Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Software Production: fundamental research for the software society

Sjaak Brinkkemper
Utrecht University

Software Engineering Netherlands
3 December 2014

Outline

- Software products
  - Software types
  - Economic context
  - Research framework
- Software production
  - Software product management,
  - Software ecosystems
  - Incremental method evolution
- Challenges for the research community
- Invitation and discussion

Societal context

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

International:
- Google
- SAP
- Oracle
- Microsoft
- Sage
- FaceBook
- SalesForce
- …

Netherlands:
- TomTom
- Planon
- GX
- Exact
- Unit4
- Afas
- Mendix
- …

What is a Software Product ?

A Software Product is defined as

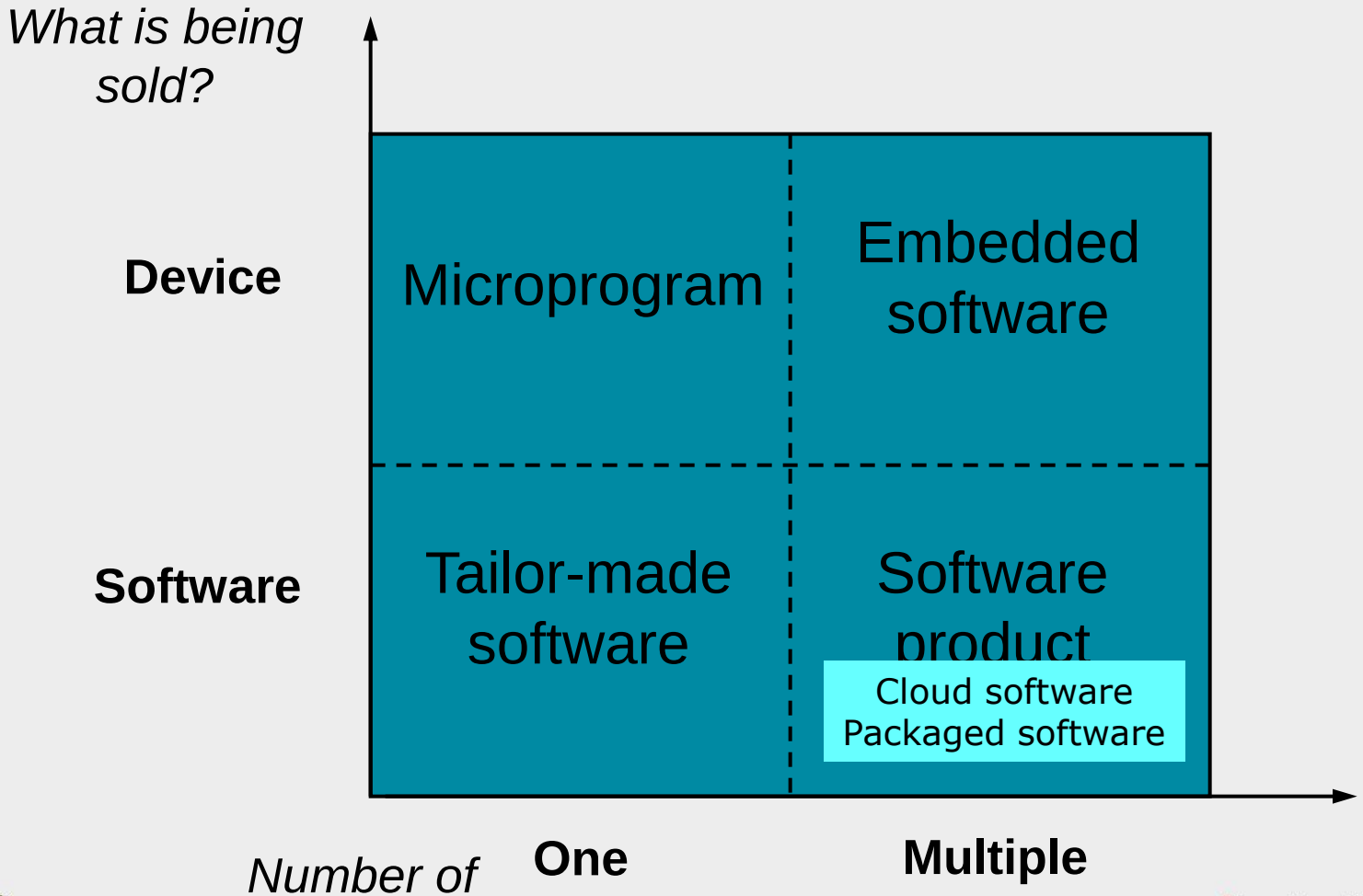
> *a packaged configuration of software components, or a software-based service with auxiliary materials, which is released for and traded in a specific market*
> *(Xu & Brinkkemper, EJIS 2007)*

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 a software product ?



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

Note: Cloud and mobile have huge impact on 2

Four main software types

1. System infrastructure software
2. Development tooling
3. Enterprise applications
4. Games and consumer apps

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. Enterprise 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

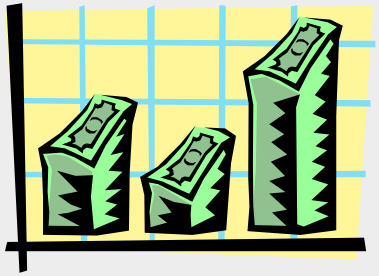
- Consumer and professional games
- Strongest shift from packaged to cloud and mobile

Segments

- Offline: Home consoles, handheld consoles, computers
- Online: Cloud and mobile apps
- 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

