FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
DEPARTMENT OF SOFTWARE ENGINEERING AND INFORMATION TECHNOLOGY

QUALITY MANAGEMENT DOCUMENTATION

COMPANY NAME: JAGUAR INTEL

PROJECT NAME: TRAVEL LAH!

SSE 4300

SOFTWARE PROJECT MANAGEMENT

GROUP 1

SEM 1 (2018/2019)

GROUP MEMBERS:

FATIN SYAKIRA BINTI MUHAMMAD - 189041 (PROJECT MANAGER)

SITI HAJAR BINTI IBRAHIM - 190631 (TASK LEADER 4)

NUR SARAH BINTI BAHARUN - 186803

NUR ALYA AMANI BINTI HARIS - 188263

CONTENTS

1.0 Purpose of Software Quality Management

Software quality management (SQM) is a management process that aims to develop and manage the quality of software in such a way so as the best ensure the product meets the quality standards expected by the customer while also meeting any necessary regulatory and developer requirements, if any. Software quality managers require software to be tested before it is released to the market, and they do this using a cyclical process-based quality assessment in order to reveal and fix bugs before release. Their job is not only to ensure their software is in good shape for the consumer but also to encourage a culture of quality throughout the enterprise.

Software quality management activities are generally split up into three core components: quality assurance, quality planning, and quality control.

2.0 Quality Planning

2.1 Project Quality Goals

Goals	Category	Goal description	How to measure
Defect removal	Efficiency	Increase defect removal efficiency (DRE) from 90% to 99.5%	Comparing all bugs found during development to those reported in the first 90 days by customers [2].
Cost of quality (COQ)	Process and software product improvement	Lower cost of quality (COQ) from 45% of development to 20% of development.	Cost of Quality (COQ) = Cost of Poor Quality (COPQ) + Cost of Good Quality (COGQ) [1]
Cyclomatic complexity	Coding structure	Reduce average cyclomatic complexity from 25% to 10%	Measuring cyclomatic complexity for all modules.
Schedule	Delivery on time	Reduce the odds of schedule delays from 50% to 5%.	Defect prevention combined with pre-test static analysis [2].
Improve support	Customer centric	Improved support delivered and also reduces need for support.	Annual population survey (APS) statistics – eg: number of bugs. [3]
Improve engineering practices	Efficiency and effectiveness	Following the coding standard or code review.	% Amount of code reviewed [3].
Team versatility	Collaboration	Enable flexibility to change workloads.	Identify weak points and set targets to increase knowledge that are lacked.
Open communication	Collaboration beyond the team	Achieve effective collaboration with other teams and roles.	Feedback from project manager or team members.

2.3 Strategy for meeting Quality Goals

Goals	Strategy	Expected Benefits
Defect removal	Perform a defect removal method that helps to increase the efficiency of review and testing activities. Example of defect removal methods that can be applied are design inspections, code inspections and unit tests [4].	10 -15 % reduction in defect injection rate and about 5% improvement in project performance
Cost of quality (COQ)	Provide quality and compliance information and metrics visible across the organization such as evaluate compliance risk from audit results (internally and externally) and complaints reported [1].	Improving the project's cost of good quality and reducing the cost of poor quality.
Cyclomatic complexity	Measuring cyclomatic complexity for all modules to ensure the system is testable and maintainable [5].	Helps to understand the complexity of the program better which useful for testing and maintainability stage.
Schedule	Manager may monitor progress in every development stage of the product and be creative in providing solutions to a problem in handling any expected delays.	Avoid to delay the schedule especially during testing if high number of defects are discovered.
Improve support	Provide good documentation or manual for user guide and installation of the product.	Increase performance of the product and reduce number of reports/complaints from client.
Improve engineering practices	The manager must identify standard that need to be followed for each development activities and provide checklist for inspections.	Approximately 10 % reduction in defect injection rate and 5 %

		improvement in overall productivity
Team versatility	The manager encourage open communication among team members during weekly meeting and post mortem.	Helps to exchange knowledge, skills and solve any misunderstanding.
Open communication		

3.0 Quality Assurance

3.1 Quality Tracking

Quality Activity	**Action**
Defect tracking	Use Dimensional Control Systems (DCS) tool for logging defects and tracking them to closure
Reviews (requirements, high level design, detailed design)	Check against project goals in quality plan
Code review	Check against limits for each program through Statistical Process Control (SPC) tool
Independent unit testing	Check against limits for each program through Statistical Process Control (SPC) tool
Integration testing/System testing	Check against project goals in quality plan

4.0 Quality Control

4.1 Quality Reviews

Review Point	**Review Item**	**Type of Review**

End of Project Planning	• Project Plan • Dimensional Control System(DCS) setup • Project milestones	• Group review • Software Quality Assurance(SQA) review
End of 90% of Requirements (This should be at the end of First Elaboration Iteration)	• Business analysis • Software Requirement Specification(SRS)	• Group review
End of 90% Design (This should be at the end of Second Elaboration Iteration)	• Software Design Document	• Group review
After implementation of the first phase of the application	• Source code	• Group review

5.0 Quality Management Tools

5.1 Check Sheet Excel Template Tools

The check sheet is a form (document) used to collect data in real time at the location where the data is generated. The data it captures can be quantitative or qualitative.

