2015 International Conference on Advances in Computer Engineering and Applications (ICACEA)
IMS Engineering College, Ghaziabad, India

Exhaustive study of SDLC Phases and their Best Praxctices to create CDP Model for Process Improvement

Rachna Lekh
Computer Science Engineering & Technology
CT Institute of Management & Technology
Jalandhar, India
rachna_lekh@ymail.com

Ms. Pooja
Computer Science Engineering & Technology
CT Institute of Management & Technology
Jalandhar, India
poojachoudhary80@gmail.com

***Abstract*— This paper reveals the study of sub methods of the phases of SDLC i.e. Software Development Life Cycle. Some methods or sub-practices are always required to execute each and every phase. There is always a need of some procedures, steps, methods and processes to build anything whether it is software or anything. Even a recipe needs a proper steps otherwise it will give a bad taste. An appropriate approach or accurate steps are to be followed always to construct a quality product. Various models, standards and processes have been introduced by ISO, CMMi etc to follow a proper approach to reach the destination. There was a need to find a method which would be suitable for company/organization. As companies have to manage each and everything with the existing resources by keeping in mind the quality and goal of customer. This paper represents the exhaustive study of the sub-methods or sub-practices of SDLC phases as per the process areas of CMMi and a model which shows the best suited combinations of various methods to achieve the desired product as per the requirements and existing resources. This is was the part of process improvement of a software organization "Saber Corp(Made-Up)". This whole paper represents the CDP model which was the output of process improvement process of company Saber Corp.**

Keywords*— *SDLC sub-practices, process improvement, SDLC Phases.

I. INTRODUCTION

A software development process is a structure imposed on the development of a software product. Synonyms include software lifecycle and software process. [19]There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process. SDLC is a systematic approach to problem solving and is composed of several phases, each comprising multiple steps. Basic SDLC constitutes – Requirement Phase, Analysis Phase, Design Phase, Development Phase, Testing & Maintenance phase.

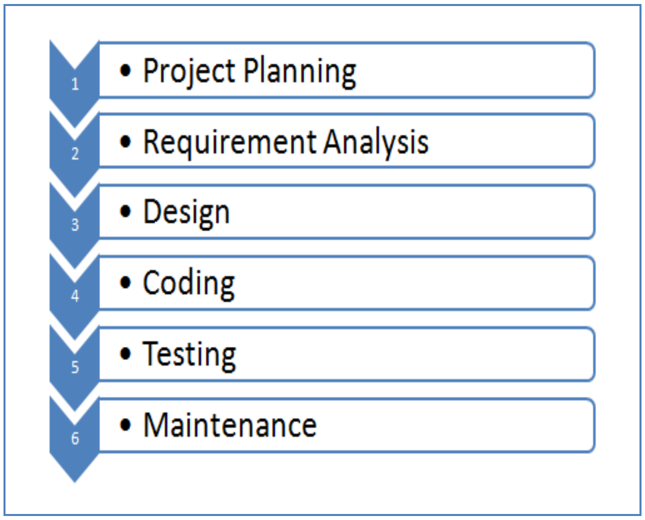


Fig. 1. Phases of SDLC

These basic phases are broken by various organizations as per their needs and approach. Following are the descriptive phases:

A. Planning

Planning is the first phase of software development. In this phase, client gives the details and concepts of his/her software and organization plan the requirement of resources, time & budget of the proposed development.

B. Requirement/Imapact Analysis

Planning This phase is concerned with capturing the requirements of the package. The requirements review is a meeting with the aim of discussing these requirements. The final output of this phase is a formal requirements document (Software Requirement Specification) and reusable templates (steps of requirement analysis part are used to create templates which can be reused across the projects), which aims to freeze the requirements at this point and will serve as input to the design phase. The impact analysis phase, tracked in release plan is concerned with tracking the impact of changes before

978-1-4673-6911-4/15/$31.00©2015 IEEE

actual changes(configurable items) are tracked in Project Management plan.

C. Design Phase

The design phase is concerned with design of the software. There are two types of designs: HLD & LLD. HLD requires Requirement Specification & Effort estimation. Database and Sequence diagrams are designed in this design. This level is reviewed by Onsite manager. In LLD, the output documents of HLD serve as input. In this design level, technical description of the design, test cases and test scripts are prepared. This level reviewed by Team Leader and QA Leader. The final output of this phase is a formal design document(Software design document), which aims to freeze the design at this point and will serve as input to the coding phase.

