SOFTWARE MEASUREMENT PLAN

Version 1.3

December 1, 2016

Name	ID	Email
Yang Zhou	27784937	zy881228@hotmail.com
Jinsong Liu	27799772	js7jumpman23@gmail.com
Kaichen Zhang	40000160	ac1102919258@gmail.com

Preface

According to the first version of measurement plan, we are familiar with the work flow and process of measurement plan. By using the methodology of previous version, and combining the actual requirement and current situation of our Online Shopping Website project, we implement the measurement plan and complete the overall workflow.

Our current situation is that key improvements to the current products which were promised to customers have not been delivered as promised, and the delivery estimation is also unknown. We have already buffered lots of backlogs of deferred enhancement.

To help the Online Shopping Website project collect, analyze data and submit a conclusion report, our measurement plan will do following procedures according to the GQM method:

1) conclude subgoal statement
2) point out operational subgoals
3) come up with indicators and success criteria by questioning subgoal
4) design strategies to describe key subgoals and activities which is used to implement the strategy
5) analyze and progress indicator
6) identifying data elements to be collected, availability, and source
7) prepare the measurement plan

Revision History

Name	Rev.	Description	Author
September 16, 2016	1.0	First Draft	Yang Zhou Jinsong Liu
November 04, 2016	1.1	The 2nd Version	Yang Zhou Jinsong Liu Kaichen Zhang
November 22, 2016	1.2	The 3nd Version	Yang Zhou Jinsong Liu Kaichen Zhang
December 1, 2016	1.3	The 4nd Version	Yang Zhou Jinsong Liu Kaichen Zhang

Table of Contents

Table of contents

Section 1. INTRODUCTION

1.1 PURPOSE

The purpose of our software measurement plan is to specify the core measurements to be required in the future. The goal is to develop a set of metrics that will meet our current issues. Our priority is to deliver promised key enhancements according to the backlog, and problems of current

1.2 ORGANIZATIONAL DESCRIPTION

There is a large scope of the organization would be affected by the software measurement branch.

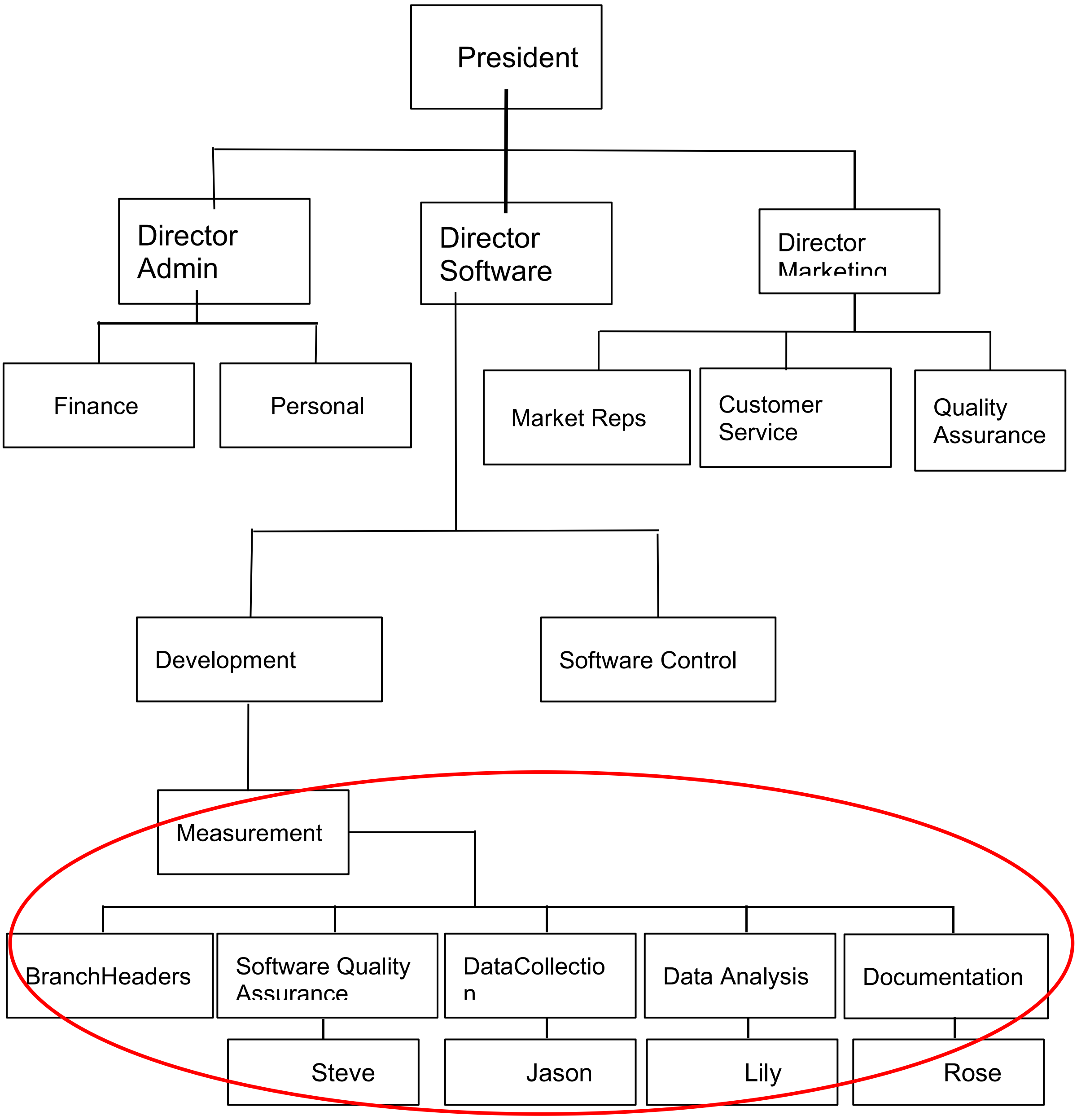


Figure 1.1

The measurement team is affecting the whole organization in a Bottom-Up way. The middle level of the organization, Director Admin, Director Software, Director Marketing, will replan their schedule or budgets, and assign the new plan to their branches. For example, when the measurement team produces a results indicating the project is ahead of plan and over budget, then the Director Admin will order the Finance branch to control the budget of future project development, and reduce the number of stuffs to meet the current situation. The Director Marketing will also forward their plan to the sub departments.

