Software Production
A new paradigm for software engineering research

Sjaak Brinkkemper
Utrecht University

SOFSEM 2013
Outline

- Software products
  - Research framework
  - Software types
  - Economic context
- Software production
  - Software product management,
  - Software delivery
  - Software ecosystems
  - Incremental method evolution
- Educating software entrepreneurship
- Challenges for the research community
- Discussion
Societal context

ICT innovations create new products, services, companies, jobs, economic value, social value

International:
  Twitter
  Google
  SAP
  Oracle
  Microsoft
  Sage
  FaceBook
  ...

Netherlands:
  TomTom
  Planon
  GX
  Exact
  Unit4
  Afas
  Mendix
  ...

[Faculty of Science Information and Computing Sciences]
What is a Software Product?

A **Software Product** is defined as

*a packaged configuration of software components, or a software-based service with auxiliary materials, which is released for and traded in a specific market* (Xu & Brinkkemper, EJIS 2007)

Examples:
- ERP software
- Bookkeeping service
- Operating systems
- Desk-top publishing
- Computer-aided design
- Software development environments
- Customer-relationship mgmt
  ...

Essentially *all* software you are using in daily life and work
What is product software?

- **What is being sold?**
  - Device
  - Software

- **Number of copies**
  - One
  - Multiple

- **Microprogram**
  - Tailor-made software
  - Product software

- **Embedded software**
  - Online software
  - Packaged software
Key differences

1. Market introduction requires precise synchronization of dependable software engineering activities

2. The same product requires installation and usage in different organizations, with different hardware and software platforms

3. The vendor company stays owner of the software and auxiliary materials, while the usage is licensed to the customers
Research Framework for Software Production

- **Societal perspective**
  - Resource Provisioning: Technology, Educational system, Capital
  - Laws and Regulations: Intellectual property, Import / export
  - Entrepreneurship: Startups, Business models, Business culture
  - Economy: Markets, Industry structures

- **Company perspective**
  - Product Strategy: Market analysis, Product lifecycle management, Technology management
  - Strategic Management: Product investment, Resource management, International organization
  - Process and Quality: Method and project management, Knowledge management
  - Sales and Services: Services portfolio, Marketing, Localization and customizations
  - Operation: Performance monitoring, Crash reporting, Usage tracing

- **Development perspective**
  - Requirements and Architecture: Release planning, Design methods
  - Development: Programming, Testing, Configuration Mgmt.
  - Implementation: Delivery, Deployment, Licensing
Software types

1. System infrastructure software
   - Foundation of software stack
   - Security, systems management
   - Not industry or function specific
   - Many in open source

Segments
   - Operating systems
   - Network, storage, security
   - Utilities, anti-virus, archiving

2. Tools
   - Development, execution
   - Allow for composition, reuse, optimization

Segments
   - Portal and collaboration: user interface infrastructure
   - Information management: data layer
   - Modeling and development: for application logic
   - Execution and Integration platforms: execution of appl. logic
Software types (2)

3. Applications
   - Used by information workers and end-users
   - Bring some level of automation in business processes
   - Most are industry or function specific
   - Few in open source

Segments
   - Office automation
   - Horizontal business apps: finance&accounting, ERP, CRM
   - Vertical business apps for industry specific solutions: billing, banking, travel reservations
   - Technical applications: design and creation of physical products

4. Games and Consumer apps
   - Consumer and professional games
   - Strongest shift from packaged to online

Segments
   - Offline: Home consoles, handheld consoles, computers
   - Online: PC and mobile phones
   - Consumer apps: editing of video and photo, hobbies
Economic context

OECD (2008): “The product software sector is among the most rapidly growing sectors in OECD countries, with strong increases in value added, employment and R&D investment.”

