



 POLITECNICO DI MILANO

Dipartimento di  
Elettronica e Informazione

Planning and Managing Software Projects 2014-15  
Session 17

## Final Review Preview

**Emanuele Della Valle**  
<http://emanueledellavalle.org>

- This slides are largely based on Prof. John Musser class notes on “Principles of Software Project Management”
- Original slides are available at <http://www.projectreference.com/>
- Reuse and republish permission was granted

- Format: Similar to last one
  - Open questions
  - An exercise on Earned Value Analysis

- Interviews
- Document Analysis
- Brainstorming
- Requirements Workshops
- Prototyping
- Use Cases
- Storyboards
- Agile methods, e.g., SCRUM user stories

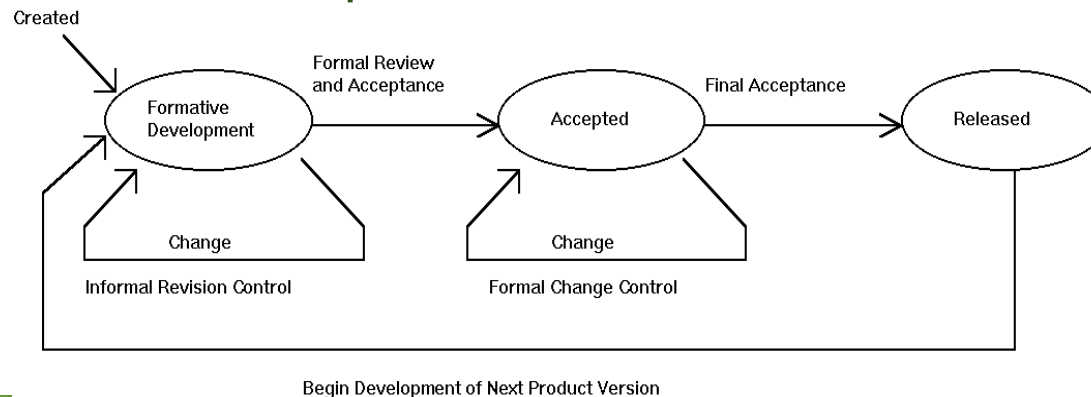
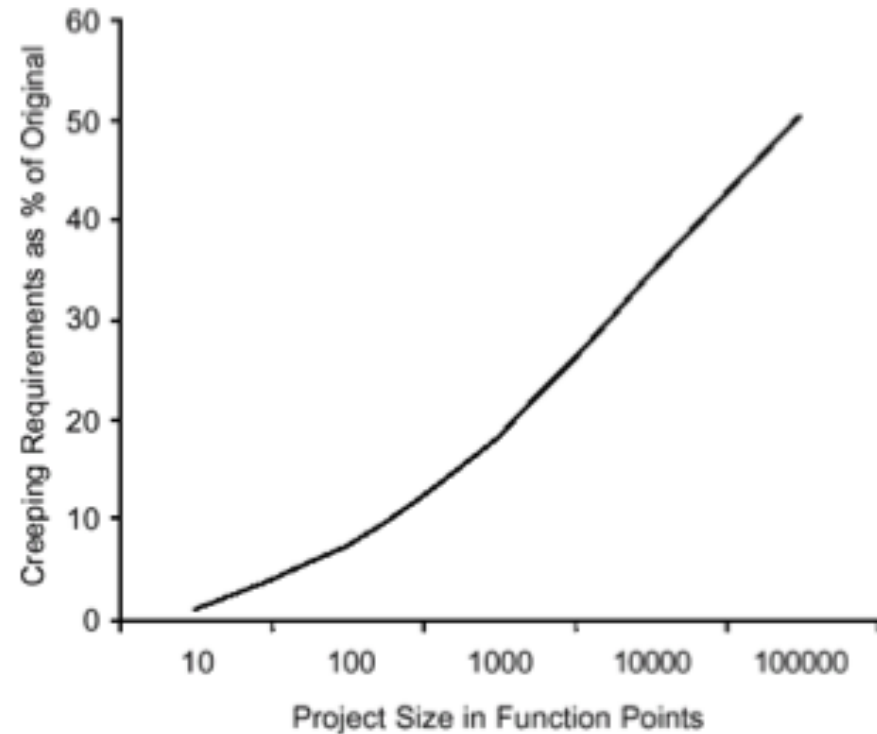
- Risk Management
  - Types of risk: schedule, cost, requirements
- Risk Identification
  - Involve the team
- Risk Analysis
  - Risk Exposure (RE = Prob. \* Size)
  - Probability is 15%, size is 10 weeks
    - $.15 * 10w = 1.5w$
- Risk Prioritization
  - 80-20 rule; large size or prob. 1st; grouping; ignoring

