Using the UML and Object Oriented Programming Techniques to Create a Lean Software Environment

By Antar A Spearmon

SELIP 695: Integrative Project

5/7/2008

Agenda

- The Problem
- Options Currently Available
- The Lean Software Approach
- The Object-Oriented Programming Paradigm
- The Unified Modeling Language
- Lean Thinking
- The Benefits
- A Brief Example
- Wrap-up
- Questions

5/7/2008

Software is Everywhere

- From Military Applications to Children’s Toys
- Market Demand Drives Increase in System Complexity
  - Automobile Industry Example:
    - Safety
    - Comfort
    - Performance
  - Leverage The Power of The Computer to Augment Mechanical Systems

5/7/2008

Increasing System Complexity

- Systems of Systems
  - Complex Systems Made up of Smaller Sub-Systems
  - Require Coordinated Efforts of Tens to Thousands of People From Various Backgrounds
- As the Role of Software Increases, Customers Demand:
  - Robust Quality Software
  - Reduced Development Cycles

5/7/2008
The Tale of Two Online Courses...
The Problem

- RISK
  - Inaccurate Understanding of End-User Needs
  - Constantly Changing Requirements
  - Late Discovery of Serious Projects Flaws
  - Schedule slips
  - Poor Communication
  - Etc.

Some Numbers

- According to the Gartner Group:
  - 74% of all IT/Software projects fail
    - Over Budget
    - Over Schedule
  - 28% Fail Altogether

Shared Characteristics of Software Failures

- Ad-hoc Requirements Management
- Brittle Architectures
- Overwhelming Complexity
- Modules That Don't Fit Together
- Software That's hard to maintain or extend
- Team members in each other's way
- Poor Communication

The Cause

- Traditional Programming Paradigms Lack the Scalability and Flexibility Required to Effectively Manage the Increasing Complexity of Current Systems

The Need

- What are Required are New Paradigms to Improve Programmer Productivity to Effectively Meet These Challenging Demands
The Problem

- RISK
  - Inaccurate Understanding of End-User Needs
  - Constantly Changing Requirements
  - Late Discovery of Serious Projects Flaws
  - Schedule slippage
  - Poor Communication
  - Etc.

Some Numbers

- According to the Gartner Group:
  - 74% of all IT/Software projects fail
    - Over Budget
    - Over Schedule
    - 28% Fail Altogether

Shared Characteristics of Software Failures

- Ad-hoc Requirements Management
- Brittle Architectures
- Overwhelming Complexity
- Modules That Don't Fit Together
- Software That's hard to maintain or extend
- Team members in each other's way
- Poor Communication

The Cause

- Traditional Programming Paradigms Lack the Scalability and Flexibility Required to Effectively Manage the Increasing Complexity of Current Systems

The Need

- What are Required are New Paradigms to Improve Programmer Productivity to Effectively Meet These Challenging Demands
Three Schools of Thought - Coding

- "Cowboy" Coding
  - Shoot From the Hip Approach to Software Development

- Task-Oriented
  - Traditional "Structured" Approach to Programming
  - Collection of Logical Functions

- Object-Oriented
  - Focus on Objects and How They Relate to Each Other

Agile Software Development Paradigm

- Focus on Rapid Continuous Delivery of Working Software

- Incremental Software Development Approach

- Centered on Self-Organizing Teams

- Constant Communication Between Business People and Developers

- Lacks Structure, Detailed Requirements Analysis, and Much Needed Documentation
The Proposal

Lean Thinking
The Unified Modeling Language
Object Oriented Programming
The Systems Engineering Process

LEAN SOFTWARE DEVELOPMENT

The Lean Software Approach

The Lean Software Approach

The Object-Oriented Programming Paradigm

The Tale of Two Online Courses...
The UML

- The UML is comprised of a set of diagrams used to form a model of the system to be created
- Expands understanding of the system under development and how it is to work
- As a Blueprint is to a house, The UML Diagrams are to Software
- Language Independent

Lean Thinking

- Do more with less by placing an emphasis on waste minimization and flexibility in the development process

