

5-2016

# Automation of Production Servers using DevOps (Puppet) Tools

Sailesh Kodali  
*St. Cloud State University*

Follow this and additional works at: [http://repository.stcloudstate.edu/mme\\_etds](http://repository.stcloudstate.edu/mme_etds)

---

## Recommended Citation

Kodali, Sailesh, "Automation of Production Servers using DevOps (Puppet) Tools" (2016). *Culminating Projects in Mechanical and Manufacturing Engineering*. 64.  
[http://repository.stcloudstate.edu/mme\\_etds/64](http://repository.stcloudstate.edu/mme_etds/64)

This Starred Paper is brought to you for free and open access by the Department of Mechanical and Manufacturing Engineering at theRepository at St. Cloud State. It has been accepted for inclusion in Culminating Projects in Mechanical and Manufacturing Engineering by an authorized administrator of theRepository at St. Cloud State. For more information, please contact [kewing@stcloudstate.edu](mailto:kewing@stcloudstate.edu).

# **Automation of Production Servers using DevOps (Puppet) Tools**

by

Sailesh Kodali

A Starred Paper

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfillment of the Requirements

for the Degreee of

Master of Engineering Management

November, 2016

Starred Paper Committee  
Dr. Hiral Shah, Chairperson  
Dr. Ben Baliga  
Dr. Kasi BalaSubramaniyam

## **Acknowledgement**

I would like to express my deepest gratitude to my mentors Dr. Ben Baliga, and Dr. Hiral Shah for all their support and encouragement at every step throughout my course of study at St. Cloud State University. I sincerely thank Dr. Hiral Shah for her valuable guidance and support throughout my Capstone project. I am extremely thankful to Dr. Balsy Kasi for having consented to be my committee member and spending time on my Project. I would like to thank the Engineering Management program for providing me with the necessary resources and an environment conducive to make my study possible.

Finally, I would like to thank my family and all my friends for constantly encouraging and supporting me in this endeavor. It is with their help that I have successfully completed my Master's Program in Engineering Management.

## **Abstract**

As the technology advances, organizations keep changing their business-related software applications. These applications were very important in daily business activities like marketing their products, selling their goods and services, billing customer and maintaining the inventory. Out of several areas of business, server maintenance is one of the sensitive department with which customers have direct link for money related issues. So, it is very important to maintain a robust automated system for any organization. Especially it is much more important in the Banking and Healthcare industry. Therefore, this project mainly focused on developing a robust automated system for such organizations and it brings all the necessary steps in the server maintenance under one roof. With this application, most of the billing related issues can be processed automatically which reduces the overall manual inputs and thus reduces the probability of causing errors.

## Table of Contents

I. Introduction .....	8
1.1 Problem Statement .....	9
1.2 Nature and Significance of the Problem.....	10
1.3 Objective of the Project.....	10
1.4 Project Questions .....	11
1.5 Summary .....	11
II. Background and Review of the Literature .....	12
2.1 Introduction.....	12
2.2 Background Related to the Problem .....	12
2.3 Literature Related to the Problem .....	13
2.4 Literature Related to the Methodology .....	16
2.5 Summary .....	19
III. Methodology.....	20
3.1 Introduction.....	20
3.2 Design of Study .....	20
3.3 Data Analysis.....	30
3.4 Budget .....	31
3.5 Project Timeline .....	31
3.6 Summary .....	32
IV. Data Presentation and Analysis.....	33
4.1 Introduction.....	33
4.2 Data presentation .....	33

	5
4.3 Data Analysis.....	34
4.4 Summary .....	38
V. Results, Conclusions, Inferences and Recommendations .....	39
5.1 Introduction.....	39
5.2 Results.....	39
5.3 Project Questions .....	40
5.4 Conclusion.....	43
5.5 Recommendations.....	44
References .....	46
Additional Works Utilized .....	47

**List of Table**

Table	Page
1. List of Technologies Used .....	22

## List of Figures

Figure	Page
1. Puppet framework .....	14
2. Puppet Architecture.....	15
3. Lean Development Flow.....	18
4. SDLC.....	21
5. Work Breakdown Structure.....	24
6. Defect raised in Bugzilla.....	28
7. Application data defects .....	34
8. Defect Arrival and Kill Rate .....	36
9. Defects in Code.....	37
10. Total number of errors from the application produced reports.....	38



## **Chapter 1: Introduction**

As the technology advances, customers expect better service from the organizations in many areas of the business. In this competitive world, any organization can sustain in its business not only by providing quality goods and services, but also by providing the better customer service. For this reason, organization keep looking for areas where they can improve their customer service levels. Not only the customer service, organization also need to improve their internal software applications in all the areas of business in such a way that it can reduce the cost to the company and which can provide a solution to various tasks. That means organizations need a centralized system for an area of the business, which can perform more number of operations at lesser amount time and with minimum amount of manual efforts.

This project was regarding one of the Fortune 500 health care services company based in Ohio. This company mainly focuses on manufacturing health care pharmaceuticals, surgical products. It also distributes medicines and other medical products to over 75% of the hospitals in the United States. It has 100,000 active operational units serving across the country currently with revenue of 102.5 Billion in 2015. It also serves many other hospitals across the world. This is all frontend which usually occurs on daily basis.

However, in the backend this company has thousands of servers and databases which were maintained by administrators and developers in order

to sync the daily activities worldwide with different time zones. The company has planned to undergo vast changes in maintaining these servers. Because, the customer base and the daily activities base has been increasing day by day which lead the management to take this decision. In this context the company has chosen DevOps(Puppet) tool which they thought suits best in their environment where servers across the world need to be synchronized automatically irrespective of time zones, tickets(issues), maintenance problems.

Puppet is an automation tool which was developed by Puppet Labs based in Portland, OR and basically it's an open-source configuration management tool, which came into limelight in recent years because of its robust and reliable technology. But why only puppet is preferred by the company over other tools will be discussed further in the project.

### **1.1 Problem Statement**

This company has several servers called production servers which were to be up and run round the clock, because those were the kind of servers which handle all the financial and website information. But the company has clients across the globe which is challenging. The time zones were different and people across the world use that information all the time. The major problem was shutting down the production servers during business hours which caused great impact on company's reliability and revenue in case any upgrades or maintenance had to be done.

## **1.2 Nature and Significance of the Problem**

The problem persisted since long time, however there were not much efforts focused on this. The company had done several projects till now, where the shutting down of servers and usage of man power were common. However, this was a new project where the company addressed the significance of the automation tool and how this tool improved the production time while reducing man power and time consumption at the same time. This project also explained how the tool was capable of syncing thousands of servers of the company across the world with very less usage of manpower and time resources.

## **1.3 Objective of Project**

- The primary objective of this project was to eliminate the shutting down of servers completely and to provide continuous access to clients all the time without any interruptions using puppet tool.
- Also, developing a solution which can be used in immediate future, that can significantly reduce the time and labor required to access information commonly used to plan inspections and maintenance. Several modules (code) were written on Linux platform for the project where the code was to be deployed on all the slave servers using this tool without having shutting down the servers. All the commands in the loops were included in those modules regarding inter connection of servers and responding to queries. The modules were included with all

the details regarding all users, their authentications, http connections, ssh-password less login details for administrators etc. This tool allowed the administrators to access remote servers in remote areas from their current working place.

