SEMS - Software Engineering Management System for Small and Medium Software Product Businesses

Goals, Means & Experiences

20.1.2004

Jarno Vähäniitty
jarno.vahaniitty@soberit.hut.fi
http://www.soberit.hut.fi/sems/

Agenda

- Background & Goals
- Means (or, our results)
- Experiences
Background & Goals

The SEMS Research Project

- Funded by Tekes and the participating companies
  - 3rd season
    - Avain Technologies
    - Mermit Business Applications
    - IPSS Oy
    - oPLAYo
    - Softatest
    - Napa Oy
  - 2nd season
    - Avain Technologies
    - QPR Software
    - Bluegiga Technologies
    - oPLAYo
    - Softatest
    - Smilehouse
  - 1st season
    - Arrak Software
    - Popsystems
    - Smartner Information Systems
    - Smilehouse

- Goal: To Develop a Business-Driven Approach for Software Process Improvement in Small Software Product Businesses
Background – Problems with Existing Work

- Most software companies are small (<50)
  - Largely overlooked in literature
  - Still, need effective software engineering practices tailored to type of business and company size
- Most existing process (improvement) models have their background in project-based SW development and originate from large company context
- Smaller companies find it hard to use existing software process improvement models and standards
  - Lack of resources (e.g. dedicated process people)
  - Fear of excessive process overhead
  - Practices designed for large businesses do not scale down easily
- Agile methodologies useful
  - However, link to business decision models for productisation (like the Stage-Gate™) or long-term product and technology planning missing

Background – Modern Approaches to Managing Software Development

- Most "modern" software process models are iterative and incremental in nature
  - The product is “grown” in small steps
  - Strengths: fast feedback, early delivery, customer satisfaction
  - Weaknesses: need for experienced people, management overhead
- Agile methodologies are the most recent
  - Suitable for
    - Small teams
    - Situations with lots of uncertainty and change
    - Face-to-face communication emphasised over documentation
- As well as for the "traditional" models, the link from models to business decision models for productisation (like the Stage-Gate™) is missing in literature
Background – Characteristics of the Software Product Business

- Based on releases of different types
  - Major, minor, maintenance, ...
- Product quality must be good (enough)
  - Otherwise, scarce resources may be tied up in maintenance work
- Sales and Marketing have to be well coordinated with Product Development

Background – Some Common Challenges in Small Software Companies

- In immature markets the company must be able to react to and utilise changes
  - The product development process needs to be flexible
    - ...thus, developers are frequently disturbed by new feature requests
- Communication between Business and Development can be difficult
  - Risk of over or under selling
    - Sales sells something Product Development cannot deliver
    - Sales does not start until the product is ready
  - Development progress is not visible, so salesmen don’t know what to promise to customers
  - Developers don’t know what was promised
- No common understanding of the (ad hoc) development process
- Difficult to plan development work, unpredictable outcomes
- New product development may be conducted through customer projects
  - Hard to keep the ‘big picture’ in mind
Creating a Software Engineering Management System for SME’s...

- We think that a company in the software product business needs to
  - Link *floor-level* product development with *long-term* product and business planning,
  - Combine *flexibility* needed in a turbulent environment with a degree of *control*
  - Understand the *right amount* of process for a given situation from the *business perspective*

Means – In other words, our results
The Cycles of Control Framework

A tool for...

1) Communicating the development process
2) Understanding software product development as a system composed of different perspectives and key areas
3) Identifying targets for process improvement
4) Adopting practices from literature as well as understanding the current ones
5) Linking business management and software development
6) Balancing development process flexibility and control


A practical guide summarising the results of the project

Often, in practice:

Question: How to get the big picture?

