Developmental Automated Testing and Software Technical Risk Assessments

Brad Neal, SimVentions

Abstract. Testing continues to represent the single largest cost associated with the development of sophisticated, software intensive, military systems. A 20% reduction in overall testing cost, as requested by the Chief of Naval Operations (CNO) in his "Guidance for 2004," will save billions of dollars over the lifecycle of the myriad of tactical software systems. Such significant savings can only be achieved if the concept of testing begins very early in the development process.

Automated Testing

As tactical software development moves more toward open standards and becomes more focused on reducing development time and creating re-usable capabilities, the need for efficient and thorough testing of software becomes more critical than ever before. Testing artifacts can no longer be used only during initial development and then discarded. Coordinated incremental software builds and highly re-usable software testing capabilities will allow developers to coordinate and reuse software tests and testing tools repeatedly. Automated software testing leverages tools to stimulate the software under test, measure its response, and assess the correctness of its response without significant human intervention.

Automated Testing (AT) tools and systems will be based on affordable COTS hardware and software tools that allow more flexible and repeatable use. These capabilities are directly enabled, and necessitated by the proliferation of Open Architecture software and computing environments and standards. This is in contrast to traditional testing methodologies and systems that require specialized hardware to test new software systems and component interfaces. AT will investigate and enable automated, affordable, rapid, high quality, and reusable testing of large software-based systems throughout the entire lifecycle.

Determining the benefit of an AT program requires the identification of the business case for investing in automated test. Simply stated, the business case for AT stems from The CNO Guidance for 2004 [1], which included direction to the Commander, Operational Test and Evaluation Force (COMOPTEVFOR) to lead a collaborative effort among Navy, OSD, and contractors to reduce the costs of Test & Evaluation (T&E) by 20%. In developing a response to the CNO Guidance for 2004, COMOPTEVFOR surveyed programs and included the following as T&E cost drivers:

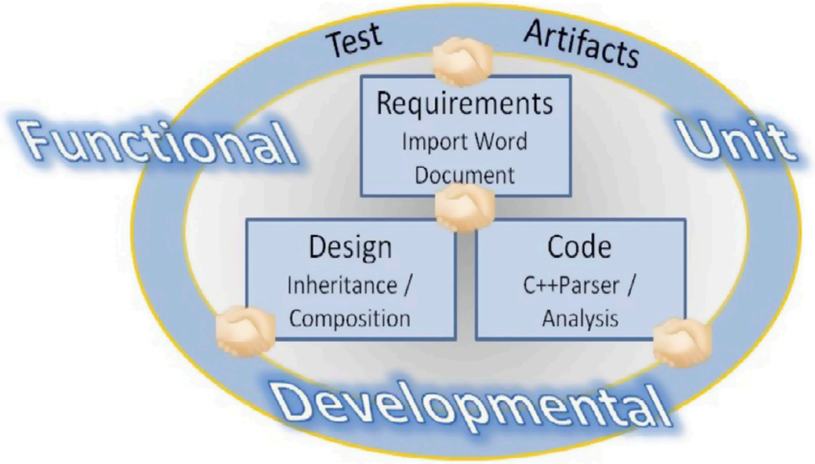
- Redundant testing
- Increased levels of regression testing driven by technology insertion
- Increasing complexity of computer software testing
- Interoperability testing and certification
- Development of unique /duplicate facilities and test beds

This helps to identify the need, the measures necessary to detect progress, the impact to the program under test and the potential ROI for the program. The technical process identifies the steps that are followed that will prove or disprove the business case. The enabling technology involves identifying the tools, targets, and standards for carrying out the technical process. Finally, implementation involves physical execution of the enabling technology, a final determination of whether or not AT is feasible, beneficial, and validates the anticipated ROI, and where AT could be implemented.

Yet, currently there are no policies governing the use of AT tools and techniques within the Acquisition Framework; the DoD 5000 series instructions do not include any reference to AT. However, with the emergence of common product lines, a relationship between the acquisition community and AT concepts needs to be firmly established before a marriage into contractual language is produced. The Navy is going into unchartered territories and studies are needed, pilots performed, and proofs of concepts explored to discover applicability and affordability.

