**Project Management Plan**

**DXC Technology Managed Services For Workloads On Public Cloud**

****

SE 4485.001

Dr. Weichen Wong

***Group 2:***

*Ihfaz Tajwar*

*Zunayed Siddiqui*

*Po-Yu Liu*

*Yeswanth Bogireddy*

*Zachary Tarell*

***Abstract****:*

Service providers are constantly challenged to accept customers' workload as-is and manage it on behalf of the customer. The management can range from monitoring, patching, and security aspects of the workload. The security could be shut off certain ports or setup security scans for Denial of Service attack, etc. The monitoring could leverage the cloud provider monitoring but would alert based on certain rules. The patching could use cloud provider specific to patch from vulnerabilities, routine service packs, etc. Initially the workloads may be un-touched as in refactored ([7 R’s](https://dev.to/wingkwong/7-strategies-for-migrating-applications-to-aws-131d)) to reduce the timelines but over time it might get refactored over time to reduce costs or increase stability and performance. Here is our teams Project Management Plan to aid DXC Technology in all the aforementioned duties of our service provider for their company.

Contents

[List of Figures 3](#_Toc50037449)

[List of Tables 3](#_Toc50037450)

[Introduction 4](#_Toc50037451)

[PURPOSE AND SCOPE 4](#_Toc50037452)

[OVERVIEW 4](#_Toc50037453)

[STRUCTURE OF THE PLAN 5](#_Toc50037454)

[Project Organization 5](#_Toc50037455)

[Life Cycle Model Used 5](#_Toc50037456)

[Risk Analysis 6](#_Toc50037457)

[RISK MANAGEMENT PLAN 6](#_Toc50037458)

[Hardware and Software Resource Requirements 7](#_Toc50037459)

[HARDWARE 7](#_Toc50037460)

[SOFTWARE 7](#_Toc50037461)

[Deliverables and Schedule 8](#_Toc50037462)

[Monitoring, Reporting and Controlling Mechanisms 8](#_Toc50037463)

[ Work Breakdown Structure (WBS) 8](#_Toc50037464)

[ Requirements Traceability Matrix (RTM) 8](#_Toc50037465)

[ Control Chart 8](#_Toc50037466)

[ Review and Status Meetings 8](#_Toc50037467)

[Professional Standards 9](#_Toc50037468)

[Evidence the Document has been placed under Configuration Management 9](#_Toc50037469)

[References 10](#_Toc50037470)

# List of Figures

[Figure 1 – Diagram of makeManaged script and how it communicates with cloud services 4](#_Toc50036100)

[Figure 2 - Waterfall Methodology Diagram and how it is implemented 6](#_Toc50036101)

# List of Tables

[Table 1 – Skeleton Table of Risk Management Plan 6](#_Toc50036297)

# Introduction

Our team has been tasked with implementing a makeManaged (Fig. 1) script for DXC Technology to monitor, make patchwork, and manage security aspects of their workload. This project management plan will include all of the aspects from beginning to end on how our team will create and implement DXC Technology’s goal from our script. Even though the requirements list out three different phases of this project, our sponsor has made clear that the minimum viable product will only need to contain the monitoring aspect of our project and if time permits move onto patchwork as well as security, load balancing, and firewall monitoring.

A screenshot of a cell phone

Description automatically generated

[Figure](#_List_of_Figures) 1 – makeManaged to Cloud Services

## PURPOSE AND SCOPE

The purpose of monitoring software through Infrastructure as a System (IaaS) is to benefit many different companies that need help desk support, patchwork, and security. Since SaaS and PaaS are already managed by cloud providers, we will only be using IaaS. The scope will range from a working monitoring script able to communicate with a cloud service and relay information, notifications, and patchwork to DXC Technology. This will lighten the workload of their employees by automating a lot of tasks and opening up free time to allocate resources elsewhere.

## OVERVIEW

A brief overview of the makeManaged program will be to use a virtual machine (VM) to communicate to AWS, Azure, and/or GCP cloud services over IaaS to monitor and the help desk and send notifications and email alerts or patch any necessary fixes.

## STRUCTURE OF THE PLAN

* Create AWS or Google account to get our VM
* The monitoring tool will use open source - Nagios/Zabbix
* Install the monitoring tool in our cloud which will be Agentless Monitoring
* Work in API which will be barebone and then add dumb UI (time permitting)
  + - OS monitoring - use case (runbook remediation - what to do with it, i.e. call an action or any abstract thing like write notification)
    - Software monitoring - using JAVA  and the VM will use name only keys and no user passwords
    - Most likely no User Interface (everything will run through the VM to push automation to it)
* Make notification and send through email
* Create an incident and use serviceNow or Remedy Software
* Patching and Security (time permitting)

# Project Organization

The team consists of five members.

Po-Yu Liu: Group Leader/Project Manager

Zunayed Siddiqui: Front-end Developer

Yeswanth Bogireddy: Back-end developer

Ihfaz Tajwar: Testing and Quality Assurance

Zachary Tarell: Technical Writer

*These are only the official roles. While they are defined, we are expecting to collaborate on different parts of the project together.* We determined these roles for the main responsibilities based on strengths determined in our introduction meeting.

# Life Cycle Model Used

Our team will implement a Waterfall Software Development Lifecycle (SDLC) to accommodate the DXC Technology, as well as the UTD requirements, for this project. At some point there might be a need to switch to a more agile approach which would then change into a hybrid model, but that depends on the process as it goes and measures up to our scheduling and guidelines put forth in this document.

