How to Perform a Rapid Assessment of any Software Architecture

Tim Kertis, Principal Software Engineer, Raytheon
3 May 2017

SATURN 2017
Who Am I?

- Tim Kertis, Principal Software Engineer/Software Architect
- Chief Software Architect, Raytheon IIS, Indianapolis
- Master of Science, Computer & Information Science, Purdue
- Software Architecture Professional through the Software Engineering Institute (SEI), Carnegie-Mellon University (CMU)
- Over 30 years of diverse Software Engineering experience
- Currently working in the V-22 Avionics department
Schedule

- **Day 1:**
  - Interview Software Technical Lead
  - Complete the Form/Checklist and Capture:
    - Software Quality Attributes
    - Key Architectural Decisions
    - References to Architectural Design Artifacts

- **Day 2:**
  - Verify Software Design Artifacts
  - Analyze the Captured SWA Information
  - Produce an Opinion/Report
  - Distribute Report to Stakeholders, Managers, SW Technical Lead
Software Quality Attributes

- List of Stakeholders
- Stakeholder Views and Opinions
- Prioritized Set of SW Quality Attributes
  - Elicited
  - Recorded
  - Analyzed

- Software Quality Attributes Workshop (documentation)
  - For large projects
Key Architectural Decisions

- Computing Platform
- Software Technology
- Software Development Tools
- Software Reuse Strategy

Capture the Decisions and the Reasons Why

5/3/2017 | 5
Computing Platform

- Hardware/Processor
- Operating System
- Graphics Cards/Drivers
- Database Application Interface (API) Drivers
- Data Bus Communication Cards/Drivers
- Real-Time Operating System (RTOS), Board Support Packages (BSPs) and Hypervisor
Software Development Tools

- Software Development Design Tools
- Software Development Graphics
- Implementation Tools
- Graphics Development Tools
- Database/Persistence Frameworks
- Bus Communications Development Tools
- IPC Development Tools
Software Technology

- Software Design Methodology
- Programming Languages and Mixed Technology
- Graphics Technology
- Persistence Technology
- Bus Communication Protocols
- Inter-Process Communications (IPC) Mechanisms
Software Reuse Strategy

- Product Line Architecture
- Internal Software Components
- External Software Components
- Software Design Patterns
- Historical Productivity Cost Estimation Data
- Software Processes
Architectural Design

- UML Use Cases
  - Use Case Diagrams
- UML Software Components and Interfaces
  - Class Diagrams
  - Component-Connector Diagrams
  - Sequence Diagrams
  - Interface Definition Description
- UML Software Component Deployment
  - Deployment Diagrams
- Other Modeling Languages
  - AADL
  - SysML
  - etc.
Software Architecture Report

- Form/Checklist
- Report
- Distribution
## Software Architecture Review:

<table>
<thead>
<tr>
<th>Project Name:</th>
<th>______________________________________________________________</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architect Name:</td>
<td>______________________________________________________________</td>
</tr>
<tr>
<td>Project Engineer:</td>
<td>______________________________________________________________</td>
</tr>
<tr>
<td>Customer:</td>
<td>______________________________________________________________</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>______________________________________________________________</td>
</tr>
<tr>
<td>Reviewers:</td>
<td>______________________________________________________________</td>
</tr>
<tr>
<td>Review Date:</td>
<td><em><strong><strong>/</strong></strong></em>/___________</td>
</tr>
<tr>
<td>Estimated Code Size:</td>
<td>_____________ K ELOC</td>
</tr>
</tbody>
</table>
# Software Quality Attributes:

