Testing in O&M
Software Does Not Wear Out, It Degrades

Jon Hagar, Grand Software Testing

Abstract. Software verification and validation remain active during software-system evolution in operations and management. There are actions in test planning and risk analysis that need to take place. This article briefly defines verification, validation and testing. It also outlines testing activities during operations and management, including integrity and risk assessment, regression testing, and handling third-party changes introduced into the operations and management phase. This article concludes with the impact of test automation.

Introduction

Most long-term software development teams understand that a majority of the costs spent on typical products happens during the operations and maintenance (O&M) phase, which can last for years. Additionally, most organizations understand that verification and validation (V&V) along with testing are necessary activities for software products. Unfortunately, V&V and testing may be seen as less important in the O&M phase since the software is "in the field and working."

However, V&V and testing can provide vital information about any software degradation during the O&M phase. In fact, some projects have more V&V and testing efforts in O&M than in development because of software "evolution."

The evolution of software in the O&M phase that can necessitate V&V and testing includes:

–Updates to operational concepts, which subject software to new environmental uses that may trigger long-hidden errors.

–Changes to software to add new features, which can have side effects that impact existing logic due to coupling.

–Changes to the software to fix bugs, which in turn will need testing.

–Updates to software elements from third-party vendors, which also require testing.

–Changes to hardware that impact the software.

Years ago, a spacecraft had been in orbit. During its O&M activities it needed a small software update, so a patch was uploaded that consisted of only a couple words. The change was reviewed (using peer review verification), but not subjected to real validation or testing.

After the upload, the spacecraft went dark (stopped communicating) for a few minutes until the system's safing software activated and re-established communications. Thereafter, the team and the whole organization was instructed that no change to either software or hardware in the O&M phase should ever be fielded without being validated, verified or tested first.

Unlike hardware in the O&M phase, which with repeated use can "wear out," software does not wear out. However, some software experts find that as software evolves, the possible negative effect on the software as a whole can be very similar to the effects of hardware wearing out. The code becomes hard to maintain. It may break and, therefore, be unstable. A set of solutions to minimize the wear-out factor can be provided by V&V or testing. Some of these solutions are examined briefly below.

Defining V&V and Testing

V&V and testing provide information to stakeholders. Verification answers the question "Did we build the product right?" Validation answers the question "Did we build the right product?"

When you verify software, you evaluate whether it satisfies the conditions defined at the beginning of an engineering activity (e.g., checking requirements implemented in a design). When you validate software, you evaluate a product to determine whether it satisfies the user or customer needs.

Classically, V&V uses four approaches: test, inspection (e.g., visual examination), demonstration (e.g., executing tests on a delivered system), and analysis (e.g., modeling) to gather information. V&V and testing provide information on these qualities:

–Meeting all requirements.

–Creating trusted software with no errors.

–Meeting timing expectations.

–Verifying that the user interface works.

V&V and testing include a variety of activities throughout the life cycle to provide information on these qualities and others.

There are generic industry standards on V&V and testing, including IEEE 1012 V&V plans [1] and ISO 29119 software testing [2]. Such standards can be used as a starting point, but they must be tailored for any project, even during O&M. There are other useful industry references to consider, including "Black Box Testing: Techniques for Functional Testing of Software and Systems" [3], "The Art of Software Testing" [4], "Lessons Learned in Software Testing," [5] and "Systematic Software Testing" [6]. Refer to these standards and references for more information.

V&V and Testing Activities within O&M

While V&V and testing should occur during initial development, many of the qualities assessed remain of interest during the O&M phase. V&V and testing efforts should be reduced and tailored for O&M, but dropping them to zero effort is not advised. A conceptual O&M V&V and testing process flow is shown in Figure 1.

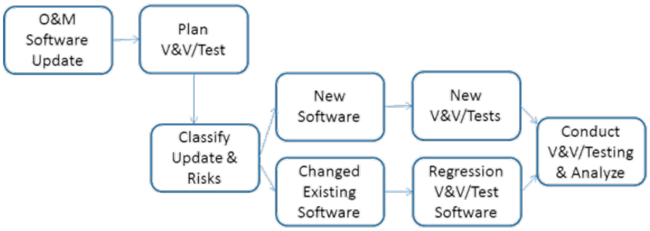

O&M Software Update
Plan V&V/Test
Classify Update & Risks
New Software
Changed Existing Software
New V&V/Tests
Regression V&V/Test Software
Conduct V&V/Testing & Analyze

Figure 1: Basic V&V and Testing Flow for Software O&M Evolution

Teams within the O&M cycle should conduct V&V and test planning and life cycle activities as the software evolves for all changes. There is no one best set of activities or tests for O&M, so critically thinking testers are still needed. Planning, classifying the update, and identifying changes in risks for testing are all necessary for an O&M update.