Project Name: _____

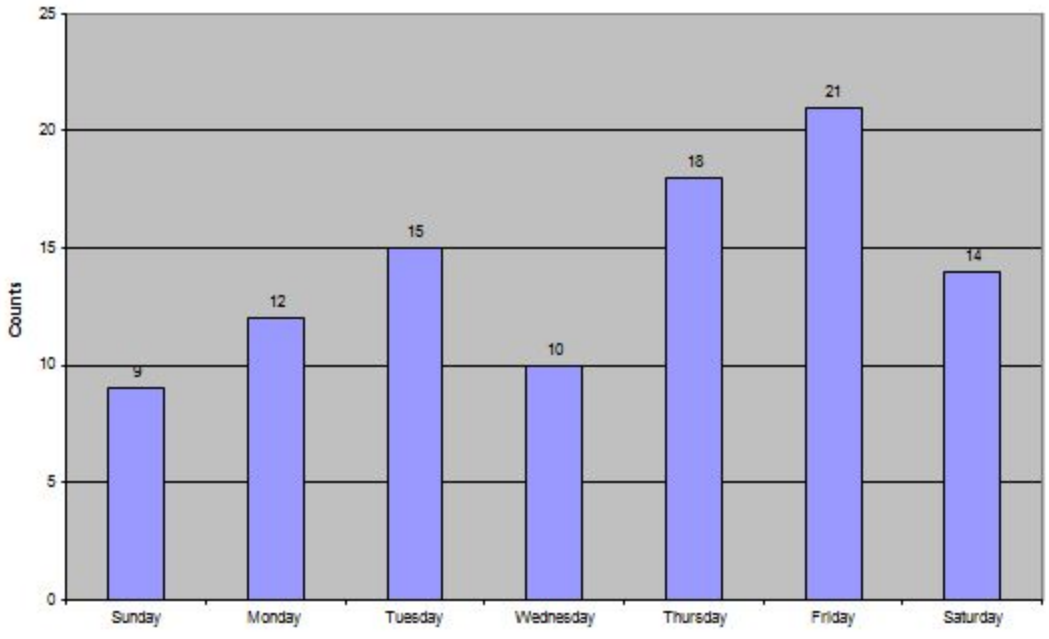
Name of Data Recorder: _____

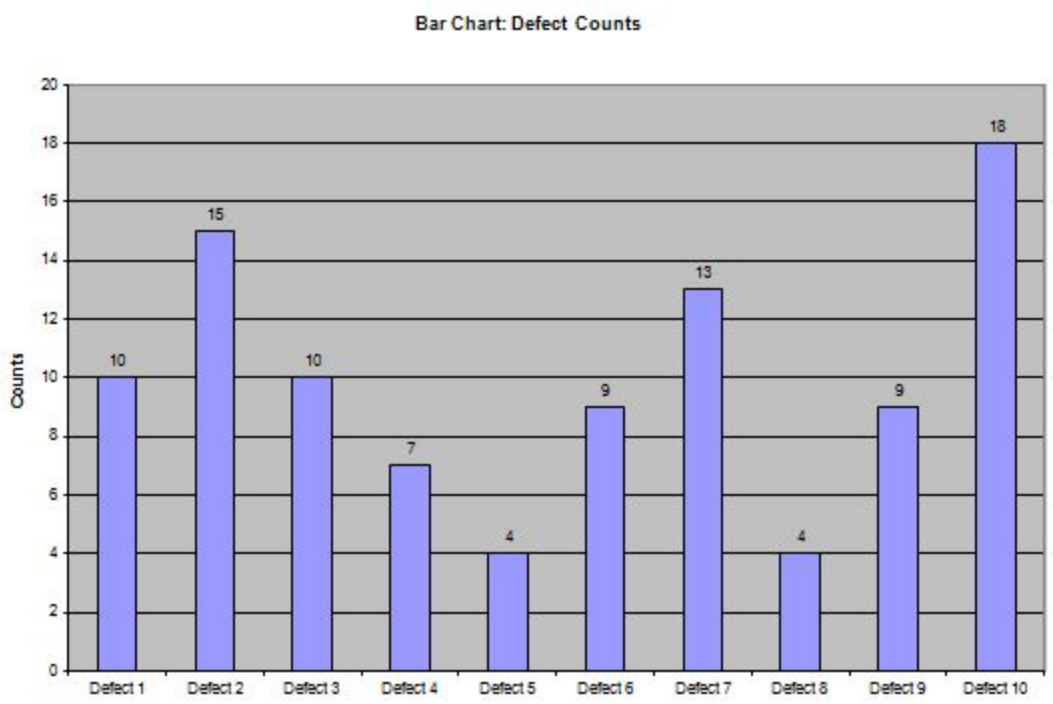
Location: _____

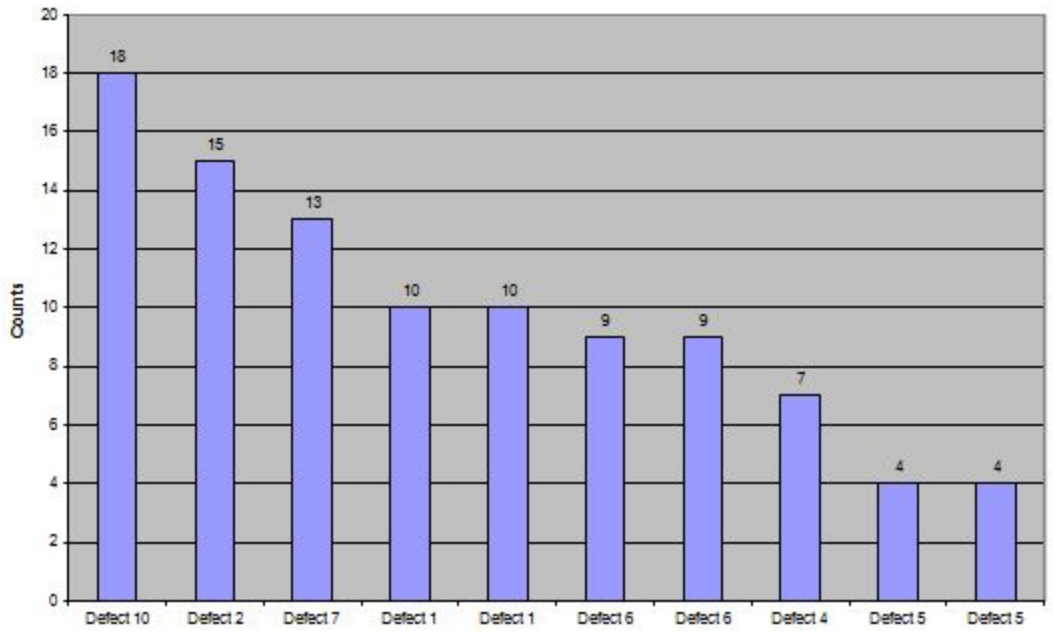
Data Collection Dates: _____

Defect Types/ Event Occurrence	Dates							TOTAL
	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	
Defect 1								
Defect 2								
Defect 3								
Defect 4								
Defect 5								
Defect 6								
Defect 7								
Defect 8								
Defect 9								
Defect 10								
TOTAL								

Histogram: Defects Over Time







5.2 Usage of tool

Kaoru Ishikawa, guru of Software Quality identified five uses for check sheets in quality control:

- To check the shape of the probability distribution of a process

- To quantify defects by type
- To quantify defects by location
- To quantify defects by cause (machine, worker)
- To keep track of the completion of steps in a multistep procedure (in other words, as a checklist)

5.3 Steps to use tool

The "Check Sheet-Weekly" worksheet can be printed for use by individuals in their data collection. Once data is collected on printed forms, type either the combined data or data for each individual into this Excel worksheet.

Then, the following charts will automatically be generated:

- Histogram that will shows the number of defects over time