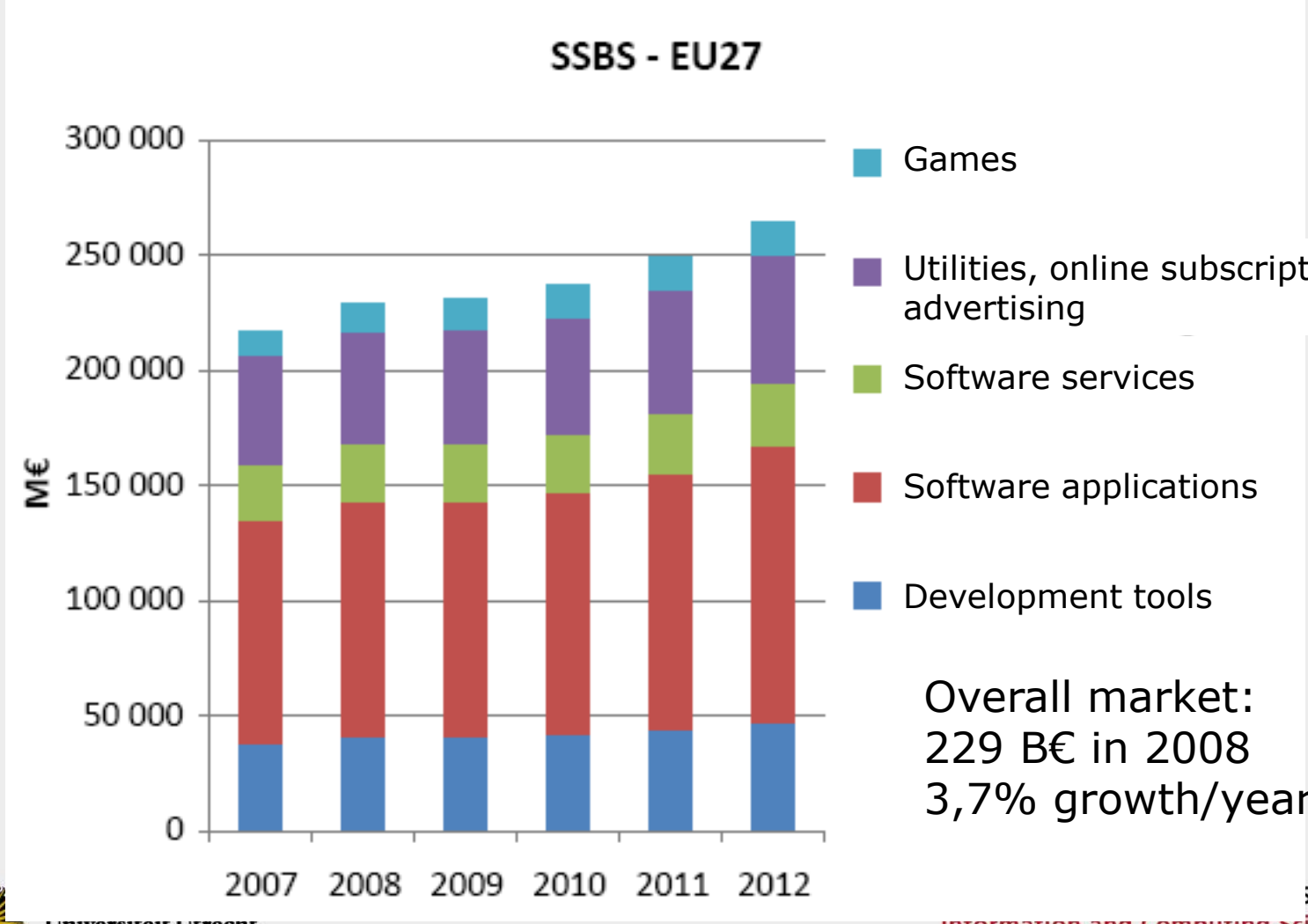
- 2.7 M employees
- Few good statistics

NL

- Dialogic research 2014
- 12.300 companies with 10,5 B euro revenue
- Export: 3.5 Beuro
- NL: 74.000 employees in NL-owned companies
- Total software: 2.8% GDP

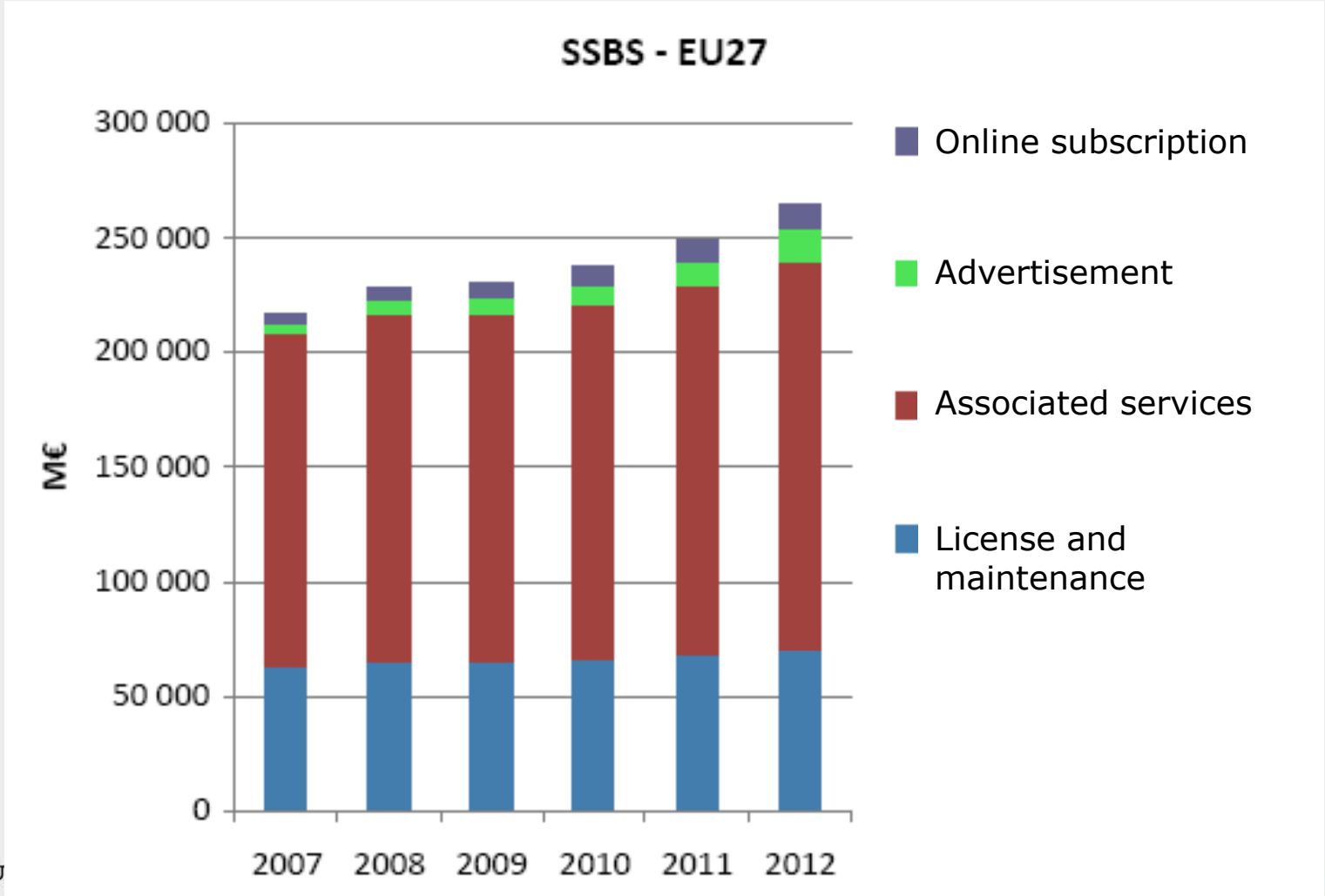
http://www.nederlandict.nl/Files/TER/De%20Nederlandse%20softwareindustrie%202014.pdf

European Software Market



Overall market:
229 B€ in 2008
3,7% growth/year

Software business model



Top 100 in EU software industry

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

Source: Truffle 100, see www.truffle100.com

R&D jobs per country

Wolters Kluwer
Unit4
Gemalto
Exact
Centric
TSS
AFAS

Countries	Number R&D employees	% of total	R&D investment (€M)	% of total
Germany	23 783	37.6%	2 837.7	41.5%
France	9 962	15.7%	960.6	14.0%
UK	8 375	13.2%	1 043.4	15.3%
Netherlands	6 321	10.0%	713.2	10.4%
Sweden	4 503	7.1%	438.2	6.4%
Poland	3 569	5.6%	75.2	1,1%
Italy	1 634	2.6%	99.4	1.5%
Finland	1 232	1.9%	59.9	0.9%
Switzerland	1 098	1.7%	65.1	1.0%
Norway	536	0.8%	244.4	3.6%
Czech Republic	512	0.8%	70.8	1.0%
Denmark	506	0.8%	56.3	0.8%
Belgium	465	0.7%	91.7	1.3%
Slovakia	332	0.5%	32.4	0.5%
Spain	322	0.5%	33.2	0.5%
Austria	140	0.2%	19.6	0.3%
Total	63 290	100 %	6 841.1	100 %

