



## A Tentative Framework for Managing Software Product Development in Small Companies

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

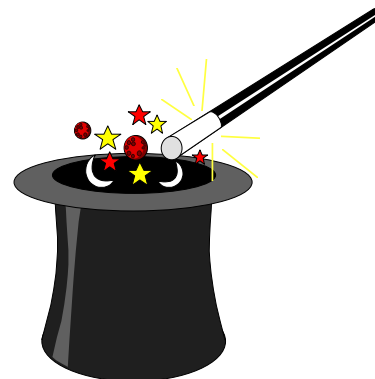
*Helsinki University of Technology  
Software Business and Engineering Institute*

<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

3



## 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

4



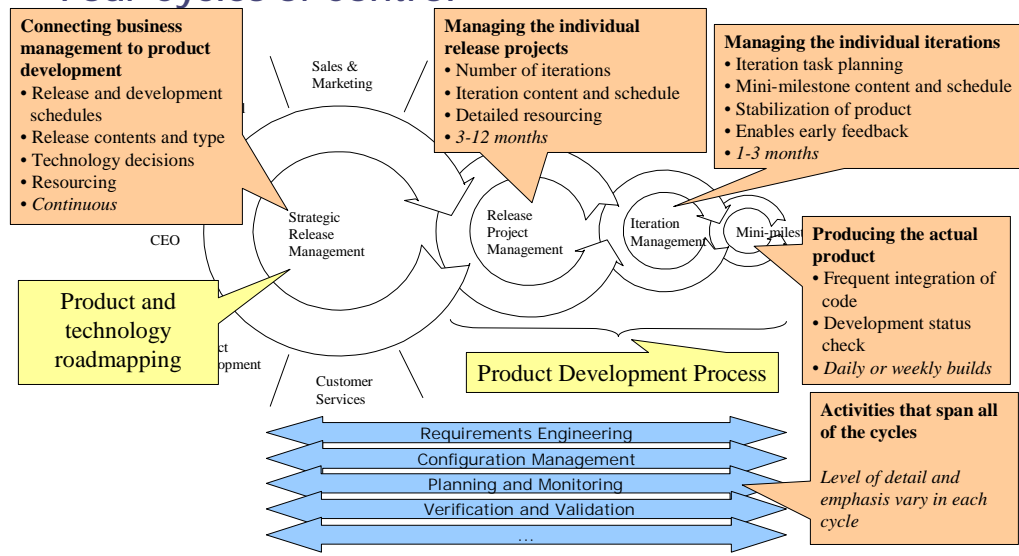
## Purpose of the Four Cycles of Control Framework

- ❑ 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

5

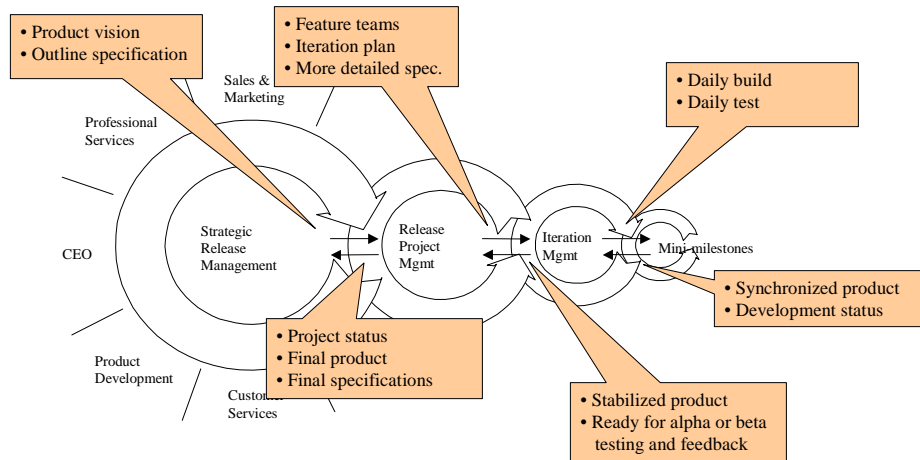


## Four Cycles of Control





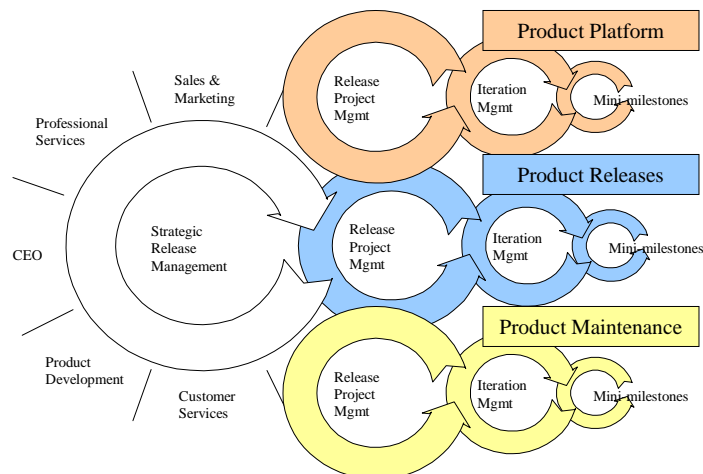
## An Example: Microsoft's Synch-and-Stabilize



7



## Different Types of Projects



Implementation details vary between different types of projects

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

8



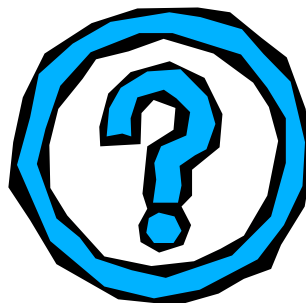
## Lessons Learned

- ❑ 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

9



## Questions and Comments



10