Haaga-Helia
University of Applied Sciences Ltd.

Dedicated Server Infrastructure Transformation to AWS Cloud In Lean Organisation

Harri Pyy
11.5.2017

Opinnäytetyö
Liiketalouden ylempi
ammattikorkeakoulututkinto
Tietojärjestelmäosaamisen
koulutusohjelma
2017

Tiivistelmä

Tekijä
Harri Pyy

Koulutusohjelma
Tietojärjestelmäosaamisen koulutusohjelma

Opinnäytetyön nimi
Dedicated Server Infrastructure Transformation to AWS Cloud In Lean Organisation

Sivu- ja liitesivumäärä
63 + 8

Opinnäytetyön tarkoituksena oli selvittää voidaanko Schibsted Media Group:ssa(Tori.fi) luopua nykyisistä ylläpidettävistä servereistä ja ostaa palvelimien hosting palvelut Amazon:n pilvestä hyödyntäen Amazon Web Services -komponentteja.

Opinnäytetyön lähtökohtana on selvittää voidaanko Tori.fi -palvelu siirtää Amazon:n pilvipalveluksi. Tarkoitus on myös tutkia voidaanko kaikki vaatimukset täyttää, ja onko siirtyminen pilvipalveluksi realistinen hanke. Työ pyrkii myös ratkaisemaan sellaisia esteitä, jotka lähtökohtaisesti aiheuttaisi ongelmia palvelun siirtämiseksi AWS:n pilvipalveluksi.

Opinnäytetyön tärkeimpänä tavoitteena olikin tuottaa SCM Suomi Oy:lle Proof-of-Concept tietoa palveluiden siirtämisestä ja palvelujen kehittämiseksi. Lisäksi tavoitteena oli selvittää minkälainen Lean:ia kehitysmalli liiketoimintatarpeiden toteuttamiseen yrityksessä oli käytössä projektin aikana.

Opinnäytetyössä sovellettiin lähestymistapana innovaatioiden tuottamista Leanissa ympäristössä, joka soveltuu erittäin hyvin kokeellisen mallin tuottamiseen. Aineiston keruumenetelminä käytettiin Proof-of-Concept kokeiluista saatua tietoa ja haastatteluja. Aineiston analysointi menetelminä käytettiin useita ketteriä Lean menetelmiä ja työvälineitä. Uusien liiketoiminta mahdollisuuksien määrittelyssä käytettiin Lean Canvas menetelmää.

Opinnäytetyön tuloksena luotiin Proof-of-Concept Hybrid malli pilveen menosta, jossa lähes 70% Internet liikenteestä siirtyy Amazon Web Services:n (AWS) pilvipalveluiden läpi. Schibsted:n tulevaisuuden suunnitelma on siirtyä tulevaisuudessa uuteen platformiin, jossa ohjelmistoarkkitehtuuri on suunniteltu paremmin pilviarkkitehtuuriin skaalautuvaksi.

Opinnäytetyössä tiivistettiin AWS:n pilvipalvelumallin perusominaisuudet seuraaviksi: kustannustehokkuus, skaalautuvuus, ylläpidettävyys, päivitettävyys, automatisoitavuus ja tietoturvallisuus. Opinnäytetyön tuloksena oli saatiin Proof-of-Concept konsepti ja suunnitelma koko palvelun viemisestä Amazonin pilvipalveluksi ja kuvaus Startup-henkisestä tuotekehitysmallista jota käytettiin toteutuksessa.

Johtopäätöksinä todetaan, että opinnäytetyöllä pystyttiin luomaan tietoa jota käytetään tulevaisuudessa uuden infrastruktuurin kehittämiseen ja lisätä ymmärrystä minkälainen ohjelmistoarkkitehtuuri parhaiten skaalautuisi pilviympäristöön.

Asiasanat
AWS, Cloud, Lean, Proof-of-Concept, Dedicated Server

Abstract

Author
Harri Pyy

Degree programme
Master’s Degree Programme in Information Systems Management

Report/thesis title
Dedicated Server Infrastructure Transformation to AWS Cloud In Lean Organisation

Number of thesis pages and attachment pages

63 + 8

The thesis tries to solve whether in the Schibsted Media Group (Tori.fi) it is possible to give up hosting dedicated servers and move the services Amazon's cloud using Amazon Web Services components.

The starting point of the thesis was to determine whether the services around Tori.fi can be moved to Amazon's Web Services(AWS). This thesis concentrates on to examine if all the requirements can be fulfilled and whether the transition to the cloud service is a realistic project. The study also aims to solve those obstacles which in principle would create problems for the transfer of service to the AWS cloud.

The main objective of the thesis was to produce Proof-of-Concept knowledge for transferring the services to AWS Cloud. Also, one goal was to find better solutions to improve practices and services. In addition, the aim was to find out what kind of Lean development model was used for the transformation. Lean methods are excellent when producing experimental Proof-of-Concept models.

Data collection methods used Proof-of-Concept experiments and interviews. Lean methods and agile tools were used for data collection. Searching for new business opportunities the Lean Canvas methodology were used.

As a result of this thesis, the Proof-of-Concept Hybrid cloud model was created. Almost 70% of Internet traffic being diverted from the Amazon Web Services(AWS). Schibsted's future plan is to move in the future to a new platform. The new platform has new software architecture which scales better way to cloud architectures.

The study was intensified the basic characteristics of AWS's cloud service model: cost-effectiveness, scalability, maintainability, upgradeability, automation and information security. As a result of the thesis, it was obtained Proof-of-Concept model and description how the operations were carried out in Start-up model.

Conclusions were that this study was able to create information to be used in the future development of new infrastructure and increase the understanding needed what kind of what architectures would scale the best way to cloud environment.

Keywords
AWS, Cloud, Lean, Proof-of-Concept, Dedicated Server

Table of Contents

1. Introduction 6
   1.1 Target 8
   1.2 Purpose of the Thesis 8
   1.3 Stakeholders 9
   1.4 Research Objectives 9
   1.5 Scope 10
   1.6 Research Questions 11
2. Research Methodology 13
   2.1 Strategy 13
   2.2 Data Collection Plan 14
3. Proof-of-Concept Cloud Infrastructure 15
   3.1 Amazon Web Services(AWS) 19
   3.2 System Architecture on Dedicated Servers 21
   3.2 Proof-of-Concept Architecture 24
   3.3 Tori.fi as Hybrid Cloud Architectures with AWS 25
   3.4 AWS Proof-of-Concept Image Servers 26
   3.5 Conclusions and Summary Of The AWS Proof-Of-Concept Infrastructure 27
4. Lean Agile Model in Organisation 28
   4.1 Lean Koju-model Explained 29
   4.3 Lean Organisation Flow 31
   4.4 Lean Six Sigma 32
   4.5 Lean Canvas 34
   4.6 Lean Loops 37
   4.7 Lean A/B Testing with Optimizely Tool 40
   4.8 Software Branching Methodologies in Lean 40
   4.9 Agile Product Development Process 42
   4.10 Lean “Kanban” Visualisation in Agile 44
   4.11 Jira Ticket Tool for Agile development 46
   4.12 Agile Way of Stand-ups 48
   4.13 Quality Assurance(QA) 49
   4.14 Conclusions And Summary of The Lean Model 50
5. Results of Data Analysis And Interviews 51
   5.1 Measurements Results 51
      5.1.1 Measurement Results of Website Latency 51
      5.1.2. Measurement Results of Work Efficiency 53
   5.2 Interview Results 54
      5.2.1 Interview Results About Website Loading Latency 54
      5.2.2 Interview Results About Flexibility Improvement 55
      5.2.3 Interviews Results of Work Efficiency 56
      5.2.4 Interview Result of the Factors Measuring Work Efficiency 57

5.3 Conclusions and Summary of the Data Analysis And Interviews 58
6. Conclusions 59
6.1 Answers to Research Questions 59
6.2 Suggestions for Cloudification 62
6.3 Suggestions for Lean model 63
References and Literature 64
Appendix 67
Appendix 1. Terminology And Abbreviations 67
Appendix 2. What is Zabbix 69
Appendix 3. XiTI tool by AT Internet 70
Appendix 4. Optimizely Tool 71

Figures:

Figure 1: Non-hierarchical organisation model(Giles H. 2016) 7
Figure 2: Present State and cloud 16
Figure 3: Server management at the present state 17
Figure 4: Future: Tori.fi concentrates on software development 17
Figure 5: AWS generic security options 18
Figure 6: AWS architecture and services(Vyas U. 2015) 19
Figure 7: Simple load balancing(AWS Elastic Load Balancing,2016) 20
Figure 8: Tori's Physical servers in one datacenter 21
Figure 9: System architecture overview 22
Figure 10: First approach to Tori's AS-IS transformation to AWS 24
Figure 11: Current AWS hybrid model 25
Figure 12: Architecture of Proof-of-Concept image servers 26
Figure 14: Examples of possible Kojus 29
Figure 15: Example of Koju-model organisation 29
Figure 16: Organisation flow in lean model 31
Figure 17: The DMAIC project methodology stages(Six Sigma, DMAIC Methodology, 2017) 32
Figure 18: Lean canvas used in Tori(Maurya A. 2010) 34
Figure 20: Moves the Needle lean loop 37
Figure 21: Lean testing if the concept is viable 38
Figure 22: Snapshot of Optimizely used for Tori experiment 40
Figure 23: Flow chart explains the GIT branching used at Tori 41
Figure 24: Tori's scrum sprint 42
Figure 25: Scrum agile development (Tal L. 2005, Chapter 1, Figure 1-5) 43
Figure 26: Example of the Kanban board in use 44
Figure 27: Creating Jira ticket 46
Figure 29: Selecting the Koju 47
Figure 30: Selecting the ticket priority 47
Figure 31: Example of velocity chart of Jira 48
Figure 32: Stand-up meeting in Tori 48
Figure 33: Tori.fi bandwidth usage from internal server monitoring 52
Figure 35: Describes the ping from office to front cloud in AWS 52
Figure 36: Describes the ping from office to local dedicated servers 53

1. Introduction

Tori.fi is the leading online marketplace in Finland. In classified ads, Tori.fi is the most efficient way to reach millions of consumers in every month.(Tori Media Homepage, 2017) Tori.fi online classified service was in the beginning of this thesis work hosted fully on virtualised XenServers. The idea of making this thesis became from the need of Schibsted Media Groups need to move the service to cloud environment. This thesis deals with the design and implementation phase of transferring the service to Amazon cloud computing environment called Amazon Web Services, AWS. The thesis covers the lean operational model that was used in the organization. The IT companies such as Tori tends to continuously improve the IT services and methods, and therefore Lean principles could offer tools to reach this goal.

Cloud computing is the on-demand delivery of computing power, database storage, applications, and other IT resources through a cloud services platform via the internet with pay-as-you-go pricing. (About AWS, 2016)

Similar Websites like Tori are being hosted in several Schibsted countries, which have generally shared Schibsted Blocket.se platform. Tori.fi's code base was split originally November 2009 from Blocket.se. Nowadays the Tori.fi service is running on dedicated servers in Telecity Groups server facilities in Helsinki.

This thesis is an initial study with its key learnings to move the services to part of Amazon Cloud Services. The purpose to move the services to Cloud comes from the need to reduce the system maintenance work. In the future, it would be ideal that service will be designed fully hardware independent. Software development will move towards futuristic DevOps thinking.

This thesis work covers how product and technology development is carried out in Schibsted Media Group Finland, and how it was used during cloudification. Also, this work describes how project management methodologies have evolved during Tori.fi's growth.

At the beginning, there didn't really exist any operational, or project management plans or tools. High level "Hoshin" strategy plan was implemented. Most of the operational plans were minimal, and not large planning was carried out. Changes and development to the service were quick and smooth. Naturally, the organization grew and Agile development methods were taken into to use. Agile development has good practices but they tend to be too slow for IT companies nowadays. It was decided to move towards the Lean principles, and reduce the bureaucracy needed in company's operations.

It is very normal that operation models change during the time the company size grows. Also, we are constantly trying to improve the operation models according to lean principles. This thesis describes how the processes have evolved, and the reasons why we have ended up to current operation model.

Our development model in Schibsted Media in Finland is nowadays more startup and experimental startup-lean driven. Latest changes in the organization have lead that we broke then traditional organization model to lean non-hierarchical Koju-model. This decision based on that when the organization grows the R&D development becomes slower. The new model will be described in this project work.

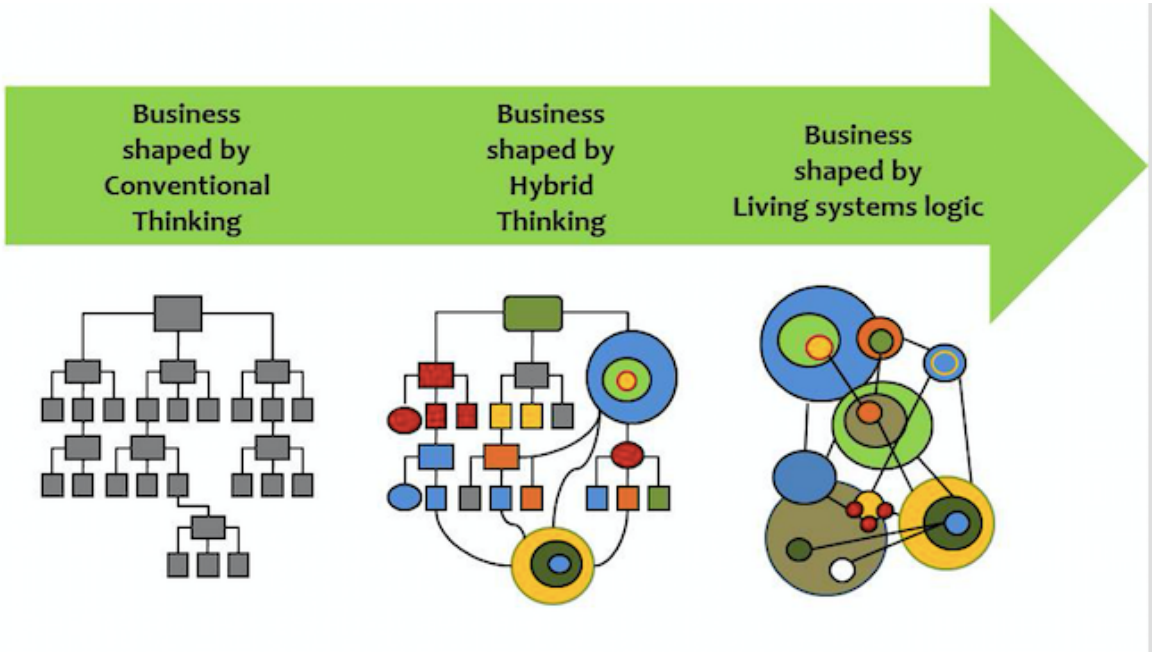


Figure 1: Non-hierarchical organisation model(Giles H. 2016)

Growth and innovation don't necessarily come from the people with lofty titles, but from the people who develop new ideas and execute them on a daily basis.(Haden J. 2014) The conventional organizational thinking was based on old hierarchical organizations, and nowadays in informational technology organizational thinking has moved towards non-hierarchical organizations.

This thesis covers product development metrologies such as Agile Scrum, Extreme Programming(XP) and Kanban. The company uses the mixture of these at the moment in Tori. Kanban would be ideal to a Startup type of operations, but the technical competencies are very spread at the moment. Best practices from Extreme Programming(XP) have been taken into usage. This covers agile stand-ups and sometimes pair programming. Agile Scrum way of working is used when tickets are created to our R&D ticketing system. Kanban board is used for agile visualization. We cannot work in real Kanban model since competence areas are too wide. It would be ideal that everyone could work in the full-stack mode, but it’s not realistic. The thesis describes these issues in more detail this in Chapter 4.

1.1 Target

Tori.fi belongs to global Schibsted Media Group, which is a Norwegian media group with operations in 29 countries, the most important being Norway and Sweden. The company has its headquarters in Oslo, Norway, and is listed on Oslo Stock Exchange. The company has around 7000 employees.(Schibsted Media Group Homepage, 2017)
Tori way of working is based on Lean-Agile, fast thinking and lean start-up way of making things happen. Tori.fi is a very popular online classified site in Finland. It has approximately 750 000 visitors weekly in the year 2015. Company internally researched that every third Finnish people use Tori.fi service weekly. (Tori Media Homepage, 2017)

1.2 Purpose of the Thesis

The purpose of the thesis is to implement the experimental model and move part of the Tori.fi services to AWS cloud. Amazon Web Services provides a highly reliable, scalable, low-cost infrastructure platform. (About AWS, 2016)

The important issue to determine whether the site is operational with known latency increases. Also, the Cloud Scalable Infrastructure and Software Architecture play an important role when moving the services to AWS.
The purpose comes from the Schibsted Media Group's local management team decisions to move toward cloudification.

In this chapter is described the main purpose of this thesis work. One of the biggest purposes of this thesis is to understand, if we can reduce the costs of hosting the services from the cloud. It is difficult to calculate forehand accurately since we need to think of maintenance required costs as well. Unlike on-site hosting the price of deploying the application in the cloud can and should be less low to hardware costs.
The Universal Access play important role in cloud computing, which allows employees remotely located to access servers and services easier.
Software upgrades from cloud provider also will generate more dependency and elasticity, since there will no longer hardware dependency. Cloud makes possible fast and easy scaling and possibility to acquire more service resources instantly. Cloud services are elastic, which means new resources can be added with few mouse clicks.

At the moment Tori's service costs running costs are divided into maintenance of the servers and operating costs of the servers. The goal is to create automated systems, where there would not be a need for old fashion way of system administrator services. The purpose is to move towards DevOps direction, where traditional sysadmin services would not be needed. Server hosting requires at the moment hosting facilities from local

internet operators, hardware servers, fiber optics, switches, and hard-drives. The purpose is in the future to get these services from the cloud environment. Also, the competence needed for the maintenance is not needed in-house.

The second goal regarding the transformation is security. High scalability gives us an opportunity to fight back against security threads such as Distributed Denial of Service(DDoS) and a help of continuos automated software updates we minimize the risk of having insecure software components.

1.3 Stakeholders

This AWS Cloud project changes the model how system infrastructure is managed in the future. All the developers notice changes how the software will be released. It will affect indirectly to the software development models as well. It results affect management team decisions and product development. Also in short term, the cloudification project employs the local team why all the local improvements will be postponed.

The stakeholders of the Lean model is all the employees in the company, since the overall the company works around the lean model principles. Mostly thesis concentrates on the research and development part of the models. Some areas are very special and concentrate only for employees working in the Product and Technology business unit.

Cloud project affects also how the Quality Assurance processes work in the future. The stakeholders for this project are business process owner, product owners, software development, testing and customer service.

As we are researching the Proof-of-Concept cloud model for current infrastructure the results could be valuable for the Media Group as well. The results could be shared with several countries facing the same problem.

1.4 Research Objectives

One of the main objectives is to study the suitability to move the service to AWS. (About AWS, 2016). The important issue is the Website page load time latency. Moving Tori's services to cloud it's experimental. The decision for moving the services can be made if the Website loading latency challenges can be defeated. Website loading time and latency play an important role of the popular and successful Web sites such as tori.fi. There are also several different kind challenges for instance if we are able to move databases with clusters to the cloud.

The second objective is to investigate how the Lean Koju model works in the company. Tori organization is operating in Koju model. The Tori Koju is Finnish word at it means marketplace stand. Tori's Koju model has similarities to e.g. Spotify Tribes. Actually, Kojus could be understood as Spotify Tribes. It's agile engineering culture which could be referred as well to company culture. Kojus could be considered in other terms projects that have independence in the operational model. In IT companies tends to rapidly improve their IT services and methods. Therefore Lean principles for the development are ideal. In the thesis, I am investigating the improvement of the flexibility in a Lean environment, and how it effects to work performance.

1.5 Scope

The AWS cloudification project has been experimental and is scope is to study whether the Tori.fi-service can be operated in AWS as a cloud service. Another scope of this thesis is to explain the lean working model that was taken into use recently. Research questions include questions how the Lean working model has affected to flexibility and working efficiency.

