RUP Iteration Planning

By Masoud Moshref Javadi

Nov 2008
Outline

- Introduction
- When, What, How, Who
- Iteration Strategies
- Phases Iteration Planning
- Resources
Two Levels of Plan

- A coarse-grained plan: the phase plan
  - Dates of the major milestones
  - Staffing profile
  - Dates of the minor milestone

- A series of fine-grained plans: the iteration plans
“A time-sequenced set of activities and tasks, with assigned resources, containing task dependencies, artifacts, and evaluation criteria for the iteration”
When should I do that?

- Built in the preceding iteration (second half)
- Maintained in the current iteration
- Two “active” iteration plan at one time:
  - The current iteration plan
  - The next iteration plan
What do I need?

- The project plan
- The current status of the project
- Previous iteration plan assessment
- Software architecture
  - List of scenarios/usecases
  - List of major classes and packages
- List of changes/fixes for preceding artifacts
How?

- Understand the Objectives of the Phase
- Determine the Deliverable(s)
- Find the related activities
- Associate Resources with Activities
- Define Monitoring and Control Processes
- Assess Iteration
What’s its constituents?

- **Plan**
  - Activities
  - Schedule
  - Deliverables
- **Resources**
  - Staffing Resources
  - Financial Resources
  - Equipment & Facilities
- **Use Cases**
- **Evaluation Criteria**
Who?

- Project Manager
- Software Architect
- Management Reviewer
Iteration Strategies (1)

- Wide and Shallow
  - When is appropriate?
    - Inexperienced team (problem domain / technology area)
    - Consequential / unprecedented architecture
  - Potential pitfalls
    - Analysis paralysis
    - Delayed result
    - Not enough of the technical details to get a sense of the real technical risks
Deep and Narrow

When is appropriate?
- Early result ➔ Overcome dominant risk, garner support, or prove viability
- Continually evolving requirements

Potential pitfalls
- Incompatible components / stovepipe
- Large part of the functionality must be sampled to achieve a balance architecture in a completely new problem domain / an unprecedented architecture
## Hybrid approach

- Narrow/Deep strategy used in Inception
- Wide/Shallow strategy used in Inception
- Wide/Shallow strategy used in Elaboration
  - With selective Narrow/Deep focus
- Narrow/Deep strategy used in Construction
Objective Drivers

- Vision
- Business case & critical requirements
- Project management (feasibility & setting up the team)
- Development environment
Elaboration Phase Iteration Planning

- **Objectives Drivers**
  - **Risk**
    - For the most damaging risks, identify a scenario in one use case that would force the development team to "confront" the risk.
  - **Coverage**
    - Toward the end of the phase, include scenarios that touch areas you know will require development (baseline an architecture)
  - **Criticality**
    - make sure to include the most fundamental function or services provided
Objectives Drivers

- **Risk**
  - New and unsuspected risks are uncovered
  - Complete some of the more critical ones early

- **Completeness**
  - Toward the end of the construction phase, the main goal will be to ensure coverage of full use cases.
The main goal is to finish this generation of the product

Bugs/Fixes/Improvements
**Summery**

- **An iteration plan:**
  - provides a detailed description of the upcoming phase of work,
  - defines the worker roles involved, necessary activities, and artifacts to be delivered in that iteration,
  - outlines a very clear set of measurement criteria by which progress can be assessed during the iteration and success can be measured at the end
  - defines specific start/end dates and delivery dates
Resources