Software Project Manager identifies and manages project issues and uses measurement results to make program decisions.

Data Analysis Team tailors measures to address program issues and collects and analyzes measurement data and reports results.

Development Team uses measurement results in software engineering efforts and provides measurement data.

1.3 REFERENCES

1) CMMI Software process assessment
2) Quality Assurance group
3) Data Defect Tracker (DDT)
4) Common Repository and Storage Hierarchy (CRASH)
5) Online Shopping Measurement Plan Version 1.0
6) GQM method document
7) SMP-instructions
8) Online Shopping Requirements
9) ISO/IEC. ISO/IEC 15939:2002. Software Engineering – Software Measurement Process. 2002.
10) Software Project and Process Measurement. Dr. Christf Eber.Jul2009

Section 2. MEASUREMENT ORGANIZATIONAL ROLES AND RESPOSIBILITIES

2.1 Software project manager's measurement responsibility

In our project organization definition, the Software Project Manager should be responsible for the completion of project within limited resources. The SMP makes decisions to ensure the successful project. So he will be responsible for directing and organizing the required formal reviews and will be responsible for the overall conduct of this measurement.

2.2 Data analysis team assignment and responsibility

The Data Analysis Team will provide supports to the project manager and software measurement project directly. The data analysis team is responsible for data collection, database operations, data processing presentation, and data analysis. The data analysis team should be composed of :

1. Data Analysis Team Leader - The Software Quality Assurance Manager will serve as the Data Analysis Team Leader. His responsibilities include directing the Data Analysis Team in performing their duties in data collection, analysis, and presentation.
2. Software Quality Assurance Team - The software quality assurance team is responsible for defining the quality standard according to the ISO, current project requirement and industry-wide standards.
3. Data Collection Team - The data collection team should be responsible for define indicators and success criteria. Define the data elements, and assign hand to collect associated data elements. Processing the data element organization, checking working list.
4. Data Analysis Team - The Data Analysis Team should include the Software Development team, the Test team, the SCM Manager and the SQA group. The team leaders and the SCM Manager are responsible for data collection, and review of presentation materials. The SQA group, under the direction of the SQA Manager, will perform data analysis, development of presentation materials, and the maintenance of the Project History Files and MS Project plans.
5. Documentation Team - The documentation team records the work of measurement group, checks the requirements and makes sure the error free tracking of the measurement work.

Section 3. SOFTWARE PROJECT MEASUREMENT SPECIFICATIONS

3.1 Step1: Brainstorming and Additional Sub-Goal Statements

As a professional software company, Commonwealth Software, Inc. (CSI), we need to deliver the product within limited resources and fulfill the customer requirements maintain and improve the quality of the product according to the business goals. Here we concentrate more on the efficiency and delivery punctuation of our product, more specifically, we set our business goal as to deliver high-efficient products on time and increase customer satisfaction.
Now we are acting as software manager for development specifically. From the software management perspective we choose how to improve customer satisfaction as our subgoal. The goal of our project is to improve the customer satisfaction quality by 40 percent for this season compared to the last season, based on the business goal stated above.

3.1.1 Questions related to the business goal

Grouping questions related to software quality

Grouping #1 (documents)	• Are the documents we produce readable? • Are the documents structure features traceable? • Are the documents concise and complete? • Is the terminology correct? • Are the documents audits successful?
Grouping #2 (software product)	• What tools do they use? • Do our people provide effective solutions to fix the problem? • Are plans and changes communicate with the customer? • Is development progress visible to the customer? • Are status and progress of change requests visible to the customer? • Is the source code consistent with the requirement documents?
Grouping #3 (project management)	• Is the number of the developers enough to produce the products the customer wants? • Are our developers meet the qualifications of developing process? • Do personal affairs affect the schedule of the product? • Where would the backlog happen? • Are requirement analysis and estimations done before starting the developing process?

Grouping #4 (risk management)	• Is the number of the testers enough to produce the products the customer wants? • Are our testers meet the qualifications of testing process? • Have potential risks been considered ahead of developing process? • Do our people give efficient measures to control the risk?
Grouping #5 (quality management)	• Is the number of the quality inspectors enough to produce the products the customer wants? • Have our quality inspectors been systematic trained to qualified to guarantee the quality of the product? • Are the requirement analysis done comprehensive? • What developing model do our people choose? • What developing structure do our people use?
Grouping #6 (Communications)	• Do our developers analyze the requirements from the customer? • Do our developers respond quickly to the requests changed from the customer? • Is the testing result visible to the customer? • Is our product delivered to the customer on time? • Do our people give efficient and timely action to the feedback?

3.1.2 Subgoals

Subgoal 1 : Improve quality of the source code

Subgoal 2 : Improve communication with the customer

Subgoal 3 : Improve the reliability of the software release

Subgoal 4 : Manage the priority of solving backlog

Subgoal 5 : Improve the quality of design and development documentation

Subgoal 6 : Improve the performance of organization

3.2 Step2: Operationalized Goals

Subgoal 1: Improve the quality of the source code
Perspective: Project Manager
Object of Interest

source code development process

Purpose

Improve source code readability and traceability in order to improve quality of the software.

