Collaborative Project Management

– A Practical and Effective Mechanism For On time Delivery

6th Annual International

Project Management Leadership Conference

(PML 2011)

Submitted by

M. Ashfaq Ahamed

Ashfaq.m@nsn.com

Nokia Siemens Networks India Pvt Ltd,

L5, Manyata Embassy Business Park,

Bangalore

Abstract

Time to Market is critical for any product development company. On time delivery requires Project Planning in an integrated manner and has to be holistic in its approach. Different groups involved in product development and delivery have to plan through better understanding of process performance and often the gap is found in planning and monitoring by involving different stake holders. Project Manager plays a pivotal role in improving organizational efficiencies and sustains competitive advantage by facilitating communication & sharing among delivery teams.

This paper intends to propose a quantitative model based on the principle of collaborative planning which will enable Project Manager or departmental Manager to plan for better resource availability and utilization and helps ensure integrated project management for on time delivery.

1. Introduction

A project management team is a group of individuals/functions working interdependently to achieve a common goal. Teamwork is cooperative effort by members of a team to achieve that common goal. The effectiveness of the project team can make the difference between the project success and project failure. Communication is most essential part in a project team; it should be good with the customers and within the project team. Project Manager plays a pivotal role in improving organizational efficiencies and sustains competitive advantage by facilitating communication & sharing among delivery teams.

Most projects experience problems and unforeseen events due to poor communication which poses a threat to successful completion. Good Project Manager can provide team members and other stakeholders with a clear overview of who is doing what, and the status of a specific task. Visibility of project progress is a critical factor for project success.[5]

In a matrix organization where the Project Manager is assigned by the Project Management Office and is responsible for overall Project execution and success and functional organizations are responsible for assigning resources for design, development and validation. The Project Manager is responsible to forecast the resource requirement and negotiate the allocation and availability with functional groups for smooth execution of the project.

This paper intends to propose a quantitative model based on the principle of collaborative planning which will enable project manager to plan for better resource availability and utilization and helps ensure integrated project management for on-time delivery.

When product realization is organized by function, the R&D, Validation, documentation, tools, release engineering, are all separate groups. Functional Managers monitor the groups' performance separately and are responsible for resource management.

Typically, Validation groups own the verification and release processes and defines what gets tested and when, and what the shipment criteria are. The Development organization plans when to develop what features according to project priorities and needs and deliver the builds to validation team for validation.

Project Manager plays a vital role in bringing these different functions together and provides a platform for collaborative project management. Where the teams identify the information needs of the project stakeholders and agree on sharing them on a periodic basis for smooth project execution and achieving collectively as a team.

2. Collaborative Project Management

Collaborative Project management is based on the best practices of Integrated Project Management. A project is established and managed by involving relevant stakeholders and according to an integrated and defined process [1]. The working interfaces and interactions among relevant stakeholders internal and external to the project are planned and managed to ensure the quality and integrity of the overall endeavor. Reviews and exchanges are regularly conducted with relevant stakeholders to ensure that coordination issues receive appropriate attention and everyone involved with the project is appropriately aware of status, plans, and activities.

2.1. Benefits of Collaborative Project Management

- Improved planning through better understanding of process performance
- Improved coordination among people involved in a project
- Commitment to the organization is improved as communication & sharing between teams is facilitated
- Institutionalizing dependencies tracking & collaboration issues

2.2. Integrated Resource Management

Resources are crucial for development and delivery of product and good program management requires that there be a process to plan for the acquisition of future resources, to allocate current and project resources to schedules, and to make shifts in resource management required.[2] A quantitative model proposed in this paper helps forecast the resource requirement using the inputs provided by different stakeholders in the projects, in this case Development and Validation teams.

The proposed model is effective in the below described organization and Project environment. Process described and data used to elucidate the model do not exactly depict the development environment at Nokia Siemens Networks.

3. Organization and Project Environment

Software Product Development environment where it is functionally organized and has development teams working in developing new features and also engaged in operational activities like fixing the defects submitted by Validation group.

Incremental development model is used for developing the product where a set of features are developed, reviewed, Unit Tested and delivered as monthly builds to independent validation team. Validation teams execute identified test cases and file defects in change management tool. At the end of development all the modules are available and after integration, System level testing is done.

The defect submitted by validation team needs to be fixed before the final delivery.

