A Successful Framework for Rapid Development, Safety and Software Reuse

Alison Joseph

Tony Ponko

July 2014

Overview

- Background
- Challenges
- Solutions
- Framework
- Lessons Learned
- Video
- Questions

Background

- Development of new electronic safe and arm device
  - Experienced product development team assembled
  - Legacy LM work products selected as baseline
    - Expectations
      - Rapid development
      - Software reuse (leverage previous safety compliance)
    - Reality
      - Good starting point
      - Time and effort still required to ensure compliance with current safety standards/requirements

Reuse is a valid approach…but safety compliance not assured

Challenges

- Safety engineering requirements/guidelines had evolved
  - Required verifying compliance with current mission requirements/safety guidelines
- Reuse – Not so fast…
  - Code not “drag and drop”
    - Reuse code baselined several years ago

Evolution of safety guidelines can impact reuse strategy

Solutions

- Reuse as a guideline
  - Manage expectations
- Safety Engineering as a design partner
  - Understands current requirements
  - Guides systems/software efforts
- Modified framework with safety in mind
  - Insert “compliance mindset” into existing development framework

Involve Safety Engineering up front

Framework Overview

- ❑ Create plan
- ❑ Form Safety Compliance Working Group (SCWG)
- ❑ Conduct Tailoring Workshop (SCWG)
- ❑ Create baseline work products
- ❑ Create design
- ❑ Implement design
- ❑ Test

- ❑ Safety Board Presentations

Basic design flow with Safety interlaced throughout entire process

Framework

- Create plan
  - Layout schedule
    - Include time for Compliance Assessments
    - Include time for Fuze Board reviews
  - Identify all work products up front
    - What is required (Systems, Software, and Safety)
    - Establish reuse strategy
    - Who “owns” work products
    - Resources required to produce work products
    - Safety Compliance Checklists to verify work products

Identify and schedule ALL work products and reviews

Framework

- Form Safety Compliance Workgroups (SCWGs)
  - Internal
    - Program Lead, Safety, Systems, Software, Electrical, Quality
  - External (Layered)
    - Internal SCWG and Customer
      - Concurrence / partnering
      - Review safety presentations before Fuze Board meetings
    - Internal SCWG, Customer and Fuze Board members
    - Become a team
    - Ask questions
      - They want to help you succeed

Team insights increase confidence

Framework

- Form Safety Compliance Workgroups (SCWGs)
  - Reviews reuse strategy
  - Reviews requirements and safety impact
  - Reviews design and safety impact
  - Reviews Safety Compliance Checklists status/progress
  - Provides multi-disciplined insight with compliance questions/concerns

Safety compliance monitored throughout

Framework

- Convene customer Technical Interchange Meetings
  - Keep customer in-the-loop
  - Discuss progress / concerns / obstacles
  - Discuss requirements / design / safety changes
  - Do not be afraid to discuss issues/ask questions
    - OK to admit you don't know how safety aspects apply
    - Sometimes "N/A" is the right answer
    - Others have experience and can help

Customer is your partner

Framework

- Create baseline work products
  - Ensure requirements are clear and testable
  - Ensure requirements are properly allocated
    - Systems, Software, Firmware, Electrical, Reliability, etc.
  - Ensure requirements assigned Safety Critical[S-C] / Safety Related [S-R] “Safety Rating Tags” (SRTs)
    - Absolutely necessary and critical
  - Review traceability and compliance matrices
  - Ensure safety requirements are traceable to code level

Safety Rating Tags absolutely necessary and critical

Framework

- Create baseline design
  - Design with testing in mind
    - Need to prove requirements are not only implemented, but are implemented correctly
  - Isolate [S-C]/[S-R] functionality using separate source code files
    - Design should consider partitioning
  - Eliminate unnecessary features from reused code
    - Irrelevant legacy functionality, obsolete/outdated debug services, etc.

Design with safety and testing in mind

Framework

- Implement design
  - Generate source code files
    - Isolate [S-C]/[S-R] functionality using separate source code files
    - Embed Software Requirement IDs and SRTs directly into source code where requirement is met
      - File headers = Good, function/procedure headers = Better, source code block = Best
      - Provides obvious requirements traceability
      - Easily determine how and where requirements implemented
  - Perform regular static code analysis checks
  - Perform design and code Peer Reviews

Generate software with safety in mind

Framework

- Test
  - Generate test cases
    - Separate test cases for [S-C]/[S-R] code
    - Test cases must be traced to every requirement (i.e., must have a Requirements Traceability Matrix)
    - Include GO paths, NO GO paths, nominal, off nominal, in limits, out of limits, and duration/stress conditions
    - Automated test tools/test sets are best
  - Create code coverage analysis
    - Code Inspections may be necessary

Test requirements apply equally to new code and reuse code

Framework

- Safety Board/Joint Services Review Board Presentations
  - Update on a regular basis
  - Expect recommendations/actions
    - These are good things!
  - Present Safety Compliance Assessment results
  - Allow time to assemble Technical Data Package (TDP)