- Bar Chart that will shows the number or count of defects
- Pareto Chart that will displays the 80/20 rule for defects

6.0 References

[1] Pilgrim an IQVIA Company. (2016). Evaluating the Cost of Quality: It's Simple Math. Retrieved December 7, 2018, from http://blog.pilgrimquality.com/evaluating-cost-quality-simple-math/

[2] Pizzutillo, P. (2014). 20 Software Engineering and Code Quality Goals You Should Nail Before 2018. Retrieved December 7, 2018 from https://www.castsoftware.com/blog/20-software-engineering-and-code-quality-goals-you-should-nail-before-2018

[3] Yeret, Y. (2012). Individual and team goals retrieved December 7, 2018 from, https://www.slideshare.net/yyeret/individual-and-team-goals

[4] Davies, K. (Ed.). (june 2008). Measuring Defect Potentials and Defect Removal Efficiency. *CROSSTALK The Journal of Defense Software Engineering, 21*, 6th ser., 11-12. Retrieved December 9, 2018, from https://pdfs.semanticscholar.org/d58a/773bf10a0fc87a9dc3b9cd5c4b615058ba7a.pdf.

[5] Karanth, D. (2016, May 19). Cyclomatic Complexity as a Quality Measure - DZone DevOps. Retrieved December 9, 2018, from https://dzone.com/articles/cyclomatic-complexity-as-a-quality-measure

[6] The 7 Basic Quality Tools for Process Improvement. (n.d.). Retrieved December 2, 2018, from https://asq.org/quality-resources/seven-basic-quality-tools