#### **1.4 Project Questions**

After successfully accomplishing the project, the following potential questions will be answered based on the results obtained from the project:

- 1) What was the reason to select puppet automation tool over other automation tools by the company?
- 2) What improvements were observed in the company after implementing the tool?
- 3) How effective was the puppet tool in completing the server management project on time and within the budget?

#### **1.5 Summary**

This chapter gives a basic idea about the problem, objectives of the project, nature and significance of the project at the business, problems being solved by the project and the assumptions made throughout the development of this project. It also explains the technical abbreviations and other terminology used in this document.

## **Chapter 2: Background and Review of the Literature**

### **2.1 Introduction**

This section gives more detailed information about the background of the problem and the literature study. This helps the user to get better information about the problem and the studies made to propose this particular solution.

### **2.2 Background Related to the Problem**

The company has numerous clients located across the world and to meet their needs, several servers were established. The servers can serve millions of users at any given time provided, there were no enhancements or patches applied during business hours. However, for any machine, software or hardware there comes a time where they need some updates, upgrades or bug fixing processes. So, these important steps lead to some unexpected disturbances or problems for the customers in accessing the websites or their data. This eventually led to the industry downfall in their income and reputation. So, the company was forced to think about the alternatives which can significantly reduce the maintenance and bug fixing time. The company came up with an idea to automate these kind of tasks which can be performed automatically at the scheduled time where the users or clients do not encounter any issues while accessing their data. For the company people there were many software tools available in the market which can automate the server tasks and some of them were puppet, chef, open stack etc.

However, among all these tools, the company opted for puppet because of its low maintenance and minimal code requirement.

### **2.3 Literature Review Related to Problem**

Per Puppet Labs, Puppet is an automation tool which is currently being practiced by the Linux Administrators across the globe. This tool will synchronize all the servers in a company which might had many different servers across the world. To simplify, what this tool basically will do is, to create a master server in the main branch and makes all the rest of servers as slaves for that master server. These slave servers will be programmed in such a way that they all sync to this master server and would always be in contact with it. Whatever operations were done on slave servers they were monitored and recorded in master server. This master server controls the slave servers giving desired permissions and resources to them.

The beauty of this tool is, each slave server's data is unique and whatsoever it communicates with the master server, the rest of the slave servers will be denied from knowing it (Arundel, 2013). Likewise, each slave has its own boundaries which deny the rest of servers from knowing what's happening in them unless and until the administrator give access. Another good thing about the usage of this tool would be unlike java or other programming languages, thousands of lines of coding was not necessary in this (Rhett, 2013). Simply one needs to know how to write modules which comprises simple codes for each operation in console. So, even within

the company everyone has private access to the servers which helps in securing their details regarding their documents or passwords.

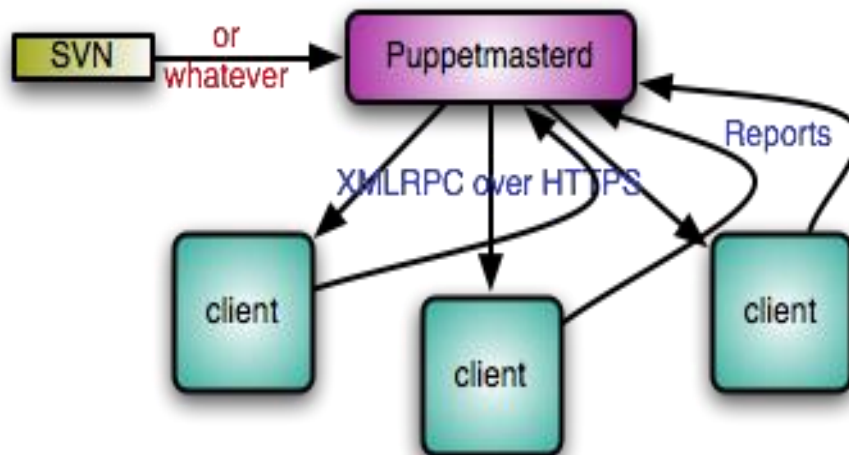
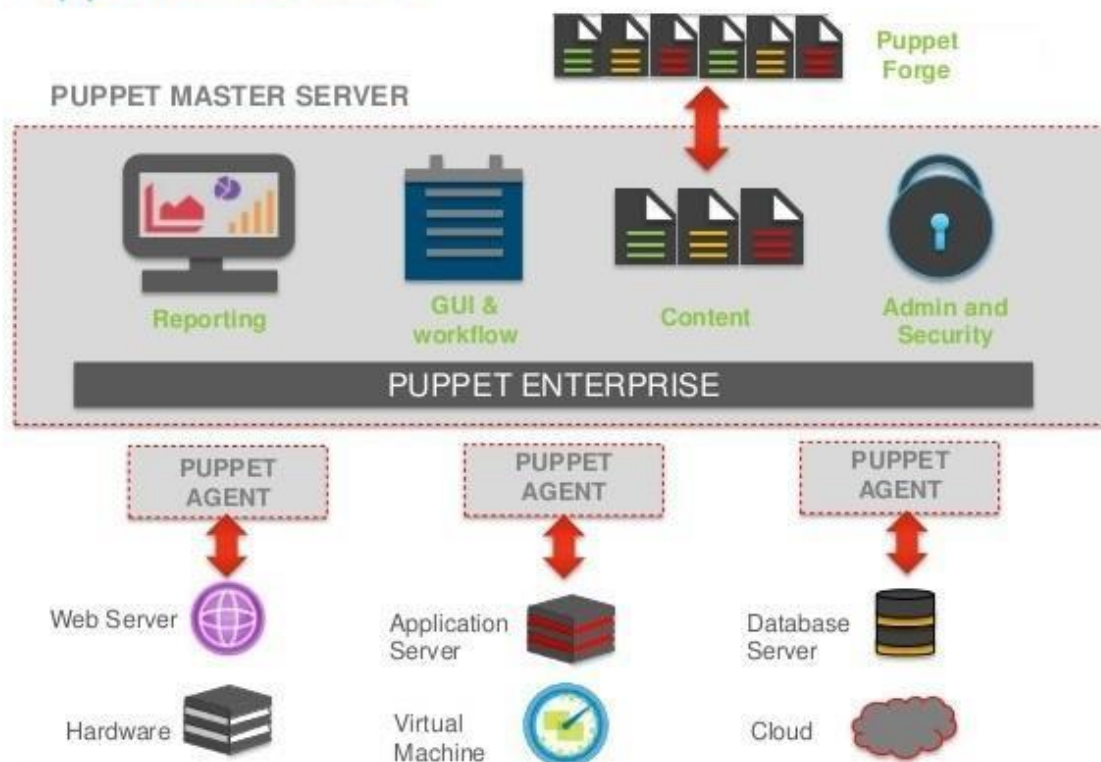


Figure 1: Puppet framework

Figure 1 describes that,

- **SVN** is the subversion console where the puppet script is written and the data is contained here. The data is deployed to master server
- **Puppet master** is nothing but a server which has all the admin rights and has a control over all the remaining servers which were declared as client servers. Usually there will be single master server for n number of client servers.
- **Client servers** were also servers which responds to the signals of master server and reports all the issues or activities to the puppet master.

