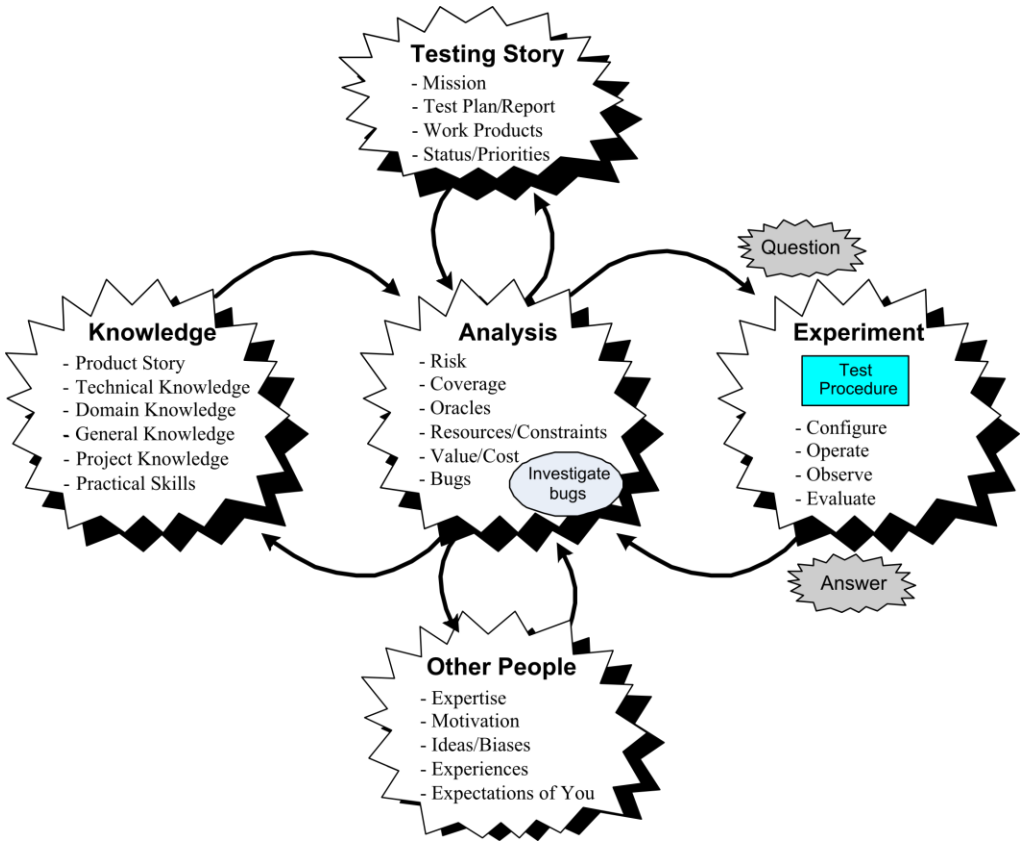
Exploratory Testing Dynamics

Created by James Bach, Jonathan Bach, and Michael Bolton[1] *v3.0*

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 diagram below shows the main elements of exploratory testing modeled as a set of cycles:



In any competent process of testing that is done in an exploratory way, you can expect to find these elements. The arrows represent dynamic influences of the elements on each other, mediated by various forms of thinking. For instance:

Learning: The cycle between analysis and knowledge might be called the learning loop. In this interaction the tester is reviewing and thinking about, and applying what he knows.

Testing: The cycle between analysis and experiment might be called the testing loop. It is dominated by questions which guide the gathering of evidence about the product.

Collaboration: The cycle between analysis and other people might be called the collaboration loop. Collaboration is not necessarily a part of exploration, but often is, especially in larger projects.

Self-management: The cycle between analysis and the testing story is self-management, by which the whole process is regulated.

---

[1] The participants in the Exploratory Testing Research Summit #1 also reviewed 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

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, refined, and possibly documented during the process.

	Test Ideas. Tests, test cases, test procedures, or fragments thereof.
	Testability Ideas. How can the product be made easier to test?
	Test Results. We may need to maintain or update test results as a baseline or historical record.
	Bug Reports. Anything about the product that threatens its value.
	Issues. Anything about the project that threatens its value.
	Test Coverage Outline. Aspects of the product we might want to test.
	Risks. Any potential areas of bugginess or types of bug.
	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, protocols, controls, and systems that provide a basis for excellent testing.
	Test Estimation. Ideas about what we need and how much time we need.
	Testing Story. What we know about our testing, so far.
	Product Story. What we know about the product, so far.
	Test Process Assessment. Our own assessment of the quality of our test process.
	Tester. The tester evolves over the course of the project.
	Test Team. The test team gets better, too.
	Developer and Customer Relationships. As you test, you also get to know the people you are working with.