Quality Focus & Perspective

Examine the quality and defects from the perspective of project development manager.

Environment and Constraints

Organizational standards and structure, customer factors, budgets, application environment, resource factors

Subgoal 2: Improve the reliability of the software release

perspective: project manager

Object of Interest

software release

Purpose

Check the software released and reduce the existing problem which affect the reliability in order to improve the quality of the software.

Quality Focus & Perspective

Examine the defect of the source code from the perspective of project development manager

Environment and Constraints

budgets, application environment, resource factors

Subgoal 3: Improve the quality of design and development documentation

perspective: project manager

Object of Interest

Traceability and readability of documentation to be delivered on time

Purpose

To trace the stage of current development, and make plans to improve the quality of our system by analyzing the documentation.

Quality Focus & Perspective

Examine the quality of the documents from the perspective of project development manager

Environment and Constraints

budgets, application environment, resource factors

3.3 Step3: Success Criteria and Indicators

Measurement Goal: Improve the quality of the source code

From the reference of "The Correlation among Software Complexity Metrics with Case Study" written by Tashtoush Yahya, Al-Maolegi Mohammed and Arkok Bassam in 2014, "Code readability: Code comments OR self-documenting code : How does the choice affect the readability of the code?" written by Nielsen Sebastian and Tollemark David in 2016, after considering the current situation of the measurement goal, the quality of the source code can be composed of code correctness, code complexity and code readability. We come up with the following questions and related indicators.

Q1: What is the summation of the defects per thousand lines of codes of the software in the developing process?

Q2: What is the ratio of the minimum cyclomatic complexity in the source code of the software?

Q3: How to evaluate the readability of the source code?

Indicator	Evaluation	Success criteria
DKLOC	defects/KLOC	≤ 0.92‰
ratio of CCMin	CC = E-N+2 CCMin/CCSum	≥80%
Code Readability	The Number of GOOD	The Number of GOOD ≥ 2

Figure 3.1: Success criteria

3.3.1 DKLOC Indicator

Indicator: The DKLOC Indicator measures the ratio of summation of all levels of defects and every thousands lines of codes throughout the developing process of the software which will reflect the quality and reliability of the source code. (3 reference).

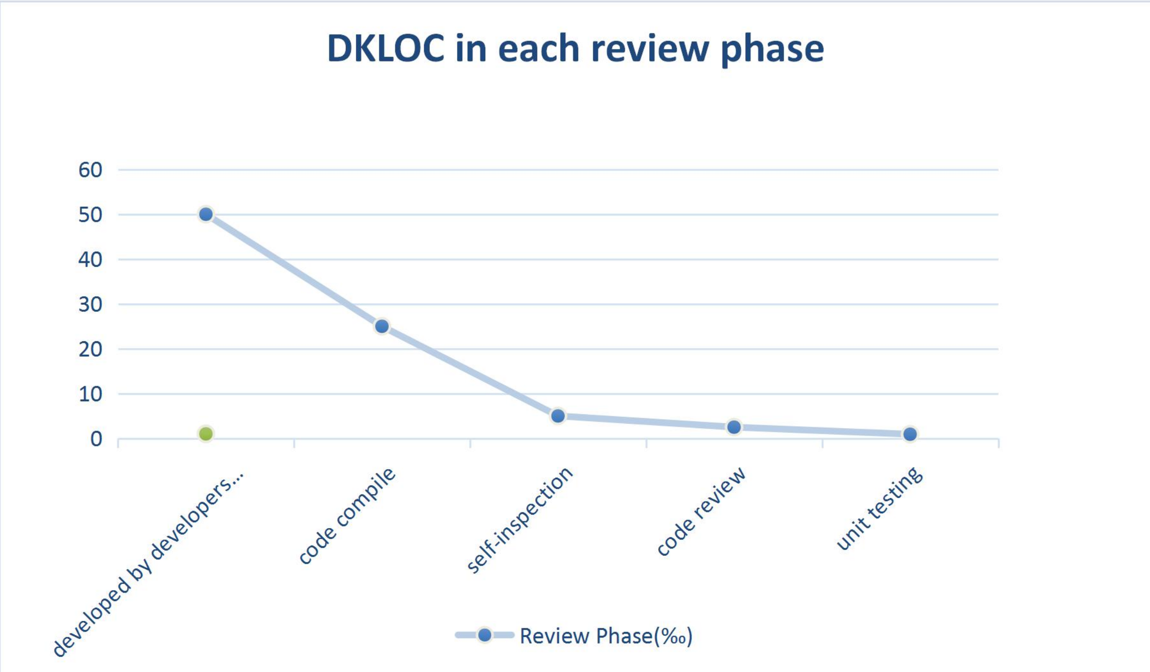
Computing formula:
DKLOC=(Defects_minor+Defects_medium+Defects_significant+Defects_critical+Defects_catastrophic)/KLOC

Success criteria: Achieve the CMM4 level which means the ratio of defects per thousand lines of codes is less than or equal to 0.92‰.