In an effort to take an expanded approach to AT, leveraging the knowledge and insight gained from the Object Management Group (OMG) standards work, an attempt at injecting automation in other areas not greatly explored within industry or Navy programs was attempted, i.e.: 1) Requirements, 2) Design, 3) Code, and 4) Test Coverage. The ability to automatically generate the Requirements Traceability Matrix from top-level requirements and intermediate designs to the actual Computer Program (CP) would eliminate the need to perform this task manually as is often done within disparate programs; DOORS, Excel, Enterprise Architect, and Integrated Development Environments. Standards organizations like the OMG will be leveraged to engage industry, academia, and government communities by developing options and approaches via Request for Information, Request for Proposal (RFP), technology pilots, and white papers. The right mix of companies, organizations, and Navy industry relationships will enable an open and streamlined process for technical standards establishment. If standards are adopted, usability is likely to increase bolstering the business case for AT.

Figure 1: Information Exchange Handshake



Software Assessments

An initial implementation, performed in FY '09, involved the demonstration of a graphical view of requirements mapped to source code. This effort resulted in the generation of a Developmental Automated Risk Testing (DART) tool, which automates the process of mapping top-level requirements to lower level requirements, software design and CP code allowing the user to traverse bi-directionally to examine all the software artifacts associated with each requirement. DART offers automated information exchange handshakes (Figure 1) coupling functional, unit, and operational testing with developmental information. If the requirements change, this tool will allow the developer to instantly see what portion of the CP is affected along with test data specifics. It will allow a tester / analyst / developer to trace an error encountered in a component under test to its specifications and verify its function; determine if the error is in the CP or in the specification; and if so, whether the parent requirements for these specifications also need to be evaluated.

Built-in automated metrics such as code complexity, afferent/efferent instability, along with code fluidity evaluations aid the developer in identifying potential problem areas early. An overall risk assessment offers probability and impact measures based on issues with the code. This analysis takes into account complexity, orphans, and density of complex code among other things. A technical risk chart provides a quick look into the software for stakeholders, providing a status that updates as bugs are fixed and the design is analyzed.

Linking requirements to code provides continuity between the beginning and the end of development and offers the ability to automate the testing of your software design. It offers high-level automated software debugging and promotes standards in requirements documentation and software development. Below are some example use cases subject to analysis for a sample program under test:

- DART has a high probability of making code/design correct by introducing a Level 0 "smoke test" which focuses on coverage rather than on functionality.
- DART facilitates the developers job of having to open up a requirements document, design, or code; all are shown in a dashboard with the automated metrics decreasing time spent using disparate support programs to view the data.
- DART's automated complexity metric determines if code is testable; high complexity potentially introduces risk into program testing and threatens quality.
- DART's automated code fluidity metric is a functional software flow check determining if pieces of code are either called or are calling another piece of code; if not, the developer needs to verify necessity or completeness.
- DART can be used during peer reviews to "grade" developers (determine how the overall code structure and design looks to determine if the programmer has successfully created a testable design).
- DART offers insight into a programmer's design mechan ics indicating their ability to meet program software compliance while assessing their coding competence.
- DART offers an immediate quick look mechanism for code and shows how a program satisfied a requirement techni cally allowing for future software reuse where capability is needed enterprise wide.

Identifying the association between the design, development, and transition of a capability in a common dashboard offers an opportunity to perform automated metric and report generation, analysis of requirements coverage, and testing as a whole. This will enable program personnel to gain firsthand experience in discovering the "need to fleet" associations, unleashing the potential of automated testing. Within the standard Systems Engineering V-chart for the development of capabilities, systems, and platforms, there should be more of a relational exchange of information, not a waterfall of data dumped from one activity to the next. Feedback loops from the transition or testing side of the V-chart should be tied directly to requirements, system design, and CP code for true testing accountability. This traceability into the systems engineering process allows for visibility into the artifacts associated with each piece of development. Only when this traceability is available can AT policies and procedures be introduced and employed to demonstrate true acquisition efficiency.

Contract Language

The majority of the contracts start at milestone B, where concept exploration is complete and technology is determined to be mature enough. Therefore, the stakeholders accountable for all T&E activity and responsible for creating the Test and Evaluation Strategy (TES), which is the initial planning document describing test activities throughout the acquisition lifecycle, are prepared to make key decisions on what is best for streamlining their test process from cradle to grave. Articulating in the TES that AT capability should be used could prove to be a major influence on contract language. This TES document is a key item to be considered when developing the contract Statement of Work for the RFP. The Office of the Secretary of Defense (OSD) T&E Contract Guide states:

"Acquisition planning is the process of identifying and describing contract requirements and determining the best method for meeting those requirements including solicitations and contracting. During the program lifecycle it is critical that the Project Manager (PM), systems engineer, and T&E personnel recognize that early and consistent incorporation of T&E considerations and requirements begins at the onset of program planning during the Material Solutions Analysis and Technology Development phases [2]."