D. Development

The development phase involves the actual coding/programming of the software as per LLD. The output of this phase is typically the library, executables and user manuals and additional software documentation.

E. Testing

The testing phase is concerned with the validation & verification of the software. At this stage, two types of testing are done : Unit Testing & Integration testing.

Unit Testing is done on single units. Integration testing is done by including each package/unit together with other packages/units and testing them altogether.

F. Evaluation Release

The pilot of the product and client evaluates the product. If he/she requires modification in the product, he suggests it and it will be done in a short span of time.

G. Beta/Interim Release

The release phase involves packaging of all sub-packages, together with all relevant documentation in a suitable format for distribution.

H. Recycle

In case of long term projects, the release phase is the starting point of recycling of the project but in short term projects release phase is the point of sign off too. In this phase Black box testing, regression testing & system testing is done. Black Box testing is the testing based on an analysis of the specification of a piece of software without reference to its internal workings. The goal is to test how well the component conforms to the published requirements for the component. Regression testing is retesting of previously tested program following modification to ensure that faults have not been introduced or uncovered as a result of the changes made; System Testing is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. It falls within the scope of Black box testing.

release phase involves packaging of all sub-packages, together with all relevant documentation in a suitable format for distribution.

I. User Acceptance Testing

In some engineering sub-disciplines, it is known as functional testing, black box testing, release acceptance, QA testing, application testing, final testing, validation testing, usability testing or factory acceptance testing.

J. Deployment

The process whereby software is installed into an operational environment . Every project should establish deployment scenarios.

II. CMMI

CMMi stands for **Capability Maturity Model Integration**. This model constitutes a set of best practices which provides essentials elements of effective processes of organization. It helps in setting up the goals and priorities for process improvement, grants guidance for quality processes and also a mode for the appraisal of ongoing or existing processes. Five maturity levels are there which act as an essential layer of the overall current improvement activities of the organization. [1][18][22]

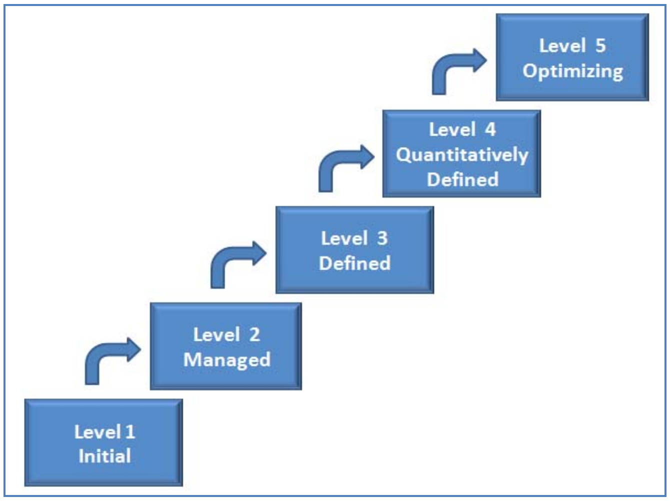


Fig. 2. Maturity Levels of CMMi

A. Maturity Level 1 : Initial

No stable atmosphere. Chaotic, adhoc and unstable processes at this stage. Success repetition is not possible because of unsteadiness of processes.

B. Maturity Level 2 : Managed

Policies and processes are portrayed as per projects and they are habitually thoughtless.

C. *Maturity Level 3 : Defined*

More characterized and standardized processes

D. *Maturity Level 4 : Quantitatively Defined*

Quantitative objectives are established to cater the needs of customers/end users, organization, process owners and all the stake-holders.

E. *Maturity Level 5 : Optimizing*

Organization is more focused on the way to Incremental process improvement which is endlessly reviewed, revised and improved.

III. Key Process Areas of CMMi

Process areas are defined as the arenas which let us know about the phases of the product development which are to be covered by the processes of organization.