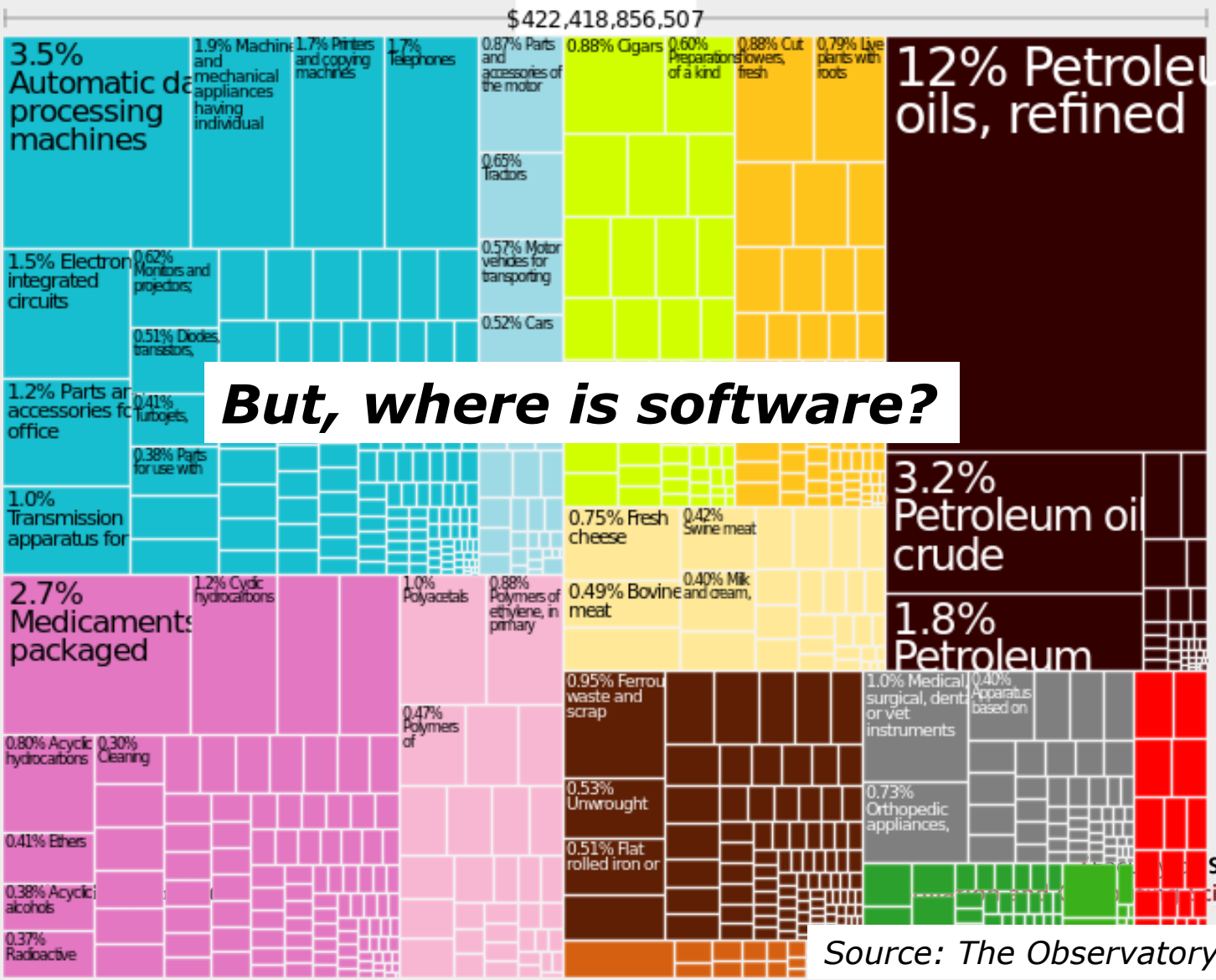
Source: Truffle 100

see www.truffle100.com

Univers

[Faculty of Science
Information and Computing Sciences]