- Risk Control
  - Plan
- Risk Resolution (5 Types)
  - Avoidance (ex: scrub)
  - Assumption (just monitor)
  - Control (contingency)
  - Knowledge Acquisition (learn/buy/prototype)
  - Transfer (off project, team, critical path)
- Risk Monitoring
  - Top 10 Risk List (McConnell's example)
    - [http://www.construx.com/Thought\\_Leadership/Books/Survival\\_Guide/Resources\\_By\\_Chapter/Sample\\_Top\\_10\\_Risks\\_List/](http://www.construx.com/Thought_Leadership/Books/Survival_Guide/Resources_By_Chapter/Sample_Top_10_Risks_List/)

# Requirements gathering techniques

- Interviews
- Document Analysis
- Brainstorming
- Requirements Workshops
- Prototyping
- Use Cases
- Storyboards

- The feature creep phenomenon
  - Average project has 25% requirements change
- Sources of change
  - Marketing
  - Developers
  - Users
- Overly detailed specs. or prolonged requirements phase are not the answer
- Change control is a process





- Early Stages
  1. Minimal Specification
  2. Requirements Scrubbing
  3. Versioned Development
  
- Mid-Project
  - Effective change control
    - Set up a Change Control Board
      - Structure
      - Process
      - **Like a Triage !!!**
    - Adopt Configuration Management Tools
  
- Late-Project
  - Feature cuts

- Staffing
  - Typical profile
  - Roll-on – Roll-off
- Hiring Guidelines
- Team Models
- Resource Allocation Tools
  - RAM
  - Skill Matrix

## Hiring Guidelines

- Hire for attitude, train for skill
- Smart, gets things done
- Balance

- Start with objective
  - Problem resolution, creativity, tactical execution
- Decentralized vs. Centralized
- Large teams
  - Decompose via hierarchy, into optimal sizes
- Optimal size?
  - 4-6 developers

- Business team
  - Technical lead + team; most common
  - Can be strong or loose hierarchy
- Chief-programmer team
  - Surgical team; star at top; ego issues
- Skunkworks team
  - Off-site; pro: buy-in; con: minimal visibility
- SWAT team
  - Highly skilled/specialized; Ex: security team

| Team Model            | Problem Resolution | Creativity | Tactical Execution |
|-----------------------|--------------------|------------|--------------------|
| Business Team         | ***                | *          | **                 |
| Chief-Programmer Team |                    | ***        | **                 |
| “Skunkworks” Team     |                    | ***        |                    |
| SWAT Team             |                    |            | ***                |

## LEGEND

\*\*\* Best suited

\* Can be used

- Responsibility Assignment Matrix
  - Who does What
  - Be able to draw one
  
- Skills Matrix
  - Who has what skills
  - Be able to draw one

- Testing “Phases”
  - Unit
  - Integration
  - System
  - User Acceptance Testing
  
- Testing Types
  - Black-box
  - White-box
  
- Integration: 2 types
  - Top down
  - Bottom up



- Open Bugs (outstanding defects)
  - Ranked by severity
- Open Rates
  - How many new bugs over a period of time
- Close Rates
  - How many closed over that same period
  - Ex: 10 bugs/day
- Change Rate
  - Number of times the same issue updated
- Fix Failed Counts
  - Fixes that didn't really fix (still open)
  - One measure of “vibration” in project

- BCWS
- BCWP
  - Earned value
- ACWP
- Variances
  - $CV (BCWP - BCWS)$ ,  $SV (BCWP - ACWP)$
- Ratios
  - $SPI (BCWP / BCWS)$ ,  $CPI (BCWP / ACWP)$
  - $CR (SPI \times CPI)$
- Benefits
  - Consistency, forecasting, early warning

- Migration Strategies
  1. Flash Cut
    - A. Immediate Replacement
    - B. Parallel Operation
  2. Staged
    - One part at a time

- Migration Plan
- Importance of 2-way communication
  - Find-out customer's key dates
- Minimize intrusiveness
- Back-out Plan
- Data Conversion

- Roll-Out
  - Release Check-List
- Training
  - More than just end-users
    - Users, systems ops, maintenance developers, sales
- Documentation
  - Many types: End-user, sales & marketing, operations, design

- 3 Approaches
  1. Cut the size of the software
  2. Increase process productivity
  3. Slip the schedule, proceed with damage control
- People Steps
  - Morale; focus; re-assign
- Process Steps
  - Fix classic mistakes; mini-milestones
- Product Steps
  - Stabilize; trim features; take out the garbage

- Focused on process not people
- Steps
  - Prepare survey form
  - Email team with survey and schedule meeting
    - Gather data
  - Conduct meeting
  - Prepare PPR report

- Capability Maturity Model
- Five levels
  - Initial
  - Repeatable
  - Defined
  - Managed
  - Optimizing
- Know the functions in each level
  - (especially in the first 2)



# Questions?