## Puppet Architecture



*Figure 2: Puppet architecture*

*Figure 2* shows that,

The puppet master does several operations like securities, reporting, application management etc. All these actions were performed on servers per the instructions provided by the admin who is responsible for maintaining the servers. Oracle is the background database for the SAP system in this project. All the raw data and the data after performing operation on it will be stored in this database. Oracle database is the number one database among the SAP customers around the globe. It has a large and growing customer base that



gains long-term cost benefits from these two different companies' integrated software technologies. Organizations can run their developed SAP applications with the Oracle databases on the same code in different operating systems like Unix, Linux and Windows operating systems. Customers who are using SAP and Oracle as a background database receive support from both SAP and Oracle. With these integrated software technologies, organizations can create more innovative applications at lesser costs and increased efficiency.

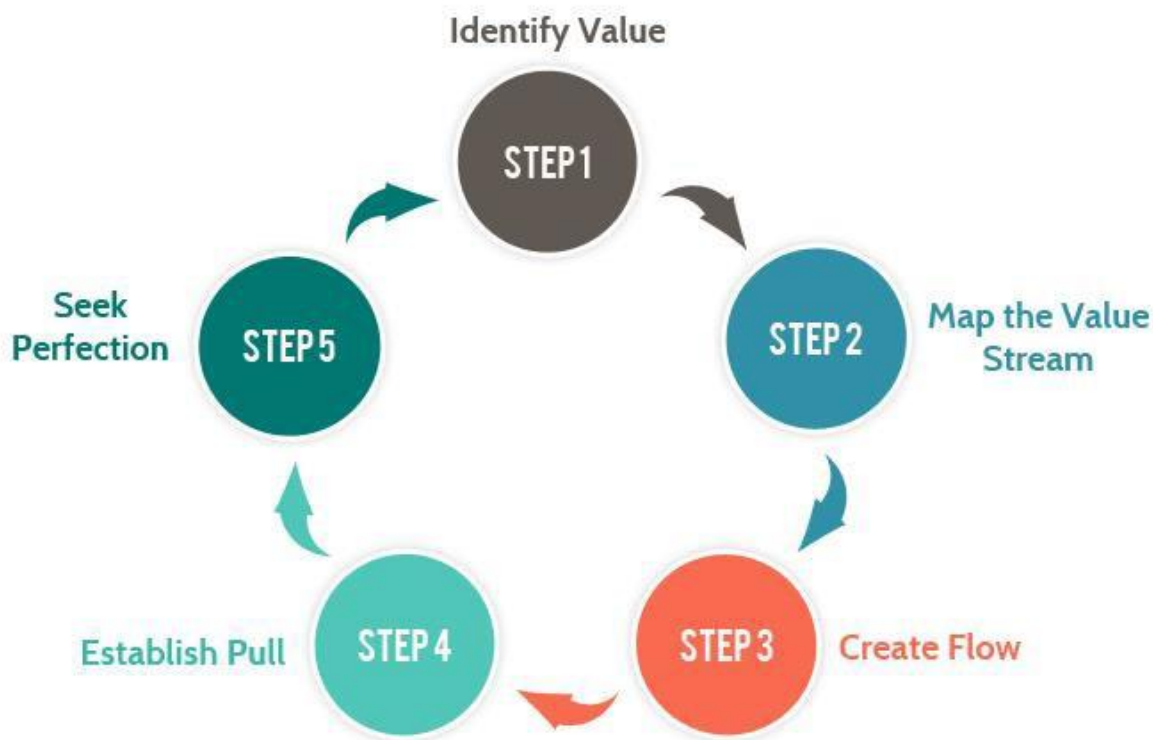
## **2.4 Literature Related to the Methodology**

Lean Development (LD) Methodology:

Lean Development methodology is employed for this project. As this project comes more under change tolerant software methodology suites this project perfectly. As this project involves dynamically changing software and up to date requirements application needs to be adjusted and modified. So this Lean Development focuses more on the dynamic stability which is similar Scrum embracing the controlled chaos. According to originator of this Lean Development, Bob Charrette, the measurable goal of LD is to build software with one-third the human effort, one-third the development hours and one-third the investment as compared to what SEI (Software Engineering Institute) CMM Level 3 organization would achieve.

12 rules which form the Lean Development were:

1. Customer satisfaction is the highest priority.
2. Providing the best value for the money always.
3. Regular interactions with customer and participation influences the success.
4. Team work and effort is the key to LD.
5. Everything is changeable.
6. Domain, not point, solutions.
7. Complete, don't construct.
8. An 80 percent solution today instead of 100 percent solution tomorrow.
9. Minimalism is essential.
10. Needs determine technology.
11. Product growth is feature growth, not size growth.
12. Never push LD beyond its limits.



*Figure 3 Lean Development Flow*

Common principles Lean Development follows per *Figure 3* are,

1. *Eliminate Waste*: In the software development industry, the most commonly identified wastages were unnecessary code or functionality, starting more than can be completed, delay in the software development process, unclear or constantly changing requirements, bureaucracy, slow or ineffective communication, partially done work defects and quality issues, task switching (Waters, 2010).
2. *Build Quality In*: One of the techniques used in this is **Pair Programming** which helps to avoid issues by combining the minds of

two developers to each specified task. So collective, combined experience of two developers can be used to yield better productivity and better solutions. Second technique is **Test Driven Development** which avoids most of the quality issues by writing tests before writing code. So basically, it is nothing but Test QA writing the test cases before the code is developed (Waters, 2010).

3. *Create Knowledge*: In software development one developer that works on everything and does the work productively creates a knowledge that no one can beat.
4. *Defer Commitment*: Decide as late as possible, is the key for this principle. This particularly applies for decisions that were irreversible, or at least will be impractical to reverse
5. *Deliver Fast*
6. *Respect people*
7. *Optimize the whole*

## **2.5 Summary**

This section gives the detailed knowledge about the problem background information and literature information about the software being used to develop the application and the literature information related to the methodology of the project. Detailed implementation of the project will be discussed in next chapters.

## **Chapter 3: Methodology**

### **3.1 Introduction**

This chapter gives detailed knowledge about how the project frame work and design were made. Implementation of the project and actual development of it is also discussed here. In addition to these, how the data was collected and time line of the project is also discussed here.

### **3.2 Design of Study**

Since it was a new tool which made its entrance into software market just few years back, every administrator needed to get aware of its importance and implementation techniques. For this batch jobs were created. Batch job was nothing but collecting all the data from the sources and running all the data at the night times with all the validations and store it into database. This process runs as a job once the batch creation is completed. Validations that were performed in this batch job before storing the data into database were as follows

- Was the data entered has all the required fields?
- Was the data format correct (Date field should consist of date format)?
- Was the time and date noted for the inspection?