- [ ] Functional Suitability
- [ ] Reliability
- [ ] Modifiability
- [ ] Security
- [ ] Scalability
- [ ] RASU (Reliability, Availability, Serviceability, Usability and Installability)
- [ ] FURPS (Functionality, Usability, Reliability, Performance and Supportability)
- [ ] RASR (Reliability, Availability, Scalability and Recoverability) [databases]
- [ ] ACID (Atomicity, Consistency, Isolation (or Integrity) and Durability [databases]
- [ ] RAMS (Reliability, Availability, Maintainability and Safety) [safety critical systems]
- [ ] Agility (Debug Ability, Extensibility, Portability, Scalability, Securability, Testability & Understandability)
- [ ] Dependability (Availability, Reliability, Safety, Integrity and Maintainability)
- [ ] Other ________________________________________________________________
### Key Software Architecture Decisions:

<table>
<thead>
<tr>
<th>Hardware/Driver/OS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[] RTOS/BSPs/Hypervisor:</td>
<td></td>
</tr>
<tr>
<td>[] Bus Communication Card/Drivers:</td>
<td></td>
</tr>
<tr>
<td>[] Database API Drivers:</td>
<td></td>
</tr>
<tr>
<td>[] Graphics Card/Drivers:</td>
<td></td>
</tr>
<tr>
<td>[] Operating Systems:</td>
<td></td>
</tr>
<tr>
<td>[] Hardware Platforms:</td>
<td></td>
</tr>
<tr>
<td>[] Other:</td>
<td></td>
</tr>
</tbody>
</table>

Capture the Hardware/Drivers/OS Selected
### Key Software Architecture Decisions (continued):

**Software Technology**
- [ ] IPC Mechanisms: ____________________________________________
- [ ] Bus Communication Protocols: ____________________________________________
- [ ] Persistence Technology: ____________________________________________
- [ ] Graphics Technology: ____________________________________________
- [ ] Programming Languages: ____________________________________________
- [ ] Mixed Language Bindings: ____________________________________________
- [ ] Software Design Methodology: ____________________________________________
- [ ] Other: ____________________________________________
## Key Software Architecture Decisions (continued):

### Software Tools
- [ ] IPC Development Tools: ____________________________
- [ ] Bus Communication Dev Tools: ____________________________
- [ ] Database/Persistence Frameworks: ____________________________
- [ ] Graphics Dev Tools: ____________________________
- [ ] Integrated Dev Environment: ____________________________
- [ ] Software Design Tools: ____________________________
- [ ] Other: ____________________________

Capture the Software Tools Selected

Approved for Public Release
<table>
<thead>
<tr>
<th>Software Reuse Strategy</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ] Software Processes:</td>
<td></td>
</tr>
<tr>
<td>[ ] Historical Productivity Data:</td>
<td></td>
</tr>
<tr>
<td>[ ] Software Design Patterns:</td>
<td></td>
</tr>
<tr>
<td>[ ] External Software Components:</td>
<td></td>
</tr>
<tr>
<td>[ ] Internal Software Components:</td>
<td></td>
</tr>
<tr>
<td>[ ] Software Product Line:</td>
<td></td>
</tr>
<tr>
<td>[ ] Other:</td>
<td></td>
</tr>
</tbody>
</table>
Review Form/Checklist page #7

Architectural Design:

Use Cases
[ ] Use Case Diagrams: ____________________________

Software Components and Interfaces
[ ] Class Diagrams: ____________________________
[ ] Component-Connector Diagrams: ____________________________
[ ] Sequence Diagrams: ____________________________
[ ] Interface Definition Descriptions: ____________________________

Software Component Deployment
[ ] Deployment Diagrams: ____________________________
SWA Report

- Powerpoint Slides
- Word Document
SWA Report Distribution

- Stakeholders
- Software Technical Lead
- Software Department Manager
Summary

- **Day 1:**
  - Interview the Software Technical Lead
  - Capture
    - Software Quality Attributes
    - Key Architectural Decisions
    - Architectural Design

- **Day 2:**
  - Verify and Document Design Documentation References
  - Analyze the Software Architecture
  - Produce a Completed Checklist and Report
  - Distribute the Report to Stakeholders, Managers, Software Technical Lead

Complete the Assessment in Two (2) Days

Approved for Public Release
How to Perform a Rapid Assessment of Any Software Architecture