There are several advantages moving services to the cloud. The biggest known issues are related to website loading time latency.

It's known fact that Website page load time loading time increases estimated around 20 - 30 ms when services are moved from Finland to Ireland. Currently, Tori's infrastructure contains around approximately 80 virtual servers that will be moved to the cloud with new system architecture. It is experimental why it is expected that unexpected and unknown issues will arise during the project.

Website page load time is an important factor and important Key Performance Indicator(KPI) within the Website companies.

There are ways that we can improve the User Experience with software architectural changes, and minimize the cloudification delays. There are plenty of tricks that can be done to get the user experience better even we have more loading latency.

The scope is also to describe the development model that is used in Schibsted Media Group, Finland. It's also possible that in the future AWS opens data center closer to Helsinki e.g Stockholm, which will minimize the Website latency.

1.6 Research Questions

The research problem was to find out whether in the Schibsted Media Group (Tori.fi) it is possible to give up hosting dedicated servers and move the services Amazon's cloud using Amazon Web Services components.
The main objective of the thesis was to produce Proof-of-Concept knowledge for transferring the services to AWS Cloud. Also, one goal was to find better solutions to improve practices and services.
Also, one goal was to find better solutions to improve practices and services. In addition, the aim was to find out what kind of Lean development model was used for the transformation.
Schibsted Media Group, tori.fi moved to Lean model. Lean methods are excellent when producing experimental Proof-of-Concept models. This work explains the working methods and concentrates on studying how the work has changed in the Lean model.

Research questions:
Study the suitability of the AWS Cloud solutions due to Website page load time latency

Question 1: How the latency effects to Website traffic?
Question 2: How the end user latency is measured?

Improvement of the flexibility for the employees to work in Lean environment

Question 3: How the flexibility for working improved in Lean?
Question 4: How the flexibility is measured?

Improving working efficiency in Lean Agile-Model

Question 5: How the work efficiency will be measured?
Question 6: What factors were used to measure work efficiency

This thesis work concentrates moving the services to AWS Cloud and the Lean model that was in use during the project.

The first pair of research questions (1-2) concentrates on the fact that Cloud services provide latency to the service. That can be problematic if the end users of Tori.fi are not willing to accept the latency, and are willing to change to another service. The latency measurement can be measured from the server to provide the content. The users will necessarily notice this change since there are latency as well with the Internet Browser renders the visible internet pages. All the content download transfer on the page is not started parallel at the same time, and it’s affected by Web browser rendering. There are differ-

ences in the speed of rendering pages between different internet browsers e.g. Internet Explorer, Opera, Firefox, Safari, and Chrome.

The second pair of research questions (3-4) concentrates on working in the Lean environment. The question is how the flexibility working in Lean has improved. In the Lean model, there is more freedom for the employees to select the working methods, and plan continuously improve what we will be implemented.

The third pair of research questions (5-6) concentrates on the work efficiency in the Lean model. The question is how the work efficiency will be measured, and what factors were used for measurement. A lean model gives freedom, and not so accurate monitoring how the tasks are carried out. A question is how this affects to work efficiency, and how it can be measured.

2. Research Methodology

The used theoretical frameworks and research methodologies are discussed in this chapter.

This chapter covers the research strategy and methods that will be used for this thesis. A big emphasis will be put into additional constructive research to support the decision making process in moving the dedicated service hosted services to AWS. The empirical data is used to make assumptions.

The research methods used in this thesis for the Lean operational model is a case study. Methods that are used are qualitative and quantitative methods. By using these methods the goal is to find out how employees work in the model.

2.1 Strategy

The strategy for this research is constructive research and experimental case study. Constructive research is very common in computer science research method performing benchmark tests with the Proof-of-Concept prototype. (Lukka, K. 2000)

The strategy for the Constructive Research model is to set objectives and tasks. In this case, the objectives are to build up the Proof-of-Concept model for AWS Cloud. The process model is designed during the Architectural review of possible alternatives. There will be no waterfall model but we are using agile steps to move small components to the cloud.
The case execution is implemented by moving parts of the current component architectures to Cloud. The organization interviews are conducted asking their visions about possible events. Interviews are included in this thesis work. The Proof-of-Concept model works as a simulation model. There is backup plan to use old dedicated servers in the cases of failovers. The results of the simulation are interpreted immediately after switching traffic to Cloud. The data collection has been made and KPI and servers are being closely monitored during this actions. According to analysis, the feedback has listened, and customer support contacts are monitored closely. If analytics or customer support feedback gives negative results we will rollback the traffic from cloud to dedicated servers.(Lukka, K. 2000)

The cloudification project nature is also experimental case study why it's important to understand the effect of the known defect Website latency to the End Users behavior in order to provide good service.

The strategy is to measure and analyze company's internal KPIs(key performance indicators) of the analytics data and estimate according to that End User behavior. Most important KPI in this project is the Website loading latency and its correlation to other KPIs during the AWS Cloudification pilot project.

Another very close monitored KPI is the daily visitor count. This is a very important factor for business and if its get negative impact the project could be postponed, redesigned or terminated. Existing dedicated servers are kept parallel with the cloud project. It gives the possibility to rollback in the case of important KPIs measures negative impact.

The lean organizational model has become a foundation of many IT organizations today. This organizational methodologies and processes are examined in this work by case study. The related methodologies discussed in this thesis have been chosen for their relation to the improvement model, which is the theoretical framework used for this case study.

Methodologies that will be discussed in chapter 4 are Lean Koju-model, Organisation Flow, Lean Methodology, Lean Canvas, Lean Loops, Lean Experimenting and Lean A/B testing.

2.2 Data Collection Plan

Data collection is done with several tools that are used to monitor the Website latencies, which is the main KPI for this pilot project. The tools used to analyze the traffic are Zabbix Server Internal Monitoring(What is Zabbix, 2017, Appendix 2) and XiTI tool by AT the Internet. (XiTI tool by AT Internet,2017, Appendix 3)

There are several KPIs measured and monitored which are very typical for the Website companies. The KPIs measured will be first-time visitors, the bounce rate of the end user, visit quality conversion rate, page view per user, unique daily sellers, and new unique daily sellers.

The Tori.fi Website is monitored very extensively for user behavioural changes. The KPIs play a very important role when Schibsted makes investment decisions to markets globally. These KPIs play an important role since it's known from previous experience how the KPIs can be monetised.

Schibsted Media Group in Tori.fi is also using XiTI tool provided by AT Internet, which gives extensive information about the End Users behaviour for the site due to any change of the KPIs recorded.

3. Proof-of-Concept Cloud Infrastructure

The project started in Schibsted Media Group Finland approximately November 2015. The required information was gathered by participating to the Architecting on AWS training and colleagues working in the same project. This course covers the fundamentals of building IT infrastructure on AWS.

During this project, we have built small infrastructure components to cloud from the existing system infrastructure. These system infrastructures will be introduced in this chapter. The plan has been gradually moving all the infrastructure services to the cloud but in small steps. This decision was made in the beginning to use small step was to avoid risks. Amazon Web Services provide hot topic today's competitive market. Today Tori.fi is a hybrid model which means that it's partly on AWS (Amazon Web Services) and party on it's dedicated servers in Helsinki. This case study concentrates on transforming the Traditional dedicated server model to IaaS (Infrastructure as a Service). The long-term future model will be Micro-services model PaaS (Platform as a Service) where the plan is to concentrate only on developing software for e.g. Market Platform, and all the hosting and infrastructure work will be hosted outside. (Sarkar A., Shah A, 2015, Chapter 1, Cloud service models) Also reduction of the operating costs plays important role. It's assumed and estimated that system administration work, which includes maintenance of the infrastructure causes less work for server maintenance. Flexibility and elasticity are provided with server resource purchases.

One important issue is also reduction of the service interruption and downtime. The AWS reliability has been around the best for cloud vendors is "five nines" (99.999 percent). This can be improved using several Amazon availability zones. Currently, Tori.fi has been running with high reliability of around 99.9%. Mostly reliability has been caused by buggy software releases and not due infrastructure failures. The expectation in the future is that services from the cloud are as well fault tolerant.

An important objective is that will we gain improved system infrastructure with cloudification. Infrastructure as a service (IaaS) model provides third-party hosting for hardware, software, servers, storage and other infrastructure components. IaaS providers also host users applications and handle tasks including system maintenance, backup and resiliency planning. IaaS platforms offer highly scalable resources that can be adjusted on-demand. This makes IaaS well-suited for workloads that are temporary, experimental or change unexpectedly.

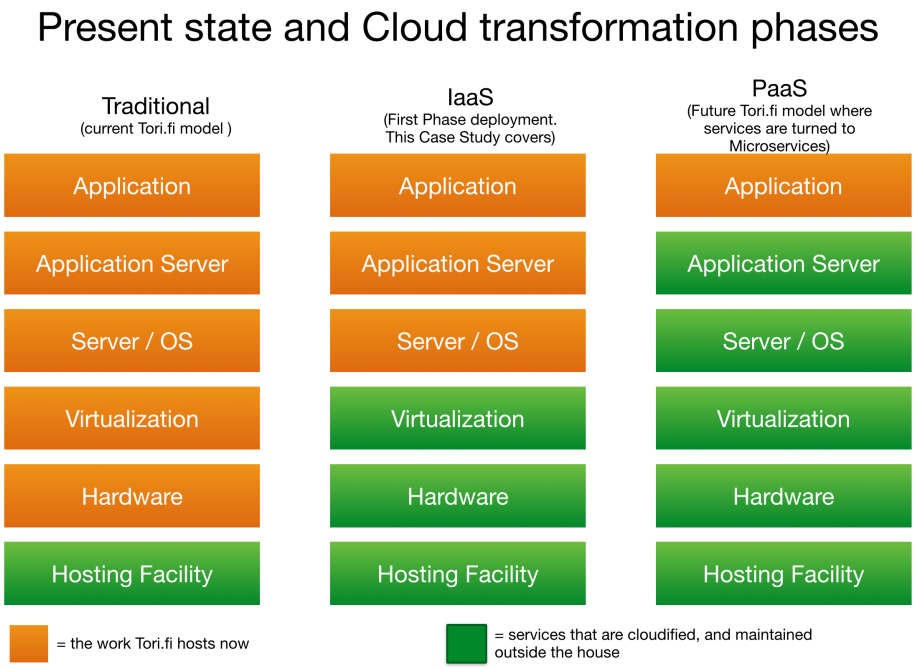


Figure 2: Present State and cloud

Figure 2 describes that at the moment Tori.fi is in the traditional model. We host all the servers ourselves and install operating systems to servers. IaaS (Infrastructure as a Service) model will be the next phase all the servers are in Cloud. All the current servers are moved to AWS EC2 instances. Schibsted plan for the future to move towards PaaS (Platform as a Service) model where application developers need to only write application code and all the unnecessary issues that core knowledge is outsourced to cloud vendors such as AWS. Spinnaker tool gives great opportunity in the future. Spinnaker is an open source, a multi-cloud continuous delivery platform for releasing software changes with high velocity and confidence.

It gives an opportunity to deploy to different cloud providers: AWS EC2, Google Compute Engine, Microsoft Azure, OpenStack and Cloud Foundry. (Spinnaker Documentation, 2016) Spinnaker provides cluster and deployment management. Spinnaker's cluster management features are used to manage resources in the cloud. Spinnaker's deployment management features are used to construct and manage continuous delivery workflows.(Spinnaker Documentation, 2016)

The future status is that we concentrate purely implementing software, and all hardware related services and task are outsourced to the cloud. (as it can be seen in Present State and Cloud, Figure 2).

The reason for this kind of change is that now for server maintenance is split to several people, and changes are slow.

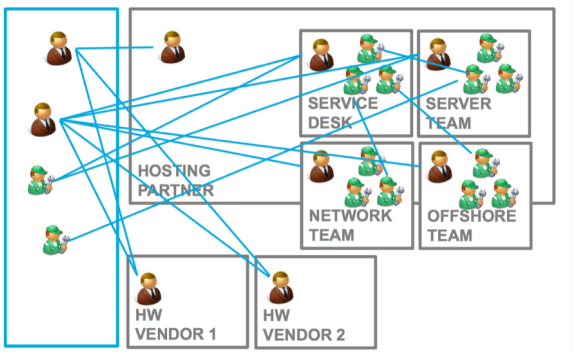


Figure 3: Server management at the present state

Currently, for the dedicated server facilities are split between two server facilities. And we have outsourced partly the hardware maintenance to an external company which is depicted in the figure 3. We need to always wait for their response to make changes there. These makes the cloud services in the future more cost efficient. Software developers do not need to maintenance servers and do not need to rely on others to generate rapidly software. We have made some cost efficiency calculations, but there are mostly random calculations where we cannot calculate how much costs if e.g. 10 developers will wait(lean methodology "wait"-waste) for the server changes to be able to implement the code.

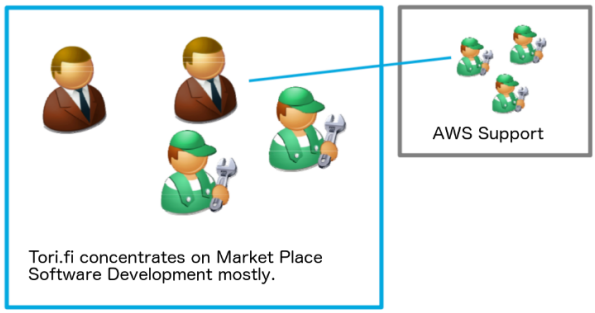


Figure 4: Future: Tori.fi concentrates on software development

At the moment we are dependable of another company to do the some of the server management tasks. The cloud-based model gives us an opportunity to concentrate purely on software development. Also using Spinnaker we not dependable of one cloud provider. The software development will be implemented using Agile development models and mainly their mixture Scrum, XP and Kanban. These models will be described later in this thesis work.

The security of the service maintains an important role as well. It will use AWS general Security features which allow generating as secure systems to cloud as to traditional service.

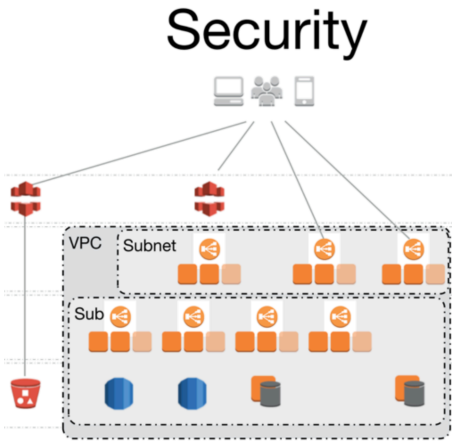


Figure 5: AWS generic security options

AWS contains VPC, Console, Resources, Network, Sever, Web Application, CDN-specific security option available. Security part is so extensive that it is left out by purpose from this thesis work.

3.1 Amazon Web Services(AWS)

The AWS Cloud provides a highly scalable and fault-tolerant infrastructure. The architecture allows deploying web-based solutions with minimal cost and administration. Also, it has more flexibility generally than in-house infrastructures or datacenter capabilities. (Vyas, 2015, Chapter 1, AWS Services) AWS has started contributing highly available infrastructure platform based on the pay-as-you-go model.

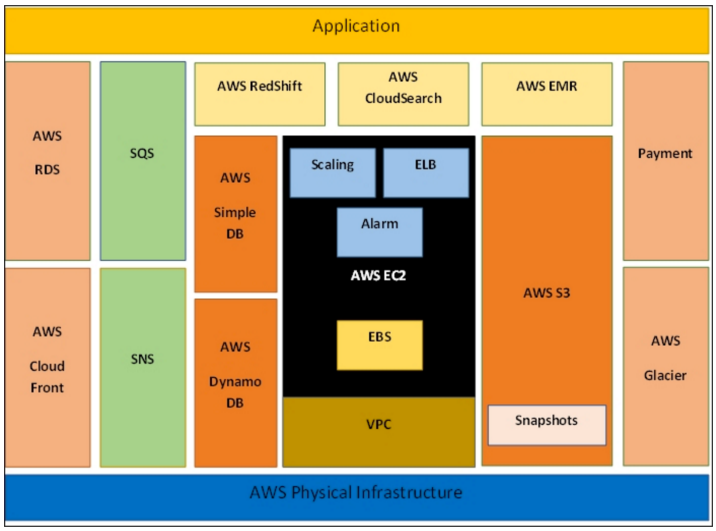


Figure 6: AWS architecture and services(Vyas U. 2015)

I am shortly defining the most important components used for Tori's AWS infrastructure.

Amazon Elastic Compute Cloud (AWS EC2) instances provide compute capacity from the AWS cloud. The operating system, application software, and associated configuration can be bundled into an Amazon Machine Images (AMIs). (Sarkar A., Shah A. 2015, Amazon Elastic Compute Cloud) Figure 6 shows that AWS EC2 are one of the main components of AWS cloudification. In Tori's case, most of the Xen Virtual servers will be transformed to EC2 instances with auto scaling possibilities. All the current Xen virtual servers will be transformed to EC2 instances. Auto scaling creates new AWS EC2 instances according to predefined rules. This gives flexibility for scaling services in Tori's case.
AWS auto scaling allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. Auto scaling helps to ensure that desired number of Amazon EC2 instances are running when required. Auto scaling can also automatically increase the number of Amazon EC2 instances. This is required during demand spikes and to maintain performance and decrease capacity during slow hours to reduce costs. (AWS Auto Scaling, 2016)

Amazon Elastic Load Balancers (AWS ELB) automatically distribute traffic across EC2 instances. ELB is used for fault tolerance which providing the required amount of load balancing capacity needed to route application traffic. (AWS Elastic Load Balancing, 2016) Figure 7 describes simple load balancing to EC2s.

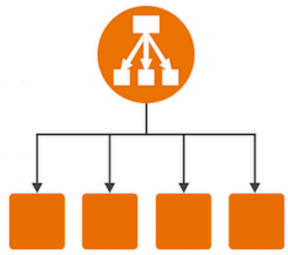


Figure 7: Simple load balancing(AWS Elastic Load Balancing,2016)

The Elastic Load Balancers were used in Tori's Proof-of-Concept model.

Amazon Virtual Private Cloud (Amazon VPC) is used to provision a logically isolated section of the Amazon Web Services (AWS) cloud where you can launch AWS resources in a virtual network that you define. They have used e.g. security reasons to limit traffic to certain database servers. In dedicated server world like Tori used several routing rules in the local area networks, and they could be thought to be "tiny" subnets in local area network giving access to have own IP ranges and configure route tables and network gateways. VPCs supports nowadays also IPv6. (AWS Amazon Virtual Private Cloud, 2016)
AWS S3 is Amazon Simple Storage Service from AWS Cloud. It provides reliable storage for data by providing redundancy. The users are charged for storage of data on S3 based on the amount of storage consumed. Download of data from S3 is also charged. Uploading data and transfer of data between AWS properties are free of charge.(Turkington G. Deshpande T., Karanth S. 2016, Chapter 9, Amazon AWS S3)
In Tori case, the image upload was initially designed to use the S3 directly. The biggest advantages were that S3 can scale automatically in disk space. Later on, we decided that we would rather use the Schibsted Infrastructure Image Servers providing images for the site. This has several advantages since the image server hosting will be taken care of separate infrastructure team. Their solution is as well based storing the data to S3.
AWS CloudFront is Content Delivery Network (CDN). Amazon CloudFront is a global content delivery network (CDN) service that accelerates delivery of the content of your website. Earlier in Tori, we used Varnish Cache, and we replaced it with AWS CloudFront. The static content is the type of data that is used for CDN mostly. Basically Tori's case it includes images, and for instance ads which content doesn't change on every page load.

FrontCloud integrates with other Amazon Web Services products to give developers and businesses an easy way to accelerate content to end users with no minimum usage commitments.(Amazon CloudFront, 2005, Chapter: Content Delivery Network)

3.2 System Architecture on Dedicated Servers

Tori.fi was hosted completely on dedicated servers when until the beginning of 2016, when the server infrastructure move to AWS cloud started.

