

## Critical Testing Processes: An Open Source, Business Driven Framework for Improving the Testing Process

When I wrote my book *Critical Testing Processes* in the early 2000s, I started with the premise that some test processes are critical, some are not. I designed this lightweight framework for test process improvement in order to focus the test team and test manager on a few test areas that they simply must do properly. This contrasts with the more expansive and complex models inherent in TPI and TMM. In addition, the Critical Testing Processes (CTP) framework eschews the prescriptive elements of TMM and TPI since it does not impose an arbitrary, staged maturity model.

What's the problem with prescriptive models? In my consulting work, I have found that businesses want to make improvements based on the business value of the improvement and the organizational pain that improvement will alleviate. A simplistic maturity rating might lead a business to make improvements in parts of the overall software process or test process that are actually less problematic or less important than other parts of the process simply because the model listed them in order.

CTP is a non-prescriptive process model. It describes the important software processes and what should happen in them, but it doesn't put them in any order of improvement. This makes CTP a very flexible model. It allows you to identify and deal with specific challenges to your test processes. It identifies various attributes of good processes, both quantitative and qualitative. It allows you to use business value and organizational pain to select the order and importance of improvements. It is also adaptable to all software development lifecycle models.

Since CTP focuses on a small number of processes, how are those processes selected? What is a critical testing process? Here's a quick way to think about it: If a test team executes the critical testing processes well, it will almost always succeed, but if it executes these activities poorly, even talented individual testers and test managers will usually fail. Let's expand this definition a bit.

First, I defined a process as some sequence of actions, observations, and decisions. Next, I defined testing as the activities involved in planning, preparing, performing, and perfecting the assessing of the quality of a system. So, with a definition of a test process firmly in hand, what makes a test process a critical test process? I applied four criteria:

- Is the process repeated frequently, so that it affects efficiency of the test team and the project team?
- Is the process highly cooperative, involving a large number of people, particularly cross-functionally, so that it affects test team and project team cohesion and cooperation?
- Is the process visible to peers and superiors, so that it affects the credibility of the test team?

- Is the process linked to project success, in such a way as to affect project team or test team effectiveness?

In other words, a critical test process directly and significantly affects the test team's ability to find bugs, build confidence, reduce risks, and generate information.

Based on these criteria, I identified the following twelve critical testing processes:

**Testing.** The overall process, viewed at a macro, strategic level. It consists of eleven constituent critical testing processes.

**Establishing context.** This process aligns testing within the project and the organization. It clarifies expectations on all sides. It establishes the groundwork for tailoring all other testing processes. As you can see, when appropriate, the creation of test policy and test strategy documents would fit into this process. However, those are not prescribed since some situations might need them and some might not.

**Quality risk analysis.** This process identifies the key risks to system quality. It aligns testing with the key risks to system quality. It builds quality and test stakeholder consensus around what is to be tested (and how much) and what is not to be tested (and why).

**Test estimation:** This process balances the costs and time required for testing against project needs and risks. It accurately and actionably forecasts the tasks and duration of testing. It demonstrates the return on the testing investment to justify the amount of test work requested.

**Test planning.** This process builds consensus and commitment among test team and broader project team participants. It creates a detailed map for all test participants. It captures information for retrospectives and future projects.

**Test team development.** Since testing is only as good as the team that does it, this process matches test team skills to the critical test tasks. It assures competence in the critical skills areas. It continuously aligns team capabilities with organizational value of testing.

**Test system development.** This process ensures coverage of the critical risks to system quality. It creates tests that reproduce the customers' and users' experiences of quality. It balances resource and time requirements against criticality of risk. It includes test cases, test data, test procedures, test environments, and other support material.

**Test release management.** If we don't have the test object, we can't test it. If the test items don't work in the test environment, we can't test them. If each test release is not better than the one before, we're not on a path for success. So, this process focuses on how to get solid, reliable test releases into the test environment.

**Test execution.** This process, the running of test cases and comparison of test results against expected results, generates information about bugs, what works, and what doesn't. In other words, this is where the value of testing is created. This process

consumes significant resources. It occurs at the end of the project and gates project completion.

Bug reporting. This process creates an opportunity to improve the system (and thus to save money). While test execution generates the value of testing, this process delivers part of the value of testing to the project team, specifically the individual contributors and line managers. It builds tester credibility with programmers.

Results reporting. This process provides management with the information needed to guide the project. It delivers another part of the value of testing to the project team, particularly line managers, senior managers, and executives. Since test results are often bad news, it separates the message from messenger. It builds tester credibility with managers

Change management. This process allows the test team and the project team to respond to what they've learned so far. It selects the right changes in the right order. It focuses efforts on the highest return-on-investment activities.

You might notice that I've described each of these processes in terms of what an optimal process will achieve. If the process does not achieve those standards of capability – and more – then it is not optimal and has room for improvement.

How do we use CTP for assessment and improvement? Test process improvements using CTP begin with an assessment of the existing test process. This assessment will identify which of the twelve test processes are currently done properly and which need improvement. The assessment results in a set of prioritized recommendations for improvements. The assessor bases those recommendations on organizational needs, not some maturity model.

Since the assessments are tailorable, they can vary depending on the specific organization and its needs. We have done narrowly focused CTP assessments that looked only at one test team, we have done CTP assessments that looked only at a one or two test processes like the test system, and we have done broad CTP assessments that looked at everything that affects quality. So, while CTP assessments vary, we tend to examine the following metrics during a CTP assessment:

- Defect detection percentage
- Return on the testing investment
- Requirements coverage and risk coverage
- Test release overhead
- Defect report rejection rate

In addition, we also tend to evaluate the following qualitative factors, among others, during a CTP assessment:

- Test team role and effectiveness

- Usefulness of the test plan utility
- Test team skills in testing, domain knowledge, and technology
- Value of the defect reports
- Usefulness of test result reports
- Change management utility and balance

Once an assessment has identified opportunities for improvement, the assessor will develop plans to implement this improvement. While the model includes generic guidelines for process improvement for each of the critical testing processes, the assessment team is expected to tailor those heavily.

