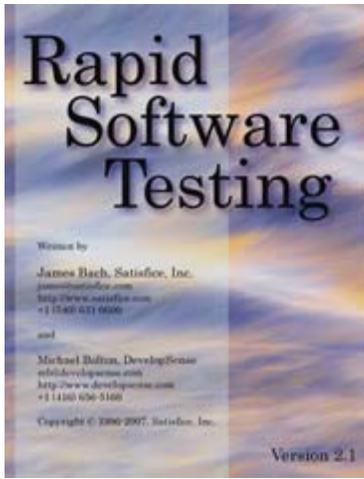


Rapid Software Testing by James Bach and Michael Bolton



Excellent software testing might seem hard. Maybe you're overwhelmed by or uncertain about approaches to test planning, design and execution. Maybe you're working in an environment where some people aren't following "the rules", or in a culture where "the rules" are overwhelming the time and resources that you have to test. It might be that you're having difficulty answering questions like "How long will you need to test?" or "Why didn't you find that bug?" Perhaps you're finding that "industry best practices" are infeasible and a poor fit for your organization, or that books, courses, and certification schemes related to testing focus on nomenclature without giving you the exercises, experiences, and thinking tools to help you learn to *test*.

Do you want to learn how to do excellent testing, effectively and accountably, in situations in which you are faced with barely sufficient information and extreme time pressure? Do you want to get *very good* at software testing? If so, then Rapid Software Testing—a methodology and a course designed by James Bach and Michael Bolton—is for you.

About the Course

Rapid Software Testing is a three-day, hands-on class that teaches testing as a sophisticated thinking art. **Rapid testing is a complete methodology designed for today's testing, in which we're dealing with complex products, constant change, and turbulent schedules.** This class presents an approach to testing that begins with developing personal skills and extends to the ultimate mission of software testing: lighting the way of the project by evaluating the product. The approach is consistent with and a follow-on to many of the concepts and principles introduced in the book [Lessons Learned in Software Testing: a Context-Driven Approach](#) by Kaner, Bach, and Pettichord.

The philosophy presented in this class is not like traditional approaches to testing, which ignore the *thinking* part of testing and instead advocate never-ending paperwork. Products have become too complex for that, time is too short, and testers are too expensive. Rapid testing uses a cyclic approach and heuristic methods to constantly re-optimize testing to fit the needs of your clients. The Rapid approach isn't just testing with a speed or sense of urgency; it's mission-focused testing that eliminates unnecessary work, assures that everything necessary gets done, and constantly asks what testing can do to speed the project as a whole.

One important tool of rapid testing is the discipline of exploratory testing—essentially a testing martial art. Exploratory testing combines test design, test execution, test result interpretation, and learning into a seamless process that finds a lot of problems quickly. If you are an experienced tester, you'll find out how to articulate those intellectual processes of testing that you already practice intuitively. If you're a new tester, hands-on testing exercises help you gain critical experience.

If you outsource development or testing...

We have taught this class all over the world to offshore and outsource organizations on behalf of clients who want their testers to improve their skills and do a better job of testing, without the expense and weak coverage associated with detailed, scripted test procedures. The rapid testing methodology is about getting value for the testing dollar—value that simply can't be reproduced by throwing untrained bodies at the problem—so that your top management won't see testing as a rote activity that some stranger could do as well as you can. Even if you outsource, you will need a core team of testers back at headquarters who can rapidly test products to check the "testing" done by outsource firms.

If you are burdened with clerical requirements...

We have taught this class in organizations pursuing the CMMI, and in organizations subject to FDA and other regulatory requirements. Rapid testing is about *thinking*, so it's not incompatible with formal process cultures. As long as the project owners still want you to think well and find important problems quickly, this is a class that applies to you. However, we do advocate a lean form of test documentation, minimized to the extent you can possibly trim it while still completely satisfying the client and accomplishing the mission. We also teach session-based test management, which allows you to measure, report upon, and document exploratory testing in a way that is entirely accountable and auditable.

About The Instructor

Michael Bolton has over 20 years of experience in the computer industry testing, developing, managing, and writing about software. Michael has delivered workshops, tutorials, and conference presentations on Rapid Software Testing and other aspects of testing methodology on five continents (and one subcontinent!). He has been writing a regular column in Better Software Magazine (formerly Software Testing and Quality Engineering) since 2005. He was an invited participant at the 2003, and 2005-2009 Workshops on Teaching Software Testing in Melbourne, Florida (hosted by Cem Kaner, James Bach, and Scott Barber). He is a founding member (with Fiona Charles) of the annual Toronto Workshops on Software Testing. He is also the Program Chair for TASSQ, the Toronto Association of System and Software Quality.

