Software Engineering Process

Large-scale industrial software development

Björn Regnell

How many software developers?
> 2 million in China
> 2 million in India
> 2 million in US
> 2 million in EU
> 10 million in the world?

How much software?
> 10 LOC per day per developer?
> 100 million lines per day

How many bugs?
> 1 error per 1000 LOC?
> 100 000 errors per day?
Building SW from components

Ever Increasing SW Complexity
From Centralized to Decentralized to Distributed

Mjukvaran är själen i svensk industri
Group Discussions

Assume that you work as a software engineer with a large company.

• What do you think are the biggest challenges that you need to cope with?

Examples of real difficulties in real industrial projects

1. What is actually the best set of requirements?
2. How much uncertainty in effort estimation can we cope with?
3. At what level of detail should we document requirements?
4. How to minimize waiting time for other parts to be ready before we can start our part?
5. How to make more parts in parallel without generating confusion and unnecessary rework?
6. How to know when the product is reliable enough to be released?
7. How to incorporate changes without generating spaghetti and excessive cost of rework?
Who are the stakeholders?

**External stakeholders**
- Customers
- Operators
- Direct customers
- Retailers
- Indirect customers
- Content providers
- Product providers
- Direct Competitors
- Mobile phone developers
- Indirect Competitors
- Cameras
- Mobile music players
- Platforms providers
- Operating Systems
- Technical Platforms
- Network system providers
- Standardization bodies
- Legislation and authorities
- National
- International
- Manufacturing sub-contractors
- Component providers
- ... find the right person to talk to ...
- ... get the deep domain knowledge ...

**Internal stakeholders**
- Marketing
- Long term branding
- Customer relations
- Product management
- Roadmapping and portfolios
- Product development
- Hardware design
- Electronics
- Analog
- Digital
- Mechanics
- Software design
- User interface
- Service logic
- Network access
- Codecs
- Platform development
- Mother, daughters, cluster
- Global functions
- Sub-contracting management
- Technical platforms
- Operating systems
- Original Design Manufacturing
- Technology forecasting
- Market research
- Customer Services
- Support
- Repair
- Legal
- Sourcing
- Accessories
- ...
SW Value chains are getting more and more complex...

Software Ecosystems
Matrix Organization

Organizing the development

Pre-development in line functions; development in projects

TIMING CHALLENGE!!!

Toll Gate
Concept
## Orders of magnitude in Requirements Engineering

<table>
<thead>
<tr>
<th>Abrev.</th>
<th>Level</th>
<th>Order of magnitude</th>
<th>Managing a complete set of interdependencies…</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSRE</td>
<td>Small-Scale Requirements Engineering</td>
<td>~10 reqs</td>
<td>requires small effort.</td>
</tr>
<tr>
<td>MSRE</td>
<td>Medium-Scale Requirements Engineering</td>
<td>~100 reqs</td>
<td>is feasible but requires large effort.</td>
</tr>
<tr>
<td>LSRE</td>
<td>Large-Scale Requirements Engineering</td>
<td>~1000 reqs</td>
<td>is practically unfeasible, but feasible among small bundles of requirements.</td>
</tr>
<tr>
<td>VLSRE</td>
<td>Very Large-Scale Requirements Engineering</td>
<td>~10000 reqs</td>
<td>among small bundles of requirements is unfeasible in practice.</td>
</tr>
</tbody>
</table>

## Dealing with very large requirements databases

- Profitable?
- Strategic?
- Related?
- Ambiguous?
- Group?
- Complete?
- Split?
- Reject?
- Expensive?
Complex Information Structures

Challenges in Large-Scale Software Engineering

Chasm between marketing and development

<table>
<thead>
<tr>
<th>State-of-practice</th>
<th>State-of-the-art</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineering and marketing are not aligned</td>
<td>Common strategy and joint future vision of RE</td>
</tr>
</tbody>
</table>

Organisational instability and market turbulence

<table>
<thead>
<tr>
<th>State-of-practice</th>
<th>State-of-the-art</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inability to reach steady state</td>
<td>Stability combined with flexibility</td>
</tr>
</tbody>
</table>
Elementary or advanced process?

- **State-of-practice**
  - To small or too large process
- **State-of-the-art**
  - Managed process evolution

Specifications grow and grow and grow...

- **State-of-practice**
  - Monolithic specifications
- **State-of-the-art**
  - Tailored Requirements Databases
Overloaded Requirements Management

State-of-practice
Process is trashed by lack of early screening

State-of-the-art
Controlled requirements selection quality and screening

Resource planning
Release planning

State-of-practice
Uncertain estimates of cost and value

State-of-the-art
Systematic learning from estimation outcomes
Living with changing requirements

State-of-practice
Engineers want to “freeze” requirements

State-of-the-art
Continuous and controlled allocation of requirements, readiness for controlled changes

Living with design infiltration

State-of-practice
Unnecessary restrictions imposed by low-level requirements

State-of-the-art
Management of appropriate requirements abstraction level
Requirements are invented rather than discovered

<table>
<thead>
<tr>
<th>State-of-practice</th>
<th>State-of-the-art</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blinded by technology or</td>
<td>• Balancing market pull and technology push</td>
</tr>
<tr>
<td>passively reacting on current</td>
<td>• Strategic innovation process connected to</td>
</tr>
<tr>
<td>market</td>
<td>requirements engineering</td>
</tr>
</tbody>
</table>

Requirements dependencies and bundling

<table>
<thead>
<tr>
<th>State-of-practice</th>
<th>State-of-the-art</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependencies are treated ad hoc</td>
<td>Most important dependencies are managed by bundling and links</td>
</tr>
</tbody>
</table>
Kravberoende [Carlshamre et al., 2001]

**AND** Ömsesidigt OCH: A kräver B och vice versa
Printer kräver drivrutin och vice versa för att funka

**REQUIRES** Ensidigt OCH: A kräver B men ej vice versa
Skicka epost kräver nätverk men inte tvärtom

**TEMPORAL** A måste implementeras före B
Att ”lägga till” bör implementeras före ”ta bort”

**CVALUE** A&B har annat kundvärde än var för sig
on-line manual minskar värdet av tryckt manual

**ICOST** A&B har annan kostnad än var för sig
Krav på högre prestanda ökar impl. kostnaden

**OR** Bara en av A eller B behöver implementeras
Ritfunktion integrerad i ordbehandlare eller
länk till annat ritverktyg

---

Layers of Platforms
Platforms as product lines =>
Synchronization Challenge

Mobile product projects

Internal system & application platform

Technical platform product

Internal platform (HW/SW code base)

Time

Milestone

Requirements in sync over borders?

Consumers

Operator

Retailer

Device Competitor

Platform Competitor

Integrator

Supplier

Product Management

Integration Platform Development

SRS System Req Spec for integration platform

Requirements Sub-domain working groups

SRS System Req Spec for technical platform

Requirements Sub-domain working groups

SoC Statement of Compliance

Hotlist

Order

Release

Change Request

Maintenance Release

Detailed Requirements

Product Management

Technical Platform Development
Summary

- Large-scale development require processes, organizations and tools that can cope with increasing complexity
- Most of you will work in
  - Large companies
  - Small- or medium-size companies that deliver software or systems to large companies
- Your engineering skills depend also on
  - How you can combine technology and economics
  - Work in teams and with big organizations

Och sedan då....

Programvarutestning

Objektorienterad modellering och design

Konfigurationshantering

Programvaruutveckling i grupp

Programvaruutveckling för stora system

Coaching av programvaruteam

Ingångsprocessen - ekonomi och kvalitet

Projektplan

Test

Design

Binär

Utv-process