TABLE I. Process Areas

Abbreviation	Name	Area
CM	Configuration Management	Support
DAR	Decision Analysis and Resolution (DAR)	Support
IPM	Integrated Project Management	Project Management
MA	Measurement and Analysis	Support
OPD	Organizational Process Definition	Process Management
OPF	Organizational Process Focus	Process Management
OT	Organizational Training	Process Management
PPQA	Process and Product Quality Assurance	Support
PI	Product Integration	
PMC	Project Monitoring and Control	Project Management
PP	Project Planning	Project Management
RD	Requirements Development	Project Management
REQM	Requirements Management	Project Management
RSKM	Risk Management	Project Management
SAM	Supplier Agreement Management	
TS	Technical Solution	Project Management
VAL	Validation	Review Process
VER	Verification	QC Process
CAR	Causal Analysis and Resolution	Support
OPM	Organizational Performance Management	Process Management
OPP	Organizational Process Performance	Process Management
QPM	Quantitative Project Management	Project Management

CMMi consists of best practices which address development activities are to be applied on products and services. It addresses practices that cover the product's lifecycle from conception through delivery and maintenance.

IV. CDP

Now after exhaustive study of the phases of SDLC and key process areas, a point comes to an individual's mind that what would be the sub-practices and sub-methods which actually execute or drive a particular phase. After a research, numerous methods have been introduced and adopted by the organizations around the world. For example : If we talk about requirement phase, SRS(System Requirement Specification) document is there, UML(Unified Modeling Language), wire-framing and many more methods for requirement phase. In same manner for testing phase, there is ad hoc testing, test case driven testing etc.

However after a lot of choices, we are still not sure about that which combination of different sub-methods gives us a better product as per our goal. So to overcome this problem, there is a need of a model which specifies some sub-methods. Consequently we will create a model which will specify the sub-methods of SDLC phases and will apply combinations/permutations to find best suitable methods. These best combinations of methods can only be defined by working on the historical data of the projects of an organization. To achieve this, a procedure was also needed: what is to be done, what is to be achieved, how is to be achieved, what elements are required etc.
Actually what we are going to execute, is a part of process improvement so there was a need to think from an

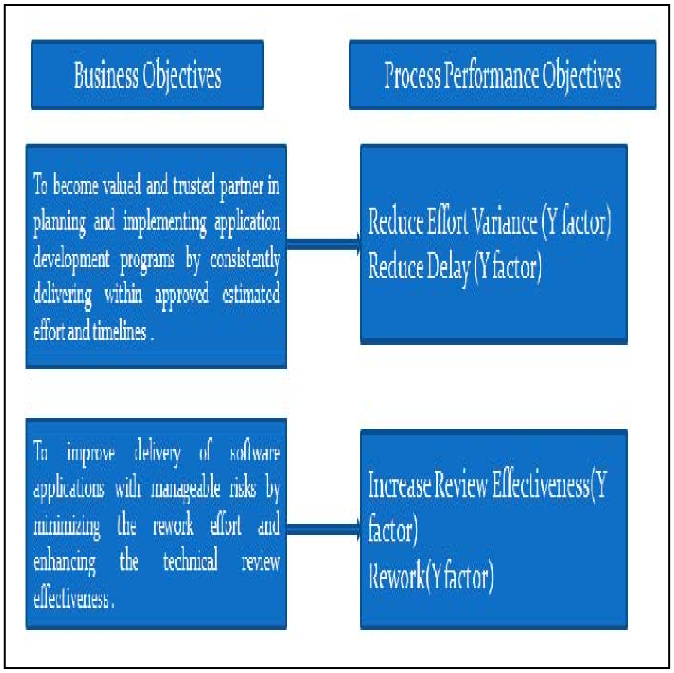
improvement perspective for an organization [10] [21]. So first of all, business objectives of the organization needed to be defined. As organization's quality policy was to satisfy all implicit & explicit requirements of the customers. Integrity & high quality is a hallmark of the organization.

As per this organization, the business objectives were:

- To become valued and trusted partner in planning & implementing application development programs by consistently delivering within approved estimated effort and timelines.
- To improve delivery of software applications with manageable risks by minimizing the rework effort and enhancing the technical review effectiveness.

As these business objectives are concerned with the process performance, so the further decomposition of these objectives were needed to find the **Process Performance objectives** of the organization for which the work is to be done. On further analysis, goal decomposition was done and found the factors which directly affect the goals/business objectives.