Review Phase	DKLOC
source code	50‰

code compile	25‰
self-inspection	5‰
code review	2‰
unit testing	≤ 0.92‰

Figure 3.2: The DKLOC in each review phase



3.3.2 Cyclomatic Complexity(CC) Indicator

Indicator: Cyclomatic Complexity is computed using the control flow graph of the program: the nodes of the graph correspond to indivisible groups of commands of a program, and a directed edge connects two nodes if the second command might be executed immediately after the first command. This indicator would help us to measure the complexity of the source code and improve the quality of the code.
CCMin: 1-10
CCMid: 11-20
CCMax: greater than 21

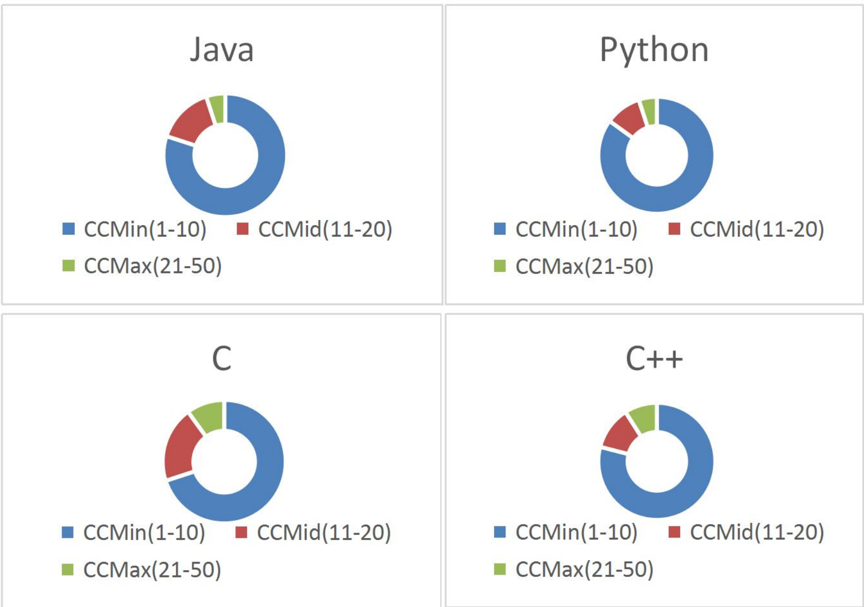
Cyclomatic Complexity	Risk Evaluation
1-10	a simple program, without much risk

11-20	more complex, moderate risk
21-50	complex, high risk program
greater than 50	untestable program(very high risk)

Figure 3.3: The risk evaluation of the CC

Language	CCMin(1-10)	CCMid(11-20)	CCMax(21-50)
Java	80%	15%	5%
Python	85%	10%	5%
C	70%	20%	10%
C++	79%	12%	9%

Figure 3.4: The distribution of the CC in each phase



McCabe showed that T-(G)is also equal to the number of binary decision nodes in G plus one. There are four basic rules that can be used to calculate TT(G):

1) Increment one for every IF, CASE or other alternate execution construct
2) Increment one for every Iterative DO, DO-WHILE or other repetitive construct
3) Add two less than the number of logical alternatives in a CASE
4) Add one for each logical operator (AND, OR) in an IF.
The three variants studied in this measurement plan are defined as follows:
a) CYCMAX: all four rules are used, as in the original McCabe version

b) CYCMID: only rules 1-3 apply, as proposed by Myers
c) CYCMIN: only rules 1 and 2 apply, as suggested by Hansen

Computing formula:
CCMin/CCSum

Success Criteria: The success criteria for the cyclomatic complexity ratio indicator would be the actual cyclomatic complexity in the source code structure which CCMin >= 80%.

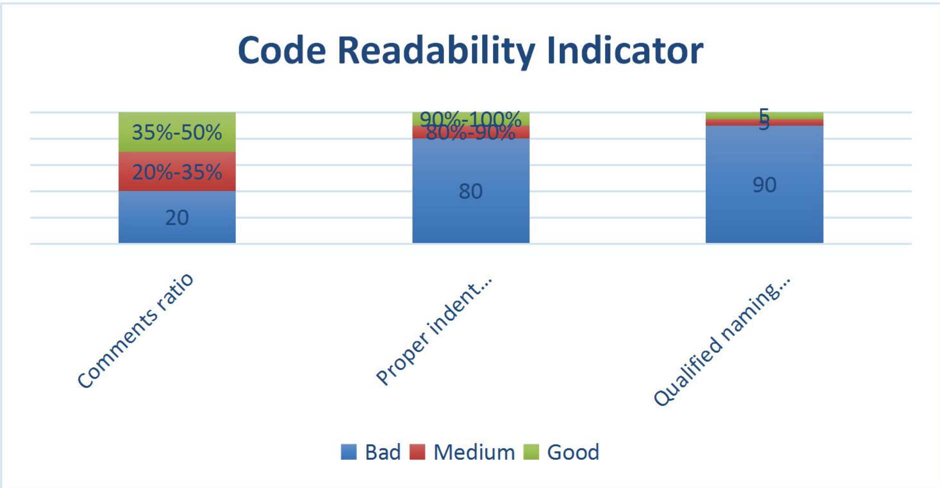
3.3.3 Code Readability Indicator

Indicator: The code matching the coding standards contains: correct and adequate comments, proper naming conventions for objects, right indent styles. This indicator will integrate these factors and decide whether the code is readable. It would help us to improve the readability of code and then improve the quality.

Success Criteria: The Readability Indicator measures the ratio of code that matches the coding standards of company. Success criteria is that at least 2 factors are good and no factor is bad.

Factor	Good	Medium	Bad
Comments ratio	35%-50%	20%-35%	< 20%
Proper indent styles	90%-100%	80%-90%	< 80%
Qualified naming conventions	95%-100%	90%-95%	< 90%

Figure 3.5: The quality of the code in each factor



Computing formula:
Comments ratio: Lines_with_comments/SLOC
Proper indent styles: Lines_with_proper Indentstyles/Lines_needIndentstyles
Qualified naming conventions: Numbers_with qualified_naming_conventions/Numbers_need qualified_naming _conventions

3.4 Step4: Subgoals to Strategies and Activities

Subgoal: Improve the quality of the source code

Strategy 1: Enhance the effectiveness of the code review and revision.
Impact of Strategy:
1. It will decrease the defects in the code.
2. It can increase the readability and traceability of code.