Raising awareness and educating the T&E stakeholders involved with creating the TES through a conclusive business case analysis and identifying what the ROI is for AT will begin to influence policy consequently effecting contract requirements. Specific metric weighting based on stakeholders, as shown in the acquisition process below (Figure 2) (PM, Warfighter, Contractor, and T&E Tester), are provided throughout the rest of the document.

Metrics

The word metric has many different meanings; the following definitions provide a basis for this discussion. A metric is, "A system of related measures that facilitates the quantification of some particular characteristic" [3]. Implicit in the use of the word metric is the notion of a standard means of measuring something in a known space. Further calibration of the term metric comes from the definition, "The application of statistics and mathematical analysis to a field of study" [4]. This definition indicates the need for sufficient rigor and sensitivity in the description of the measures to provide meaningful quantitative results. These results can then be used to support various analyses such as ROI. Within each metric, a description and explanation of how it relates to the AT business case is provided along with any relevant categories in the context of DART.

Metric 1: Time

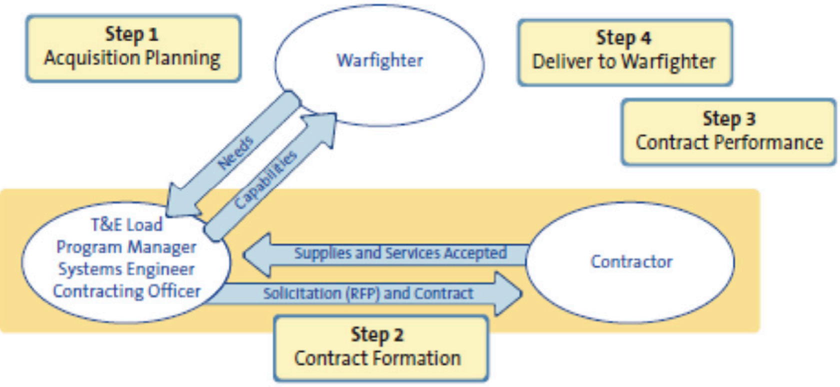
Time is a simple measure. It is well defined, understood, and standardized. While most of the systems being tested using AT technology are concerned with very small units of time such as milliseconds, the AT process must be focused on a much larger scale. In the context of AT, time has most to do with the calendar period need to perform a certain activity. As a business case metric, the timeframes of meaning typically span days, weeks, months, or quarters.

There are two primary categories of time, labor time and calendar time. Both factors are important to keep in perspective. Labor time is directly related to cost. Calendar time is directly related to delivery completion and risk. A reduction in either of these categories can be seen as a clear improvement in the ROI equation. Depending on the stakeholder, time impact varies. Time to the developer relates to earned value, contract management, and continuation. Time to the PM relates to meeting planned delivery to the platform manager and affects his measure of performance. Time to the platform manager relates to meeting planned ship delivery schedules, which can relate to significant penalty cost by a shipbuilder and/or lack of needed war fighting capability to the fleet. Time to the operational tester relates to certifying the operational effectiveness and safety associated with fleet use.

Metric 2: Delivery Risk

Delivery Risk is a measure of how likely a particular issue will negatively affect the ability to complete a project. This metric provides a qualitative adjustment to other quantitative metrics outlined in this document. Therefore, any assessment of delivery risk will be linked to an associated sub-category to which it belongs: Time/schedule impact, cost of system, and complexity/difficulty of implementation. There are several reasons that an issue can have an impact on delivery risk:

Figure 2: Acquisition Process



* If the number of functional requirements is very high
* The design complexity causes difficulty in procurement or development
* The number of defects detected is too high
* The schedule is aggressive
* There is insufficient funding

Time, schedule, and cost are well-understood factors, and are discussed in this document. There is a value in capturing the qualitative impact of these related to AT as well as the quantitative means described above. Difficulty of implementation is a category of risk that measures any extra effort, special skills, processes, equipment, or other factors that complicates (or simplifies) the execution of the project. In the context of AT, the question one must ask is, "How much delivery risk (increased or decreased) is introduced into my system as a result of AT?" This question should be asked in the context of time, schedule, and cost. Given the system under test and the stakeholder's viewpoint, the answer may have either a positive or negative affect on the ROI equation.