EU
- 2.7 M employees

NL
- Import 1.2 B€, Export 1.6 B€
- VS: 78% market share!
- NL: 40,000 employees in NL companies
- 16% of 250,000 total ICT employment
European Software Market

Overall market: 229 B€ in 2008
3.7% growth/year
Software revenue model
<table>
<thead>
<tr>
<th>Country</th>
<th>SW revenues 2010 (€M)</th>
<th>SW revenues 2011 (€M)</th>
<th>% of total</th>
<th># Software companies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Germany</td>
<td>15578.0</td>
<td>18145.7</td>
<td>48.8%</td>
<td>15</td>
</tr>
<tr>
<td>UK</td>
<td>5752.0</td>
<td>5497.3</td>
<td>14.8%</td>
<td>22</td>
</tr>
<tr>
<td>France</td>
<td>3482.0</td>
<td>4040.9</td>
<td>10.9%</td>
<td>17</td>
</tr>
<tr>
<td>Netherlands</td>
<td>1093.0</td>
<td>2187.9</td>
<td>5.9%</td>
<td>8</td>
</tr>
<tr>
<td>Sweden</td>
<td>977.0</td>
<td>2163.5</td>
<td>5.8%</td>
<td>9</td>
</tr>
<tr>
<td>Norway</td>
<td>436.0</td>
<td>1110.8</td>
<td>3.0%</td>
<td>4</td>
</tr>
<tr>
<td>Finland</td>
<td>661.0</td>
<td>1005.5</td>
<td>2.7%</td>
<td>6</td>
</tr>
<tr>
<td>Italy</td>
<td>522.0</td>
<td>618.1</td>
<td>1.7%</td>
<td>3</td>
</tr>
<tr>
<td>Switzerland</td>
<td>881.0</td>
<td>530.1</td>
<td>1.4%</td>
<td>5</td>
</tr>
<tr>
<td>Belgium</td>
<td>511.0</td>
<td>516.5</td>
<td>1.4%</td>
<td>1</td>
</tr>
<tr>
<td>Poland</td>
<td>562.0</td>
<td>486.2</td>
<td>1.3%</td>
<td>2</td>
</tr>
<tr>
<td>Denmark</td>
<td>185.0</td>
<td>248.5</td>
<td>0.7%</td>
<td>1</td>
</tr>
<tr>
<td>Czech Rep.</td>
<td>166.0</td>
<td>194.4</td>
<td>0.5%</td>
<td>2</td>
</tr>
<tr>
<td>Slovakia</td>
<td>-</td>
<td>176.7</td>
<td>0.5%</td>
<td>1</td>
</tr>
<tr>
<td>Spain</td>
<td>117.0</td>
<td>149.1</td>
<td>0.4%</td>
<td>2</td>
</tr>
<tr>
<td>Austria</td>
<td>62.0</td>
<td>116.4</td>
<td>0.3%</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>30985.0</strong></td>
<td><strong>37187.7</strong></td>
<td><strong>100%</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>

Source: Truffle 100, see www.truffle100.com
Outline

- Software products
  - Research framework
  - Software types
  - Economic context
- Software production
  - Software product management,
  - Software delivery
  - Software ecosystems
  - Incremental method evolution
- Educating software entrepreneurship
- Challenges for the research community
- Discussion
Deliverable structure

Portfolio

*Portfolio*: The complete set of products of a company
Deliverable structure

Portfolio

Product 1  Product 2  ...  Product k

Product: A packaged configuration targeted to a specific market
Deliverable structure

Portfolio

Product 1  Product 2  ...  Product k

Release 1.0  Release 1.1  ...  Release 2.0  ...

