Performance model to predict overall defect density

Dr. J. Venkatesh
Associate Professor, School of Management Studies
Anna University: Chennai 600025, Regional Office Coimbatore
Jothipuram Post, Coimbatore – 641 047, Tamilnadu, India.

Mr. Priyesh Cherurveettil
Ph.D Part Time Research Scholar, School of Management Studies, Anna University: Chennai 600025
Regional Office: Coimbatore, Jothipuram Post, Coimbatore – 641 047, Tamilnadu, India.

Mrs. Thenmozhi. S
Assistant Professor
Department of Computer Applications
Gnanamani College of Technology,
AK Sumuthiram, Namakkal District. Tamilnadu, India.

Dr. Balasubramanie. P
Professor
Department of Computer Science & Engineering
Kongu Engineering College
Perundurai, Erode - 638 052.
Tamilnadu, India

Abstract— **Management by metrics is the expectation from the IT service providers to stay as a differentiator. Given a project, the associated parameters and dynamics, the behaviour and outcome need to be predicted. There is lot of focus on the end state and in minimizing defect leakage as much as possible. In most of the cases, the actions taken are re-active. It is too late in the life cycle. Root cause analysis and corrective actions can be implemented only to the benefit of the next project. The focus has to shift left, towards the execution phase than waiting for lessons to be learnt post the implementation. How do we pro-actively predict defect metrics and have a preventive action plan in place. This paper illustrates the process performance model to predict overall defect density based on data from projects in an organization.**

Keywords- process; performance; defect density; metrics.

I. Introduction to Overall Defect Density

Number of defects leaked into production is a key metric that IT service providers will track month on month and intend to show a downward trend to their clients. Number of defects as a measure alone might not make sense, its relationship with size or effort is important.

For example, if we have 40 defects leaked in January, 30 in February and 5 in March, it doesn’t mean that there is a downward trend. 30 defects could be for a project effort of 1000 hrs whereas 5 defects could be for a project size of 50 hrs. Hence, the metric that need to be closely tracked is overall defect density, number of defects leaked against the project effort.

II. Process Performance Models In Organizations

Process Performance Models (PPM) is probabilistic, statistical and simulative in nature. It can predict interim and final outcome, it is a proactive measure of tracking the end goal instead of a reactive one. It can model the variation of factor and help us understand the predicted range or the variation of its outcomes. Mid-course correction can be made to achieve desired outcome. Interestingly, PPMs enable “What-if” analysis for project planning, dynamic re-planning and problem resolution during project execution. We can run “what if” exercises holding one or more values constant. We can see the effect of tradeoffs between schedule, effort, defects, staff and functionality.

CMMI Dev 1.2 predominantly focuses on development and enhancement type of projects. CMMI for Services focuses on production support or maintenance type of projects. At an organization level, the respective Software Engineering Process Group (SEPG) develops few standard process performance models. These models are developed based on the data gathered from different types of projects within the organization. The organization focuses on three to four key models; typically they are around defect density, productivity, and schedule variance. The important step is to agree to standard definitions for the entities and their measured attributes. When we use terms like defect, productivity, size, and even project, different teams will tend to interpret in their own context. Hence it is important to have a common definition.

Managers should have a good understanding of process performance models and should use it to pro-actively manage the customer needs. Project Managers will check the availability of organizational developed process performance models with respect to the project objectives and quality and process performance objectives. During planning phase, managers are expected to implement process models to manage the objectives. Project manager has to define the project objective and consider the organization objective if there is no client objective defined. The specification limit of the objective is derived from process performance baseline arrived at the organization level. Project Manager is expected to provide the values for the controllable factors in process model prediction section. The prediction intervals are automatically generated based on the values provided

III. Variables Associated In Prediction Model

Based on brain storming session with the project team in the organization the different parameters that influence overall defect density were looked at. The team shortlisted following factors to start with, domain experience, technical experience, defects identified during design and coding phase, overall review efficiency and usage of tools. Operational definitions for these parameters were baseline and data was collected from projects in a particular account against these parameters. Linear regression was performed against the data to find out which are the key variables that influences the overall defect density.

