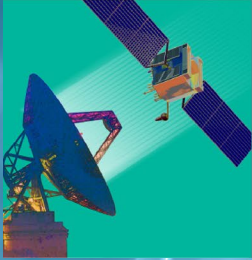


Working Group B Outbrief



Ground System Architectures Workshop Driving Innovation for Enterprise Integration

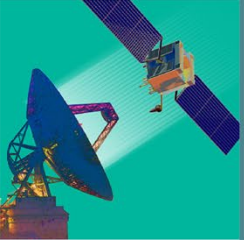
February 23–March 3, 2022 | Virtual Event

Integration and Test Approaches for Modern Ground Software Systems: What Works and What doesn't at the Code, System, and System of System Levels

*Leads:
Robert Crombie and
Jason McKenney,
The Aerospace Corporation*

February 28, 2022

© 2022 The Aerospace Corporation

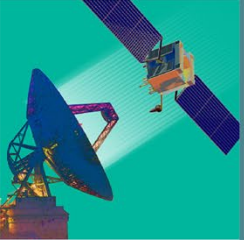


Session Goals

Integration and Test Approaches for Modern Ground Software Systems: What Works and What Doesn't

- Learn how participants have addressed the rapid pace and increasing integration and test frequency when Continuous Integration and Continuous Delivery (CI/CD) methods are used.
- Discuss how automated tests can cover important functionality and learn if participants have found more frequent I&T of smaller changes to be cost effective.
- Present and discuss topics related to the developer level where the software branches need to be merged into a software component and tested automatically.
- Present and discuss topics related to the system level, where many components of the ground system need to be demonstrated together.
- Present and discuss topics related to the system of systems level, where the ground system needs to interoperate with other systems, such as the space system or a data processing system.

Working Group B

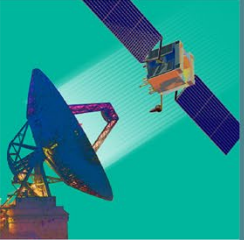


Presenters/Panelists

Integration and Test Approaches for Modern Ground Software Systems: What Works and What Doesn't

- Presenters:
 - *Jason McKenney*
 - *Robert Crombie*
- No additional panelists

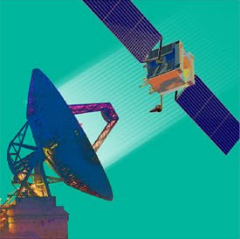
Working Group B



Key Points

Integration and Test Approaches for Modern Ground Software Systems: What Works and What Doesn't

- Compared traditional I&T approaches to Agile or more modern approaches.
- Challenges encountered in the modern build, test, and deploy processes.
- Determining which test cases to automate.
- System integration techniques with respect to regression testing, configuration management, and non-functional testing.
- Ways to handle cross-organizational I&T planning challenges.

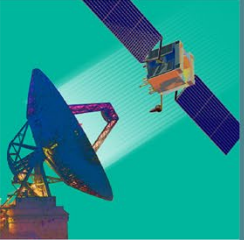


Conclusions

Integration and Test Approaches for Modern Ground Software Systems: What Works and What Doesn't

- **Developer Level**
 - *Work closely with CIO/cyber security to ensure security is implemented properly*
 - *Automated testing does not come free*
 - *Regression testing is crucial*
 - *Maintaining the regression test suite is important; remove tests no longer needed, break up regression suites by functionality or feature*
 - *Having a bug fix triage process up front so everyone knows when a fix or patch will be released, also an aspect of configuration management*
 - *Working with contractors – each may have their approach of the best way to implement the development, integration and test*
- **System Level**
 - *Increased cost from test events leads to less rework at the end and getting the right features to the user*
 - *Deployment of smaller changes helps to identify root causes of anomaly*
 - *For regression testing, you don't have to validate everything. Managing what is tested and removing duplication is necessary*
 - *How you handle non-functional testing is always interesting when implementing continuous delivery or iterative development*
 - *There are advantages of testing on cloud systems, containers, or VMs, but there are limits*
 - *Implement a set of tests that try to break the security of a system (hacking, denial of service, etc.) that targets security vulnerability and violations*
 - *Dealing with people can be challenging, especially when objectives are not aligned. Always look for ways to improve communication and improve (common) objectives*

Working Group B



Conclusions

Integration and Test Approaches for Modern Ground Software Systems: What Works and What Doesn't

- **System to System Level**

- *Test cultures between systems/organizations may differ*
- *Create an Integration & Test Working Group with representatives from all organizations, as soon as possible*
- *When exchanging simulators, consider links between labs if all-in-the-same-place doesn't work*
- *We develop using Agile, but we test a system to a traditional project specification - they don't line up cleanly*
- *Analysis of volumetrics is not acceptable - run at capacity, run at the expected volume, and for the duration they expect*
- *Provide representative data for the system-to-system test*
- *Communicating with contractors – define a common approach to integration and test*
 - *Established a cross-team, cross-contractor process to address those differences in cadences and styles*
- *Schedule - If different contractors' teams are agile, consider aligning the sprint start/end dates and durations, for easier exchange of S/W, test schedules, etc.*

Working Group B