Release: A formalized sellable version
**Deliverable structure**

- **Portfolio**
  - Product 1
  - Product 2
  - …
  - Product k
  - Release 1.0
  - Release 1.1
  - …
  - Release 2.0
  - …
  - Requirement 1
  - Requirement 2
  - …
  - Requirement n

*Requirement*: Wish for a future product feature
SPM Competence Model

Van de Weerd et al., RE 2006
# Maturity matrix – Maturity profile

<table>
<thead>
<tr>
<th>Focus area</th>
<th>Maturity levels</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements gathering</td>
<td></td>
<td>A</td>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C</td>
<td></td>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements identification</td>
<td></td>
<td>A</td>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements organizing</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements prioritization</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>Requirements selection</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>Release definition</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release validation</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>Launch preparation</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scope change management</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Theme identification</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core asset identification</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Roadmap construction</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Market trend identification</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Partnering &amp; contracting</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product lifecycle management</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product line identification</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Van de Weerd et al., ICSOB 2010*
# Maturity in SPM

Many organizations have low maturity

<table>
<thead>
<tr>
<th>Focus Area</th>
<th>None</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Portfolio management</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Market analysis</td>
<td>30.2</td>
<td>32.6</td>
<td>16.3</td>
<td>4.7</td>
<td>7.0</td>
<td>9.3</td>
</tr>
<tr>
<td>Partnering &amp; contracting</td>
<td>16.3</td>
<td>23.3</td>
<td>37.2</td>
<td>2.3</td>
<td>11.6</td>
<td>9.3</td>
</tr>
<tr>
<td>Product lifecycle mgmt</td>
<td>47.6</td>
<td>14.3</td>
<td>19.0</td>
<td>7.1</td>
<td>0.0</td>
<td>11.9</td>
</tr>
<tr>
<td><strong>Release planning</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Roadmap intelligence</td>
<td>46.5</td>
<td>23.3</td>
<td>4.7</td>
<td>2.3</td>
<td>14.0</td>
<td>9.3</td>
</tr>
<tr>
<td>Core asset roadmapping</td>
<td>48.4</td>
<td>21.0</td>
<td>19.4</td>
<td>6.5</td>
<td>4.8</td>
<td>-</td>
</tr>
<tr>
<td>Product roadmapping</td>
<td>14.5</td>
<td>25.8</td>
<td>12.9</td>
<td>33.9</td>
<td>3.2</td>
<td>9.7</td>
</tr>
<tr>
<td><strong>Product planning</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements prioritization</td>
<td>21.0</td>
<td>35.5</td>
<td>21.0</td>
<td>3.2</td>
<td>9.7</td>
<td>9.7</td>
</tr>
<tr>
<td>Release definition</td>
<td>9.7</td>
<td>45.2</td>
<td>8.1</td>
<td>33.9</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td>Release definition validation</td>
<td>25.8</td>
<td>38.7</td>
<td>16.1</td>
<td>19.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Scope change management</td>
<td>59.0</td>
<td>9.8</td>
<td>6.6</td>
<td>8.2</td>
<td>16.4</td>
<td>-</td>
</tr>
<tr>
<td>Build validation</td>
<td>9.3</td>
<td>32.6</td>
<td>55.8</td>
<td>2.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Launch preparation</td>
<td>12.9</td>
<td>45.2</td>
<td>11.3</td>
<td>1.6</td>
<td>3.2</td>
<td>6.5</td>
</tr>
<tr>
<td><strong>Requirements management</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements gathering</td>
<td>0.0</td>
<td>22.6</td>
<td>32.3</td>
<td>1.6</td>
<td>6.5</td>
<td>19.4</td>
</tr>
<tr>
<td>Requirements identification</td>
<td>25.6</td>
<td>11.6</td>
<td>14.0</td>
<td>46.5</td>
<td>2.3</td>
<td>-</td>
</tr>
<tr>
<td>Requirements organizing</td>
<td>17.7</td>
<td>21.0</td>
<td>38.7</td>
<td>22.6</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Intriguing data

Intriguing data

Bekkers et al., RE 2012
### Binary Priority Listing in large scale RM

![Excel spreadsheet showing priority listing](attachment:image)