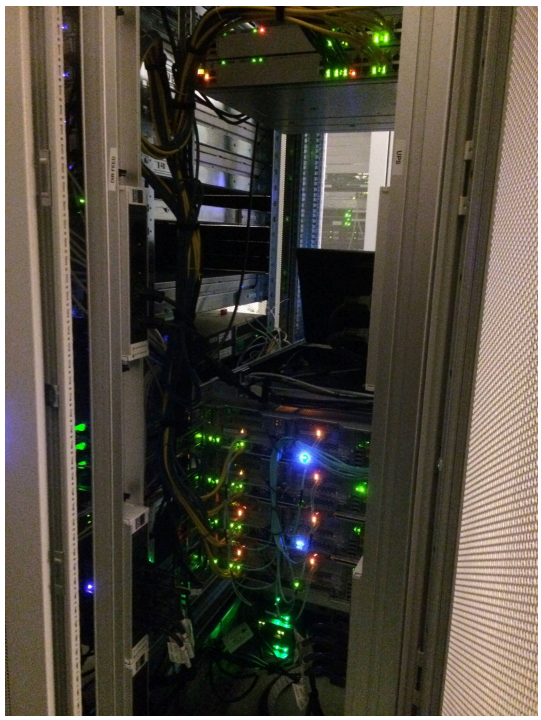


Figure 8: Tori's Physical servers in one datacenter

The servers are virtual servers build on top of the XenServer. For the current architecture, there are 4 physical servers per datacenter hosting Tori.fi service. There are two datacenter's used hosting the service. The project started slowly moving infrastructure towards cloud with a Hybrid solution. Currently, in hybrid mode source of the traffic comes partly from dedicated servers and from the AWS cloud server.

The current setup of Tori's System architecture is described here. Front-end servers at Tori's current infrastructure were split two logical of front-end servers that serve the presentation of data.
Front end part in Tori contains two kinds of Apache Web servers. Pre-fork and worker are two type of Multi-Processing Modules Apache provides. There are front end servers which execute C compiled template language in worker mode. The template language is compiled to executable C code which is extremely fast and efficient. On the other side, it's not very easy to read the code, and it's very primitive programming language.
Another Front End servers execute PHP code in pre-fork mode. (Helmke M. 2017, Chapter 24, Apache Multiprocessing Modules)
This division was made in a long time ago when the performance was considered as the most important part, and there weren't really available cloud architectures.

Front-end servers are considered in Tori as Template Modules executing Apaches, PHP servers, and image servers. Front-end servers and PHP and image servers are used to display the content of the Website. Back-end servers are considered as Data Access Layer.

Partly it is difficult to separate front-end code from the back-end since the currently the presentation part contains as well features that should not be in the presentation layer.

This is due that in some phase it was considered that all the back-end codes would be as well written with template language.

Generally, transaction server, search engines, Postgres database and NonSQL database as Redis are considered to be a back-end of the site.

I will briefly explain the back-end server purposes in the current Architecture.
Transaction server: Contains access layer to the database. Transaction server works as connecting layer to database access.

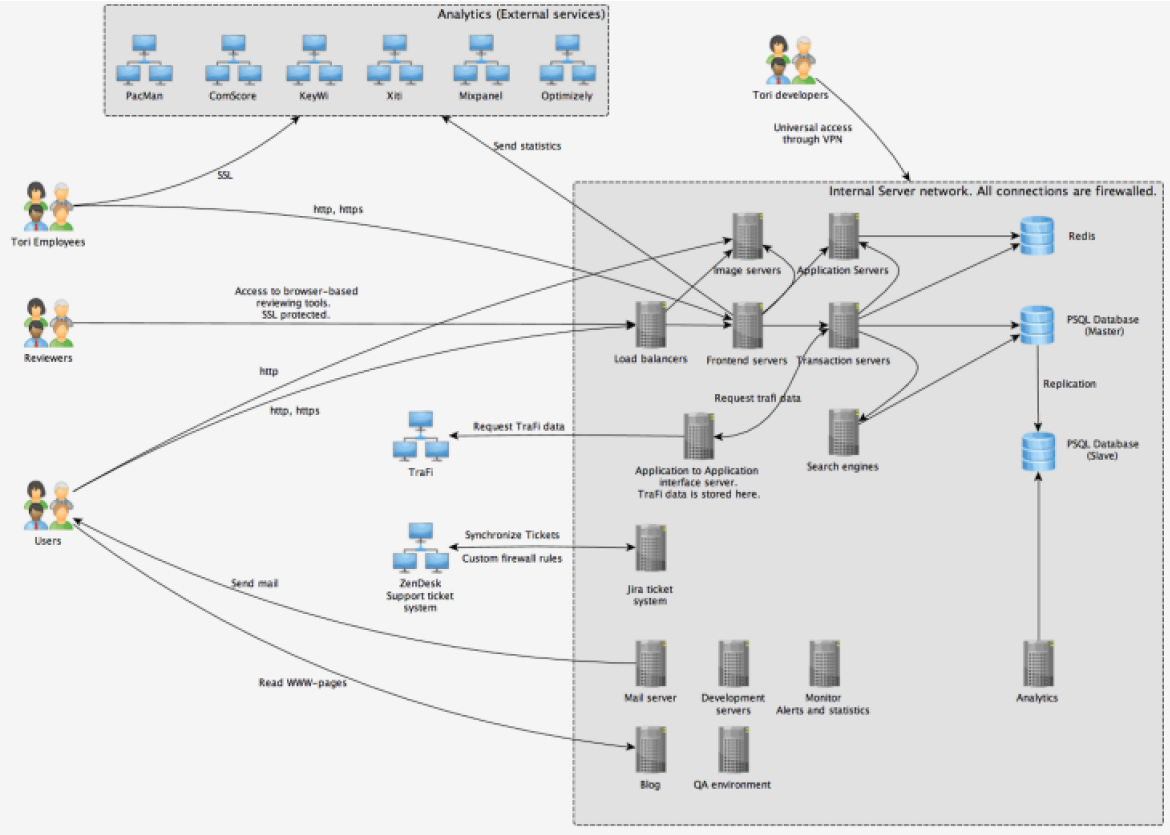


Figure 9: System architecture overview

It prioritizes how the transactions are executed. It has a secure layer that prevents unauthorized access. They are mostly written with C-language. There is factory transaction created for easy way implementation of it.
Search Engines: Contains data with fast access. All the ads in Tori are indexed by Search Engines. Data synchronized with the database to Search Engines with Indexers. Data is organized to search engine such way that it fetched there much faster than from Database.
Postgres SQL: PostgreSQL is a powerful, open source object-relational database. It has been around more than 15 years of active development and a proven architecture that

has earned it a strong reputation for reliability, data integrity, and correctness. (Postgres SQL 2017, Official Website)

NoSQL: Two types of NoSQL are used: Redis and Memcache. Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache and message broker. (Redis Homepage) Memcache is Free, open source, high-performance, distributed memory object caching system, generic in nature, but intended for use in speeding up dynamic web applications by alleviating database load. (Memcache Homepage, 2017) Redis service improves the performance of web applications by letting you access information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases. NonSQL databases are very popular today since they make performance driven Web application development easy.

3.2 Proof-of-Concept Architecture

First AS-IS transformation approach is that we will move first all the virtual servers from the XenServer virtualisation environment to AWS. This will be the Proof-of-Concept Architecture for the Cloud. The architecture would be similar as in XenServer, but build up with AWS components.

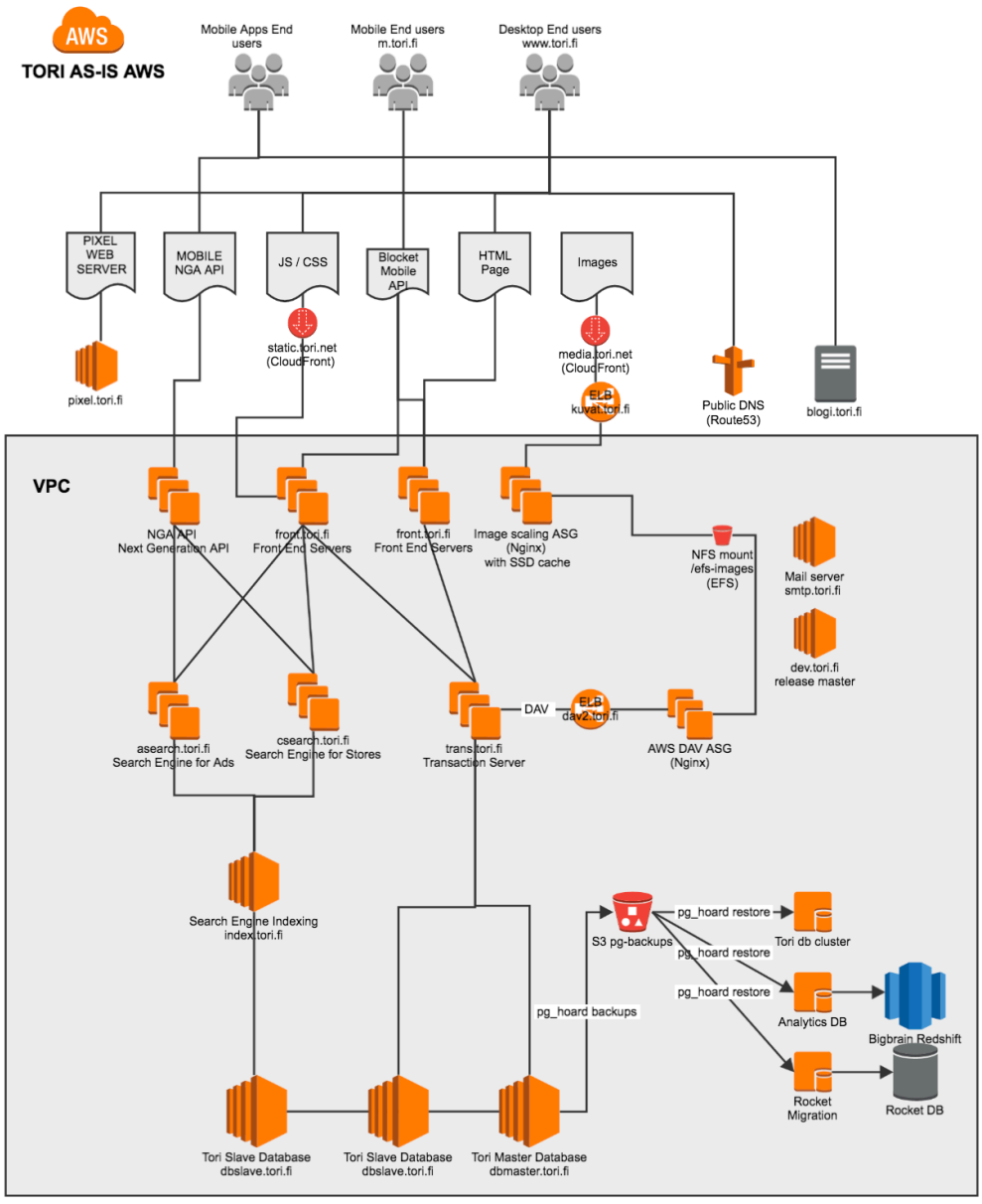


Figure 10: First approach to Tori's AS-IS transformation to AWS

This is the plan for first phase transformation, but it's not the intention of the cloudification. The current architecture is not scalable to the desired AWS typical auto scaling architecture. Therefore we first do the AS-IS migration before we plan the new way of deploying the software to cloud. Also, we need the support for Spinnaker deployment which requires software architecture changes.

In the first phase AS-IS migration we have plans to move the Redis instances to Amazon ElastiCache. ElastiCache supports Memcached and Redis in-memory caching solutions. Amazon ElastiCache is a web service that makes it easy to deploy, operate, and scale an in-memory data store or cache in the cloud.(Sarkar A., Shah A, 2015, Chapter 3, Elastic Cache) Tori has currently eight(8) different Redis instances running on the present infrastructure. Redis is used as NoSQL fast storage for parameters used in the code. It's used to store information what needs to be fetched quickly. It's really fast compared storing information to the relational database.

First AS-IS migration we would transfer the Postgres SQL databases into EC2 instances.

The plan is to move the database in the future with AWS Amazon Relational Database Service(RDS). Currently, there are built-in Postgres modules installed with RPM's that are installed to Postgres SQL servers. Amazon RDS doesn't support external modules to be integrated.

Amazon Relational Database Service (Amazon RDS) provides an easy way to setup, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks. It supports as well several databases: MySQL, Oracle, SQL Server, PostgreSQL (Sarkar A., Shah A. 2015, Chapter 3, Amazon RDS)

3.3 Tori.fi as Hybrid Cloud Architectures with AWS

The current project has progressed so that we have moved the Domain Name System (DNS) and content delivery network (CDN) services to AWS already.

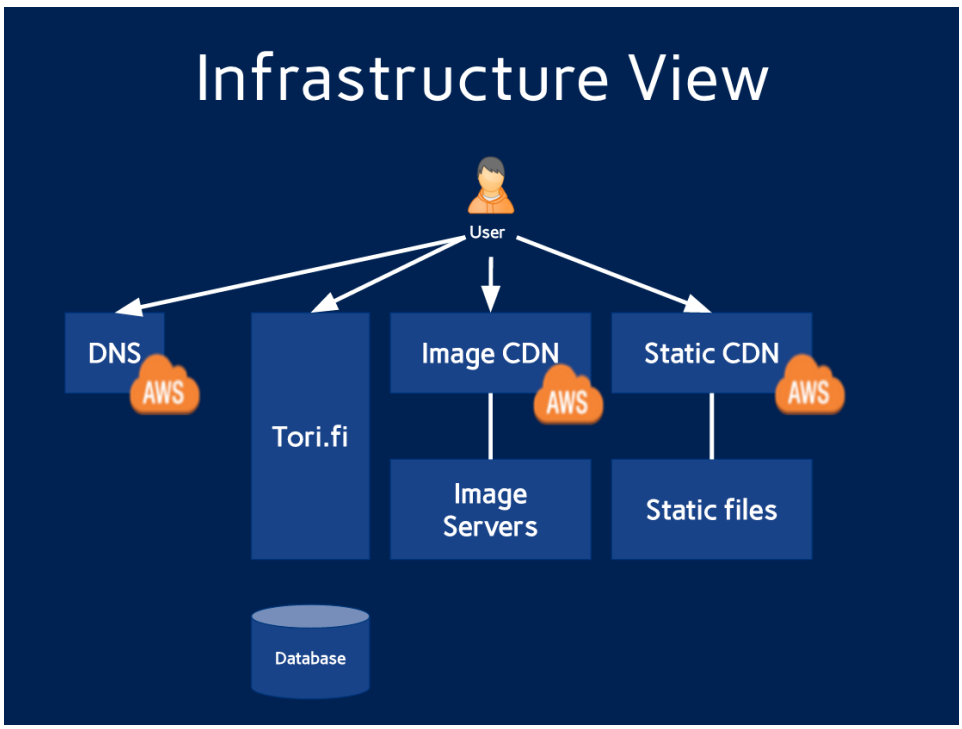
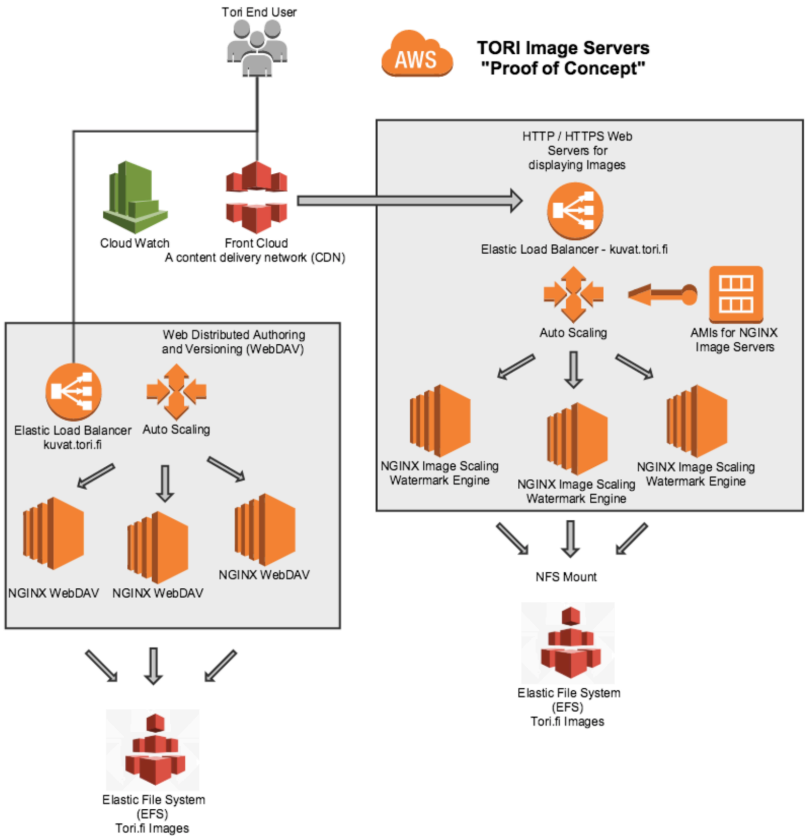


Figure 11: **Current AWS hybrid model**

Currently, we have moved the AWS services to Cloud in gradually. It's safer to move the services to Cloud in small steps. In the first phase, we used Lizard project to move the services to the cloud. Unfortunately, the Schibsted is not using that project anymore, and that is why we planned to abandon the plan using it.
Currently, we have switched image cache to Amazon CDN. We planned to shut down Varnish servers from the dedicated servers. Also, the DNS services have been moved to AWS. The next phases for the that we are working is to build up reliable replication of databases to AWS. We will start working moving the image servers to Cloud. Also, we need to convert the XenServer virtual images to AWS EC2 instances. It's not decided yet will we used in the first phase the Spinnaker deploying artifacts to the cloud. This will change the nature of the project since we need most likely software architecture changes.

3.4 AWS Proof-of-Concept Image Servers

Building up new image servers was also one interesting concept of this thesis. The first version was implemented with Linux S3FS, which is a FUSE filesystem that allows you to mount an Amazon S3 bucket as a local filesystem. (Fuse Over Amazon, 2017)



We implemented the first version with S3FS and stored the images to AWS S3. During the implementation, it was realized that AWS offers for this purpose new kind of service Amazon Elastic Filesystem(EFS). It was natural to switch there, since if we would counter technical issues we could contact AWS Technical Support.

Figure 12: **Architecture of Proof-of-Concept image servers**

Figure 12 describes the logical Architecture for the Proof-of-Concept WebDAV image server. This Proof-of-Concept version of the image server was tested for operation for two weeks. There were some synchronization problems with the images, which was mostly caused by the platforms DAV protocol handler which is written to the platform. It is possible that we continue using this with proxy DAV server, which reflects data to both old and new image servers. Also during the implementation of this Schibsted Global infrastructure team has to build general image server solution. It's very likely that we will take that solution to part of our image servers.

3.5 Conclusions and Summary Of The AWS Proof-Of-Concept Infrastructure

During this project, we have built small infrastructure components to cloud from the existing system infrastructure. The plan has been gradually moving all the infrastructure services to the cloud but in small steps. In that sense, we are working in an Agile mode in building the system infrastructure to cloud. If we would have worked in waterfall model then we would have deployed all the services to the cloud at once. The decision was made in the beginning to use small step was to avoid risks.
As described in the chapter we have moved the DNS service to AWS Route 53. We have moved the static content to AWS CDN. We have the build the Proof-of-Concept Image server solution which was introduced in this chapter. We have described the plans to move the current infrastructure to cloud and depicted the AWS instances needed for that purpose. During these Proof-of-Concept transformations, we have proven that e.g. the latency is not the problem moving the services to the cloud. Already in the current Hybrid model, we are having the same the latency effect as all the services will be running in the cloud. During the project, the competence for AWS services has improved while transforming the services there. We could have moved the services with old structure to AWS already earlier, but we have decided to continue and improve our in a Lean way our infrastructure architecture so that is more cloud-like. This means the issues defined in this chapter e.g. taking the auto scaling into use for EC2 server instances. The plan has been also to release the software in more cloud way e.g. Canary Releases and with Blue Green deployments.
I feel we have been successful in making the Proof-of-Concept model. It would require still more work to implement the changes. Currently, Schibsted is in parallel developing new Micro-service (Newman S. 2015) based architecture for the Market place platform. Naturally, that will change the scope of this project as well in the future.