Activities:
1. Improve the self-inspection effectiveness.
2. Assess the coverage and the consistence of peer review.
3. Team leader should be in charged of each phase of development.
4. Check the inner consistency of all related documents.
5. Use refactoring to lower coupling and improve cohesion of code.
6. Consult the efficient algorithms that have been used in the similar situation to lower the complexity of algorithms.

Strategy 2: Improve the readability of source code.
Impact of Strategy:
1. It will help programmers comprehend other people's work.
2. It construct solid basis for later maintaining work.

Activities:
1. Use design patterns.
2. Avoid bad code smells and anti-patterns.
3. Use annotations and documentations.
4. Follow universal programming syntax.

3.5 Step5: Analysis and Progress Indicators

At this step we are going to develop some analysis and progress indicators to address the questions quantitatively and communicate the results to others.

Strategy 1: Enhance the effectiveness of the code review and revision

Impact of Strategy: 1. It will decrease the defects in the code
2. It can increase the readability and traceability of code.

Activities:
1. Improve the self-inspection effectiveness.
2. Assess the coverage and the consistence of peer review.
3. Team leader should be in charged of each phase of development.
4. Check the inner consistency of all related documents.
5. Use refactoring to lower coupling and improve cohesion of code.
6. Consult the efficient algorithms that have been used in the similar situation to lower the complexity of algorithms.

Questions about the Activity

Activity	Quantifiable question related to the activity
Improve the self-inspection effectiveness	• How many defects are fixed through self-inspection?
Assess the coverage and the consistence of peer review	• How many defects are detected in the peer review process? • What is the minimum cyclomatic complexity in the peer review process?
Team leader should be in charged of each phase of development	• How many phases are under control by team leader? • How many phases' quality are improved after team leaders' review?
Check the inner consistency of all related documents	• What's the rate of inner consistency varied in different kinds of documents?
Use refactoring to lower coupling and improve cohesion of code	• How many kinds of bad smells are detected and what's the impact on software performance.
Consult the algorithms	• What algorithms can be consulted?

used in the similar situation to lower the complexity	• What time complexity of algorithm is acceptable?

Figure 3.6: The distribution of the CC in each phase

Indicators

Defects/KLOC Indicator
Cyclomatic Complexity Indicator

Analysis

Peer review is a well defined process that examined by its authors and one or more colleagues. It is carried out by peers representing areas of life cycle affected by material being reviewed. The purpose of this activity is to evaluate its technical quality and provide a disciplined practice for locating and correcting defects in software artifacts. With the help of peer review, software quality can be ensured and problems can be identified and fixed in early lifecycle.

Peer Evaluation Form for Group Work

Write the name of each of your group members in a separate column. For each person, indicate the extent to which you agree with the statement on the left, using a scale of 1-4 (1=strongly disagree; 2=disagree; 3=neither agree nor disagree 4=agree; 5=strongly agree). Total the numbers in each column.

Evaluation Criteria	Group member:	Group member:	Group member:	Group member:
Number of minor defects every KLOC				
Number of medium defects every KLOC				
Number of significant defects every KLOC				

Number of critical defects every KLOC				
Number of catastrophic defects every KLOC				
Minimum Cyclomatic Complexity				
Mean Cyclomatic Complexity				
Maximum Cyclomatic Complexity				
TOTALS				

Figure 3.7: Peer Evaluation Form for Group Work

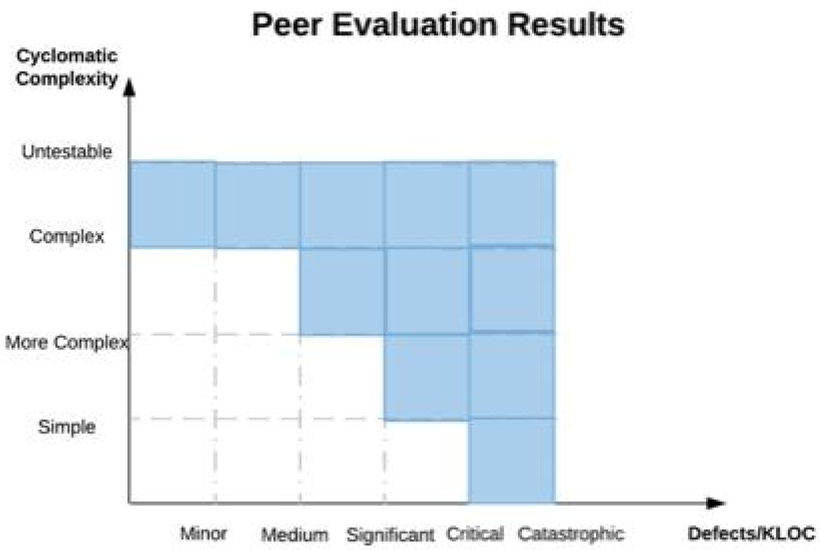


Figure 3.8: Peer evaluation results

The peer evaluation results showing a positive relationship between Cyclomatic Complexity and Defects from KLOC, which is the more larger the KLOC the more difficult to fix the Cyclomatic Complexity.

Progress

As the project is processing the states of DKLOC and min Cyclomatic Complexity should be changed in the way showing in the figure below. The DKLOC shold decrease and approach the bottom approximately, because there should be less error after we process the measurement and planning.But the Cyclomatic Comlexity should slightly increase because the whole system is growing larger than before.