After many trial and error methods the below three variables were established as the x factors.

1. Y - Overall defect density – No of defects identified in the entire life cycle of the project against total effort for the project
2. X1 - Technical experience – Average technical experience of the team, in person months
3. X2 – DDD - Design Defect Density - Defects attributed to design identified during design review against effort spent for design
4. X3 – CDD – Coding Defect Density - Defects attributed to coding, identified during code review against effort spent for coding.

IV. Defect Density – Regression Equation

The project data collated for the x and y factors are as shown in the Table 3.1. Data points from 25 projects in an organization were collected and considered for analysis. Projects factored in were similar in nature.

Y	X1	X2	X3
Overall Defect Density	Technical Experience (in months)	Design Defect Density DDD	Coding Defect Density CDD
0.092	36	1.813	0.231
0.093	31	0.158	0.052
0.095	35	0.258	0.140
0.114	35	0.044	0.083
0.120	47	0.425	0.192
0.119	42	0.458	0.069
0.128	33	0.510	0.250
0.134	40	0.520	0.125
0.126	34	1.525	0.310
0.139	22	0.650	0.055
0.093	30	0.058	0.057
0.064	38	0.143	0.022
0.066	44	0.035	0.213
0.074	54	0.079	0.051
0.080	41	0.390	0.310
0.083	44	0.090	0.290
0.131	33	0.540	0.150
0.134	39	0.510	0.125
0.136	36	1.625	0.300
0.322	12	0.026	0.004
0.565	20	0.558	2.125
0.350	50	5.000	0.500
0.089	59	0.014	0.071
0.083	58	0.435	0.424
0.320	38	0.500	0.500

Table 3.1 – Project data values

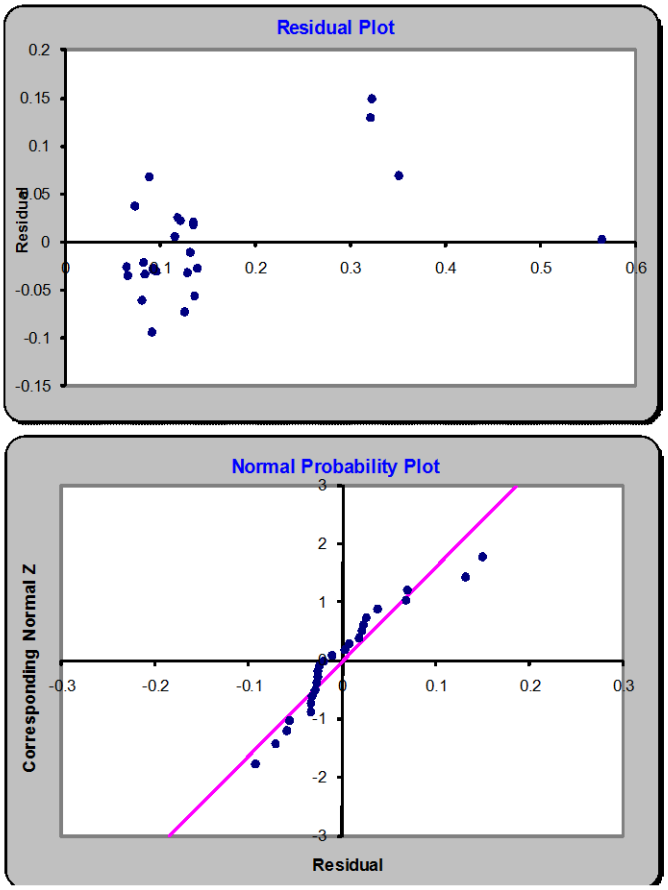

Figure 3.1 – Residual Plot

Mirror pattern is not found in Figure 3.1, Residual Plot and hence no heteroscedasticity is found. The normal probability plot is approximately linear. This would indicate that the normality assumption for the errors has not been violated.