The Key objective in all the phases of the project was to provide the reports more reliably because the processed data was directly accessed by the customer. Data with errors could give inconvenience to the customers for additional or wrong charges that were billed to them. So throughout the

process, the approach was a combination of both qualitative and quantitative approaches. So the data that needs to be processed was huge and results given were error free data to the customers.

**Approach followed:**

Life cycle used to develop this process was **systems development life cycle (SDLC)**, also referred to as the **application development life-cycle**.

This is a term used in software development systems and software engineering to describe a process which includes all the stages from planning, designing, developing, testing and deploying and application. SDLC concept can be applied to a wide range of software and hardware integrations and also to system consisting of both.



*Figure 4 SDLC*

As mentioned above in the *Figure 4*, methodology involves four different phases which has different purposes as follows.

### **a) Project Assignment and Roles Identification**

Initially this project was selected and a specific team was assigned to it. Later, the team members were given appropriate roles such as testers, administrators, by the team lead. Agile methodology was followed in this project where project manager assigned various roles such as team lead, business analysts, developers and testers to the team members. This completed the team formation and assigning project.

Different tools required for the project were identified at this stage and availability of those tools and also the cost effectiveness were identified. Then the intrigued individuals who were part of this project were identified and all the documented details were provided. So altogether end user was made very clear about the requirements and business team should be clear about the expected output. The tools involved in completion of the project were mentioned in Table 1.

<b>Tool</b>	<b>Version</b>	<b>Source</b>	<b>Used For</b>
MySql	8.0	mySQL	This database is used for data storage.
SVN	3.1	SubVersion	This framework is used for extracting transforming and loading data.
Puppet	4.6	Puppet Opensource	This script is used for coding.

JQuery	4.1	HP	This is used in the frontend to display the data and defects.
--------	-----	----	---

*Table 1. List of Technologies Used.*

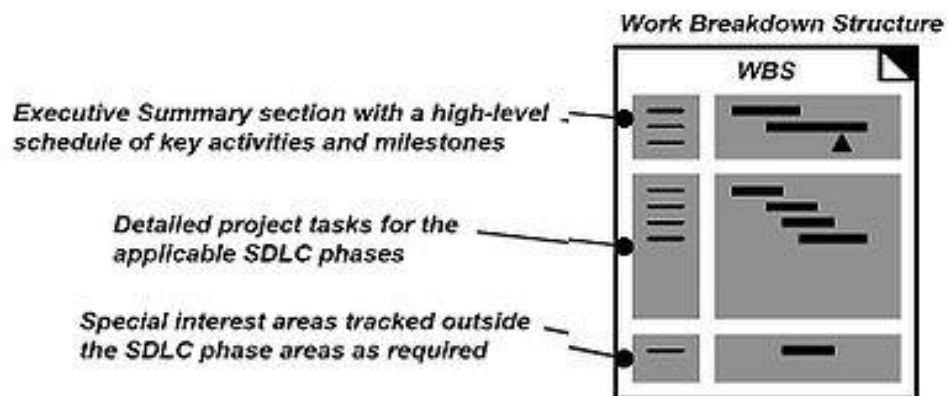
Hence the key activities included defining the scope and establish scope dimensions. Agile methodology was being followed in this specific project where project assigned various roles such as team lead, business analysts, developers and testers to the team members. This completed the team formation and assigning project.

#### **b) Initial Project Planning**

Initial project meeting was held to discuss various life cycles of the project with all the team members. This also included the design of layout of the project and design of the database to store all the data needed for this. Appropriate and quality data is very much essential for this project as discussed in the data analysis stage. Hence validations for the data were planned well advanced to rectify the before it is stored in the database. Gaps in the data were also identified and is essential to achieve project bounded deadlines and its quality. This project was done in two week sprints where a daily scrum meeting was conducted for all the team members every morning to discuss the tasks they had accomplished yesterday, obstacles they were facing today, and tomorrow's goals. At the end of every scrum project was presented to



the business to compare the requirements and output. Work was broken down into small parts and tasks were assigned to team members for every sprint. Work break down structure was designed by manager. Following was the work break down structure planned.



*Figure 5 Work Breakdown Structure*

As shown in *Figure 5*, A work breakdown structure is a key project deliverable that organizes the team's work into manageable sections like high-level schedule of key activities, project tasks, out of the box issues which usually occur in SDLC process. The Project Management Body of Knowledge (PMBOK) defines the work breakdown structure as a "deliverable oriented hierarchical decomposition of the work to be executed by the project team."

The work breakdown structure had several benefits in addition to defining and organizing the project work. The project budget was allocated to the top levels of the work breakdown structure, and department budgets was quickly calculated based on each project's work breakdown structure. By allocating

time and cost estimates to specific sections of the work breakdown structure, a project schedule and budget was quickly developed. As the project executed, specific sections of the work breakdown structure was tracked to identify project cost performance and identify issues and problem areas in the project organization.

### **c) Requirements Gathering**

Post the data was collected the project started with designing a database which stores all this data. Designing a database was comprised of creating tables and developing a relation among them. Once the database design was done the functional lead, business analysts, project manager and development team met to check whether the customer specifications to develop the web application could be developed within estimated time and project manager assigned tasks to the development team. Various software, hardware and tools that will be required to develop the software were identified; manage and track the progress of the web application development.

### **d) Requirements Analysis**

Once all the data was collected the project started with designing a database which stores all this data. Once the database design was done, functional lead, business analysts, project manager and development team will meet to check whether the customer specifications to deploy the tool and could be developed within

estimated time and project manager assigns tasks to the development team.

### **e) Code Design**

Since this was a tool, there will not be much complicated tasks in designing a code for this project. A simple code which lasts only few lines was deployed. However, each member in a team was assigned with a module where he needs to develop a simple code for that module which was nothing but a specific task which manages the servers.

### **f) Code Testing**

Each developer performs unit testing before they check in their code or commit their code. After every developer had finished their part of coding, all these pieces of codes were integrated together. After integrating the code, it was tested again as a whole module to find if there were any complications in the functionality matching. When the code was deployed into test environment QA and testing team came into picture where they perform aggressive testing. Each and every team member's code must be checked before they were actually delivered. For User Interface Selenium IDE testing and for code JUnit testing has been employed. Generally, development and testing were done parallel in order to eliminate delayed code errors, bugs and functionality mismatch.

System testing had been performed which involves both the

functional and non-functional testing.

Functional testing:

This was the initial testing where risks were analyzed based on the requirement specifications. Six steps involved in the Functional testing were

1. Expected output per the business was identified.
2. Based on the requirement specifications, input data creation was identified.
3. Output determinations per the functional specifications.
4. Already prepare test case execution.
5. The comparison of actual and expected outputs
6. Customer needs were verified as per specifications.

So finally, with the functional testing what the developed application does was verified and answered.