Through Rhythm!
The Cycles of Control is a Tool for Understanding and Communicating SW Product Development through Rhythm

The Cycles of Control

- The Cycles of Control is an abstraction of an iterative and incremental development process
  - Developed in the SEMS research project

- Developed for and tested in improving the product development processes of small to medium software product companies that operate in turbulent environments
  - Flexibility => react to changes in the market
  - A degree of control => hinder over-reacting to perceived change
An Overview

- Provides the interface between business management and product development
- Deals with the long-term plans for the product and project portfolio
- The value of long-term planning is to have a baseline to guide short-term decisions and trade-offs
- Continuous, both time- and event paced

Issues of Concern in Different Cycles

- Concerned with the individual release projects in which actual product versions are created
- Planning release projects according to the higher-level priorities set in the long-term release plan
- 1 – 4 months

- Building the product as a series of reasonably stable, working intermediate versions having a part of the functionality of the final release
- In each increment, a set of features is identified, specified in detail, designed, implemented, and tested
- Sets the interval during which developers are allowed to concentrate on the tasks at hand
- 2 weeks – 1 month

- The smallest pace at which the efforts and status of the release project are synchronized and reflected against a plan.
- Yield up-to-date information at regular intervals about project progress and system status
- From daily to weekly
**Issues of Concern in Different Cycles (Example)**

**Long-term planning of product releases**
- Product and technology roadmap, updated twice a year
  - Product vision(s)
  - Timing, role, type and content of releases
  - Resource planning
  - Oversees project and increment planning

**Incremental development of the releases**
- Increment planning
  - Increment goal setting
  - Converting requirements to tasks
  - Requirement freeze!
- Developing a tested product increment
  - Increment demo at the end
  - Show progress to all stakeholders
-- Time box typically 1 month

**Developing a release of the product**
- Project planning
  - Project goal setting (e.g. release criteria)
  - Timing, role and content of increments
  - Resource allocation
  - Testing strategy
- Developing a stable release candidate
-- Time box 3-6 months (2-4 releases/year)

**Synchronising the daily work**
- Status meetings
  - What has been done?
  - What will be done?
  - Any foreseeable problems?
- “Daily” routines and practices
-- Time box 1 day – 1 week

---

**Different Types of Development, Different Processes**

- **Product Architecture & Technology**
- **Major Product Releases**
- **Minor Product Releases**
- **Maintenance Releases**

**Possible variables:**
- Roles and resourcing
- # and duration of cycles
- Communication patterns
- Decision-making rights
- …
Example Implementation – Overview

The **R&D process** is the process for developing the software components for the product and only concerns the R&D team.

The **release process** is the process for all product-release-related activities. The model resembles the Stage-Gate™.

**“Strategic Release Management”**

**“Release Project Cycle”**

**Sprint**

**1 month**

**3 months**

**Commercialisation sprint (1 month)**

**Example Implementation – Development Rhythm**

New releases are scheduled **every three months**

A **commercialisation sprint** is done, when a release candidate is accepted by the Management Team to qualify for commercial release. In the commercialisation sprint the software is tested more thoroughly and product documentation and other accompanying stuff is finalized.

**“Strategic Release Management”**

**“Release Project Cycle”**

**Sprint**

**1 month**

**Commercialisation sprint (1 month)**

**Release process with go/kill gates**

**Releases are built incrementally in one month sprints**
Example Implementation – Control Points

Daily Scrum meetings are held to check development status and synchronise the efforts of the development team. The meeting takes a maximum of 10 minutes.

Sprint demos are done at the end of each sprint. The plans and results for the sprint are presented and a demo of the working software is shown to all interested. Discussion is encouraged. After the sprint demo sprint planning takes place, except for the last sprint, which ends in the sprint (and release candidate) demo.

At the gates, the Management Team makes go/hold decisions based on the entire project portfolio.

