INTEGRATING ECLIPSE WITH FLEXELINT AND BEYOND

Jerry Williams: NASA IV&V Software Assurance Tools (SWAT) Team
John Schipper: Ares J-2X IV&V Analyst
Jim Chamberlain: Ares J-2X IV&V Analyst
What We’re Going to Talk About

- Introduction
- Ares J-2X Example
- The Ares Solution
- Ares Results
- Implications for IV&V
- Takeaway Points
  - IV&V Tool Benefit Increased through Integration
  - Actively Look for Opportunity to Integrate Tools
  - SWAT Can Help
- Wrap-Up
What is a “tool integration”?

A tool integration increases the effectiveness of individual IV&V tools by combining them in complementary ways.

Analysts use and create tool integrations regularly.

Tool integrations are certainly not a novel idea, but they can be the “low-hanging fruit” of IV&V tool innovation we all want at the IV&V Facility.
A Real World Example: Ares J-2X Project

- IV&V would soon receive an engine controller code drop from the developer.
- Multiple code drops would be received on an accelerated schedule to support the first hotfire test at Stennis Space Center (SSC).
- IV&V task was to verify and validate that the code correctly implements requirements, met the operational need under nominal and off-nominal conditions, and introduced no unintended features.
- IV&V would perform static code analysis and requirements-to-code traceability in conjunction with NASA J-2X Avionics Insight Team reviews.
- IV&V analysts considered how the current IV&V toolset could help with these tasks, posed “what if?” questions, and requested tooling assistance from SWAT.
Solution: SVN to Manage Code Drops

Subversion (SVN):
- An open source tool widely used by IV&V projects
- Access to code is controlled through permissions.
- Provides a central repository for analyst use.
- Maintains version history between versions.
- Source code is easily available for checkout to environments where it can be used for analysis.
- Integrates with Eclipse through the Subversive plug-in.
Solution: FlexeLint for Static Code Analysis

- FlexeLint:
  - A powerful static analysis tool for C/C++ code to identify bugs and other potential code issues.

- Initial Manual Solution:
  - Analyst pulled the code from SVN and executed the FlexeLint run script against the code.
  - The results file was then made available to the team.

- Later Automated Solution:
  - A server-side script automatically pulled the code from SVN, executed FlexeLint, and then placed the results back into the SVN repository for access by the analysts.
Solution: Integration with Eclipse

- The IV&V team discovered a technique by which FlexeLint output messages could be ingested into Eclipse to “mark up” the source code being displayed to the user.

- As the tool integrations were streamlined, the eventual result was the ability to request FlexeLint analysis within Eclipse, which would cause the server-side analysis script to run FlexeLint, and finally allow the analyst to bring the most recent results onto his or her workstation.
Eclipse/FlexeLint Integration Screenshot
Solution: Conducting Analyses

- The IV&V team used the tool integration through manual execution by one of the team members to conduct the static analysis activities for the project.
- Eclipse provided an intuitive method for viewing source code that was tagged with FlexeLint messages.
- The capabilities of SVN allowed differences between developer code drops to be assessed.
Solution: Requirements Traceability

- IV&V analysts identified a method of tracing requirements to code by using the Eclipse Task view.
- The project was able to use this simple technique to intuitively and efficiently associate lines of code with specific requirements.
- Area for further improvement – semi-automated “promotion” of traces in eclipse from current to new code versions
  - Utilize patch files?
Solution: Exporting Data to Excel

- Without customization there is a limited set of attributes with a tag that are, in general, not IV&V related.
- The requirements traceability references were exported from Eclipse to Excel:
  - Currently manual and experimental
  - Another potential valuable tool integration
- The Excel spreadsheet was then used to complete the analysis in typical IV&V fashion:
  - Documentation of issues
  - Integration of analyst results
Results of the Ares Tool Integration

- The integrated toolset provided a powerful, relatively simple interface for quickly performing the code analysis.
- The code analysis identified dozens of issues of various severity, and one new risk.
- At the MSFC avionics insight team reviews over the past 6 months, IV&V was the primary provider of code analysis feedback to the developer.
- The IV&V project has been extended by the J-2X project office through FY11 as a reimbursable project, and FY12 funding is in planning stage.
- Part of this success is the code analysis expertise that IV&V brings to the J-2X project.
General Application to IV&V

- The solution developed for Ares may be one which could be customized for existing and future projects.
- The types of integrations performed for Ares can also be used to support similar analysis needs on other projects.
The benefit of individual IV&V tools can be increased by developing integrations that:

- Make up for shortcomings in the tool (e.g., difficulty of use or non-intuitive display of results).
- Better adapt the tool for use in an IV&V environment (many tools used by IV&V are intended for use by developers).
- Free the analyst from manual or labor-intensive tasks, allowing more time to be spent on analysis (automation).
- Allow analysis results produced by tooling to be automatically captured “in context”.

Point: Increased IV&V Tool Benefit
Point: Identifying Potential Integrations

- Ask yourself, “How could we do this better?”
- Recognize labor-intensive, repetitive tasks.
- Recognize any shortcomings of the tools and the methods a project uses to overcome them.
- Recognize any tool integrations developed within IV&V teams that might be useful to others.
- Analysts and those who support them should be looking to identify potential integrations.
Point: SWAT Can Help

- The SWAT team possesses knowledge of the IV&V toolset as well as the engineering skill to integrate tools to help with a custom solution for your project.
- Tooling solutions and integrations developed within IV&V teams are valuable! Work with SWAT to improve them and allow them to be shared with analysts on other projects.
- Just ask! If you are considering a tool integration for your project, SWAT may be aware of an existing solution or be able to offer guidance or support.
Wrap-up: Requesting SWAT Assistance

- The easiest way to request assistance from SWAT is to email the team, but you can use any of the methods below:
  - Email: ivv-swat@lists.nasa.gov
  - Phone: (304) 367-8404
  - Intranet: http://tools/ or www.tools.ivv.nasa.gov
  - In Person: Building II, 5000 NASA Blvd

- For detailed questions about the SWAT support of the Ares J-2X tool integration or this presentation, contact Jerry Williams directly:
  - Email: jerry.c.williams@ivv.nasa.gov
  - Phone: (304) 816-4191
Wrap-up: Demonstration

- A demonstration of the tool integrations described for the Ares J-2X project will be provided in a follow-on session.
- Please join us if you would like to see the live demonstration of:
  - SVN source code access from Eclipse
  - Remote execution of FlexeLint from Eclipse
  - Ingestion of FlexeLint results into Eclipse
  - Eclipse source code display with FlexeLint results
  - Other example tool integrations
Summary

- The Ares Solution and Results
- Implications for IV&V
- Takeaway Points
  - IV&V Tool Benefit Increased through Integration
  - Actively Look for Opportunity to Integrate Tools
  - SWAT Can Help