A Tentative Framework for Managing Software Product Development in Small Companies

Kristian Rautiainen, Casper Lassenius, Jarno Vähäniitty, Maaret Pyhäjärvi and Jari Vanhanen

Helsinki University of Technology
Software Business and Engineering Institute

http://www.soberit.hut.fi/sems/

Presentation Outline

- Background
- Approach
- The Four Cycles of Control Framework
- Lessons Learned
Background

Problems
- There are many SME’s that struggle with the quality of their software products
- Money and time is lost when products are re-developed from scratch or patched until the code is degenerated
- Many good business ideas can fail through poor quality of execution
- Still, SME’s find it hard to allocate resources for improving their software engineering activities

Existing solutions
- No holistic approach exists for managing the software engineering activities of SME’s
- Business aspects are not considered together with product development aspects
- Process-centered approaches do not take into account the differences between project types, for instance,
  - Requirements-driven projects
  - Schedule-driven projects
- CMM, for example, is too heavy for SME’s and best suited for limited types of software engineering

Approach

Our framework is based on:
- Our previous research on improving the controllability of product development
- A literature study on
  - SW process models
  - SW business models
  - Strategic management of NPD
- Interviews, discussions and observations at the participating companies
  - Implementation experience from one case

Focus
- SME’s
- Software Product Business
- Mass-market products
  - Degree of customer-specific tailoring small
Purpose of the Four Cycles of Control Framework

- Combines business management and software product development
- Provides both a long-term and short-term view to software product release management
  - Release role, timing and content
  - ...
  - Daily or weekly builds within release projects
- Provides a common language and understanding of the way SW development is organized and the structure of the product(s)
- Brings a degree of control into software product development and at the same time accommodates faster response to change

Activities that span all of the cycles

Level of detail and emphasis vary in each cycle

Connecting business management to product development
- Release and development schedules
- Release contents and type
- Technology decisions
- Resourcing
- Continuous

Managing the individual release projects
- Number of iterations
- Iteration content and schedule
- Detailed resourcing
- 3-12 months

Managing the individual iterations
- Iteration task planning
- Mini-milestone content and schedule
- Stabilization of product
- Enables early feedback
- 1-3 months

Producing the actual product
- Frequent integration of code
- Development status check
- Daily or weekly builds

Product and technology roadmapping

Strategy
- Release Project Management
- Iteration Management
- Mini-milestones

Product Development Process
- Requirements Engineering
- Configuration Management
- Planning and Monitoring
- Verification and Validation
- ...
An Example: Microsoft’s Synch-and-Stabilize

- Product vision
- Outline specification

- Feature teams
- Iteration plan
- More detailed spec.

- Daily build
- Daily test

- Project status
- Final product
- Final specifications

- Synchronized product
- Development status

- Stabilized product
- Ready for alpha or beta testing and feedback

Different Types of Projects

- Roles and resourcing
- # and duration of cycles
- Communication patterns
- Decision-making rights
- ...

Implementation details vary between different types of projects
Lessons Learned

- Our framework forces management to consider business and development aspects at the same time
  - Helps make informed decisions
  - Prevents falling into a short-term reactive mode
- Prioritization is a must with scarce resources
  - There is never enough time to do everything
- Visualising the long-range plan with a roadmap communicates the priorities and helps in decision-making about e.g. resource allocation trade-offs
- The importance of a common language cannot be over-emphasized
  - Process and Product
- Suggestions for paths of improvement should be provided to aid and motivate improvement efforts

Questions and Comments