AGILE IA

A Modus21 Case Study

Abstract

A case study on the application of Agile software development techniques to the Information Assurance team of a large, federal, software development project.

Modus21, LLC
http://www.modus21.com
August, 2015

To engage in a discussion regarding the
material contained within this case study
or to get more information,
please contact:

Authors
Peter Woodhull
CEO
843-513-1272
Peter.woodhull@modus21.com

Contributors
Randy Adkins
Consultant
843-991-6945
Randy.Adkins@modus21.com

John Dornisch
Senior Associate
843-991-9603
john.dornisch@modus21.com

The information contained in this document is for general guidance on matters of interest only. The application and impact of laws can vary widely based on the specific facts involved. Given the changing nature of laws, rules and regulations, there may be omissions or inaccuracies in information contained in this document. This document is provided with the understanding that the authors and publishers are not herein engaged in rendering legal, accounting, tax or other professional advice and services. It should not be used as a substitute for consultation with professional and competent advisers. Before making any decision or taking any action, you should consult with a Modus21 professional.

While we have made every attempt to ensure that the information contained in the document has been obtained from reliable sources, Modus21 is not responsible for any errors or omissions, or for the results obtained from the use of this information. All information in this document is provided "as is", with no guarantee of completeness, accuracy, timeliness or of the results obtained from the use of this information, and without warranty of any kind, express or implied, including, but not limited to warranties of performance, merchantability and fitness for a particular purpose. In no event will Modus21, its partners, principals, agents, or employees thereof be liable to you or anyone else for any decision made or action taken in reliance on the information in this document or for any consequential, special or similar damages, even if advised of the possibility of such damages.

Contents

Executive Summary

- The VA VBMS program needed an IA program that fit into an Agile Lifecycle Management (ALM) structure.
- The program's IA team imported NIST 800-53 controls into an ALM tool and managed them as Agile work items.
- This enabled the IA team to work as a Scrum, report performance in accordance with the project sprint schedule, and automate the generation of the System Security Plan (SSP).
- It also made the project's IA efforts compliant with VA requirements for Continuous Monitoring per VA Handbook 6500.3.

Background

The Veterans Benefits Management System (VBMS) initiative is the cornerstone of VA's technology transformation strategy. The purpose of VBMS is to develop and implement a comprehensive solution that integrates a 21st-century web-based, paperless processing solution complemented by improved business processes. The goal of the Paperless Initiative is to deliver automated Claims Processing services for the Veterans Benefit Administration (VBA) with a current focus on the Compensation and Pension (C&P) business line as an integrated investment. C&P, the largest VBA business line, depends on paper-based claims processing, with a substantive inventory and increasing backlog. The primary software component in the Paperless Initiative is the VBMS. VBMS is a paperless claims processing system that replaces legacy claims processing software, implements improved standard business practices for a paperless claims workflow, provides enterprise data services for external and internal communications, and is enabled by an enterprise, Service Oriented Architecture (SOA) framework. The VBMS SOA-based technology platform provides VBA an enterprise set of capabilities including an Enterprise Portal, Data Integration, Imaging, Forms Service, Rules-Based Processing, Correspondence, Messaging, Workflow Services, and Content Management. As an integrated investment, the SOA framework contributes to the mission by delivering the VBMS system that will provide C&P the ability to increase the automated handling and processing of over a million claims submitted a year and reduce inventory backlog, while continuing to ensure $3.7 billion in claims are paid each month. VBMS enhances services to Veterans and their families by facilitating business line integration for automated, rules-based claims processing for Education, Vocational Rehabilitation and Employment, Insurance, and Loan Guaranty, recognizing that some are highly automated through the VBMS SOA-based framework. This standardization of capabilities will reduce long term costs and align VBA enterprise requirements, which will align with the Office of Management and Budget requirements for integrating program acquisitions.

Problem Statement

The VBMS solution was developed and implemented utilizing an Agile Scrum software engineering lifecycle. The program was challenged to execute mandated Information Assurance (IA) practices within the context of that Agile lifecycle management structure. In accordance with standard Agile software development standards, the project team developed and deployed new functionality on a regular basis, either quarterly or bi-annually depending upon the scope of the new functionality.