Non-Functional testing:

This was the second type of testing where the developed software application was tested for its non-functional requirements. These include how the application operates rather than how the application behaves. In functional requirements application was tested to check whether it meets the specific functions described or specified by the user. This non-functional testing was exact opposite to functional testing. So, this included reliability, scalability, load testing, performance testing, stress

analysis, portability testing, load balancing, high availability, robustness and even more. So, with this non-functional testing the main answer going to be answered was how the application works.

Code was developed and deployed to test environment where the testing takes place. So there was a scope of having different set of data or different environment properties between the Test and Developer environments. So both the environments should be as much similar as possible. Test team was responsible for testing and writing the test scripts. Test team logged the defects as they found them and assigned them back to developers for fixing them. Below was the *Figure 6*, showing the defect being logged.

Summary								The invalid file format error message for invalid photographs is appearing after the upload is complete							
Submitter		vijay s		Affected Release/Build		Release 3		Severity		Average		Fixed Release/Build			
Submitted Date		2/27/2014 12:41:28 AM		Priority		Medium		Root Cause		Other		Module		qConnect - Sample Project Module root	
Assigned To				Status		New		Type		Bug		Target Release/Build		Release 3	
Reason				Category		Implementation - Logical Defect		Target Date				Closed Date			
Environment		Windows XP + Firefox + Staging													
Description															
<ol style="list-style-type: none"> <li>1. launch OrangeHRM and login as a employee</li> <li>2. Click on the photograph displayed at the top left corner of the personal details page</li> <li>3. Click on "Choose a file" button and choose a ".exe" file that is less than 1 MB</li> <li>4. Click Upload</li> </ol> <p>Expected result: An error should be displayed that it is an invalid file format.  Actual Result: The message of 'Invalid file type' should appear (as only JPG, GIF and PNG files supported) after the Upload button. But it uploads the complete exe file and then gives a message that the file type is invalid.</p>															
LINKED TEST RUNS								ATTACHMENTS							
								<p>Click here to pick from a folder (Maximum Size: 50MB)</p> <p>OrangeHRM issue 2.jpg</p>							
© www.SoftwareTestingHelp.com															

*Figure 6 Defect raised in Bugzilla*

### **g) Improve Phase**

After the QA team found the defects and logged them, they were assigned back to the developers responsible for that. At this stage initially it was assessed whether the defect was due to the code or test error or environment issue. Once it was determined development team worked on fixing the bug and reassigned it back to the QA team for retesting.

After the Test environment, there were two other environments this application was tested. They were Pre-production and Production stages. This gave ample scope to the developers and testers to improve and get the error free application.

So, to have the right decision made, testing team employed small testing cycles called "PDCA" or Plan Do Check Act. This helped in redefining the ideas of both the developers and QA team and collecting the feedback from the business. To find out whether these small improvements were effective and reliable these small cycles were the great way.

To have these improvements getting successfully implemented careful planning was required. So here both the developer and QA teams were required to have the frequent communications, training and planning as well. The more time the team spent on planning, the faster they reached the total adaptation to the improvements by their process.

## **h) Deployment and Maintenance**

As per the customer requirement, once the basic functionalities of the web application were developed, and all the defects raised by the QA team were fixed and the web application software was deployed and the web application went live so that it can be used by the end users of the client. In case the client experienced any errors or bugs while using the web application, these were reported to developers/testers. The assigned developers and testers took the responsibility to solve the issues. Once high priority issues were solved, software releases were made to affect the changes made to the web application.

### **3.3 Data Analysis**

All the collected data after the validations were done were stored in a MySQL database. Application which is developed analyses the data using different techniques per the business requirements and project the data in a user readable format. Data entered the tables represent a model object and its properties. So SVN is the tool used to retrieve the data in the form of an object and display it per the requirements. To analyze the gathered data, data analysis techniques were performed.

Data mining technique has been used to model the data rather than describing the data. BI (Business Intelligence) team focused more on the predicting the data outcome rather than purely describing the data. Two types of analysis were performed EDA (Explorative Data Analysis) and CDA

(Confirmatory data analysis). Main difference between both techniques is EDA concentrates more on discovering the new features in the data and CDA concentrated more on the analyzing and confirming the existing data hypotheses.

Data collected from the both the sources as mentioned above goes into the same place before going through different validation checks. Data cached here will be process through a nightly batch jobs that run throughout the nights and puts them into the database in the required format. Both these techniques were then used to retrieve the data to transform it into a simply understandable form, habitually with visual graphs, without using any statistical model or a formulated hypothesis.

### **3.4 Budget**

All the costs that were incurred during this project like licensing, server costs, equipment costs, database installation and handling costs, resources, maintenance and other miscellaneous costs were covered by the company.

### **3.5 Project Timeline**

The following is the anticipated timeline to complete this project:

Project Idea Formulation and Research.....	January 2015
Project Proposal Write-up.....	February 2015
Gathering the requirements.....	March 2015
Analyzing the requirements.....	March 2015
Designing the code.....	April 2015



Testing the developed code.....	June 2016
Deploying the tested code.....	July 2016
Evaluating the Agile Methodology.....	July 2016
Project Write-up.....	August 2016
Final Defense.....	Nov 2016

### **3.6 Summary**

This section discusses about the study made prior to the start of this project and the methodology followed in implementation of this project. It also discusses about the collection of the data on which this application is going to run and methods, tools used to analyze the data collected.

## **Chapter 4: Data Presentation and Analysis**

### **4.1 Introduction**

This chapter explains about the data presentation and analysis of the data that is being presented. Data presentation is a key aspect of this project. It is very much important to arrange the data in meaningful manner to portray the analysis of the presented data. With the help of the presented data users will be notified of their scheduled inspections and patches on the servers. In this chapter, it is clearly explained that how the data is presented and what way the data has been analyzed.

### **4.2 Data Presentation**

As discussed in the above chapter data has been collected in two ways. First one is through the manual process. In the developed application there is a page with all the required fields to be entered and also validations done automatically on the front end side of the application. Below is the image showing the page where details of the servers were entered and validated.

In one part of the screen fields were present to enter the inspections details performed on the servers and on the second part of the screen there were fields present to enter the modification details for that server. For each inspection, there is a possibility of entering the defect related to the inspection. After entering the details there is an option for saving these fields into the database.