<table>
<thead>
<tr>
<th>ID</th>
<th>Requirement</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>MR1000731</td>
<td>Raw material expiry date</td>
<td>9</td>
</tr>
<tr>
<td>MR1000732</td>
<td>Rest Times in Quality</td>
<td>3</td>
</tr>
<tr>
<td>MR1000733</td>
<td>Test Procedure - Quality</td>
<td>13</td>
</tr>
<tr>
<td>MR1000734</td>
<td>Version control on the test procedure - Quality</td>
<td>1</td>
</tr>
<tr>
<td>MR1000735</td>
<td>Multi-Dimensional Inventory</td>
<td>10</td>
</tr>
<tr>
<td>MR1000736</td>
<td>Quality - Selecting Inventories based on Customer</td>
<td>2</td>
</tr>
<tr>
<td>MR1000737</td>
<td>Configurator For Formulas</td>
<td>6</td>
</tr>
<tr>
<td>MR1000738</td>
<td>Pricing and Containerization</td>
<td>12</td>
</tr>
<tr>
<td>MR1000739</td>
<td>Yield for end items</td>
<td>8</td>
</tr>
<tr>
<td>MR1000740</td>
<td>Yield dependant operation</td>
<td>4</td>
</tr>
<tr>
<td>MR1000741</td>
<td>Yield dependant materials</td>
<td>11</td>
</tr>
<tr>
<td>MR1000742</td>
<td>Yield calculation for material, capacity and cost</td>
<td>15</td>
</tr>
<tr>
<td>MR1000743</td>
<td>Modifications on batches, addition of operation</td>
<td>7</td>
</tr>
<tr>
<td>MR1000744</td>
<td>Actual yield in pspmg001</td>
<td>14</td>
</tr>
</tbody>
</table>
Customer Involvement Factory

- Reports of questions, complaints or bugs
- Approximately 60,000 reports per year
- Handled by helpdesk and consultancy
- 85% can be solved by referring to the manual
- 15% is a bug or shortcoming

Kabbedijk et al., RE 2009
The **Product Backlog** contains a prioritized list of all items relevant to a specific product. This list can consist of bugs, customer requested enhancements, competitive product functionality, competitive edge functionality and technology upgrades.

The **Product Management Software Backlog** consists of tasks that can be finished by the SPM team within the sprint.
Alternating Sprints

- Development sprint backlog is always **up-to-date** and ready for use once the software development sprint starts
- **Reduced time** between inception of requirements and realization in the product
- **Information flow-back** from development to PM
Release planning with ILP

- Mathematical formalization using Integer Linear Programming

- Given are:
  - Lists of requirements
  - Revenue estimates
  - Resource cost estimates
  - Development teams and capacities

- Managerial steering mechanisms
  - Team transfers
  - Deadline extension
  - External resources
  - Mandatory requirements

Akker et al., REFSQ 2005; Li et al., REJ 2010
**Development by one pool of developers**

There are $n$ requirements $R_j$.

- $v_j = \text{estimated revenue of } R_j$
- $a_j = \text{estimated resources need for } R_j$
- $Q = \text{number of developers in pool.}$
- $T = \text{development period}$
- $d(T) = \text{number of working days in planning period}$

\[
x_j = \begin{cases} 
1 & \text{if requirement } R_j \text{ is selected;} \\
0 & \text{otherwise.}
\end{cases}
\]

\[
\max \sum_{j=1}^{n} v_j x_j \quad \text{subject to}
\]

\[
\sum_{j=1}^{n} a_j x_j \leq d(T)Q,
\]

\[
x_j \in \{0, 1\},
\]

for $j = 1, \ldots, n$. 
Team development and transfers

\( a_{ij} = \) estimated resource need for \( R_j \) from team \( i \). There are \( m \) teams.

\( Q_i = \) number of developers in team \( i \).

\( U_{\text{cap}} = \) atomic transfer capacity unit

\( y_{ik} = \) number of transfer capacity units from team \( i \) deployed in team \( k \).