Fig. 3. Business Objective to Process Performance Objectives

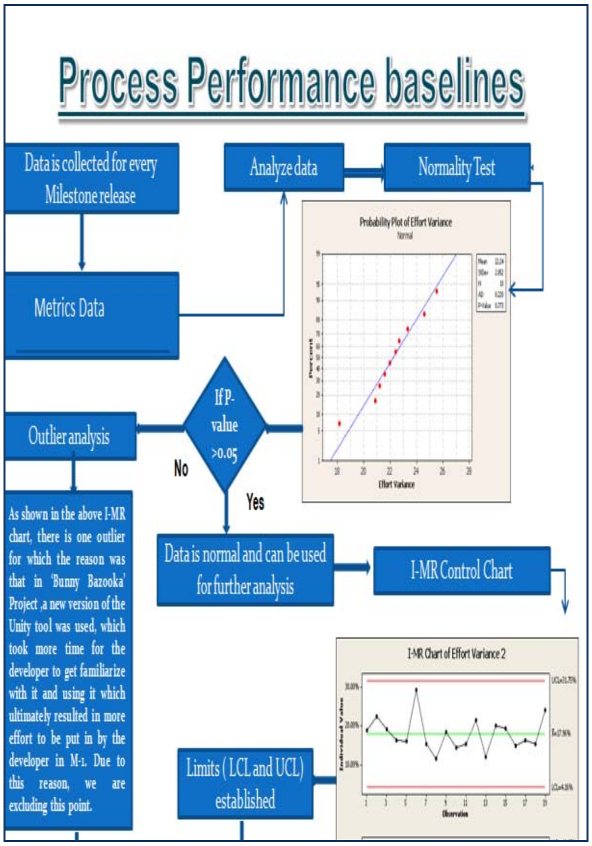

Business Objectives
Process Performance Objectives
To become valued and trusted partner in planning and implementing application development programs by consistently delivering within approved estimated effort and timelines .
Reduce Effort Variance (Y factor)
Reduce Delay (Y factor)
To improve delivery of software applications with manageable risks by minimizing the rework effort and enhancing the technical review effectiveness .
Increase Review Effectiveness(Y factor)
Rework(Y factor)

Y factors were needed to be derived from company's business objectives by GDM process i.e. Goal Decomposition process. Y factors were analyzed and corresponding metrics were identified so that Y factors can be monitored and controlled.

The sub-processes to be monitored were:

- Requirement Analysis
- Design
- Coding
- Quality Control(Testing)

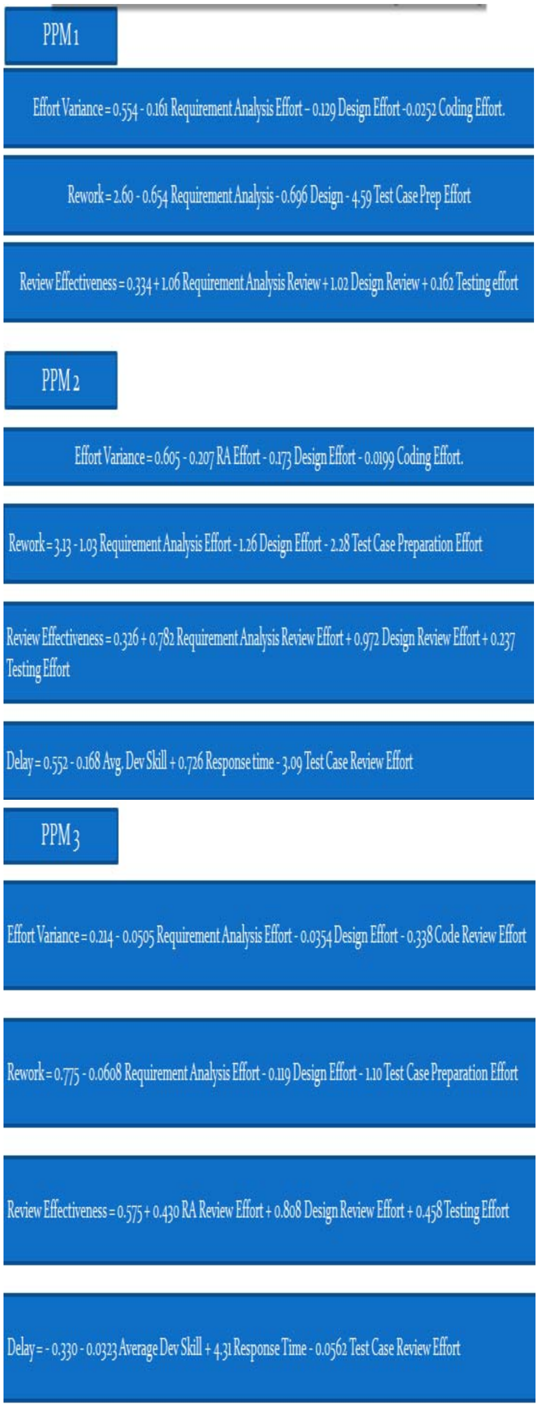
After identifying the metrics, projects data was collected and function points were calculated. Consequently, statistical data analysis tools were used to analyze the data for setting up the Process Performance baselines. Process performance baselines are the standard or the bench marks which could be set up by the organization on the basis of their historical data so that the organizations can judge their future product's performance. Minitab tool is used to create the PPBs.