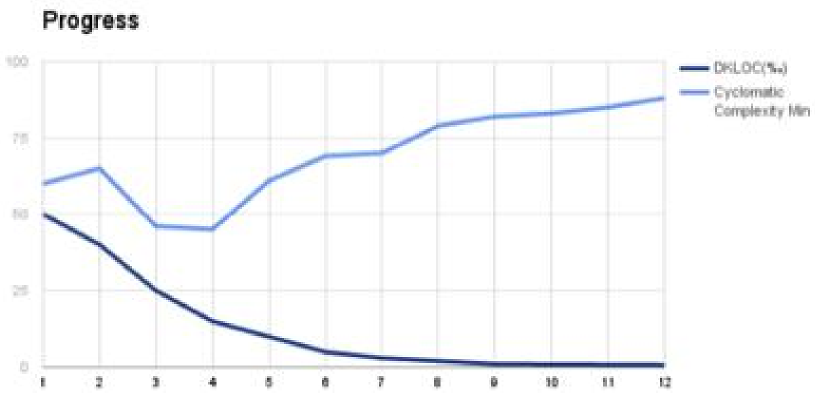


Figure 3.9: Peer review progress

Section 4. MEASUREMENT PROCESS

4.1 Step6: Identifying Data Elements to be Collected, Availability, and Source

Data Elements Required

Data Elements Required	Indicator		
	DKLOC Indicator	**Cyclomatic Complexity(CC) Indicator**	**Code Readability Indicator**
Defects_minor	×		
Defects_medium	×		
Defects_significant	×		
Defects_critical	×		
Defects_catastrophic	×		
SLOC	×		×
Statements		×	×
Lines_with_comments			×
Lines_with_proper Indentstyles			×
Lines_needIndentstyles			×
Numbers_with qualified_naming_conventions			×
Numbers_need qualified_naming			×

_conventions			
Nodes(control flow)		×	
Edges(control flow)		×	
The number of the set of the graph(control flow)		×	
CCMin		×	
CCMid		×	
CCMax		×	

Data Elements Availability

Data Elements Required	Avail	Source
Defects_minor	00	From users' review and customers' feedback
Defects_medium	00	From team leaders' review and self-inspection
Defects_significant	00	From code review and peer review
Defects_critical	00	From code review, peer review and unit testing
Defects_catastrophic	00	From code review, peer review and unit testing
LOC	+	The source code line display on Eclipse
Nodes(control flow)	00	From the control flow graph derived from the functional module of source code
Edges(control flow)	00	From the control flow graph derived from the functional module of source code
The number of the set of the graph(control flow)	00	From the control flow graph derived from the

		functional module of source code
CCMin	-	From the results of the excel documents computation
CCMid	-	From the results of the excel documents computation
CCMax	-	From the results of the excel documents computation
Lines of codes with comments	+	The number of the source code lines with comments in the statistic documents
Lines with proper Indent styles	+	The number of the source code lines with proper indent styles in the statistic documents
Lines need Indent styles	+	The number of the source code lines needed indent styles in the statistic documents
Numbers of entities with qualified naming conventions	00	The number of entities with qualified naming conventions in the statistic documents
Numbers of entities that need qualified naming conventions	00	The number of entities that need qualified naming conventions in the statistic documents

Code	Meaning
+	Available
0	Can be derived from other data
00	Can be obtained via minor effort
-	Not available now

--	Impossible to obtain or extremely difficult

4.2 Step7: Planning Tasks

This step will review the action item checklist and analyze these tasks to see if they are sufficient to collect, store, analyze, etc. the required measures (data elements) for our indicators.

Action Item Checklist

Planning Tasks	**Data Element**																
	Defects_minor	**Defects_minor**	**Defects_significant**	**Defects_critical**	**Defects_catastrophic**	**LOC**	**Nodes(control flow)**	**Edges (control flow)**	**The number of the set of the graph (control flow)**	**CCM in**	**CCM id**	**CCM ax**	**Lines of codes with comments**	**Lines with proper Indent styles**	**Lines need Indent styles**	**Numbers of entities with qualified naming conventions**	**Numbers of entities that need qualified naming conventions**
Data elements defined	Y	Y	60 %	Y	Y	N	Y	90 %	Y	Y	Y	Y	N	Y	N	80 %	Y
Data collection frequency and points in the software process defined	50 %	N	60 %	Y	N	30 %	Y	N	Y	N	Y	Y	Y	Y	70 %	Y	Y
Timelines defined for measurement result to	Y	Y	30 %	Y	Y	Y	70 %	Y	Y	N	N	N	Y	Y	60 &	Y	80 %

data bases and users																	
Data collection forms defined	Y	N	N	N	N	N	Y	30 %	90 %	N	Y	Y	90 %	Y	Y	Y	N
Data collection procedure defined	Y	N	79 %	Y	N	Y	15 %	Y	Y	N	N	N	Y	Y	Y	Y	Y
Data storage, database design, and data retention responsibilities defined	Y	Y	88 %	N	Y	Y	Y	Y	80 %	N	Y	50 %	Y	Y	80 %	Y	50 %
Who will collect and who will access defined data	N	N	56 %	Y	N	N	80 %	Y	Y	N	30 %	N	60 %	Y	Y	N	Y
Analysis process defined	Y	Y	N	Y	Y	40 %	Y	70 %	?	N	Y	N	Y	60 %	Y	N	Y
Reporting process defined	Y	Y	Y	Y	Y	N	Y	Y	N	78 %	Y	90 %	N	Y	90 %	Y	Y
Supporting tools identified and made available	N	N	Y	Y	Y	Y	N	N	70 %	Y	Y	N	Y	N	Y	N	Y

Process Guide for data definition and collection prepared	Y	Y	Y	Y	Y	Y	90 %	N	80 %	N	Y	N	N	Y	N	Y	N

The following form modify the existing tasks and add new task as required and list the rational for each.