Further, not all changes are equal. Even small changes can have large impacts, as the spacecraft story illustrated. New team members may not understand historic software designs and features, so errors can be introduced even during a small change.

Many programmers also suffer from the "not invented here" syndrome, so they will change software to make it better even when the change is not necessary. These changes usually happen outside the configuration control system unless good configuration management controls are in practice. Such added refactoring changes can be dangerous unless properly controlled, since their impact may not be known or fully understood. This results in incomplete V&V and testing. As Figure 1 shows, risk analysis and V&V are necessary for changed and new software. The following are a few generic recommended test activities (further explained in "Software Test Attacks to Break Mobile and Embedded Devices" [7]).

Integrity and Risk Assessment V&V and Test Planning Activity for O&M

IEEE 1012 [1] defines integrity levels and addresses how different levels impact V&V efforts. If integrity levels were used during initial development, they may evolve and change during the O&M phase. If integrity levels were not used, they can be introduced during the O&M phase. Refer to IEEE 1012 if you are interested in using integrity levels to manage O&M-phase software activities.

Next, ISO 29119 [2] is a risk-based testing standard made up five parts, which can be used during O&M. The ISO 29119 set of standards defines process activities, documentation and techniques. New risk assessments may be needed during the O&M phase. Information about software qualities can change, introducing or reducing risk. If risk-based testing was not used, the O&M phase can be a good time to introduce it, allowing V&V and testing to focus on the software's higher-risk qualities and areas. In risk-based testing, not all areas and qualities of software will need the same levels of activities. Risk-based testing can be used to manage O&M V&V and testing costs and scheduling.

Updated risk areas that may change during O&M include:

–Security and privacy threats. Not many years ago, concerns about privacy were minimal for many software O&M efforts. Almost daily now, we hear about software systems being hacked, which costs millions of dollars and can destroy a company's integrity. Thus, software that was once viewed as safe from a security perspective can become a high risk.

–Safety impacts to humans from new use of software systems. Fitness watches were designed to help with health and exercise, but they are now being used to make medical decisions [8]. Thus, fitness watches have become more of a safety risk than developers might have originally planned.

–Hazards introduced by system changes. A programmable logic control (PLC) for power systems was installed in an isolated network, but then a router was added to the network to allow communications. Now the PLC is threatened by hacking from a virus, such as STUXNET [9]. Electric utilities are now at risk due to just such a scenario.

–Performance needs growth. Initially, web systems were focused on functions and design. Now most websites worry about crashing under high volumes as they become popular. This changes the V&V and testing focus from functional testing to performance analysis and V&V.

–Environment and hardware changes. An existing vendor provided a new computer system and supporting software to a project, but the vendor's team did not understand the new environment the software or system was to be used in. Project management thought their software was proven because the vendors were an existing third party. That is, until failures started happening during V&V and testing.

These examples are risk areas that went from "low risk" during initial development to "high risk" in O&M phases.

In integrity and risk assessment during O&M, ask what is changing. Changes can be in the software, hardware, users, threats and environment. Be cautious during O&M V&V and testing activities.

Regression Testing Activities Important for O&M

Most software V&V teams are familiar with regression testing [10]. This testing helps you discover if you broke something that was already working during development — which happens as the software changes in areas that were already tested. Regression testing happens in software partially because of coupling [11] or lack of team understanding of design.

Teams develop different strategies for regression testing, including:

–Rerunning all testing.
–Rerunning tests in the code areas related by function.
–Rerunning a standard regression test suite.
–Rerunning what the team thinks needs to be run.

Each of these strategies has positive and negative considerations [7]. I will not define a complete regression test strategy and approach, but there is debate in the test community about the validity of regression testing [12].

The validity question comes from the pesticide paradox [13]. In the pesticide paradox, the ability of a test case to provide information diminishes the more times it is run without changes. Running regression tests over and over is therefore of questionable benefit, but most organizations employ some level of regression testing and analysis. The debate about regression tests will go on.

I ran a test with a mix of regression items 2, 3 and 4 on a personal progress and did risk-based testing planning. I changed "old" tests to some extent to avoid aspects of the pesticide paradox, while still addressing regression. I made sure a test existed that could assess the error or changes in the software. During O&M phases, I did a lot more testing because I employed V&V and test automation (see automation section below).

During one O&M cycle, I had the following percentage allocations:

–Standard regression tests: 20 percent.
–Expert team-selected regression tests: 30 percent.
–New tests (including developer tests): 30 percent.
–Tests traced to coupling that needed to be executed: 10%.
–Historic system tests not run over a long period of time: 10%.

