

Implementing Phase Containment Effectiveness Metrics at Motorola

Ross Seider
On-Fire Associates

Phase Containment Effectiveness (PCE) is a project measurement technique that provides timely and accurate predictions of the project's current state and future risks. PCE methods deliver project insights beyond those provided by the Gantt and Program Evaluation Review Techniques (PERT). PCE provides data that increases the probability of a successful project outcome. This article describes the way in which PCE was implemented at one Motorola business unit.

Two of the most commonly used project management tools are the PERT and Gantt charts. When either is employed, a project's progress can be measured and extrapolations can be made about future project schedule accuracy. These tools focus on milestone achievement and task interdependencies.

The practical shortcomings of PERT and Gantt charts are well known to project managers. Among these are the following:

- **Milestone achievement can be ambiguous.** Often, milestones are qualitative events and opinions on attainment may vary. This ambiguity is not handled well by either tool. The project manager must rely on insight and experience to discern the true state of such milestones.
- **Neither tool inherently measures the quality of the milestone deliverable.** Nor do the tools anticipate downstream risks arising from poor quality of upstream deliverables.

Despite these and other limitations, PERT and Gantt techniques are popular because they do provide useful insight. Project managers can supplement these with other tools to gain further insight into a project's true state.

PCE is a quantitative, real-time measurement technique that addresses these limitations. PCE techniques provide an excellent means to unambiguously judge the completeness and quality levels of certain development milestones. PCE is an inherent indicator of downstream performance giving program managers time to adjust plans.

The PCE method described herein relies on the existence of foundation processes such as repeatable project planning, project tracking and oversight, and quality assurance. At the time of the events described in this article, the Motorola business unit was a Software Engineering Institute Capability Maturity Model® (CMM®)-Level 2 organization with many

established Level 3 capabilities. For Motorola, PCE was a valuable next step in process maturation. Without the foundation processes, taking this step would not have been fruitful.

Theory of Phase Containment

It is a well-established project management axiom that the longer problems go undiscovered, the more costly they will be to correct.

For example, it is far less costly to correct an error in the architect's drawing than in the finished building. The architect's drawing error could have been discovered during a review of the building's design. It also could have been uncovered when the drawings were sent out to potential contractors. The building inspector might have detected the error before issuing the permit. Finally, it might have been spotted by the builders who reviewed the plans prior to the start of construction. A design error going unnoticed at all these review points may be very expensive to correct.

The thesis of phase containment is similar to the previously stated axiom: The most effective projects identify and fix mistakes at the earliest possible point in the project life cycle. This thesis is backed by substantial evidence across virtually all types of development activities. Stated another way: If mistakes are escaping early detection, then the project is at increased risk for missing its time and cost goals. Statistically, if the probability of escapes is increasing, the probability of costly and time-consuming downstream corrections is also increasing.

It was this thesis that convinced senior division management to sponsor a PCE improvement initiative at Motorola. PCE was an attractive initiative because it provided actionable management data that could further improve the effectiveness of the engineering operations.

PCE provides an early warning signal to diligent project managers. It allows them time to react; for instance, by

adding resources to the system testing activities.

Applying PCE

As applied to software projects, PCE is a statistical measurement of the problems (bugs) uncovered at the earliest possible review (or containment) point compared to the total faults uncovered. The more effective the containment (the higher the PCE metric), the more likely the project will proceed smoothly and not experience downstream delays, quality problems, and/or cost overruns.

The PCE metric differentiates between problems discovered at the earliest possible review point to problems that are discovered at subsequent review points. The former problems are referred to as errors and the latter are referred to as defects. The sum of errors and defects represents the total project faults expressed mathematically:

$$\begin{aligned} &\text{Errors (problems discovered at} \\ &\quad \text{the right time)} \\ &\quad + \\ &\text{Defects (problems discovered at} \\ &\quad \text{the wrong time)} \\ &\quad = \\ &\text{Total Project Faults} \end{aligned}$$

Well-executed projects strive to discover a high proportion of the total faults as errors and not as defects. When properly conducted, project design reviews yield this type of data and the risk of downstream problems is reduced.

When faults are discovered during reviews, they must be classified as either errors or defects. This classification is made through a root cause analysis. Once faults are logged into the bug tracking database, the engineers and Quality Assurance (QA) team determine how and when each significant bug was injected into the design. If the root cause analysis determines that the fault should have been uncovered at an earlier review point, the fault is logged as a defect. If the root cause analysis determines that the fault

* Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

could not have been caught earlier, it is logged as an error. While there was incremental effort associated with this categorization, the Motorola development teams rapidly learned how to evaluate the sources of the faults, and the process took little additional time.