Planning Tasks	Rationale
Process planning: identify the needs for measurements	For any measurement need, it is important to recognize that there may be several audiences or users of the measurement data, each with different perspectives and needs for measurement.
Identify the objectives of the organization.	This task ensures that measurement is aligned with and will support decision making with respect to the business objectives of the organization. Also, these are the methods that need to be measured in order to give the measurement users insight to make decisions.
Identify the methods that will be used to achieve the objectives.	This task will help to ensure that there are methods to achieve the objectives, and that the objectives are feasible. Also, these are the methods that need to be measured in order to give the measurement users insight to make decisions.
Identify the issues that need to be managed, controlled or observed.	These issues should refer to products, processes, and/or resources that are traceable, and relate to the methods. Some common issues are size, cost, and schedule.
Translate each measurement issues into precise, quantifiable and unambiguous measure goals..	These goals may involve measurement to help individuals understand an issue (i.e., to evaluate, predict, or monitor) or to address a need for improvement (i.e., to increase, reduce, achieve, or stabilize).
Data elements defined	Data elements describe the logical unit of data, fields are the actual storage units, and data items are the individual instances of the data elements.

Data collection frequency and points in the software process defined	Data collection frequency refers to the time frequency at which data is collected at regular interval. It is useful in analyzing things like if the transactions were legal or illegal and many other useful information.
Timelines defined for measurement result to data bases and users	Timelines is to help understanding the order or chronology of events or tasks for the project. By showing time on a specific scale on an axis, a timeline is used to visualize time lapses between tasks, durations and the simultaneity or overlap of spans and tasks.
Data collection forms defined	The data collection form serves several important functions. First, it links directly to the review question and criteria for accessing eligibility. Second, it is the historical record of the multitude of decisions that occur throughout the review process.
Data collection procedure defined	Accurate data collection is essential to maintaining the integrity of the project. Both the selection of appropriate data collection instruments and clearly delineated instructions.
Data storage, database design, and data retention responsibilities defined	It is the process of producing a detailed data model of database. The data model contained all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language.
Who will collect and who will access defined data	It refers to the persons who collect and process data as data controllers.
Analysis process defined	It is a step-by-step breakdown of the phases of a process, used to convey the inputs, outputs and operations that take place during each phase. It can be used to improving understanding of how the process operates and to determine potential targets for process improvements.
Reporting process defined	It is the set of steps and requirements to standardize the process.
Supporting tools identified and made available	Supporting tools can help to streamline the process. For the development, supporting tools are used to create, debug, maintain or otherwise support other programs or applications.
Process Guide for	It is the process of collecting data into one file or data table for use in

data definition and collection prepared	analysis. The process of preparing data generally entails correcting any errors, filling in nulls and incomplete date, and merging data from several sources or data formats.
Prepare and presente report	Report provides insight into issues, status of goals obtained from report. Report is understood and acceptable (list criteria, e.g., approved by audience)
Review and revise procedure	Analysis procedures are adequate (list criteria, e.g., additional issues captured)
Evaluate measurement process	Audience reviews and accepts measurement report. Inconsistencies or concerns to be addressed are identified

4.3 Effort and Budget Estimation

The technique that we used in the effort estimating is **Delphi**. The detailed steps is described as follows:

1. Coordinator presents each expert with a specification and an estimation form.
2. Coordinator calls a group meeting in which the experts discuss estimation issues with the coordinator and each other.
3. Experts fill out forms anonymously.
4. Coordinator prepares and distributes a summary of the estimates
5. Coordinator calls a group meeting, specifically focusing on having the experts discuss points where their estimates vary widely
6. Experts fill out forms, again anonymously, and steps 4 to 6 are iterated for as many rounds as appropriate.

The results form is as follows:

No.	Planning Tasks	Effort (Hours)	Start time	End time	Predecessor	Budget (1k Dollars)
0	Process planning: identify the needs of measurements	16	09/24/16	09/25/16		0.5

1	Identify the objectives of the organization.	24	09/26/16	09/28/16	0	0.8
2	Identify the methods that will be used to achieve the objectives.	16	09/29/16	09/30/16	1	0.4
3	Identify the issues that need to be managed, controlled or observed.	16	10/01/16	10/02/16	0	0.4
4	Translate each measurement issues into precise, quantifiable and unambiguous measure goals..	16	10/03/16	10/04/16	0	0.5
5	Data elements defined	65	10/24/16	11/03/16	0	1.5
6	Data collection frequency and points in the software process defined	64	11/04/16	11/13/16	5	1.5
7	Timelines defined for measurement result to data bases and users	67	11/14/16	11/23/16	6	2.0
8	Data collection forms defined	64	11/24/16	12/03/16	7	1.5

9	Data collection procedure defined	65	12/04/16	12/13/16	8	1.5
10	Data storage, database design, and data retention responsibilities defined	64	12/14/16	12/23/16	9	1.5
11	Who will collect and who will access defined data	65	12/24/16	01/03/17	10	1.5
12	Data collection	80	01/04/17	02/15/17	11	2.3
13	Analysis process	45	10/24/16	12/24/16	0	1.1
14	Reporting process	40	12/25/16	02/15/17	13	1.0
15	Supporting tools identified and made available	10	12/24/16	12/25/16	10	0.2
16	Process Guide for data definition and collection prepared	20	10/24/16	10/31/16	0	0.4
17	Prepare and present report	10	01/05/17	01/15/17	11	0.2
18	Review and revise procedure	60	01/16/17	02/15/17	17	1.1
19	Evaluate measurement process	30	10/24/16	02/15/17	0	0.8

