The New Frontier for Software-Based Product Lines

James Cezo
Principal Member Engineering Staff, Lockheed Martin

Charles Krueger, PhD
CEO, BigLever Software
Agenda

• Joint presentation by Lockheed Martin and BigLever Software

• Achieving a mission critical business and engineering objective at Lockheed Martin, by taking a new perspective

• Software Product Lines – a new frontier for innovation in product line engineering
The New Frontier for Innovation in Software-based Product Line Engineering

Competitive Advantage ($)

Discontinuous Competitive Jump

New Frontier for Software-based Product Lines

Time

Competition
Lockheed Martin’s “Common Product Line” Initiative
A Goal at Lockheed Martin
Maritime Systems and Sensors

“At Lockheed Martin, the timely and cost-effective delivery of the latest technological advances to our customers is mission critical. Our goal is to constantly ‘push the envelope’ in employing state-of-the-art product development tools and methods.”

– Norman Malnak, Chief Engineer & Vice President of Technical Operations
Lockheed Martin Maritime Systems & Sensors (LM MS2)
Common Product Line Initiative at LM MS2

• In pursuit of this core business objective
  - Finalizing a multi-year initiative to identify, evaluate and implement a leading-edge Common Product Line reuse strategy
  - Streamline systems and software engineering in each of its core product lines.

• Driving goal is to satisfy strong customer demand to reduce the time, cost and effort required to create, deploy and maintain products
In Search of a Solution

• The solution must ultimately
  - Minimize duplicate effort and maximize commonality among design and implementation assets
  - Optimize reuse of effort across similar products in our product lines

• Emerging discipline referred to as Software Product Lines (SPL) offered great promise

• Recent breakthroughs in the SPL field enabled the advanced capabilities desired by LM MS2
  - Strategic reuse of engineering assets across the entire systems and embedded software development lifecycle
  - Spanning requirements, architecture, design, source code, documentation and test artifacts
Software Product Line Engineering
Software-based Product Lines?

• In the world of hard goods, a **product line** refers to variations on a common theme
  - Multiple products combined into one “line”, offering different **features**
  - Addressing **diversity in customer needs** for a particular kind of item

• **Economy of scale** is key
  - Greater profitability achieved when common product or manufacturing assets can be used for different “flavors” of a product

• Today, **product differentiation** relies heavily on **software**

• **Software Product Line** engineering is a growing trend
  - Helping organizations better achieve **product diversity**
  - With the **speed and efficiency** it takes to satisfy today’s cost-conscious “want it new, want it now” consumers and businesses
Why Software Product Line Engineering?

• Compared to product-centric development approaches, Software Product Line (SPL) engineering allows you to:
  - Better capitalize on **strategic reuse** of commonality across a portfolio
  - More efficiently **minimize the complexity of managing variation** among products
  - Take a more **holistic approach for product line delivery**, across the full systems and software development lifecycle
3 Keys to SPL Success at Lockheed Martin

• A shift in perspective
  - “The right point of view saves 20 points of IQ”
• The SPL Lifecycle Framework
• Incremental Transition to SPL Practice
A Shift in Perspective
Shift in Perspective from Product-centric to Product Line
– Before –

Order $N^2$ Complexity

“Vertical” 
Product 
Perspective
Shift in Perspective from Product-centric to Product Line

– After –

Simplicity of a Single System

“Horizontal” Core Asset Perspective

Reusable Core Assets

| Product A |
|---|---|---|
| Requirements | Design Models | Source Code |
| Test Cases |

| Product B |
|---|---|---|
| Requirements | Design Models | Source Code |
| Test Cases |

| Product N |
|---|---|---|
| Requirements | Design Models | Source Code |
| Test Cases |

Product Marketing

Feature Profiles

Requirements Engineers

Architects

Developers

Quality Assurance

Gears

Product Configurator

Source Code

Architects

Requirements Engineers

Test Cases

Requirements Engineers

Test Cases

Requirements Engineers

Test Cases
Engineer Your Product Line as a Single System rather than a Multitude of Products
Software Product Line Lifecycle Framework
SPL Lifecycle Framework
Supporting the Full Systems & Software Development Lifecycle

Example LM Tools

- DOORS
- Rhapsody
- ClearCase
- Eclipse
- Custom Local Test System
- OS File System
- Bridge for Local Test System

Product Configurator

- Feature Profiles
- Product A
- Production Line Development Environment

Example Assets

- Requirements
- Design Models
- Source Code
- User Docs
- Test Cases

Example LM Tools

- DOORS/ Gears Bridge
- Rhapsody/ Gears Bridge
- Universal CM Bridge
- Eclipse Plugin
- Bridge for Local Test System

Product A

Product N
SPL Lifecycle Framework Solutions from BigLever Software, Telelogic and Rational

- **BigLever Software Gears** SPL Framework
  - Software product line engineering tool & framework
  - Powerful patent-pending technology
  - Award winning

- **Bridge integrations**
  - Rhapsody/Gears Bridge
  - DOORS/Gears Bridge
  - Eclipse/Gears Plugin
  - Universal Configuration Management Bridge (including full support for Synergy and ClearCase)
  - ...

Ready for IBM Rational software
Incremental Transition to SPL Practice
Getting Started with SPL Practice

- Start with what you’ve got
- Use an *incremental transition* strategy
  - Think big. Start small.
  - Incremental return on incremental investment
- Create a *pilot as the catalyst* for change
- Stage transition to be *non-disruptive* to production schedules
- Utilize best practices in the *3-Tiered SPL Methodology*
3-Tiered SPL Methodology

Capabilities and benefits in each tier enable the capabilities and benefits at higher tiers

Leverage. Feature Based Portfolio Evolution
Business-wide management of portfolio by features rather than by products leads to optimized:
Scalability and Time-to-Market

Simplify. Core Asset Focused Development
High levels of reuse, deep asset expertise, stable organization structure from focus on single system rather than multitude of products leads to optimized:
Quality

Consolidate. Variation Management & Automated Production
Applying SPL Lifecycle Framework to eliminate duplication, divergence, merging, ad hoc variation techniques, lifecycle silos, and manual production leads to optimized:
Productivity and Cost
Insights from Lockheed Martin’s Experience
Insights from Lockheed Martin's Practical Experience with SPL Approach

- Insights on the SPL shift in perspective
- Leveraging the SPL Lifecycle Framework
- Experiences with Incremental Transition to SPL Practice
Conclusions
Lockheed’s Business Rationale for Adopting SPL Engineering

| Available Benefits |  
|-------------------|---|
| Low               | High |

<table>
<thead>
<tr>
<th>Transition Time, Cost &amp; Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return-on-Investment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slow</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cost of Doing Nothing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
</tr>
</tbody>
</table>
Thank You!

--------

Questions and Discussion?