The screenshot shows a software application with three main panels:

- Scheduled Inspections Table:**

	Subdivision/Yard	Asset Type	Asset
✓	SUBDIV_1	Main Track	MAIN-MA
✓	SUBDIV_1	Main Track	MAIN-MA
✓	SUBDIV_1	Main Track	MAIN-MA
✓	SUBDIV_1	Main Track	MAIN-MA
✓	SUBDIV_1	Main Track	MAIN-MA
✓	SUBDIV_1	Main Track	MAIN-MA
✓	SUBDIV_1	Turnout	EAST TR SWITCH LAKE
✓	SUBDIV_1	Storage Track	TRACK 1 LAKE-BA YRD
✓	SUBDIV_1	Storage Track	TRACK 2 LAKE-BA YRD2
✓	SUBDIV_1	Turnout	WEST TR SWITCH LAKE
✓	SUBDIV_1	Storage Track	OLD WESTBC MAIN PA OLD MAI
✓	SUBDIV_1	Spur	SKUNK HOLLOW ST 1 OR 15
- Inspection Details Form:**
  - Asset: SIDING LOGAN-LOG SDG
  - Start MP: 456.4, End MP: 458.1
  - Asset Type: Siding
  - Subdivision/Yard: SUBDIV\_11
  - Inspection Reason: Regulatory
  - Earliest Date: 01/25/2011, Latest Date: 01/29/2011
  - Inspection Type: Regular
  - Completed Date: 05/09/2013
  - Entering Own Inspection:
  - Inspected By: LEVEL 1 LAST, LEV
  - Entered By: LEVEL 1 LAST, LEVEL 1 FIRST
  - Accompanied By: [Empty]
  - From MP: 456.4 - 456.4, To MP: 458.1 - 458.1
  - Inspection Mode: Observe
  - Comments: test
- Defect Details Form:**
  - Select Defect: New FRA Regulatory
  - 1. Subpart: C-Geometry
  - 2. Section: -- Select --
  - 3. Reg Code: 213.053 - Gage
  - Defect Value: 213.053 - Surface
  - MP: (456.4 - 458.1)
  - Repeated:
  - Defect Date: 05/09/2013
  - Immediate Action: -- Select --
  - Correct Before Date: [Empty]
  - Defect Comments: [Empty]
  - Status: Open
  - CORRESPONDING CORRECTIVE ACTION:
    - Corrected Date: [Empty]
    - Corrected By: [Empty]
    - Corrective Action: -- Select --
    - Corrective Action Comments: [Empty]

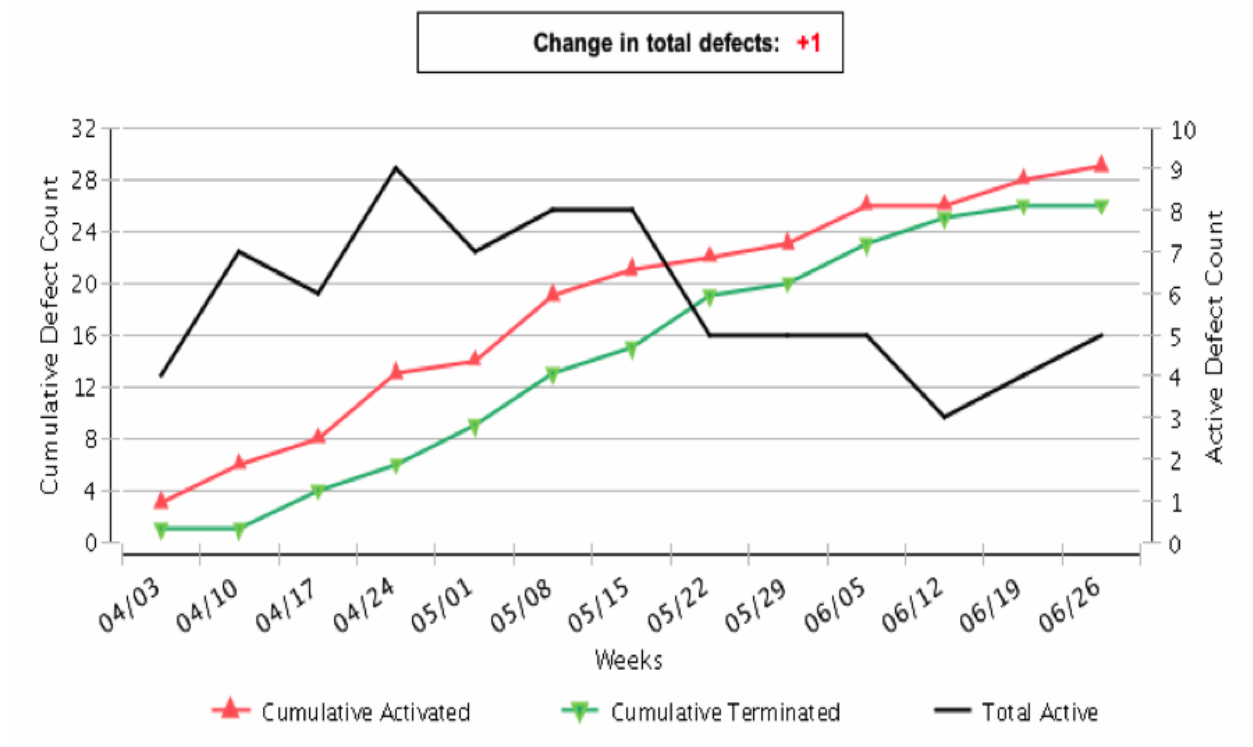
Figure 7 Sheet for entering the data

Once the data is collected this is displayed per the inspection or defect or schedule selected as shown in Figure 7. Once any one of these is selected details about that will be displayed in the tabular format. This table includes division or sub division where the defect or inspection occurred, scheduled timeline, asset number and location of the detail in terms of mileposts.

### 4.3 Data Analysis

In this stage, Data collected using various means by using different techniques and by using different methods were divided into small pieces of data where the application can analyze and made an inference. As seen above application

uses the collected data to report the results to the user and have achieved an unmistakable part to accomplish evaluated results. Following Business and user requirements were the key things responsible to achieve these results. Below was the chart showing the defects found from the server after the tool has analyzed the data and scheduled enhancements. This was in the time frame of April through June. Red bar indicates the total cumulative defects activated and green bar indicates total cumulative defects terminated. Black bar indicates the total number of cumulative defects still active. This was high in this time frame as the web application was in development and testing stages. Hence total number of defects have been increased in this time frame.



*Figure 8 Application data defects*

After this time frame when the application was moved to production stage after all the defects in the script were closed there has been a lot of improvement. Defects that are shown in *Figure 8*, are produced in the application and the kill rate after each defect in the code is fixed. This is in the time frame of month July through August.

As indicated in *Figure 9*, Red bars indicate the defects arrival rate and green bars indicate the kill rate of those defects. At the end of month August kill rate has been too high and literally there were no new defects raised. These were defects from the results produced by the application after analyzing the data and producing the output in the server.