3.1. Challenges

1. No visibility into Validation Plan, as the Validation team decides when and what features are tested (See Figure 1)
2. Inadequate inputs for Development Team to plan resources for defect fixing
3. Inadequate inputs for Validation Team to plan resources for Test Case Execution

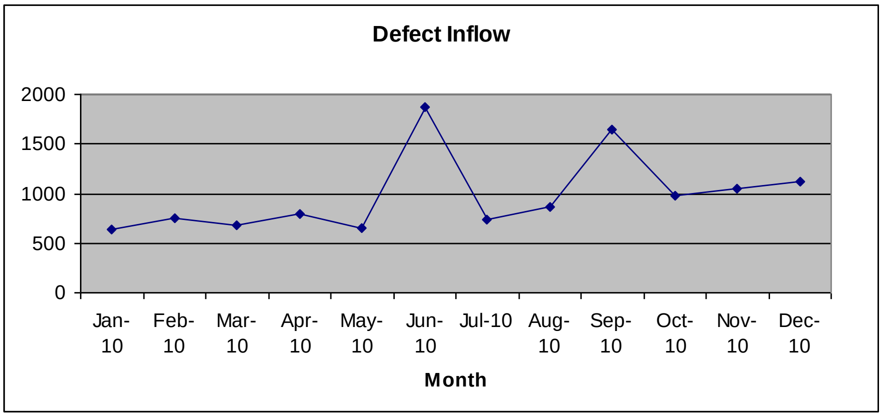

Figure 1: Variation in Defect Inflow due to varying set of features tested

In the above Figure, we can see variations in Defect Inflow as the monthly builds are tested on regular basis and in June and September additional features are validated

4. Resource Forecasting and Defect Inflow Prediction Model

The proposed model helps alleviate the above challenges when the development and validation teams share project planning and execution information on a periodic basis enabling each others to plan resources and meet the project deadlines.

In order to be effective the information sharing needs to be two-way process, the development team and validation team needs to share data on a regular basis. Inputs required from Development team and Validation Team are as follows

From Development Team to Validation Team

- Features Planned for delivery to Validation Team and estimated Code Size

Above input can help the validation team to plan the resource required for executing the test cases based on the features delivered or code size and Test Execution Productivity.

From Validation Team to Project Manager/Development Manager

- Number of Test Cases Planned for execution during that period
- Test Case Effectiveness based on Historical Baseline

By using the model proposed with above inputs the Project Manager can plan resources for defect fixing activity

4.1. Steps involved in building a model to predict defect inflow and resource requirement

- **Method –I**

1. Estimate the number of Test Cases planned for execution (Input from VAL)
2. Identify the Test Case Efficiency from historical baseline
3. Multiply the Planned Test Cases with Test Case Efficiency percentage to arrive at Predicted Defect Inflow.
4. Estimate the Defect Closure Productivity and forecast the resource for Defect fixing activity.

- **Method – II**

1. Estimate the Code Size based on the feature deliveries planned
2. Identify the Defect Density Delivered to Validation Team from historical baseline
3. Multiply Estimated Code Size with Historical Defect Density Delivered to Validation Team to arrive at predicted defect inflow
4. Estimate the Defect Closure Productivity and forecast the resource for Defect fixing activity.

Step 4 is common in both the methods

5. Illustration

In Method –I, to predicting defects two inputs are required from Validation team, i.e. Test Cases Planned and Test Case Effectiveness from Historic data. The validation team has visibility into these inputs, if these inputs are provided to Development teams; they can estimate the defect inflow.

Test Case Effectiveness can be computed as in *Equation 1*

$$\textbf{Test Case Effectiveness} = \left[\frac{\textbf{Valid Defects Filed by Validation Team}}{\textbf{Total Test Cases Executed}} \right] \textbf{x 100}$$

Equation 1: Test Case Effectiveness

Having arrived at Test Case Effectiveness, *Defect Inflow* (DI) prediction can be done by multiplying Test Cases Planned with Test Case Effectiveness percentage.

Defect Inflow (DI) arrived can also be verified with the latent defect delivered to testing team with the help of defects found in review and MT and Code Review Effectiveness. For instance, Defects found in Code Review is 100 and Code Review Efficiency is 65% then the latent defects delivered to validation team are 35.

Defect Inflow can let the development team plan the resources for defect fixing based on the Productivity figures of Defect Closure rate.

