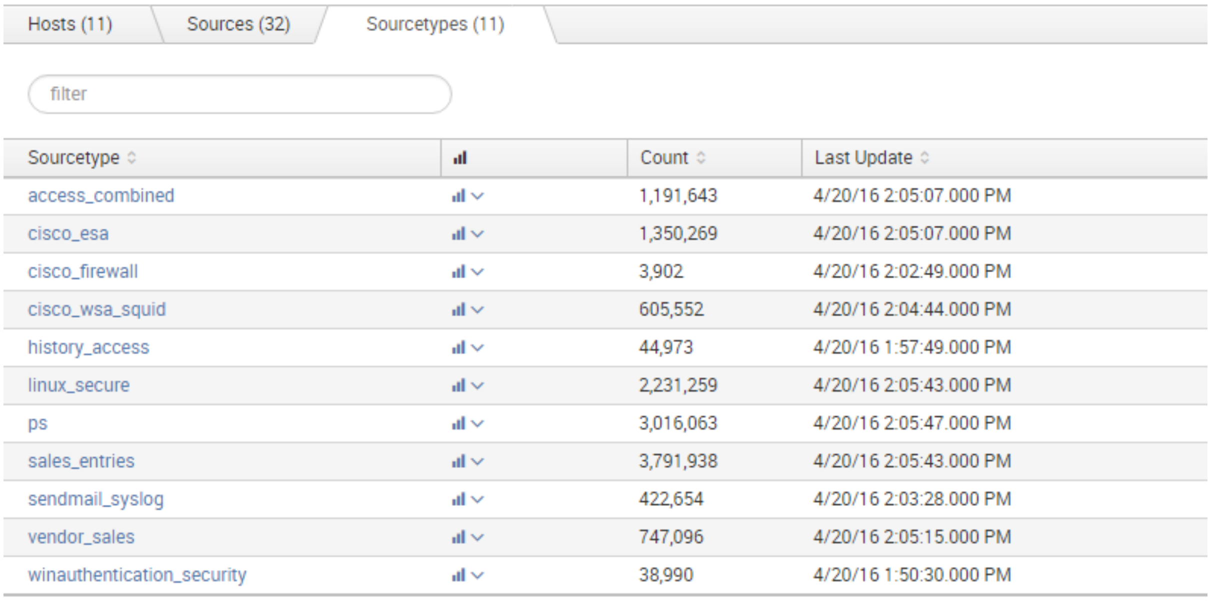
To resolve these issues, an attempt was made to extract the test data straight from Performance Center's database and get away from the SiteAdmin dependency. This was not within the original scope of this thesis work but it quickly became apparent that this was an essential requirement if LoadSplunker should ever became a realistic replacement for Analysis tool. This solution is documented in more detail in chapter 5.

Another issue with the method implemented by the previous students was that it extracted the test data in bulk. This meant a loss of structure along the way, resulting in the user having to do field extractions on *all* available data. The only sources entering Splunk were two tables called "Event_meter" and "Event_map" with blank field values, instead of the expected structure with hosts, sources and sourcetypes, similar to figure 5. To put this in perspective, if one would normally want to get a figure for the sales revenues one would simply enter into the Splunk syntax something along the lines of:

"sourcetype=vendor_sales | stats sum(price)".

Instead, the user is now forced to filter out the data at search time so the search instead becomes:

“source="\\xml\\splunk\\Event_meter.xml" EventName=Transaction_Action_Step_5 | join type=inner EventID [search source="*\\xml\\splunk\\Event_map.xml" | where EventType="Transaction"]*
|rename Transaction as vendor_sales
|rename NULL as price |stats sum(price)”



Sourcetype		Count	Last Update
access_combined		1,191,643	4/20/16 2:05:07.000 PM
cisco_esa		1,350,269	4/20/16 2:05:07.000 PM
cisco_firewall		3,902	4/20/16 2:02:49.000 PM
cisco_wsa_squid		605,552	4/20/16 2:04:44.000 PM
history_access		44,973	4/20/16 1:57:49.000 PM
linux_secure		2,231,259	4/20/16 2:05:43.000 PM
ps		3,016,063	4/20/16 2:05:47.000 PM
sales_entries		3,791,938	4/20/16 2:05:43.000 PM
sendmail_syslog		422,654	4/20/16 2:03:28.000 PM
vendor_sales		747,096	4/20/16 2:05:15.000 PM
winauthentication_security		38,990	4/20/16 1:50:30.000 PM

Figure 5. Example of a well structured Splunk database.

This would therefore also have to be resolved.

1.2 Goal

The primary goal of this thesis is to replace the HP LoadRunner Analysis tool with LoadSplunker. Furthermore the goal is to implement Loadsplunker in a format that will be user friendly and efficient for the needs of the IT department at IKEA. This entails the following:

- Reconfigure Loadsplunker to run on the correct operating system used by the IT department at IKEA, namely Linux.
- Allow the test reports to be viewable in an understandable format and easily modified by the test specialists according to their personal needs.
- Implement a role-based system that makes certain information viewable only for intended users (e.g. developers, test specialists, business administrators). There is a duality to this goal as graphs and statistics intended for a test specialist will be too detailed and complex for say a business administrator.

1.3 Problem Specification

In order to make the objective of this thesis work clearer, there is a need to clarify and specify the problems that this thesis work will attempt to answer. They are as follows:

Review of implemented software:

- How is *LoadSplunker* currently implemented (structure, database, Splunk)?
- How complicated is it to learn and further develop *LoadSplunker*?
- How can LoadSplunker be converted from Windows-compatibility to Linux-compatibility?

Automatic report creation (Dashboard):

- How can the generated XML-file be loaded into the dashboard to present the test results?
- How can *LoadSplunker* determine the type of test data and choose a suitable template for the dashboard to present it with?
- How can *LoadSplunker* be made easier to use than the currently used *Analysis tool*?

Role based system:

- How to gain access to user privileges at IKEA?
- How can *Splunk* regularly be updated of changes to user privileges at IKEA?
- How to Implement user restriction in *LoadSplunker* based on the user privileges?

Installation:

- How can this system be installed in IKEA's environment?

Research:

- How can a survey be formulated to determine the current shortcomings of the Analysis Tool and ensure that *LoadSplunker's* dashboards meet the requested requirements?
- What development options can be undertaken by future developers to make *LoadSplunker* a complete analysis tool?

1.4 Delimitations

At the start of this thesis work the authors aspired to completely replace HP Analysis with LoadSplunker. However, due to the size and complexity of HP Performance Center and the Analysis tool, the authors primarily focused on implementing LoadSplunker as a proof-of-concept, rather than a comprehensive replacement of HP Analysis. This decision stems from the fact that HP Analysis is a large tool with thousands of features and visualization options and it would be infeasible to implement all of them. The scope was therefore narrowed to show that all of them *can* be implemented.

1.5 Resources

During this thesis work the authors were aided by Jacek Goralski who acted as the supervisor on behalf of IKEA. The authors also received expertise from other IKEA employees. The authors were each given a rental laptop with the prerequisite software needed to complete the project, along with the necessary access cards to navigate the office. While the laptops have been an invaluable tool during this thesis work, they were unfortunately severely delayed. They were expected to be delivered by the end of January for the project launch, but due to technical difficulties and an unusually high demand for rental laptops they weren't received until the end of March. While crippling the initial development stage the blow was softened seeing as there was a high volume of material to learn and take in before implementation.

In order to complete the project the authors of this thesis work needed to have access to IKEA's internal network. This network can only be accessed if one is physically present at an IKEA office, or via a VPN service from home. The received laptops came equipped with a VPN software, as well as the appropriate security credentials to work within the IKEA network.

2 Technical Background

This section aims to provide the technical insight required to better understand this thesis work. The reader is required to have a basic understanding of software development, Java and XML code.

2.1 Application Lifecycle Management (ALM)

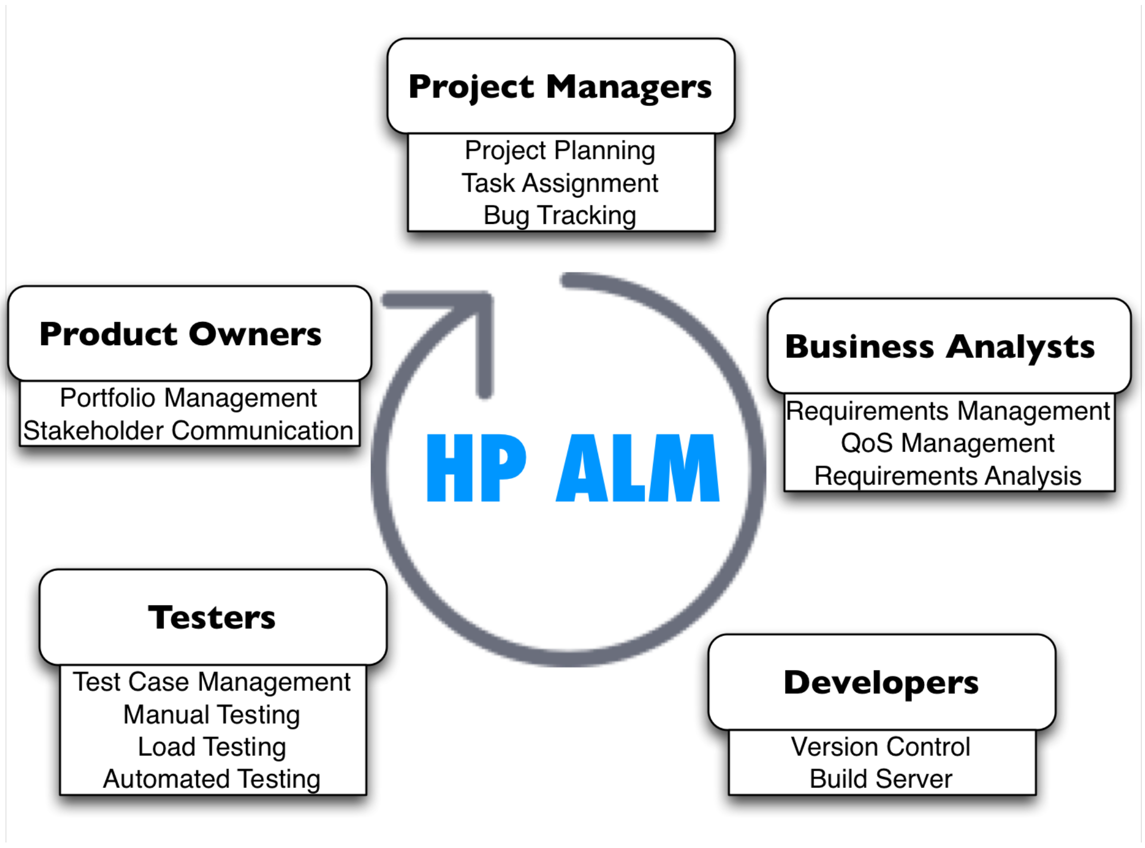


Figure 6. Tools and possibilities provided by the HP ALM Suite [32].

The ALM suite [7] was originally developed by Mercury Interactive in 2006 and later bought up by Hewlett Packard, and is used by IKEA IT as their primary software testing tool. It is one of the most widely used test environments in the world today. HP ALM provides tools for managing application development and application testing, see figure 6. It can be used throughout a project where project managers can initiate all necessary parts with project planning and task assignments. Business analysts can specify the project requirements, developers can share code through version control, testers can perform load- and software tests on the implemented application and lastly product owners can manage portfolios and communicate with stakeholders.

Performance Center, which is the tool for running load tests, integrates with the ALM suite. IKEA IT primarily uses six parts of the ALM suite for performance testing; Virtual User Generator, Performance Center, LoadRunner, Analysis and SiteScope, with the components being displayed in figure 7.

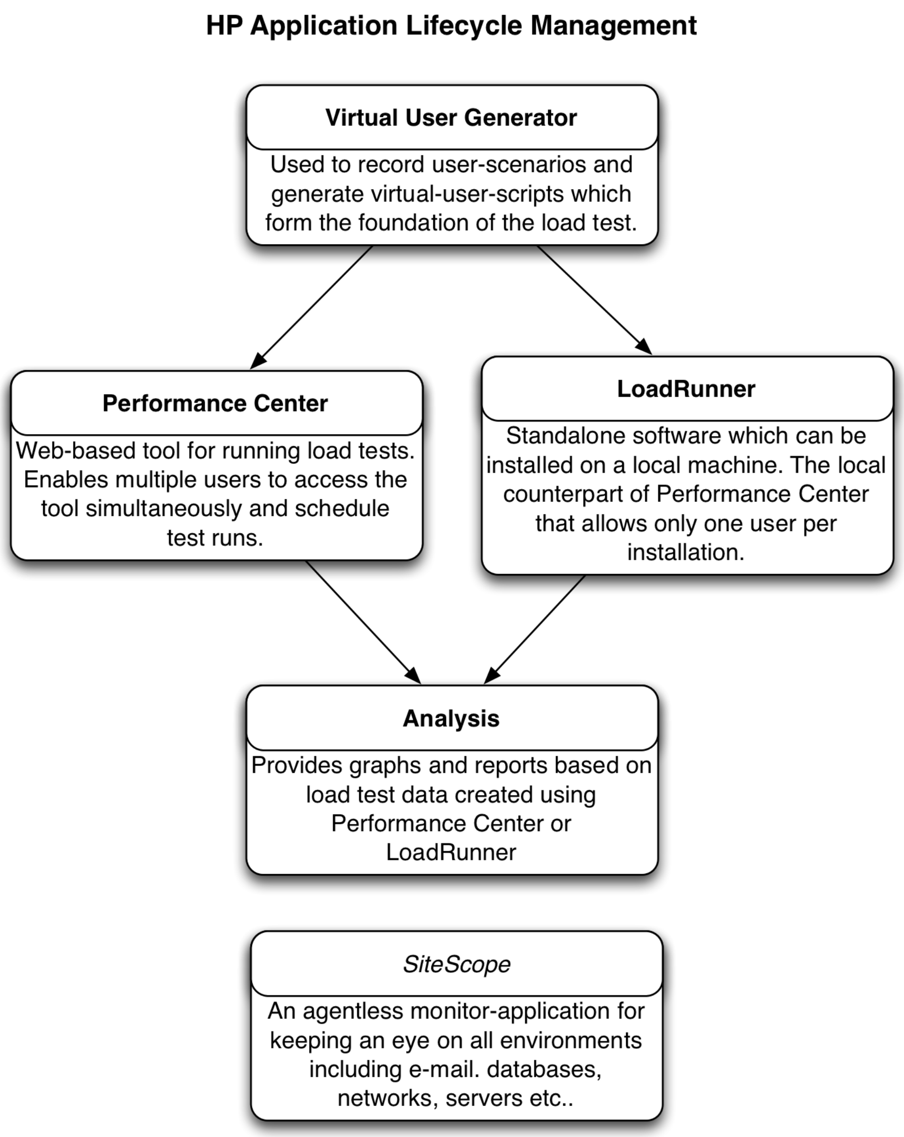
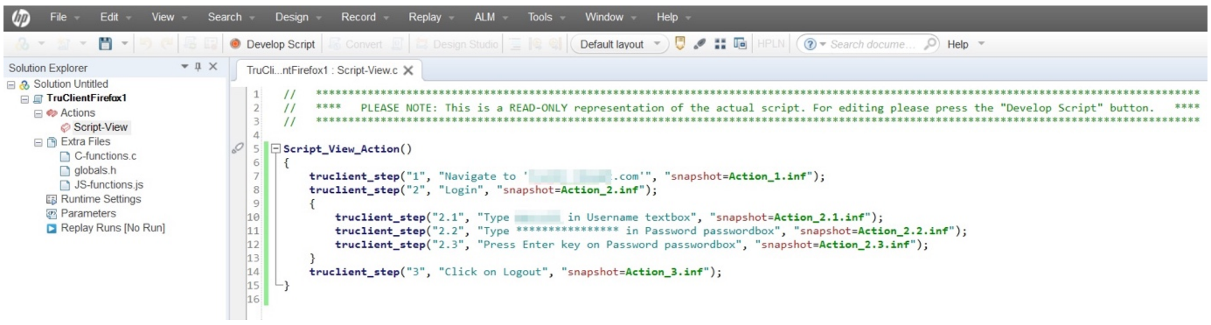

HP Application Lifecycle Management
Virtual User Generator
Used to record user-scenarios and generate virtual-user-scripts which form the foundation of the load test.
Performance Center
Web-based tool for running load tests. Enables multiple users to access the tool simultaneously and schedule test runs.
LoadRunner
Standalone software which can be installed on a local machine. The local counterpart of Performance Center that allows only one user per installation.
Analysis
Provides graphs and reports based on load test data created using Performance Center or LoadRunner
SiteScope
An agentless monitor-application for keeping an eye on all environments including e-mail. databases, networks, servers etc..

Figure 7. Components of the HP Application Lifecycle Management used at IKEA IT.

2.1.1 Virtual User Generator

In Virtual User Generator the tester develops scripts using the programming language C [8] to simulate the actions of real users. The action of real users can vary greatly within the same project. The goal of a script is to simulate a specific scenario, checking that the software fits the requirements for it to be put into production. The simplest way for a tester to create a script is to use the built-in recording tool and record their actions by simply performing some of the possible scenarios, for example navigating to a homepage, browsing a product, adding to cart and finally checking out, see figure 8 for an example of a recorded script. Each script represents a scenario and once all necessary scripts have been created and uploaded to a load generator, each script can run hundreds, thousands or any number of times to represent users performing the specified actions [9]. To run the scripts and simulate a load on the system being tested, IKEA has a set of virtual servers acting as load generators, which are allocated dynamically when needed [1]. These can be scheduled for use inside the Performance Center GUI.



```
//   ***************************************************************************************************************
//   ****   PLEASE NOTE: This is a READ-ONLY representation of the actual script. For editing please press the "Develop Script" button.   ****
//   ***************************************************************************************************************

Script_View_Action()
{
    truclient_step("1", "Navigate to '[redacted].com'", "snapshot=Action_1.inf");
    truclient_step("2", "Login", "snapshot=Action_2.inf");
    {
        truclient_step("2.1", "Type [redacted] in Username textbox", "snapshot=Action_2.1.inf");
        truclient_step("2.2", "Type *************** in Password passwordbox", "snapshot=Action_2.2.inf");
        truclient_step("2.3", "Press Enter key on Password passwordbox", "snapshot=Action_2.3.inf");
    }
    truclient_step("3", "Click on Logout", "snapshot=Action_3.inf");
}
```

Figure 8. A recorded script of navigating to a web-page, signing in with valid credentials and lastly signing out.

2.1.2 Performance Center/LoadRunner

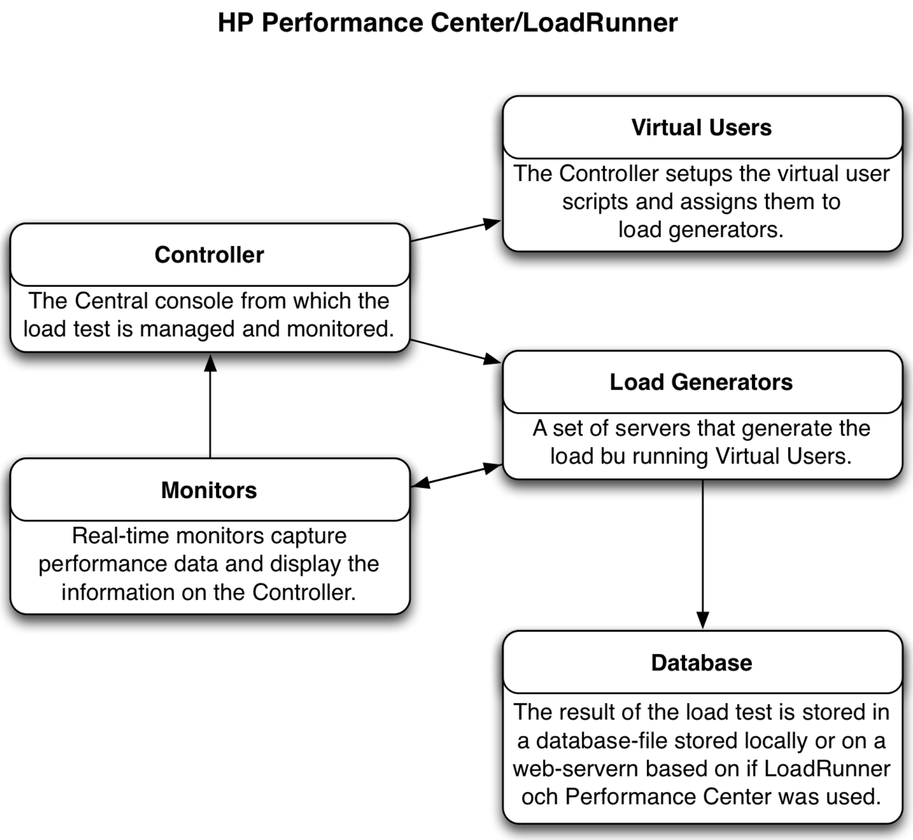

HP Performance Center/LoadRunner
Virtual Users
The Controller setups the virtual user scripts and assigns them to load generators.
Controller
The Central console from which the load test is managed and monitored.
Load Generators
A set of servers that generate the load bu running Virtual Users.
Monitors
Real-time monitors capture performance data and display the information on the Controller.
Database
The result of the load test is stored in a database-file stored locally or on a web-servern based on if LoadRunner och Performance Center was used.

Figure 9. An overview of the interactions between the LoadRunner components.

Performance Centers main functionality is to schedule and run test runs. Test specialists choose a Controller unit when scheduling a test run which produces several different types of test results, such as an output log, raw results, analyzed results, a rich report as well as an HTML report. The analyzed results contain the data used by the Analysis Tool. This is the same data which will be extracted by LoadSplunker for analysis in Splunk.

Performance Center and LoadRunner are basically the same tool where LoadRunner is a standalone software which can be installed on a local machine running Windows and Performance Center is the web version of LoadRunner where multiple users can access the GUI and schedule test runs simultaneously [10].

The first step in running a load test using Performance Center/LoadRunner after Virtual Users have been created, is to set up the Controller. The scripts created using the Virtual User Generator can be uploaded into the Controller upon which the Controller assigns a set of Load Generators to run the virtual user scripts on [11]. The Controller reserves the needed number of Load Generators based on the number of Virtual Users to be executed. At IKEA there are a set of dynamically allocated servers dedicated to serving as Load Generators which can be reserved and scheduled to run by the test specialists.

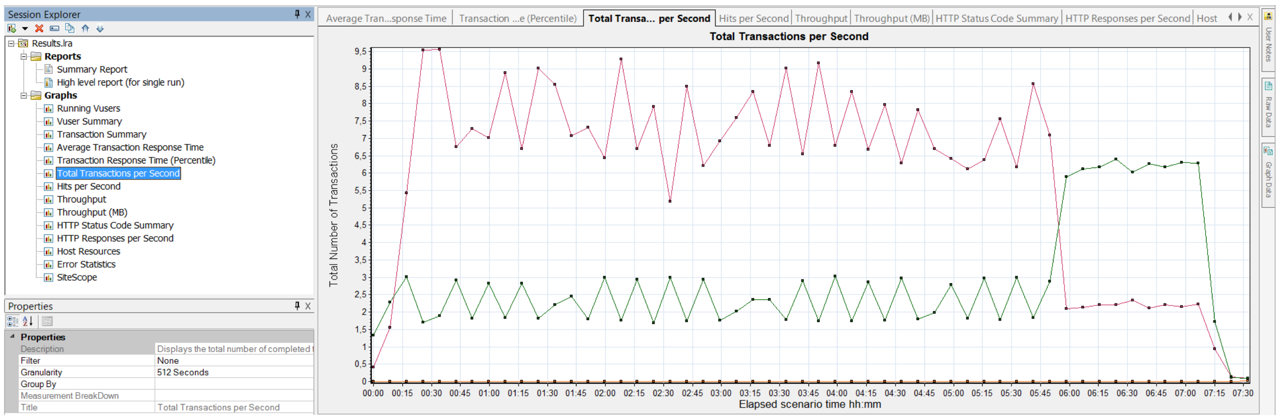
During the execution of the test runs started in Performance Center/LoadRunner, there are a number of real-time monitors that capture performance data from all tiers, servers and network resources (CPU usage, memory utilization, network traffic and other statistics are measured and gathered) and sends the information to the Controller. The monitors make sure that everything runs as planned and keeps constant logs of the progress throughout the test run.

When all Virtual User scripts have been executed, the test run is considered to be done. The data is collected, structured and stored in a database-file. This file is either accessible from a web server if Performance center was used to run the test, or locally if LoadRunner was used.

After the test run is finished the results can be opened in HP Analysis for an analysis of the test result, see figure 9 for an overview of HP Performance Center's components.

2.1.3 HP Analysis

Test specialists use HP Analysis to analyze the test results visually. The tool allows test specislists to break down and visualize test data for their own sake, as well as present and explain it to stakeholders. The visualization options consist of multiple types of graphs and charts [12]. The Analysis tool reads and analyzes the test result file created by Performance Center/LoadRunner. Once the operation completes a session explorer window will open; here you can see information about the scenarios and their duration. Analysis tool will also generate a set of default graphs after each test run; *Running Vusers, Hits per second, Throughput, Transaction Summary, Average Transaction Response Time*, with the possibility of configuring and specifying additional graphs, see figure 10 for a configured graph over *Total Transactions per Second*.


Session Explorer

- Results.lra
  - **Reports**
    - Summary Report
    - High level report (for single run)
  - **Graphs**
    - Running Vusers
    - Vuser Summary
    - Transaction Summary
    - Average Transaction Response Time
    - Transaction Response Time (Percentile)
    - Total Transactions per Second
    - Hits per Second
    - Throughput
    - Throughput (MB)
    - HTTP Status Code Summary
    - HTTP Responses per Second
    - Host Resources
    - Error Statistics
    - SiteScope

Properties

Properties	
Description	Displays the total number of completed
Filter	None
Granularity	512 Seconds
Group By	
Measurement BreakDown	
Title	Total Transactions per Second

Average Tran...sponse Time
Transaction ...e (Percentile)
Total Transa... per Second
Hits per Second
Throughput
Throughput (MB)
HTTP Status Code Summary
HTTP Responses per Second
Host
Total Transactions per Second
Total Number of Transactions
Elapsed scenario time hh:mm
User Notes
Raw Data
Graph Data

Figure 10. Example of a graph generated by HP Analysis.

2.1.4 HP SiteScope

SiteScope is an agentless overall monitor that resides in the background and keeps track of all the environments connected to Performance Center. It collects system utilisation data from application, database and system utilities. The data can be reviewed by users in form of graphs and reports. SiteScope monitors all necessary resources at all time and also notifies if any unusual behaviour is detected [13].

2.2 Linux

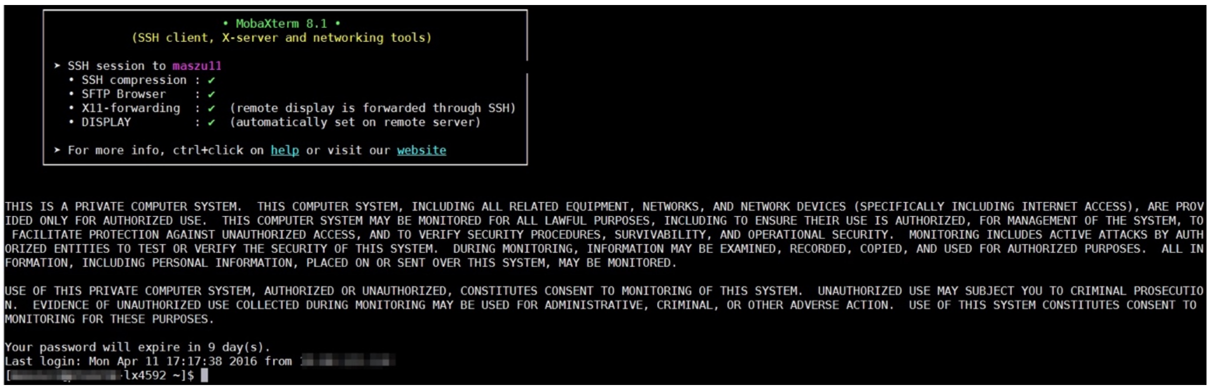
Though the majority of development took place in Windows, the end system for LoadSplunker is Linux, since the Splunk installation at IKEA runs on Linux [1]. Splunk for Linux was downloaded, extracted and installed on the Linux server assigned for this project, via the Bash shell. The authors were given a crash course in the Linux and the Bash syntax by IKEA supervisor Jacek Goralski who oversaw the installation of Splunk and LoadRunner to one of IKEA's Linux servers.

The operating system used on IKEA's servers is Red Hat Enterprise Linux (RHEL) which is a distribution developed by Red Hat and has the commercial market as its target. Like most Linux distributions it is based on Unix and includes the Bash Shell command tool which gives the user full control over the operating system. The reason Red Hat is used at IKEA is because of its high reliability, stability and security. Red Hat is widely used by companies across the world and has become an ecosystem with more than 4000 product certifications and a huge community of developers. The Enterprise version of Red Hat targeted at companies also includes an easily accessible customer support with high technical expertise [14].

Since Red Hat Linux is a subscription based operating system with an expensive yearly cost, it was not an OS that the authors could install on their private computers for testing. Instead a similar but free OS called *CentOS Linux* was downloaded and installed for educational purposes. The purpose was initially to get familiarized with the Linux environment, but more importantly to learn how to use the Bash Shell commands.

Bash is a Unix Shell, the most common one used in the Linux distributions. A Unix Shell is an interface which lets its users type in text commands to control and manipulate the OS's behaviour. The Shell issues a prompt which indicates its readiness to acquire a new commands. The user can enter commands using either text or running a script upon which the shell can interpret and execute them [15].

It was essential for the authors to learn how to use the Bash command tool since the Linux server on which Splunk was installed had to be accessed remotely and had no graphical user interface.
To connect to the Linux server a program called MobaXterm 8.1 was used, which is an extended terminal/command tool for Windows that has tools for X11 server, SSH client and other network features [16], see figure 11.



```
                         • MobaXterm 8.1 •
             (SSH client, X-server and networking tools)

  ➤ SSH session to maszu11
    • SSH compression : ✔
    • SFTP Browser    : ✔
    • X11-forwarding  : ✔  (remote display is forwarded through SSH)
    • DISPLAY         : ✔  (automatically set on remote server)

  ➤ For more info, ctrl+click on help or visit our website


THIS IS A PRIVATE COMPUTER SYSTEM.  THIS COMPUTER SYSTEM, INCLUDING ALL RELATED EQUIPMENT, NETWORKS, AND NETWORK DEVICES (SPECIFICALLY INCLUDING INTERNET ACCESS), ARE PROV
IDED ONLY FOR AUTHORIZED USE.  THIS COMPUTER SYSTEM MAY BE MONITORED FOR ALL LAWFUL PURPOSES, INCLUDING TO ENSURE THEIR USE IS AUTHORIZED, FOR MANAGEMENT OF THE SYSTEM, TO
 FACILITATE PROTECTION AGAINST UNAUTHORIZED ACCESS, AND TO VERIFY SECURITY PROCEDURES, SURVIVABILITY, AND OPERATIONAL SECURITY.  MONITORING INCLUDES ACTIVE ATTACKS BY AUTH
ORIZED ENTITIES TO TEST OR VERIFY THE SECURITY OF THIS SYSTEM.  DURING MONITORING, INFORMATION MAY BE EXAMINED, RECORDED, COPIED, AND USED FOR AUTHORIZED PURPOSES.  ALL IN
FORMATION, INCLUDING PERSONAL INFORMATION, PLACED ON OR SENT OVER THIS SYSTEM, MAY BE MONITORED.

USE OF THIS PRIVATE COMPUTER SYSTEM, AUTHORIZED OR UNAUTHORIZED, CONSTITUTES CONSENT TO MONITORING OF THIS SYSTEM.  UNAUTHORIZED USE MAY SUBJECT YOU TO CRIMINAL PROSECUTIO
N.  EVIDENCE OF UNAUTHORIZED USE COLLECTED DURING MONITORING MAY BE USED FOR ADMINISTRATIVE, CRIMINAL, OR OTHER ADVERSE ACTION.  USE OF THIS SYSTEM CONSTITUTES CONSENT TO
MONITORING FOR THESE PURPOSES.

Your password will expire in 9 day(s).
Last login: Mon Apr 11 17:17:38 2016 from [redacted]
[[redacted]lx4592 ~]$
```

Figure 11. Running bash commands on the Linux server using MobaXterm 8.1.

2.3 Splunk

Splunk is the intended replacement environment for HP's Analysis Tool. It boasts of dynamic graphs, meaning, they are fully zoomable with the user having a wide array of tools and options to choose from while viewing graphs and panels, see figures 12 and 13 for an example of a graph being zoomed in on. As an example, if a test manager is viewing a chart with data from the last 24 hours, but becomes curious about data from the previous week, this can be configured with a simple drop down menu instead of having to contact a test specialist and potentially re-run an entire test analysis. Splunk also has intuitive drill-down functionality and a test specialist will be able to configure specific drill-down functions. An example of drill-down functionality in Splunk is if a user clicks on a chart containing the number of active system users, Splunk will intuitively extract the relevant fields for active users and list the details in a new statistics panel on the fly. "Lookups" are another highly usable and resource managing function. A system heavy search, such as mapping every payment transaction in the last 24 hours, while important, can be a tremendous waste of system resources to run every time a stakeholder wants to review the data. Splunk can instead be configured to run a scheduled "lookup" with a chosen interval, for example once a day, that saves this data in the Splunk database for review at any time [17]. Splunk will then display the data from the lookup when called, instead of running a fresh search. This results in the panel being displayed significantly faster while at the same time saving system resources.

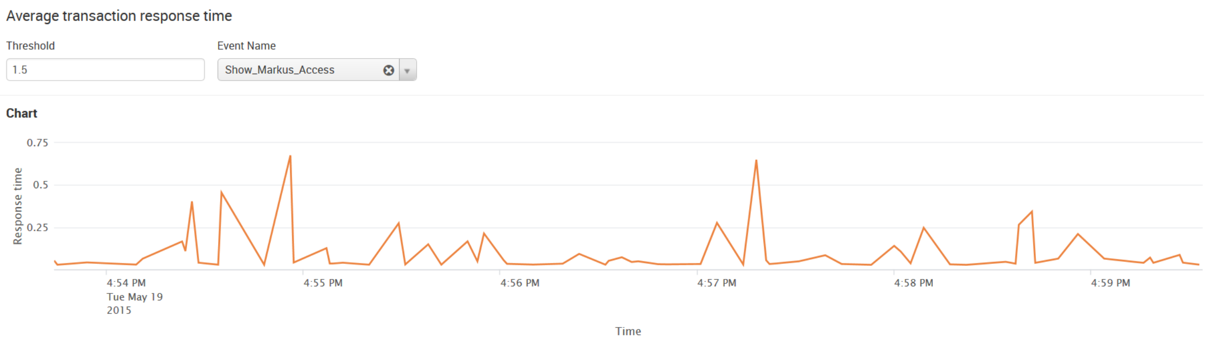


Figure 12. A graph in Splunk displaying transaction response time for a test run.

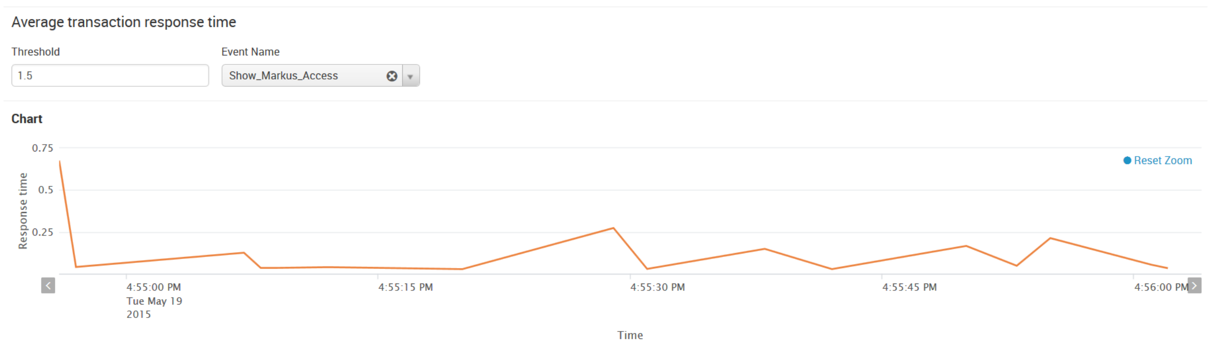


Figure 13. Figure 12 zoomed in.

2.3.1 Dashboards in Splunk

A Splunk dashboard consists of any number of panels defined by its creator. These panels range from a simple statistical panel with the transaction information of every system user in the last hour to an interactive pie chart with the server response times from the latest test run and anything in between. The individual panels can be set to perform a fresh search and load the latest data every time it is displayed. It can also be set to simply load them from a predefined and performed lookup [18], see figures 14 and 15.

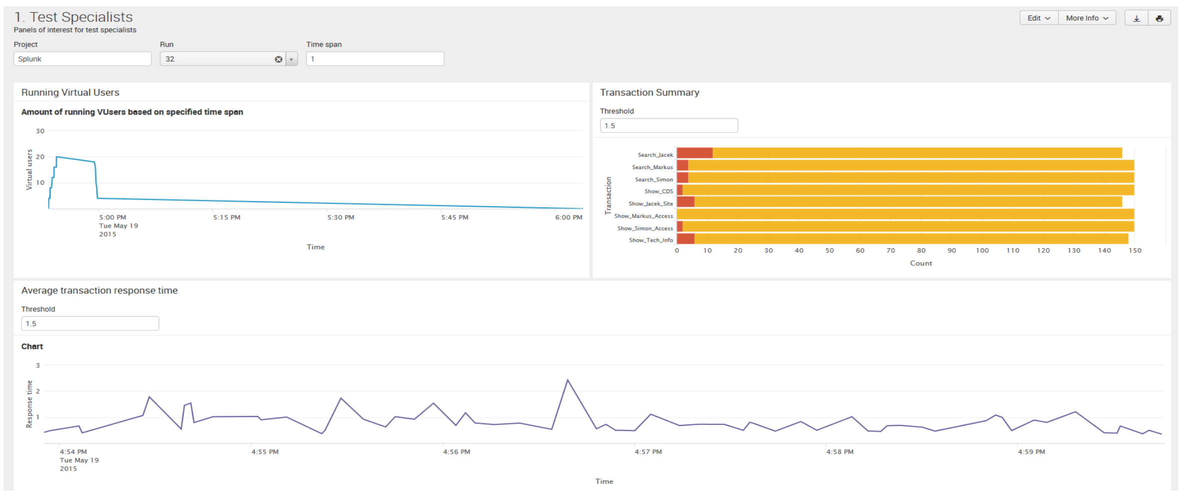


Figure 14. Extract from a Dashboard created in Splunk

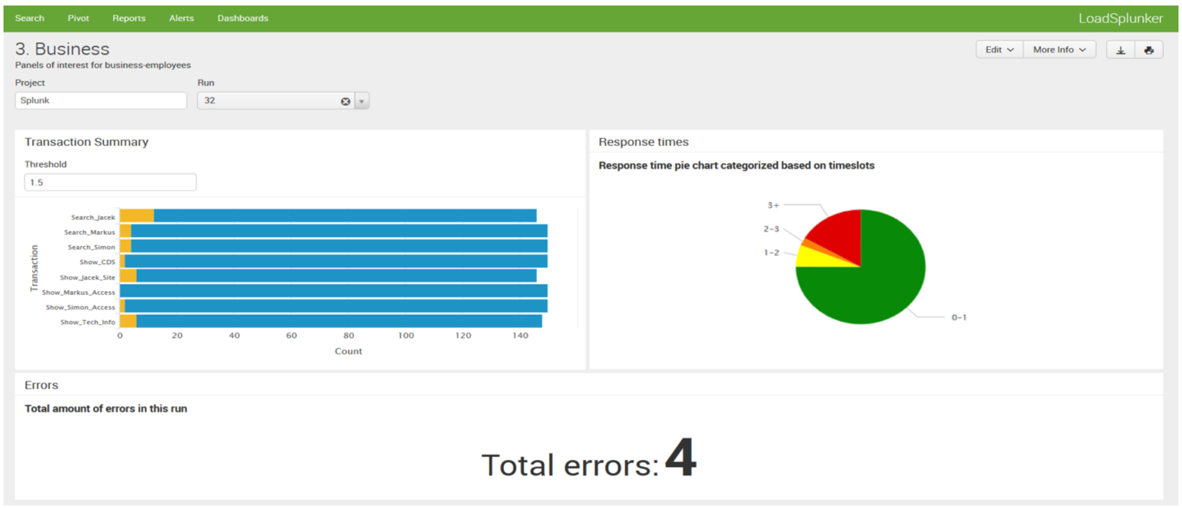


Figure 15. Extract from a Dashboard created in Splunk

2.4 LoadSplunker

LoadSplunker is the solution implemented by previous students at Lund university. Their aim was to create an integration between HP Performance Center and Splunk. The result was a real-time based Java-application that in the background continuously searches the HP Performance Center server at IKEA for new test results. If a test result is found, the raw test data is extracted from the Performance Center server and converted into human readable XML-files. LoadSplunker stores the XML-files in structured folders based on project and run ID's (information extracted from the test data). This folder structure becomes the source path which is used in Splunk to find the specified data.
Upon successful conversion of the test data into XML structure, the XML-files are automatically uploaded to the Splunk server with the help of a Splunk-forwarder, where searches can be run and panels/dashboards can be built based on the uploaded data.

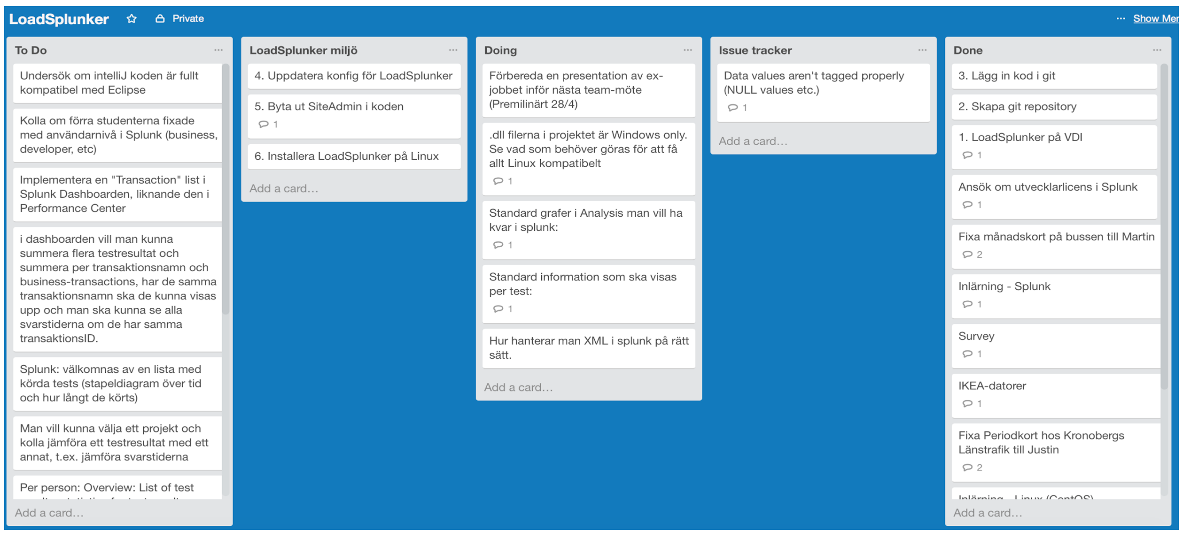
LoadSplunker uses two API's developed by Hewlett Packard, a REST API and a SiteAdmin API. The REST API is used to send search-queries into the Performance Center database to retrieve the raw test data. This is the data that later is converted into XML-structure.
The SiteAdmin API is used to monitor the Performance Center database and continuously check for new test results. As soon as a new test run has finished, the SiteAdmin API triggers a flag and LoadSplunker retrieves the data using the REST API. Another used feature of the SiteAdmin API is that of retrieving user privileges and user-account information from the Performance Center database which was meant for implementing a role-based system into LoadSplunker, unfortunately the previous students did not find time to complete this part of their thesis work [4].

3 Method

This section will cover the methods and tools used during this thesis work.

3.1 Kanban

Kanban [19] was used as the software development process, other methodologies were considered, but seeing as the development team only consisted of two people with varying work schedules, other alternatives were discarded due to being infeasible.
Working agile and lean is an improvement from the confined and restrictive models that the authors have experienced in the past. The focus has been on development and maintaining good communication. An agile methodology has worked well to facilitate this, especially considering the constantly changing circumstances and requirements of the assignment at hand.
A digital Kanban board was set up using the Trello online tool [20] with a standard three column solution consisting of *To do, Doing, Done*. This was quickly expanded with an additional two columns, a separate *To do* column specifically for LoadSplunker and an *Issue Tracker* column, see figure 16.
The work to be done was visualised and planned initially so the *to do* list could be filled. The authors then decided to limit the work in progress (WIP) to a maximum of five tasks being worked on simultaneously (two per person and one cooperatively, for example). A WIP-limit of five was evaluated to be sufficient since it would keep work flow high without too many items being worked on simultaneously, and without limiting the workflow to too few items, letting the authors start on another task in case one was stalled or delayed. The one significant part of Kanban that was modified for this assignment was the lead time. Since this thesis work was not the authors only occupation and had to be balanced with other studies and work. Studying for exams and planning around work as well as adapting to IKEA's schedule has made the specification of a lead time unfeasible. However the authors strived to maintain a steady delivery time of no more than 2 weeks per task.



To Do
- Undersök om intelliJ koden är fullt kompatibel med Eclipse
- Kolla om förra studenterna fixade med användarnivå i Splunk (business, developer, etc)
- Implementera en "Transaction" list i Splunk Dashboarden, liknande den i Performance Center
- i dashboarden vill man kunna summera flera testresultat och summera per transaktionsnamn och business-transactions, har de samma transaktionsnamn ska de kunna visas upp och man ska kunna se alla svarstiderna om de har samma transaktionsID.
- Splunk: välkomnas av en lista med körda tests (stapeldiagram över tid och hur långt de körts)
- Man vill kunna välja ett projekt och kolla jämföra ett testresultat med ett annat, t.ex. jämföra svarstiderna
- Per person: Overview: List of test

Add a card…

LoadSplunker miljö
- 4. Uppdatera konfig för LoadSplunker
- 5. Byta ut SiteAdmin i koden 💬 1
- 6. Installera LoadSplunker på Linux

Add a card…

Doing
- Förbereda en presentation av ex-jobbet inför nästa team-möte (Premilinärt 28/4)
- .dll filerna i projektet är Windows only. Se vad som behöver göras för att få allt Linux kompatibelt 💬 1
- Standard grafer i Analysis man vill ha kvar i splunk: 💬 1
- Standard information som ska visas per test: 💬 1
- Hur hanterar man XML i splunk på rätt sätt.

Add a card…

Issue tracker
- Data values aren't tagged properly (NULL values etc.) 💬 1

Add a card…

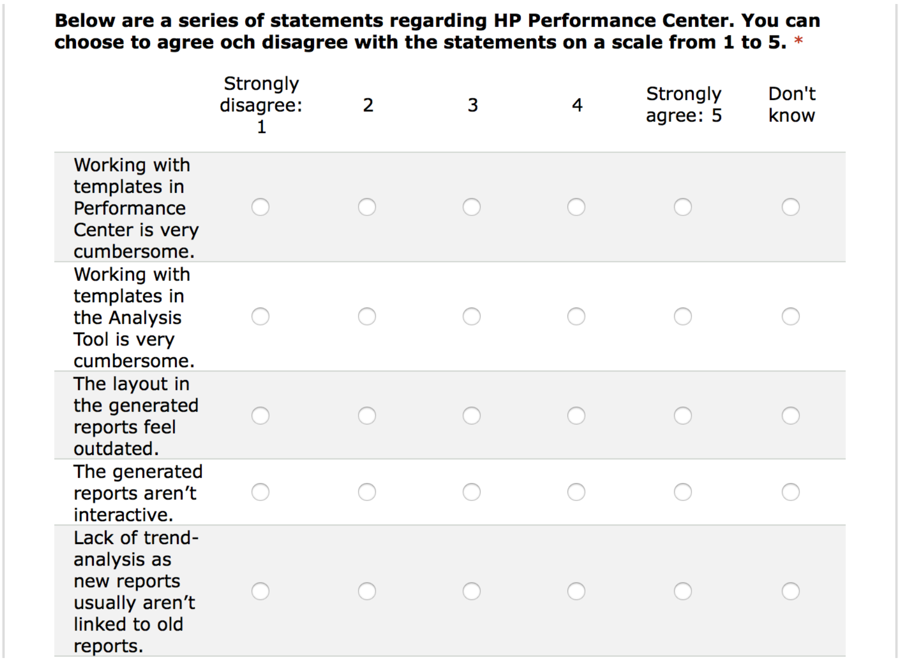
Done
- 3. Lägg in kod i git
- 2. Skapa git repository
- 1. LoadSplunker på VDI 💬 1
- Ansök om utvecklarlicens i Splunk 💬 1
- Fixa månadskort på bussen till Martin 💬 2
- Inlärning - Splunk 💬 1
- Survey 💬 1
- IKEA-datorer 💬 1
- Fixa Periodkort hos Kronobergs Länstrafik till Justin 💬 2
- Inlärning - Linux (CentOS)

Add a card…

Figure 16. A snapshot of the Trello board used during development

3.2 Survey

In order to further improve the software it was deemed essential to gain an overview of the current shortcomings of the HP Analysis Tool and better understand what changes needed to be made in order to improve the serviceability of the tool at the workplace. Therefore, a survey was conducted using Google Forms [21] to gain insight into employees current perception of the ALM suite and the LoadRunner Analysis Tool, see figure 17. To help formulate questions, interviews were conducted with various stakeholders about what they perceived to be the pros and cons of the Analysis tool. The interviewed stakeholders had a very negative view of the Analysis Tool which made it very difficult to gain objective criticism or opinions about the software. The survey was therefore written with these opinions formulated as statements about the software that respondents can agree with on a scale ranging from strongly disagree to strongly agree. These statements were based on what was gathered from the aforementioned interviews.



Below are a series of statements regarding HP Performance Center. You can choose to agree och disagree with the statements on a scale from 1 to 5. *

	Strongly disagree: 1	2	3	4	Strongly agree: 5	Don't know
Working with templates in Performance Center is very cumbersome.	○	○	○	○	○	○
Working with templates in the Analysis Tool is very cumbersome.	○	○	○	○	○	○
The layout in the generated reports feel outdated.	○	○	○	○	○	○
The generated reports aren't interactive.	○	○	○	○	○	○
Lack of trend-analysis as new reports usually aren't linked to old reports.	○	○	○	○	○	○

Figure 17. Extract from the survey. Complete survey can be found under Appendix.

3.3 Performance Center

The features and possibilities of Performance Center were demonstrated by the supervisor at IKEA during several meetings. The authors were allowed to participate in real world scheduling and deployment of test runs as well as analyzing them in the Analysis Tool. This gave the authors experience in using these tools which aided the development of the Splunk dashboards, based on what parts of Performance Center and the analysis tool that were perceived as good or bad.

3.4 Splunk

The authors were fortunate enough to have the opportunity of consulting with IKEA's resident Splunk expert Magnus Johansson during several meetings. Mr. Johanson demonstrated the use of the Splunk syntax as well as sharing his expertise and good sources of knowledge, such as the answers.splunk.com website as well as several tutorials and lectures.

The authors also attended an online Splunk e-learning introduction course offered on the official Splunk website. The aim of this course is to introduce new users to the Splunk syntax and cover the basics. This course consisted of virtual lectures by Splunk experts as well as quizzes, see figures 18, 19 and 20.

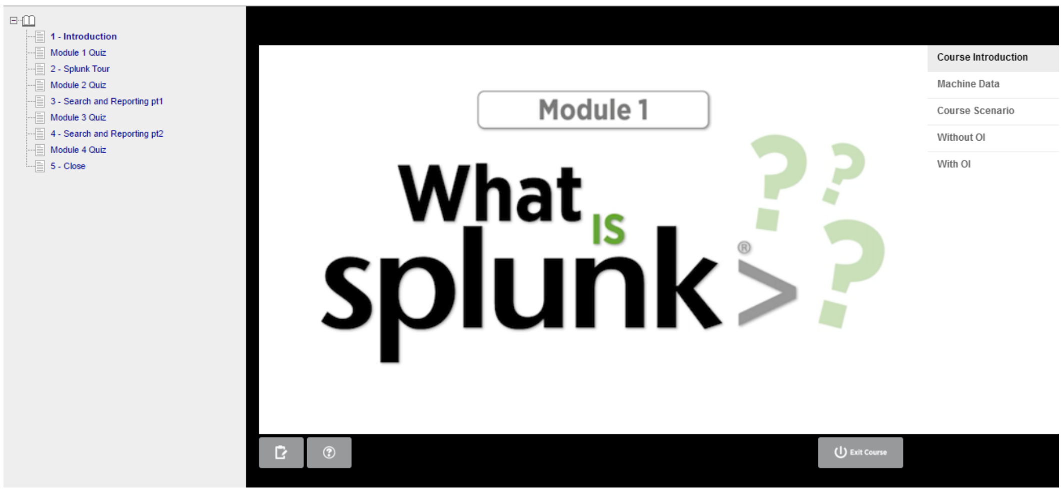
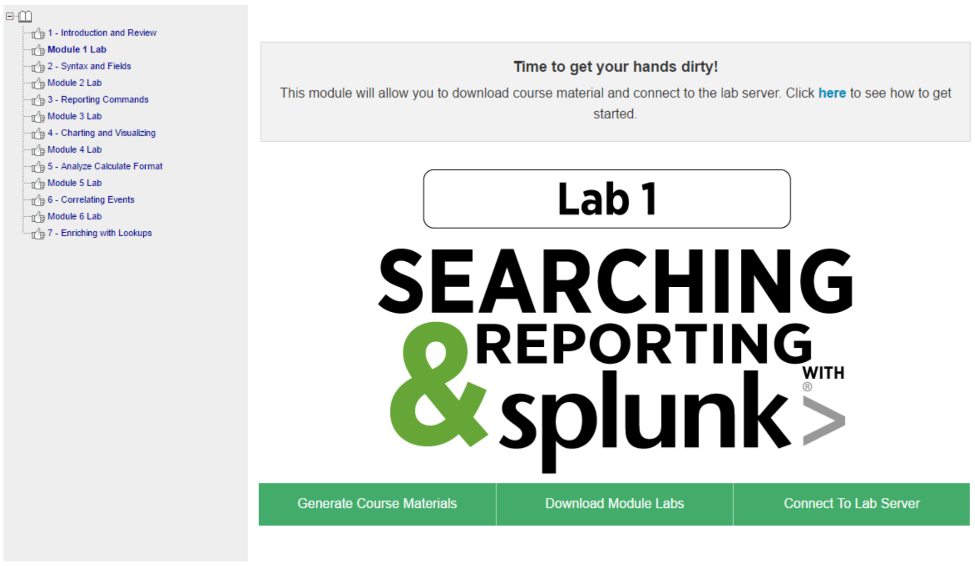


Figure 18. Splunk introduction course.

Upon completion, the authors were allowed to attend an in-depth course called "Searching And Reporting With Splunk", at IKEA's expense [22]. This paid-for course is aimed at intermediate users and is quite comprehensive. Just like in the introduction course the searching & reporting course consists of virtual lectures and quizzes, but this module also includes a live lab environment. The lab environment consists of realistic data from a mock-up global company. This course proved vital for the success of this thesis work as the test data proved to be quite similar to the test data extracted from IKEA.

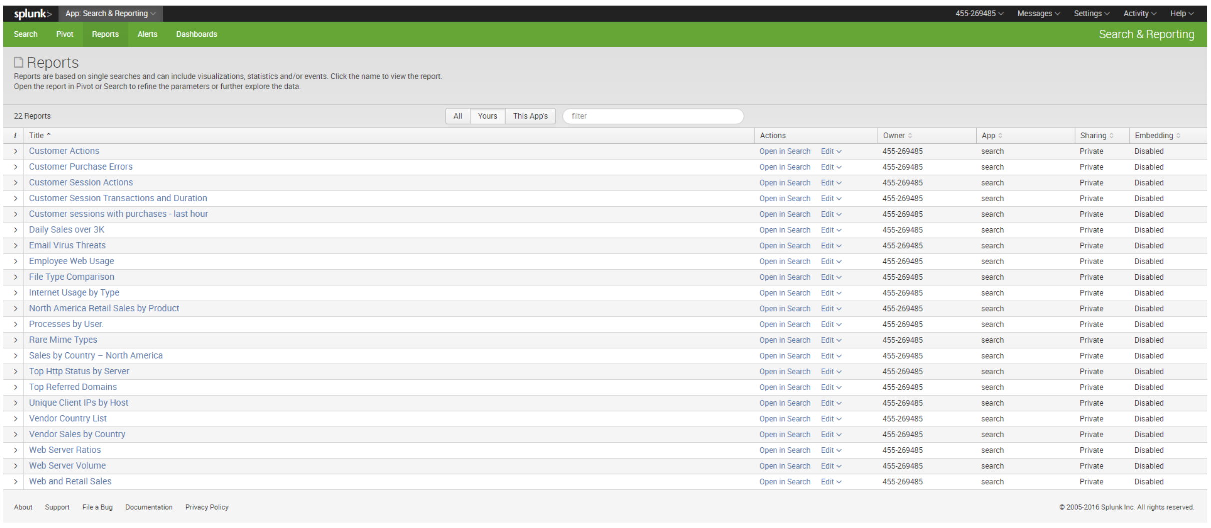
The two courses gave the authors an essential knowledge about searching in Splunk, how to build dashboards and its overall usage. This knowledge was later applied to formulate the search queries needed for extracting data from the XML files and enabled the authors to design and create dashboard panels to present the data in. The Splunk courses also educated the authors on adding, indexing, extracting and modifying data.



Time to get your hands dirty!

This module will allow you to download course material and connect to the lab server. Click here to see how to get started.

Figure 19. Intermediate Splunk Course.


Reports

Reports are based on single searches and can include visualizations, statistics and/or events. Click the name to view the report.
Open the report in Pivot or Search to refine the parameters or further explore the data.

22 Reports

i	Title ^	Actions	Owner	App	Sharing	Embedding
>	Customer Actions	Open in Search Edit	455-269485	search	Private	Disabled
>	Customer Purchase Errors	Open in Search Edit	455-269485	search	Private	Disabled
>	Customer Session Actions	Open in Search Edit	455-269485	search	Private	Disabled
>	Customer Session Transactions and Duration	Open in Search Edit	455-269485	search	Private	Disabled
>	Customer sessions with purchases - last hour	Open in Search Edit	455-269485	search	Private	Disabled
>	Daily Sales over 3K	Open in Search Edit	455-269485	search	Private	Disabled
>	Email Virus Threats	Open in Search Edit	455-269485	search	Private	Disabled
>	Employee Web Usage	Open in Search Edit	455-269485	search	Private	Disabled
>	File Type Comparison	Open in Search Edit	455-269485	search	Private	Disabled
>	Internet Usage by Type	Open in Search Edit	455-269485	search	Private	Disabled
>	North America Retail Sales by Product	Open in Search Edit	455-269485	search	Private	Disabled
>	Processes by User.	Open in Search Edit	455-269485	search	Private	Disabled
>	Rare Mime Types	Open in Search Edit	455-269485	search	Private	Disabled
>	Sales by Country – North America	Open in Search Edit	455-269485	search	Private	Disabled
>	Top Http Status by Server	Open in Search Edit	455-269485	search	Private	Disabled
>	Top Referred Domains	Open in Search Edit	455-269485	search	Private	Disabled
>	Unique Client IPs by Host	Open in Search Edit	455-269485	search	Private	Disabled
>	Vendor Country List	Open in Search Edit	455-269485	search	Private	Disabled
>	Vendor Sales by Country	Open in Search Edit	455-269485	search	Private	Disabled
>	Web Server Ratios	Open in Search Edit	455-269485	search	Private	Disabled
>	Web Server Volume	Open in Search Edit	455-269485	search	Private	Disabled
>	Web and Retail Sales	Open in Search Edit	455-269485	search	Private	Disabled

© 2005-2016 Splunk Inc. All rights reserved.

Figure 20. Summary reports of some exercises completed in the intermediate course.

3.5 IKEA's Requirements

IKEA's requirements were gathered in various ways, explicitly via consulting IRW (IKEA Retail Website) employees, our IKEA supervisor, as well as conducting a survey with stakeholders from both the business and development sectors of IKEA.
Requirements were also deduced implicitly via the author's own observations and conclusions while working with the IKEA staff.
The gathered requirements became the backbone for the Trello board where they were formulated into tasks and put into the *to do* list, which ultimately served as a backlog controlled by the authors of this thesis assignment. The tasks in the *To Do* list were sorted by priority which was based on interviews conducted with IKEA employees and the results from the survey described in section 3.2.

3.6 Implementation

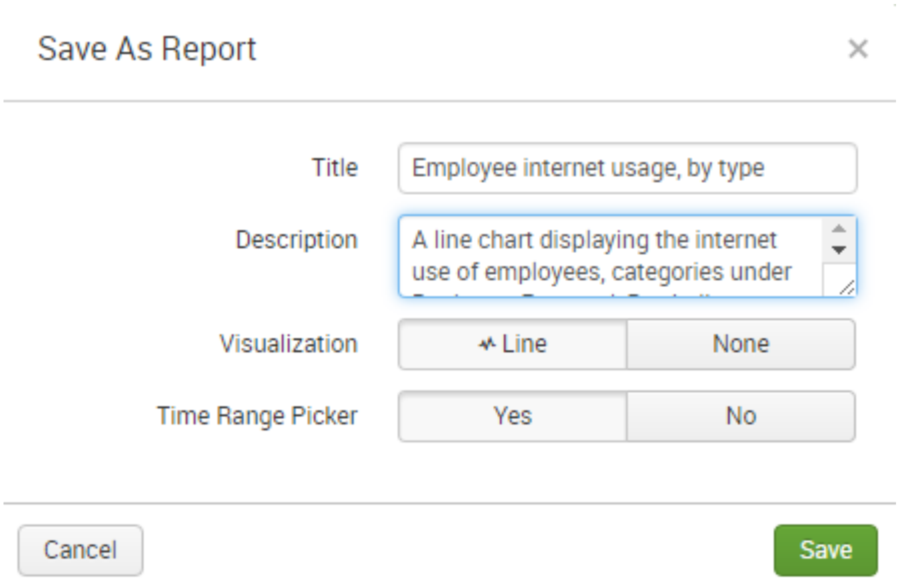
In order to achieve the defined goals, this thesis originally intended to re-use what was produced by the previous project and develop it further. To realize this, the previously created software needed to be reviewed and the current version of *LoadSplunker* had to be installed in order to gain familiarity with its functionality and in order to be able to determine the best way to implement a dashboard for it.

The current version of LoadSplunker is developed to work with ALM 12.01 and Splunk 6.2.0. LoadSplunker works on machines with a Java Virtual Machine installed. Both the ALM suite and Splunk have had several updates since the previous implementation of LoadSplunker. Adaptation to newer software was therefore necessary.

After this was achieved it was necessary to implement and modify the result from the Windows only solution of the previous thesis assignment so that it would fit into IKEA's current systems, which run on Linux Red Hat Enterprise version 7. Target system for LoadSplunker (and Splunk at IKEA) is therefore Linux (Red Hat Enterprise version 7) only. The previous students assignment was developed in- and runs on Windows, which cannot be kept as IKEA's Splunk installation runs on Linux. However, Windows can be used during development.

Secondly, it was desired to allow test reports from LoadSplunker to be easily understandable and viewable in order to ease the test report process. To achieve this it was therefore necessary to automate each step from when a test run is completed to viewable graphs and statistics. These steps include extracting the raw test result data from a test run, then interpret, categorize, structure and finally upload it into Splunk where the data can be manipulated into viewable results in the form of graphs and statistics. The hope was to reduce the amount of manual work currently needed to create a test report.

For Java development the IntelliJ [23] environment was used, seeing as it was the environment of choice for the previous students. Though the majority of the implementation phase was spent within the Splunk environment there was nevertheless a need to edit the previous student's code and solution. GIT [24] was used for version control, with a repository running at an IKEA Linux server.
Several different dashboards and dashboard panels (see figures 21 and 22 for examples) were created in Splunk as a proof-of-concept. The purpose was to show that Splunk can be a comprehensive replacement for the Analysis Tool, without any data loss.


Save As Report

Title	Employee internet usage, by type
Description	A line chart displaying the internet use of employees, categories under
Visualization	Line / None
Time Range Picker	Yes / No

Figure 21. Saving a search as a report.

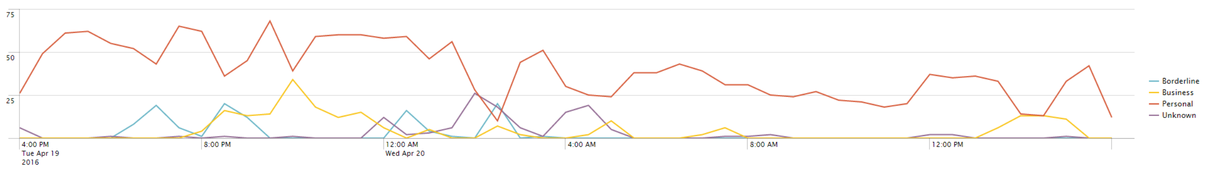


Figure 22. A search appearing as a dashboard panel.

3.7 Testing

During development a separate Splunk server running on Linux was set up at IKEA for testing and implementation. MobaXterm [16] (see chapter 2.2) was used to configure the server and for remote access. This meant the authors had their own Splunk environment to work within, containing real test data from Performance Center. One of the benefits of the Splunk syntax is the instant feedback it provides. If a search term doesn't exist or is entered incorrectly no results will be returned and a dialogue box detailing the error will be shown. As soon as a search is successful and the relevant information extracted the user can choose to save the search as a report, or add it to a dashboard.

3.8 Validation

Each new dashboard change and solution was evaluated and validated by the IKEA supervisor. Upon extracting new information from the converted XML files that could be used for a new dashboard panel, the supervisor at IKEA was contacted and asked if the proposed panel would be of interest and relevant enough to add into a dashboard. If the suggested panel was accepted, the authors begun with the implementation.
Each successfully developed new dashboard panel was then validated by the IKEA supervisor, who would perform minor tests using real test data for the new panel and give constructive feedback.
To display and validate this much information the authors spent one to two days a week working with the supervisor at the IKEA main office in Älmhult, with less critical communication handled via email. This amount of live feedback and guidance has been vital for the successful completion of this thesis work, and wouldn't have been possible if IKEA hadn't approved of the supervisor Jacek Goralski allocating 20% of his working contract for this project.

3.9 Communication

The authors have worked closely with IKEA supervisor Jacek Goralski on site at the IKEA headquarters in Älmhult 1-2 times a week. The rest of the time the authors have developed either together in person at Lund University or from the comfort of their own homes while communicating over Skype [25]. Less critical information has been handled via email or text messages. The workload has been equally divided between the authors of this thesis assignment with a few minor exceptions where for example Szuter's responsibility was slightly shifted towards finalizing all design work, whereas Rees was in charge of overlooking and managing the report writing process.

4 Analysis

4.1 Kanban

A good structure and overview of what has been done and what is left to be done given by the usage of a Trello-board, has been of great help to the authors, where not only the authors but also stakeholders could take part of the development and add their standpoints to the progress.
The Kanban way of being adaptable to changing requirements was essential for this thesis work due to the high initial learning curve. A lot of information, tutorials and laborations had to be ingested by the authors at the beginning of this work which resulted in requirements having to be modified and restructured during the development progress. Using Kanban makes it easy to change priorities during the progress of a project, which was another valuable asset during this thesis work since the authors had to wait for the survey results to be submitted before being able to completely specify the requirement priorities.
Another pro of using the Kanban development model was that it usually results in less defects in the final product, since the developer can return to a previous stage and review it whenever he/she finds it suitable, as opposed to more strict development-models where everything has to be made in exact order. This thesis work relied on the possibility to be able to jump between research, development, review and documentation which was made possible by using Kanban.

However, the Kanban-model required a lot self discipline from the authors. Kanban is a very flexible development model with adjustable rules, which relies on the developers being able to carry the burden of responsibility. The authors of this thesis work had to make sure that communication was maintained between them as well as with the supervisor at IKEA and the supervisor at school. The reason for this was that it was very important for everybody involved to be updated as soon as changes to the requirements or the code were made.
The lack of strict documentation can sometimes also be a negative aspect of the Kanban model since documents specific for requirements, test, evaluation and overall logs can prove to be useful during the final stages of a project. The authors of this thesis nevertheless managed to keep all requirements gathered on the Trello board and all necessary information and meetings were documented in Google Drive.

