Continuous Integration in the Cloud

Improving Cost, Schedule and Technical Performance

Elfriede Dustin, Innovative Defense Technologies (IDT)
Kevin Caldwell, Innovative Defense Technologies (IDT)

Abstract. Program Managers need to continue to seek ways to improve cost, schedules and technical performance. This article provides a summary of industry best practices we have applied successfully that enable program managers to implement processes and practices that can result in the improved cost, schedule and technical performance that the DoD is continuously seeking.

Introduction:
For nearly 70 years, the Department of Defense (DoD) has engaged in a constant process of acquisition self-assessment, striving to continuously improve the way it acquires weapons systems. Generally, the concern is that acquisition costs are too high and the process is too slow, involving too many stove-pipes. According to the Performance of the Defense Acquisition System 2015 Annual Report, some positive change has taken place with various contracts and initiatives; however, program managers are still encouraged to actively seek ways "to save money and to set targets for doing so, not just to stay within their budgets [1]."

This article provides a summary of best practices we have implemented that are gaining momentum in the industry. These practices have resulted in the types of improved cost, schedule and technical performance that the DoD is seeking. Some of these systems engineering best practices include: virtualization, continuous integration, automated testing using Automated Test and ReTest (ATRT), and hosting continuous integration solutions in the cloud. These best practices have been applied to over 80 programs at IDT with significant, measurable results. The outcomes summarized below are taken from two "approved for public release" case studies from IDT’s work with NAVAIR [2] and NAVSEA [3].

NAVAIR
• Increased testing efficiency by greater than 75%.
  The result is significantly less time and manpower are required to conduct testing.
• The number of requirements, permutations and configurations being tested has increased along with consistency of the testing. In addition, test teams have been able to identify software defects earlier in the schedule.
• Automated test cases are being shared and reused across the responsible contractor and government teams (e.g. removing stove-pipes). Besides the efficiency of reusing test cases, the time and scope of incorporating automation is also being accelerated.

NAVSEA
• Increased testing efficiency for those critical system and software requirements where automation was applied. The result is significantly less time and manpower being required to verify the associated requirements and system performance.
• Improved collaboration among test teams. The application of ATRT facilitated efficient sharing of analysis cases between the various AEGIS test entities. As a result, each test team gained the ability to conduct more thorough analysis at each testing stage.
• Improved software quality and reduced risk. Automation has increased requirements coverage and expanded the data able to be evaluated to assess the system performance. Additionally, sharing of analysis methods between test teams has enhanced defect resolution.

Acquisition program managers face the challenge of not only grasping all practical business concerns, but also of understanding and managing a diverse range of topics, including: risk identification and mitigation, selection and integration of commercial off-the-shelf (COTS) components, process capability, program management, architecture, survivability, interoperability, source selection, continuous integration, software development tasks, verification and validation, and contract monitoring.

The use of a comprehensive suite of management capabilities is designed to orchestrate and optimize complex software engineering oversight, Continuous Integration (CI), and human-centric acquisition processes across the value chain. Next, we will provide an overview of a few of the technologies and best practices that in our experience can increase efficiencies and reduce the work load an acquisition program manager faces.

Virtualization and Continuous Integration (CI):
Virtualization and Continuous Integration are two of the biggest time and cost savers we have implemented for our customers. We've discussed virtualization and CI in detail in our article "Efficiencies of Virtualization in Test and Evaluation" [4] in the July 2013 edition of Crosstalk. CI is one of the best industry-adopted software engineering practices in which any change to the code or environment is tested and reported on as soon as feasible. In most cases this involves nightly software builds and nightly automated test runs to allow for quick look reporting on any newly introduced issues. Virtualized development and test environments play a major role in this CI practice. In “eating our own dog-food,” we’ve continuously expanded on these best practices. For example, we have implemented an increasingly efficient automated CI solution as a pluggable framework of CI applications which includes an automated process and the capability of being hosted in the cloud. We call this solution/methodology the CI-Cloud. Additionally, CI-Cloud orchestrates a tool-independent environment, and tools such as Jenkins, SVN and GIT version control systems are hosted and seamlessly integrated with project scheduling and management tools such as Redmine and requirements management tools. The features are described in further detail below.
**CI-Process Modeling:**