\( \alpha_{ki} = \) transfer effectiviness factor (from team \( k \) to team \( i \)).

\[
\sum_{j=1}^{n} a_{ij} x_j \leq d(T) Q_i, \quad \text{for } i = 1, \ldots, m.
\]

\[
\max \sum_{j=1}^{n} v_j x_j \text{ subject to } \sum_{j=1}^{n} a_{ij} x_j \leq U_{\text{cap}} [y_{ii} + \sum_{k:k \neq i} \alpha_{ki} y_{ki}] \quad \text{for } i = 1, \ldots, m,
\]

\[
\sum_{k=1}^{m} y_{ik} = m_i, \quad \text{for } i = 1, \ldots, m,
\]

\[
x_j \in \{0, 1\}, \quad \text{for } j = 1, \ldots, n,
\]

\[
y_{ik} \text{ nonnegative and integral,} \quad \text{for } j = 1, \ldots, n.
\]
Key decisions in Software Start-ups

Software Business Start-Up Memories
Key Decisions in Success Stories

Roderick van Cann, Slinger Jansen and Sjaak Brinkkemper

Released: 17 Jan 2013
See Amazon
Outline

- Software products
  - Research framework
  - Software types
  - Economic context
- Software production
  - Software product management,
  - Software delivery
  - Software ecosystems
  - Incremental method evolution
- Educating software entrepreneurship
- Challenges for the research community
- Discussion
CCU - Customer Configuration Updating

- Vendor
- Vendor Repository
- Product

Vendor

Vendor Repository

Product

Uninformed Customer

Advertise Update

Informed Customer

Receive Info

Receive Update

Customer Possesses Update

Deploy/Install Update

Customer

Possesses Update

Install Customer

Activate Update

Installed Customer

Remove

Rollback/Deinstall

Deactivate

Reconfigure

Activated Customer

Jansen et al, ICSE 2006
Multi-tenancy in Product Lines

Need for resource sharing

Need for Variability

Standard Multi-tenant Solution

Configurable Multi-tenant Solution

Custom Software Solution

SPL Solution

a = Business Growth
b = Customer Requirements Growth

PAAS

IAAS
Patterns for Multi-tenancy

Kabbedijk et al., WICSA/ECSA 2012
The Catalogue of Customization Realization Techniques

<table>
<thead>
<tr>
<th>Customization Realization Technique</th>
<th>Latest introduction time</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model change</td>
<td>Design</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View change</td>
<td>Detailed design</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Controller change</td>
<td>Detailed design</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>System connector change</td>
<td>Architecture design</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System component change</td>
<td>Architecture design</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) Infrastructure centered architecture, (b) Run-time variant component specialization, (c) Variant component implementations, (d) Condition on variable, (e) Code fragment super-imposition
Outline

- Software products
  - Research framework
  - Software types
  - Economic context
- Software production
  - Software product management,
  - Software delivery
  - Software ecosystems
  - Incremental method evolution
- Educating software entrepreneurship
- Challenges for the research community
- Discussion
Microsoft Internet Explorer has encountered a problem and needs to close. We are sorry for the inconvenience.

If you were in the middle of something, the information you were working on might be lost.

Restart Microsoft Internet Explorer

Please tell Microsoft about this problem.
We have created an error report that you can send to help us improve Microsoft Internet Explorer. We will treat this report as confidential and anonymous.

To see what data this error report contains, click here.

Send Error Report  Don't Send
Feedback in Software Supply Chains
## Software Ecosystem Modeling

- Developed at Utrecht University
- Software Supply Network (SSN)
- Product Deployment Context (PDC)

### Company of Interest (CoI)
- The CoI delivers the PoI in the business model. Its color is blue and its border line is solid.

### Supplier
- An actor that supplies one or more required products or services to a Supplier. Its color is orange and its border line is solid.

### Customer
- An actor that directly or indirectly acquires or makes use of the PoI. Its color is yellow and its border line is solid.

### Intermediate
- Actors like Distributors, Retailers, etc., act as intermediaries between two parties. Their color is gray and its border line is solid.