The rationale for the Waterfall (Fig. 2) approach was taken into great consideration, and after two meetings with our team and DXC Technology, we concluded that the UTD requirements out-weighed any other plan in order to comply with meeting the SE 4485 standards given to us first. Also, learning the new monitoring technology was given great consideration and we all concluded that it would be best to attack this project one step at a time seeing as none of us are too familiar with the inner workings yet.

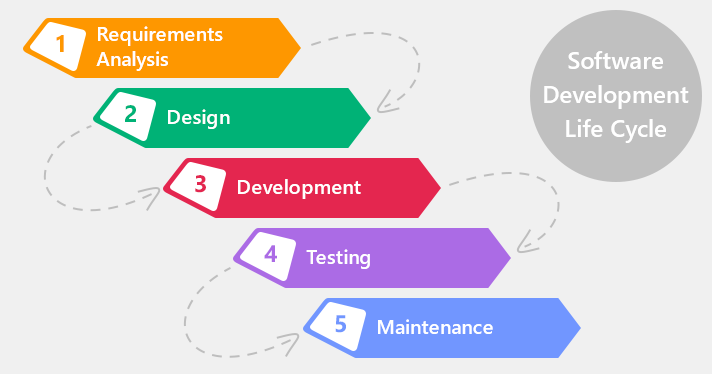


Figure 2 – WaterFall Methodology

# Risk Analysis

## RISK MANAGEMENT PLAN

|  |  |
| --- | --- |
| Requirement Change | Change Request Form |
| Delay in requirement completion. | Group meeting and split the work. |
| Monitoring mechanism failure | Create script to email everyone when the failure happens |
| Unit testing delay | Group meeting and every member test different component |

Table 1 - RMP

# Hardware and Software Resource Requirements

## HARDWARE

Our project is fully cloud based which means we would not be requiring any hardware other than a computer and an internet connection for the programmer and the user to use our services.

## SOFTWARE

* Jira testing
* VM - AWS
* GitHub
* JAVA - C
* Ac2 or Google
* Monitoring tool - Zabbix
* Nagios
* **User Story 1**: Choose VM - AWS or Azure or GCB
* **User Story 2**: Choose monitoring tool, i.e. Nagios
* **User Story 3**: Decide on agent/agent less monitoring
* **User Story 4**: Instrument monitoring given a machine (Invoke an API to start monitoring)

Epic

Postman for API calls

Swagger - for API calling that is connected to spring boot and we will also have a UI

Spring boot

Jenkins - deploying

JDK/JRE

IntelliJ

Maven - dependency management

# Deliverables and Schedule

We will have an estimated twelve weeks of time to work in. These will be divided up into deliverables in two week segments. While this is the initial plan, we may need to edit to stay consistent with our workflow.

* Deliverable 1 (due 9/11) - Finished plans (epic/user story creation)
* Deliverable 2 (due 9/25) - First iteration
* Deliverable 3 (due 10/9) - First iteration (continued)
* Deliverable 4 (due 10/23) - Finish first iteration & review with sponsor
* Deliverable 5 (due 11/06) - Second iteration, bug fixes
* Deliverable 6 (due 11/20) - Testing and quality assurance

# Monitoring, Reporting and Controlling Mechanisms

Monitoring and control keep projects on track. The right controls can play a major part in completing projects on time. The data gathered also lets project managers make informed decisions. They can take advantage of opportunities, make changes, and avoid crisis management issues. Some tools that can be used are:

* Work Breakdown Structure (WBS) to break a project down into small units of work, or sub-tasks.
* Requirements Traceability Matrix (RTM) This will map, or trace, the project’s requirements to the deliverables. This makes the project’s tasks more visible. It also prevents new tasks or requirements being added to the project without approval.
* Control Chart monitors the project’s quality. There are two basic forms of control chart – a univariate control chart displays one project characteristic, while a multivariate chart displays more than one.
* Review and Status Meetings further analysis problems, finding out why something happened. They can also highlight any issues that might happen later.

# Professional Standards

Our standards are developed and approved through a consensus-based process that ensures that all interested stakeholders can participate. PMI publishes 13 globally recognized project management standards which will be followed by our team for this project:

* A Guide to the Project Management Body of Knowledge (PMBOK® Guide)
* The Standard for Program Management
* The Standard for Portfolio Management
* Organizational Project Management Maturity Model (OPM3®)
* Practice Standard for Project Risk Management
* Practice Standard for Earned Value Management
* Practice Standard for Project Configuration Management
* Practice Standard for Work Breakdown Structures
* Practice Standard for Scheduling
* Practice Standard for Project Estimating
* Project Manager Competency Development Framework
* Construction Extension to the PMBOK® Guide Third Edition
* Government Extension to the PMBOK® Guide Third Edition

# Evidence the Document has been placed under Configuration Management

How da fuck do I prove we are using GitHub?!

# References

Ritter, Danielle. “SDO: Project Management Institute.” *PMI - Project Management Institute*, www.standardsportal.org/usa\_en/sdo/pmi.aspx.

“Project Monitoring and Control Techniques.” *PMP® Blog - Project Management Professional Training*, 21 Oct. 2019, [www.projectmanagementqualification.com/blog/2019/10/21/project-monitoring-control/](http://www.projectmanagementqualification.com/blog/2019/10/21/project-monitoring-control/).

“What are microservices really all about? - Microservices Basics Tutorial.” YouTube. Dec 31, 2018. <https://www.youtube.com/watch?v=j1gU2oGFayY>

“Getting Started | Applications on the ServiceNow Platform.” YouTube. Sept 27, 2018. <https://www.youtube.com/watch?v=ofKrvQ32AXo>