Looking at the p value, since it is 0.0003 which is < 0.05, null hypothesis is not valid, which means the variables selected have an impact to overall defect density

Intercept	Technical Experience (in months)	Design Defect Density DDD	Coding Defect Density CDD
0.21375	-0.0035	0.0295	0.1892

Table 3.2 – Regression Equation

As shown in Table 3.2, technical experience has a negative influence on overall defect density. As the team's technical experience increases the overall defect density is reduced. The influence of Design Defect Density and Coding Defect Density is positive.

This means that when the values of Design Defect Density and Coding Defect Density are low the overall defect density will be low and vice versa.

V. Defect density – components of prediction model

Based on project data analyzed it is evident that overall defect density is critically influenced by design sub process, code sub process and technical experience. Organization Metrics group would help with the baseline data for these metrics. For code sub process and CDD metric, based on the technology (Java, .Net) and review type (manual, tool, FxCop) the baseline values can be tabled. Organization Metrics group will share the baseline values for these combinations. Baseline values will include lower specification limit (LSL), goal and upper specification limit (USL).

The same can be gathered for Design Defect Density as well. Project team needs to choose the process that they would be following for coding or design sub process. Based on the composition of sub process, project goal for DDD and CDD would be calculated. It is also important for the project team to justify why they have gone with a particular sub process and the rationale. Table 4.1 gives the sub process performance baseline for Coding Defect Density and Design Defect Density. The values are represented by A1, A2, A3 and so on. Organization Metrics team would have the actual baseline values for LSL, Goal and USL for these identified metrics. Based on the current project context, the type of technology and review type will selected as shown in Table 4.2, Selected Sub process performance baseline.

Sub process	**Metric**	**Technology**	**LSL**	**Goal**	**USL**
Code Review	CDD	.Net	A1	A2	A3
Code Review	CDD	Java	C1	C2	C3
Code Review	CDD	J2ee	D1	D2	D3
Design Review	DDD	.Net	E1	E2	E3
Design Review	DDD	Java	F1	F2	F3

Table 4.1 - Sub process performance baseline

Sub process	**Metric**	**Technology**	**Goal**	**Comments**
Code Review	CDD	.Net	A2	
Design Review	DDD	.Net	E2	

Table 4.2 – Selected Sub process performance baseline

VI. Defect density model – practical usage

One of the current releases in design phase was considered for the practical usage of this model. The below steps will illustrate the prediction model.

1. X factors baseline data was used as input. Technical experience goal is 28 months with LSL of 14 months and USL of 48 months
2. Sub process performance baseline data was reviewed and based on the current project context the below selection was made. As shown in Table 5.1 the sub process code and design review were selected. Based on the project usage, .net technology was selected. The goal, upper specification limit and lower specification limit are chosen from organization baseline report.

Sub process	**Metric**	**Technology**	**LSL**	**Goal**	**USL**
Code Review	Coding Defect Density (defects/ person day)	.Net	0.10	**0.38**	1.5
Design Review	Design Defect Density (defects/ person day)	.Net	0.15	**0.22**	0.45

Table 5.1 – Selected Sub process

3. Update the actual technical experience in the team and predict the overall defect density. The predicted value is based on Monte Carlo simulation.

Average Technical Experience (in months)	DRDDE (Defects/ Personday)	CRDDE (Defects/ Personday)	Predicted Overall Defect Density
24	0.22	0.38	0.21

Table

Table 5.2 – Predicted Overall Defect Density

4. The data was compared against goal. The client goal for overall defect density is 0.15 whereas the predicted value is 0.21.
5. Perform what-if analysis and look at various combinations of the x factors and analyze the predicted overall defect density based on these factors. Based on the project experience choose the one which is close to reality. In this case the option of 24months technical experience, DDD 0.22 and CDD 0.38 was considered as the closest option.
6. List down the assumptions considered when the final decision is made on the values of x factors. Ensure all the relevant assumptions are documented. As need be, the assumptions need to be validated with the relevant stakeholders before the baseline process.
7. Understand the deviation and prepare preventive action plan

Expected client overall defect density	Predicted overall defect density	Preventive Action	Responsibility
0.15	0.21	List down the top three preventive action items	Project Manager