Implementing PCE served another Motorola objective: Management was committed to holding comprehensive design reviews. They were keenly interested in evaluating the quality and thoroughness of the design reviews. Phase-containment metrics provided data for this purpose, too. The fault information was combined with 1) the size of the work product being reviewed, 2) the number of people participating in the review, and 3) the time spent conducting the review. The combined data yielded benchmark ratios that allowed QA and project managers to evaluate the comprehensiveness of the review. Review milestones were challenged if the resulting data showed major variances from expected results.

Historical PCE Data

In principal, if a project is uncovering an *unusually large number* of defects, it is probable that additional latent problems exist and will be found downstream. But what constitutes an *unusually large number of defects*? Without a pre-existing comparison, this threshold is subjective. To solve this problem, Motorola reviewed historical fault discovery performance from earlier projects.

In many software organizations, new projects can bear strong resemblance to already completed projects. The resemblance can take many forms:

- The project is a variant of a past project.
- The project is similar in concept to past projects.
- The engineers executing a new project have worked together on previous programs.
- The software tools and processes are

similar to the tools and processes used in the past.

- The new functionality is an add-on to the legacy functionality.

If two or more projects are similar, historical PCE data can add significant insight when interpreting new PCE ratios. Deviations from benchmark trajectories, either positive or negative, form the thresholds for comparison. When the current project's PCE ratios are compared to the norms measured on previous projects, negative deviations are a warning indicator. On the other hand, if the current project's PCE is better than historical benchmarks, then the project manager has evidence of reasonable progress.

PCE initiatives do not automatically require historical data reconstruction. Many companies choose to not implement PCE in this manner. Instead, they opt to accumulate historical data from new projects. While this approach is less burdensome, it takes longer to establish thresholds.

Case Study

Consider a hypothetical project named Reliant Release 4.0. Assume there have been previous releases of Reliant and the historic PCE records of past projects have been saved.

At an early milestone point in the project, the Reliant 4.0 development team created a Release 4.0 functional specification and a high-level design. Both of these have been design reviewed, and the results of the reviews have been captured in Figure 1.

The diagram's rows relate to the different Reliant 4.0 work products. The diagram's columns relate to the design review points. The spreadsheet's cells represent the data collected and analyzed at each review point. The four right-hand columns summarize the results of all the reviews by work product. The final column summarizes the total effectiveness of each review.

At the release of 4.0 functional specification review, six faults were uncovered. This number has been inserted into the chart in the upper left cell.

Sometime thereafter, at the high-level Reliant 4.0 design review, 139 total faults were uncovered. When evaluated for root cause, nine of these faults were attributable to Functional Specification problems (i.e. problems that should have been caught in the earlier functional specification review), and 130 were attributable to problems in the high-level design. Therefore, the numbers nine (the defects that escaped the earlier review) and 130 (the errors uncovered during the high level design review) are inserted into the second column. In a similar manner, as the Reliant 4.0 project continues, further data will be added to the chart.

Based on the data available at this early project stage, the PCE of the Functional Specification is equal to the ratio of the total errors/total faults. Mathematically:

$$\text{PCE} = \# \text{ of errors} / \# \text{ of faults}$$

$$\text{PCE} = 6/15 \text{ (40 percent)}$$

This ratio is shown in the far right-hand side of the table's first row. Notice too that the PCE for the Functional Specification will degrade if further downstream reviews uncover new problems that are attributable to defects in the Functional Specification.

On one level, this PCE data suggests that a moderate amount of time may have been lost in high-level design implementing the wrong functionality. A more refined question to pose to project management is whether this situation is typical of the company's prior projects at a similar point in the development life cycle.

Given the availability of other project data, and the project manager's ability to assess the similarities and differences between Release 4.0 and earlier projects, Release 4.0's data can be put into a richer historical context. For the sake of illustra-

Figure 1: *Current Project*

Work Products	Phase Review Points							Total Errors	Total Defects	Total Faults	Phase Containment Effectiveness
	Functional Spec Review	High-Level Design Review	Low-Level Design Review	Code Review	Module Test	Integration Test	Alpha Test				
Functional Specification	6	9					6	9	15	0.40	
High-Level Design		130					130		130	1.00	
Low-Level Design							0	0	0	N/A	
Code							0	0	0	N/A	
Test Plan							0	0	0	N/A	
Faults by Phase	6	139									

Project Reliant Release 3.0

Phase Review Points

Work Products	Functional Spec Review	High-Level Design Review	Low-Level Design Review	Code Review	Module Test	Integration Test	Alpha Test	Total Errors	Total Defects	Total Faults	Phase Containment Effectiveness
Functional Specification	5	2	1	0	0	0	0	5	3	8	0.63
High-Level Design		78	34	5	15	11	12	78	77	155	0.50
Low-Level Design			112	32	22	13	11	112	78	190	0.59
Code				89	34	12	2	89	48	137	0.65
Test Plan					22	5	2	22	7	29	0.76
Faults by Phase	5	80	147	126	93	41	27				

Figure 2: Benchmark Project

tion, assume that Reliant Release 3.0 is a good comparative benchmark and that Figure 2 shows the phase containment data for the recently completed release 3.0 project.