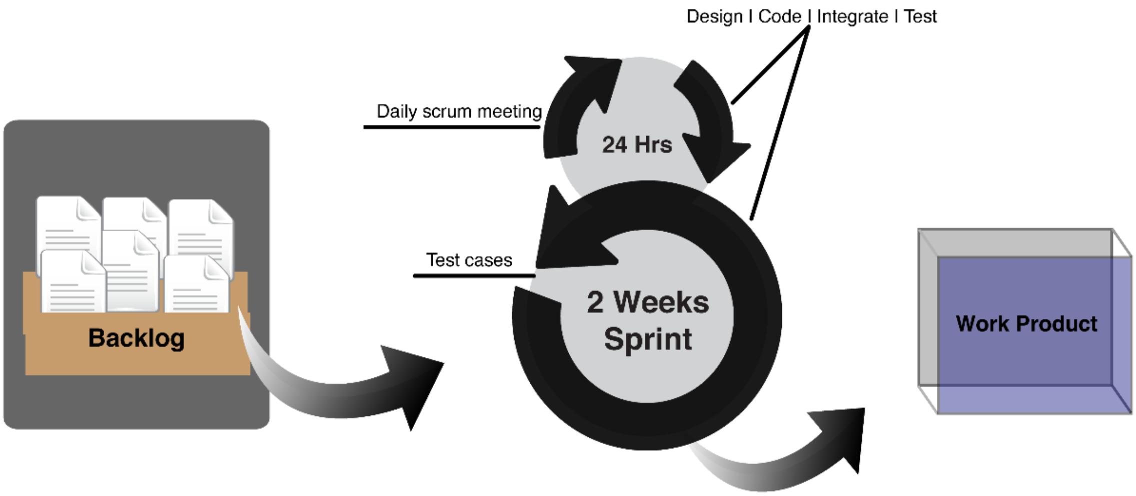


Figure 1: Sprint Based Development

Newly deployed functionality often included new architectural components, design and usability features, or integration capabilities. Additionally, numerous security controls and objectives were met via either VA infrastructure inheritance or Terremark hosting inheritance. As such, each application release contained material and structural modifications which required IA scrutiny and the potential for re-accreditation. However, the Agile nature of the project lifecycle also meant that testing happened on a recurring basis due to continuous integration capability. Often, stabilization Sprints only left 2 weeks for User Acceptance Testing (UAT) prior to deployment. Under these circumstances, the IA team would have been unable to successfully verify the security posture of the system during the time allotted between development testing and deployment.

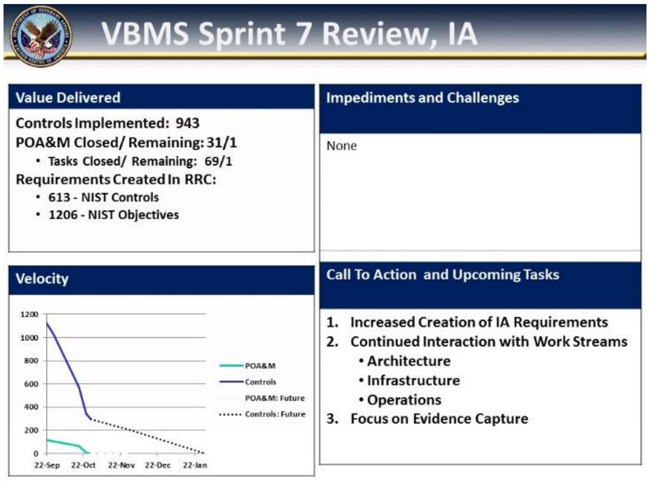
Need

With only a few weeks of time between integration testing and production deployment, the IA team needed a quick way to flexibly perform system security tasking that did not inject weeks or months of work into the project release cycle. The system security tasting included:

- Supporting Certification and Accreditation (C&A) activities to achieve and maintain the system Authority to Operate (ATO);
- Tracking the performance and progress of recurring tasks within development and DevOps;
- Demonstrating execution of tasks via audit tracking;
- Automating the generation of security documentation.

Because the entire project (a team of more than 250 individuals) was regularly meeting at Sprint and release reviews to demonstrate performance and review metrics, the IA team needed the capability to provide Agile metrics for activities which were not traditionally performed mid-stream. In effect, the IA team needed to illustrate its performance and the value delivered to the project on a bi-weekly Sprint and quarterly release cycle.

Additionally, due to the distributed and Agile nature of work on the project, the IA team needed a means of allocating work to all project team members within the context of their tasks as defined for a given project sprint.