Process Performance baselines
Data is collected for every Milestone release
Analyze data
Normality Test
Metrics Data
Probability Plot of Effort Variance
Normal
Percent
Effort Variance
If P-value >0.05
No
Yes
Outlier analysis
Data is normal and can be used for further analysis
I-MR Control Chart
I-MR Chart of Effort Variance 2
Individual Value
Observation
UCL=31.75%
X̄=17.96%
LCL=4.16%
Limits (LCL and UCL) established

As shown in the above I-MR chart, there is one outlier for which the reason was that in 'Bunny Bazooka' Project ,a new version of the Unity tool was used, which took more time for the developer to get familiarize with it and using it which ultimately resulted in more effort to be put in by the developer in M-1. Due to this reason, we are excluding this point.

Fig. 4. Process Performance Baseline values

Now the PPBs were established and the time was to establish the Process Performance model (PPM). PPMs were established by considering the projects data of various quarters for all the process performance objectives.



PPM1

Effort Variance = 0.554 - 0.161 Requirement Analysis Effort – 0.129 Design Effort -0.0252 Coding Effort.

Rework = 2.60 - 0.654 Requirement Analysis - 0.696 Design - 4.59 Test Case Prep Effort

Review Effectiveness = 0.334 + 1.06 Requirement Analysis Review + 1.02 Design Review + 0.162 Testing effort

PPM2

Effort Variance = 0.605 - 0.207 RA Effort - 0.173 Design Effort - 0.0199 Coding Effort.

Rework = 3.13 - 1.03 Requirement Analysis Effort - 1.26 Design Effort - 2.28 Test Case Preparation Effort

Review Effectiveness = 0.326 + 0.782 Requirement Analysis Review Effort + 0.972 Design Review Effort + 0.237 Testing Effort

Delay = 0.552 - 0.168 Avg. Dev Skill + 0.726 Response time - 3.09 Test Case Review Effort

PPM3

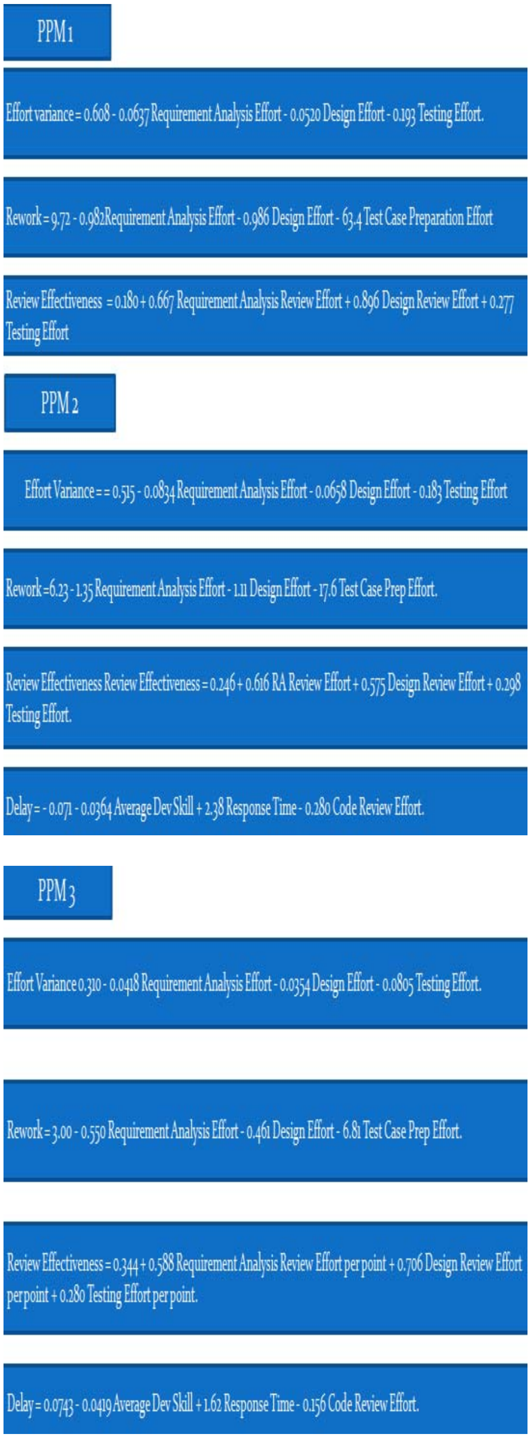
Effort Variance = 0.214 - 0.0505 Requirement Analysis Effort - 0.0354 Design Effort - 0.338 Code Review Effort