Communicate with Safety Board on a regular basis

Framework Review

- ✓ Create plan
- ✓ Form Safety Compliance Workgroups (SCWGs)
- ✓ Convene customer Technical Interchange Meetings
- ✓ Create baseline work products
- ✓ Create design
- ✓ Implement design
- ✓ Test
- ✓ Safety Board Presentations

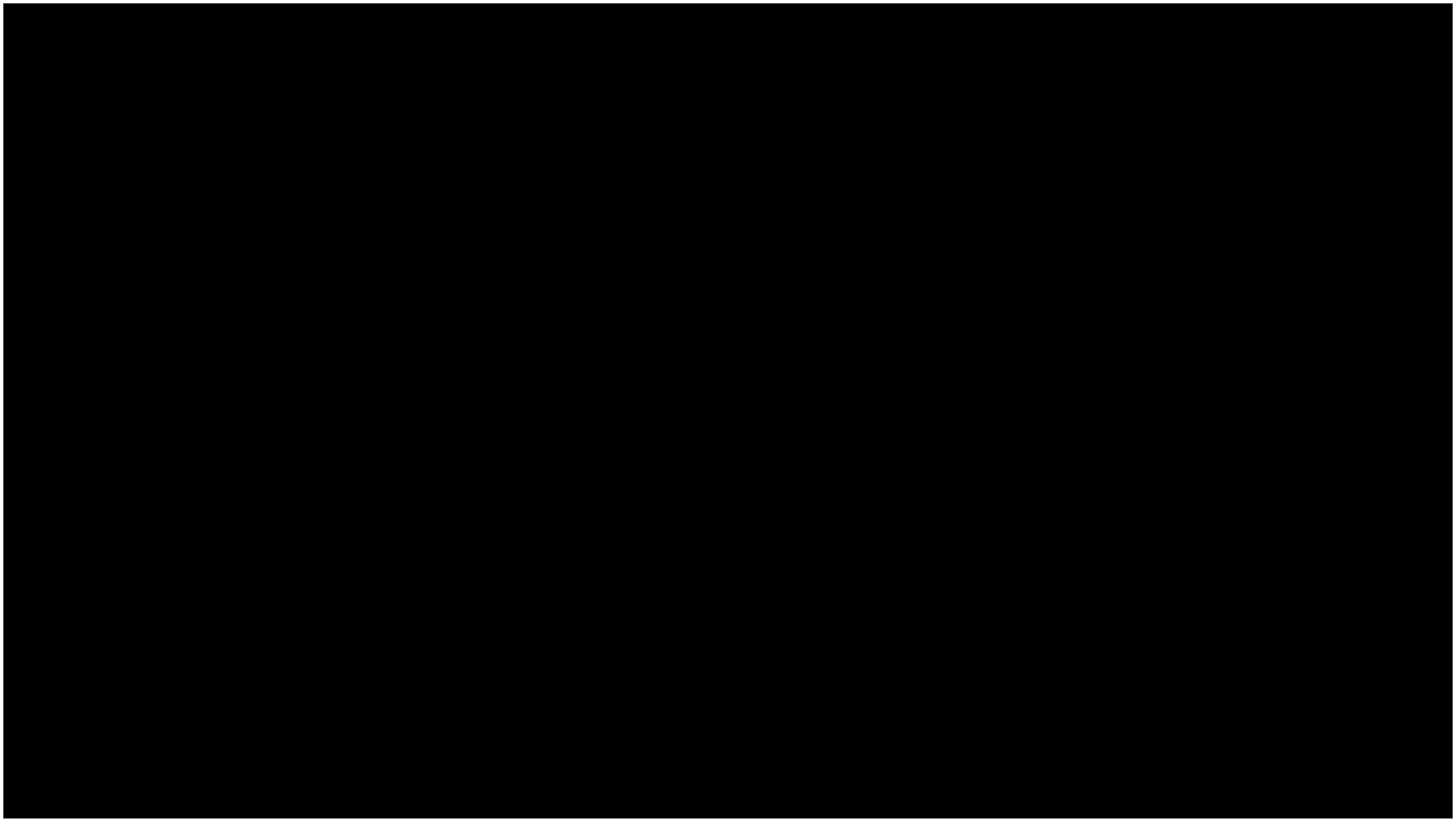
Standard engineering practices with Safety interlaced throughout entire process

Lessons Learned

- Modify standard framework
- Follow basic design practices with Safety from start
  - Always better to understand what is required up front
    - Understand current safety guidelines/requirements
    - Things change over time
  - Reuse isn't free
    - Don't over estimate "savings"
    - Remember to assess technical "debt"
- Include Safety in all phases
  - Do not be afraid to interact with Safety Boards
  - Traceability is key (document, document, document!)

Ask questions! Safety and Safety Boards are assets!

Proof of Principal Testing



Questions

Contact Information

- Alison Joseph
  - alison.joseph@lmco.com
    Lockheed Martin Corporation
    Missiles and Fire Control
    5600 Sand Lake Road
    Mail Point 157
    Orlando, FL 32819
    Phone: 407.356.9654

- Tony Ponko
  - tony.m.ponko@lmco.com
    Lockheed Martin Corporation
    Missiles and Fire Control
    5600 Sand Lake Road
    Mail Point 205
    Orlando, FL 32819
    Phone: 407.356.9587