VBMS Sprint 7 Review, IA

Value Delivered

Controls Implemented: 943

POA&M Closed/ Remaining: 31/1

- Tasks Closed/ Remaining: 69/1

Requirements Created In RRC:

- 613 - NIST Controls
- 1206 - NIST Objectives

Impediments and Challenges

None

Velocity

Call To Action and Upcoming Tasks

1. Increased Creation of IA Requirements
2. Continued Interaction with Work Streams
   - Architecture
   - Infrastructure
   - Operations
3. Focus on Evidence Capture

Figure 2: Sprint Review Slide

Finally, during the project lifecycle, the Secretary of Veterans Affairs certified the requirement that all VA programs be compliant with NIST SP 800-137, *Information Security Continuous monitoring for Federal Information Systems and Organizations*. The VA Handbook 6500.3 includes the directive that all programs must implement systems to facilitate continuous monitoring as a means of ensuring the security posture of IT systems. Although a grace period of implementation was established, the VBMS IA manager wanted to utilize the status of the project to demonstrate how continuous monitoring could be implemented effectively.

Solution

Unfortunately, there was no standard for how to execute Information Assurance within an Agile project, so the IA team took it upon themselves to develop one. They started by agreeing that they would operate like an Agile Scrum Team, and integrate into the overarching Agile model of the project. Then they analyzed the security requirements of the VBMS system to determine which NIST 800-53 controls and objectives were relevant for the project. The IA team worked with one of the project resources whose responsibility was to administer and configure the project's version of the IBM Rational JAZZ Platform. Together, they were able to quickly import the appropriate security controls and objectives into Rational Requirements Composer (RRC) as project requirements. For each of the objectives, they created a work item (Security Control) in Rational Team Concert (RTC), and linked the Security Control in RTC to the control requirement in RRC. Within RTC, the team defined a comprehensive Security Control lifecycle. Figure 3 illustrates the work items and relationships implemented by the VBMS IA team. This method enabled the team to define a periodicity for each Security Control, and allowed the team to assign the Security Control to a project team member for execution and/or evaluation.

