

# What is Exploratory Testing?

## And How it Differs from Scripted Testing

By James Bach

Exploratory software testing is a powerful and fun approach to testing. In some situations, it can be orders of magnitude more productive than scripted testing. I haven't found a tester yet who didn't, at least unconsciously, perform exploratory testing at one time or another. Yet few of us study this approach, and it doesn't get much respect in our field. It's high time we stop the denial, and publicly recognize the exploratory approach for what it is: scientific thinking in real-time. Friends, that's a good thing.

### Concurrent Test Design and Execution

The plainest definition of exploratory testing is test design and test execution at the same time. This is the opposite of scripted testing (predefined test procedures, whether manual or automated). Exploratory tests, unlike scripted tests, are not defined in advance and carried out precisely according to plan. This may sound like a straightforward distinction, but in practice it's murky. That's because "defined" is a spectrum. Even an otherwise elaborately defined test procedure will leave many interesting details (such as how quickly to type on the keyboard, or what kinds of behavior to recognize as a failure) to the discretion of the tester. Likewise, even a free-form exploratory test session will involve tacit constraints or mandates about what parts of the product to test, or what strategies to use. A good exploratory tester will write down test ideas and use them in later test cycles. Such notes sometimes look a lot like test scripts, even if they aren't. Exploratory testing is sometimes confused with "ad hoc" testing. Ad hoc testing normally refers to a process of improvised, impromptu bug searching. By definition, anyone can do ad hoc testing. The term "exploratory testing"--coined by Cem Kaner, in *Testing Computer Software*--refers to a sophisticated, thoughtful approach to ad hoc testing. In the last decade, James Whittaker (at Florida Tech), Cem Kaner and I have worked to identify the skills and techniques of excellent exploratory testing. For one example of a fully defined and articulated process of exploratory testing, see the General Functionality and Stability Test Procedure for Microsoft's Windows 2000 Compatibility Certification program. This document is publicly available on Microsoft's web site, or on my site at <http://www.satisfice.com/tools/procedure.pdf>.

### Balancing Exploratory Testing With Scripted Testing

To the extent that the next test we do is influenced by the result of the last test we did, we are doing exploratory testing. We become more exploratory when we can't tell what tests should be run, in advance of the test cycle, or when we haven't yet had the opportunity to create those tests. If we are running scripted tests, and new information comes to light that suggests a better test strategy, we may switch to an exploratory mode (as in the case of discovering a new failure that requires investigation). Conversely, we take a more scripted approach when there is little uncertainty about how we want to test, new tests are relatively

## White Paper

unimportant, the need for efficiency and reliability in executing those tests is worth the effort of scripting, and when we are prepared to pay the cost of documenting and maintaining tests.

The results of exploratory testing aren't necessarily radically different than those of scripted testing, and the two approaches to testing are fully compatible. Companies such as Nortel and Microsoft commonly use both approaches on the same project. Still there are many important differences between the two approaches.

### **Why Do Exploratory Testing?**

Recurring themes in the management of an effective exploratory test cycle are tester, test strategy, test reporting and test mission. The scripted approach to testing attempts to mechanize the test process by taking test ideas out of a test designer's head and putting them on paper. There's a lot of value in that way of testing. But exploratory testers take the view that writing down test scripts and following them tends to disrupt the intellectual processes that make testers able to find important problems quickly. The more we can make testing intellectually rich and fluid, the more likely we will hit upon the right tests at the right time. That's where the power of exploratory testing comes in: the richness of this process is only limited by the breadth and depth of our imagination and our emerging insights into the nature of the product under test. In the rapid testing classes at Satisfice, Inc., we have equipment that watches testers invent tests in real-time. When the instructor makes a new suggestion for what to test, or provides new information to the testers about the product, we observe and measure how a roomful of exploratory testers reacts to that information. Free from the encumbrance of pre-documentation, they immediately incorporate new ideas into their tests.

Scripting has its place. I can imagine testing situations where efficiency and repeatability are so important that we should script or automate them. For example, in the case where a test platform is only intermittently available, such as a client-server project where there are only a few configured servers available and they must be shared by testing and development. The logistics of such a situation may dictate that we script tests carefully in advance to get the most out of every second of limited test execution time. Exploratory testing is especially useful in complex testing situations, when little is known about the product, or as part of preparing a set of scripted tests. The basic rule is this: exploratory testing is called for any time the next test you should perform is not obvious, or when you want to go beyond the obvious. In my experience, that's most of the time.

## White Paper

### **About QualiTest**

QualiTest Group is a leading global provider of Quality Assurance and Software Testing services. With over 800 Software Testing professionals, 10 branches and a strong presence in the USA, Europe, Israel and Asia, QualiTest provides services to global Fortune 500 companies, large and small organizations. The company is committed to improving quality by providing superior solutions to customers and partners worldwide as well as advancing QA and testing methodologies.

QualiTest offers a wide variety of staffing services such as: onshore testing, quality assurance (QA) and testing outsourcing and flexible staffing. QualiTest expert services promote global practices including: Test Process Improvement (TPI), Keyword Driven Testing (KDT), Scrum Based Testing and Test automation.