

Exploratory Testing Dynamics

Created by James and Jonathan Bach¹

v1.5

Copyright © 2005-2006, Satisfice, Inc.

Exploratory testing is the opposite of *scripted* testing. Both scripted and exploratory testing are better thought of as test *approaches*, rather than techniques. This is because virtually any test technique can be performed in either a scripted or exploratory fashion. Exploratory testing is often considered mysterious and unstructured. Not so! You just need to know what to look for. The following are some of the many dynamics that comprise exploratory testing.

Evolving Work Products

Exploratory testing spirals upward toward a complete and professional set of test artifacts. Look for any of the following to be created or refined during an exploratory test session.

	Test Ideas. Tests, test cases, test procedures, or fragments thereof.
	Testability Ideas. How can the product be made easier to test?
	Bugs. Anything about the product that threatens its value.
	Risks. Any potential areas of bugginess or types of bug.
	Issues. Any questions regarding the test project, or matters to be escalated.
	Test Coverage Outline. Aspects of the product we might want to test.
	Test Data. Any data developed for use in tests.
	Test Tools. Any tools acquired or developed to aid testing.
	Test Strategy. The set of ideas that guide our test design.
	Test Infrastructure and Lab Procedures. General practices or systems that provide a basis for excellent testing.
	Test Estimation. Ideas about what we need and how much time we need.
	Test Process Assessment. Our own assessment of the quality of our test process.
	Testing Narrative. The story of our testing so far.
	Tester. The tester evolves over the course of the project.
	Test Team. The test team gets better, too.
	Developer Relations. As you test, you also get to know the developer.

¹ The participants in the Exploratory Testing Research Summit also helped shape and test this document. They included: James Bach, Jonathan Bach, Mike Kelly, Cem Kaner, Michael Bolton, James Lyndsay, Elisabeth Hendrickson, Jonathan Kohl, Robert Sabourin, and Scott Barber

Exploration Skills and Tactics

These are the skills that comprise professional exploration of technology:

	Modeling. Composing, describing, and working with mental models of the things you are exploring. Identifying relevant dimensions, variables, and dynamics. A good mental model may manifest itself as having a “feel” for the product; intuitively grasping how it works.
	Resourcing. Obtaining tools and information to support your effort. Exploring sources of such tools and information. Getting people to help you.
	Questioning. Identifying missing information, conceiving of questions, and asking questions in a way that elicits the information that you seek.
	Chartering. Making your own decisions about what you will work on and how you will work. Understanding your client’s needs, the problems you must solve, and assuring that your work is on target.
	Observing. Gathering empirical data about the object of your study; collecting different kinds of data, or data about different aspects of the object; establishing procedures for rigorous observations.
	Manipulating. Making and managing contact with the object of your study; configuring and interacting with it; establishing procedures for better control of experimental conditions.
	Collaboration. Working and thinking with other people on the same problem; elaborating off of other people’s ideas.
	Generating/Elaborating. Working quickly in a manner good enough for the circumstances. Revisiting the solution later to extend, refine, refactor or correct it.
	Overproduction/Abandonment. Producing many different speculative ideas and making speculative experiments, more than you probably need, then abandoning what doesn’t work. Examples are brainstorming, trial and error, “bracketing” in photography, genetic algorithms, free market dynamics.
	Abandonment/Recovery. Abandoning ideas and materials in such a way as to facilitate their recovery, should they need to be revisited. Maintaining a “boneyard” of old ideas.
	Refocusing. Managing the scope and depth of your attention. Looking at different things, looking for different things, in different ways.
	Alternating. Switching among different activities or perspectives to create or relieve productive tension and make faster progress. See <i>Exploratory Testing Polarities</i> .
	Branching/Backtracking. Allowing yourself to be productively distracted from one course of action in order to explore an unanticipated new idea. Identifying opportunities and pursuing them without losing track of the process.
	Conjecturing. Considering possibilities and probabilities. Considering multiple, incompatible explanations that account for the same facts.
	Recording. Preserving information about your process, progress, and findings. Taking notes.
	Reporting. Making a credible, professional report of your work to your clients in oral and written form.

Exploratory Testing Polarities

To develop ideas or search a complex space quickly yet thoroughly, not only must you look at the world from many points of view and perform many kinds of activities (which may be polar opposites), but your mind may get sharper from the very act of switching from one kind of activity to another. Here is a partial list of polarities:

	Warming up vs. cruising vs. cooling down
	Doing vs. describing
	Doing vs. thinking
	Careful vs. quick
	Data gathering vs. data analysis
	Working with the product vs. reading about the product
	Working with the product vs. working with the developer
	Product vs. project
	Solo work vs. team effort
	Your ideas vs. other peoples' ideas
	Lab conditions vs. field conditions
	Current version vs. old versions
	Feature vs. feature
	Requirement vs. requirement
	Test design vs. execution
	Coverage vs. oracles
	Testing vs. touring
	Individual tests vs. general lab procedures and infrastructure
	Testing vs. resting

Testing Considerations

This is a compressed version of the Satisfice Heuristic Test Strategy model. It's a set of considerations designed to help you test robustly or evaluate someone else's testing.

Project Environment

- Customers.* Anyone who is a client of the test project.
- Information.* Information about the product or project that is needed for testing.
- Developer Relations.* How you get along with the programmers.
- Test Team.* Anyone who will perform or support testing.
- Equipment & Tools.* Hardware, software, or documents required to administer testing.
- Schedules.* The sequence, duration, and synchronization of project events.
- Test Items.* The product to be tested.
- Deliverables.* The observable products of the test project.

Product Elements

- Structure.* Everything that comprises the physical product.
- Functions.* Everything that the product does.
- Data.* Everything that the product processes.
- Platform.* Everything on which the product depends (and that is outside your project).
- Operations.* How the product will be used.
- Time.* Any relationship between the product and time.

Quality Criteria Categories

- Capability.* Can it perform the required functions?
- Reliability.* Will it work well and resist failure in all required situations?
- Usability.* How easy is it for a real user to use the product?
- Security.* How well is the product protected against unauthorized use or intrusion?
- Scalability.* How well does the deployment of the product scale up or down?
- Performance.* How speedy and responsive is it?
- Installability.* How easily can it be installed onto its target platform?
- Compatibility.* How well does it work with external components & configurations?
- Supportability.* How economical will it be to provide support to users of the product?
- Testability.* How effectively can the product be tested?
- Maintainability.* How economical is it to build, fix or enhance the product?
- Portability.* How economical will it be to port or reuse the technology elsewhere?
- Localizability.* How economical will it be to publish the product in another language?

General Test Techniques

- Function Testing.* Test what it can do.
- Domain Testing.* Divide and conquer the data.
- Stress Testing.* Overwhelm the product.
- Flow Testing.* Do one thing after another.
- Scenario Testing.* Test to a compelling story.
- Claims Testing.* Verify every claim.
- User Testing.* Involve the users.
- Risk Testing.* Imagine a problem, then find it.
- Automatic Testing.* Write a program to generate and run a zillion tests.