I designed the CTP model to be very flexible. It does assume the primary use of an analytical risk-based testing strategy, balanced with a dynamic testing strategy. You can adapt CTP to use other test strategies primarily, such as requirements based, checklist based, or model based, but that would have a systemic impact on the model. In the next edition of *Critical Testing Processes*, I will build that additional element of flexibility into the model.

In Figure 1, you see the summary results of a CTP assessment for a client. As you can see, our evaluation is fine grained. In addition, we don't rate the process areas as fulfilled or unfulfilled, but rather given them a qualitative evaluation that helps interpret the assessment. We specifically identify opportunities to improve, sometimes in specific areas of the process. The assessment is not only a snapshot in time, but it captures the momentum of the organization in a specific process area.

1. Value of the Test Team and Process: Good.
2. Alignment of the Test Team with the Organization: Needs improvement.
3. Alignment of Test with Quality: Opportunity to improve.
4. Test Estimation: Opportunity to improve.
5. Test Planning: Generally strong, but regression strategy needs improvement.
6. Test Team Development: Evolving positively; optimize with direction.
7. Test System Development: Improving.
8. Test Releases Management: Excellent.
9. Running Tests: Improving.
10. Reporting Bugs: Inconsistent; presents opportunities to improve.
11. Test Results Reporting: Incomplete; inadequate for many stakeholders.
12. Change Management and Bug Triage Process: Needs improvement.

## Figure 1: CTP assessment summary

In Figure 2, you see an excerpt from RBCS' assessment guidebook that addresses a specific quantitative metric which we use to measure the test development process. The metric looks at test case detail. You can substitute "test procedure" if that's the term you use for the actual unit of work assigned to a tester executing a manual test. The more words per unit of execution time in the written test case or procedure, the more explicitly specified the test is and thus the more instruction and guidance we are giving the tester.

- ✦ Metric: Test Case Detail Density
- ✦ Use: Evaluates level of detail in test cases
- ✦ Data needed: Number of words in a test case and the number of minutes required to execute it (manual tests only), for at least ten test cases spanning testers and applications

- ✦ Formula: 
$$\text{detail density} = \frac{\text{\# of words}}{\text{\# of minutes}}$$

- ✦ Notes
  - ❖ The larger the sample, the better
  - ❖ Calculate average and standard deviation

## Figure 2: Measuring processes in CTP

The point is to evaluate whether the level of detail in the test cases is suitable for needs. You see, CTP prescribes neither a lightweight documentation regime nor a heavyweight one but instead measures to see the actual current state of documentation and then evaluates that against the business needs. For this specific metric, we are looking at test cases or test procedures, depending on the naming convention at the assessed organization.

To calculate this metric, we take a sample of at least ten test cases. We count the number of words in each test case and the number of minutes required to execute it. We also calculate the average and standard deviation. What we often see during assessments is an order-of-magnitude variation between tests, or sometimes even a two- or three-orders-of-magnitude variation. That's fine if the variation is carefully considered and managed, but generally it arises from the lack of clear management direction on this

topic as well as the lack of any understanding by the individual contributors as to what drives documentation density needs in one direction or another.

There are some five to ten metrics for each critical testing process, along with some qualitative evaluations. Now, we don't typically look at every metric for every process on every single assessment because the selected metrics, like the model itself, are tunable. This results in a customer-centric assessment.

So, what is the value of a CTP assessment? CTP, like any other process model, provides a starting point, a standard framework, and a way of measuring your processes. A process assessment using a process model identifies opportunities to improve your current process that could not be identified by applying continuous process improvement techniques, such as Deming's Plan-Do-Check-Act (PDCA), to your current existing processes. This is true because techniques such as PDCA are incremental improvement techniques, while process models allow provide a method for quantum leaps in process improvement through the introduction of known best practices.

CTP assessments deliver specific recommendations, along with the order in which to implement them. When my associates and I carry out CTP assessments for clients, we typically deliver a 50 to 100 page report with our assessment of the critical testing processes, our recommendations for improving them, the business justification for the improvements, and a roadmap for implementing those improvements. If you look at any two of our assessment reports, you might see very similar recommendations but in very different order. Why? Because each client has a different level of opportunity associated with the recommendation. In some cases, constraints or preconditions can influence the order. Those constraints and levels of opportunity tend to be unique from one organization to another, and a non-prescriptive model adapts to those unique organizational needs.

Let me close with an important note: Whether you use CTP or some other test process assessment model, don't use process assessments as a one-time activity. Having done an assessment and found opportunities to improve, you should – of course – improve. Furthermore, at regular intervals, you should reassess to see the effect of the changes. Based on that reassessment, you should course correct your process improvement.

## **Bio**

With a quarter-century of experience, Rex Black is President of RBCS ([www.rbc-us.com](http://www.rbc-us.com)), a leader in software, hardware, and systems testing. For over fifteen years, RBCS has delivered consulting, outsourcing and training services to clients ranging from Fortune 20 companies to start-ups. Rex is also the immediate past President of the International Software Testing Qualifications Board and the American Software Testing Qualifications Board. Rex has published six books which have sold over 50,000 copies, including Japanese, Chinese, Indian, Hebrew, and Russian editions. He has written over thirty articles, presented hundreds of papers, workshops, and seminars, and given

about fifty speeches at conferences and events around the world. Rex may be reached at [rex\\_black@rbc-us.com](mailto:rex_black@rbc-us.com).