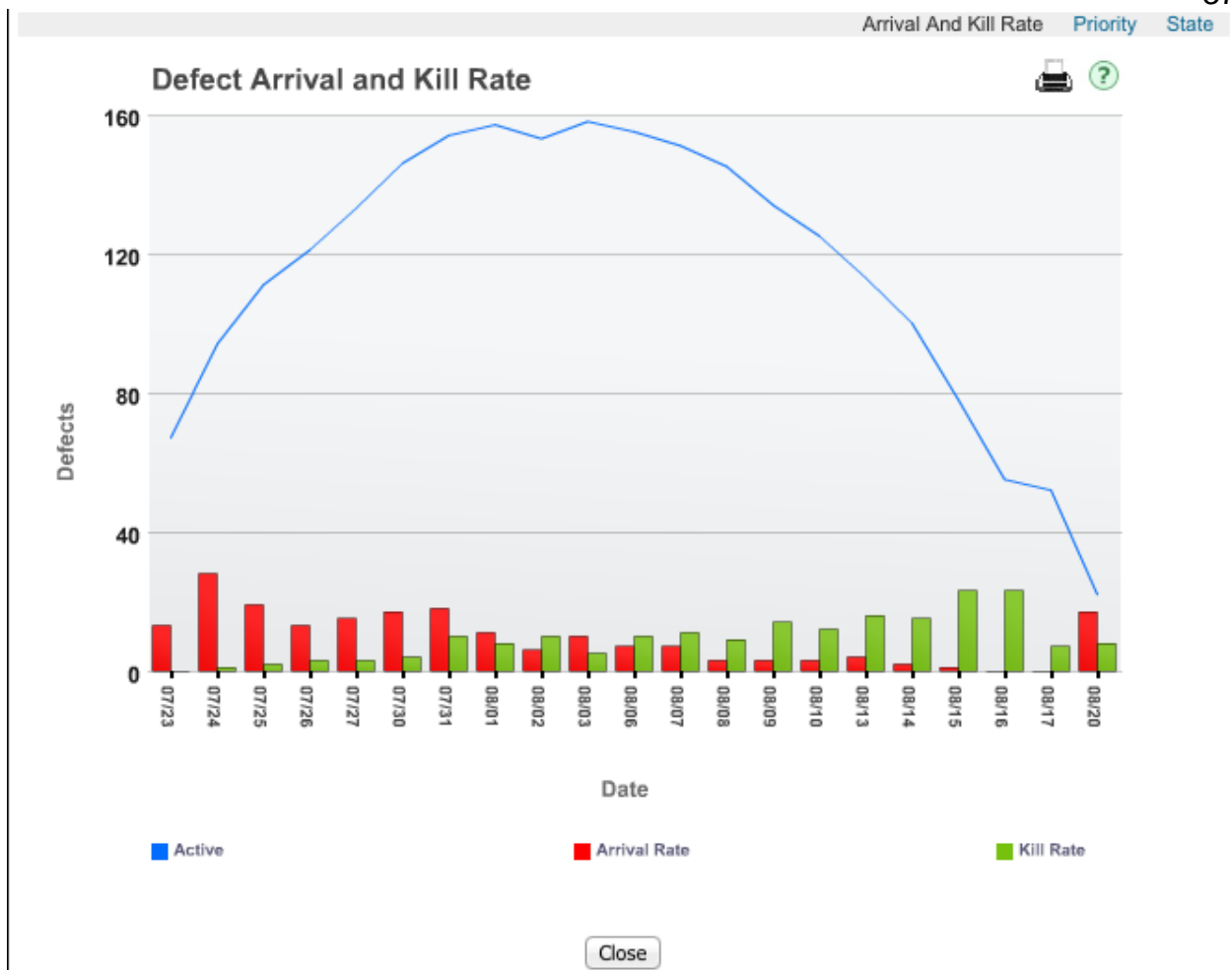
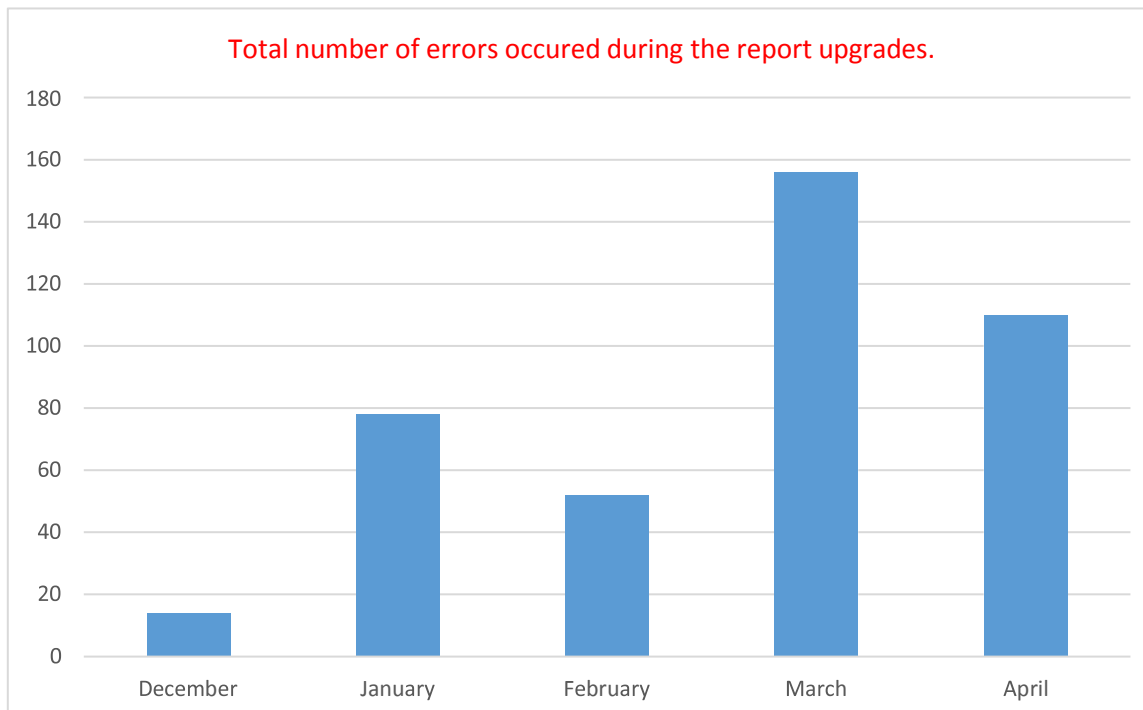


Figure 9 Defect Arrival and Kill Rate

### Report analysis generated by the application:

Figure 10 shows the total number of errors occurred on different month's data during the testing process. These were the errors occurred from the reports generated by the application monthly wise. April is the month where the code has been to UAT testing where all the defects has been rectified as they were raised and generated new reports.



*Figure 10 Total number of errors from the application produced reports*

#### **4.4 Summary**

This chapter explains the way how data flows in the business and how this project can be used to analyze the company's data. This automated reports which runs as jobs can generate these results at given point of time. Next chapter discussed about the results achieved by this project at the client location.

## **Chapter V: Results, Conclusions, Inferences and Recommendations**

### **5.1 Introduction**

This chapter discusses the results obtained from the data collected and effectiveness of these observations for the organization by implementing this project. This chapter also discusses the recommendations and conclusion of the project and how this project answered questions raised at the initial stages. Potential future works and recommendations for the extension were also discussed.

### **5.2 Results**

At the end of this project, robust web application has been developed which handles the huge chunks of data which can be automatically fed into servers without putting down the production time. This application is designed in such a way that it can be operated from a company system or the personal laptop from where the admins can enhance right from their home. This is because admins need to enter the data around the clock. So, this application acts as an interface for various platforms and yet stores the entered data into the specified database. Again, by using this data this application transforms it into graphs, charts and diagrams and notifies users about the servers and their properties. So, this application helped the organization in eliminating various legacy systems which were being used to perform the tasks independently



and manually. This application brought all the applications under the same roof.

