May 2006 $9.95 www.StickyMinds.com

# BETTER SOFTWARE

The Print Companion to StickyMinds.com



## WHAT LIES BENEATH

*Hunt Down Security Vulnerabilities with Penetration Testing*

# Time for New Test Ideas

by Michael Bolton

• On a recent project, after a long weekend, our production system began crashing. The database was running out of connections. From time to time, all testers get the sinking feeling of having missed something; this time my worry was that we had diligently examined the month-end reports, but we hadn't simulated a bunch of people running them at once. As it turned out, the problem had nothing to do with month-end reports or volume, but we immediately changed our strategy such that some tests were more representative of real operational load profiles.

• A task for one of our iterations was to develop and test an automated task to remove a transaction glitch. The application helped to broker transactions between customers. In one case a customer had died; consequently the transaction was never completed, and money hung between accounts because the deceased could neither receive nor reject the transaction.

• When I'm traveling, I like to keep my personal information managers on my laptop and handheld computer in sync. However, if I change the time zone on both devices and then synchronize, the software fails to account for the time zone for which the appointment was made. The result is a large pile of duplicate entries—one reflecting the original appointment, and one reflecting the appointment as it would appear in the new time zone. Restore both devices to the original time zone and the duplicates aren't deleted or recognized.

• My stepson called me into the living room where he was playing a popular hockey simulation, saying, "I *thought* I saw that happen . . . check this out!" He replayed an incident in which one of his players had been checked by an opponent and then careened into the glass above the boards. The glass had shattered. In slow-motion replay, he was able to show me from several angles that the glass had begun to fall apart long before the player had hit it.

• James Bach posed a puzzle to me a while ago. "Supposing someone enters two plus two in a calculator program and the answer is four, how might that not be satisfactory?" I made a few suggestions. He then replied, "There's an obvious one that you missed: The answer took thirty seconds." A few nights later, I posed the same question to my friend Wil. He didn't guess the delay problem, but when I told him James's answer, Wil pointed out that in his business—networking software for telcos—there were times when getting the answer back in less than thirty seconds was a sure symptom of trouble. It is important for testers to consider opposites; getting an answer too soon can be as bad as getting one too late.

What did each of these incidents, presented on different platforms and in different contexts, have in common? One thing, for sure, is time.

When we think about computer software, we tend to think about quick processing and nearly instantaneous response. We tend to think less about the other relationships that our software might have with time.

Rapid Testing encourages us to develop guideword heuristics, words that prompt us to think indirectly and expansively to help solve problems. I decided to collect time-related words that might be valuable to a test strategy. I found it easy to get to one hundred, and colleagues supplied many more. I've provided a partial list below, divided into a few arbitrary categories. As an exercise, pick some guidewords at random. Then think about how each guideword applies to your current product or process. Try to find at least three ideas about tests or test strategy. If the word doesn't seem to fit, try thinking about it more abstractly, concretely, or broadly until you get those three ideas.

For example, when I think about a holiday guideword like "New Year's Day," I think about certain reporting periods closing (year-end reports, but also month-end reports—are they consistent with each other?); the fact that few developers will be at work (What are my alternate sources of information?); that banks are closed but interest still accumulates (Have we remembered that business days and calendar days are necessarily different?); and that there will typically be an extra day attached to the weekend (Things that usually happen on Monday might happen on Tuesday, but with higher than usual volume).

## Time-Related Guidewords

*Data-flow* guidewords relate to the pacing and marshalling of data. They include blocking, buffering, bursts, concurrency, queuing, dequeuing, spooling, and timeouts.

*Financial and business* guidewords are closely related to functionality for many applications. This large class includes annual percentage rate, business day, business week, calendar day, close of business, compound interest, due date, end of quarter, fiscal year, interest, month end, opening day, payment period, quarter, sell-by date, service level agreements, simple interest, start of business, tax year, warranty period, and year end.

*Holiday* guidewords cross political, cultural, and social boundaries and tend to trigger exceptional business rules. A handful of holiday terms immediately relevant to North America are civic, religious, and statutory holidays, Christmas, holiday pay, spring break, New Year's Eve and Day, vacation, and weekend. If some of your team is offshore, what might holidays in other countries do to your development schedule?

*Instrument* guidewords refer to the means by which we can observe and measure time: atomic clock, alarm clock, calendar, chronograph, chronometer, clock, egg timer, fuse, hourglass, metronome, microwave, speedometer, stopwatch, sundial, tachometer, wristwatch, and VCR (that flashing 12:00 triggers all kinds of ideas about user interfaces and configuration issues).

*Lifetime* guidewords relate to time on a human scale: adult, age, aging, archaic, birth, child, death, inheritance, legacy, minor, senior, retired. However, several of these ideas also can apply to applications, data, and records.

*Operating systems* have a whole vocabulary of time-related guidewords: concurrency, critical region, idle time, mutex, preemption, refresh rate, round-robin, semaphore, synchronization, threads, time sharing, time slicing, and timer ticks. Many of these could apply to other contexts, such as business rules or test scheduling. If you're unfamiliar with some of these terms, look them up on Wikipedia.org.

*Rates* refer to some count or quantity divided by some unit of time. Examples are acceleration, deceleration, degradation, frequency, speed, hits per second, transactions per hour, bits per second, and velocity.

*Relationships* place something in time with respect to something else: after, before, current, during, early, faster, fastest, frequency, future, interruptions, late, past, present, slower, slowest, slowing down, speeding up.

*Schedules* include alarms, alerts, appointments, deadlines, and slack time.

*Time zones* resonate in daylight saving time (DST), GMT, local time, standard time, time zone exceptions (like Saskatchewan and Arizona that don't observe DST, and India and Newfoundland with time zones offset by one half-hour), and UTC.

*Units of measure* scale from picoseconds to nanoseconds, microseconds, milliseconds, seconds, minutes, hours, days, weeks, months, quarters, seasons, years, decades, centuries, millennia, epochs, hertz, and baud.

In many cases, we'll have covered most time-related risks in other coverage models. Time's arrow files rapidly in one direction—toward the next ship date—but guidewords about time can help us broaden our test coverage quickly and inexpensively. **{end}**



*Michael Bolton lives in Toronto and teaches heuristics and exploratory testing in Canada, the United States, and other countries as part of James Bach's Rapid Software Testing course. He is also program chair for the Toronto Association of System and Software Quality. He is a regular contributor to* Better Software *magazine. Contact Michael at mb@developsense.com.*

STAR EAST SPEAKER

**Don't Stop Now!**

Log on to **StickyMinds.com** and join Michael Bolton and your peers in a conversation about this topic. At the end of the digital column, add your views or just read what others have to say.