Netherlands Export

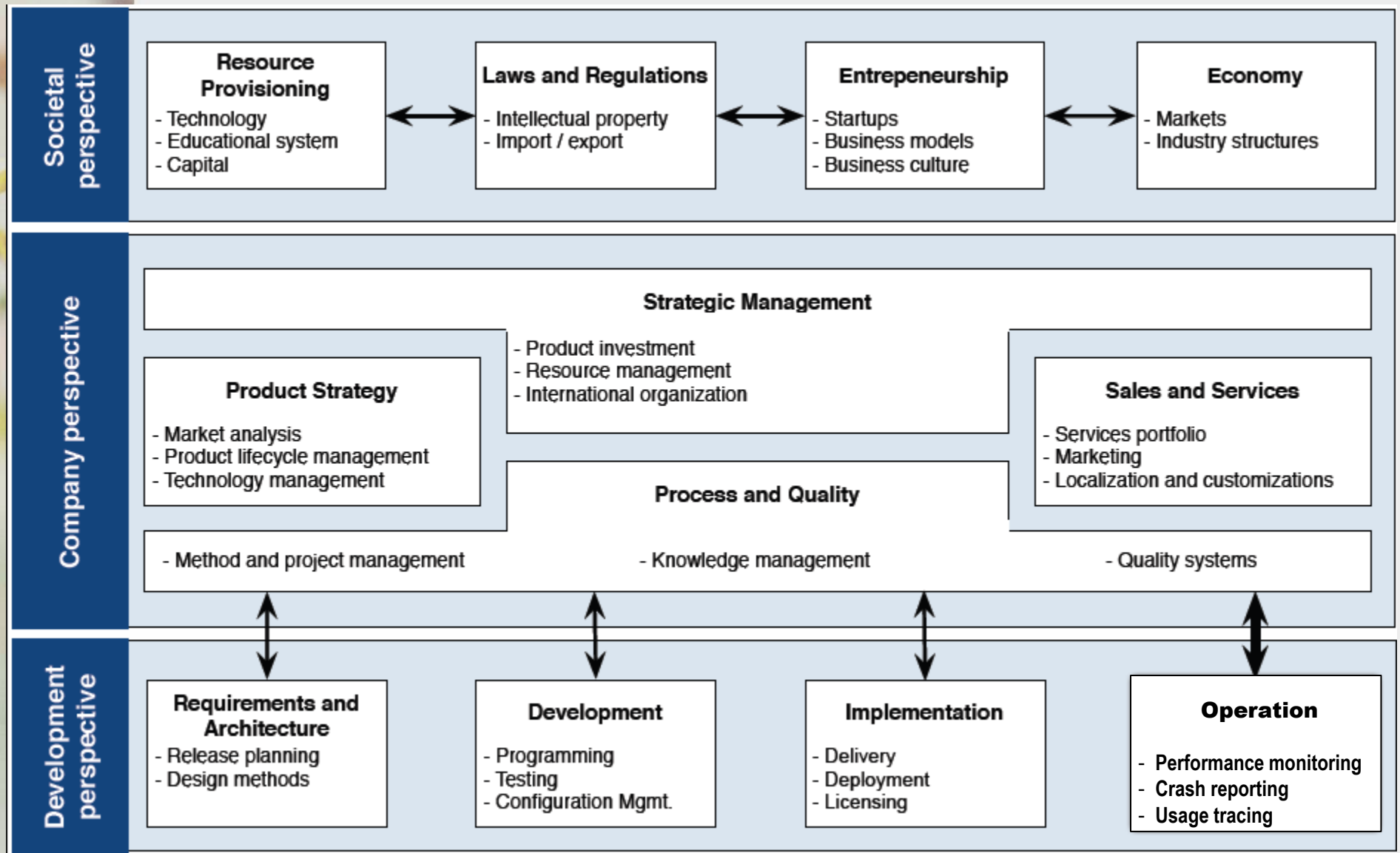


Source: The Observatory

SoftwareVOC

- Network of software industry in NL
  - About ~300 companies
  - Admin and meetings: Nederland ICT
  - Board: AFAS, Unit4, Exact, VitalHealth, Hydrologic, AgroVision, Levi9, UUtrecht
  - Annual conference SINC: run by CKC-seminars
- Industry invests heavily in Research and Development
  - ~25% of annual revenue (NL 2.5 Beuro ?)
  - Constant need for functional and technical innovation
  - Opportunity: Perform empirical and experimental research at industry premises
  - Industry is willing to pay for PhD research and match with cash
- Let's not waste taxpayers' money

Research Framework for Software Production



Research at Utrecht University

- Software Production
- Software Product Management
- Software Ecosystems and Delivery
- Software Architecting
- Game Production
- Knowledge networks

See www.cs.uu.nl

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

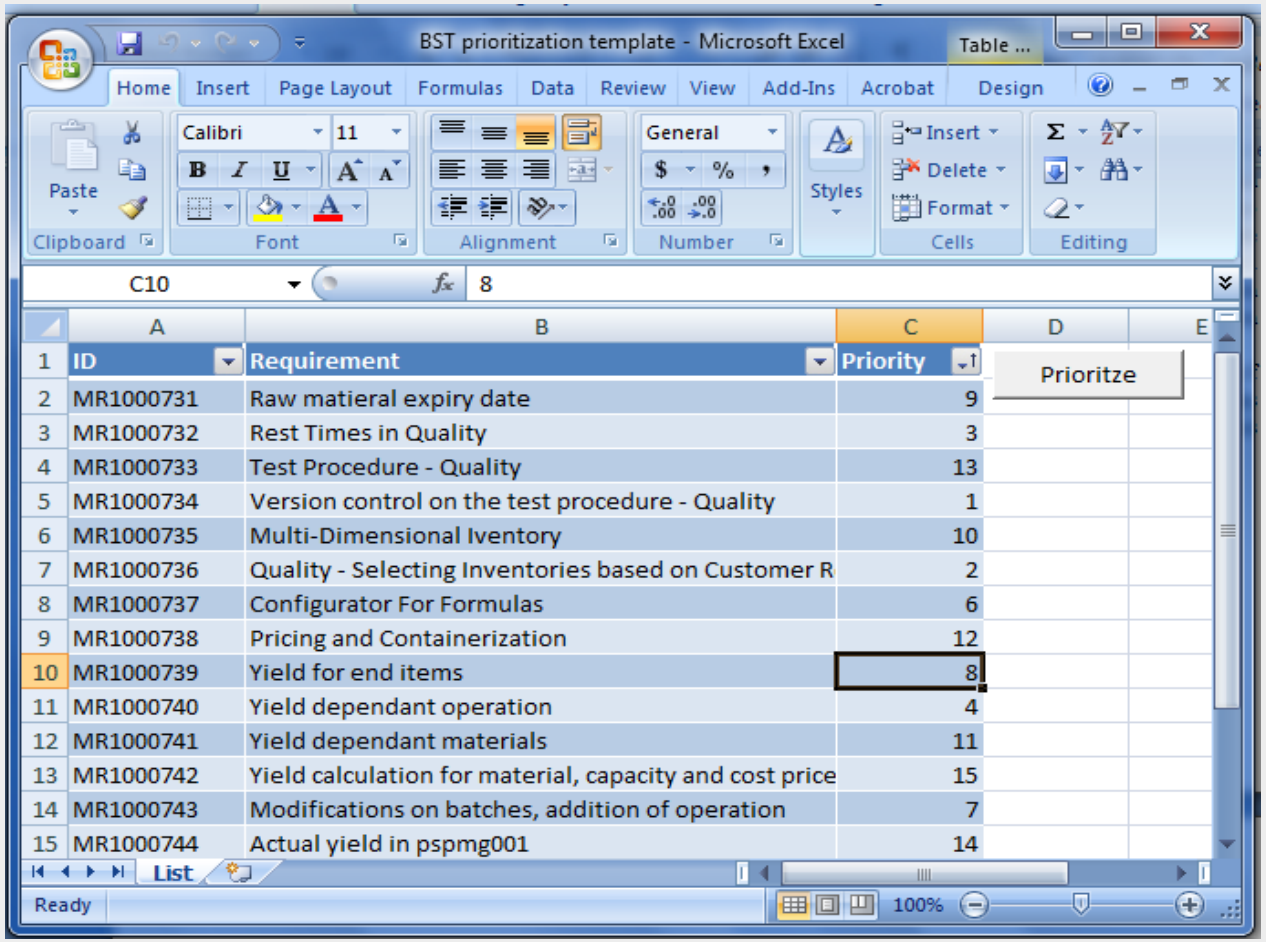
Take home message

Scientific research in the field of software production is fundamental for creating innovative products and services, for new companies, jobs, economic and social value, but cooperation with the software industry is key to get well founded theories.

Outline

- Software products
  - Software types
  - Economic context
  - Research framework
- Software production
  - Software product management,
  - Software ecosystems
  - Incremental method evolution
- Challenges for the research community
- Invitation and discussion

