FRITS J. BOS, PMP

1602 - 145 Hillcrest Ave., Mississauga, ON, L5B 3Z1 | C: (416) 524-7790 | frits@writeforhire.com

Quality Assurance is the Achilles Heel of software development, there are more ways for logic to be coded in error than to get it right. Testers have a major challenge trying to anticipate all the ways in which data can affect the application under test. Many test automation "solutions" are nothing but a "capture and replay" process that is often of limited value for keeping test cases current. In this seminar we explain a focused testing approach that covers all the bases, and that runs on a PC with MS-Office installed (using the powers of Excel™-VBA to deliver great results).

Quality Assurance in IT

Do you expect more from an astronomical investment in software test-tools? Do you find the cost/benefits ratio disappointing? You are not alone, but there are alternatives! You may not know that the software you need for testing is likely already installed on each PC.

The most influential business software product is **Excel™**, but most people use only a fraction of its capabilities. It is remarkable, since most testers already use Excel™ to build laborious scripts. The problem is that these scripts are static – as time passes and/or transactions are entered the conditions in the test data change, invalidating the scripts, and requiring updates. Even costly test management systems that import Excel™ worksheets miss this boat on critical test features.

We added the intelligence to Excel™ to fix that problem!

We offer a training seminar that builds on what your testers already know: how to create scripts that can be processed by our Excel™-VBA software to deliver testing prompts or audible feedback to facilitate testing without going back-and-forth between the script and the application screens. We offer a complete training program to empower your QA department with test automation in 9 sessions, and as a bonus each participant gets a copy of both **GATES** and **TILT** software for their personal use, complete with detailed user-guide documentation.

Session 1: Creating manual scripts

We begin by exploring how business requirements and functional requirements are itemized as input to testing, and how to describe the testing steps in sufficient detail. This is based on our **GATES** application (**Generate Automated Test Scripts**) that puts the potential of automation in the hands of every tester with access to Excel™ (on a PC workstation or laptop). **Secret #1**: the tester can be anywhere, so long as they can access the application to be tested. There is no need for a centralized server or special security (beyond what you use for your regular users). Best of all this provides the closest, and most realistic, simulation of a production user base as possible. **Secret #2**: you control what information testers have access to, an increasingly important data security consideration. **Secret #3**: you can use **GATES** as an automated tester that works after hours to extend your day, and to optimize the time available for software development.

Session 2: Creating dynamic manual scripts

The next stage is focused on creating "variables" within the script that keep it current. We will show later how the data are updated, but to use that data we need to explore "variables" within the script to receive that data and reflect the updated information in the tester prompts. This can be a traditional script, but this also works with pop-up window prompts, or with audible test instructions, well before you are ready to consider test automation.

Session 3: Using the power of Excel™ cell formulas

The third stage explains how "variables" within the script can be dynamically manipulated using Excel™ formulas. This introduction shows how powerful Excel™ is, and how you can use Excel™ formulas to control aspects of the test execution. For example, you can calculate expected test results that the tester can confirm against the application responses.

Session 4: Introducing automated testing with Excel™

There are many fancy tools on the market that exploit different paradigms for test automation, the most painful of which is "capture-and-replay", that suffers from the same problems as static scripts. **GATES** has an enhanced version of the VBA "SendKeys" method, including mouse control and screen-image reading to interact with the application under test. **GATES** "Test Automation" uses simple commands provided in a parallel worksheet column to match the manual script that then explains what the automation commands are up to. You can also use this tool to validate the application responses to the test input. Note that we use a flexible screen-coordinates tool to adapt **GATES** to changes in the GUI layout with a minimal amount of effort.

Session 5: Introducing data-driven testing with Excel™

With scripts that are already designed to contain "variables" that keep the data current the next step is to use Excel™ as a database engine that feeds data into these variables. This gets a little more complicated: for every application dialog instance we only have one script instance. Using simple selection commands, we process rows of data in Excel™ that then trigger portions of the script depending on the data contents. While we normally receive that data from **TILT** we have the option to build scripts in data format, which dramatically reduces the script writing efforts.

Session 6: Building a test-data library with Excel™

We use a second component to the testing called **TILT** (**Timed Input & Logic Testing**) that, among other things, builds the input data for **GATES**, but it is not meant to be distributed: it is designed for your test manager/analyst who builds, maintains, and protects the "golden database" which is a combination of all valid and invalid data the application must process or reject. There are 3 stages to the **TILT** process: building the test-data library is a prerequisite step based on business requirements and functional requirements. In a project time-line, you can start building the test-data early in the process to validate that all requirements are accounted for, including any master files that you build so you do not violate privacy laws by using production data extracts. **TILT** has provisions to redact or scramble data so that it no longer reflects identifiable customer data, and you can use boundary testing and edit checking by introducing "invalid" test conditions. For each application you will normally employ a separate "golden database" **TILT** workbook.

Session 7: Generating the test collateral with Excel™

Not only do we have ability to generate input data for **GATES**, but we also can generate data for the application to work on. We can generate SQL upload files, XML files, text files, even cheques and other documents you might require for your simulation. Since the file generation is template driven we can even deliver a wide variety of proprietary formats (such as OPEX OXI files). As **TILT** suggests, all data are current, and can also be throttled by using a timer feature in **GATES** that will invoke the next transaction based on the PC clock. **TILT** can create different arrival patterns and allocate specific transactions to a specific **GATES** instance: this way inputs can be fired off from any location to simulate peak processing times, for example.

Session 8: Orchestrating the test execution with Excel™

TILT can trigger batch processes to copy files or initialize databases, and so on, to make sure the data contents are current. It can also wait for test completion, to perform a comparison between the current test session and a previous test session for regression testing. This makes it ideal for daily after-hours verification that software changes have not broken any functionality, or (if any errors are found) developers can fix those errors before additional code changes are made.

Session 9: Review and summary of how to manage testing

TILT will normally be used in the domain of the test analyst in charge of the application, while all testers will have access to **GATES** for test execution. It is easy to build the test collateral with the **TILT** tools that can also initialize an updated **GATES** worksheet that we can distribute to testers. **GATES** can produce an output worksheet with the actual values used in the scripts that provides a feedback instance to audit individual test sessions, which can be returned and uploaded to a test management repository for reference just as if an old-fashioned detailed script was used.

Low cost introductory offer

For this ***introductory offer*** we charge $1,500.00 per day (for out-of-town venues add $400.00 for travel time, plus reimbursement of transportation and lodging expenses at cost). This seminar is intended for up to 10 participants. Additional attendees can be accommodated at $100 each for extra materials. The client is also responsible for providing a meeting room with an LCD projector.

Schedule: (+ break)	**2 days** - $3,000.00	**3 days** - $4,500.00	**4 days** - $6,000.00
Day 1	Sessions 1,2,3 + 4,5	Sessions 1,2 + 3	Sessions 1,2 + 3
Day 2	Sessions 6,7 + 8,9	Sessions 4,5 + 6	Sessions 4 + 5
Day 3	N/A	Sessions 7,8 + 9	Sessions 6 + 7
Day 4	N/A	N/A	Sessions 8 + 9
Features:	Limited hands-on	Hands on exercises	Hands on exercises

Out-of-town only (venues outside the greater Toronto, On., Canada area) budget:

Travel:	$400.00	$400.00	$400.00
Transportation:	1 Economy return *	1 Economy return *	1 Economy return *
Lodging:	2 Nights Economy	3 Nights Economy	4 Nights Economy
Cab fares:	As required	As required	As required

* We will use the most economical air-fare or other public transportation practical to reach the destination within one day.

Respectfully,

Frits J. Bos, PMP