IDT has automated the modeling of the CI process, termed CI-Process Management (CPM), to provide a bridge between the customers receiving a delivery and developers and engineers implementing and testing a solution. Our customers can now receive continuous development status via access to the CI-Cloud, software with the automated test cases, and virtualized hardware (versus having to purchase their own hardware). This CI Process, built into the CI-Cloud, offers the following advantages:

- **The CI Process enables users to model Continuous Integration and Application Delivery goals via a flow chart which describes the steps needed and the order required to achieve that goal.**
- **The CI Process improves the visibility, monitoring and agility of software delivery logic, resulting in higher-level and domain-specific representations that can be understood by DoD customers and DoD contractors.**
- **Corporate and domain-specific CI-processes can be plugged into a modifiable palette, making the CI process more easily understood.**
- **This CI Process Management is not an isolated process engine. Complex CI logic can be modeled as a combination of CI processes with conversion and migration rules between existing CI environments and the CI-Cloud.**

(See Figure 1: “Process Modeling” for more details)

With this automated process modeling we are achieving a goal of 80% reduction in manual interactions and faster issue resolution by allowing DoD agencies to design, integrate, deploy, execute, monitor and optimize their critical software engineering acquisition processes and operations. This process will:

- Automatically prioritize and route work and tasks to stakeholders
- Guide users, contractors, developers, and program managers through decisions
- Standardize resolutions across geographies
- Leverage existing CI and Program Management systems and data
- Monitor for business events and initiate action
- Provide real-time visibility and process control

**CI Pluggable Framework/Application Store:**

This CI-cloud consists of a pluggable framework that allows for adding/removing CI applications and capability with ease.

- All CI-Cloud applications are portable and self-contained.
- Archives can be deployed via the CI 'appstore'.
- Upgrade, downgrade, stop, start, deploy, undeploy, as easy as clicking a button

This pluggable framework allows customers to choose their development environment with specifically preferred tools. For example, users can choose between a Java based/Eclipse development environment and a C/C#/Visual Studio/Team Foundation Server development environment. This framework comes with the build server of choice, along with the source control and automated testing solution. For example, it automatically provides access to Jenkins, SVN, and ATRT.

**CI in the Cloud:**

Much has been written about the need for and benefits of cloud computing, such as quicker and cheaper delivery and reduced hardware costs. “Tech historians will look at October 22, 2015 as a watershed,” according to New York Times reporter Quentin Hardy. He goes on to say, “Cloud computing is no longer on the way, just a contender, or even a competitor to traditional enterprise technology companies. Instead, it is here, full force, and all the signs are that it is about to get a lot bigger, fast [5].”
INTEGRATION AND INTEROPERABILITY

Our customer’s goal of being able to access CI in the cloud was various. A few of the goals were:

1. To allow for better coordination. Now, their Sprint backlogs and schedules can be accessed and modified in CI-Cloud via Redmine by Government and contractors alike;
2. To grant access to both developers and testers. Now, both groups can use CI-Cloud for development, testing, building, nightly automated tests, and results reporting;
3. To save money on hardware;
4. To increase visibility and insights into development / test progress; and
5. To move from manual disk / software delivery to an automated pushdown download and install.

(For more detail on this last goal, see Figure 3: “Before CI-Cloud and After”)

For CI-Cloud we chose to use Amazon Web Services (AWS) as our cloud service provider (though the CI-solution is cloud or environment agnostic).

This CI solution can be hosted in the cloud and it allows program managers to:

• Manage the entire software engineering lifecycle – from design to optimization
• Ascertaining continuous process improvement using closed-loop control
• Reduce time to introduce new CI and acquisition processes
• Improve stakeholder productivity
• Cut operational costs by up to 40 percent by automating and standardizing CI and reducing hardware cost
• Improve mission delivery and drive CI and software acquisition process agility
• Extend the ROI of existing CI and Program Management technology investments
• Ensure continuous compliance with internal best practices and regulatory standards
• Increase competitive advantage and DoD agency satisfaction

(For more detail on this last goal, see Figure 4: “How CI Cloud works”)

ATRT CI-CLOUD

Our article “Efficiencies of Virtualization in Test and Evaluation” [6] in the July 2013 edition of Crosstalk also provides detailed examples of automated software testing in a virtualized test environment which include: 1.) Automatic provisioning of a virtualized automated test environment; 2.) Automatic provisioning of the entire automated testing lifecycle for any type of SUT; and 3.) Continuous Integration using virtualized environments. By implementing those solutions we have been able to remove

CI-Cloud and Automated Test and ReTest (ATRT):

Our article “Efficiencies of Virtualization in Test and Evaluation” [6] in the July 2013 edition of Crosstalk also provides detailed examples of automated software testing in a virtualized test environment which include: 1.) Automatic provisioning of a virtualized automated test environment; 2.) Automatic provisioning of the entire automated testing lifecycle for any type of SUT; and 3.) Continuous Integration using virtualized environments. By implementing those solutions we have been able to remove
the stovepipes of testing, allowing vendors and government facilities located on opposite coasts to share their automated test procedures, reduce the repetition of work by reusing tests, and minimize silos.