One can observe in the right-most column that the PCE for Reliant Release 3.0's Functional Specification was 63 percent for the entire duration of the project. But at the time of Release 3.0's high level design review, (the point in time we seek to compare to Release 4.0's data), the Functional Specification's PCE was:

$$\begin{aligned}
 \text{PCE} &= \# \text{ of errors} / \# \text{ of faults} \\
 \text{PCE} &= \# \text{ of errors} / (\# \text{ errors} + \# \text{ defects}) \\
 &= 5 / 7 \text{ (71 percent)}
 \end{aligned}$$

This data comes from the top row on the left-hand side of the Release 3.0 chart.

When comparing the new project to the benchmark project in their early stages, Release 4.0 is experiencing lower phase containment effectiveness than its benchmark project (40 percent compared to 71 percent). Even more glaring, the Release 4.0 Functional Specification PCE is already substantially lower than the final Release 3.0 Functional Specification PCE (40 percent compared to 63 percent). This should be a cause for concern on the new project.

Furthermore, the total number of faults discovered in the Release 4.0 high-level design review raises additional concern. This data is available on the bottom row of each chart.

High-Level Design Review Results:

- Release 4.0 139 faults uncovered
- Release 3.0 80 faults uncovered

This data raises further questions for project management, and possibly greater concerns. For example, is it reasonable that 59 additional faults should be discovered at this phase of Release 4.0 compared to Release 3.0? Were the two high-level design reviews equally thorough? Are there major functionality or complexity

differences between the two releases? Is it possible that the Release 4.0 engineers have been exceptionally thorough and successfully uncovered a higher percentage of latent problems, thereby minimizing the escapes? Or is the data suggesting something more sinister?

Some of these questions cannot be conclusively answered at this point. However, the concern has been noted and the results of the next containment opportunity – the low-level design review – ought to be more closely monitored. If this upcoming review also reveals an unusually high number of escapes from the high-level design, one can reasonably conclude that Release 4.0 is getting into serious trouble.

If Release 4.0 milestones are tracking to expectations, PERT and Gantt tracking tools would not indicate any cause for concern. PCE metrics give the Reliant 4.0 project manager an early warning indicator.

By now, it should be obvious to the reader how phase containment metrics complement Gantt and PERT charts and how they better illuminate certain types of information. But one of the most important benefits is that this data is available early in the project's life cycle.

As organizations become better at collecting and archiving project metrics, their ability to accurately interpret phase containment data improves. Further root cause analysis and process re-engineering can address additional ways to improve operations, namely how to avoid the injection of faults. Ultimately, problem avoidance is the most cost effective way to speed a project.

Conclusions

Understanding the true state of an ongoing project affords organizations the opportunity to take timely corrective action. A project's successful outcome is best ensured by making adjustments at the earliest point possible. When Gantt and/or PERT metrics are combined with

phase containment metrics, far richer project data can be discerned and much of the subjective problems with milestone-based tools can be avoided. If for no other reason than its early warning capability, PCE would have been a valuable addition to Motorola.

PCE's benefits extend beyond this. PCE metrics quantified the effectiveness of the business unit's design reviews. Improving reviews was a key element in Motorola's quality road map. Finally, over time and across multiple projects, PCE results provided quantitative evidence of continuous improvement. Project outcomes were more predictable and fewer faults escaped into customer environments. ♦

About the Author



Ross Seider is president of On-Fire Associates, an executive management consultancy agency that focuses on engineering execution excellence. His career spans 35 years in Boston's high-technology engineering community. Previously Seider was vice president of product development and network operations for Cambridge-based Akamai Technology, and spent 12 years at Motorola in a variety of executive roles. In the 1980s, he co-founded and was vice president of engineering for two successful networking start-ups. Seider holds a bachelor of science in electrical engineering from Rensselaer Polytechnic Institute and a master of business administration from Northeastern University.

184 Brookline ST
 Needham, MA 02492
 Phone: (617) 680-3600
 E-mail: ross@on-fireassociates.com