4. Lean Agile Model in Organisation

Tori used to work in the beginning in Extreme Programming(XP) methodology. (Crookshanks E. 2015, Chapter 4, Extreme Programming)
Later on, the model was switched to Agile development model. As the company continued growing in engineering people it was decided to take into using more Spotify Tribe way of working that we call Lean Koju model. In Koju model employees don't need to work according to the certain predefined Agile rules.
Tori have several business units e.g. Marketing and Communications, Growth-Insight, Sales operations, and Product development and technology.

Each business unit has its own line-manager. Mostly all the development is carried out in Lean Koju model. Tori were split into several business units which are described below.

Marketing and Communications include advertising, direct marketing, branding, online presence, printed materials, and PR activities. Marketing communication channels focus on the way a business communicates a message to its desired market or the market in general.
Growth-Insight includes analytics team, Big Data team, fraud management is handled in this team. Growth-Insight team's mission is to increase traffic to the website through multiple online marketing channels (Facebook Ads, Apps Installs campaigns, Google Adwords) and retention (Push notifications, emails, and retargeting).
Sales operation is a set of business activities and processes that help a sales organization run effectively, efficiently and in support of business strategies and objectives.
Product development and technology carries out product designing, UX designing etc.
Technology includes software front- and back-end development, testing, and QA assurance.
Finance generally includes budget, salaries and Koju budget allocation. It's responsible for knowing the status of the cash flow and reporting.

4.1 Lean Koju-model Explained

Tori organization is operating in Koju model. Tori's Koju model has similarities to e.g. Spotify Tribes. (Tracey E. 2015) Kojus could be understood as Spotify Tribes. It's agile engineering culture. Kojus could be considered in other terms projects that have independency in working methods.

An organization that is planned to achieve a particular aim generally works in several parallel projects. Generally, projects require people from several business units e.g. Sales, Growth-Insight and Product and Tech. In the organization, people are organized to Koju teams. It's CEOs and management role to set strategy and decide what kind of Kojus are needed. In other terms, Kojus could be referred as projects, but compared to projects Kojus are independent in decision making. They tend to work flexibly in a Lean way. Figure 14 depicts example Kojus that are set to the organisation. Koju is Finnish word and means individual stand on a marketplace.

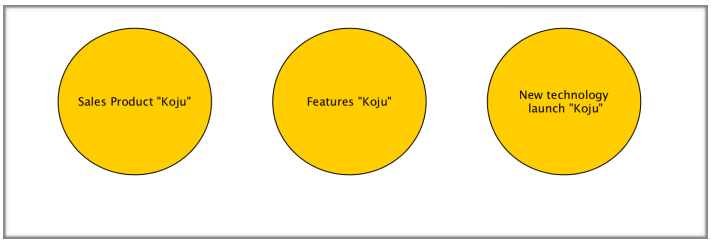


Figure 14: Examples of possible Kojus

Kojus are self-organising groups with sets of skills team together with the purpose developing a product or a feature. Kojus have minimal dependencies with each other and operate as autonomous units. They are allowed to make independent decisions, which doesn't need to be with a line each other. Good common practice is to use existing Agile methodologies and development tools. Kojus are not eternal, and they reflect the strategic priorities in the short to term. Once Koju has met its purpose then it can be set to standby.

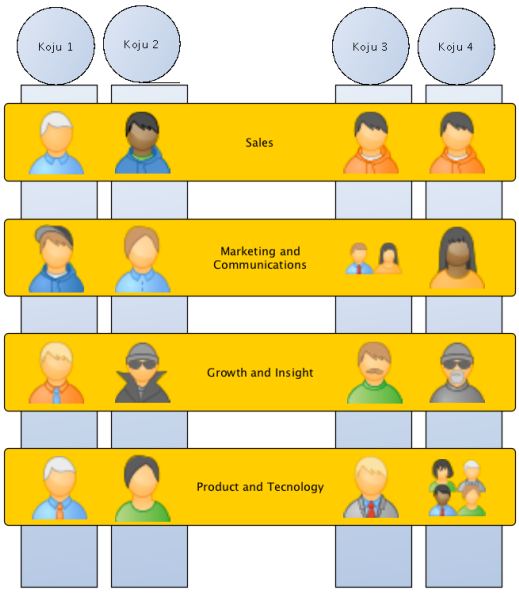


Figure 15: Example of Koju-model organisation

Kojus are meant to avoid wasting resources thanks to the use of lean methodology tools.

Normally one Koju team has approximately 6 persons. All the employees are not working directly in the Kojus. For instance, customer support, fraud management, and part of the analytics team are not working in the Kojus. There are some overlapping with the Kojus since e.g. special technical knowledge requires some people to operate in parallel Kojus at the moment.

Kojus are responsible its resource allocation and decision making independently. Every Koju has its autonomy how it can work lean model the best suitable and most efficient way. Also, Koju has complete freedom, and it's not forced to follow any general development Agile models since Kojus allowed to make autonomous decisions.
We used to work more in the hierarchical organizational model earlier, but since the development processes are wasting a lot of time our management team wanted to split organization to rather work around smaller teams or Koju projects. (This is a modern way to operate and it took its root from Spotify development model.) Also working in the small teams there is minimal micromanagement needed to carry out projects.

Kojus have few types of roles. The first type is Koju Sponsor who sets the overall targets for selected Koju. The roles main task is to support Koju and clarify the opportunity for the Koju and ensures that Koju has a growth environment available. Koju sponsor makes sure that product development is focusing right tasks. He or she prioritizes tasks and removes tasks that are not important. Koju Sponsor communicates product vision with the management team.

The second role is Koju Owner. Koju Owner creates a roadmap for Koju, and makes sure that operations are in line with parent company's (Schibsted Global) product roadmap. Koju Owner is responsible for overall business case and confirms that Scrum backlog is being prioritized through Jira. This includes as well communication and prioritization with scrum master. Scrum master ensures that the reflection and evaluation are carried out frequently for the Koju members. Sets Key Performance Indicators(KPIs) and makes sure they are shared to Koju members.

The third role is developers role. A software developer is a person involving with software development process including the research, design, programming, and testing of software. Developer participates retrospective, sprint planning, and planning poker. The developer is responsible for actual implementation of user stories.

4.3 Lean Organisation Flow

The organization flow is depicted in Figure 16. CEO + management team decides on the strategy. They set up several different Kojus, which tries to archive strategic goals.

One example of the Koju could be monetization team. Koju's task would be to implement features that will make revenue for the company. This team is independent, and they can generate new products using "Lean"-loop model that is explained later in this document.

Operational flow explains that CEO and management team setups strategy. Kojus implement the business needs. Kojus lean loop the ideas, and with several iterations, and produce the strategic goals.

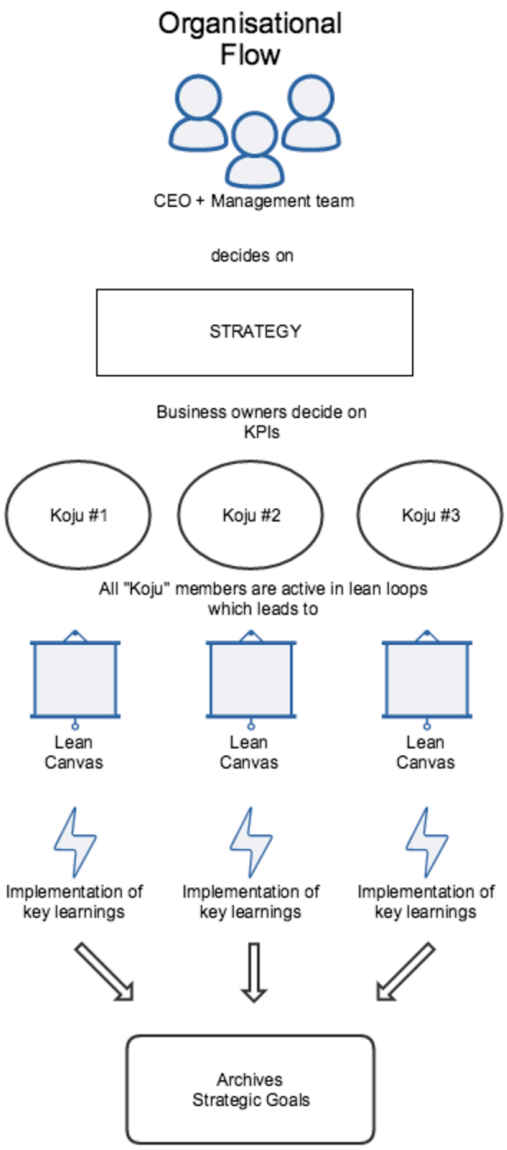


Figure 16: Organisation flow in lean model

Also in Tori we are using a term as "Hoshin". It is strategic planning ensure that the mission, vision, goals and annual objectives are communicated throughout an organization, and implemented by everyone from top management to the bottom level.
Management organizes business meetings where the strategic goals are presented.
Tori's management is responsible for implementing "Hoshin" plan. According to that plan the Koju models are established.

4.4 Lean Six Sigma

Lean Six Sigma is a methodology that relies on a collaborative team effort to improve performance by systematically removing waste. Lean Six Sigma's principle is to eliminate the eight kinds of waste (Muda): Transportation, Inventory, Motion, Waiting, Overproduction, Over-processing, Defects, and Skills. (Bell S., Orzen M. 2011, page 284)

A lean idea in Tori is to create more value for customers with fewer employee resources.

The goal of Tori's lean organization understands customer value and focuses its key processes to continuously increase it. The ultimate goal is to provide perfect value to the customer through a perfect value creation process that has zero waste. (What is Lean, 2017)

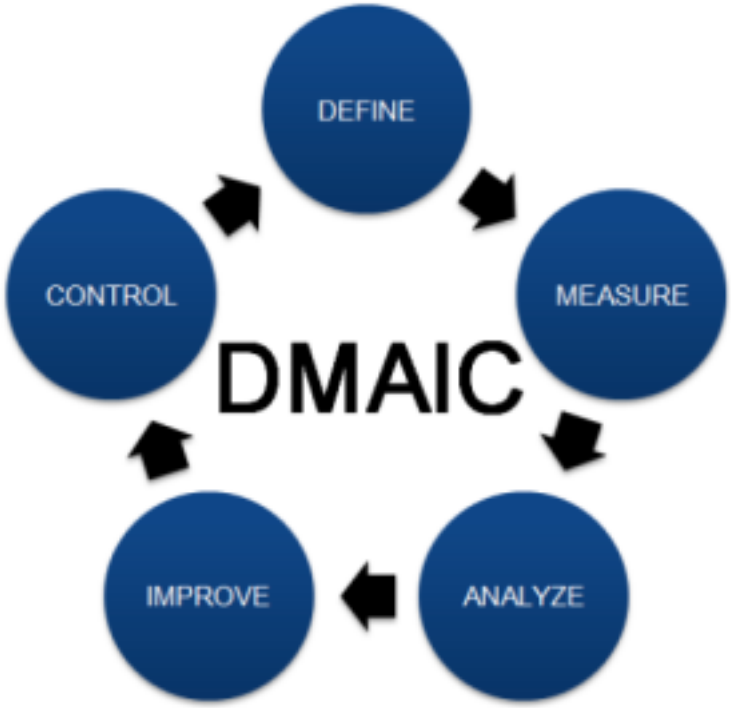


Figure 17: The DMAIC project methodology stages(Six Sigma, DMAIC Methodology, 2017)

The DMAIC project methodology comes from five words: define, measure, analyze, improve and control.(Six Sigma, DMAIC Methodology, 2017)

Define parts purpose in Figure 17 of this step is to clearly the business problem, goal, potential resources, project scope and high-level project timeline. In this lean AWS server move project the business problem was that are we able to move the present service reliably to AWS cloud. This project had quite limited resources since there was another prioritization ongoing for other Koju's as well. The project timeline was not tightly decided since the project was experimental.

The plan was to build a Proof-of-Concept model of the current software.

Measure part in Figure 17 determines how the process currently performs. Look and research for what might be causing the problem. Create a plan to collect the data. During the project, we collected plenty of data. The results are described in this work in the Analysis part chapter 5.

Analyze part in Figure 17 concentrated closely examine the process. It generally visually inspects the data. Brainstorm potential causes of the problem. Verify the causes of the problem. In the analyzing part, we, for instance, were inspecting the measurements for the website latency, and it gave references that we are able to move the service to cloud with disturbances.

Improve part in Figure 17 is used for brainstorm solutions that might fix the problem. Select the practical solutions. Develop maps of processes based on different solutions. Select the best solutions. Implement the solutions, and measure the improvement. For instance, we came to conclusion that we are able to improve the latency with Front End improvements so that it doesn't make much of the change for the User Experience.

Control part in Figure 17 concentrates continuously improve the process using Lean principles. Ensure the process is being managed and monitored properly. Expand the improved process throughout the organisation. Apply new knowledge to other processes in your organization. Share and celebrate your success.

4.5 Lean Canvas

Lean Canvas is mainly used when looking for new business opportunities.(Varma T. 2015, Chapter 3, Section 16)

Lean Canvas is used to "getting to know the path". Lean canvas is the first step towards testing whether the developed idea has the potential to be a viable business needs or not. The Business model canvas is specifically targeted for entrepreneurs and start-ups to help them account for uncertainty and risk. Most of the startup companies fail from (lack) paying customers. Lean Canvas will help map out the problems product should address, measure progress by identifying KPIs, define competitive advantage, define a product as a solution to the customer's problem. It is ideal to fill it thoughtfully since it will save time and money in the future. Lean canvas should be altered dynamically, add improved during its usage. Lean Canvas is very important tool nowadays in Koju teams.

Figure 18 describes the example of the Lean Canvas. It explained with imaginary problems to explain how it can be used.

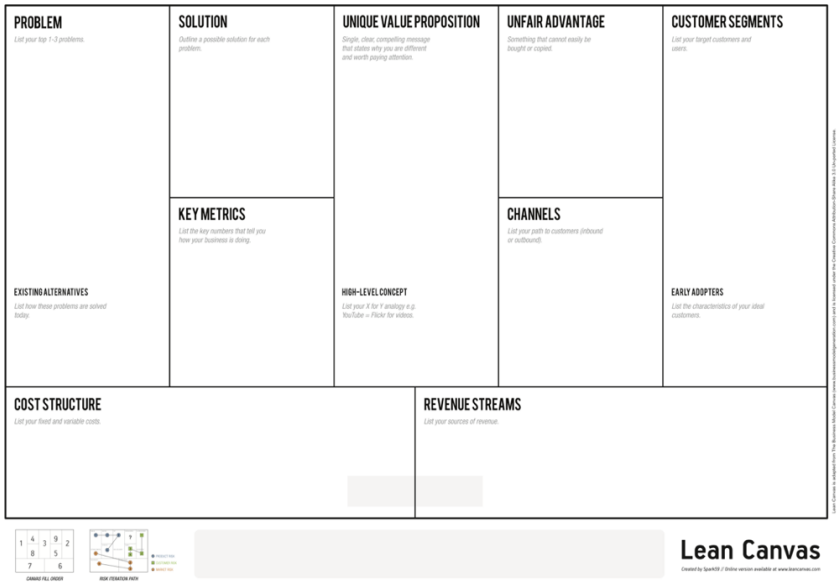


Figure 18: Lean canvas used in Tori(Maurya A. 2010)

Lean Canvas and relevant metrics are presented here. Lean Canvas contains following areas: Problem, Solution, Unique value proposition, Customer segments, Channels, Unfair advantage, Revenue streams and Cost structure. (O'Reilly Media 2016, Designing for Product Strategy)

Problem part describes the respondents who have this need, respondents who are aware of having the need. What is the crucial problem faced by Tori customers? In the cloudifi-

cation project this it is understood that website loading latency might traffic lost and therefore cause revenue drop in advertising amount.

"Solution part describes the respondents who try the MVP, engagement, churn, most-used/least-used features, people willing to pay." (O'Reilly Media 2016, Designing for Product Strategy) What is the solution to Tori's customers problems? Present the defining elements of your service: what makes it the top tool for addressing consumer needs? In the cloudification project, this could be implemented as front end solution that hides the latency effect from the end users.

"Unique value proposition part describes the feedback scores, independent ratings, sentiment analysis, customer-worded descriptions, surveys, search, and competitive analysis." (O'Reilly Media, 2016, Designing for Product Strategy) This message should explain what you do, how you are different, and why you are worth investing in. What you promise is to Tori's customers? The value proposition is innovation, service, or feature intended to make a company or product attractive to customers. In the cloudification project, this could for instance optimising the front end speed to gain more speed for the service.

"Unfair Advantage part describes the respondents understanding of the UVP (Unique Value Proposition), patents, brand equity, barriers to entry, the number of new entrants, exclusivity of relationships" (O'Reilly Media 2016, Designing for Product Strategy) How do you stand ours from competitors? What puts you ahead of the pack? Why should consumers have confidence in your service above others? In the cloudification project, this could be that for instance, Amazon changes the prices of the cloud services, and we are stuck with the service without able to change it when the implementation is done.

"Customer Segments part describes the how easy it is to find groups of prospects, unique keyword segments, targeted funnel traffic from a particular source" (O'Reilly Media, Designing for Product Strategy, 2016) Who can you help? In the cloudification project, this part includes for instance seasonal performance scaling. For instance the biking season creates lot of traffic to the site, and it can be easily scaled from cloud to end users.

Early Adopters part describes the specific characteristics of the early adopters. In the cloudification project this could give competitive advantage for the competitors who are not using the e cloud solutions to run the services.

Key metrics part describes how will you track consumer engagement and usage of your product.

Channels part describes the leads and customers per channel, viral coefficient, and cycle, net promoter score, open rate, affiliate margins, click-through rate, PageRank, message reach. How will you interact with consumers, inform them about developments and services? Print ads, social media platforms, promotional events.

The cost of Structure part describes the fixed costs, the cost of customer acquisition, the cost of servicing the nth customer, support costs, keyword costs. What will it cost to launch and maintain your business? Consider each stage of the start-up from creating a website and acquiring users, to hiring employees and producing goods, to marketing products and getting them to consumers.

"Revenue streams part describes the lifetime customer value, average revenue per user, conversion rate, shopping cart size, click-through rate" (O'Reilly Media 2016, Designing for Product Strategy) What monetary sources will fuel your company? How will you generate income? Present a pricing model or service, and then highlight other sources of revenues, ad sales, subscription fees, or asset sales.

4.6 Lean Loops

Schibsted Media Group Finland hired Moves the Needle consultants from the US to give consultation for working in Lean.

One of the consultants Eric Ries has written book called The Lean Startup, which is very popular. (Ries E. 2011) Brant Cooper Co-founder of Moves the Needle was visiting Finland as well in the consultation session. Brant Cooper has also strong background with Lean. He wrote NYT Bestseller book called The Lean Entrepreneur.

Moves the Needle bring Lean Startup methodologies to the Tori. Moves the Needle provides the principles, tools, tactics and strategies that allow product teams, innovators, support groups, and leaders to immediately act bolder, move faster and become more customer-focused. (We bring entrepreneurial spirit to the enterprise, 2016)

Moves the Needle (MTN) is the only firm that combines lean innovation thought leadership with practical "been there, done that" experience.

A lean looping in one of the concepts what "Moves the Needle" uses. It equips teams with the right framework to experiment. We conducted these lean loops in Tori to test our experiments.