Binary Priority Listing in large scale RM



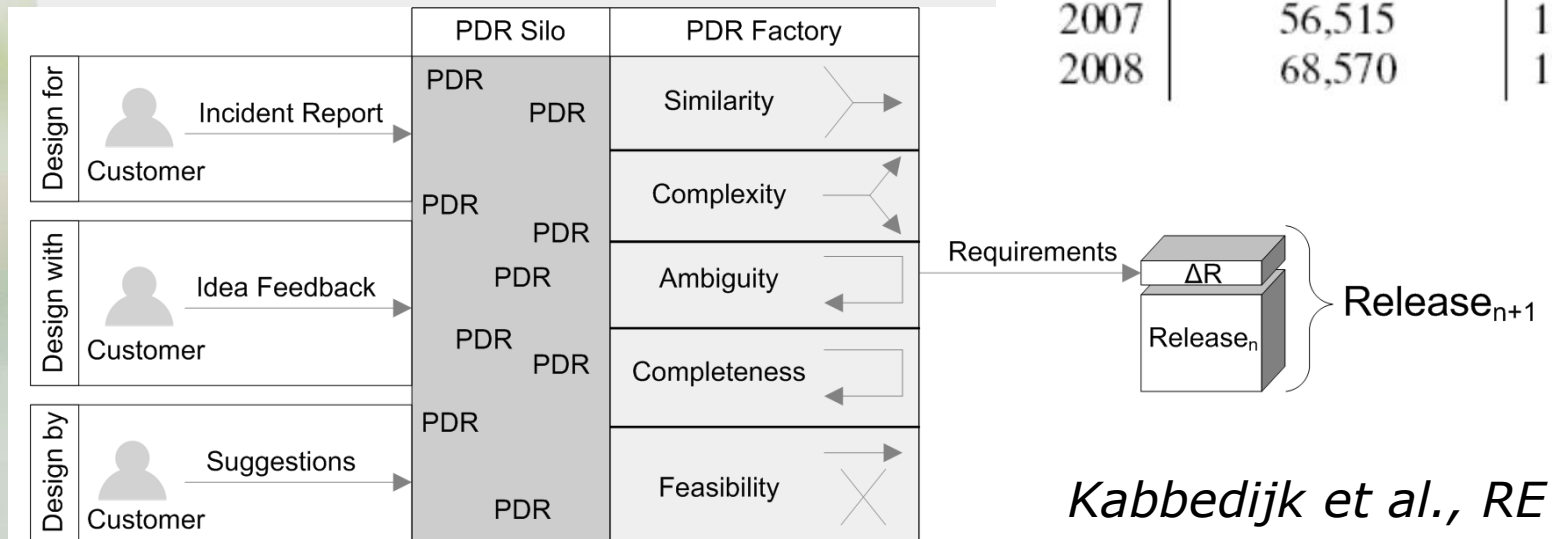
	A	B	C	D	E
1	ID	Requirement	Priority		
2	MR1000731	Raw matieral expiry date	9		
3	MR1000732	Rest Times in Quality	3		
4	MR1000733	Test Procedure - Quality	13		
5	MR1000734	Version control on the test procedure - Quality	1		
6	MR1000735	Multi-Dimensional Iventory	10		
7	MR1000736	Quality - Selecting Inventories based on Customer R	2		
8	MR1000737	Configurator For Formulas	6		
9	MR1000738	Pricing and Containerization	12		
10	MR1000739	Yield for end items	8		
11	MR1000740	Yield dependant operation	4		
12	MR1000741	Yield dependant materials	11		
13	MR1000742	Yield calculation for material, capacity and cost price	15		
14	MR1000743	Modifications on batches, addition of operation	7		
15	MR1000744	Actual yield in pspmg001	14		

Bebensee et al., REFSQ 2010

Customer Involvement Factory

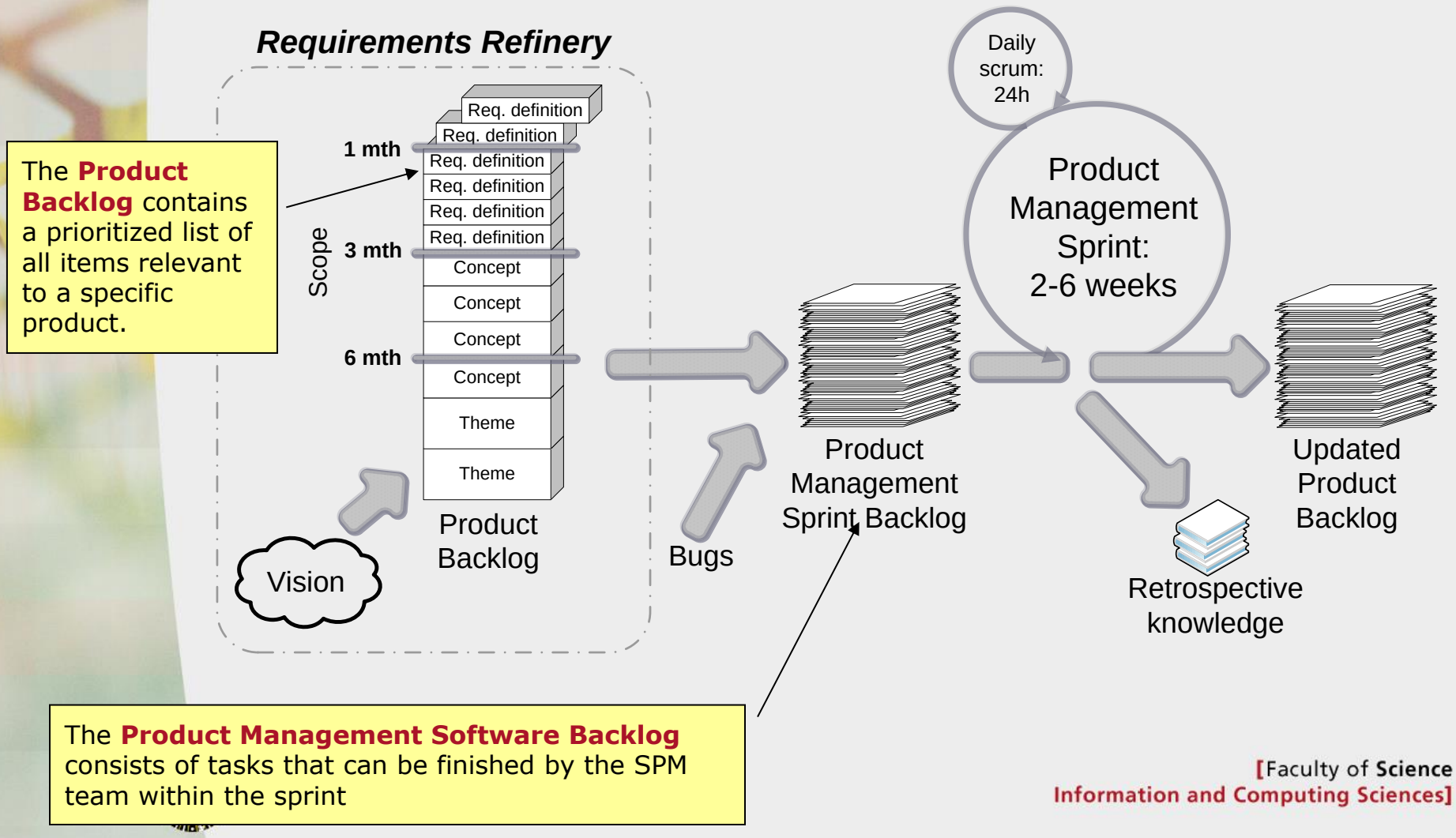
- Reports of questions, complaints or bugs
- Approximately 60.000 reports per year
- Handled by helpdesk and consultancy
- 85% can be solved by referring to the manual
- 15% is a bug or shortcoming

	Incident Reports	PDRs
2005	64,541	15,411
2006	62,981	12,913
2007	56,515	15,346
2008	68,570	17,904



Kabbedijk et al., RE 2009

Scrum extension: Agile Requirements Refinery



Vlaanderen et al., IST 2011

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 = estimated revenue of R_j
a_j = estimated resources need for R_j
Q = number of developers in pool.
T = development period
d(T) = number of working days in planning period

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

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

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

$$x_j \in \{0, 1\}, \qquad \text{for } j = 1, \ldots, n.$$

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_{cap} = atomic transfer capacity unit
y_{ik} = number of transfer capacity units from team i deployed in team k.
α_{ki} = transfer effectiviness factor (from team k to team i).