Defect Closure Productivity can be computed as in *Equation 2* below

$$\textbf{Defect Closure Productivity} = \frac{\textbf{Defects Closed in a period}}{\textbf{Resources deployed}}$$

Equation 2: Defect Closure Productivity

Defect Closure Productivity helps the development team to compute the resource requirement as in *Equation 3* below

$$\textbf{Resource Required for Defect Fixing} = \frac{\textbf{Estimated Defect Inflow}}{\textbf{Defect Closure Productivity}}$$

Equation 3: Resource Required for Defect Fixing

In Method –II, Defect Inflow can be predicted based on the feature release plan in monthly builds. As the typical project duration is around a year Monthly builds are used for incremental delivery of features. Development teams can estimate the code size either in KLOC or Function Points. In the Figure 2, the Code (Added/Modified/Deleted) for the month of January is 32 KLOC and based on the Process Performance Baseline (PPB) the Defect Density delivered to Validation team is 30 Defects per KLOC. Based on these measure development teams can compute Defect Inflow as the product of KLOC delivered and Delivered Defect Density to Validation team, and the value arrived at is 960.

Below is the snapshot of the model in Figure 2, where Defect Inflow is predicted based on both the methods and resource requirement for defect fixing is forecasted, this helps the Project Manager to plan and allocate the resource to deliver defect free product

	Jul-2010	Aug-2010	Sep-2010	Oct-2010	Nov-2010	Dec-2010	Jan-2011
Code (Added/Modified/Deleted) in KLOC	24	28	26	32	26	29	32
Estimated Delivered Defect Density per KLOC to VAL Team	30	30	30	30	30	30	30
Planned Resources for Defect Fixing	18	21	20	24	21	25	28
Defect Closure Productivity	40	40	40	40	40	40	40
INPUTS FROM VALIDATION TEAM							
Test Cases Planned	1920	2240	2080	2560	2080	2320	2560
Test Case Effectiveness	38%	38%	38%	38%	40%	45%	45%
PREDICTED VALUES							
Predicted Defect Inflow (Based on Test Case Efficiency)	730	851	790	973	832	1044	1152
Predicted Defect Inflow (Based on Delivered Defect Density to VAL team)	720	840	780	960	780	870	960
Resources Required for Defect Fixing	18	21	20	24	21	26	29
Resource Gap	0	0	0	0	0	-1	-1

Figure 2: Snapshot of the Model with hypothetical data

The model explained above has lead the project teams to successfully predict the defect inflow and resource requirement. As the model is deterministic the output is a single value but not range of results as in the case of probabilistic models. It is recommended that the Project Manager uses the above model to start with and having gained better understanding of process performance data advanced modeling techniques like Monte Carlo simulation can be used on the same model to predict range of defect inflow.

6. Advance Prediction Model Based on Monte Carlo Simulation

High maturity organizations use process performance models to understand the process behavior and predict the process performance. Monte Carlo simulation is widely used modeling technique.

Monte Carlo Simulation is a technique that converts uncertainties in input variables of a model into probability distributions. By combining the distributions and randomly selecting values from them, it recalculates the simulated model many times and brings out the probability of the output.

Why Use Monte Carlo Simulation? [3]

- Allows modeling of variables that are uncertain
- Provides a range of values instead of single value
- Enables more accurate sensitivity analysis
- Analyzes simultaneous effects of many different uncertain variables
- Provides a basis for confidence in a model output

The model illustrated above can be simulated to predict the Defect Inflow using Monte Carlo simulation method. The process of constructing a model in Crystal Ball [4] Software involves

Step 1: Determine the variables that are used within the model.
Step 2: Identify the type of distribution and for each variable
Step 3: Defining Assumptions
Step 4: Defining a Forecast
Step 5: Run Simulation
Step 6: Create a Forecast Chart

Snapshot of the Forecast Charts are provided below from Crystal Ball Software,
the variables used to predict the Defect Inflow based on TCE are

1. Number of Test Cases Planned
2. Test Case Effectiveness

The type of distribution used is Triangular Description as the triangular distribution is commonly used when you know the minimum, maximum, and most likely values. It is a continuous probability distribution.[4]

The parameters for the triangular distribution are minimum, likeliest, and maximum.

There are three conditions underlying the triangular distribution:

1. The minimum value is fixed.
2. The maximum value is fixed.
3. The most likely value falls at a point between the minimum and maximum values, forming a triangular shaped distribution, which shows that values near the minimum and maximum are less likely to occur than those near the most likely value.