4.2 Development

Because of the requirement that LoadSplunker should run on Linux only, a lot of unexpected modification to the previous students solution was necessary. This lead to a development delay of this thesis work's main goal which was the creation of Splunk dashboards, and divided the scope of this thesis assignment into two significantly different tasks. The previous students used libraries that were Windows-only which made the execution on Linux impossible. This had to be modified by the authors of this thesis work which proved not to be an easy task and unfortunately quite time consuming. It was essential for the authors to get the previous students LoadSplunker to run before being able to implement the dashboards, since it is LoadSplunker that is responsible for retrieving the data from Performance Center, converting it to XML-structure and forwarding it to Splunk. The plan was to initially run LoadSplunker as it was, on a Windows machine, to not have to stall the development of the dashboards. Unfortunately, was this not possible due to the Performance Center servers having been replaced since year 2015, and the new ones not being compatible with the connection-libraries and API's used in LoadSplunker by the previous students. This lead to the final decision of first modifying LoadSplunker to work in Linux and the new Performance Center-environment. This proved to be very time consuming since about 80% of the LoadSplunker code had to be discarded or modified. All the Windows-only connection-libraries like Com4J and API's like the SiteAdmin API had to be removed. The new solution instead relies on a script that polls the Performance Center database every 20 minutes to check if new test result data has become available since the previous poll time-stamp.

The LoadSplunker modifications took weeks to complete which delayed the development of the Splunk dashboards. To narrow the scope of the dashboards and make the development realistic, the authors had to cut back on some functionalities like cross-comparison between different load tests in one dashboard. However three different dashboards were developed as planned with easy to understand graphs and tables. Drill-down functionality was implemented as planned and even comparison between different test runs was partly implemented. All in all the authors are pleased with the development progress and satisfied with the overall result.

4.3 Documentation of meetings

A meeting journal was created and maintained throughout the course of this thesis work. All meeting with supervisors and resident experts at IKEA were documented and uploaded to Google Drive. This has been a valuable learning tool and a point of reference during development and reporting.

4.4 Trello

The Trello board [20] (see figure 16, section 3.1) used during development has been a valuable tool, giving a clear overview and structure during this phase. Although the authors feel it could have been utilised more and to greater effect, the goals have been clear from the start and working in such a small team there has not been as high of a need for division of tasks as would be expected in larger development teams.

4.5 HP Analysis and Splunk

A large part of this bachelor thesis work was to find a significantly improved replacement for the HP Analysis tool. Since this was a continuation of a previous project, part of this goal was already decided and developed. The chosen strategy of the previous students was to extract the test data from HP Performance Center and forward it to Splunk. Other options were considered, for example using Graylog instead of Splunk based on its better understanding of the XML file structure, or converting the raw test data to JSON (Java Script Object Notation) instead of XML since Splunk is better at handling data presented as JSON-files. However considering that a great amount of work from the previous students had focused on handling and converting the data from Performance Center to XML and then forward it to Splunk, the authors agreed to continue on this path instead of revamping the entire process.
The first step was to design the Splunk dashboards to be used in way that was easy to use but at the same time included all the necessary information. The authors needed insight into the pro's and cons of the HP Analysis tool. To gather relevant information the authors used three methods. The first being overall communication and interviews with the test specialists at IKEA that were frequent users of the Analysis tool. The second method was sending out a survey to different stakeholders consisting of test specialist, developers and some business oriented employees at IKEA. The third and final method for gathering information about the Analysis tool was to download the software and personally test it.

All employees interviewed agreed that the visualization options in the HP Analysis-generated report are aesthetically unappealing and feel outdated. It also had the additional disadvantage of being static images, meaning they aren't zoomable and lack drill-down functionality. A real world example of why such a disadvantage would cause issue in the workplace occurred when an IKEA employee was charting server response times and a stakeholder became curious about the servers individual response times. Since this wasn't specified in the original test run the entire test run had to be rescheduled, wasting hours of manpower and system resources when a simple drill-down could have been performed on a dynamic chart. Cases like this are a common occurrence and display the need for our thesis work.

The survey results described in section 4.7 were of great help when deciding how to design and develop the Splunk dashboards and gave the authors a good understanding of what parts needed to be prioritised.

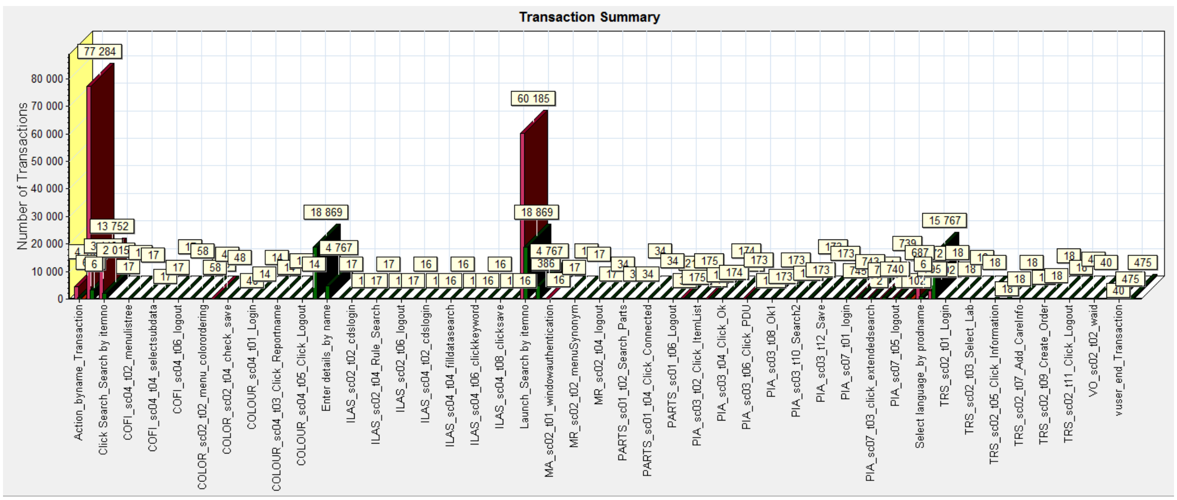


Figure 24. Example of graph generated by HP Analysis.

Personally testing the HP Analysis tool proved to be a worthwhile experience since it gave the authors a much better understanding of what was good with the tool and what needed improvement. The authors were guided by IKEA supervisor Jacek Goralski on how to download and install the HP Analysis software. A finished test-run result was handed to the authors for uploading into the tool. The authors got the possibility to familiarize themselves with the software and to analyse the generated graphs and results. The drawbacks of the Analysis tool were at once evident with its outdated design and aesthetically unappealing graphs. The lack of a proper drill-down functionality made some of the graphs unreadable as there was too much information being displayed which conveyed a congested image in the graphs, see figures 24 and 25.

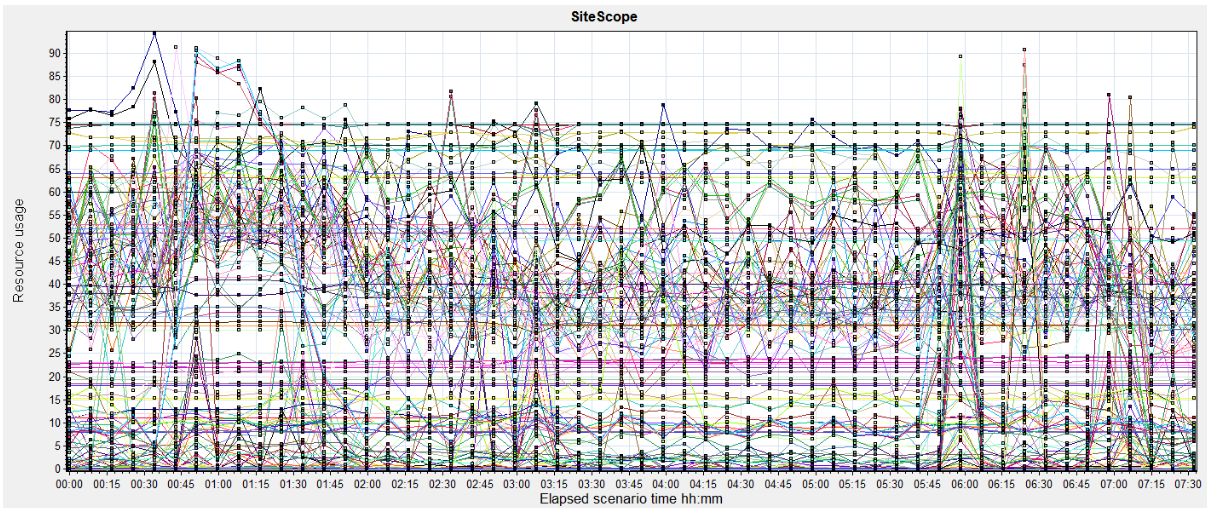


Figure 25. Example of graph generated by HP Analysis.

4.6 Survey

The survey was sent out through Google Forms [21] to 40 varying stakeholders at IKEA. A total of 27 responses were received. The results of the survey gave valuable insight into the direction of the thesis and what areas should be focused on. Answers with the highest percentage were prioritised and focused on during development. The full questionnaire can be found in the appendix.

Question	**Strongly disagree: 1**	**2**	**3**	**4**	**Strongly agree: 5**	**Don't Know**
1	0%	4%	11%	52%	33%	0%
2	4%	7%	7%	56%	26%	0%
3	0%	11%	37%	45%	7%	0%
4	0%	4%	19%	59%	19%	0%
5	4%	4%	34%	19%	41%	0%
6	11%	4%	26%	30%	22%	7%
7	0%	7%	15%	63%	15%	0%
8	0%	11%	19%	48%	11%	11%
9	7%	0%	7%	22%	59%	4%
10	4%	7%	44%	26%	19%	0%
11	0%	11%	22%	32%	32%	11%
12	0%	7%	19%	26%	52%	0%
13	11%	4%	26%	33%	26%	0%
14	0%	11%	11%	33%	30%	15%
15	4%	7%	15%	33%	22%	26%
16	0%	11%	15%	19%	56%	0%

Figure 23. Survey results. The full questionnaire can be found in the appendix.

As can be seen in the survey results most answers lean towards agree and strongly agree. A few strong examples would be questions 9, 12 and 2, see figure 23.

The second statement of the survey, "Working with templates in Performance Center is very cumbersome", also received a strong agreement from the respondents. 26% chose Strongly Agree(5) while 56% selected Agree(4). This lead to the specification of a requirement that the developed Splunk dashboards had to be easy to use. As a response to this criticism, the authors developed a solution by displaying a field where the user could select which project to view. All other panels and fields will then be automatically filled in and generated. The minimization of user-input made the dashboards easier to understand and use.

Question 9; “I have to manually extract relevant information for a TES (Test evaluation Summary)” received a majority answer of 59% choosing Strongly Agree(5), and 22% choosing Agree(4). This made the authors prioritise the development of the Splunk dashboards since exporting a Splunk dashboard that automatically filled in graphs and tables would eliminate most of the manual labour involved in writing a TES.

Question 12; “The lack of a drill-down function in system logs leads to unnecessary manual work”. Here 52% answered Strongly Agree(5) and 26% answered Agree(4). In other words, the majority of the survey respondents agreed to this statement which conveyed the importance of drill-down functionality in the Splunk dashboards to the employees. This made the authors put particular focus on it and led to all the developed graphs being zoomable and some having the option to modify the time span or limit the results showed.

4.7 Source Criticism

Source number [1] is from a Power-point presentation during orientation for new IKEA IT employees that was given to the authors at the beginning of this bachelor’s thesis work. The presentation took place at IKEA’s global headquarters in Älmhult and is therefore deemed to be credible.

Sources [2], [3], [5], [6], [7], [10], [13], [14], [16], [20], [21], [23], [24], [25] and [30] originate from companies own homepages for the various products used during this thesis work, and are evaluated as credible. While companies are likely to have a biased towards their own products it is nevertheless unlikely that such major companies would be outright dishonest about their products.

Source number [4] M. Jönsson, S. Karlsson. LoadSplunker - Integration between HP Performance Center and Splunk. Bachelor thesis work. Lund 2015, is the written bachelor thesis report of the previous students who developed LoadSplunker. The work has been reviewed by supervisors and examiners and is therefore viewed as credible.

Sources [8], [9], [11], [12], [15], [26], [31] and [32] are from different sources of internet lexicons (Wikibooks, Guru99, TutorialsPoint, Webopedia, Northwaysolutions). These are usually written by expert users of the different products. However, since this is not something that can be verified, these sources should be considered less credible. It should be noted that while these sources are less trusted, every followed tutorial or tip gathered from these sources have proven to work during this thesis.

Sources [17], [18] and [27] are from the official Splunk Docs. which are documents gathered by Splunk employees and presented as a lexicon, due to this they are viewed as credible.

Sources [28] and [29] are Splunk references retrieving directly from the Splunk website. As they are extracted from the developer’s own official site they are deemed to be credible.

Source [19] is a Kanban reference/tutorial from a website called dZone. dZone is a large community with over 1 million members that publishes technical documentation for software professionals. However, the sites community forum is open for contribution by all members and is therefore not evaluated as credible.

Source [22] is a reference to an official Splunk course undertaken by the authors of this thesis. The course was paid for by IKEA and delivered by the engineers and software architects employed by Splunk. This source is therefore deemed to be credible.

5 Result

The authors were successful in removing the SiteAdmin dependency from LoadSplunker and making the software run on Linux only. LoadSplunker now runs completely within the Linux sphere. The root cause of LoadSplunker being Windows-dependant was that the previous solution relied on SiteAdmin that utilised Windows-only connection libraries such as COM4J. All the code that was auto-generated by SiteAdmin in the previous solution was therefore cut out or replaced, while ensuring that the classes responsible for converting the raw test result data to XML remained intact or operational after rework. While successfully implemented it took more time and resources than anticipated with 80% of the previous code being discarded or reworked. Reworking virtually an entire thesis assignment before working on the intended one inevitably narrowed the scope and size of the Splunk dashboards, which was the original main goal of this thesis assignment. The reason for so much for the old solution being discarded was that the previous students relied on the SiteAdmin API to inform LoadSplunker when new test results are available in Performance Center and trigger their extraction software. While it is true that SiteAdmin is one of the only ways of instantly triggering an alert when a new test result is available, this approach didn't factor in that the vast majority of the test runs aren't time-critical. The new solution instead relies on a small script running in the Performance Center environment that polls the test result database every 20 minutes and sends the results to LoadSplunker, getting away from the SiteAdmin and Windows dependencies.
Any test results that are newer than the previous poll timestamp will be uploaded to LoadSplunker and the test results extracted to Splunk. Most test scenarios take several hours to run and a worst case scenario of 20 minutes extra waiting time was regarded as irrelevant by all the stakeholders in question at IKEA IT. The new solution is much more lightweight and simple, see figures 26 and 27.

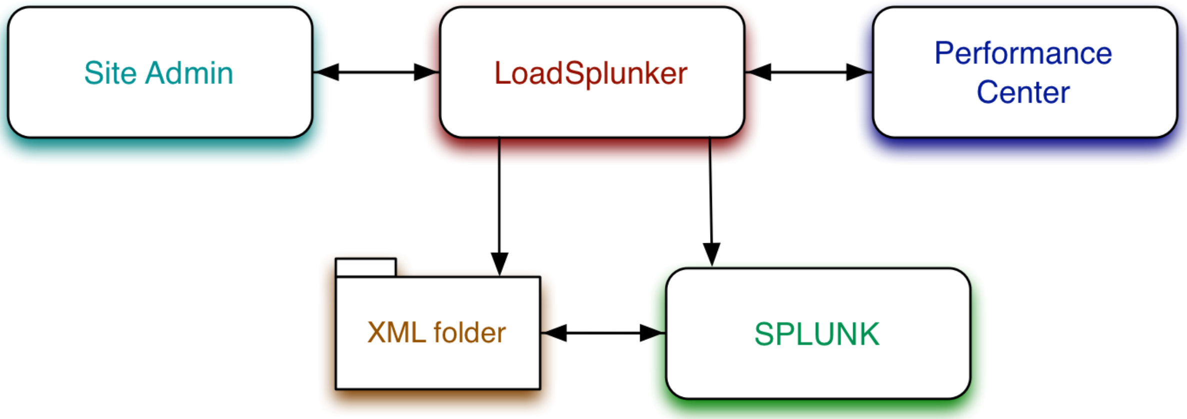


Figure 26. The communication flow in the previous thesis work.