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

$$\max \sum_{j=1}^{n} v_j x_j \text{ subject to}$$

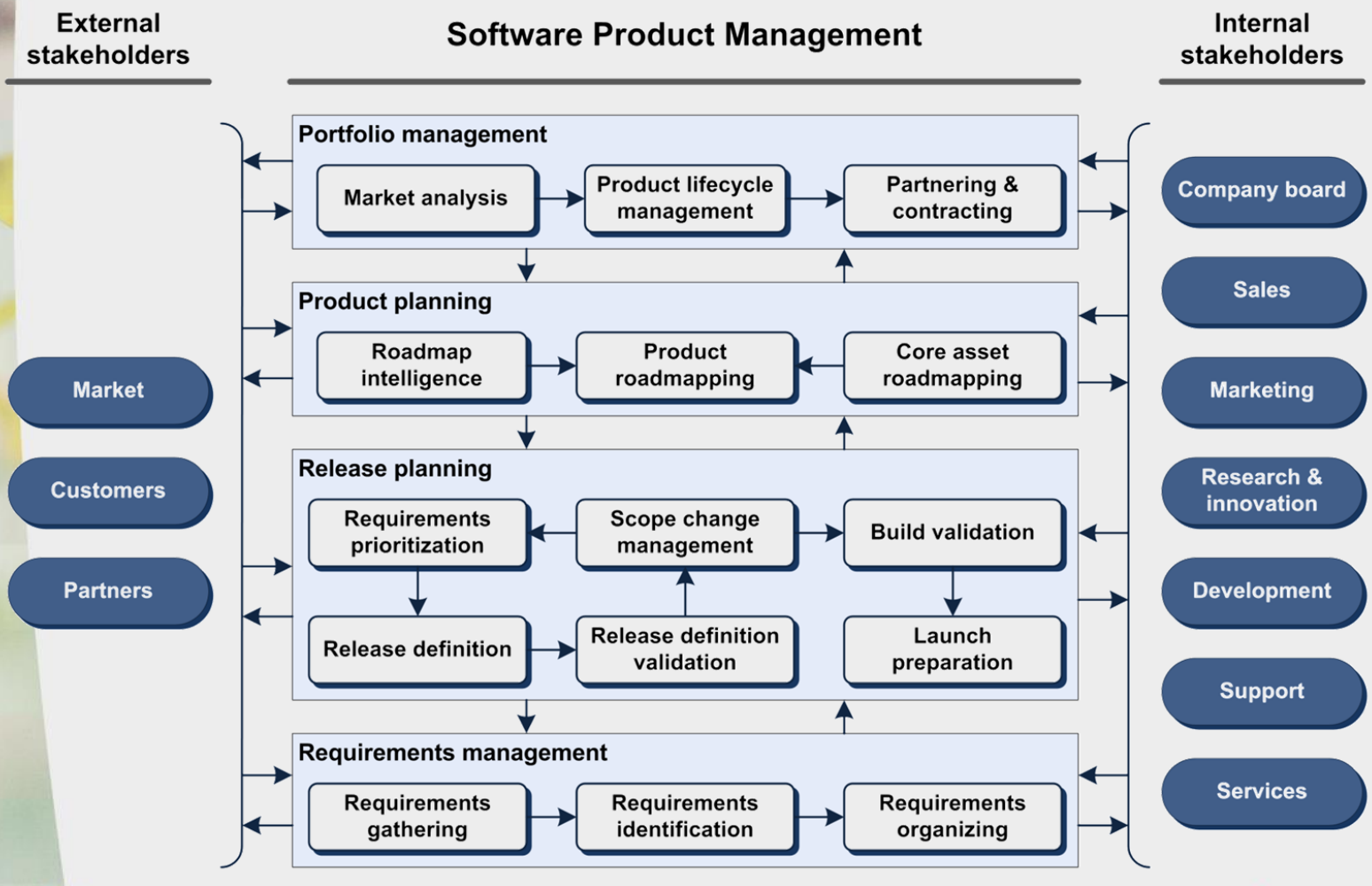
$$\sum_{j=1}^{n} a_{ij}x_j \leq U_{cap}[y_{ii} + \sum_{k:k\neq i} \alpha_{ki} y_{ki}] \qquad \text{for } , i = 1, \ldots, m,$$

$$\sum_{k=1}^{m} y_{ik} = m_i, \qquad \text{for } i = 1, \ldots, m,$$

$$x_j \in \{0, 1\}, \qquad \text{for } j = 1, \ldots, n,$$

$$y_{ik} \text{ nonnegative and integral}, \qquad \text{for } j = 1, \ldots, n.$$