(See Figure 5: “Reducing Test Silos” for more details.)

Cloud Security and Authority To Operate (ATO):

Many security measures will be shared or inherited due to CI-Cloud building systems on top of the AWS Cloud infrastructure. CI-Cloud will provide security for its software components, and Amazon AWS GovCloud will provide security for its infrastructure. CI-Cloud is able to leverage security controls from AWS's security, meaning that CI-Cloud will not have to provide those controls for its components since Amazon AWS GovCloud is already providing them.

CI-Cloud assumes responsibility for, and management of, the guest operating system (including updates and security patches), other associated application software, and the configuration of the AWS-provided security group firewall.

Figure 6: Amazon vs CI-Cloud security

As illustrated in Figure 6: “Amazon vs CI-Cloud security,” Amazon AWS GovCloud will provide security controls from the virtualization layer down to the physical security of the facilities in which the service operates.

For more on CI-Cloud and related ATOs, stay tuned for a follow-on article that discusses security in the AWS cloud in detail.

Implementing the CI solution in the cloud is just one step towards automating the many facets of the acquisition program. The acquisition program is subject to numerous influences, both internal and external to the program. Some influences such as budget constraints, schedule constraints, and performance requirements are well quantified and easily understood. Other influences, such as stakeholder agreements, requirements stability, and contractor capability, are more difficult to assess and less obvious. These influences, or program drivers, are sources of risk to an acquisition program. For these challenges, methodologies that identify and mitigate some of these risks are available. Much more can be done to streamline and automate the acquisition process. In this article we presented a subset of some of the proven best practices that have led to saving the government money and improving efficiencies.

REFERENCES


ABOUT THE AUTHORS

Elfriede Dustin is Director of Solutions at Innovative Defense Technologies (IDT) where she works on developing new ideas and discovering new approaches to software engineering challenges. Software development is still an art and that makes automated software testing and software engineering a special challenge. IDT (www.idtus.com) strives to meet that challenge by producing edge of technology solutions, starting with requirements through the entire secure software testing lifecycle to defect closure. Elfriede has a B.A. in Computer Science and over 20 years of IT experience, implementing effective software engineering processes and testing strategies, both on government and commercial programs. Together with IDT CEO Bernie Gauf and IDT FSO and Sys Admin Guru Thom Garrett, Elfriede wrote her latest book “Implementing Automated Software Testing.” Elfriede’s goal is to continue to help further the software engineering/automated software testing advances.

Book list: amazon.com/author/elfriededustin

Kevin Caldwell is a leading security scientist working at Innovative Defense Technologies (IDT). He has over 20 years of experience leading the design, development, and delivery of innovative security and IT solutions for the DoD and commercial entities. His government experience includes hands-on lead roles in security engineering, development, and the production of information systems and cloud technologies for NAVAIR, NAVSEA, Internal Revenue Service, Department of Labor/OSHA, FBI, FCC, and SPAWAR. In 2015, Kevin designed and developed a portable application hypervisor solution, based on top of the Amazon Web Services (AWS) Cloud, for Continuous Integration (CI) activities for NAVAIR and other DoD agencies. As part of these activities, he developed and secured AWS for a CI Cloud for JMPS/NAVAIR and he prepared ATOs based on DoD RMF (8510.01) for Cloud Services. Additionally, he designed, developed, and managed a secure, cloud-based data platform and CI environment solution for the U.S. Navy which was based on AWS infrastructure, but portable to other clouds. In 2013, he developed the concepts, architecture, and capabilities for the SPAWAR/NMCI/NGEN Information Security Manager (ISM), a custom Puppet infrastructure with visual controls, necessary to support the U.S. Navy’s complex Security Vulnerability Management and Automated Remediation of Vulnerabilities across the entire enterprise of over 500,000 endpoints.