	Total					17.7

Work Breakdown Structure

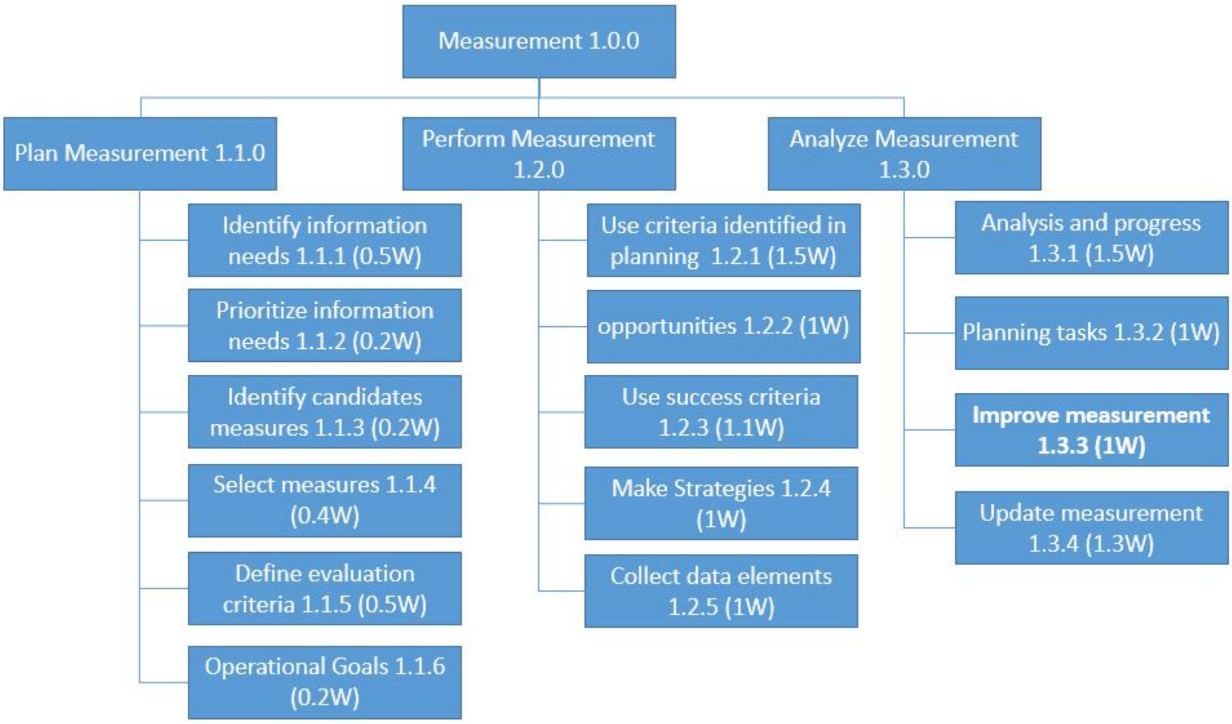
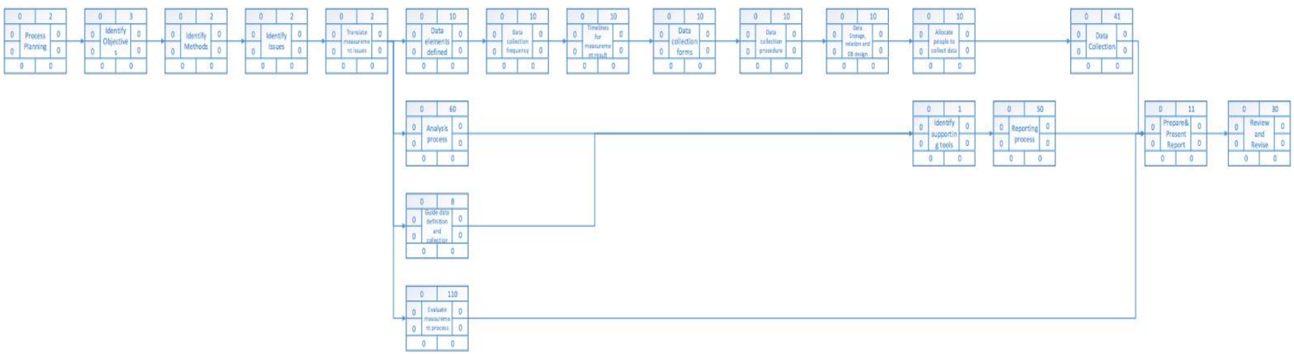


Figure. Measurement Work breakdown structure

CPM



Appendix: Definition

Defects Degree	Description	Number of Defects / KLOC
Defects_minor	comments or advices on improving the software	>10
Defects_medium	cause users to make mistakes or confused while use the product but have no effects on software's job function, primary function or secondary function	5~10
Defects_significant	influence the normal operation or the usage of the software	1~5
Defects_critical	functions incomplete or return incorrect results; slow performance; software unstable or causing data damage; inevitable primary problems which are unable to fix in normal operations; must be fixed ASAP	0.1~1
Defects_catastrophic	software crashes, hangs; software can not work due to data loss, data damaged, a serious shortage of resources, unable to use the primary function	<0.1

Reference

[1] Agrawal M, Chari K. 2007. Software effort, quality, and cycle time: A study of CMM level 5 projects. IEEE Transactions on software engineering. 33(3):145-156.
[2] M. Diaz and J. Sligo. 1997. "How Software Process Improvement Helped Motorola," IEEE Software. 14(5):75-81.
[3] C. Fox and W. Frakes. 1997. "The Quality Approach: Is It Delivering?" Comm. ACM. vol. 40:25-29.