SPM Competence Model



Van de Weerd et al., RE 2006

Maturity profile in Product Management

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

Van de Weerd et al., ICSOB 2010

Maturity analysis of 68 software producers

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

Many organizations have low maturity

Percentage of organizations achieving the level

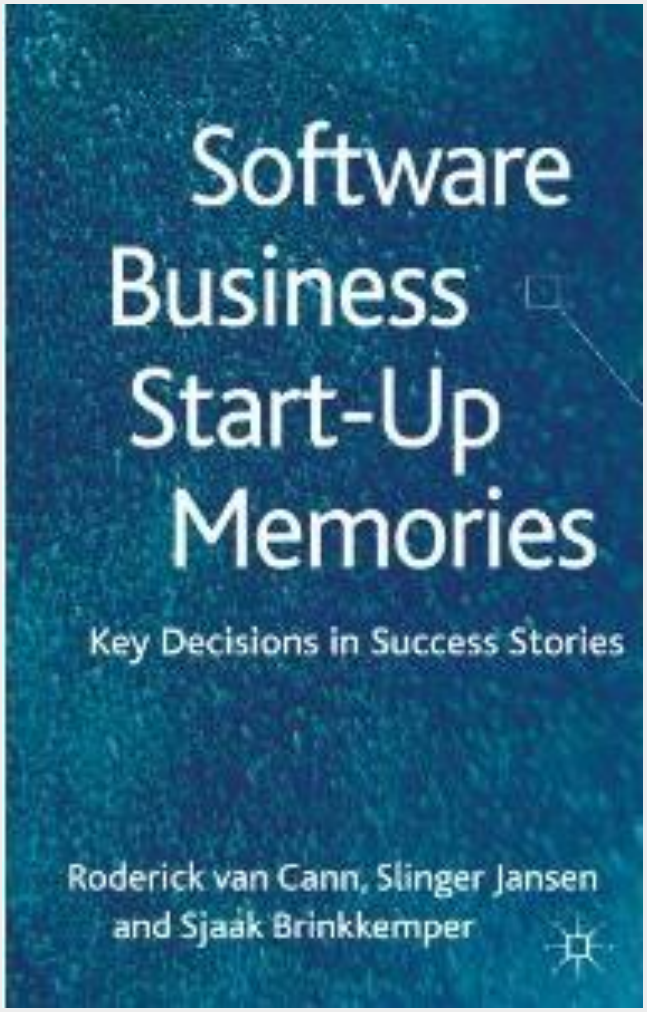
Intriguing data

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Bekkers et al., RE 2012

Key decisions in Software Start-ups



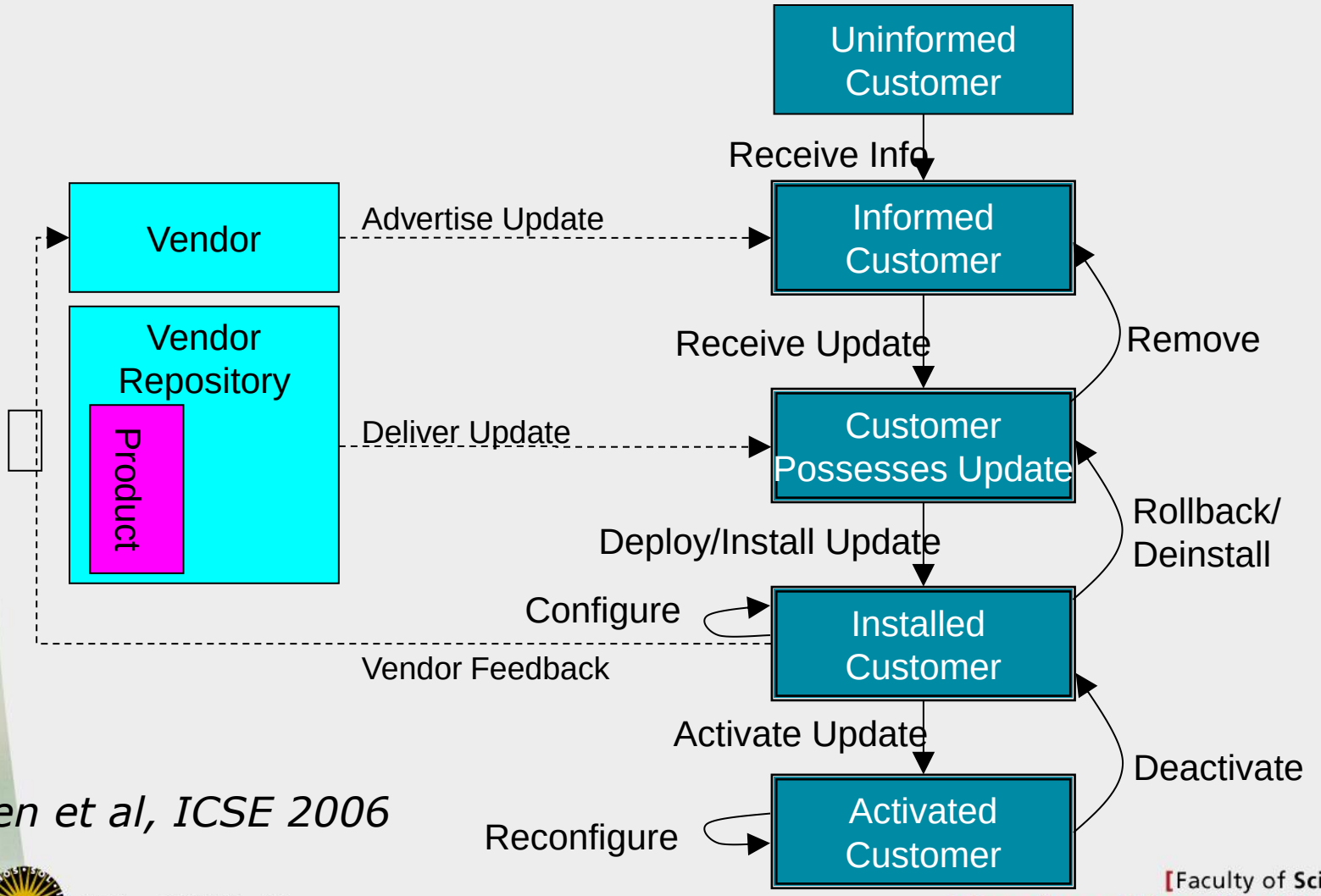
Released: 17 Jan 2013

See Amazon

Outline

- Software products
  - Software types
  - Economic context
  - Research framework
- Software production
  - Software product management,
  - Software ecosystems
  - Incremental method evolution
- Challenges for the research community
- Invitation and discussion

CCU - Customer Configuration Updating



Jansen et al, ICSE 2006

The Catalogue of Customization Realization Techniques

Customization Realization Technique	Latest introduction time	a	b	c	d	e
Model change	Design		✓		✓	
View change	Detailed design		✓		✓	✓
Controller change	Detailed design				✓	✓
System connector change	Architecture design		✓	✓	✓	
System component change	Architecture design	✓		✓	✓	

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

Jansen et al., ICWE 2010

Microsoft Internet Explorer

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

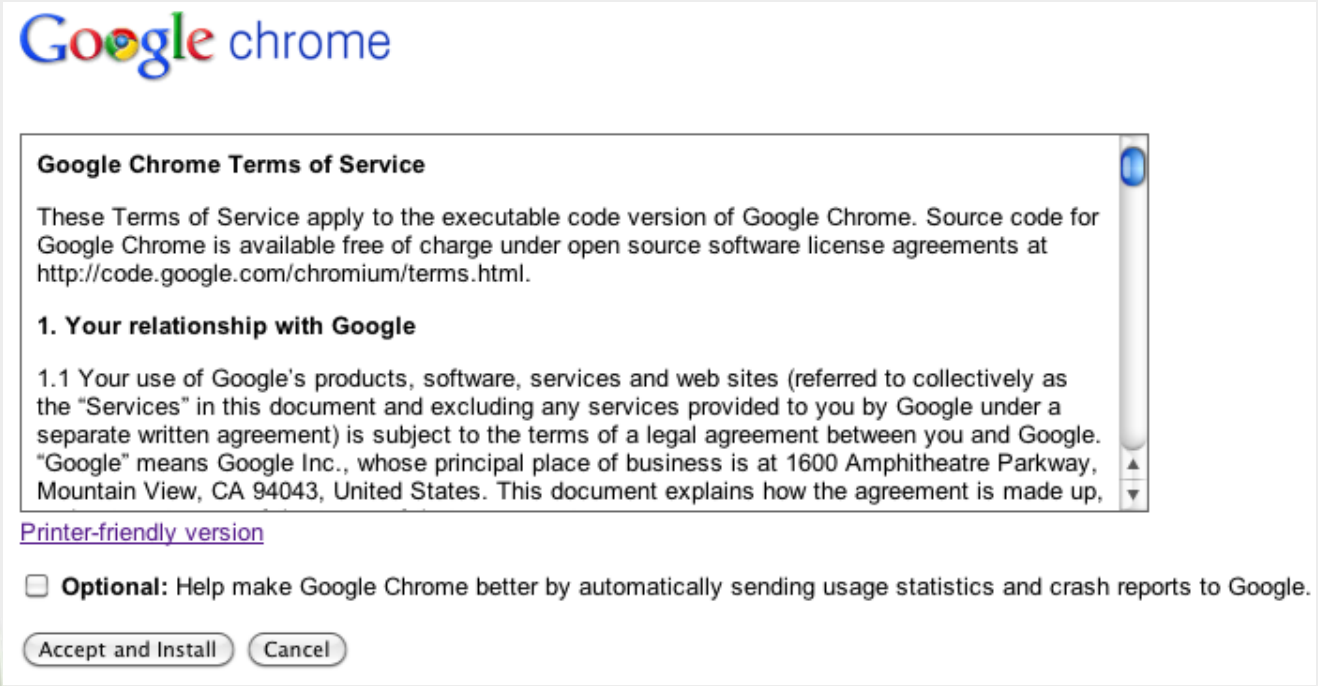
Please tell Microsoft about this problem.

