Expanding Legacy Systems Using Model Driven Engineering (MDE)

Kevin Nguyen & Billy Smith
Software Engineers
Overview

• Who we are

• MDE Process

• Results

• Challenges of expanding a legacy software system using MDE

• Lessons Learned
Northrop Grumman Products
## Model Driven Architecture Development Highlights

<table>
<thead>
<tr>
<th>Design Phase</th>
<th>Purpose</th>
<th>Collaboration</th>
</tr>
</thead>
<tbody>
<tr>
<td>System architecture</td>
<td>Understand customer’s requirements</td>
<td>- System engineers work with customers to understand their requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Start a concept model to support the effort of breaking the system requirements into hardware, and software requirements</td>
</tr>
<tr>
<td>CSCI architecture</td>
<td>Create architecture of hardware and software</td>
<td>- System engineers work with senior architects and test engineers to convey knowledge as well as clarify any misunderstandings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Refine/derive requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Develop test plan</td>
</tr>
<tr>
<td>CSC architecture</td>
<td>Expand architecture to software components</td>
<td>- Senior architects work with subject matter experts (SME) to further break the architecture down into CSCs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Generate ICDs and data information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Refine test plan</td>
</tr>
<tr>
<td>Detailed Design</td>
<td>Develop software units</td>
<td>- SMEs work with software engineers to design and develop software units</td>
</tr>
</tbody>
</table>
Design Phase Lifecycle

- Each phase of architecture development follows the same basic procedures
- Output from one phase becomes input for the next phase
  - Iterative process helps to catch design deficiency early

- Customers get the opportunity to participate in design and development efforts
- Architecture artifacts are developed using UML, making them easy for all parties to understand
Requirements Elicitation

• Inputs are from customers, users, and management
Requirements Allocation and CSCI Interfaces

Decompose System Requirements to CSCI Level
Computer Software Component (CSC) Interfaces
Computer Software Unit (CSU) Detailed Design

Drive Changes to Software Units

CXFORM package is downloaded from http://nssdcftp.gsfc.nasa.gov/selected_software/coordinate_transform/

Approved for Public Release; NGAS 14-0819, 4/28/14
CXFORM package is downloaded from http://nssdcftp.gsfc.nasa.gov/selected_software/coordinate_transform/
Results

- Passed PDRs and CDRs with full award fee and minimal customer Request for Information (RFI)

- Created a framework for collaboration and open communication between system, software, test, architects, and customers

- Minimized software integration issues

- Increased software testability

- Obtained the benefits of object oriented design and analysis for new architecture built on top of a non-OO system
  - Reusability, testability, maintainability
  - Highly cohesive, low coupling
Challenges of Adapting MDE to Legacy System

• Acquiring new tools and training
  – UML tools, process training

• Integrating the MDE process into existing tools and processes
  – Source control, requirements management, in-house tools

• Gaining full commitment and support from leadership
  – Selling the long term benefits of using MDE

• Changing the existing culture
  – Removing the engineering group stovepipe
  – Partitioning legacy software components
  – Stubborn engineers
Lessons Learned

- Allocate enough time for design
  - Spend more time on algorithm analysis
  - Perform more trade analysis

- Prototype often

- Fully flesh out interfaces early on

- Revisit and evaluate use cases at every design level
  - Make sure critical design elements are not eclipsed or forgotten

- Do not underestimate the effort of changing the existing culture
THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN