

USING THE PRAGMATIC SYSTEMS TESTING^Ô METHODOLOGY

Targeted Audience: Executives/Managers/Testers

I. Definition of Pragmatic Systems Testing^Ô

Pragmatic Systems Testing™ (PST) is an engineering approach to testing software and systems. It's formulated on a balanced approach to testing coupled with the real-world need to ensure that testing is high quality, relevant, and defined. PST incorporates ideas and methods from many testing experiences, techniques, methodologies, standards, and practices, and distills it into a set of value-added test tasks, processes, and activities. It works within the constraints of scope, personnel, and schedule using a holistic approach with a minimal set of requirements and documentation.

PST embraces value-add over useless process, creativity over control, trade-offs over soapboxing, risk mitigation over disregard, and pragmatism over idealism. PST is especially suited for environments, teams, and companies that are creative and agile but still need a formal test and release project phase for their new products. It can be incorporated into existing company processes and methodologies using applicable parts or even just using the basic philosophy of plan, execute, and improve.

II. Pragmatic Systems Testing^Ô Goals

The goals of system and software testing are to uncover product defects, verify product requirements and functionality, and exercise the system. It is imperative that the costs associated with testing personnel, schedule, tools, and infrastructure be identified and contained while the testing scope, test activities, and responsibilities of system test are properly identified and agreed to by suppliers and consumers.

In the real world, product requirements are vague, schedules shift, development is problematic, and not enough time and resources are typically available to the test group. Some of the typical testing consequences of this reality are:

- Open-ended testing with no clear completion goals;
- Reliance on unplanned and ad hoc testing for test coverage;
- Unrealistic test schedules and testing expectations;
- Testing is not focused in the high-risk areas of the product.

PST defines a minimal set of tasks, activities, and infrastructure for system testing that can ameliorate typical testing problems, reduce program risks, and contain testing costs. It is based on a flexibility to adapt the system testing efforts to factors such as company culture, type of product, test personnel experience and knowledge, and availability of tools. The end results are to have a plan, agreement, and cost/schedule estimates for systems testing prior to execution, and then criteria and processes for test execution and completion.

III. The Pragmatic Systems Testing[®] Methodology

The PST approach is to plan the test effort, execute the test plans and test procedures, complete the systems testing effort, and then improve the plans, tools, and processes. This simple approach is difficult to implement without some basic testing activities, components, and infrastructure. PST has the following components:

1. **System Architecture, Design, & Usage Docs** – Systems test planning requires a starting point for identifying what needs to be tested. It's expected that architecture, functional design, and use case documents (as a minimum) will have been written and can be used as the basis for developing the System Test Plan.
2. **Test Plan** – A Test Plan document defines **what** system testing will be done for this product. The Plan covers and addresses functionality and compatibility testing and also security testing and usability testing. Additionally, sections covering load testing, performance testing, and reliability testing (and any other applicable "...ilities" testing) are included if appropriate. Test tools that are needed for system test are identified within the Test Plan and an assessment determines whether tools will be purchased, developed, or revised.
3. **Test Procedures** – Test Procedures are a set of documents covering **how** key areas and complex parts of the product will be tested. Test Procedures are written to a level of detail that is appropriate for the experience level of the test execution personnel and to a level that allows an estimate of the number of test cases.
4. **Testing Infrastructure** – A system testing environment and infrastructure needs to be architected to ensure that it is effective. The testing infrastructure should be representational (replicates, simulates, or emulates) of the production or customer environment. Key attributes of the environment should include the capability to collect data and perform measurement during test, ability to scale the testing infrastructure, and flexibility to reconfigure.
5. **System Test Schedule** – The System Test schedule uses the number of test cases as the basis for determining a test schedule and resource costs. At least one, and preferably two, system test cycles are needed to be included in the schedule. Another shorter test cycle is used for a "golden" test run. Finally a user or customer acceptance test is a short set of tests demonstrating the product at the end of system testing. The schedule estimates are combined with testing resource estimates to yield an overall system test schedule and test manpower loading profile.
6. **Test Completion Goals & Criteria** – Test completion goals and completion criteria are formulated based on a number of factors such as the type of product and usage, domain experience/knowledge, quality goals, test coverage, and identified risks and mitigation plans. For each systems test cycle, testing progress data is collected and evaluated against the completion goals. For systems test completion, testing progress, product readiness, and failure rates are some of the criteria that can be used to make that decision.
7. **Defect Tracking System** – A Defect Tracking System is required for collecting, managing, and reporting product defects during system test. Data can be analyzed

to determine testing efficacy, defect densities, and other metrics associated with system testing.

8. **Core Team** – During formal system testing cycles, a multi-functional group meets as needed to triage open defects. This team has members from Test, Development, Marketing, and Project Management as a minimum. The Core Team assesses open defects and determines the product impact and the business necessity, priority, and timing to fix these defects.

PST does not preclude expected changes to the product requirements, design, or delivery schedules during product design and development. It's anticipated that things change in any new product development. A pragmatic approach to systems testing can work within the constraints of time, money, and resources to deliver a high quality test effort that has been properly engineered.

IV. Summary

Pragmatic Systems Testing™ is an approach and methodology for planning, executing, and completing systems testing. This document is a brief introduction and overview of the methodology. More details, samples, examples, and descriptions will continue to be made available on the Round Rock Systems Test web site. <http://www.rrstinc.com/>