ITEC420: Software Engineering
Lecture 7: Recap &
Project & Risk Management

Box Leangsuksun
SWECO Endowed Professor,
Computer Science
Louisiana Tech University
box@latech.edu

CTO, PB Tech International Inc.
naibox@gmail.com

Recap:

Towards creating survivable architecture
## Product Life cycle (RUP)

### Workflows
- Business Modeling
- Requirements
- Analysis & Design
- Implementation
- Test
- Deployment
- Configuration & Change Mgmt
- Project Management
- Environment

### Phases

<table>
<thead>
<tr>
<th>Phases</th>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Initial</td>
<td>Elab #1</td>
<td>Const #1</td>
<td>Const #N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Elab #2</td>
<td>Const #2</td>
<td>Tran #1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Const #N</td>
<td>Tran #2</td>
</tr>
</tbody>
</table>

### View of Software Engineering

- User Needs/Inputs – Specification or Requirement
- Technical: Design/Analysis Implementation
- Project/Product Management:
- Manufacturing/Quality Control
- Marketing
- Sale
Typical SE life cycle process

- **User Inputs -> Requirements and Specification**
- **Design & Analysis -> Analysis & Design documents**
- **Implementation -> Programming, integration, Tools -> Code or system**
- **Testing -> Test Plan & Test results**
- **Manufacturing, Installation/Deployment and Quality Control -> Change Management & Configuration Management.**
- A classic water model. Not good..why?
- the above SE workflow can be divided into smaller iterations

Supporting Workflow Overview
Objectives: Supporting Workflows

- Define the supporting workflows
- Understand how to apply the supporting workflows
- Understand the activities necessary to configure a process for a project
- Comprehend the basic concepts of configuration and change management
- Understand the context of Project Management within the scope of a project

INTRODUCTION
Supporting Workflows in Context

Process vs Supporting Workflow

- Process Workflow
  - Development activities
  - Stronger emphasis in specific phases

- Supporting workflow
  - Management and infrastructure activities
  - Equal emphasis in all phases
Supporting Workflows in Context

<table>
<thead>
<tr>
<th>Workflows</th>
<th>Phases</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Modeling</td>
<td>Inception</td>
<td>Initial</td>
</tr>
<tr>
<td>Requirements</td>
<td>Elaboration</td>
<td>Elab #1</td>
</tr>
<tr>
<td>Analysis &amp; Design</td>
<td>Construction</td>
<td>Elab #2</td>
</tr>
<tr>
<td>Implementation</td>
<td>Transition</td>
<td>Const #1</td>
</tr>
<tr>
<td>Test</td>
<td></td>
<td>Const #2</td>
</tr>
<tr>
<td>Deployment</td>
<td></td>
<td>Const #N</td>
</tr>
<tr>
<td>Configuration &amp; Change Mgmt</td>
<td></td>
<td>Tran #1</td>
</tr>
<tr>
<td>Project Management</td>
<td></td>
<td>Tran #2</td>
</tr>
<tr>
<td>Environment</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Environment Workflow
Objectives: Environment Workflow

- Focuses on the activities necessary to configure the process for a project
- Describe the activities required to develop the guidelines in support of a project
- Provide the software organization with the processes AND tools that will support the development team
Towards creating survivable architecture
Process Configuration

- Two levels at which the software engineering process can be configured
  - Organization Level
  - Project level

Developing Guidelines

- Guidelines are captured at the beginning of the process and also as the project progresses
- guidelines for...
  - Business modeling
  - Use Case modeling
  - Design modeling
  - Programming
  - Testing
Towards creating survivable architecture
Definition: Configuration & Change Management

- Configuration and Change Request Management control change to, and maintain the integrity of, a project’s artifacts
  - (SEI CMM)
  - Configuration of your project – toward software construction/build
  - Control and manage new features/enhancements as well as defects

Configuration & Change Management Artifacts

![Configuration & Change Management Artifacts Diagram]
Configuration & Change Management Overview

The Major Aspects of a CM System

- Change Request Management
- Configuration Management
- Change Tracking
- Version Selection
- Software Manufacture
**Change Request Management**

- Provide a mechanism to associate any modifications to your project
- Provide multiple interfaces for mixed platform support
- Allow for easy submission from any member of the team
- Flexible customization of request lifecycle
- Understanding of assignments and priorities

---

**Change Request Management**

- Metrics and reporting to assess quality of product
- Easy to access to project status information
- Understanding of resource allocation
- Integration with CM for full traceability
Configuration Management (CM) Plan

- Roles & Responsibilities
- Tools
- Environment, Infrastructure
- Identification methods
- Change Control Board
- Change Request Processing and Approval
- Configuration Status Accounting

Creating the CM Report

- Use the MS Word template provided in the Rational Unified Process
Change Request

- Identification
- Problem Description
- Proposed Change
- Resolution
- Assessment
- Disposition

Automated SCM Repository (Functions and Tools)

- Versioning
- Dependency tracking
- Change management
- Requirements tracing
- Configuration management
- Audit trails

These slides are from Dr. Jay-Evan J. Tevis's software engineering lecture ppt
**Change Management in Software engineering**

- Quality states & the state diagram
- Examples
  - Initial/created
  - Assigned
  - Submitted (developer)
  - Unit tested
  - IT_passed, SVV_passed/final

**Quality Transition**

- Code submission to repository
- Assign to a developer
- Unit test Passed
- Intg Passed
- SVV Passed
- SV
Submitting a Change Request