Exploration Skills

These are the skills that comprise professional and cost effective exploration of technology. Each is distinctly observable and learnable, and each is necessary for excellent exploratory work:

Self-Management

	Chartering your work. Making 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.
	Establishing procedures and protocols. Designing ways of working that allow you to manage your study productively. This also means becoming aware of critical patterns, habits, and behaviors that may be intuitive and bringing them under control.
	Establishing the conditions you need to succeed. Wherever feasible and to the extent feasible, establish control over the surrounding environment such that your tests and observations will not be disturbed by extraneous and uncontrolled factors.
	Maintaining self-awareness. Monitoring your emotional, physical, and mental states as they influence your exploration.
	Behaving ethically. Understanding and fulfilling your responsibilities under any applicable ethical code during the course of your exploration.
	Monitoring issues in the exploration. Maintaining an awareness of potential problems, obstacles, limitations and biases in your exploration. Understanding the cost vs. value of the work.
	Branching your work and backtracking. Allowing yourself to be productively distracted from a course of action to explore an unanticipated new idea. Identifying opportunities and pursuing them without losing track of your process.
	Focusing your work. Isolating and controlling factors to be studied. Repeating experiments. Limiting change. Precise observation. Defining and documenting procedures. Using focusing heuristics.
	De-focusing your work. Including more factors in your study. Diversifying your work. Changing many factors at once. Broad observation. Trying new procedures. Using defocusing heuristics.
	Alternating activities to improve productivity. Switching among different activities or perspectives to create or relieve productive tension and make faster progress. See *Exploratory Testing Polarities*.
	Maintaining useful and concise records. Preserving information about your process, progress, and findings. Note-taking.
	Deciding when to stop. Selecting and applying stopping heuristics to determine when you have achieved good enough progress and results, or when your exploration is no longer worthwhile.

Collaboration

	Getting to know people. Meeting and learning about the people around you who might be helpful, or whom you might help. Developing a collegial network within your project and beyond it.
	Conversation. Talking through and elaborating ideas with other people.
	Serving other testers. Performing services that support the explorations of other testers on their own terms.
	Guiding other testers. Supervising testers who support your explorations. Coaching testers.
	Asking for help. Articulating your needs and negotiating for assistance.
	Telling the story of your exploration. Making a credible, professional report of your work to your clients in oral and written form that explains and justifies what you did.
	Telling the product story. Making a credible, relevant account of the status of the object you are studying, including bugs found. This is the ultimate goal for most test projects.

Learning

	Discovering and developing resources. Obtaining information or facilities to support your effort. Exploring those resources.
	Applying technical knowledge. Surveying what you know about the situation and technology and applying that to your work. An expert in a specific kind of technology or application may explore it differently.
	Considering history. Reviewing what's been done before and mining that resource for better ideas.
	Using Google and the Web. Of course, there are many ways to perform research on the Internet. But, acquiring the technical information you need often begins with Google.
	Reading and analyzing documents. Reading carefully and analyzing the logic and ideas within documents that pertain to your subject.
	Asking useful questions. Identifying missing information, conceiving of questions, and asking questions in a way that elicits the information you seek.
	Pursuing an inquiry. A line of inquiry is a structure that organizes reading, questioning, conversation, testing, or any other information gathering tactic. It is investigation oriented around a *specific* goal. Many lines of inquiry may be served during exploration. This is, in a sense, the opposite of practicing curiosity.
	Indulging curiosity. Curiosity is investigation oriented around this *general* goal: to learn something that might be useful, at some later time. This is, in a sense, the opposite of pursuing a line of inquiry.
	Generating and elaborating a requisite variety of ideas. Working quickly in a manner good enough for the circumstances. Revisiting the solution later to extend, refine, refactor or correct it.
	Overproducing ideas for better selection. Producing many different speculative ideas and making speculative experiments, more than you can elaborate upon in the time you have. Examples are brainstorming, trial and error, genetic algorithms, free market dynamics.
	Abandoning ideas for faster progress. Letting go of some ideas in order to focus and make progress with other ones.
	Recovering or reusing ideas for better economy. Revisiting your old ideas, models, questions or conjectures; or discovering them already made by someone else.

Testing

	Applying tools. Enabling new kinds of work or improving existing work by developing and deploying tools.
	Interacting with your subject. Making and managing contact with the subject of your study; for technology, configuring and operating it so that it demonstrates what it can do.
	Creating models and identifying relevant factors for study. Composing, decomposing, describing, and working with mental models of the things you are exploring. Identifying relevant dimensions, variables, and dynamics.
	Discovering and characterizing elements and relationships within the product. Analyze consistencies, inconsistencies, and any other patterns within the subject of your study.
	Conceiving and describing your conjectures. Considering possibilities and probabilities. Considering multiple, incompatible explanations that account for the same facts. Inference to the best explanation.
	Constructing experiments to refute your conjectures. As you develop ideas about what's going on, creating and performing tests designed to disconfirm those beliefs, rather than repeating the tests that merely confirm them.
	Making comparisons. Studying things in the world with the goal of identifying and evaluating relevant differences and similarities between them.
	Detecting potential problems. Designing and applying oracles to detect behaviors and attributes that may be trouble.

	Observing what is there. 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.
	Noticing what is missing. Combining your observations with your models to notice the significant absence of an object, attribute, or pattern.

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
	Training (or learning) vs. performing
	Product focus vs. project focus
	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
	Coverage vs. oracles
	Testing vs. touring
	Individual tests vs. general lab procedures and infrastructure
	Testing vs. resting
	Playful vs. serious

Test Strategy

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

- ❑ *Mission.* The problems you are commissioned to solve for your customer.
- ❑ *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 it 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 Checking.* Write a program to generate and run a zillion checks.