NASA’s Software Architecture Review Board’s (SARB) Findings from the Review of GSFC’s “core Flight Executive/Core Flight Software” (cFE/CFS)

Lorraine Fesq and Dan Dvorak
Jet Propulsion Laboratory, California Institute of Technology
NASA Software Architecture Review Board

Flight Software Workshop
November 7-9, 2012
Southwest Research Institute
San Antonio, Texas

Copyright 2012 California Institute of Technology. Government sponsorship acknowledged. The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.
Background

Software Architecture Review Board

- SARB establish in 2009 based on recommendation from FSW Complexity study to Office of Chief Engineer
- Funded as a NESC technical discipline team by Michael Aguilar
- Several reviews conducted, varying in duration and depth
cFE/CFS Background

- Developed by GSFC Flight Software Systems Branch in response to growing costs and schedule for SW development due to increasing system complexity
- Project-independent FSW provides run-time environment and services for hosting applications
- Targeted for Class B FSW for Robotic s/c and instruments
- Domain: C&DH, GN&C, thermal, power, instrument control
- Users: ARC/LADEE, JSC/Morpheus, APL/RBSP
“Lollipop” Diagram shows cFE core applications and software bus (green), plus CFS applications that plug into the bus (blue and purple).
cFE/CFS Review Team

- Michael Aguilar (NESC, NASA Software Discipline Expert)
- Dan Dvorak (JPL, SARB Lead)
- Lorraine Fesq (JPL, review chair)
- Robyn Lutz (Iowa State University) – Product Line expert
- Michael Madden (LaRC)
- Pedro Martinez (JSC)
- Alex Murray (JPL)
- John Weir (MSFC)
- Steve Williams (APL)
Review Objectives & Focus

- Objectives:
  - Help project identify issues that may have been overlooked
  - Recommend actions to minimize downstream problems

- Focus will be on software architecture
  - not detailed design, not code, not avionics

- This is an engineering peer review
  - Tabletop review style, not primarily presentations to board

- Report:
  - Board report finalized January 2012
  - Report restricted to GSFC 582 management unless they permit broader release
What an architecture review is NOT

An architecture review is …

- not a gate, not a mandatory review
- not a pass/fail judgment
- not an audit for a cancellation decision
- not an evaluation of architect’s performance
- not a tutorial
- not a code review
Findings

Well thought-out, perhaps partly due to systems engineers and FSW engineers in same organization, promoting collaboration

Four categories of findings

- Governance
- Use on Projects
- Architecture
- Documentation
Findings: Governance

Meets a need across NASA, used by several projects at multiple Centers

• Has potential to become a dominant architecture framework for NASA FSW

• Lacks a business model - requires formal support for full benefit of product line to be realized
Findings: Use on Projects

Users at Multiple Centers were interviewed

- Technology viewed as mature – easy to build and test
- Promotes collaboration across Centers
- Code violates some standards
- Applications outside of original scope likely will require enhancements
- Could provide valuable training for pipelineing students – open-source availability
Findings: Architecture

Highly regarded by the Board

- Development guidelines for app layer exert a positive influence on architecture

- Use of pub/sub SW bus
  - allows for distributed development and easy integration
  - Well-encapsulated apps improve abstraction, flexibility, reuse, division of concerns
  - Could result in non-deterministic/non-repeatable execution
Findings: Architecture – cont.

- Modular components, well-defined I/Fs
- cFE shields apps from data structure formats
- OSAL allows easy use of different Operating Sys
- cFE can be used Stand-alone
- Message queue overflow handling
  - Drops newer messages
  - Subscriber not notified
- Seconds and sub-seconds derived from different sources, which could lead to timing issues
Findings: Documentation

SARB often find that the documentation doesn’t describe all the key aspects that future users ought to know. Utility/longevity limited by quality, depth, maintenance of architectural description

- ADD incomplete
- ADD uses ad-hoc graphical notation
- Discrepancies in representation and terminology
- Document what has been used on projects
- ADD does not identify required vs optional
Findings: Documentation – cont.

- Distinction between cFE and CFS components not clear in ADD
- Need view of connections between publishers and subscribers
- Need description of dependencies among source packages
- Need rationales for design decision and underlying assumptions
- Need testing guidelines
- Conceived to meet GSFC’s Earth-orbiter needs; no insight into architectural
Findings: Documentation – cont.

- QoS attributes not well documented
- Need guidance for complex, FT, autonomous control systems
- Need definition of FM philosophy – Limit Checker meets EO needs
- Need start-up procedures
- Need expanded time-services description
- Provide info to configure, execute, analyze performance data
- Document/analyze flight/ground division
Conclusions/Summary

- cFE/CFS Architecture highly regarded by the SARb
- Well-thought out – much potential
- Needs improved documentation
- Needs Governance and support to reach full potential
- Users outside of EO community should proceed with caution
SARB’s website is a sub-Community of the Software Engineering Community of Practice
https://nen.nasa.gov/web/software/sarb
SARB Mission:
Manage flight software complexity through better software architecture

Charter

- Provide constructive feedback to flight projects in the formative stages of software architecting
- Focus on architectural improvements to reduce and/or better manage complexity in requirements, analysis, design, implementation, verification, and operations
- Spread best architectural practices, principles, and patterns across flight software centers
Benefits of Architecture Reviews

• "Architecture reviews tend to increase quality, control cost, and decrease budget risk."

• "In our experience, the average [architecture] review pays back at least twelve times its cost."
  ◦ [Daniel Starr and Gus Zimmerman, *STQE Magazine*, July/August 2002]

• Beneficial side effects:
  ◦ The review process trains people to be better architects
  ◦ Cross-organizational learning is enhanced
  ◦ Architectural reviews get management attention without personal retribution
  ◦ Architectural reviews assist organizational change
## Perceived benefits of architecture review

<table>
<thead>
<tr>
<th>Benefits/goals of architecture review</th>
<th>Responses (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Identifying potential risks in the proposed architecture</td>
<td>88</td>
</tr>
<tr>
<td>B Assessing quality attributes (for example, scalability, performance)</td>
<td>77</td>
</tr>
<tr>
<td>C Identifying opportunities for reuse of architectural artifacts and components</td>
<td>72</td>
</tr>
<tr>
<td>D Promoting good architecture design and evaluation practices</td>
<td>64</td>
</tr>
<tr>
<td>E Reducing project cost caused by undetected design problems</td>
<td>63</td>
</tr>
<tr>
<td>F Capturing the rationale for important design decisions</td>
<td>59</td>
</tr>
<tr>
<td>G Uncovering problems and conflicts in requirements</td>
<td>59</td>
</tr>
<tr>
<td>H Conforming to organization’s quality assurance process</td>
<td>55</td>
</tr>
<tr>
<td>I Assisting stakeholders in negotiating conflicting requirements</td>
<td>43</td>
</tr>
<tr>
<td>J Partitioning architectural design responsibilities</td>
<td>40</td>
</tr>
<tr>
<td>K Identifying skills required to implement the proposed architecture</td>
<td>40</td>
</tr>
<tr>
<td>L Improving architecture documentation quality</td>
<td>40</td>
</tr>
<tr>
<td>M Facilitating clear articulation of nonfunctional requirements</td>
<td>31</td>
</tr>
<tr>
<td>N Opening new communication channels among stakeholders</td>
<td>31</td>
</tr>
</tbody>
</table>

---

What’s different about this review?

- cFE/CFS is a reference architecture, not a point design architecture
  - It is intended for a class of missions that share commonalities despite differing requirements
  - It has variation points to address differences
  - The developers in customer missions are not the architecture’s developers

- Some reviewer questions will focus on such aspects