Release postmortem and success evaluation to collect lessons learned

Release process with go/kill gates

Example Implementation – Product Management

All product related ideas are entered into the product backlog (may include bugs)

The product mgmt team specifies (in more detail), prioritises and allocates items from the product backlog for the upcoming release projects

In the beginning of a release project, the requirements are specified (again, in more detail), (possibly) re-prioritised and allocated to sprints
**Example Implementation – Product Mgmt Control Points**

- **Product backlog**
  - Allocated into roadmap as
  - Release backlogs
    - Parts allocated into
    - Sprint backlog

- **Development**
  - Business

- **Release process with go/kill gates**

- **Release planning** is done in cooperation with key customers and partners + the company’s different stakeholders for the product. The Management Team of the company makes the decision to start projects in the appropriate release process gate.

- **Sprint planning** is done as a dialogue between the Sprint Board and the development team. The Sprint Board includes the Head of the Product Team, the Head of Professional Services, the Product Manager, and the Head of the R&D Team (representing different viewpoints to the product).

- **A comprehensive roadmap update** taking an in-depth look at future releases’ contents, timing, roles, and types with participants from the company board after a product release (twice per year). Also, the product roadmap is reviewed quickly after each increment to assess progress and impact of scope changes.

---

**SoberIT**

**Softmation, Business and Engineering Institute**

**Approaches to testing in iterative and incremental software development (viewed through the Cycles of Control framework)**

- Automated regression approach
- Stabilizing phase approach
- Separated system testing approach
Example: An Approach to Testing in I&I Dev

Strategic Cycle

Release Project Cycle

Increment + stabilizing phase

Separated, concurrent system testing increments + stabilizing phase

Lightweight Process Documentation ~ 1 picture / perspective

R&D Portfolio Management

Design and Implementation

Product Management

Construction

Product Development Rhythm

Testing and Quality Assurance

Development Organisation

Technical product management

Resourcing
Experiences

Experiences From Our Case Companies (1/2)

- The instantiated process is really used
  - “It just feels so natural” (Developer, Case A)
  - “In the last project we did everything by the book” (Chief architect, Case A)
  - “[the new process]...has made my life a lot easier” (R&D team leader, Case B)

- Reasonable Effort recorded for process improvement in 2002
  - Case A
    - R&D team leader 120 hours
    - Development team 49 hours
  - Setting up the process in Case B
    - R&D team leader ~1 man-month

- Control and flexibility
  - Requirements frozen during sprints
    - More stable work environment for the developers, fewer interruptions, improved satisfaction
    - Management commitment required
  - New requirements can be included in the following sprints
Experiences From Our Case Companies (2/2)

- Long-term vs. short-term
  - Product roadmaps with preliminary release backlogs and product vision
    - Concrete near-term development timetable
    - Refined at release process gates
  - Development process control points mapped to release process gates
    - Informed decision making
- Communication and visibility
  - Roadmap shows high-level plans (~1 year into the future)
  - Sprint planning communicates business goals to development
    - Different product perspectives represented
  - Sprint demonstrations have increased development visibility
    - All stakeholders can follow progress on a monthly basis
- Improved intra-company communication
  - Business understands Development and product status better
  - Ramp-up time for new employees shorter than before
- Developers report improved quality

Experiences – Success Factors for Improvement

- The Cycles of Control helps envision
  - how product development could be organised and paced
  - how strategic product planning and development can be integrated
  - How practices can be combined from different agile methodologies
- A driving force / champion is needed
- Management commitment is needed
  - Involvement through planning their own processes
- Stepwise process improvement with frequent feedback
  - “We learned something new in every sprint and made small adjustments to the process in feedback sessions. I don’t think we could have done it any better or faster.” (Chief architect, Case A)
Partnership Opportunities

Benefits in a Nutshell:

- Co-operation with the leading edge of software product development process research in Finland

- A highly motivated team with
  - Experience from tens of cases
  - 5+ years of experience from doing this type of coaching
  - A proven track record (ask our partners)

- A network of skilled professionals from the industry, academia and teaching
  - The latest knowledge from the field seeks you out (typically, the other way around)

Our next project starts in May 2004, contact me to hear more