These percentages are nothing magical. In fact, on other O&M cycles the percentages looked different. The key was that I applied critical thinking and planned tests early in the O&M cycle. The stakeholders accepted the V&V and test plans, and I modified test plans as needed during the cycle.

You may notice the last category percentage seems new. The team added this, and it was different than standard regression testing concepts. I learned over years of O&M that it was a good idea to pull out historic tests that had not run for a while and then rerun them with changes for the new O&M cycle. I did this during most O&M cycles, and slowly over the years (it was a long-term project), the team would cycle through the majority of historic system tests.

We were looking for a regression test that had escaped previous test cycles. In at least one case, we found an unexpected regression error using this method. If I'd had the time and high levels of automation, I could have avoided this last step. However, I did not have the extra time and budget, so cycling over the composite test suite was a compromise.

Introducing New V&V and Testing in O&M

In the O&M phase, many projects will add new functionality to the existing software. This in effect resets the testing efforts back to where they were during development, so new tests are needed. This makes sense to teams and managers. However, there are a variety of overlooked sources where new tests may become necessary during O&M, including:

–New operational concepts for the software system, where the existing tests do not address the use cases.

–Changed usage environments, where software may encounter external situations not tested during development.

–Updates to hardware, where the new hardware behaves differently than what was originally tested.

–Changes to incoming or output data, where data boundaries and classes have not been tested.

In new testing, different test plans, risks, designs and implementations may become necessary. Risk-based testing should be redone, and the team may want to consider new exploratory tests [7] in addition to the planned tests.

The Need in O&M V&V and Testing to Address Third-Party Software Changes

An interesting aspect of O&M V&V and testing is third-party software and hardware. These components can be commercial-off-the-shelf (COTS) or customized software and hardware. It is common to have such third-party elements change and be updated during the O&M phase. While it may be tempting to think that the third-party vendor owns the V&V and testing of such elements, this can be dangerous for the product.

On one project, I had some vendor hardware components. One of them wore out and was replaced. The configuration management numbers were the same, but in our test lab, I started having test failures during our regression tests. At first, I thought it was a software issue. After much analysis V&V, the team found the vendor had changed a transistor in the hardware!

They should have rolled the configuration part number, but they did not because the part was considered to be equivalent (which it was not). If I had not been running regression tests in the lab with all the updated elements, including the third-party elements, I might have fielded a bad release during the O&M cycle.

In third-party testing, consider running new and regression tests. Having a good configuration control system for hardware and software is a good start; however, the team must still "trust but verify." The development project owns the software system, which includes all of the third-party elements.

V&V and Test Automation

Much of the IT industry loves the idea of V&V and test automation, as if it is a panacea that will solve many problems (e.g., schedule, budget, testing completion and regression). Automation has a mixed track record. Sometimes it works, saves time or money, and is a good idea. At other times, automation efforts have failed, because it takes more initial setup time (learning curves) and money (tooling and consulting fees) up front during development to make automation work. Often the payback is only realized after several test and regression cycles.

My experience is that V&V and test automation's real payback can occur during O&M. If I must do new and regression tests manually, my labor costs remain high during each O&M cycle. If I have automation, rerunning regression suites and modified new tests can be very cost effective. I can spend the savings on new and exploratory tests. There will be some O&M costs to maintain automation tests and tooling, but these can be reasonably managed.

The automation payback gets better as a team moves into the cloud, where they can add test execution resources quickly. Automation efforts will still include the costs of maintaining test automation, running new tests and thinking critically, but automation during O&M efforts becomes quite attractive.

Of course, automation must be planned, and it should be started during initial development. My experience is that it takes three to five O&M regression V&V and testing cycles to start to see a payback on automation. However, if the team factors in the costs of automation and the savings during O&M and development, automation becomes a viable path.

Summary

I have outlined V&V and testing impacts to O&M with a focus on software, but you can never forget the hardware and system in test planning. Many of my students wanted to get jobs doing new development because they thought it was more fun. However, I spent most of my career doing O&M V&V and testing. Most of us will spend much more time working in O&M than in new development. O&M comes with its own V&V and testing challenges, some of which I have outlined in this paper.

For more information, see the references section below. O&M V&V and testing is a big subject, and a team must address bugs, regression, new testing, third-party vendors and automation during O&M. V&V and testing assessment includes checking for software degradation during evolution and looking for errors that may still exist in software even after years of use. You should now have a starting point for O&M V&V and testing budgeting, scheduling and detailed planning.