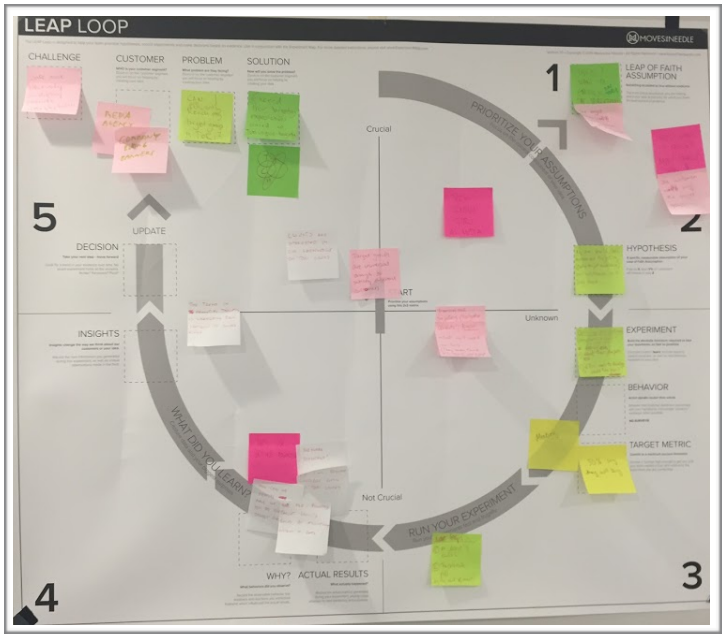


Figure 20: Moves the Needle lean loop

Lean loops are used continuously to improve the product. Below is demonstrated how the looping is used in practice.

First issue is to prioritize your assumptions: Start by listing all your leap of faith assumptions.
They should describe something you know what is true, but there is no market currently for it. You can collect the list of them, and once they are listed, you can start organizing them from known to unknown and crucial to not crucial. At this phase, you need to select your core leap of faith assumption. Leap of Faith assumption will be the ones which are most crucial and unknown.
Prepare your experiments: Map out your hypothesis. This phase you measurable description of your core assumption. To demo the feature you can build Minimum Viable Product (MVP) go out there and test it with real users. It is better to implement the real test feature.
Run your experiment. Identify key Learnings and updating your lean loop accordingly. Looping continues and is started again from the beginning. Loop the situation as long as you can improve the results.

Lean experimenting has a two-fold purpose. According to Moves the Needle testing your core "leap of faith" assumption and avoiding "waste".

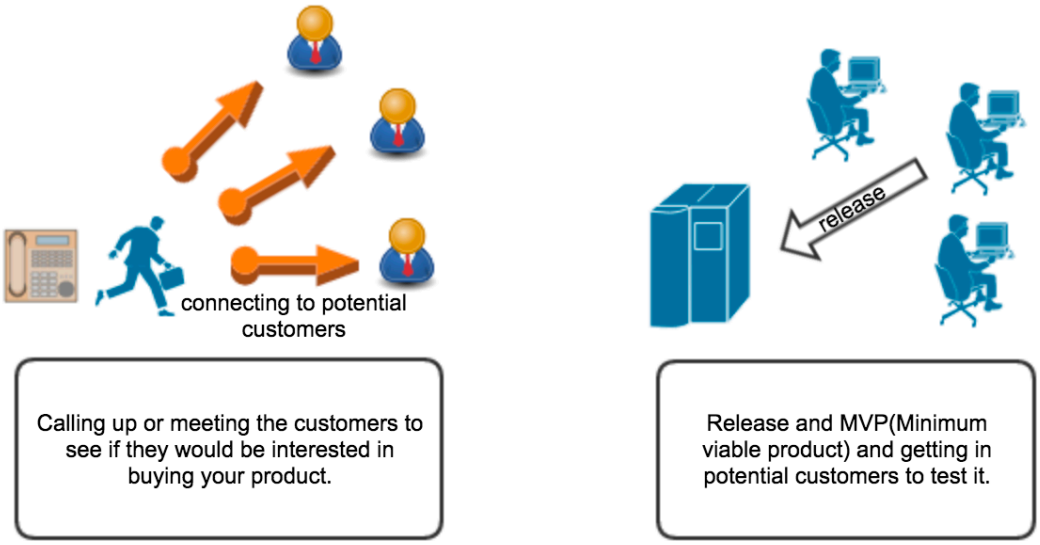


Figure 21: Lean testing if the concept is viable

Figure 21 explains a testing Lean concept. Testing the concept is important since there is a huge difference between what customers say they want, and what they will actually buy.

Testing with real customers gives valuable feedback. It gives you important ideas and the key learnings. Experimenting allows you to test your product through Minimum Viable Product(MVP). It's also important to iterate the idea until it is ready to be properly developed.

Muda comes from Japan and it literally means "waste". Areas of waste that can be avoided through efficient lean management. The target is to create a minimum viable product (MVP) A minimum viable product has just those core features that allow the product to be deployed and no more.

To test your core assumptions and see whether you have a market and if you are heading in the right direction. If you are not going to a right direction, you can simply drop the idea. It is also important to learn about why the customers are not willing to purchase the idea.

Benefits of the MVP are to test a product hypothesis with minimal resources accelerating learning, reducing wasted engineering hours, get the product to early customers as soon as possible and base for new products.

4.7 Lean A/B Testing with Optimizely Tool

Tori uses "Optimizely" tool for Lean A/B testing. This is a powerful tool for creating MPV models e.g. for Lean loops. It is very powerful and intelligent tool for testing new features.(Optimizely Product, 2017)

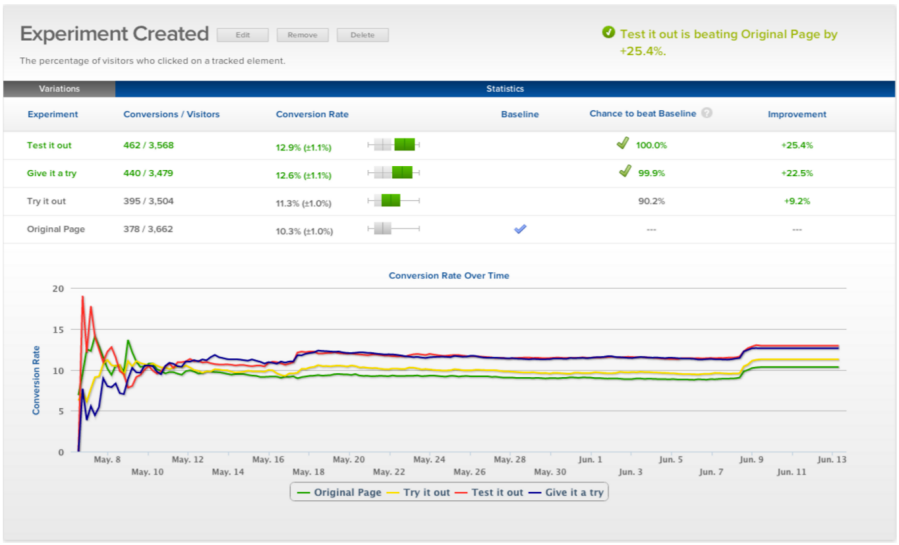


Figure 22: Snapshot of Optimizely used for Tori experiment

Optimizely is easy to use Web-tool to run A/B experiments. It helps to make easily data-driven decisions. Optimizely Tool makes possible to do easily modification to Web-page on real time without releasing Tori's code base. Optimizely is used mostly for User Experience purposes.

A/B testing is at the heart of the lean methodology as it enables teams to test assumptions and most importantly avoid wasting time and resources. More details about the tool in Appendix 4.

4.8 Software Branching Methodologies in Lean

Tori uses GIT version control system for software development to guarantee e.g. better Quality and Assurance. Git is a popular and widely used for source management system that greatly simplifies the software development cycle. (Git, Everything is local, 2017)

It makes possible users create, use, and switch between software development branches for content development as easily as people create and switch between files in their daily workflow.

As Tori works in the Koju model the development tasks are split to individual development branches. All the tasks are developed in separate branches, so this means that multiple developers can work simultaneously on the dedicated tasks without interfering others.

Tori uses three main branches. Master branch is Tori's production branch. All the code changes merged to this branch are normally released to production. It works partially as rollback branch if the faulty code is released to production.

In our software development model persons who are working in Quality and Assurance operations are generally taking care that correct features are merged to the branch. The persons are also up-to-date what features are merged to this branch, and when the branch is ready for release. Pre-release is used for merging ready developed features with pull requests to this branch. Software developed here will go trough Quality and Assurance. Develop branch is used for software feature development.

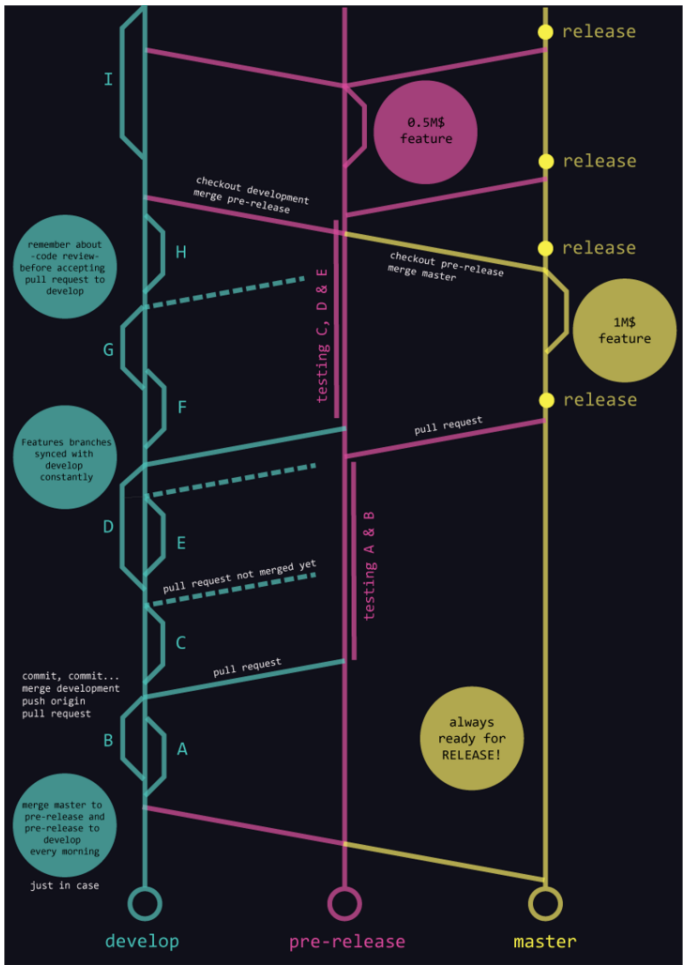


Figure 23: Flow chart explains the GIT branching used at Tori

Generally, software development is split from the develop branch. Quality Assurance(QA) makes sure that code is working and tests it in a pre-release branch. When it's approved to be working it's merged to master branch when it is ready to be released to production.

4.9 Agile Product Development Process

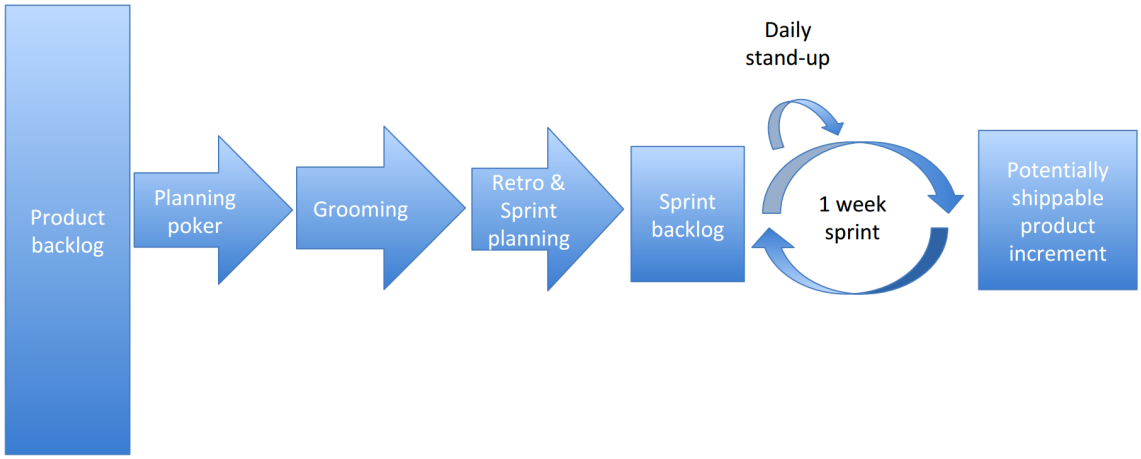


Figure 24: Tori's scrum sprint

Schibsted Media Group Finland, Tori.fi uses Scrum sprints. Scrum is a lightweight agile framework that is focused on making small teams work efficiently together to deliver a working product by employing methodologies and processes to streamline the work of the Scrum team for a successful software project. (Tal L. 2005, Chapter: Practicing Scrum Agile Development)
It's an iterative and incremental agile software development methodology for managing product development. It challenges assumptions of the traditional, sequential approach to product development, and enables teams to self-organise by encouraging physical co-location or close online collaboration of all team members, as well as daily face-to-face communication among all team members and disciplines in the project.

Scrum is facilitated by a scrum master. The scrum master is who is accountable for removing impediments from the teamwork. That helps a team to deliver the product goals and deliverables as efficiently as possible.

The scrum master is not a traditional team lead or project manager but acts as a "buffer" between the team and any distracting influences.

It has been internally agreed that ideal sprint would be one week long in time. It was also decided that after the sprint we would release the sprint features.

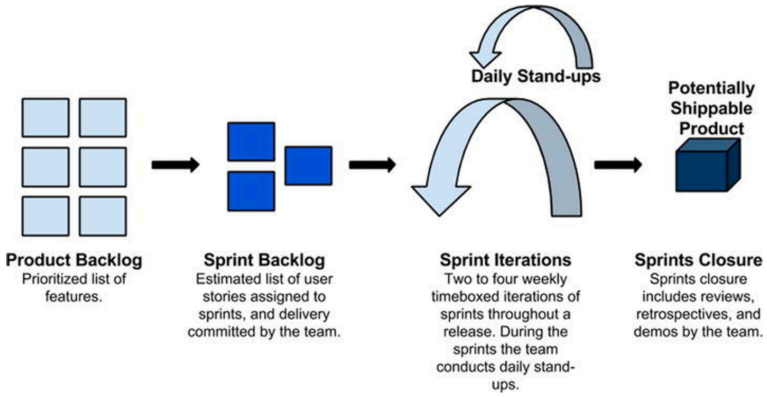


Figure 25: Scrum agile development (Tal L. 2005, Chapter 1, Figure 1-5)

Tori's agile model include Product backlog. Currently, all the product development ideas are collected to the product backlog. Sprint log is created from the product backlog. Sprint backlog and sprint log are both handled in Jira ticket system. Koju daily stand-ups are used to ensure that everyone knows and are able to solve what are the obstacles for development.

Planning poker meetings are organized once a week. The planning pokers are used for developers to estimate tasks work amounts. They are used to make the estimations how long will the tasks takes to implement. The make the estimates more accurate it's decided that one task should be the longest one day, else the task should be splitter to several tasks. An ideal would be that tasks should not be split bigger than to take time 2-3 hours each. The velocity calculation would be much more accurate in these cases.

Grooming is used as prioritization meeting with the person who is the main responsible for the product development. Grooming is meeting where the task is prioritized and taken into short term Sprint. Also grooming meetings takes care of elimination of the user stories that no longer appear relevant. It's very normal that during the development new tasks will arise, and they are prioritized in these meetings as well. Schibsted's product development model also has a retrospective and next Sprint Planning sessions. Retro is abbreviation from the word retrospective, what means looking back how past events or situations occurred against the plans. In retrospectives, we try to solve understand problems and improve the development model for the tasks.

In the Sprint Planning meeting, we are generally selecting and prioritizing the needs for the next sprint. On the grooming meetings, a person who is in a change of the product development may change the prioritization.

Some parts in Tori.fi's the development process is adopted from Extreme Programming. This is due to history since earlier we worked more in the extreme programming model. Good parts were adopted to present Scrum model.

Sometimes practices of pair programming with extensive code review has taken into practice. It's tended to use lean flat management structure, simplicity, and clarity in code, expecting changes in the customer's requirements as time passes and the problem is better understood, and frequent communication with the customer and among programmers. The development team is responsible for delivering potentially shippable increments (PSIs) of product at the end of each sprint. We don't have defined in Tori sprint team size, so in that sense, we work more in the Kanban style.

The development team is responsible for analyze features, design features, develop features, testing own codes, and QA for final testing, technical communication, and documentation.

4.10 Lean "Kanban" Visualisation in Agile

Even we are at the Agile Scrum development model with characters of Extreme Programming we tend to use Kanban visualization for the tasks.

The word Kanban in Japanese roughly translates to 'cards you can see'. (Mittu N. 2016) Today's lean thinking it is understood as providing visual clues for reducing unnecessary inventory. In Tori we use "Jira"-ticket system for visualization. Also, Trello is used in some Koju-projects since they are independent to choose the development tools.

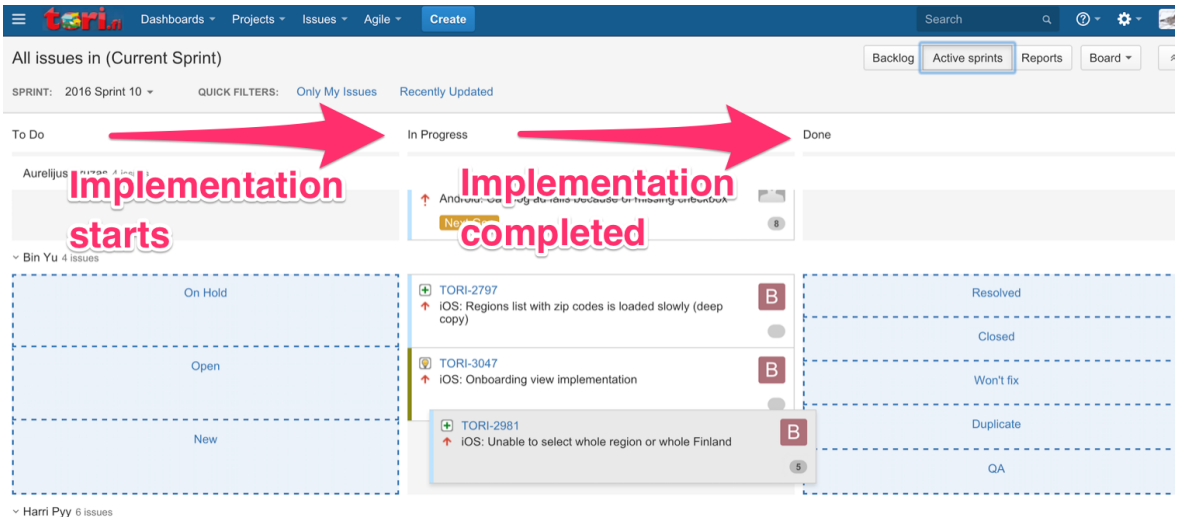


Figure 26: Example of the Kanban board in use

Figure 26 describes the practical Kanban board that is used for development. In our own model we have task that are "To do", "In progress", and "Done". Tasks that are "To do" are basically the tasks need to be implemented. When a developer starts implementing task he or she moves the task to "in progress" mode. When the task is implemented the developer moves the task for mode "Done".

“To do” tasks can be in different modes e.g on hold, open or new. On hold means that developer needs something before the implementation can start. Open means that when the developer is ready he can pick up a task. New means that developer has created new task needed to implement.

Tasks or tickets can be closed for “Done” stage. There are several options which are self-explanatory, Problem was resolved, Closed, Won’t fix, Duplicate, or move for Quality Assurance(ready for testing).

4.11 Jira Ticket Tool for Agile development

JIRA Agile or agile software development is a new and revolutionary way of developing software. JIRA is our main tool for Agile development as it lets development teams produce software with better quality, higher customer satisfaction, and improved efficiency. JIRA Agile brings the power of agile to Atlassian JIRA. JIRA is one of the most popular enterprise issue tracking and project management system. (Li P. 2015, Preface)

I am explaining in this thesis chapter the Schibsted Media Group uses Jira in the product development. Our lean agile product development uses Jira for Kanban visualization. This thesis demonstrates how the Jira-ticket system works.