### Customer’s Customer
- A customer might have his own customers being provided with a product or service directly or indirectly from the CoI. Example: Product Support, Updates, etc. Its color is gray and its border line is solid.

### Trade Relationship
- Connects two actors. A relationship might be complex, consisting of many flows of arbitrary directions. Drawn as a black solid line.

### Flow
- Represents an artifact or service flow from one actor to another. One or more characters subscript X, denoting the nature of the transferred artifact (Flow Type). V is subscripted by a number, denoting different artifacts of the same nature. Its color is white and its border line is solid.

### OR Gateway
- Enables one or more or all Trade Relationships and their Flows between the Input Trade Relationships and the Output Trade Relationship. Its color is black; the color of the text is white.

### XOR Gateway
- Enables only one Trade Relationship and its Flows between the Input Trade Relationships and the Output Trade Relationship. Its color is black; the color of the text is white.

---

Lucassen et al., ICSOB 2012
SECO models
**Software Operation Knowledge**

*Definition:*

**Software Operation Knowledge** is knowledge of in-the-field performance, quality and usage of software, and knowledge of in-the-field end-user software experience feedback

*Vd Schuur et al., CSMR 2011*
Book on Software Ecosystems

Available March 2013
Outline

- Software products
  - Research framework
  - Software types
  - Economic context

- Software production
  - Software product management,
  - Software delivery
  - Software ecosystems
  - Incremental method evolution

- Educating software entrepreneurship
- Challenges for the research community
- Discussion
The software startup

Start with a simple process
Incremental Process Capability

Gradual more extensive processes are added dependent on:
- product
- development culture
- internal maturity
- external stakeholders
On-line Method Engine

Van de Weerd et al., CAiSE 2007
Formalization of Method Incrementation

- **Definition 3.1**
The mapping method: \( T \rightarrow M \), where 
\( m = \text{method}(t) \) means that the method \( m \in M \) is the valid method at time \( t \).

- **Definition 3.6**
A method increment is a method fragment 
\( f \in F \) such that \( \exists i \) contains\( (f, \text{method}(t_i)) \) \( \land \neg \text{contains}(f, \text{method}(t_{i-1})) \)

- **Definition 3.7**
The mapping insert

\( F \times M \rightarrow M: \text{insert}(f, m_1) = m_2 \)

means that the method fragment \( f \)
has been inserted in the method \( m_1 \)
resulting into method \( m_2 \)

- **Rule 3.1**
Insertion of concepts

\( \text{insert}(c, m_i) = m_{i+1} \Rightarrow \neg \text{concept}(c, m_i) \land \text{concept}(c, m_{i+1}) \)

- **Example**

insert(RELEASE TABLE,BaanIncr2) = BaanIncr3 \( \Rightarrow \)
\( \neg \text{concept}(\text{RELEASE TABLE,BaanIncr2}) \land \text{concept}(\text{RELEASE TABLE,BaanIncr3}) \)

The release table was not a concept in BaanIncr2 and therefore inserted to get BaanIncr3
Collaboration patterns

Kristjansson et al., JKE 2011
Product Knowledge Center

Nonaka spiral
Outline

- Software products
  - Research framework
  - Software types
  - Economic context
- Software production
  - Software product management,
  - Software delivery
  - Software ecosystems
  - Incremental method evolution
- Educating software entrepreneurship
- Challenges for the research community
- Discussion
Educating ICT Entrepreneurship

- Course ICT Entrepreneurship at Utrecht University, since 2002
- Part of MSc in Business Informatics
- The students develop their own product or service ideas in a self composed team
- Role play in a virtual company as if there is one year in the company
- Students play a horizontal role in a staff dept: HRM, Finance, Marketing, Project Office, Technology, Investor Relations
- Mid review and End review for external Supervisory Board
- Deliverables as assignments: Product Definition, Market Plan, Project Plan, Business Model, Architecture, Business Plan, Prototype
- After the course, the students decide for continuation:
  - office in incubator
  - coaching and network
  - company registration
Course Organization

Horizontal teams for company experience

Finance
HRM
Project Office
Marketing

Business Unit
CEO

Management Team

Supervisory Board

Team 1 P1
Team 2 P2
Team 3 P3
Team n Pn

Product teams for own product development
Welcome to our website!

