Beyond the Code: Lessons Learned in Software Reuse

GSAW

2 March 2011

Amanda Ragan
Mission Systems Manager

Copyright, Northrop Grumman Corporation, 2011
Beyond the Code: Lessons Learned in Software Reuse

• Software reuse example
  – From large existing program to a new program

• Software reuse success factors
  – Code Design
  – Process
  – Stakeholders
  – Standards

• Success achieved by planning for these factors

• 83.7% of final codebase was reuse
  – Leveraging embedded software not explicitly designed for reuse
Our Challenge: Achieve high reuse across two architectures
83.7% of final codebase was reuse
Reuse of Legacy Design Guidelines

- Message based Inter-task Communication Model

- No shared memory
  - All memory pre-allocated and locally managed (no dynamic objects)

- Keep it simple, stupid (KISS)
  - No Templates
  - No Multiple Inheritance

- Isolate HW and System dependencies
  - Wrappers for OS
  - APIs for all CSCs
Lessons Learned (1 of 2)

• Process
  – Establish process and enforce across lifecycle
  – Maintain full alignment between Legacy Engineering Review Board (ERB) and New project ERB
  – Provide incremental builds to HW Integration and Test to enable early identification of problems/disconnects
  – Strengthen FQT effort by
    • Advanced preparation and coordination key to success
      – Entry/Exit criteria, roles, expectations, rules of engagement
    • Conducting Dry Runs in parallel to software development
      – Protect schedule
      – Facilitate early identification of problems
Maintaining Common Reuse Alignment

Start

Conducts periodic reviews of **ALL** Legacy SW Common Reuse SPRs & CRs

SCCB

Applicable & No CR/SPR Exists?

Yes

Stop

No

Determine if there is a Legacy companion (CR/SPR)

**SPR:** Category 1-3a?

No

No

Document As Not Implemented

Yes

SW IPT Decision – take To ERB?

No

Engineering Review Board (ERB)

No

ERB Approves?

No

Directed Change

Write CR/SPR

Severity=<severity level>

Defect Type="Legacy SPR" or "Legacy CR"

Injected In="Reuse Legacy"

Detected In="N/A"

Plan and Implement

Yes

CCB Approves?

Yes

Yes

CCB (Change Control Board)

Proactively Focused on Maximizing Legacy Alignment Benefits

Copyright, Northrop Grumman Corporation, 2011
Lessons Learned (2 of 2)

• Stakeholders
  – Build on both product reuse and knowledgeable staff from Legacy program
  – Understand and negotiate early what constitutes acceptance with customer

• Standards
  – Develop requirements with verification in mind
  – Address Firmware early in order to prevent schedule and cost uppers
  – Document Assumptions to enable Program and Working Group visibility into key areas requiring resolution
Summary

• Software reuse success factors
  – Code Design
  – Process
  – Stakeholders
  – Standards

• Success achieved by planning for these factors

• 83.7% of final codebase was reuse
  – Leveraging embedded software not explicitly designed for reuse

Lessons can be applied to other reuse programs