Anyone in the organization can create Jira task. It's up to Agile Grooming processes to prioritize what tickets will be taken into Sprint log. Figure 27 describes how the new task or ticket can be created to Jira.

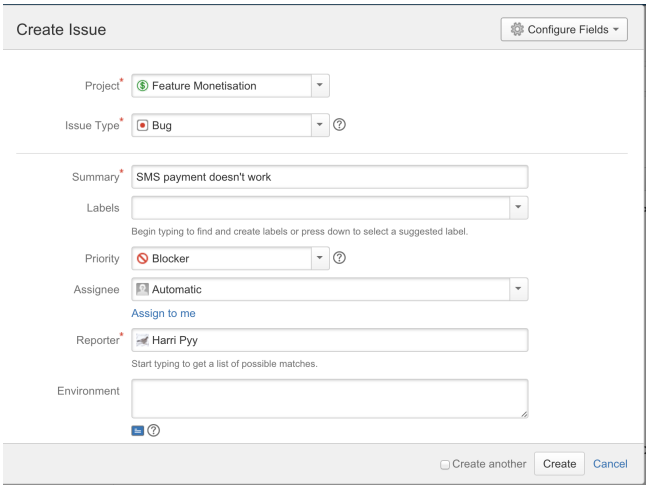


Figure 27: Creating Jira ticket

When there is a need for the new feature, bug fixing or other task for product development to carry out the ticket system is used for that purpose. Normally new task creating starts with creating new User Story or Ticket with Jira. User Story can include several subtasks. This means that bigger tasks are splitter to several subtasks. During the ticket creation, the priority of the ticket is requited. Normally tickets are critical or blocker bugs. This helps to prioritize what tasks will be implemented in practice. Normally when creating ticket the code changes are reviewed by another team member. Tori uses Github tool for code pull-requests. Code pull-requests means that another team member checks the code changes, and gives comments about them. When a developer implements the new feature the user Story is completed. Developer moves the ticket to Quality Assurance queue,

and it's tested manually by another team member. Functionality is verified on different platforms and browsers if applicable Transaction tests are created or updated, if a new transaction is created. Automated tests are executed during the Quality Assurance process.

During the ticket creation the selection of the project is done as depicted in figure 22.

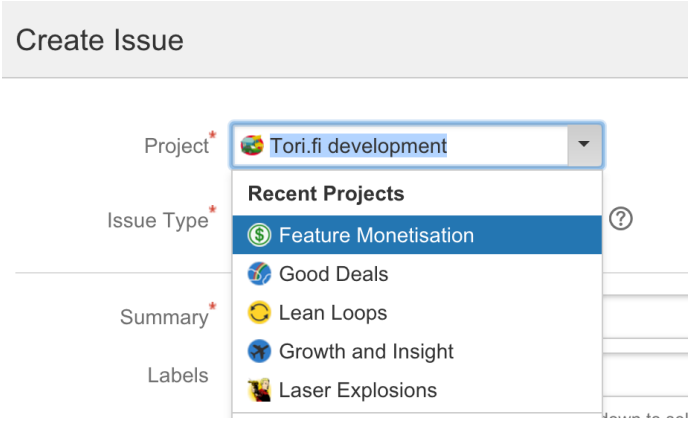


Figure 29: Selecting the Koju

Project selection organises the tickets so that different Koju's know what are their tasks. It's also easier to search tickets according to a project. Ticket creation has other fields as well. There is summary, priority, reporters name. For the summary section is described the summary of the implemented task. Priority can be selected to ticket initially, but the grooming decides when the ticket will be implemented. Alternatives for ticket priority choices are Blocker, Critical, Minor, Trivial. Figure 29 describes the priority selection in Jira ticket system.

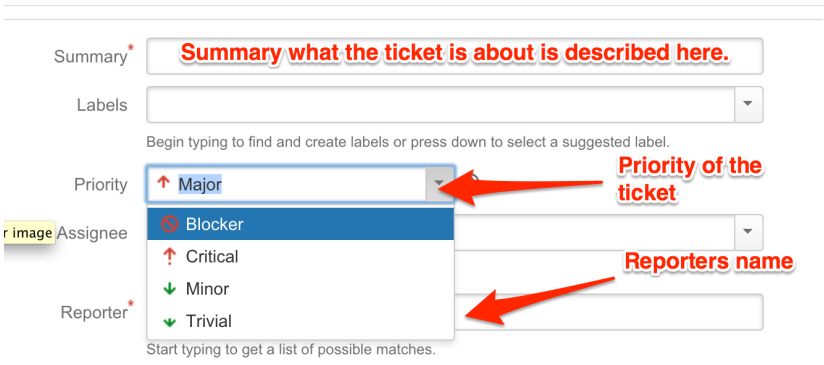


Figure 30: Selecting the ticket priority

Jira Ticket system is used to estimate the velocity of the team. Team velocity shows the actual work amount implemented in the Agile Sprints. Jira can be used for velocity estima-

tion of the Sprints in the future. As we can see the from the Figure 20 velocity in not currently stable, and the estimation of the feature development is not straightforward.

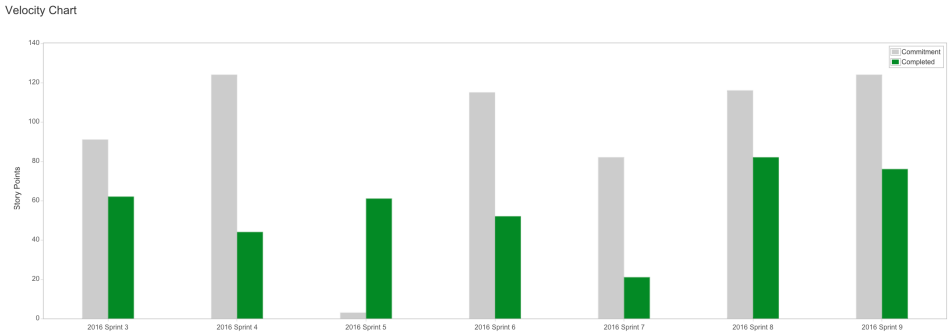


Figure 31: Example of velocity chart of Jira

4.12 Agile Way of Stand-ups

Every Monday and Friday at 9.20am all Schibsted Media Finland gathers upstairs in a circle and share their highlight of the day, and what they are planning on achieving. They are counted as well Tori's culture.

Figure 32: Stand-up meeting in Tori

Stand-ups should be very fast. It's not designed, but to tell the focus of the work in general level. Normally stand-ups should be over in 15 minutes.

This involves all the employees currently approximately 55 employees.(February 2017) The purpose of the stands is personal focus and top-of-mind project reflection. Simply everyone in the circle explains what they plans to be achieving. Also this includes sharing from learnings from mistakes and successes. The point is also getting together as a team on Monday morning and get the "drive" up for the week. This is also part of company culture in a way of for all employee to meet, share, caring and celebrating each other. To promote the company culture of get things done. Also one of the principle in the company is that "things that are promised are done". Generally stand-up is the first time meeting colleagues in the beginning of the week. Also Friday stand-up closes the week.

4.13 Quality Assurance(QA)

Quality assurance is a vital process. The term quality, when applied to intelligence really rates timeliness, accuracy, relevance, and actionability.
In many organizations, the software testing teams are referred to as QA teams. QA of software can be achieved by testing its reliability (functionality), recoverability, resiliency (security), and interoperability. (Mano P. 2015, Chapter 5.3 Quality Assurance)
Quality assurance (QA) is a way of preventing mistakes or defects in manufactured products and avoiding problems when delivering solutions or services to customers.
Quality Assurance has many responsibilities in Tori product development. The main responsibilities of the Quality Assurance are to write new User Stories and Bugs to JIRA. Also, keep the understanding of the ongoing bugs on the site, and inform developers about them.
Also, Quality Assurances responsibilities are to arrange design meetings. Design meetings are needed so that User Stories can be created.
Quality Assurance is responsible for the unit tests. Also, ensure that automated testings are working properly. Normally developers do regression testing before merging the codes to version control.
One Quality Assurances important task is to involve of code review. QA makes sure that codes are reviewed properly, and makes statistics about the code review. At the moment other developers are doing code review using Github Pull requests.
Integration testing needs to be done when features are merged to new branches. Integration testing happens when individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Code changes are reviewed by another team member using Github pull requests. The fix is manually tested by another team member. Functionality is verified on different platforms and browsers, if applicable. Automated tests are fixed if they got broken.

4.14 Conclusions And Summary of The Lean Model

Adopting an iterative and incremental development approach has changed in working practices of the management team and employees involved in the Tori Kojus.

The changes of the new development model adoption and acceptance in the organization have been sometimes difficult. All the employees have not obtained and accepted the new development policies. Kojus are not required to use any common practices, if not wanted.

The competencies are widespread in the organisation, and why the development of one thing blocks another easily. Lean flow control is blocked due to interruption of another Koju or maintenance work if certain competencies are required to do unexpected issues. For long term planning, we could increase the development resources.

These competence issues have some reflection to work estimations. It has been difficult to provide constant velocity and lean flow for the R&D development.(As seen earlier in the earlier that getting constant velocity has been problematic.)

Also, our strategic goals have changed frequently due to business changes, and will also in the future since the business sector we are in is very aggressive to changes.

5. Results of Data Analysis And Interviews

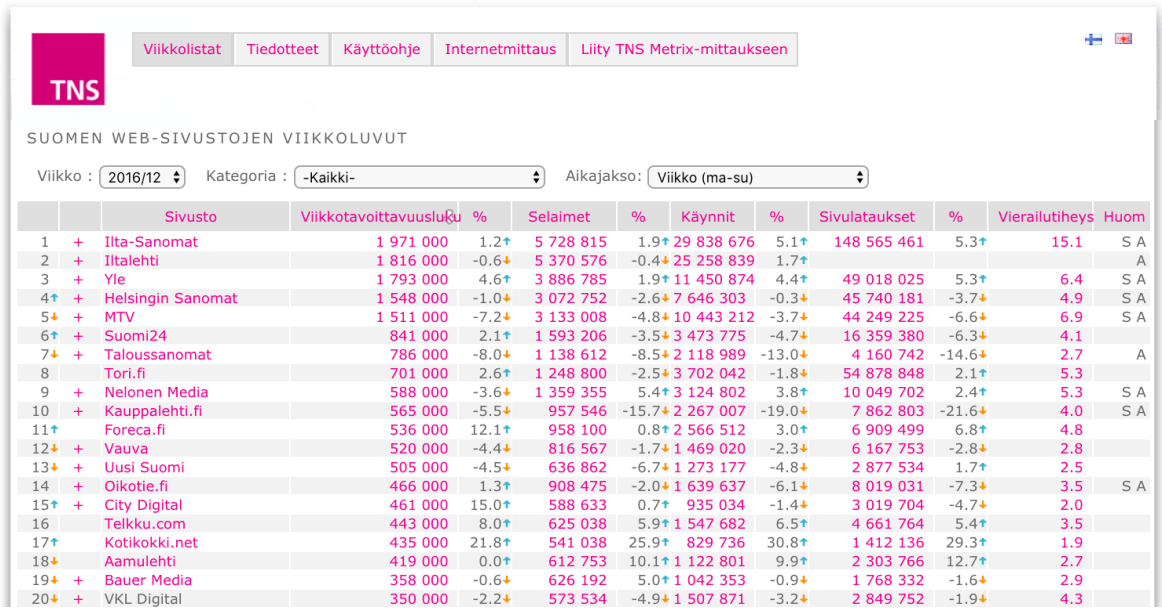
This chapter concentrates on the measurement data analysis results and personal interview results of the management team.

5.1 Measurements Results

Measurement results concentre on the website visitors and traffic, a bandwidth of the traffic, posted ads to the site, latency measurements of the ping and work efficiency measurements.

5.1.1 Measurement Results of Website Latency

Traffic figures from the TNS metrics have not measured changes. TNS metrics is public service measuring Website's visitors(Figure 35). The latency effect was already seen in week 12 the year 2016 when approximately 70% of the traffic was switched to hybrid cloud AWS and dedicated model. Front cloud servers are providing the most of the data for the service. The latency has not disturbed the ends users. Figure 35 shows that there was not effect in visitor counts.



		Sivusto	Viikkotavoittavuusluku	%	Selaimet	%	Käynnit	%	Sivulataukset	%	Vierailutiheys	Huom
1	+	Ilta-Sanomat	1 971 000	1.2↑	5 728 815	1.9↑	29 838 676	5.1↑	148 565 461	5.3↑	15.1	S A
2	+	Iltalehti	1 816 000	-0.6↓	5 370 576	-0.4↓	25 258 839	1.7↑				A
3	+	Yle	1 793 000	4.6↑	3 886 785	1.9↑	11 450 874	4.4↑	49 018 025	5.3↑	6.4	S A
4↑	+	Helsingin Sanomat	1 548 000	-1.0↓	3 072 752	-2.6↓	7 646 303	-0.3↓	45 740 181	-3.7↓	4.9	S A
5↓	+	MTV	1 511 000	-7.2↓	3 133 008	-4.8↓	10 443 212	-3.7↓	44 249 225	-6.6↓	6.9	S A
6↑	+	Suomi24	841 000	2.1↑	1 593 206	-3.5↓	3 473 775	-4.7↓	16 359 380	-6.3↓	4.1	
7↓	+	Taloussanomat	786 000	-8.0↓	1 138 612	-8.5↓	2 118 989	-13.0↓	4 160 742	-14.6↓	2.7	A
8		Tori.fi	701 000	2.6↑	1 248 800	-2.5↓	3 702 042	-1.8↓	54 878 848	2.1↑	5.3	
9	+	Nelonen Media	588 000	-3.6↓	1 359 355	5.4↑	3 124 802	3.8↑	10 049 702	2.4↑	5.3	S A
10	+	Kauppalehti.fi	565 000	-5.5↓	957 546	-15.7↓	2 267 007	-19.0↓	7 862 803	-21.6↓	4.0	S A
11↑		Foreca.fi	536 000	12.1↑	958 100	0.8↑	2 566 512	3.0↑	6 909 499	6.8↑	4.8	
12↓	+	Vauva	520 000	-4.4↓	816 567	-1.7↓	1 469 020	-2.3↓	6 167 753	-2.8↓	2.8	
13↓	+	Uusi Suomi	505 000	-4.5↓	636 862	-6.7↓	1 273 177	-4.8↓	2 877 534	1.7↑	2.5	
14	+	Oikotie.fi	466 000	1.3↑	908 475	-2.0↓	1 639 637	-6.1↓	8 019 031	-7.3↓	3.5	S A
15↑	+	City Digital	461 000	15.0↑	588 633	0.7↑	935 034	-1.4↓	3 019 704	-4.7↓	2.0	
16		Telkku.com	443 000	8.0↑	625 038	5.9↑	1 547 682	6.5↑	4 661 764	5.4↑	3.5	
17↑		Kotikokki.net	435 000	21.8↑	541 038	25.9↑	829 736	30.8↑	1 412 136	29.3↑	1.9	
18↓		Aamulehti	419 000	0.0↑	612 753	10.1↑	1 122 801	9.9↑	2 303 766	12.7↑	2.7	
19↓	+	Bauer Media	358 000	-0.6↓	626 192	5.0↑	1 042 353	-0.9↓	1 768 332	-1.6↓	2.9	
20↓	+	VKL Digital	350 000	-2.2↓	573 534	-4.9↓	1 507 871	-3.2↓	2 849 752	-1.9↓	4.3	

Figure 35: Public statistics about Tori.fi visitors

Figure 35 depicts the public statistic that can be seen from the public service. The figures are trackable from http://tnsmetrix.tns-gallup.fi/. Also the nature of the Website traffic is that it varies. Possible marketing campaign affects to visitor statistics. It's impossible to keep track all the marketing campaigns, since they are executed constantly. Traffic has only grown when the traffic has been switched to cloud.

From the figure 33 we can see that the Bandwidth of the Website dropped in one data-center about 300 Mbits/sec to under 100 Mbits/sec during the switch over of the traffic. This proves that data has been fetched from the cloud mostly with latency and not from dedicated servers.

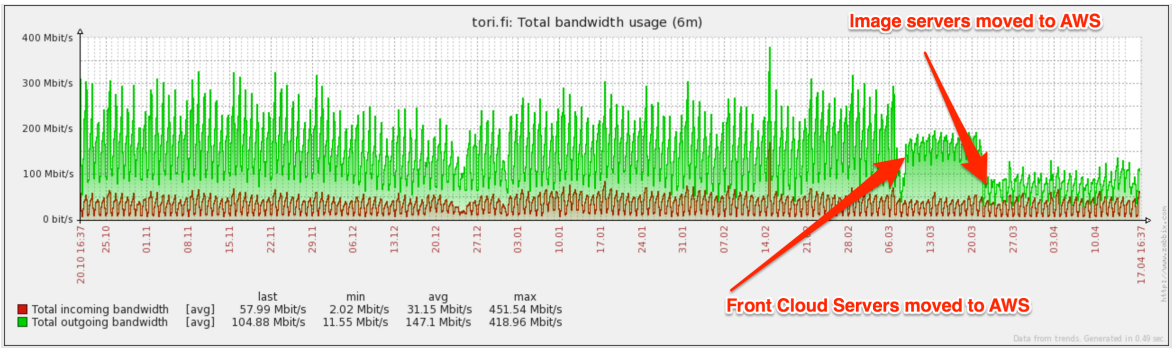


Figure 33: Tori.fi bandwidth usage from internal server monitoring

Bandwidth is the bit-rate of available or consumed information capacity expressed typically in metric multiples of bits per second.

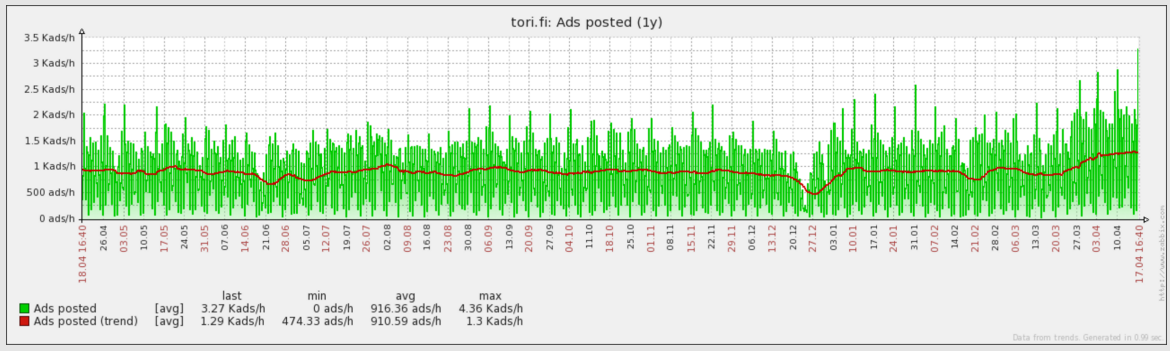


Figure 34: Tori ads posted

Other KPIs measured were be ads posted(Figure 34), first time visitors, bounce rate of the end user, visit quality conversion rate, page view per user, unique daily sellers, and new unique daily sellers. These KPIs didn't notice any difference in data.

```
16:02:49 harri@dev02:~ >> $ ping d38a5rkle4vfil.cloudfront.net
PING d38a5rkle4vfil.cloudfront.net (52.222.174.94) 56(84) bytes of data.
64 bytes from 52.222.174.94: icmp_seq=1 ttl=242 time=26.7 ms
64 bytes from 52.222.174.94: icmp_seq=2 ttl=242 time=26.9 ms
64 bytes from 52.222.174.94: icmp_seq=3 ttl=242 time=27.0 ms
64 bytes from 52.222.174.94: icmp_seq=4 ttl=242 time=26.9 ms
```

Figure 35: Describes the ping from office to front cloud in AWS

Figure 35 depict that the latency is on average approximately ~ 27 ms when the data is sent from the office.

```
PING www.tori.fi (87.108.21.143) 56(84) bytes of data.
64 bytes from front01.tori.fi (87.108.21.143): icmp_seq=1 ttl=57 time=2.10 ms
64 bytes from front01.tori.fi (87.108.21.143): icmp_seq=2 ttl=57 time=1.99 ms
64 bytes from front01.tori.fi (87.108.21.143): icmp_seq=3 ttl=57 time=2.01 ms
64 bytes from front01.tori.fi (87.108.21.143): icmp_seq=4 ttl=57 time=2.00 ms
64 bytes from front01.tori.fi (87.108.21.143): icmp_seq=5 ttl=57 time=2.08 ms
```

Figure 36: Describes the ping from office to local dedicated servers

Figure 36 depicts that ping from local office to dedicated servers is approximately ~2 ms. The difference between latency is approximately 25 ms, where AWS is as known slower in latency since the Cloud servers are physically located in Dublin, Ireland.