Course Outline

In this course, you will learn:

- How to focus on the mission and hit the ground running, tackling the testing of any product or product idea instantly
- How to analyze, select, and use concise and effective heuristics, models, and practices for rapid test design
- How to test expertly despite ambiguous, missing, or constantly changing specifications
- How to deal with potentially overwhelming complexity or confusion
- How to decide when to start, suspend, or stop testing—or how to continue while steering in a different direction
- How to prepare and deliver credible test reports at any time

Key Ideas

The Themes of Rapid Testing

Put the tester's mind at the center of testing.
Learn to deal with complexity and ambiguity.
Develop testing skills through practice, not just talk.
Use heuristics to guide and structure your process.
Be a service to the project community, not an obstacle.
Consider cost vs. value in all your testing activity.
Diversify your team and your tactics.
Dynamically manage the focus of your work.
Your context should drive your choices, both of which evolve over time.

Rapid Testing Starts With You

Identifying your context
Testing under time pressure
Excellent rapid technical work begins with you...
...but you don't have to be good at everything
Skills vs. alternatives

Scientific Thinking

How do we know what we know?
Confronting complexity with general systems thinking
Linking observations and inference
Testing magic tricks and sufficiently advanced technology
Introducing heuristics

Finding and Recognizing Bugs

Know your oracles
Consistency as a theme for oracles
Coping with difficult oracle problems
Know your coverage
Handling coverage disputes
Addressing coverage problems with testability
Reporting and reporting coverage

Comments on the Course

"Last week I took Rapid Software Testing from Michael Bolton. The three days of stuffing my brain in the beautiful downtown campus of the University of Toronto was loads of fun...If you don't go to play each day, or you don't think you can break the rules, or you simply want to become a better tester, give Rapid Software Testing a try. I think you will find, as I did, that it is three days well spent!" *Michael Hunter (The Braidy Tester), Microsoft, "This is your Brain on RST"*

"I thoroughly enjoyed your class on Rapid Software Testing. The more time I have to think about the class the more I find value in it. It has really changed my approach and ideologies around testing. The only thing left now is to get everyone in my office on the same page... Before I thought testing was just a stepping stone to move into development but now that I have more of an appreciation for testers, I think I'm going to stick with it and try to get others to realize how important testers are to software development." *James Swanson, Minneapolis*

"We are already starting to put into practice what you taught us—here is a mini case study. Four of the group sat down last Friday and tested another product. Dan (*name changed for confidentiality*) guided and made suggestions. None of the 4 knew the product under test. The product test lead spent half the day being a live oracle. Results: Another 50 defects. Several were crashes (buffer overrun—thanks Perlclip!). Many UI and usability defects. By the afternoon the team was starting to find more specific defects in what the product should do, but wasn't doing. However, by this time they were getting very baked. This sort of testing is really hard work! However, the product lead was amazed by what was found, and the defects found per hour invested was—once again—orders of magnitude more effective than the testing that was currently going on with the product. We are going to cycle this much more frequently, and the same four are going to dig deeper on the same product later this week as well." *A test manager at a commercial utility software company, somewhere in Canada*

Contact Information

Michael Bolton
61 Ashburnham Road
Toronto, ON M6H 2K4
CANADA
+1 (416) 992-8378
<http://www.developsense.com>
mb@developsense.com

James Bach
150 Marina Heights Lane
Eastsound, Orcas Island, WA 98245
USA
+1 (360) 376-4367
<http://www.satisfice.com>
james@satisfice.com

Exploratory Strategies and Procedures

Focusing and defocusing
Exploiting variation
Telling the testing story
The process of test design
Testing to search vs. testing to learn
Blending scripted and exploratory approaches

Diversifying Your Strategy and Your Tactics

Considering cost vs. value
Quick tests vs. careful tests
Focusing on risk
The role of repetition

Rapid Recording and Reporting

Session based test management for high accountability
Rapid estimation
Reporting progress in a test cycle

Selected Exercises (throughout the class)

Test The Famous Triangle
Find the Bug in a Calendar Program
Test the Mysterious Sphere
The Wason Selection Task
The Simplest Possible User Interface
Testing a Wine Glass
A Critique of a Test Report
What Does It Mean To Save a File?
Use Exploratory Modeling on a Small App
Breaking Your Confirmation Bias
Discover the Role of Repetition in Test Strategy
Report on the Completeness of Testing
Exploratory Testing with Playing Cards
Exploratory Testing with Dice

In a management position? Ask about Rapid Software Testing for Managers, a one-day course focused on managing a diversified, effective, and inspired test team!