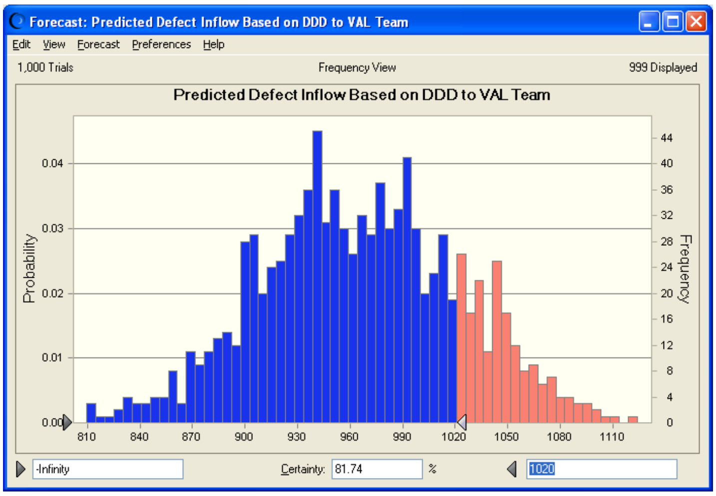

Figure 3: Defect Inflow Prediction based on DDD using Crystal Ball

The Monte Carlo Simulation provides the range of result along with the certainty percentage, in the above chart the certainty of Defect Inflow of 1020 defects is around 82%. The Project Manager can tweak the predicted defect inflow to arrive at desired certainty.

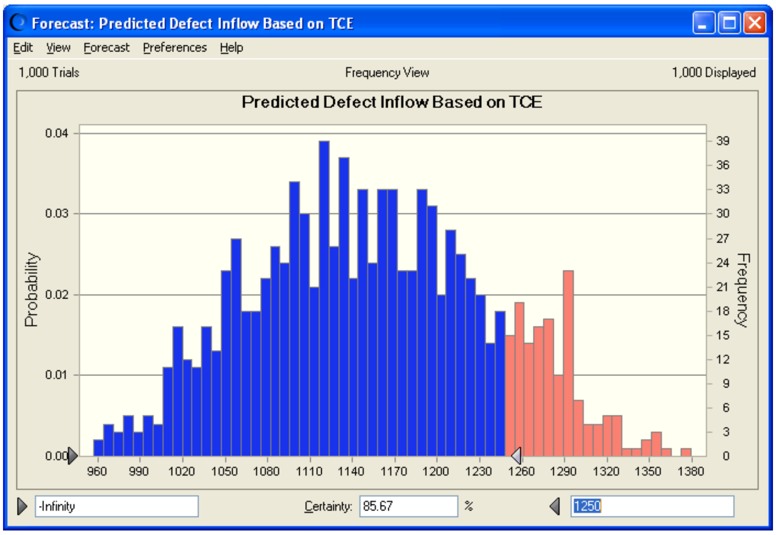

Figure 4: Defect Inflow Prediction based on TCE using Crystal Ball

4. Conclusion

The goal of every software development organization is to deliver good quality product within budget and on time to retain market leadership. Project Manager plays a very significant role in on time delivery, collaborative project management is a practical and effective mechanism in achieving Operational and Business Excellence.

5. References

1. CMMI® for Development, Version 1.3
2. Bruce Barkley, *Integrated project management* 2006
3. Robert W. Stoddard *Exactly What are Process Performance Models in the CMMI?*, SEI June 07
4. Crystal Ball Tool http://www.oracle.com/appserver/businessintelligence/crystalball/index.html
5. S. McConnell, Rapid Development. Redmond: Microsoft Press, 1996.

6. Author's Biography

Ashfaq Ahamed M is an MBA (IT) and has several years of experience in Software Quality Assurance domain. He is currently working as a Program Quality Manager at Nokia Siemens Network, Bangalore. He is well versed with CMMI, ISO 9001:2008, Six Sigma implementation and has played key roles in RNC group as Program Quality Manager, Metrics Consultant and Assessment Team Member.

He is a Certified Scrum Master (CSM), Certified ISO 9001:2008 Lead Auditor, Certified ITIL-SM Foundation v2 and Six Sigma Green Belt.

7. Acronyms

Abbreviation	Description
DDD	Delivered Defect Density
DI	Defect Inflow
KLOC	Kilo Lines Of Code
PPB	Process Performance Baseline
TCE	Test Case Effectiveness