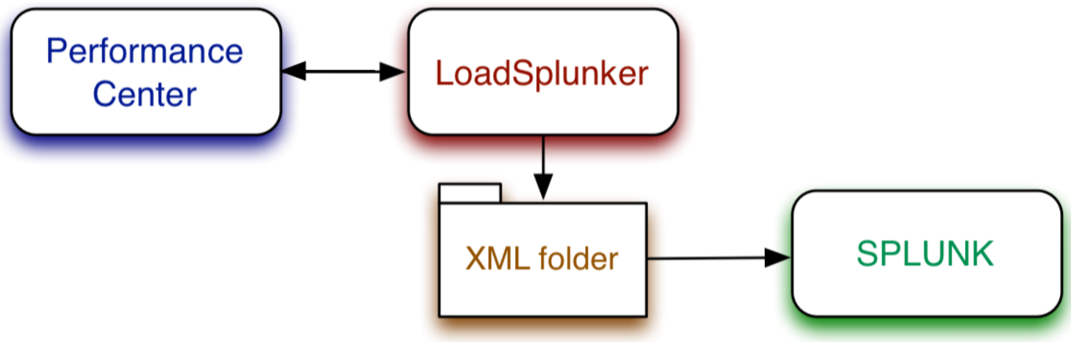


Figure 27. New communication flow after removing the SiteAdmin dependency

While the XML structure of the data being uploaded to Splunk has been improved, it must be noted that it is still unnecessarily complex. A simple reason for this is that the data from Performance Center was never intended to be extracted to a third-party software. You could even go so far as to say that there are safeguards in place in Performance Center to prevent such tampering, with a lot of field headers simply being "NULL" or "Value", so an observer won't know what the field values represent seeing as there is no header for reference, but HP Analysis knows because of the destination of the data. This lead to a lot of manual deduction and blind trial and error to determine what data is actually being processed and worked with.

The main goal of this thesis assignment was to create dynamic and easy to understand dashboards in Splunk, taking into consideration the varying levels of technical proficiency of the stakeholders at IKEA. In the end it was decided to create three separate dashboards, one for the more business orientated side, one for developers and finally one for test managers. As it stands today the dashboards aren't a complete replacement for the Analysis Tool, seeing as it has several thousand features and visualization options and it is outside the scope of this thesis work to implement all of them, see figure 28 for an overview over the workflow.

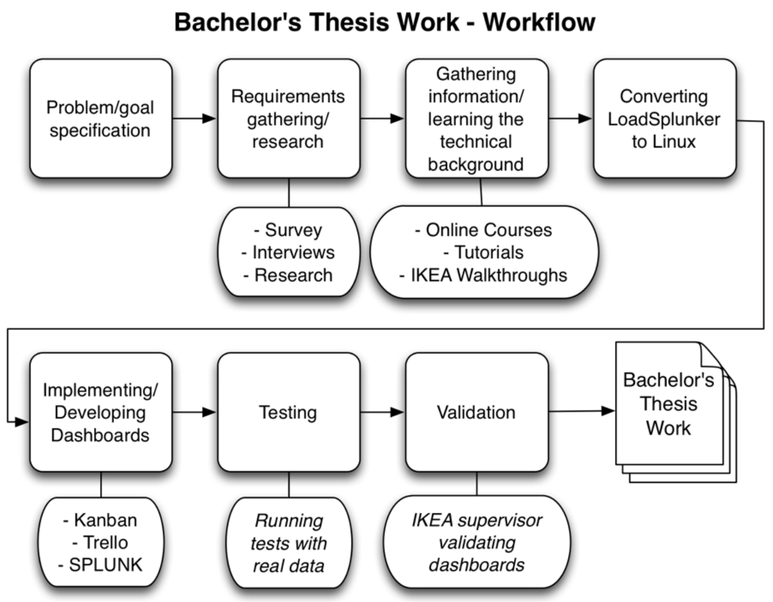
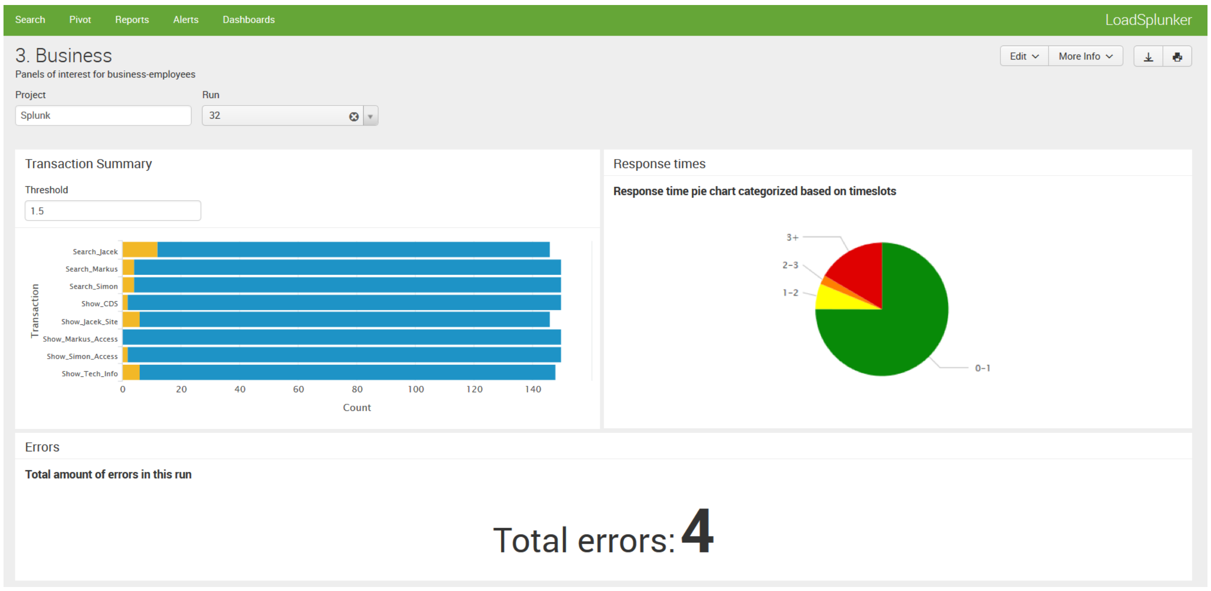

Bachelor's Thesis Work - Workflow
Problem/goal specification
Requirements gathering/ research
Gathering information/ learning the technical background
Converting LoadSplunker to Linux
- Survey
- Interviews
- Research
- Online Courses
- Tutorials
- IKEA Walkthroughs
Implementing/ Developing Dashboards
Testing
Validation
Bachelor's Thesis Work
- Kanban
- Trello
- SPLUNK
Running tests with real data
IKEA supervisor validating dashboards

Figure 28. Overview over this thesis work's workflow.

5.1 Dashboards

5.1.1 Business



3. Business

Panels of interest for business-employees

Project: Splunk

Run: 32

Transaction Summary

Threshold: 1.5

Response times

Response time pie chart categorized based on timeslots

Errors

Total amount of errors in this run

Total errors: 4

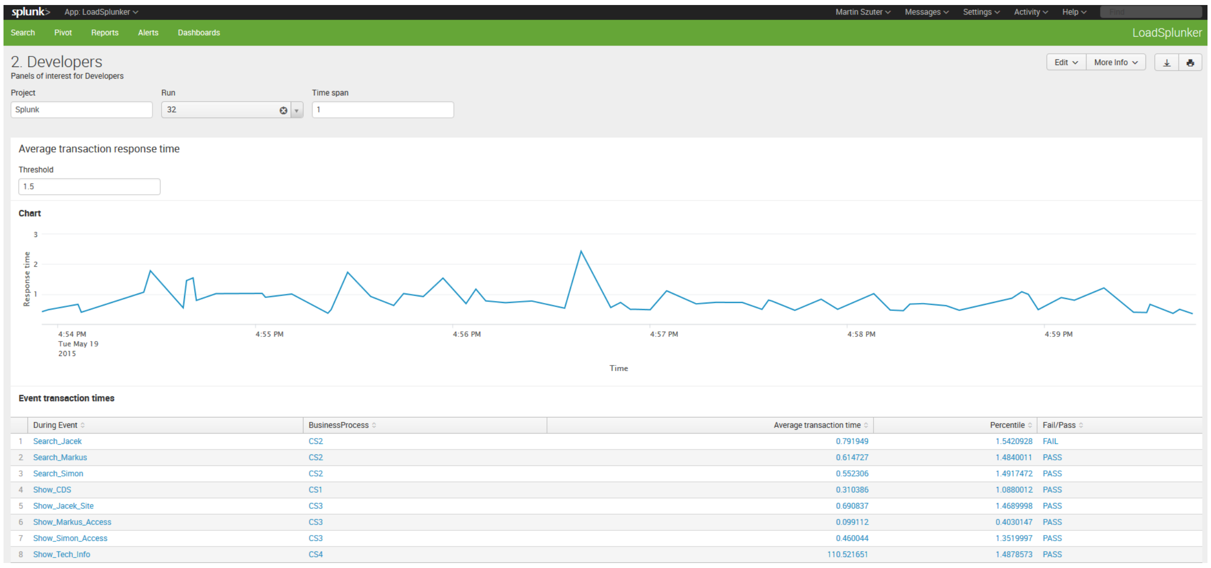
Figure 29. Example panel from the business dashboard.

The business dashboard is intended to give a general overview of the latest test runs, sticking to basic pass/fail, green/red parameters and avoiding going into too technical detail. To achieve this most of the test results are sorted, grouped and colored based on result. The majority of the graphs have had drill-down functionality disabled, see figure 29.

To be able to present the panels shown above, two specific data-files had to be searched and joined in Splunk, a file named Event_meter.xml and the other being Event_map.xml. Event_meter is the bigger file including all the important measurements as transaction times, transaction values, number of Virtual Users and so forth.
Event_map is needed to gather information about Event Types and Event Names of all the transactions. As can be seen in the search example below, to display the Transaction Summary in the dashboard, the two files are being joined with the *join inner* command with EventID as their mutual field. Then the statistics, count and evaluate commands are being used to group the data into a table where values of 1.5 or lower are presented with the label PASS and all values that are above 1.5 will be labeled as FAIL. Lastly everything is sorted and presented based on the EventName which is specified in the search by the last keyword: *by EventName.*

Search for retrieving and presenting Transaction Summary as shown in Figure 29:
source="\\xml\\Splunk\\run_32\\Event_meter.xml" | join type=inner EventID [search source="*\\xml\\Splunk\\run_32\\Event_map.xml" | where EventType="Transaction"] | stats count(eval(Value<=1.5)) AS Pass, count(eval(Value>1.5)) AS Fail by EventName*

5.1.2 Developers



2. Developers
Panels of interest for Developers

Project: Splunk
Run: 32
Time span: 1

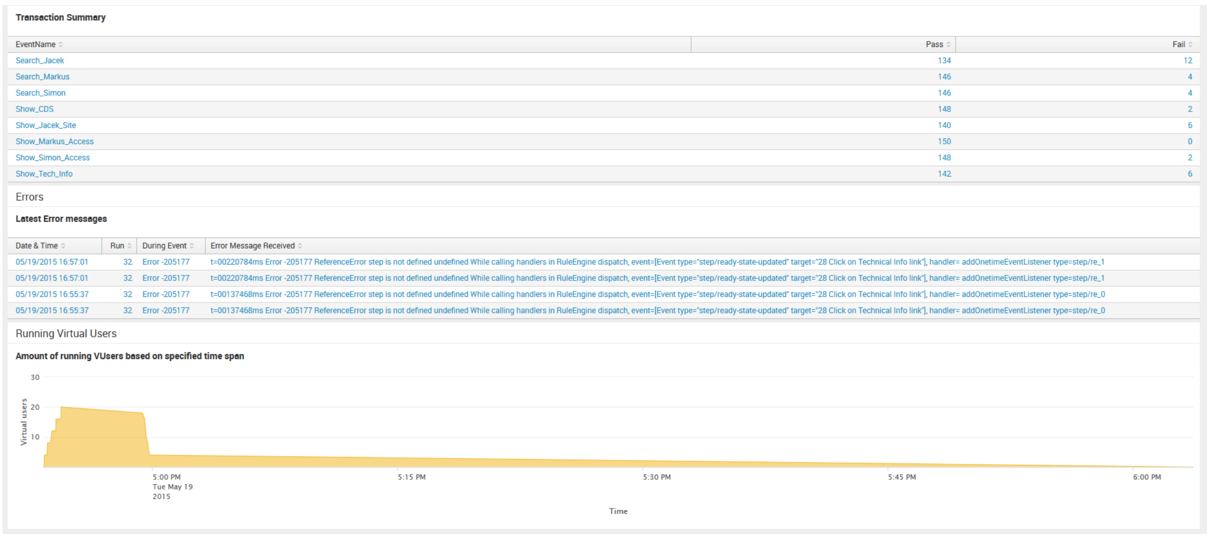
Average transaction response time
Threshold: 1.5

Chart

Event transaction times

	During Event	BusinessProcess	Average transaction time	Percentile	Fail/Pass
1	Search_Jacek	CS2	0.791949	1.5420928	FAIL
2	Search_Markus	CS2	0.614727	1.4840011	PASS
3	Search_Simon	CS2	0.552306	1.4917472	PASS
4	Show_CDS	CS1	0.310386	1.0880012	PASS
5	Show_Jacek_Site	CS3	0.690837	1.4689998	PASS
6	Show_Markus_Access	CS3	0.099112	0.4030147	PASS
7	Show_Simon_Access	CS3	0.460044	1.3519997	PASS
8	Show_Tech_Info	CS4	110.521651	1.4878573	PASS

Figure 30. Example panel from the Developers dashboard.



Transaction Summary

EventName	Pass	Fail
Search_Jacek	134	12
Search_Markus	146	4
Search_Simon	146	4
Show_CDS	148	2
Show_Jacek_Site	140	6
Show_Markus_Access	150	0
Show_Simon_Access	148	2
Show_Tech_Info	142	6

Errors

Latest Error messages

Date & Time	Run	During Event	Error Message Received
05/19/2015 16:57:01	32	Error -205177	t=00220784ms Error -205177 ReferenceError step is not defined undefined While calling handlers in RuleEngine dispatch, event=[Event type="step/ready-state-updated" target="28 Click on Technical Info link"], handler= addOnetimeEventListener type=step/re_1
05/19/2015 16:57:01	32	Error -205177	t=00220784ms Error -205177 ReferenceError step is not defined undefined While calling handlers in RuleEngine dispatch, event=[Event type="step/ready-state-updated" target="28 Click on Technical Info link"], handler= addOnetimeEventListener type=step/re_1
05/19/2015 16:55:37	32	Error -205177	t=00137468ms Error -205177 ReferenceError step is not defined undefined While calling handlers in RuleEngine dispatch, event=[Event type="step/ready-state-updated" target="28 Click on Technical Info link"], handler= addOnetimeEventListener type=step/re_0
05/19/2015 16:55:37	32	Error -205177	t=00137468ms Error -205177 ReferenceError step is not defined undefined While calling handlers in RuleEngine dispatch, event=[Event type="step/ready-state-updated" target="28 Click on Technical Info link"], handler= addOnetimeEventListener type=step/re_0

Running Virtual Users

Amount of running VUsers based on specified time span

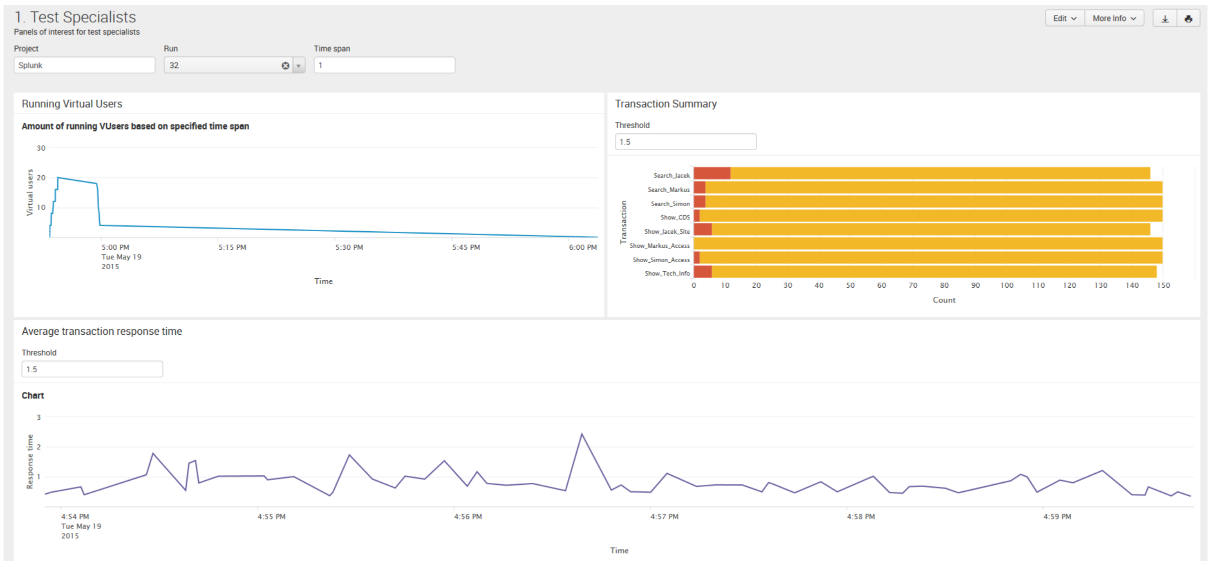
Figure 31. Another example panel from the Developers dashboard.

The developers dashboard is intended to be more of an intermediate dashboard, providing a general overview of the latest test runs while still going in to some technical detail. A lot of information overlaps here between business and test specialists, see figures 30 and 31.

One distinguished search in the developer dashboard is the one for presenting the amount of running Virtual Users. To retrieve information about the Virtual Users used in the test run, two XML-files had to be searched and joined in Splunk, namely VuserEvent_meter.xml and VuserStatusID.xml. As can be seen in the search example below, the two files are joined and evaluated based on if the Virtual User's status name is set to *Run,* which is done with the if-statement that can be found in the search.
Since the results came out in reversed order, the *reverse* command had to be used to get it back to the prefered order. Lastly the sum of the running Virtual Users is being displayed in a chart with the time span of 1 second, as specified in the last two commands of the search.