5.1.2. Measurement Results of Work Efficiency

Though we can measure the changes that have been done to GIT version control from Github tool depicted in Figure 37. This doesn't necessarily give right direction since e.g. infrastructure improvements don't require code changes. Also, the developer's count was growth during the last years. But adding manpower to late software project according to Brooks's makes it later.(Hsia P., Hsu C., Kung D. 1999, p. 370) In this sense we have proven that at the same time moving to a Lean model we have been able to implement more that can be visualized from the figure 37.

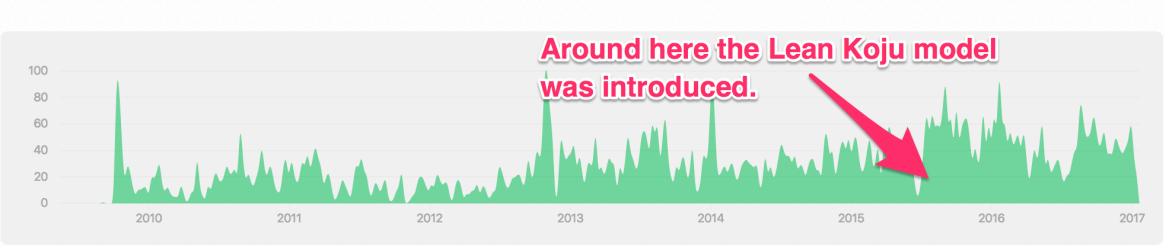


Figure 37: Displays the contributions to develop, excluding merge commit

Figure 37 is taken from the Github Version tool. It counts the software commits that have been done to version control. This measurement doesn't guarantee that the efficiency has improved, but gives reflection that most likely more features have been developed.

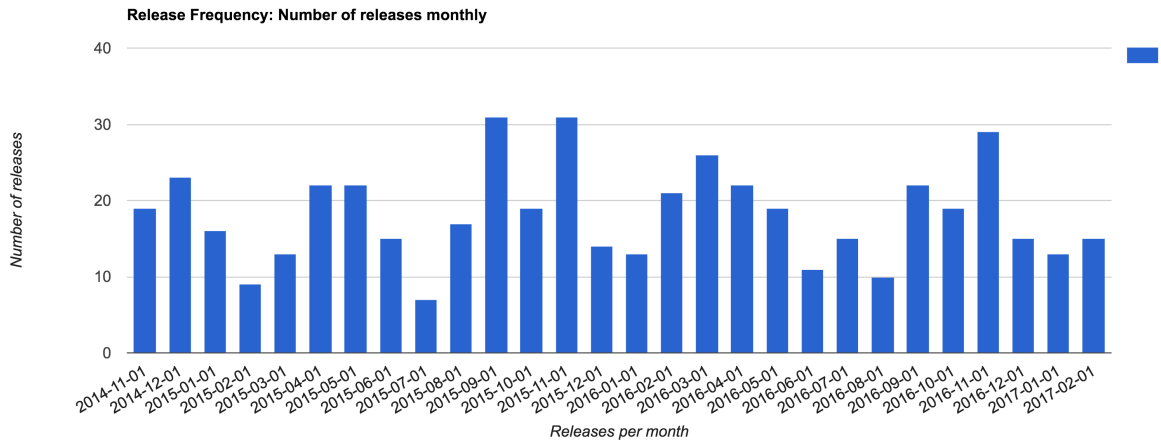


Figure 38: Number of production releases monthly

Figure 37 shows that the amount of code commits to GIT has improved, but figure 38 shows the release cycle frequency have been quite stable. The data was fetched from the GIT tags. Release frequency was measured the GIT version control. Every time there is release it is tagged with GIT tag to version control.

5.2 Interview Results

Interview questions were related to Research Questions to get perspective for the conclusions made with this thesis.

5.2.1 Interview Results About Website Loading Latency

Interviewing Lauri Halkosaari who is CTO of Schibsted Media Group (Finland) about the Website loading latency. According to Lauri Halkosaari Latency is an important factor for websites. Longer latencies mean that it takes longer for users to get responses and that in turn leads to unhappy users (and fewer users overall). In that sense extremely important for a Website.

Based on our measurements the average latency caused by our Finnish user base by hosting in clouds in Ireland is between 20-30ms. That means 0.02 or 0.03 seconds, and it is less than a normal human being can even notice.

However, even this latency can cause issues if it is multiplied: for example doing multiple calls between servers in the cloud & datacenter in Finland before returning the result for the end user. For this reason, it is important to think about the architecture of the distributed system & avoid unnecessary calls between datacenter.

Interviewing Lauri Halkosaari who is CTO of Schibsted Media Group (Finland) about the measuring the latency. According to Lauri the End user latency is measured with continuous monitoring from fixed metering points as well as with real user monitoring. Mea-

surement is done tracking how quickly on average the end users have a full page rendered in their browsers.

5.2.2 Interview Results About Flexibility Improvement

Interviewing Jussi Lystimäki who is Vice President of Schibsted Media Group. (Barcelona) According to Jussi Lystimäki, when companies and size of the team grows, that usually leads to lower efficiencies and much higher efforts in coordination of the focus in operations. Usually, companies create hierarchies, departments, and silos to make this scaling of a number of employees happen in an effective way. But for us, this opportunity led to a thinking about liquid organizational models and Lean methods to make sure the growth momentum and passion to work could be sustained throughout the growth of the company and growing complexity of the business environment the company was facing. We visited several models from lean teams to pure self-managing teams. In addition, the business environment was becoming more and more uncertain and unstable to be able to predict 12-18 months ahead or do heavy upfront investments with a validated data to have a decent payback rather fast.
In Tori, the lean model was applied with a mindset of a team of teams, namely to create a model where we can scale up and down a number of people focusing on certain key projects or problem-solving. One day would work with few colleagues to solve a problem as a small start-up and next week having 20-30 people around you to give more muscles and scale to run bigger efforts when required. Namely, there would not be fixed silos in the organization, but quite a flexible structure, also giving flexibility for employees to develop themselves during their career. This approach helped us to become more fluid with the adaptation of the focus in resourcing, and gave an opportunity to scale the level of efforts across main priorities of the company and key projects on the table. That resulted in having more speed to solve problems, employees felt more connected with their work and company, there were much more engagement and excitement thru actual work in this way, and we could adapt the company focus from zero to 100% on main projects. Namely, we could dedicate only a few persons to solve and develop a case, or even have one week with 100% team just to crack one key problem in the business together. We called these 100% weeks, as “take-over weeks”. One additional benefit was that during those weeks, we all felt very connected as a team, our brains could focus only on few things and make them well, and the level of team collaboration and learning was many times higher than in normal weeks.

The lean approach meant, that we could start solving business issues and problems with low efforts, low friction and high pace without heavy investments. The lean approach helped to test and justify the problem solving and use cases to the level, where the com-

pany felt "ok" to invest heavier and heavier based on accumulating data and feedback from the market, users, and business systems. For an employee, this brought in more diversity in the contents of work, a scope of the role and challenged to learn new things and skills. Also, each project was more like an exploration and finding your way forward as an individual, which gave an employee much more flexibility to organize your work and working time in the way it worked the better.

The lean approach meant, that we could start solving business issues and problems with low efforts, low friction and high pace without heavy investments. The lean approach helped to test and justify the problem solving and use cases to the level, where the company felt "ok" to invest heavier and heavier based on accumulating data and feedback from the market, users and business systems. For an employee, this brought in more diversity in the contents of work, scope of the role and challenged to learn new things and skills. Also, each project was more like an exploration and finding your way forward as an individual, which gave an employee much more flexibility to organise your work and working time in the way it worked the better.

Interviewing Lauri Halkosaari (CTO of Schibsted Media Group, Finland) about flexibility for working improved in Lean. According to Lauri Halkosaari the main source for flexibility in lean come from shorter cycle times: the time from the start of the development process to shipping the ready feature to users. Changes during the development cycle are always a bad thing, because that leads to confusion, quality problems and less efficient processes. So a shorter development cycle means better reaction times (and shorter time to market) without compromising quality or efficiency.
In waterfall projects the cycle time from planning to release is usually measured in months, some agile processes like Scrum focus on sprints (cycle time measured in weeks) and with lean we have been aiming for continuous releases. Continuous releases we mean that we release immediately when a feature is ready & plan/adjust continuously. This means that we can make changes to the development plans daily as well - a big difference.

5.2.3 Interviews Results of Work Efficiency

According to Jussi Lystimäki (Vice President of Shibsted Media Group) one key topic in our working approach was to create good flow and identify possible blockers upfront, to keep the team working effectively. The efficiency was mainly measured by velocity of the work, how fast we could solve possible blockers before they started to slow us down and what was the throughput of the efforts to make the needles move.

According to Jussi Lystimäki, when companies grow, there is more and more friction to make things happen and done. Also, making changes in organization structure and focus is usually slow and requires lots of extra energy and effort. The lean model helped us to lower this friction significantly, resulting in higher speed, a quicker adaptation of organizational focus, solve key problems much faster, assign fast more energy and efforts to areas which became issues to be solved, the employee satisfaction went up and stayed there. However, the main measure was to see that we had much higher level of collective learning to solve problems, which meant that thru this model the organization was able to navigate, adapt and grow healthy in changing business environment. In the end, this all leads to much high productivity and ability to keep on growing despite the evolution and revolution of the internet business and technology. Also, it helped to position yourself as a front-running employer, and excited people to apply for jobs in the high competitive recruitment market.

Interviewing Lauri Halkosaari who is CTO of Schibsted Media Group (Finland) about how the flexibility is measured. According to Lauri Halkosaari the key indicators for flexibility in our processes are the number of releases (more the better) as well as the length of the cycle time.

5.2.4 Interview Result of the Factors Measuring Work Efficiency

According to Jussi Lystimäki (Vice President of Schibsted Media Group), each discipline and domain area was measured a bit differently, to use relevant and meaningful measures. For example, the product and technology team was measure thru estimating velocity of the team and how much throughput were we able to make happen with a good feel and work satisfaction. Small autonomous teams were measured by the level of weekly experience and learnings, and how much progress they can do and on which level their Proof-of-Concept was evolving. The management measured the total efficiency of working by monitoring tens of key metrics of the user traffic, brand measures and our marketplace liquidity. This was to see if the efforts were allocated on right things to make those key needles move, and Tori would become better and better marketplace to sell consumer goods, cars, and other products.
Interviewing Lauri Halkosaari (CTO of Schibsted Media Group, Finland) about factors that were used to measure work efficiency. According to Lauri Halkosaari measuring efficiency in software development is hard to do in a way that provides real value. Things like measuring story points in a sprint are possible, but they usually suffer from the "you get what you measure" - syndrome. How do we know that we've actually improved the process instead of just changing how we measure size? And story points are only usable in tracking

the progress of a single team - they are not useful when teams change often, like in Tori's environment.
For this reason, we have focused more on tracking the change in business goals - are we reaching our goals in terms of usage & revenue? Are projects on time? For supporting these we are also measuring the number of releases as well as the number of tasks/stories done, since over a longer time period the average size of these will be even: the are smaller & bigger tasks, but over a month's time the average will be close to same. Another benefit is that the cost of measuring these things is very low (not much time spent) - measurement in itself is always "waste" in terms of lean: it is must to have to be able to improve, but it is not producing direct value for the customer. We will always need to measure things, but we need to be mindful of the benefit we get from the measurement itself.

5.3 Conclusions and Summary of the Data Analysis And Interviews

Measurement and interviews concentrated on the Website visitors and traffic, a bandwidth of the traffic, latency measurements of the ping, work flexibility and work efficiency.
With the measurements, the Website Traffic did not affect negatively when the traffic was switched to a cloud. Bandwidth showed that the traffic was actually switched to a cloud, and measurements prove that users are as active as earlier. The ad insert rate KPI didn't have an effect. The Website is facing extra 25 ms latency. Latency will improve the situation right now, but lately Apr 04, 2017 AWS announced that they have plans to opening data centers to Sweden. (Darrow B, 2017) There have been rumours that Amazon open data centers in Finland as well in the future.(Juvonen A, 2016) Overall the feeling is that the project has been successful, and the future looks as well better when the data centers come closer to Finland.
Interviews with Jussi Lystimäki (Vice President of Schibsted Media Group) and Lauri Halkosaari (CTO of Schibsted Media Group, Finland) gave good insight that the Lean environment has been successful towards the organization. Figure 37 also proved that the working efficiency has improved when moving toward Lean principles. Overall which was not measured but was mentioned in the interviews was understood that when the employee satisfactory improved when we switched working in the Lean environment. It was also stated that when the employee satisfaction improves it reflects in a positive way to work efficiency.

6. Conclusions

Conclusions of the research questions are discussed in this chapter. A research was conducted in Schibsted Media Group, Finland, and they concerns for Tori Website. Thesis aimed at identifying what factors were affected when moving hosted services to Cloud environment. Also work described the Lean working methodologies in the company as well.

6.1 Answers to Research Questions

In this chapter, there are answers to the research questions. The first pair of questions were related to the suitability of the AWS Cloud solutions due to Website page load time latency. Answers and conclusions to research questions are made from the statistics, employee interviews, and own conclusions during the implementation.

Answer 1: How the latency affects to website traffic?

The latency effect was already seen when approximately 70% of the traffic was switched to hybrid cloud and dedicated model. These results revealed that website loading latency did not impact negatively to daily visitors, ad insert counts to website or other KPIs measured. The measurement results are seen in Chapter 5.1.1. The popularity of the site remains without disturbances. Traffic has grown to be even higher. The AWS migration has not caused any feared issues to website traffic. TNS metric also points out (Figure 35) that tori.fi are the 8th biggest website in Finland. According to interview results of Lauri Halkosaari (Chapter 5.2.1) the latency difference is so small that users do not most likely notice the difference.

Answer 2: How the end user latency is measured?

The latency was measured with the ping tool with the location of Schibsted Classified Medias office. The Physical servers are located in near Kalasatama, Helsinki and second server facilities are located in close to downtown of Helsinki. The distance and internet connection type affects from the latency. AWS latency is measured from the office to be around 26 ms. The ping was made to front cloud servers.

The difference between latency is approximately 25 ms, where AWS is as known slower in latency since the Cloud servers are physically located in Dublin, Ireland.

I concluded during this thesis work that latency doesn't give the whole truth about the loading the content of the Website.

The users will necessarily notice this change since there is latency as well when the Internet Browser renders the visible internet pages to graphical form. All the content download transfer on the page is not started parallel at the same time, and it is affected to Web browser rendering. There are differences in the speed of rendering pages between differ-

ent internet browsers e.g. Internet Explorer, Opera, Firefox, Safari, and Chrome. Each of these Browsers renders the data with different speed.

Measuring the different browsers for latency is not the scope of this thesis work since it is so vast area of information and has several factors affecting it. But we can see from the figure 35 that the latency did not have affect on user experience. Therefore it did not impact to Website loading amounts in short term. This is still assumption since the service is partly on AWS. Currently impacts were not noticed. It is further analysed when rest of the services are moved to cloud.

Improvement of the flexibility for the employees to work in Lean environment

Lean part of this thesis work concentrates on research case study methodology. Interviews were carried out from Tori's employees mostly working in the management part.

Answer 3: How the flexibility for working improved in Lean?

Working in Lean it's continuous experimentation. Also, ideas that are not worth of developing will be terminated in the early phase. Normally as we have operated in a lean model we have gathering quick insight on how the users behave and react as well as give feedback.
One of the advantages has been the development speed. The advantage has been to quickly discover whether there is real demand or interest for your service, functionality or feature. Also, the cost factor has had a positive impact. We have been concentrated on developing the right things, and the plan has been forgetting the wrong things quickly.
Analysing the interviews(Chapter 5.2.2) and work efficiency measurements(Chapter 5.1.2) it could be interpreted that the results moving to Lean model seems to be positive. Lean has become partly of the company's culture. Also concluding from the interview results described in Chapter 5.2.2 that self-leadership plays important role with the satisfaction of employees.
As my final conclusion adding everything together, the working flexibility has improved when we switched to lean model.

Answer 4: How the flexibility is measured?

Conclusions based on the interviews (Chapter 5.2.4) made during this thesis. It is more flexible to work in Koju model than in the pure Agile model. Koju model requires less control for employees to function and therefore less micromanagement is required. Generally, product development gets friction when the organization grows. In lean Koju the pur-

pose is to lower that friction. The purpose in Lean is to shorten the development cycles which can be measured as release cycles.
Koju model tends to lead to improved employee satisfaction, which can be measured the way that it has given employees more flexibility to organise their work and working time.

Answer 5: How the work efficiency will be measured?

It is difficult to measure the work efficiency improvement directly. Working in the Lean model means that hierarchical working model is dropped, and it is not necessarily everyone follows the Agile methods creating User Stories. The efficiency can be seen improved, which is proven in investigating the statistics e.g. figure 37 in Chapter 5.1.2. According to measurements, the development has become more efficient with the same time when the company has grown. According to Brooks's law, the efficiency should be lower when adding more employees to the late software project. (Hsia P., Hsu C., Kung D. 1999) This has not been the case in Lean organization model. Release cycle though have been quite constant in time, which can be seen from figure 38 in chapter 5.1.2. This can be due to corrective releases and it's measurements does not necessarily give the whole truth. But figure 37 in chapter 5.1.2 shows that amount of GIT contributions to develop has increased. This means that we are able to produce more code. But measuring R&D efficiency is generally very difficult. Generally lines of code is very bad measurement of the efficiency. If it is measured as "de facto" than software developers will start to produce more lines of code. Also the more talented software developers write more compact code. My personal view is that lines of code measurement is bad, since it may lead to bad quality of implementation. In that sense it makes sense what Lauri Halkosaari's interview was revealed that you get what you measure(Chapter 5.2.4). My final conclusions are that it is very difficult to measure the organisations work efficiency in software development. Based on the interviews and intuition I feel that generally the working in Lean is more comfortable than in pure Agile.

Answer 6: What factors were used to measure work efficiency

The work efficiency was measured with the software contributions develop. (Figure 37 in Chapter 5.1.2.) One measurement was also the figure 38 in chapter 5.1.2 for the release cycles. More conclusions were that the interviews played the most important role explaining that work efficiency is a sum of many things. Conclusions were made that work efficiency in engineering work is almost impossible to measure directly. Also working on the Lean environment is based on trust. The micro management is dropped, and accurate monitoring what people are done is not measured. In the future as a conclusion, it would be better to concentrate on rather measuring the employee satisfaction than work effi-

ciency. Reinventing wheel several times might look from outside efficient if we measure what steps were required to do the implementation.
According to interviews and by intuition the conclusion was made that it is not necessarily important to measure the work efficiency in the way that it is measured quite often in the software companies. It's not sensitive to measure engineering work as efficiency to produce software. In that sense, my conclusions are that working in Lean is the way more comfortable, and not all the time goes reporting what be done and how.