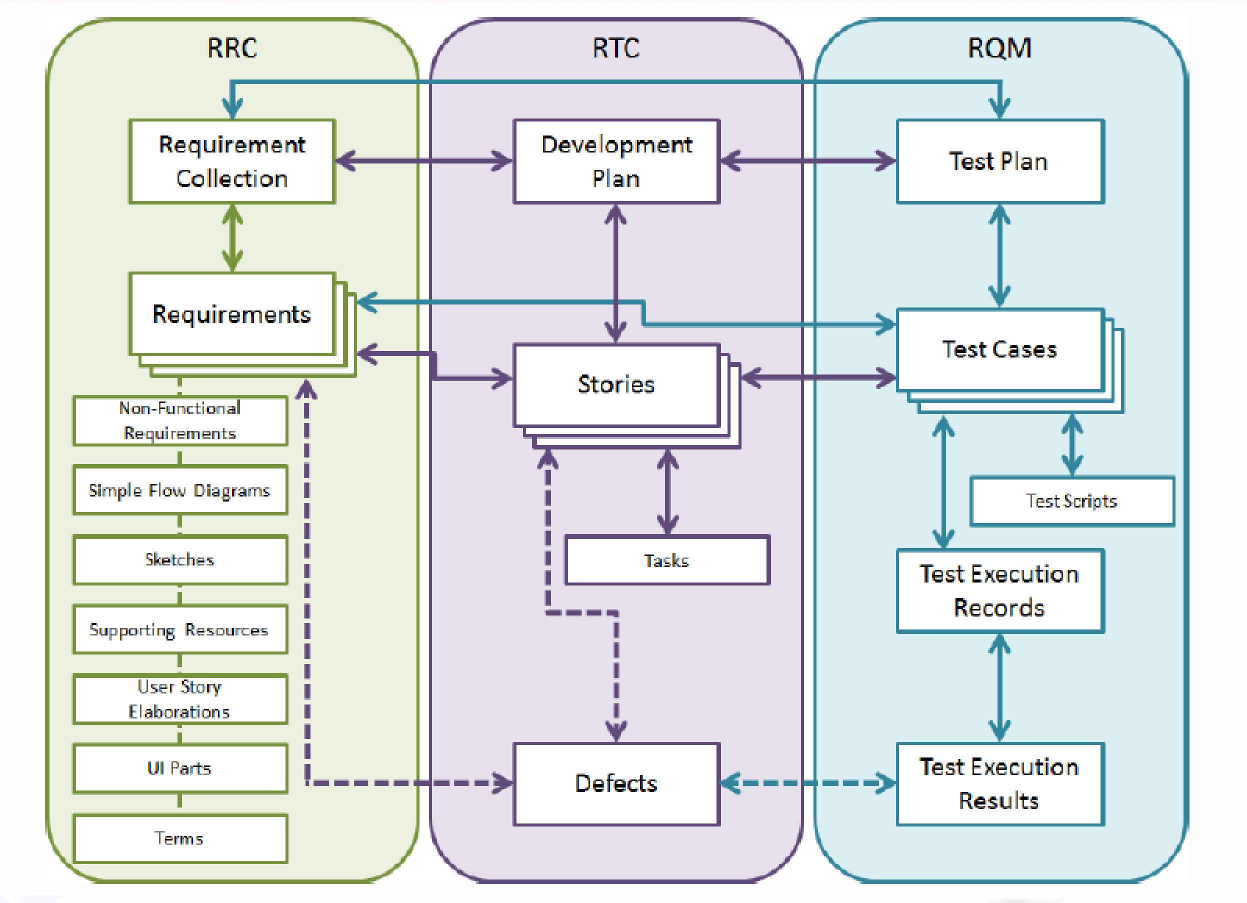


Figure 3: Work Item Traceability within project tools

The IA team then created a test script for each RTC Security Control. The test scripts were created within Rational Quality Manager (RQM), and linked to the appropriate Security Control within RTC. Each test script was executed to validate the status of the Security Control, and the test results were stored within RQM. If the test passed, then the Security Control was pushed forward to an appropriate state. If the test failed, then the Security Control was pushed back to a re-assessment state in which an IA analyst would work to determine why the test failed, and institute an appropriate remediation. Based upon the linkages maintained between RRC, RTC, and RQM, the IA team had a complete Requirements Traceability Matrix (RTM) that showed transactional details about how each requirement was implemented, tested, and verified.

Accordingly, a Security Control was able to be assigned, validated or tested, and pushed to a pending state until such time as it needed to be re-assessed. Appropriate results from the test script were maintained within RTC, as well as detailed audit logs for the Security Control. In effect, this model created a continuous monitoring capability that enabled the IA team to maintain constant visibility into the security posture of the project. Each control was evaluated on a regular basis determined by the project security policy. Some Security Controls had specific re-assessment periods based upon project or agency policy. Other Security Controls were re-assessed based upon random selection by RTC. Regardless of the mechanism, all Security Controls were re-evaluated on a recurring cycle with less than 12 months between re-assessments.

Inherent functionality of the IBM Rational JAZZ Platform includes the logging of all user activities associated with work items. As such, it was easy to verify “who did what” to which Security Controls, and when those actions were taken. Based upon this functionality, the IA team was able to automate the generation of systematic reports for all Security Controls and their current states.

Benefit

Based upon the ingenuity of the VBMS IA team, and utilization of the Agile Lifecycle Management (ALM) tools at their disposal, the project realized an Agile IA capability that had not existed prior. Additionally, because the ALM was used by every facet of the project, IA activities were able to be assigned and performed by many different resources across the program. The electronic Security Controls were created and managed by the IA team, but were assigned to appropriate resources for execution or assessment. The execution of the electronic Security Controls was directly integrated into their normal course of business as work items completed during Sprints. This substantially increased the speed at which security controls were assessed or verified while reducing the impact to project velocity by spreading the execution across the project team.

Additionally, the utilization of available technology to track and maintain the status of the NIST 800-53 Security Controls facilitated the automated generation of project security documentation, including the System Security Plan (SSP). Ultimately, a real-time SSP was able to be created via an export of information directly from the IBM Rational JAZZ Platform into a template document. This saved incredible amounts of time for both the IA team, as well as other resources throughout the project.

Ultimately, Agile practices enabled a small team of resources (~7 people) to perform the necessary Information Assurance tasking for a large program (~250 people) without impacting the deployment and release schedule. Continuous Monitoring was achieved via the recurring processing of electronic Security Controls throughout a defined lifecycle within a Commercial Off The Shelf (COTS) ALM tool.