### **5.3 Project Questions**

*1) What was the reason to select puppet automation tool over other automation tools by the company?*

Automation helps the decision makers in insurance industry includes underwriters, actuaries, financial institutions, agents and the customers themselves and so the insurance companies need high quality information consistent across all levels. This need a very high quality business solution at the company's disposal available always to operate more efficiently, manage risk and to increase their revenues. Further, these companies need to account for risk factors, natural and human disasters to rising inflation and medical costs, and all this should be hand and available instantly. This information is available in the form of data as a complete and powerful automation solution which insures the company's financial security through proactive risk management and growth strategies. Puppet was the only tool which required minimum code designing and was more concentric into scripting. Also, the maintenance of this tool was highly cost effective and simple. In computing, Puppet was an open-source configuration management tool. It runs on many Unix-like operating systems as well as on Microsoft Windows, and includes its own declarative language to

describe system configuration. Puppet provided a standard way of delivering and operating software, no matter where it ran. With the Puppet approach, company could define what they wanted their apps and infrastructure to look like using a common easy-to-read language. From there they could share, test and enforce the changes they wanted to make across their datacenter. And at every step of the way, the company had the visibility and reporting that they need to make decisions and prove compliance. Considering all these factors, the company selected puppet to automate their servers.

2 *What improvements were observed in the company after implementing the tool?*

Before adapting to this automation step, the company made huge investments on labor and time. However, there were drawbacks like shutting down the servers for making any enhancements, bug fixes, data backups, etc. Eventually, the time line of these steps went higher with version changes and the income of company gradually got affected. People also started raising their complaints of server unresponsiveness sometimes when they were in the middle of their work. In the olden scenario, billing process must collect data from multiple systems. Data collection, analysis and reporting were done by different systems, because of these multiple systems it used to take three times more time than the current developed application. As the same system is doing

multiple tasks, thus reducing the cost of the entire billing process. Here in the current application majority of the work is done by a more reliable system with possible less number of errors. Even though system gets some errors, there are back up mechanism like batch verification which helps the entire billing process to achieve error free centralized system. For addressing these issues, the company decided to automate their servers. After automating, the servers were never actually turned off for any updates or upgrades. Whenever there is minimal usage of the servers, the tool automatically runs the jobs as per pre-scheduled timings every alternative day without manual inspection. This helped to manage time and money to depend on labor which ultimately reflected in profit increase for the company.

*3) How effective was the puppet tool in completing the server management project on time and within the budget?*

The puppet tool was very effective in completing the server management project on time and within the budget. Considering its low cost maintenance and the jobs performed by the tool is significantly noticeable. The tool brought all the servers and their tasks under one roof and made the tasks simpler for the administrators to maintain the servers with more care and reduced the chaos. The tool also had long sustainability and could be updated occasionally which was be taken care by administrators. The tool was also free of cost in the market to

use and which made companies to adapt it and the creators of puppet tool provided free technical assistance for companies for certain amount of time to show the characteristics of this tool to the rest of the corporate world.

#### **5.4 Conclusion**

To achieve this automation robust scripting language was required which can be run on any platform. Hence puppet tool was employed for this which can be written in one language and run on any operating systems. All the logics, calculations and conditions were coded in this language which gives the desired input based on the input given. So, the data was processed in this language. Next step was to present the processed data. So, for this front end tools such as JQuery is selected to display the data in the required or desired format. MySQL database is used to store the vast amount of data from servers. With the help of all these technologies the company could regularize the process of automation and to reduce the extra labor. With the help of this insight with in the industry was increased with which they can increase their profits, identify and decrease potential and ongoing losses and comply with the industry regulation and standards. This tool has many capabilities that can handle vast amount of data from different sources. As all the technologies used are common one there is always a scope of expanding its capabilities in the future. There were very limited tools in the present market which can automate the servers thereby reducing the production shutdown time and

putting company reputation at stake. So, the company had to choose wisely among the limited options regarding which tool can bring the best performance with low budget and time investments. Puppet was one of the best tools available currently to which this company has adapted. Without a surprise, this tool managed to automate all the servers and made the tasks easier for the administrators and clients who can now access the servers and data more securely around the clock. To conclude, after utilizing this tool the company servers were working at full performance without getting halted for any kind of updates or patches. Every single minute thousands of users access these servers now and they were served to the fullest extent.

## **5.5 Recommendations**

This project mainly focuses on developing reports and scheduling inspections using the application developed with the puppet tool and MySQL as background database. Here were few recommendations to implement similar kind of project.

- Data was collected from different means and data stored in the database was always validated before it is stored.
- Users get hold of the application and its developments at the end of every sprint and provide their inputs if there were any changes.
- Data collected was stored in the database after it is validated. It was done by nightly batch jobs which should run every night to keep up the updated data.

- All the environments where the application was running should have similar set of data and configurations. Any changes to one environment should be notified and changes should be done in other servers as well.
- Scheduled patches should always be performed and should be entered into the system once the inspection is done to schedule next one.
- Most calculations done were in the puppet tool. So they should be done at the back end to promote faster responses.
- Any questions with the application user can go to help provided in the applications. Server related issues can be directly logged into the system.
- The code for this tool was more reliable and less cost effective, as it was an open source and can be run on any platform which mean run time may be lesser than the custom code because the standard code gone through various levels of performance tuning.

## References

- Arundel, J. (2013) *Puppet 3 Beginner's Guide*. Retrieved from <https://play.google.com/books/reader?id=Sbkr6mgFzAC&printsec=frontcover&output=reader&hl=en&pg=GBS.PA1>
- Rhett, J. (2013) *Instant Puppet 3 Master*. Retrieved from [https://play.google.com/books/reader?id=AtAXTOHc2v4C&printsec=front cover&output=reader&hl=en\\_GB&pg=GBS.PT2](https://play.google.com/books/reader?id=AtAXTOHc2v4C&printsec=frontcover&output=reader&hl=en_GB&pg=GBS.PT2)
- Waters, B. (2010). *Lean Principle #1 – Eliminate Waste*. Retrieved from <http://www.allaboutagile.com/lean-principles-1-eliminate-waste/>

### **Additional Works Utilized**

- Garvey, P. (2008). *Analytical Methods for Risk Management: A Systems Engineering Perspective*. Boca Raton, FL. Francis Group, LLC.
- Ponnaiah, P. (2007). *Data Modeling Fundamentals: A Practical Guide for IT Professionals*. New Jersey, USA. John Wiley & Sons, Inc.
- Schwartz, B., Zaitsev, P., Tkachenko, V., Zawodny, J., Lentz, Arjen., Balling, Derek. (2008) *High Performance MySQL: Optimization, Backups, Replication, and More*. Sebastopol, CA. O'Reilly Media, Inc.
- U.S. House of Representatives (1999). *Systems Development Life-Cycle Policy*.
- Ward, A., Sorbek, D. (2014) *Lean product and process development*. (2nd Ed.) Cambridge, MA. Lean Enterprise institute.