Search for retrieving and showing amount of running virtual users:
source="\\xml\\Splunk\\run_32\\VuserEvent_meter.xml" | join type=inner VuserStatusID [search source="*\\xml\\Splunk\\run_32\\VuserStatus.xml"] |eval running=if(VuserStatusName=="Run", InOutFlag, 0) | reverse | streamstats sum(running) as sum | timechart span=1 last(sum) as VUsers*

5.1.3 Test Specislists


1. Test Specialists
Panels of interest for test specialists

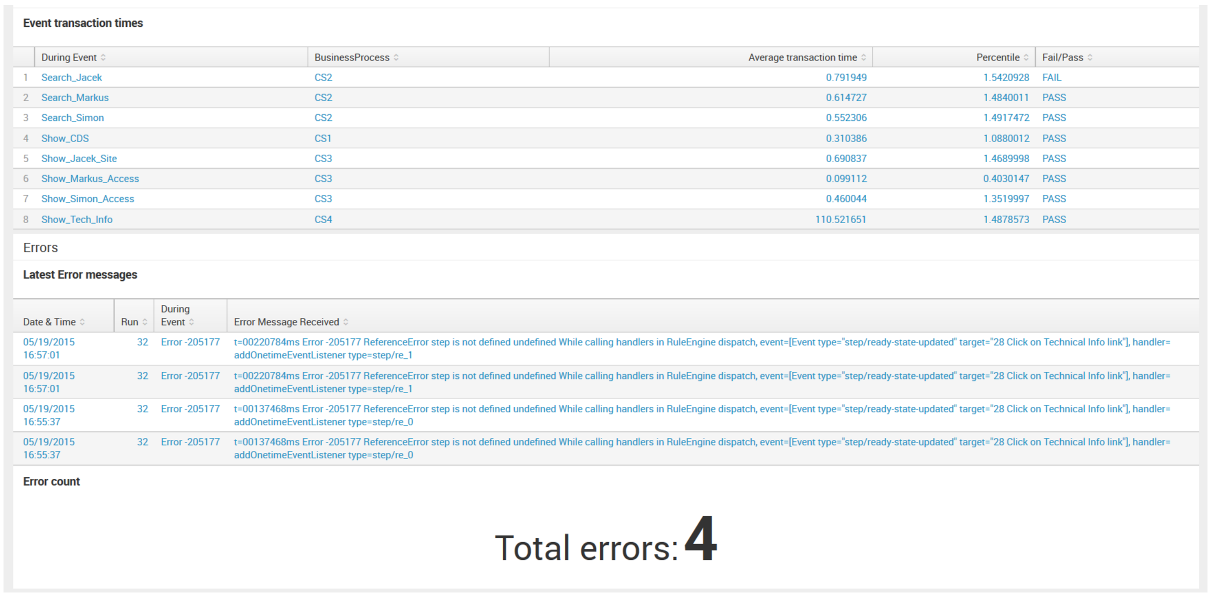
Project: Splunk
Run: 32
Time span: 1

Running Virtual Users
Amount of running VUsers based on specified time span

Transaction Summary
Threshold: 1.5

Average transaction response time
Threshold: 1.5
Chart

Figure 32. Example panel from the Test Specialist dashboard.


Event transaction times

	During Event	BusinessProcess	Average transaction time	Percentile	Fail/Pass
1	Search_Jacek	CS2	0.791949	1.5420928	FAIL
2	Search_Markus	CS2	0.614727	1.4840011	PASS
3	Search_Simon	CS2	0.552306	1.4917472	PASS
4	Show_CDS	CS1	0.310386	1.0880012	PASS
5	Show_Jacek_Site	CS3	0.690837	1.4689998	PASS
6	Show_Markus_Access	CS3	0.099112	0.4030147	PASS
7	Show_Simon_Access	CS3	0.460044	1.3519997	PASS
8	Show_Tech_Info	CS4	110.521651	1.4878573	PASS

Errors

Latest Error messages

Date & Time	Run	During Event	Error Message Received
05/19/2015 16:57:01	32	Error -205177	t=00220784ms Error -205177 ReferenceError step is not defined undefined While calling handlers in RuleEngine dispatch, event=[Event type="step/ready-state-updated" target="28 Click on Technical Info link"], handler= addOnetimeEventListener type=step/re_1
05/19/2015 16:57:01	32	Error -205177	t=00220784ms Error -205177 ReferenceError step is not defined undefined While calling handlers in RuleEngine dispatch, event=[Event type="step/ready-state-updated" target="28 Click on Technical Info link"], handler= addOnetimeEventListener type=step/re_1
05/19/2015 16:55:37	32	Error -205177	t=00137468ms Error -205177 ReferenceError step is not defined undefined While calling handlers in RuleEngine dispatch, event=[Event type="step/ready-state-updated" target="28 Click on Technical Info link"], handler= addOnetimeEventListener type=step/re_0
05/19/2015 16:55:37	32	Error -205177	t=00137468ms Error -205177 ReferenceError step is not defined undefined While calling handlers in RuleEngine dispatch, event=[Event type="step/ready-state-updated" target="28 Click on Technical Info link"], handler= addOnetimeEventListener type=step/re_0

Error count

Total errors: 4

Figure 33. Another example panel from the Test Specialist dashboard.

The test specialist dashboard is the largest and most technical, seeing as very little data is irrelevant to a test specialist after a run is completed. Drill-down functionality is specified and supported in virtually every panel, see figures 32 and 33.

As an example, below is the search used for retrieving and displaying a chart over average response times as can be seen in the Test Specialist dashboard.

The search begins with joining two XML-files including the necessary information about response times, namely the Event_meter.xml and Event_map.xml. These are joined in Splunk with the *join type=inner* command which gathers all the fields from the two files and makes them available for searching. As specified in the search below, only events with EventType Transaction are being joined and displayed. Lastly the values are presented in a chart with a default time span of 1 second and sorted by EventName. The last two commands tell Splunk not to display null values and sets the limit of hits to unlimited.

Search for retrieving and displaying a chart over average transaction response times:

```
source="*\\xml\\Splunk\\run_32\\Event_meter.xml" | join type=inner EventID [search
source="*\\xml\\Splunk\\run_32\\Event_map.xml" | where  EventType="Transaction"] |
timechart span=1 avg(Value) by EventName usenull=f limit=0
```

5.2 Lookups

Due to the large overhead cost of a lot of the heavy searches a lot of lookups were configured. To recap: a lookup can perform a search with set intervals and save the results into the Splunk database for later review. This massively speeds up response times of a lot of dashboards and panels and helps save system resources. An example of this is server response times which is represented in some form in every dashboard and is a relatively heavy search. Instead of performing this search fresh for every single view of the dashboards it is simply performed once every hour with the results being updated in the Splunk database.

Splunk also supports panels and dashboards to be exported in PDF and many other formats, if a static report still needs to be generated for any purpose. Test specialists also still have the option of extracting more detailed or separate data of interest from Splunk and generate static PDF reports for stakeholders on the fly, instead of configuring a new test run in the Analysis Tool.

5.3 Role-based system

This thesis work was unsuccessful in implementing a role-based system for the Splunk dashboards. This goal had the lowest priority of the initial goals and was de-prioritized due to time constraints and the technical and time-consuming solution required to implement it. The previous students had also explored the possibility of implementing a role-based system and reached the same conclusion. IKEA has an active directory database over all their employees and the hope was to use an LDAP (Lightweight Directory Access Protocol) [26] filter to automatically provide Splunk with the correct access right information. However, access control in Splunk is based on user roles and can only be assigned to users already in the Splunk database. Users would therefore have to be created manually and have roles assigned to them, but with over 150.000 employees worldwide[1] this would quickly prove to be quite a tedious and unavailing task. The previous students had explored the possibility of using the SiteAdmin and Splunk REST API:s to create a new user in Splunk for every user registered only in the Performance Center database. The cloned users would have the same user names but a default password that needs to be changed. While feasible and fully viable, this approach uses SiteAdmin which is dependant on Windows, which this thesis work has strived to distance itself from. While the role-based system fell outside the scope of this thesis work the authors hope it will be implemented by future students or employees at IKEA.

6 Conclusion

During the implementation of this thesis work, the real world demand for a replacement of the Analysis tool became apparent. The authors witnessed employees having to manually copy and paste test result data into excel spreadsheets on a regular basis and bounce test reports back and forth between stakeholders for weeks at a time with employees left having their hands tied in between reports. Replacing the Analysis Tool with Splunk could eliminate almost all of this downtime and the authors believe that this thesis assignment have successfully implemented a proof-of-concept with LoadSplunker. While there is potential for test reports being replaced completely with Splunk for recurring or regularly scheduled tests, test reports can probably never be fully automated for large projects or new test runs. For large projects, focus should instead be shifted towards creating individual dashboards in Splunk on a per-project basis. This would remove a lot of the overhead and downtimes associated with large projects and give all stakeholders a much better overview. Splunk has the added advantage of being virtually unrestricted in its customizability, with the option of running JavaScript and CSS under the hood [27], although it does not aggregate data which makes trend analysis over a longer period impossible, which is a drawback of Splunk.

During the initial stages of this thesis assignment, a problem specification was written, see section 1.3. The following are conclusions to those specifications.

6.1 Review of implemented software

How is LoadSplunker currently implemented (structure, database, Splunk)?
After reviewing the previous students' thesis work and getting a hands-on with the LoadSplunker-application, the authors got a good understanding of LoadSplunker's functionality and its code structure. The conclusion the authors came to was that there is a good code-structure, with Java-classes being distributed in logically named folders and all methods having explicit comments, which aided development and minimized trial-and error. The connection to the Performance Center database was made through the SiteAdmin REST API. A large part of the code implemented by the previous students was dedicated to monitoring Performance Centers' database and informing LoadSplunker of when new test results were available. When new raw test data was available LoadSplunker would extract and process on a by-line basis and convert the data to XML format. The XML files were then uploaded to a folder on the windows server that Splunk forwarder had been configured to monitor. As soon as new XML files were available Splunk would then upload the files in batch mode with a sinkhole preference, meaning the source XML files would be deleted upon successful upload. This was done with the purpose of maintaining server storage space and was the correct choice as the memory would otherwise rapidly fill up.

How complicated is it to learn and further develop LoadSplunker?
While the source code was well structured and well commented, there was nevertheless a large amount of unnecessary complexity due to auto-generated code by SiteAdmin. The authors found it time-consuming to deduce and sort out which classes handled monitoring of the database, and which classes handled the extraction and conversion of the raw test data. The authors originally attempted to implement new dashboards straight away, but seeing as the format of the data uploaded to Splunk was unnecessarily complex, this proved very difficult. Instead it resulted in focus being shifted towards improving the previous

solution before expanding upon it. This lead to the overall difficulty and complexity of learning and developing LoadSplunker to be perceived by the authors as quite high.

How can LoadSplunker be converted from Windows-compatibility to Linux-compatibility?
The root cause of LoadSplunker being Windows-dependant was that the previous solution relied on SiteAdmin that utilised Windows-only connection libraries such as COM4J. All the code that was auto-generated by COM4J as well as the .dll files from SiteAdmin in the previous solution would therefore have to be cut out or replaced, while ensuring that the classes responsible for converting the raw test result data to XML remained intact and operational. This was the only time during this thesis work that the authors had to resort to an unstructured trial-and-error approach to determine how much of the code could be cut out while keeping the XML format intact. The authors approach involved manually downloading a raw test result file from HP Performance Center and uploading it to LoadSplunker. The authors would then cut out or replace as much code as possible and regularly attempt to upload it to Splunk. As long as the structure and data being uploaded remained the same further improvements and modifications were made. While unstructured, this approach proved effective in removing the SiteAdmin dependency.

6.2 Automatic report creation (Dashboard)

How can the generated XML-file be loaded into the dashboard to present the test results?
The previous Splunk forwarder running on Windows was replaced with a new Splunk forwarder running on Linux, which has been set up to forward data to the IKEA Linux Server where Splunk currently is installed. The XML files being uploaded have had their structure improved, but it is still more complicated than desired to present the test results in the dashboards. While the authors feel they have become proficient in manipulating and displaying the data, there is certainly room for improvements in the XML structure, and future developers should consider working with other file structures altogether, for example JSON as mentioned in chapter 4.6. The possibility of extracting the test result data from the master T-table could also be explored, which includes additional information about a test run.

How can LoadSplunker determine the type of test data and choose a suitable template for the dashboard to present it with?
During development it emerged that Splunk unfortunately does not have a pre-installed template for XML-structured files. A specific configuration file instead had to be created that uses *Regular Expression* statements that manipulates, sorts and structures the XML files into events during the uploading and forwarding process. While Regular Expressions are a powerful tool in the right hands, they take a long time to master. The Regular Expression statements used in this current LoadSplunker solution proved helpful in improving the XML structure, there is certainly room for improvement in the future.

How can LoadSplunker be made easier to use than the currently used Analysis tool?
One of the big advantages of LoadSplunker and Splunk is that the panels and dashboards are pre-defined by the developers and then simply displayed to users and stakeholders. While this can lead to a higher threshold in complexity for the developer, who has to configure all the dashboards and panels in advance, which can be a time-consuming process, it is only a one-time investment. There is a similar feature in the analysis tool called “templates”, but test specialists report that it does not function properly and do not use it. The results will then be updated automatically for every test-run. This makes it is

much easier for the average stakeholder to use, since they don’t have to configure the Analysis tool during every test run. As was discussed in chapter 4.7, the average user now only has to select which test run they wish to view the results from, before all relevant graphs and statistics are displayed, further increasing the usability of LoadSplunker. This means that stakeholders do not have to install or learn how to utilise the analysis software, seeing as LoadSplunker is web-based.

6.3 Role-based system

How to gain access to user privileges at IKEA?
As was discussed in chapter 5.3, this thesis work was unable to implement a role-based system. It is possible to implement a role-based system using the Splunk SiteAdmin and REST API’s by establishing a connection between LoadSplunker and the Performance Center database to retrieve user credentials. However, it was not developed during this thesis work due to time-constraints and the Linux-only requirement.

How can Splunk regularly be updated of changes to user privileges at IKEA?
LoadSplunker would have to poll the Performance Center user database regularly, though no solution to this was discovered that did not ultimately rely on SiteAdmin.

How to Implement user restriction in LoadSplunker based on the user privileges?
A separate login-session would have to be developed in LoadSplunker that would use the user credentials retrieved from the Performance Center database. However, new passwords would have to be assigned and deployed to all employees, which made this solution unfeasible to complete during this thesis work. All usernames are stored on the Performance Center database in clear text, which makes them retrievable. However, the passwords are encrypted and cannot be extracted for use in LoadSplunker. Having to create new passwords for all Performance Center users at IKEA would be very time consuming and wouldn’t be appreciated by the employees since they would have to sign in a second time everytime they wanted to use LoadSplunker to view test results. The Splunk web interface also requires a sign in already, meaning employees would in a worst-case scenario have to sign in three times.

6.4 Installation

How can this system be installed in IKEA’s environment?
LoadSplunker together with Splunk have been installed on a Linux Server at IKEA IT. LoadSplunker is initiated through IntelliJ and runs in the background polling for new test results at specified time intervals. When a new test result is available, LoadSplunker retrieves it from the Performance Center database and converts it to suitable XML-files and uploads it to an output folder. A Splunk forwarder has been set up on the Linux server that monitors the output folder of LoadSplunker where the converted XML-files are stored. Upon noticing that a new XML-file has been added to the folder, the forwarder uploads it to the Splunk database. The Splunk web environment is reached by navigating through a web-browser to the correct port of the Linux server on which Splunk is installed.

6.5 Research

How can a survey be formulated to determine the current shortcomings of the Analysis Tool and ensure that LoadSplunker's dashboards meet the requested requirements?
See Appendix, section 9.1 for the completed survey. The survey was written with the aid of the IKEA supervisor and has indeed helped to determine the current shortcomings of the analysis tool, as well as ensuring that LoadSplunkers dashboards meet IKEA's requirements

What development options can be undertaken by future developers to make LoadSplunker a complete analysis tool?
During this thesis work, LoadSplunker has been made Linux-only and dashboards have been developed in Splunk to present the test results. While the authors of this thesis work are pleased with the end result, there is ample room for future improvements. These improvements and suggestions are discussed in more detail in chapter 6.6.

6.6 Future Improvement suggestions

6.6.1 Overall Dashboard improvements

The dashboards created during this assignment serve as proof of concept prototypes where the extracted data is presented in three different dashboards aimed at test specialists, developers and business-employees. This is naturally not a limit for Splunk and the possibilities future improvements are huge. The dashboards could be expanded to show more information and be more user-modifiable where the user decides what is to be presented in a given dashboard. The dashboards could be supplemented with more types of dashboards aimed at more specific types of employees and target groups.
Another feature that is frequently requested by both stakeholders and test specialists is the ability to cross-reference test results originating from different systems. An example of a product where this would be valuable is a bottle of IKEA soda, that passes through multiple systems during production. Stakeholders frequently request an overview of these types of business flows, something that is reportedly very complicated to achieve in the analysis tool, which reportedly tends to break down when measuring metrics originating from more than four systems, whereas test specialists have received requests to measure metrics from hundreds of systems.

6.6.2 Dashboard Design

The authors, unfortunately did not have enough time to apply custom made design to the created dashboards, however some possibilities for dashboard design were researched and is something that should be considered for future improvements. It is possible to create a CSS-file, upload and assign it to a specific dashboard. Each panel, text-box, graph and search can be assigned a unique identifier which can be used inside the CSS file to set the design of the given element. Everything from colors, sizes, fonts to overall structure and form can be specified and experimented with.

6.6.3 Web-based reports

Splunk is very competent when it comes to presenting raw computer data, but that is not always enough for a load test report. Often graphs must be complemented with text explaining the content, static figures or user-generated tables. As this text will be have to be stored in Splunk, that makes for a lot of extra information. A way to overcome this is to create a website where all reports can be found. All the extra information mentioned above would be stored on the web page server and the Splunk dashboards and panels would be added as a plug in on the website. SDK's for PHP, HTML and Java Script can be downloaded from Splunk's website which makes such a solution possible [28]. With the help of the SDK's and a Splunk API, specified panels/dashboards could be extracted and implemented into the report-website, maintaining their interactivity.

6.6.4 Mobile application

Currently the smartphones owned by IKEA employees cannot be connected to their private internal network, as mobile phones currently aren't supported. However, if this would change in the future, there is a possibility of making the Splunk Dashboards mobile. Splunk has developed an API for mobile applications [29] with which the user-created dashboards can be implemented into an app for Android or iOS. This would make the sharing of reports and dashboards with stakeholders substantially easier where everybody involved could download the mobile application onto their smartphone and have access to the test results in the palm of their hand.

6.6.5 Graylog

Since the previous students chose to extract the test data into XML-files, there is a possibility that Splunk is not the most suitable choice for presenting the data. There exists an open source management software called Graylog created by Graylog inc [30]. which basically has the same functionality that of Splunk with the difference of being able to better handle XML-files.
Structuring XML-data into Splunk has proven to be an unnecessarily complex process where complicated Regular Expression had to be used. This is something that Graylog handles in a more flexible way by providing an integrated template for reading and structuring XML-data.
Graylog has not been tried personally by the authors of this thesis work and is therefore only a suggestion that would need further exploration.

7 Terminology

Kanban	Technique for planning, tracking and managing a software development process
GIT	Version Control System for managing software development.
LDAP	Lightweight Directory Access Protocol, used to connect, search and modify internet directories
regEx	Regular Expression, a protocol for specifying searches
Token	An input field representing a value inside Splunk dashboards.
Load generator	A server used for generating and running scripts.
API	Application Program Interface
GUI	Graphical User Interface
XML	Extensible Markup Language, a markup language that encodes documents in a format that is both human- and machine-readable
Agile	Set of principles for software development
MobaXterm	Extended terminal/command tool for Windows. Used in this assignment for connection to Linux-servers.
Script	Sequence of instructions that is interpreted or executed by another program.
TES	Test Evaluation Summary, the report written after a Test Run is complete.
HTTP	HyperText Transfer Protocol - the foundation protocol for data communication for the world wide web.
SDK	Software Development kit - a set of development tools for a certain software package, framework, operating system or similar development platform.
iPass	The login software used by IKEA to access their VPN service.

ALM	Application Lifecycle Management, the software suite used by IKEA to perform Load Tests.
HTML	HyperText Markup Language. Markup language for describing web documents (web pages).
CSS	Cascading style sheet - used for adding design features to web pages.
SSH	Secure Shell - Encrypted network protocol that allows remote login and other network services.
IRW	IKEA Retail Website, IKEA branch in charge of the IKEA website.
JSON	JavaScript Object Notation - programming language structured with attribute-value pairs.
PHP	Hypertext Processor, server side scripting language.

8 Sources

[1] Power-point presentation during orientation for new IKEA IT employees that was given to the authors at the beginning of this bachelor's thesis work.

[2] http://www8.hp.com/us/en/software-solutions/performance-center-testing/
Hp Performance Center homepage. Electronic resource, read 2016-05-10

[3] http://www.splunk.com
Splunk homepage. Electronic resource, read 2016-05-10

[4] M. Jönsson, S. Karlsson. LoadSplunker - Integration between HP Performance Center and Splunk. Bachelor thesis work. Lund 2015.

[5] http://alm-help.saas.hpe.com/en/12.50/api_refs/site_admin/webframe.html#topic3.html
Site Administration API Reference. Electronic resource, read 2016-05-16

[6] https://com4j.java.net
COM4J reference. Electronic resource, read 2016-05-16

[7] http://www8.hp.com/us/en/software-solutions/application-lifecycle-management.html
HP Application Lifecycle Management homepage. Electronic resource, read 2016-05-15

[8] https://en.wikibooks.org/wiki/C_Programming
C-language reference from Wikibooks. Electronic resource, read 2016-05-17

[9] http://www.guru99.com/understanding-vugen-in-loadrunner.html
Guru99 - Understanding Virtual User Generator in LoadRunner. Electronic resource, read 2016-03-15

[10] http://www8.hp.com/us/en/software-solutions/loadrunner-load-testing/
HP LoadRunner homepage. Electronic resource, read 2016-04-10

[11] http://www.guru99.com/how-to-use-controller-in-loadrunner.html
Guru99 - How to use Controller in LoadRunner. Electronic resource, read 2016-03-15

[12] http://www.guru99.com/how-to-use-analyzer-in-loadrunner-12-0.html
Guru99 - How to use Analyzer in LoadRunner. Electronic resource, read 2016-03-15

[13] http://www8.hp.com/us/en/software-solutions/sitescope-application-monitoring/
HP SiteScope homepage. Electronic resource, read 2016-04-01

[14] https://www.redhat.com/en/about/value-of-subscription
Red Hat Inc. homepage. Electronic resource, read 2016-03-10

[15] http://www.tutorialspoint.com/unix/unix-shell.htm
TutorialsPoint, Unix - What is Shells? Electronic resource, read 2016-03-10

[16] http://mobaxterm.mobatek.net
MobaXterm homepage. Electronic resource, read 2016-03-10

[17] http://docs.splunk.com/Documentation/Splunk/6.0.7/SearchTutorial/Usefieldlookups
Splunk Docs, Search tutorial, use field lookups. Electronic resource, read 2016-04-20

[18]
http://docs.splunk.com/Documentation/Splunk/6.2.9/SearchTutorial/Createnewdashboard
Splunk Docs, Search tutorial, Create Dashboards and dashboard panels. Electronic resource, read 2016-04-10

[19] https://dzone.com/refcardz/getting-started-kanban
dZone - getting started with Kanban. Electronic resource, read 2016-05-17

[20] https://trello.com
Trello homepage. Electronic resource, read 2016-05-17

[21] https://docs.google.com/forms
Google forms homepage. Electronic resource, read 2016-05-17

[22] Searching And Reporting With Splunk (eLearning) - Self paced eLearning with live lab access for 30 days. Product ID: EDU-SRPT-1

[23] https://www.jetbrains.com/idea/
IntelliJ homepage. Electronic resource, read 2016-05-17

[24] https://git-scm.com
GIT homepage. Electronic resource, read 2016-05-17

[25] https://www.skype.com/en/
Skype homepage. Electronic resource, read 2016-05-17

[26] http://www.webopedia.com/TERM/L/LDAP.html
Webopedia - what is LDAP. Electronic resource, read 2016-04-20

[27] http://docs.splunk.com/Documentation/Splunk/4.2.2/Developer/UseCSS
Splunk Docs, Developing Dashboards, views, and apps for Splunk web, Custom CSS. Electronic resource, read 2016-05-01

[28] http://dev.splunk.com/sdks
Splunk - Available SDK's. Electronic resource, read 2016-05-14

[29]
http://docs.splunk.com/Documentation/MintAndroidSDK/latest/DevGuide/Requirementsandinstallation
Splunk Android API. Electronic resource, read 2016-05-14

[30] https://www.graylog.org
Graylog homepage. Electronic resource, read 2016-05-14

[31] https://northwaysolutions.com/products/hp-loadrunner/
Northwaysolutions – HP Performance Center. Electronic resource, read 2016-05-18

[32] http://www.guru99.com/hp-alm-introduction.html
Guru99 - HP ALM introduction. Electronic resource, read 2016-05-19

8.1 Figures

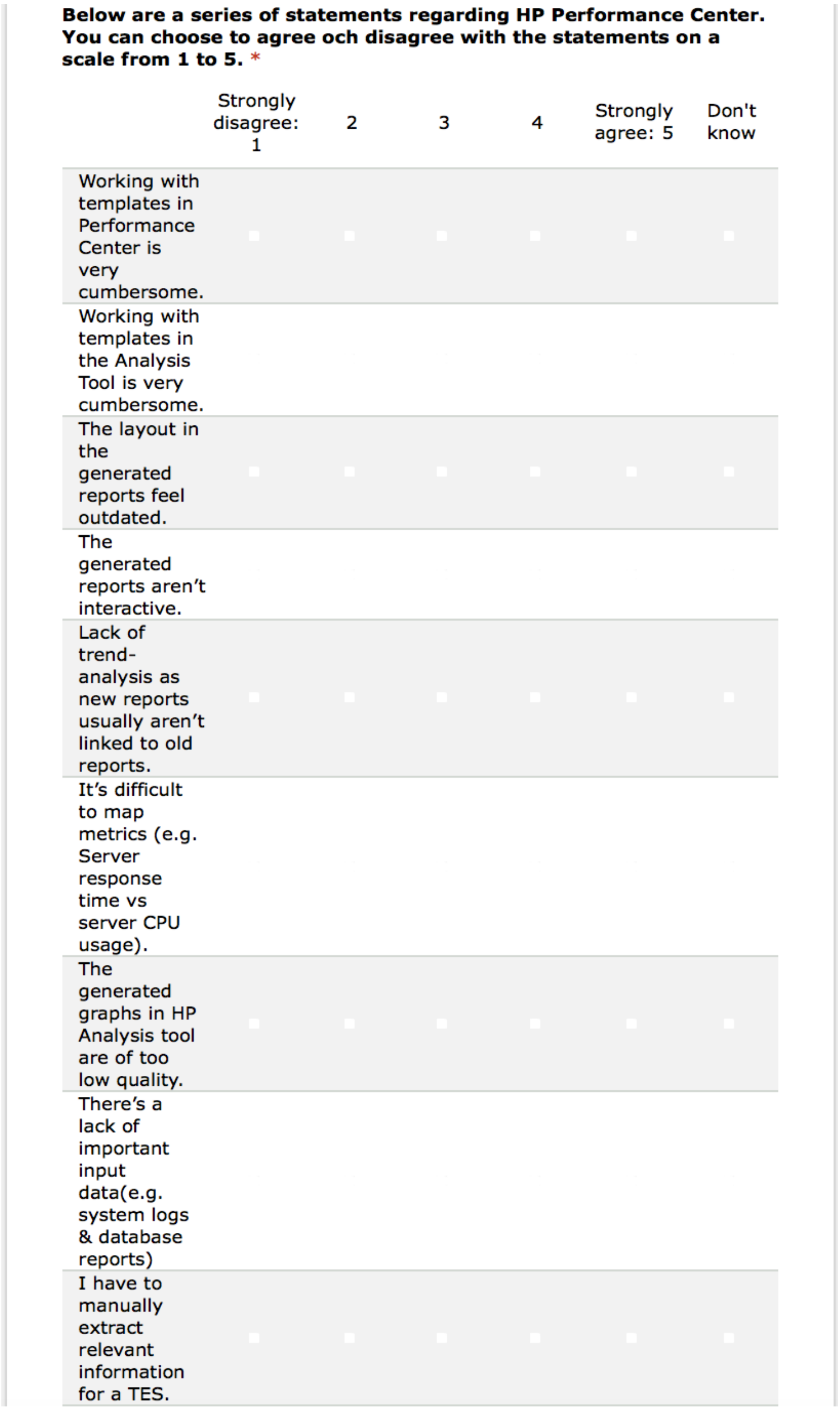
Figures 2-5 and 7-32 are original works by the authors, Justin Rees & Martin Szuter, based on observations made during this thesis work.

Figures 1 and 6 are redesigned from images that originate from external sources which are specified in the source chapter.

9 Appendix

9.1 Survey

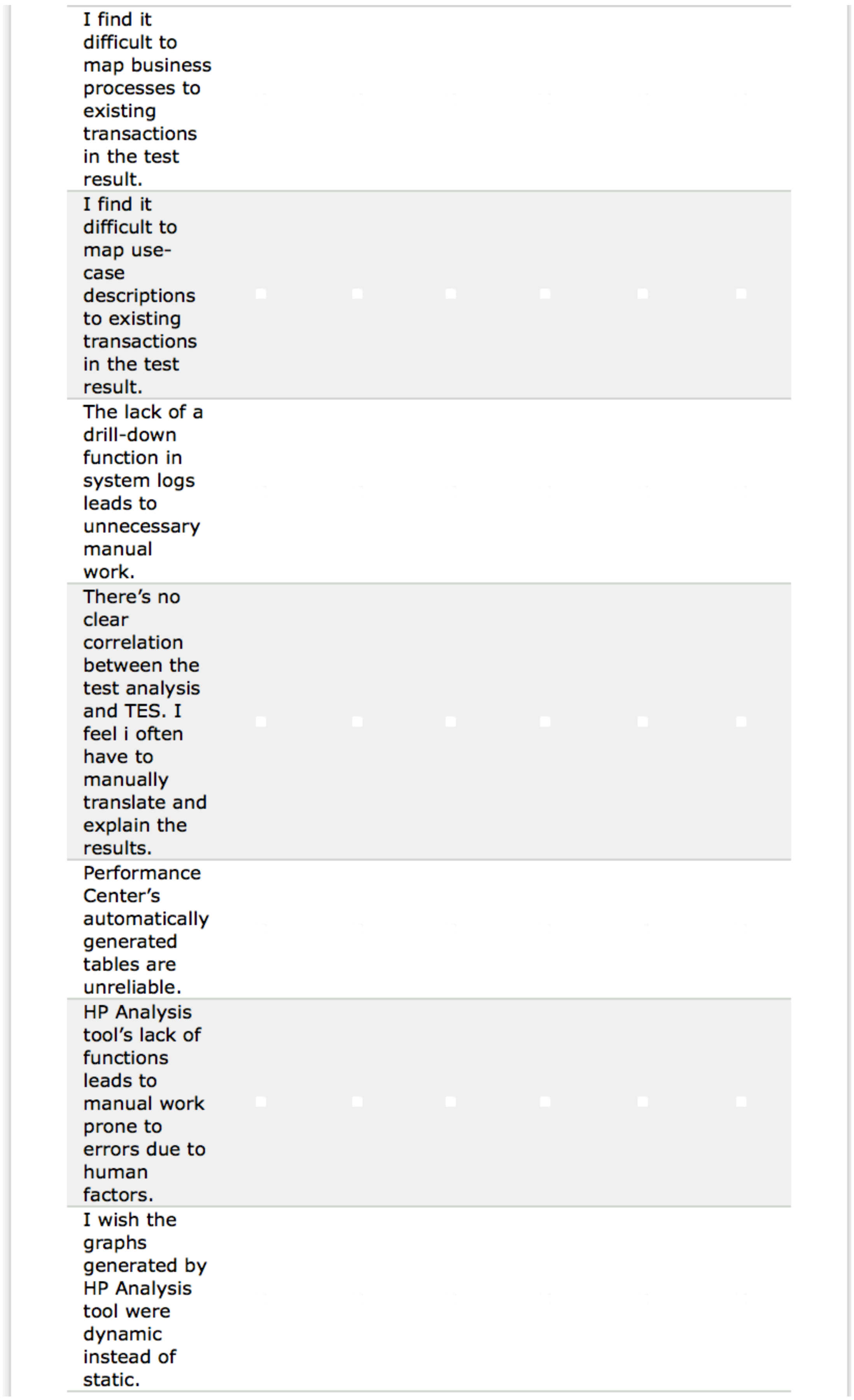
Below are the complete questions used in the survey.



Below are a series of statements regarding HP Performance Center. You can choose to agree och disagree with the statements on a scale from 1 to 5. *

	Strongly disagree: 1	2	3	4	Strongly agree: 5	Don't know
Working with templates in Performance Center is very cumbersome.						
Working with templates in the Analysis Tool is very cumbersome.						
The layout in the generated reports feel outdated.						
The generated reports aren't interactive.						
Lack of trend-analysis as new reports usually aren't linked to old reports.						
It's difficult to map metrics (e.g. Server response time vs server CPU usage).						
The generated graphs in HP Analysis tool are of too low quality.						
There's a lack of important input data(e.g. system logs & database reports)						
I have to manually extract relevant information for a TES.						

Figure 34. Screenshot of the survey, part 1.



I find it difficult to map business processes to existing transactions in the test result.						
I find it difficult to map use-case descriptions to existing transactions in the test result.	▫	▫	▫	▫	▫	▫
The lack of a drill-down function in system logs leads to unnecessary manual work.						
There's no clear correlation between the test analysis and TES. I feel i often have to manually translate and explain the results.	▫	▫	▫	▫	▫	▫
Performance Center's automatically generated tables are unreliable.						
HP Analysis tool's lack of functions leads to manual work prone to errors due to human factors.	▫	▫	▫	▫	▫	▫
I wish the graphs generated by HP Analysis tool were dynamic instead of static.						

Figure 35. Screenshot of the survey, part 2.