Rework = 0.775 - 0.0608 Requirement Analysis Effort - 0.119 Design Effort - 1.10 Test Case Preparation Effort

Review Effectiveness = 0.575 + 0.430 RA Review Effort + 0.808 Design Review Effort + 0.458 Testing Effort

Delay = - 0.330 - 0.0323 Average Dev Skill + 4.31 Response Time - 0.0562 Test Case Review Effort

Fig. 5. PPMs for Mobile projects of three quarters of a year



PPM1

Effort variance = 0.608 - 0.0637 Requirement Analysis Effort - 0.0520 Design Effort - 0.193 Testing Effort.

Rework = 9.72 - 0.982Requirement Analysis Effort - 0.986 Design Effort - 63.4 Test Case Preparation Effort

Review Effectiveness = 0.180 + 0.667 Requirement Analysis Review Effort + 0.896 Design Review Effort + 0.277 Testing Effort

PPM2

Effort Variance = = 0.515 - 0.0834 Requirement Analysis Effort - 0.0658 Design Effort - 0.183 Testing Effort

Rework =6.23 - 1.35 Requirement Analysis Effort - 1.11 Design Effort - 17.6 Test Case Prep Effort.

Review Effectiveness Review Effectiveness = 0.246 + 0.616 RA Review Effort + 0.575 Design Review Effort + 0.298 Testing Effort.

Delay = - 0.071 - 0.0364 Average Dev Skill + 2.38 Response Time - 0.280 Code Review Effort.

PPM3

Effort Variance 0.310 - 0.0418 Requirement Analysis Effort - 0.0354 Design Effort - 0.0805 Testing Effort.

Rework = 3.00 - 0.550 Requirement Analysis Effort - 0.461 Design Effort - 6.81 Test Case Prep Effort.

Review Effectiveness = 0.344 + 0.588 Requirement Analysis Review Effort per point + 0.706 Design Review Effort per point + 0.280 Testing Effort per point.

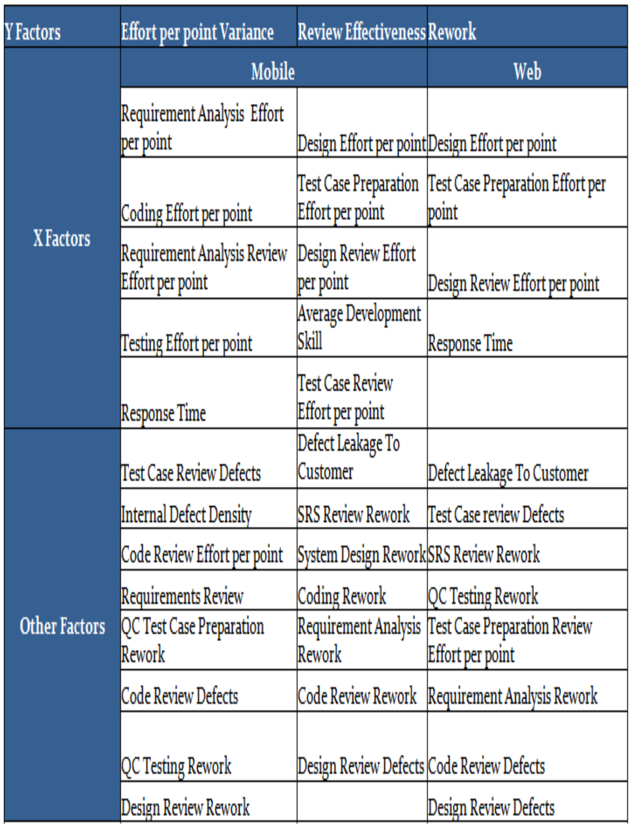
Delay = 0.0743 - 0.0419 Average Dev Skill + 1.62 Response Time - 0.156 Code Review Effort.