Table 5.3 – Deviation Analysis

8. Estimated effort in person days for the project is 1000 person days. Based on the predicted defect density and organizations standard effort distribution across phases, the defects that could be injected at each phase are predicted as show in Table 5.4

Phases	**Expected Injection**	**Actual Defects Captured**	**Remarks**
Requirements	12	10	
Design	38	28	
Coding	70		
Unit Testing	62		
System Testing	108		

Table 5.4 – Predicted-Actual Defects phase wise

9. Based on the actual data collated, keep updating Table 5.4 to compare the expected and actual defects captured. Based on the actual value in each phases, the predicted value for next phases are accordingly impacted. If there any specific inputs or considerations on the actual values, those are highlighted in the remarks column.
10. Prepare the detailed defect prevention plan. Against each phase, list down the defect type, defect cause, root cause, preventive action planned, responsible person, target date and the status. Defect types could be incorrect functionality or missing functionality or incorrect user interface or missing user interface. Defect causes could be lack of knowledge, missing information or incidental. Root cause should be as detailed as possible to plan for preventive and corrective action. 5-Why analysis can be used to identify the root causes.

    Defect prevention plan is an on- going document that need to be tracked very closely. It is meant both for planning and tracking defect prevention activities. This plan has to be revisited after completion of each stage. If defects detected during the completed stage fall under different defect types and defect causes not identified for preventing at that stage, then these new types need to be included in the on-going phases.

VII. Conclusion

IT organizations focus on customer satisfaction is the key for survival. Unfortunately the element of predictive behavior during planning phase is very minimal and subjective. While Capability Maturity Model recommends pro-active management using quantitative models, the practical implementation is very low. The context of the organization is important in building these models. It is also important to understand that the project managers need to be equipped with the right information, metrics baseline and subject matter expertise. Defect leakage is a standard concern in the industry. The practical case study demonstrated the influence of Coding Defect Density and Design Defect Density. The case study also helped us understand how the values need to be determined, the steps around what-if analysis, the defect prevention plan and the tracking mechanism. This illustration gives us the confidence that the predictive mechanism can be planned and executed well in an organization.

References

[1] CMMI® Product Development Team, CMMI for Development, Version 1.2, CMMI-DEV, V1.2. Carnegie Mellon University, Pittsburgh, PA., 2006.

[2] Stephen H. Kan, Metrics and Models in Software Quality Engineering 2nd ed., 2002

[3] Richard D. Stutzke, Estimating Software-Intensive Systems: Projects, Products, and Processes, Addison-Wesley, 2005

[4] Robert W. Stoddard, CMMI Process Performance Models and Reliability, Carnegie Mellon University, IEEE Reliability Society Annual Technology Report, 2007

[5] Umesh Kumar Mishra, K. Harihara sudhan, Shalu Gupta. Establishing Process Performance Baselines and Models for Statistical Control of Software Projects, Proceedings of ASCNT-2011, CDAC, Noida, India

[6] Tim Kasse, Practical Insight into CMMI, 2nd ed., Artech House, 2008

[7] Bill Self, Greg Roche and Nigel Hill, Customer Satisfaction Measurement for ISO 9000: 2000, 2002

[8] Michael Johnson and Anders Gustafsson, Improving Customer Satisfaction, Loyalty and Profit, 2000

[9] Margaret K. Kulpa and Kent A. Johnson, Interpreting the CMMI: A Process Improvement Approach, 2nd ed., 2008

[10] Tim Kasse, Practical Insight into CMMI, 2nd ed., 2008

[11] CMMI® Product Development Team, CMMI for Development, Version 1.2, CMMI-DEV, V1.2. by Carnegie Mellon University, Pittsburgh, PA, 2006

[12] Don Peppers and Martha Rogers, Managing Customer Relationships: A Strategic Framework, John Wiley & Sons, 2004

[13] Forrest W. Breyfogle III, Implementing Six Sigma: Smarter Solutions Using Statistical Methods, John Wiley & Sons, 2nd ed., 2003