REFERENCES

1. 1012 – IEEE Standard 1012–2012. “IEEE Standard for Software Verification and Validation™”
2. 29119 – IEEE Standard on Software Testing – Five parts covers all of software testing (published September 2013)
3. Beizer, Boris. (1995.) “Black Box Testing: Techniques for Functional Testing of Software and Systems.” John Wiley & Sons Inc., USA.
4. Myers, Glenford. (1979.) “The Art of Software Testing.” Wiley.
5. Kaner, C.; Bach, J. & Pettichord, B. (2002.) “Lessons Learned in Software Testing.” John Wiley & Sons.
6. Craig, R. & Jaskiel, S. (2002.) “Systematic Software Testing.” Artech House Publishers.
7. Hagar. (2013.) “Software Test Attacks To Break Mobile and Embedded Devices.” CRC press.
8. http://www.mirror.co.uk/news/uk-news/teens-life-saved-fitbit-after-7154991
9. https://en.wikipedia.org/wiki/Stuxnet
10. https://en.wikipedia.org/wiki/Regression_testing
11. https://en.wikipedia.org/wiki/Coupling_(computer_programming)
12. http://kaner.com/pdfs/gui_regression_automation.pdf
13. https://en.wikipedia.org/wiki/Paradox_of_the_pesticides

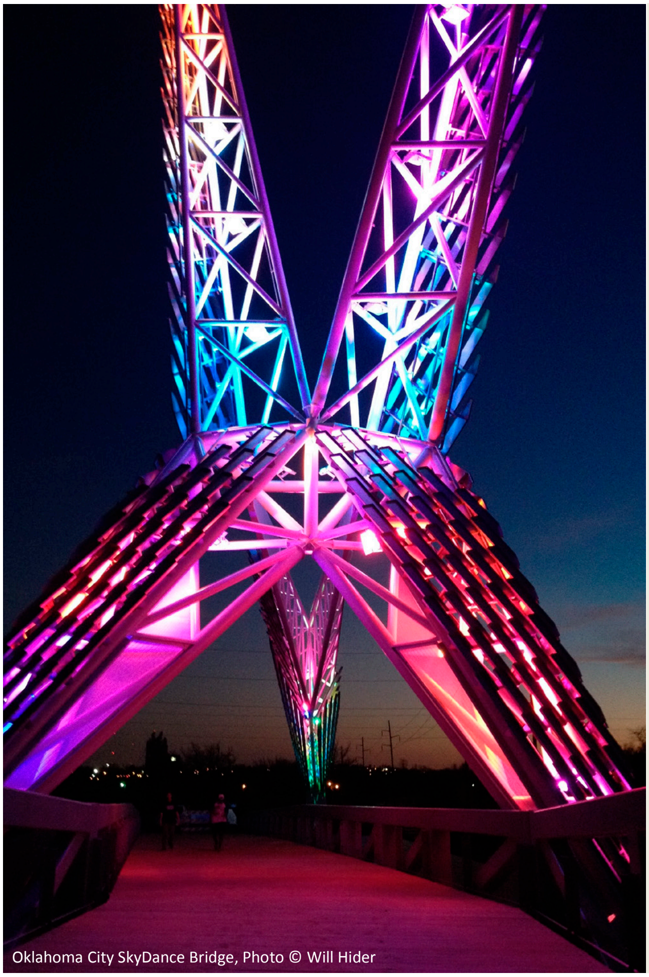
ABOUT THE AUTHOR

Jon Hagar is a systems-software tester consultant at Grand Software Testing. He supports software product integrity, verification and validation, specializing in embedded, IoT and mobile software systems.

Jon has worked in software engineering, particularly testing, for four decades. Embedded projects he has supported include control system (avionics and auto), spacecraft, mobile-smart devices and ground systems (IT). He also worked on attack-based testing in the mobile/IoT space.

Jon wrote “Software Test Attacks to Break Mobile and Embedded Devices” and has published an IoT testing e-book and more than 50 papers. He is the lead editor and author on international standards (ISO 29119), co-chair of the OMG UML test profile standard, and a contributor to IEEE 1012 V&V plans.

Jon.d.hagar@gmail.com



Oklahoma City SkyDance Bridge, Photo © Will Hider

WE ARE HIRING

ELECTRICAL ENGINEERS AND COMPUTER SCIENTISTS

As the largest engineering organization on Tinker Air Force Base, the 76th Software Maintenance Group provides software, hardware, and engineering support solutions on a variety of Air Force platforms and weapon systems. Join our growing team of engineers and scientists!

BENEFITS INCLUDE:

- Job security
- Potential for career growth
- Paid leave including federal holidays
- Competitive health care plans
- Matching retirement fund (401K)
- Life insurance plans
- Tuition assistance
- Paid time for fitness activities

Tinker AFB is only 15 minutes away from downtown OKC, home of the OKC Thunder, and a wide array of dining, shopping, historical, and cultural attractions.

76th SOFTWARE MAINTENANCE GROUP

Send resumes to:
76SMXG.Tinker.Careers@us.af.mil

US citizenship required