Metric 3: Quality

Quality is defined as the level by which a developed item meets or exceeds the requirements, endurance, and reliability specified in the design. The overall prevalence of quality software can be directly correlated to the magnitude of what has already been discovered, and the extent to which the software meets mission objectives can be measured. There are several characteristics of quality that can be considered:

- Opening and closing rate of defect reports
- Prevalence of defects
- Predictable/consistent performance
- Operational (reliable, available)
- Functional requirements coverage

Metric 4: Cost

Cost is the amount of expenditure required to complete an activity. There are many task contributors that impact the amount of cost incurred. This section divides the types of cost typically incurred and tracked in activities related to test efforts. These categories are not mutually exclusive, nor exhaustive of the cost elements that are possible.

There are many ways to stack and compare cost, or colors of money:

- Non-recurring versus recurring
- Operations and maintenance versus development
- Research and development maturity level (6.1, 6.2, 6.3, 6.4, etc.)
- Labor versus other direct costs

Various stakeholders' views provide different perspectives on which cost trade off is relevant, and what they feel is most important. Below are some categories of cost identified: Development cost, labor cost, license/equipment cost, and facility cost. Cost is simply measured in dollars. The scale of cost-relevance is proportional to the scale of the project. Cost relative to time saved or time lost will be shown in accessing each separate metric in parallel.

ROI

All relevant metrics need to be considered when the end goal is to deliver a better quality capability to the fleet faster, cheaper and at lower risk. In order to establish this level of ROI, a relationship between the different metrics must be established to determine a relative weighting of one metric to the next. This is a difficult task since different factors will weigh the importance of the metrics differently. The relationship is, therefore, stakeholder and T&E step dependent. An initial weighting may be established by polling the stakeholders, yielding a composite weighting, and using that through the life of the program. An alternative would be to have each stakeholder group perform their relative weighting of metrics and compute an ROI for each step based on the expectation from preceding programs.

An additional consideration must be a predicted effect on metrics for each stakeholder's responsibility and how this affects their desire to implement AT. As an example, DART would require developers to implement traceability in support of development and testing, therefore delivery may incur an additional cost to the program. This negative effect on cost and potentially time spent must be offset by a combination of positive effects on the other metrics of interest to them. If a relationship can be shown that utilizing DART will effectively reduce completion risk and increase quality sufficiently, then it becomes of interest to the PM. If it does not have this effect, it will be an uphill struggle to get PMs to accept the requirement.

Another benefit to automating testing is a higher confidence in the coverage of requirements during test. Many current and traditional testing methodologies, despite considerable effort, yield a wide degree of variability in the function points actually tested. It is believed that by utilizing automated tests (i.e. DART), one will know what has been tested, yielding higher confidence that critical items have been checked.

Summary

Based on the COMOPTEVFOR findings and the imminent change in paradigm to open architecture, there is a need for a more flexible and innovative way to improve synergy throughout the development and T&E communities within the acquisition framework. Traditional systems engineering process and contractual language evolution is needed; more effective approaches for testing are required to meet the CNO guidance for reducing T&E costs by 20%. However, as the business case for AT is evaluated more closely, it is apparent that there are metrics beyond cost that need to be considered when looking at the overall ROI associated with the desire to deliver tactical war fighting capability faster and cheaper while at the same time providing a higher quality product with equivalent functionality.

As the AT community grows and is leveraged throughout DoD, the business case for utilizing tools and techniques such as DART will need to be assessed along with its ease of use. Technical and enterprise maturity will need to be cataloged to establish a baseline for progress throughout component and system development. A program willing to identify and support development of standards for AT and bring key stakeholder metrics to the forefront will establish a true overall benefit and identify the impacts (both positive and negative) that AT has on its development. ROI metrics will help discover and establish the technical process and contract language for the application of AT to major software development efforts.

ABOUT THE AUTHOR

Mr. Brad Neal has 13 years of systems engineering experience supporting various DoD program offices and is currently the engineering services group lead at SimVentions. Efforts involve developing technologies to facilitate rapid technology transition and conducting business case analysis on current and traditional T&E processes to identify areas of opportunity for automation to maximize ROI. Brad has performed many automated software assessments discovering technical risk within a programs acquisition lifecycle.

REFERENCES

1. Department of Defense. CNO Guidance for 2004 . 2004. February 2010 <http://www.globalsecurity.org/military/library/policy/navy/cno-guidance_2004.htm>.
2. Department of Defense Directive 5000.1. Directive. Washington, D.C.: Department of Defense, May 12, 2003.
3. Dictionary.com. Metric. 2010. February 2010 <http://dictionary.reference.com/browse/Metric>.
4. Metrics. 2010. February 2010 <http://dictionary.reference.com/browse/Metrics>.