- Submit a change request using ClearQuest (as example in IBM RUP tools)

Change Control Board

- Ensures all proposed changes receive technical analysis and review
- Approves all changes to baseline configuration items
- Prioritizes all approved changes
- Enforces the change request process
Project Management Workflow

Project Management Workflow in Context

<table>
<thead>
<tr>
<th>Workflows</th>
<th>Phases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Modeling</td>
<td>Inception</td>
</tr>
<tr>
<td>Requirements</td>
<td>Elaboration</td>
</tr>
<tr>
<td>Analysis &amp; Design</td>
<td>Construction</td>
</tr>
<tr>
<td>Implementation</td>
<td>Transition</td>
</tr>
<tr>
<td>Test</td>
<td></td>
</tr>
<tr>
<td>Deployment</td>
<td></td>
</tr>
<tr>
<td>Configuration &amp; Change Mgmt</td>
<td></td>
</tr>
<tr>
<td>Project Management</td>
<td></td>
</tr>
<tr>
<td>Environment</td>
<td></td>
</tr>
</tbody>
</table>

Iterations

Towards creating survivable architecture
Objectives: Project Management

• To provide a framework for managing software-intensive projects
• To provide practical guidelines for planning, staffing, executing and monitoring projects
• To provide a framework for managing risk

Objectives: Project Management

• NOT
  • Managing people
  • Managing budget
  • Managing contracts
Project Management Artifacts

Project Management Overview
Inception Phase

- **Evaluation Criteria:**
  - Stakeholder concurrence on scope definition and cost/schedule estimates
  - Requirements understanding for the primary use cases
  - Credibility of the estimates, priorities, risks, development process, etc.

- Depth and breadth of architectural prototype (optional)
- Actual expenditures versus planned expenditures
- Milestone: Lifecycle Objectives (LCO)
Elaboration Phase

• Evaluation Criteria:
  • Vision of the product is stable
  • Requirements understanding (use-case model $\geq 80\%$ complete and supplementary specifications complete)
  • Executable architecture baselined and stable
  • Major technical risks ‘retired’
  • Sufficient software development plan for construction

Elaboration Phase

• Stakeholder agreement that the vision can be met with the plan and architecture in place
• Actual expenditures versus planned expenditures
• Milestone: Lifecycle Architecture (LCA)
Construction Phase

- Evaluation Criteria:
  - Complete product of sufficient quality available
- Deployment plan ready
  - Packaging
  - Pricing
  - Roll-out
  - Support

Construction Phase

- Training
- Production
- Transition strategy
- User documentation available
- Milestone: Initial Operational Capability (IOC) “beta”
Transition Phase

- Evaluation Criteria:
  - Previous products and artifacts, updated as necessary
  - Customer/user acceptance of released product
  - Post-mortem analysis of the organization’s performance; additional assets; potential evolutions
  - Milestone: Product Release ("GA" or general availability release)

Major Milestones: Business Decision Points
Concept: Iteration

- One pass through a sequence of process workflows. From a development perspective the software lifecycle is a succession of iteration, through which the software develops incrementally.
Strategies for Iterative Development

Minor Milestones: Technical Visibility Points

<table>
<thead>
<tr>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preliminary Iteration</td>
<td>Architect Iteration</td>
<td>Architect Iteration</td>
<td>Devel Iteration</td>
</tr>
</tbody>
</table>

Minor Milestones: Releases
**Minor Milestones: Iteration Feedback Loop**

- **Iteration N Cost and Schedule Actuals**
  - Quality assessment for iteration N
    - Test results
    - Defect density
    - Architecture stability
    - Other metrics

- **Revision N Assessment**
  - Compare iteration actual cost, schedule and content with iteration plan
  - Determine what needs to be done
  - Assign to future iterations

- **Iteration N+1 Plan**
  - Use revised risk list and select

- **Revised Project Plan**
  - Overall schedule
  - Scope/Content

**Iteration Planning**

- Define Objective evaluation criteria
- Identify what concrete, measurable artifacts will be developed or updated and the activities required to build them
- Use a standard work breakdown structure
- Use estimates to assign duration and effort to each activity
- Adjust as necessary to keep all numbers within resource constraints
Towards creating survivable architecture

How many and How Long?

- How many iterations should be included in the project plan?
- How long should each iteration take?
- Depends on a number of factors:
  - Size of the system being built: The larger the system, the longer the duration
  - Number of people: The larger the number of people, longer the duration.

Iteration Scope

- The scope of an iteration is driven by four factors:
  - The functionality required of the system
  - The time allocated to the iteration in the Project Plan
  - The top risks to the project
  - The phase and its specific objectives
Risk Terms

- RISK- whatever may stand in the way of our success
- DIRECT RISK- the project has a large degree of control
- INDIRECT RISK- the project has little or no control
- Risk attributes:
  - Probability of occurrence
  - Impact on the project (severity)
- Risk magnitude indicator:
  - High, Significant, Moderate, Minor, Low

Risk Strategies

- Risk Avoidance- reorganize the project so that it will not be affected by the risk
- Risk Transfer- reorganize the project so that someone else bears the risk
- Risk Acceptance- live with it
- Risk Mitigation- reduce the probability or impact
- Definition of a contingency plan- what course of action to take if the risk becomes an actual problem (“Plan B”)
Checkpoints

• Does your team understand the process you will implement for the project?
• Have you set up a Change Control Board?
• Have you defined procedures for managing change?
• Do you know what your most important risks are?
• Have you decided how you will manage the project risk?