6.2 Suggestions for Cloudification

As the migration was partly implemented from existing Xen-based infrastructure to AWS cloud, it was realized that we are able to move the current platform and it's services to the cloud. Moving services to AWS cloud will cause some latency for content loading. There are Front End development ideas that can produce a user experience that latency effect is minimal.
In the future, we would like to have a new container-based architecture in the public clouds using related CI/CD tools. The good plan is to move towards DevOps thinking in the Lean environment.
As development suggestion, we would like to also automatise the operations and productions environment to large-scale distributed systems, so no manual intervention is not needed eventually.
Ideal target would be to develop the best possible continuous delivery pipelines supporting features like an automated promotion to production, automated canary releasing or blue-green deployments.
Also we could develop further the analysis and troubleshooting statistics. There we would notice quickly when things go wrong, and opportunity to locate the faults efficiently.
Also improving the team work with local development team making sure that they follow the infrastructure guidelines. This helps us to make reliable design and implement systems that fail over gracefully and are transparent to clients.
Plan could be also to improve monitoring and logging solutions that enables the production systems to be monitored 24/7.
Also, we could take into account in the DevOps development that we make sure that technology solutions are in the future robust, will scale, and failover.
Also deploying the system infrastructure so that maintenance of databases and data store clusters across multiple datacenter.
We could as well give up maintaining partly some internal development tasks, and purchase them as service.

6.3 Suggestions for Lean model

Generally working the Lean model have been a positive effect. It has given freedom to work flexibly and efficiently. Lean tools have been useful when creating new concepts and ideas for development. Generally, all the new features go trough this lean path.

Perhaps should be more effort to persuade people to use the Agile models. All the employees don't necessarily create process model Jira tickets but interrupt development where the flow control is interrupted. Development "traffic" is jammed or interrupted and waiting waste is created. Koju model gives freedom for this, and therefore it's not required to use any methodologies.

Also, our R&D development resources with certain competences are quite low why interruption can shuffle the whole timetable. A lot of waste is created due to another waiting to finish another task.(Lean waste of "waiting")

My proposal for these problems would be that we would take Kanban into usage. The company has already adopted a Lean way to develop new features. We could also use less process oriented startup way to develop features. This could improve the team work as well. The problem still is that the competences are wide spread, and it's not sure if this model would work in practice.

The current Lean model is very flexible. Its good from the point of view of developer satisfactory. It doesn't force employees to bureaucracy in software development.

We could drop partly parallel management. The Koju organisation is a bit unclear, and there are lot of controversy. The version control is not used the check the status, but rather than personal contact. The culture around it is not yet fully established.

From the Extreme Programming(XP) we could use more features of pair programming. This would ensure that all the developers would understand better the development stack. But this would waste development resources in short term, but long term would improve the competencies.

Successful iterative and incremental development requires a progressive and adaptive approach to be taken to the management of the project and requires the whole team to embrace change and the continual improvement that this change will hopefully produce.

References and Literature

About AWS, 2016. Readable: https://aws.amazon.com/about-aws/ Read: 10-JAN-2016

Amazon CloudFront, 2017 – Content Delivery Network (CDN), Readable: https://aws.amazon.com/cloudfront, Read: 22-Jan-2017

AWS Amazon Virtual Private Cloud, VPC, 2016, Readable: https://aws.amazon.com/vpc Read: 16-JAN-2016

AWS Auto Scaling, 2016, Readable: https://aws.amazon.com/autoscaling/ Read: 16-JAN-2016

AWS Elastic Load Balancing, 2016, Readable: https://aws.amazon.com/elasticloadbalancing/ Read: 16-JAN-2016

Bell S., Orzen M. 2011, Lean IT, Enabling and sustaining your lean transformation. ISBN: 978-1-4398-1756-8, Publisher: CRC Press 2011, Originally published: September 14, 2010

Crookshanks E. 2015, Practical Software Development Techniques: Tools and Techniques for Building Enterprise Software, Publisher: Apress, Publication Date: 03-APR-2015, Insert Date: 04-MAR-2015

Darrow B, 2017, Amazon Cloud Goes Nordic, Readable: http://fortune.com/2017/04/04/aws-swedish-data-center, Read: 16-APR-2017

Fuse Over Amazon, 2017. Readable: https://github.com/s3fs-fuse/s3fs-fuse/wiki/Fuse-Over-Amazon, Read: 29-JAN-2017

Git, Everything is local, 2017. Readable: https://git-scm.com/ Read: 24-2-2017

Giles H. 2016. The Nature of Business. Readable: https://thenatureofbusiness.org/2016/09/23/the-organisation-is-a-purposeful-living-system/ Read: 15-DEC-2016

Haden J. 2014. Traditional Leadership Hierarchies Are Dead, Or Should Be. Readable: https://www.inc.com/jeff-haden/traditional-leadership-hierarchies-are-dead-or-should-be.html Read: 14-FEB-2016

Helmke M. 2017. Ubuntu Unleashed 2017 Edition, Publisher: Sams, Publication Date: 21-OCT-2016, Insert Date: 12-OCT-2016

Hsia P., Hsu C., Kung D. 1999, "Brooks's law Revisited: A System Dynamics Approach,", Twenty-Third Annual International Computer Software and Applications Conference, 1999.

Juvonen Anna, 2016, Amazon Plans to Finland, Readable: http://www.tekniikkatalous.fi/tyoelama/amazon-on-ehka-tulossa-suomeen-salamyhkaisesta-hankkeesta-kuultiin-sivukautta-6572698, Read: 16-APR-2017

Li P. 2015. JIRA Agile Essentials, Publisher: Packt Publishing, Publication Date: 25-JUN-2015, Insert Date: 01-JUL-2015

Lukka, K. 2000. The Constructive Research approach. Metodix. Accessed 12 December 2015. Readable: https://metodix.wordpress.com/2014/05/19/lukka-konstruktiivinen-tutkimusote/. Read: 15-MAR-2017

Mano P. 2015, Official (ISC)2 Guide to the CSSLP, Publisher: CRC Press Publication Date: 15-SEP-2015, Insert Date: 16-DEC-2014, Table of Contents, Start Reading

Maurya A. 2010, Business Model Toolbox, 2017. Readable: http://bmtoolbox.net/tools/lean-canvas/ Read: 5 -March-2017

Memcache Homepage, 2017. Readable: https://memcached.org/ Read: 21-Jan-2017, Redis Homepage, https://redis.io/ Read: 22-Jan-2017

Mittu N. 2016, Top 5 Criteria To Consider While Choosing A Kanban App, Readable: http://www.dkrimmer.de/2016/12/09/top-5-things-to-consider-while-choosing-a-kanban-app/ Read: 10-FEB-2017

Newman S. 2015, Building Microservices, Publisher: O'Reilly Media, Inc., Publication Date: 10-FEB-2015, Insert Date: 06-MAY-2014

O'Reilly Media 2016, Designing for Product Strategy, 2016. Publisher: O'Reilly Media, Inc. Publication Date: 16-MAY-2016, Insert Date: 28-JUL-2016

Optimizely Product, 2017. Readable: https://www.optimizely.com/products, Readable: 26-3-2017

Orzen Michael; Thomas A. Paider, 2015, The Lean IT Field Guide, Publisher: CRC Press, Publication Date: 14-OCT-2015, Insert Date: 03-DEC-2015

Postgres SQL, 2017, Official Website. Readable: https://www.postgresql.org/about/ Read: 21-Jan-2017

Ries E. 2011, The Lean Startup, Originally published: 2011, Author: Eric Ries, ISBN: 0307887898, Country: United States of America, Genres: Business, Non-fiction, Entrepreneurship

Sarkar A., Shah A. 2015. Learning AWS, Publisher: Packt Publishing, Publication, Date: 30-JUL-2015, Insert Date: 01-AUG-2015

Schibsted Media Group Homepage, 2017. Readable: http://schibsted.com Read: 5-JAN-2017.

Six Sigma, DMAIC Methodology, 2017, Villanova University, Readable: https://www.villanovau.com/resources/six-sigma/six-sigma-methodology-dmaic, Read: 1-March-2017

Spinnaker Documentation, 2016. Readable: http://www.spinnaker.io/docs Read: 11-JAN-2016

Tal L. 2005. Agile Software Development with HP Agile Manager. Publisher: Apress, Pub. Date: March 27, 2015, ISBN: 978-1-4842-1035-2

Tori Media Homepage 2017. Readable: http://media.tori.fi/tori/ Read: 1-JAN-2017

Turkington G. Deshpande T., Karanth S. 2016, Data Processing and Modelling, By: Garry Turkington; Tanmay Deshpande; Sandeep Karanth, Publisher: Packt Publishing, Publication Date: 31-AUG-2016, Insert Date: 06-SEP-2016

Tracey E. 2015. What's it like to work at Spotify, Readable: http://blog.honeypot.io/2015-11-13-what's-it-like-to-work-at-Spotify/ Read: 16-APR-2017

Varma T. 2015. Agile Product Development: How to Design Innovative Product That Create Customer Value, Publisher: Apress, Publication Date: 04-NOV-2015 Insert Date: 07-NOV-2015

Vyas U. 2015, Mastering AWS Development, Publisher: Packt Publishing, Publication, Date: 30-JUN-2015, Insert Date: 04-JUL-2015

We bring entrepreneurial spirit to the enterprise, 2016. Move the Needle, Readable: http://www.movestheneedle.com/ Read: 15-JUN-2016

What is Cloud Computing, 2017, Amazon Web Services. Readable: https://aws.amazon.-com/what-is-cloud-computing/. Read: 8-JAN-2017.

What is Lean, 2017. Readable: https://www.lean.org/WhatsLean/, Read: 2-MAR-2017

What is Zabbix, 2017. Readable: http://www.zabbix.com/product, Read:1-MAR-2017

XiTI tool by AT Internet, 2017. Readable: http://www.atinternet.com/en/web-analytics-solution/ Read: 1-MARCH-2017

Appendix

Next table contains this thesis terminology and abbreviations.

Appendix 1. Terminology And Abbreviations

Terminology, Abbreviations	**Explanation**
AWS EC2	Amazon EC2 is a web service that provides compute capacity in the AWS cloud. It's virtual server that can be utilised from cloud.
AWS AMI	Amazon Machine Images (AMIs) contains different operating systems, or software bundles.
AWS S3	Amazon Simple Storage Service is storage for the Internet. It is designed to make web-scale computing easier for developers.
AWS	Amazon Web Services
AWS	Amazon Web Services (AWS) is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality to help businesses scale and grow.
AWS ELB	AWS Elastic Load Balancers (ELB) automatically distribute traffic across EC2 instances
DevOps	DevOps (a clipped compound of development and operations) is a term used to refer to a set of practices that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes.
Extreme Programming, XP	Extreme programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements.
GIT	Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
Hoshin	Hoshin is Strategic planning ensure that the mission, vision, goals, and annual objectives are communicated throughout an organisation, and implemented by everyone from top management to the bottom level.
IaaS	IaaS: Cloud infrastructure services, known as Infrastructure as a Service (IaaS), are self-service models for accessing, monitoring, and managing remote datacenter infrastructures, such as compute (virtualized or bare metal), storage, networking, and networking services (e.g. firewalls). Instead of having to purchase hardware outright, users can purchase IaaS based on consumption, similar to electricity or other utility billing.
Kanban	Kanban is an inventory-control system to control the supply chain. A Kanban board in software development is a work and workflow visualization tool that enables you to optimize the flow of your work
MVP	Minimum Viable Product(MVP) has just those core features that allow the product to be deployed and no more.

Terminology, ***A*bbreviations**	**Explanation**
PaaS	PaaS: Cloud platform services, or Platform as a Service (PaaS), are used for applications, and other development, while providing cloud components to software. What developers gain with PaaS is a framework they can build upon to develop or customize applications. PaaS makes the development, testing, and deployment of applications quick, simple, and cost-effective.
Route 53	Route 53 provides local and global DNS look-ups
S3FS	s3fs allows Linux and Mac OS X to mount an AWS S3 bucket via FUSE. s3fs preserves the native object format for files, allowing use of other tools like s3cmd.
SaaS	Saas: Cloud application services, or Software as a Service (SaaS), represent the largest cloud market and are still growing quickly. SaaS uses the web to deliver applications that are managed by a third-party vendor and whose interface is accessed on the clients' side. Most SaaS applications can be run directly from a web browser without any downloads or installations required, although some require plugins.
Scrum	Scrum is an iterative and incremental agile software development framework for managing product development.
Spinnaker	Spinnaker is an open source, multi-cloud continuous delivery platform for releasing software changes with high velocity and confidence.
Spotify Tribe	Spotify's engineering and product organization is split into several large groups, called tribes. Multiple squads form a tribe. And Tribe Leads are the first level of full time management.
VPC	Virtual Private Cloud (VPC) isolated Cloud networks provide internal networks
WebDAV	Web Distributed Authoring and Versioning (WebDAV) is an extension of the Hypertext Transfer Protocol (HTTP) that allows clients to perform remote Web content authoring operations. Tori uses WebDav to transfer images to image servers.
XenServer	XenServer is the leading open source virtualization platform, powered by the Xen hypervisor. It is used in the world's largest clouds and enterprises.

Appendix 2. What is Zabbix

Zabbix is the ultimate enterprise-level software designed for monitoring availability and performance of IT infrastructure components. Zabbix is open source and comes at no cost.

With Zabbix it is possible to gather virtually limitless types of data from the network. High performance real-time monitoring means that tens of thousands of servers, virtual machines and network devices can be monitored simultaneously. Along with storing the data, visualization features are available (overviews, maps, graphs, screens, etc), as well as very flexible ways of analyzing the data for the purpose of alerting.

Zabbix offers great performance for data gathering and can be scaled to very large environments. Distributed monitoring options are available with the use of Zabbix proxies. Zabbix comes with a web-based interface, secure user authentication and a flexible user permission schema. Polling and trapping is supported, with native high performance agents gathering data from virtually any popular operating system; agent-less monitoring methods are available as well.

Web monitoring as well as monitoring of VMware virtual machines is possible with Zabbix. Zabbix can automatically discover network servers and devices, as well as perform low-level discovery with methods of automatically assigning performance and availability checks to discovered entities.

Why Choose Zabbix

There are many reasons to choose the Zabbix solution over its competitors. The best way to make sure it is the #1 choice for your organization is to give it a try. Before you start though, consider the following benefits of using Zabbix:

Zabbix offers the freedom of using an open-source solution with no vendor lock-in and freely accessible source code. This includes not only Zabbix itself, but also required components (Linux, Apache, MySQL/PostgreSQL, PHP) Zabbix setup and configuration is quite easy ensuring a low learning curve and therefore low cost of ownership

Highly efficient Zabbix agents for UNIX and Windows (x32, x64, Itanium) based platforms provide wider monitoring capabilities with greater speed. A centralized monitoring system allows to store all information (configuration and performance data) in a relational database for easier processing and re-use of data

Rich visualization capabilities allow to work with your data faster and smarter

Built-in housekeeping procedures allow to keep your data well organized

Appendix 3. XiTI tool by AT Internet

ANALYTICS SUITE

WEB ANALYTICS SOLUTION

The Analytics Suite is a web analytics solution designed to meet your specific needs, no matter your business type or model, job role, experience level or project. Measure and analyse your web and mobile traffic (mobile sites and mobile applications) to optimise your digital performance. The Analytics Suite brings together several web analytics applications that are simple to use and understand, for all types of users. Whether you're a beginner or an advanced digital analytics expert, you'll discover an intuitive experience with all our web analytics tools, web analytics dashboards, web analytics reporting and web analytics management. (XiTI tool by AT Internet, 2017)

REPORTS APPLICATION: WEB ANALYTICS FOR EVERYONE

With the Reports application, you can easily track analytics KPIs for all of your initiatives using ready-made business reports. Pre-built web analytics report templates mean you don't need to have sophisticated web analytics knowledge in order to make data-driven decisions.

DASHBOARDS APPLICATION: SIMPLE YET POWERFUL

Track and manage your websites, mobile sites, and mobile applications simply yet efficiently, thanks to AT Internet's Dashboards application. This application – an integral part of our web analytics solution – helps you create, customise and share web analytics dashboards within your organization, from sales and marketing teams to top management.

CONSULTANT TEAMS CLOSE TO YOU

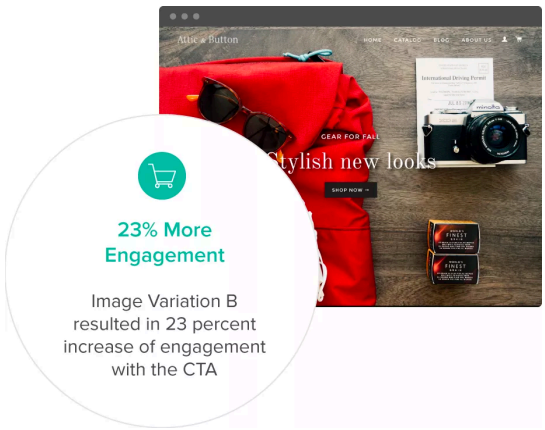
Much more than just a powerful web analytics solution, AT Internet is also a team of web analytics experts and consultants present on all continents who are ready to answer your analytics questions, train your teams on the Analytics Suite, and help you tap into the full potential of your web analytics data.

Appendix 4. Optimizely Tool

Experiment to uncover customer insights and create high-performing experiences

Increase engagement and revenue

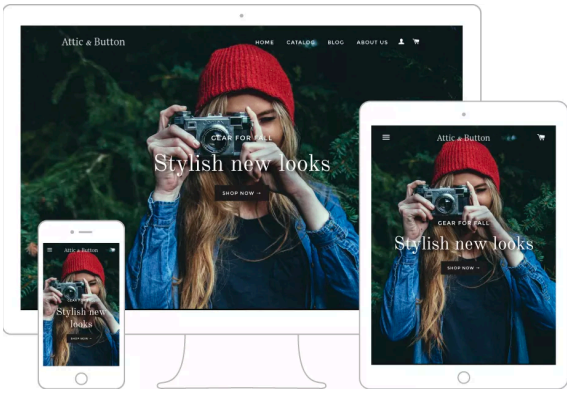
Robust, easy-to-use A/B, multivariate, and multi-page experimentation for any stage of the customer experience.



Experiment on any device.

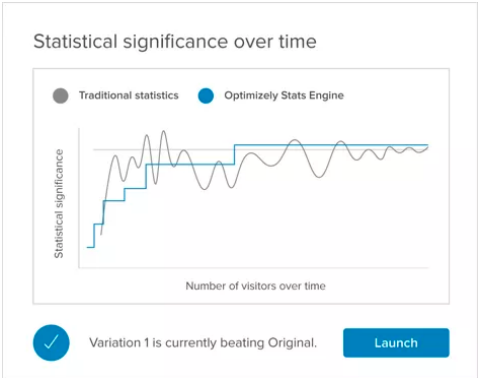
Optimizely X Web Experimentation supports optimized experiences across web, mobile web, iOS, and Android mobile apps. Take action on results with confidence

Optimizely's industry-leading Stats Engine helps you achieve the balance of making experiment decisions quickly with necessary statistical rigor.



Take action on results with confidence

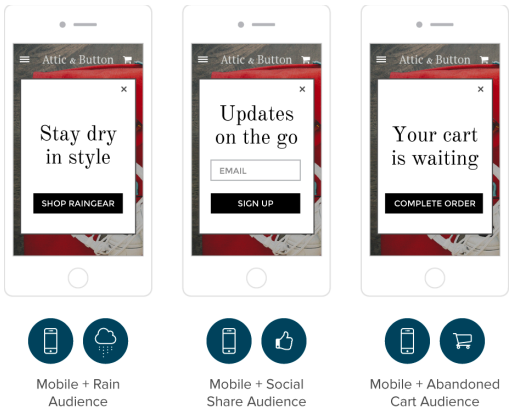
Optimizely's industry-leading Stats Engine helps you achieve the balance of making experiment decisions quickly with necessary statistical rigor.



Personalization lets you keep up with evolving customer demands

Deliver targeted content in real time

Translate what you know about your customers into revenue. Identify interesting behaviors with visual tags — no developers necessary. Connect



that browsing behavior, demographic information, contextual clues, and 1st- and 3rd-party data into a complete picture of your customer that you can use to power personalized experiences.

Make decisions backed by results you can trust

No army of analysts required. Understand the impact of each campaign you run across your audiences in real time with results powered by our industry-leading Stats Engine. Increasing average order value, reducing acquisition costs — whatever your goal, Personalization is designed to help you reach it with measurable results you can understand.