Welcome to the Netherware website! Here we will publish all the latest and hottest news concerning Netherware, the software incubator. With various innovative products to be released soon news may pop up when you least expect it.

The goal of Netherware is to enable student entrepreneurship in the ICT industry. Students can come with any idea for a product or service company, and elaborate this into a prototype with a business plan for a startup.

Netherware provides the following for the to be entrepreneurs:

News topics

Netherware Bedrijven Nieuws Er is goed nieuws te melden: één van de eerste Netherware bedrijven Elnr, behaald deze week haar 100ste klant. We feliciteren Elnr met het geweldige resultaat. [2008-04-24]Bron ...

Posted Dec 16, 2008 8:34 AM by Paulus Schoutsen

Interview Campusradio Op 21 juni is een interview met Slinger Jansen uitgezonden over Netherware en het vak ICT-Entrepreneurship. Klik op play om het interview te beluisteren. [2007-07-06]

Posted Dec 16, 2009 8:34 AM by Paulus Schoutsen
Example products

Examples
1. Hours accounting and invoicing for law firms
2. Warehousing for the retail industry
3. Utility and website to quit smoking
4. Dancing lessons in a browser
5. Camera placement for movie directors
6. Music distribution channel
7. Scanning with mobile phone
8. Simple multi-authoring tool for publishers

- 30+ start-ups with more than 100 jobs
- Permanent Software incubator
- Office space on campus
Wat weet jij over je financiën?
Yunoo geeft je inzicht in je persoonlijke financiën, vergroot je financiële kennis en helpt je geld besparen. **Meld je nu gratis aan!**

gratis aanmelden of inloggen

---

**Al je financiën inzichtelijk**
Yunoo categoriseert automatisch al je inkomsten en uitgaven zodat je direct weet waar je geld heen gaat.

**Ontdek de Yunoo community!**
Deel je ervaringen, discussieer over verschillende financiële onderwerpen en haal voordeel uit de gezamenlijke kennis.

**Tips om geld te besparen**
Yunoo geeft pro-actieve tips om geld te besparen, maar men kan elkaar ook informeren over besparende tips.
Course milestones

- Deliverables as assignments: Product Definition, Market Plan, Project Plan, Business Model, Architecture, Business Plan, Prototype

- All horizontals report bi-weekly to the lecturers, and prepare sections to the Year-end report

- Mid review to Supervisory Board (SuB)
- End review session with SuB

- The year-end report includes an spin-out plan for the next phase:
  - Which products will be continued: extended, modified,
  - Which products will be discontinued
  - Which products will be nominated for external financing
Mid and End review

All students present their products and business plans to **Board of Supervisors**

5 M€ Virtual Venture Capital

**Members** of Board of Supervisors
- Venture capital fund
- IT entrepreneur
- Institutional investment
- IT journalist
- Marketing specialist
Outline

- Software products
  - Research framework
  - Software types
  - Economic context
- Software production
  - Software product management,
  - Software delivery
  - Software ecosystems
  - Incremental method evolution
- Educating software entrepreneurship
- Challenges for the research community
- Discussion
Research Challenges

- What are the fundamental theories of Software Production?
- How can optimal multi-tenant customizations be deployed in a secure manner?
- How can high volume requirements selection be performed in techno-economic alignment?
- How can maturity growth in software production be supported by incremental method evolution?
- Can we build fault tolerant components in the software supply chain, both from a technical as well as from an organizational perspective?
- What are the essential development tools for product development and deployment?
Discussion and Invitation

Questions?

- Papers see: www dblp org
- Email: S Brinkkemper@uu nl
- Contact me in case you are interested in joining a European project for Education in Software Entrepreneurship