Benefits of Lean

- Enterprise Improvements (Manufacturing):
  - Waste reduction by 80%
  - Production cost reduction by 50%
  - Manufacturing cycle times decreased by 50%
  - Inventory reduction by 80% while increasing customer service levels
- Goal is to leverage these same benefits in the software environment by applying the 5 principles of lean
The Five Principles of Lean

- Specify value
- Identify all the steps in the value stream
- Make the value flow
- Let customers pull value
- Pursue perfection through continuous improvement

The Benefits of Lean Software Development

- A formalized approach to solving complex problems and ensuring the realization of successful software systems
- Considers the entire life-cycle
- From Requirements to Future Upgrades
- Business and technical aspects of system development
- Combines SE with LEAN, UML, and OOP

The Benefits

- Cost savings yield:
  - Increased throughput
  - Improved cash-flows
  - Higher quality
  - Increased market response
  - Repeat business
  - Enterprise growth
  - Increased profits
Two Ways to Increase Profits: Increase sales or cut costs

- In the face of fierce competition, price is nearly fixed, thereby limiting your options

<table>
<thead>
<tr>
<th>Enterprise Impact – Strengthen Strategic Advantage</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="graph.png" alt="Graph showing profit increase with price and cost adjustments" /></td>
</tr>
</tbody>
</table>

- Quality robust code that is reusable
- Fully Tested
- Tried & True
- No more re-inventing the wheel
- Extensible
- Continued success by leveraging past innovations
- More responsive to constantly changing customer needs

NGC 7’x10’ Low-Speed Wind Tunnel

A Brief Example...
Statement of Work

- Current System
  - Combination of 1950's and 1970's electronics
  - Analog System
  - Manually operated
  - Decreasing reliability
  - Limited availability of spares

- Proposed System
  - Updated Real-Time Controller System
  - Digital Based
  - Automatic Capabilities
  - User Touch Screen
  - Increased System Visibility

Requirements

- Upgrade the Model Control System (MCS)

From Blank Sheet to Architecture

- Constant Communication Required
- Original Designers Have Long Retired
- Operators/Technicians Primary Requirements Source
  - Test Engineers Have Input But Know Far Less Than Techs Regarding System Operation
  - Useful for Technical Specs
  - Have Operators Test Early Iterations

Use of UML & OOP

- Aggressive Budget & Schedule Constraints
- Needed Quick Effective Tool to Generate Requirements
  - Used Minimal Set of UML Diagrams
- Small Software Team (2.5 Members)
  - 1 Cowboy
  - 1 OOPer
  - .5 Impressionable Newbie
- Needed Way to Have 3 Programmers Work on 1 System
OOP Is Given Due to Complexity

- Because Requirements Were Self-Generated
  - Software Must Be:
    - Flexible
    - Extensible
    - Reliable
    - Robust
  - OOP Paradigm Became Essential For Project Success
- Code Reuse to Shorten Design Time

The Tale of Two Online Courses...
The Tale of Two Online Courses...

Reasons for Lean Software Approach

- Because of limited knowledge of current control system functionality
- Needed to focus on objects & their interactions with a systems view
- Full reanalysis required for traditional coding approach
- Forced to do more with less
- Less resources, budget, schedule
- Produce effective system solution under fire

Wrap-Up

- Software is everywhere
- With growing complexity, increased need for systematic approach to ensure delivery of successful software systems
- The UML, OOP, Lean Thinking, and the Systems Engineering Process
- Coupled together to ensure successful delivery of robust, high quality software components that amply meet customer needs while reducing overall software life-cycle costs
Questions?

Resources

- www.defense-aerospace.com/laterfiles/commerciapiecefighterCostFinal-Jul06.pdf
- www.opr.y.us/osp-tech/computer-on-wheels/
- www.incose.org.uk/Downloads/AA04%20SE%20trends.ppt
- Lean Methods, 2006 Rodney W. Oppeheim, LMU, Los Angeles
- www.incose.org/practice/techactivities/wg/leansewg/docs/LeanSEWorkingGroupINC0SE_Jan2006WorkshopAZ.ppt
- http://java.sun.com/docs/books/tutorial/java/concepts/object.html
- http://www.lean.org/WhatsLean/Principles.cfm
- www.incose.org.uk/Downloads/AA04%20SE%20trends.ppt