Fig. 6. PPMs for Web projects of three quarters of a year

As the PPMs were established, we came to know about the Y factors and identified the X factors which directly affects the Y factors.

Following figure is depicting the X & Y factors identified after establishing the PPMs.

TABLE II. X & Y FACTORS OF AN ORGANIZATION



Y Factors	Effort per point Variance	Review Effectiveness	Rework
	Mobile		Web
X Factors	Requirement Analysis Effort per point	Design Effort per point	Design Effort per point
	Coding Effort per point	Test Case Preparation Effort per point	Test Case Preparation Effort per point
	Requirement Analysis Review Effort per point	Design Review Effort per point	Design Review Effort per point
	Testing Effort per point	Average Development Skill	Response Time
	Response Time	Test Case Review Effort per point	
Other Factors	Test Case Review Defects	Defect Leakage To Customer	Defect Leakage To Customer
	Internal Defect Density	SRS Review Rework	Test Case review Defects
	Code Review Effort per point	System Design Rework	SRS Review Rework
	Requirements Review	Coding Rework	QC Testing Rework
	QC Test Case Preparation Rework	Requirement Analysis Rework	Test Case Preparation Review Effort per point
	Code Review Defects	Code Review Rework	Requirement Analysis Rework
	QC Testing Rework	Design Review Defects	Code Review Defects
	Design Review Rework		Design Review Defects

PPMs are treated as regression equations. Now the CDP model is created on the basis of PPMs and all other X-factors/Non X factors in order to perform the following:

- Performance prediction against goals
- Also to choose the best composition of the sub-processes among the defined sub-processes for all SDLC phases.

Following is the CDP model which consists of the sub-practices of the phases of SDLC as per the process performance objectives, process performance baselines & process performance models:

TABLE III. CDP MODEL WHICH CONSTITUTES ALL SUB-PROCESSES OF SDLC

SDLC Phases	**Method 1**	**Method 2**	**Method 3**
Requirement Gathering or Analysis (SRS Preparation)	Wire-framing	Document of Understanding (with screens)	Scope of Work (SOW)
SRS Review	Intra Team Review	Inter Team Review	
System Design	HLD	LLD	
Design Review	Walkthrough	Peer Review	
Coding(Including Unit Testing)	Reusable Code	Non-Reusable Code	Third Party component
Code Review	Sample based	Run Complete Checklist	Expert Review
QC Test Preparation	Detailed scenario based test cases	Simplified Checklist	Separate for Functional & Non-Functional/UI
QC Testing	Test Case driven testing	Ad-hoc testing	Functional & Performance Testing

V. STATISTICAL TOOLS

Statistical tools which are used for the processes to be performed are as follows:

1) Minitab

.

It is used to perform statistical tests/analysis. Following tests were performed :

a) Normality Test: Anderson Darling test is been performed to determine whether the data is normalized (normally distributed) or not. The test was executed to check the normality of data after collecting enough data points to run a Trial or Confirm Cycle. This data was used to derive control limits, consequently the PPBs and then PPMs.

b) Two Sample T Test: This test was performed to execute if the two sets of data are significantly different from each other.

This test was actually performed to check whether the Trial & Confirm limits are not significantly different as baseline can be established only if the data sets are same.

c) ANOVA: Analysis of Variance : (ANOVA) is a collection of statistical models used to analyze the differences between the group means their associated procedures. This test is actually for analyzing the defects to check which types of defects have higher mean values.

d) Control Charting: Control charts specifies the lower and upper control limits and often includes a central (average) line, to help in detecting trend of plotted values. This test was used to analyze the metrics for establishing the baselines and to analyze the processes to monitor variations of process performance over a specific period.

2) Crystal Ball: CDP Modeling and Analysis

It is used to create a model for prediction and to identify the composition of defined processes to be used in the project execution in order to get the desired results.

3) MS Excel

This was used to capture all the data such as efforts, function points etc. It was used to calculate all the function points and metrics by applying formulas on the cells of a worksheet.

Conclusion

These best practices have been extracted after a deep analysis of the phases of a software development life cycle as per the CMMi standards. CMMi standards lead the organization to the success of the product because of defined processes and key areas. With the practices, a model has been created and now these practices will be used for projects to find the values of find the values of the efforts in every process. These values or data will be used to find the best combination of sub-practices which will lead to a software quality and customer satisfaction.

References

[1] AedahAbd Rahman, ShamsulSahibuddin&Suhaini Ibrahim, "A Study of ProcessImprovement Best Practices", IEEE(2011).

[2] Shubo XU, Dishi Xu, "Project Management methodologies": Are they sufficient to develop quality software", IEEE(2011)

[3] Mejhem Yousef Al – Tarawneh, Mohd. Syazwan Abdullah &JasemAlostad, "Software Development Process Improvement Framework(SDPIFs) for small software development firms(SSDFs)" , International Journal of Computer Science Issues (IJCSI), Vol.10, Issue 1, (2013).

[4] T.R. Gopalakrishnan Nair, V. Suma, P. Kumar Tiwari, "Significance of depth inspection & inspection performance metrics for consistent defect management in software industry", The Institute of Engineering & Technology(IET), Vol. 6, Issue 6, pp 524-535,(2012)

[5] Changli Sun, "Software Document quality Measurement – A Fuzzy approach" , IEEE(2010).

[6] Li Wei, lin Xiao , Yu Wuyi, Yin Zhao, Cai, Jianhuai& Li Maoqing, "The Research Appliance of Multi Layer Fuzzy Comprehensive evaluation in the Appraisal of software quality", IEEE(2008).

[7] AhmadJavanBakht, Mohsen Fallah Rad &FarshadAkbari, "Using Fuzzy multiple criteria decision making in evaluation of software quality", IEEE(2010).

[8] ISO/IEC 9126-3: Software Engineering – Product Quality – Part 3 : Internal Metrics,2002

[9] Nithya G. Nair, Suma V, T.R. Gopalakrishnan, "Estimation of Characterstics of software team for implementing effective process through Inspection Performance Metrics".

[10] Bob McFeeley, "A User Guide for Software Process Improvement",Software Engineering Institute (SEI), Feb 1996.

[11] ISO/IEC 9126-1: 2001, Software Engineering-Product Quality-Part 1: Quality Model

[12] N.R. Shashi Kumar, T.R. Gopalakrishnan, V. Suma, "An Analytical Approach for project Manager in Effective Defect Management in Software Process", IEEE, 2011.

[13] Poornima U.S, Suma V, "Significance of Quality Metrics during Software Development Process"

[14] V. Suma, T.R. Gopalakrishnan, ""Defect Management using DI & IPM.

[15] Tom Kendrick, ""Defining & Implementing Metrics for Project Risk Reduction".

[16] Caper Jones, "Three Harmul Metrics & Two Helpful Metrics", 2012

[17] TopiHaapio, " Improving Effort Management in Software Development Projects", Publications of University of eastern Finaland, 2011.

[18] Sylvie Trudel, Jean Marc Lavoie, Marie-Claude Pare, WitoldSuryn, "PEM: The Small Company Dedicated Software Process Quality Evaluation Method combining CMMI & ISO/IEC 14598, Springer, 2006"

[19] Stephen H.Kan, "Metrics & Models in software quality engineering", Addison-Wesley Longman Publishing Co. Inc. Boston, MA, 2002.

[20] Jozef Pajzinka, "Quality optimization in IT organization using strategic process management methods", Masaryk University, 2013.

[21] Alan Ramias & Cheric Wilkins, "Building Metrics for a process", Process Improvement, Sept 2010.

[22] Sandro Felder, "Achieving CMMi compliance with Scrum focusing on ML2 Project management", Business Information systems, University of Applied Sciences & Arts, Northwestern Switzerland, 2013.

[23] Kanen A. LaFond, "The Evaluation of Project management performance on two software maintenance projects based on a CMMi Framework" , 2006.