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

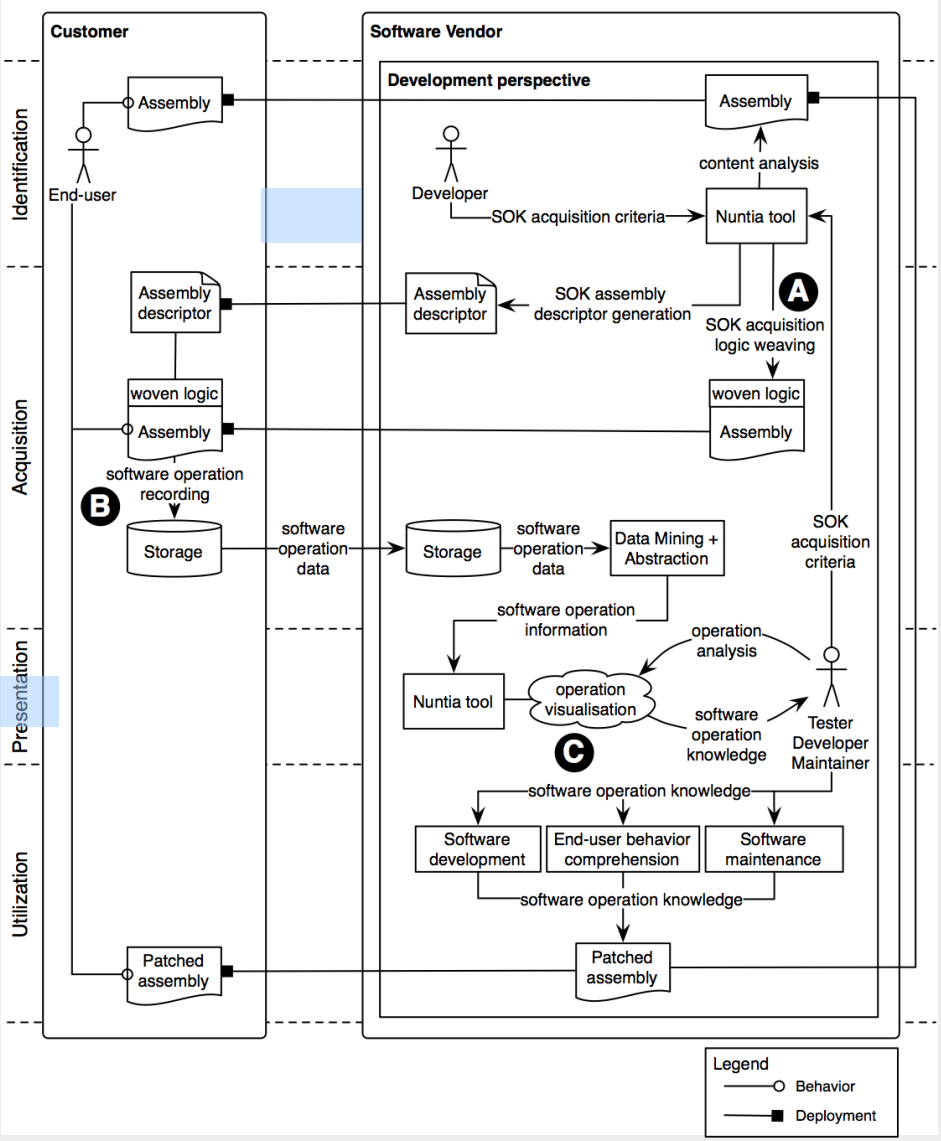
Feedback in Software Supply Chains



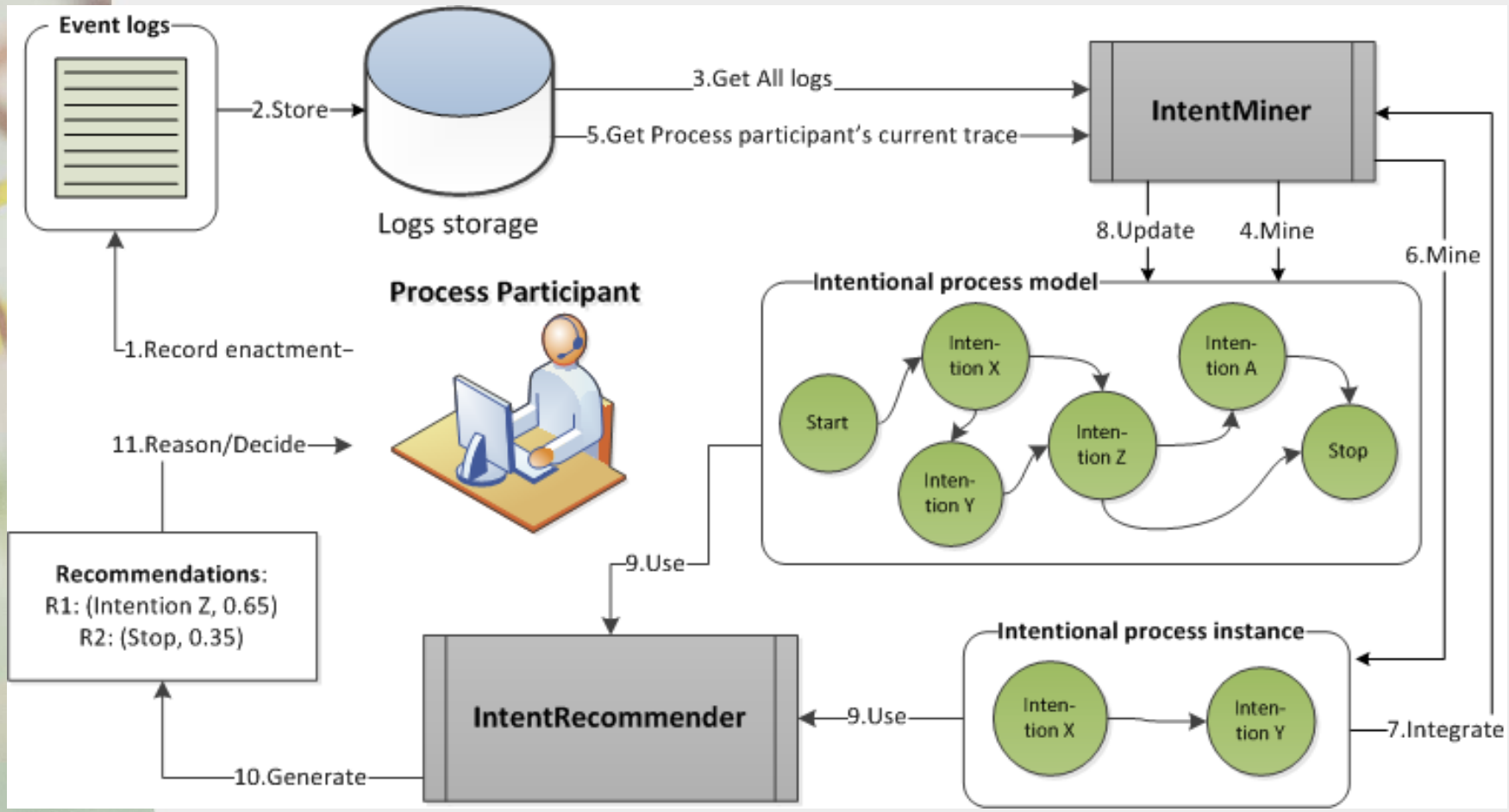
Software Operation Knowledge

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



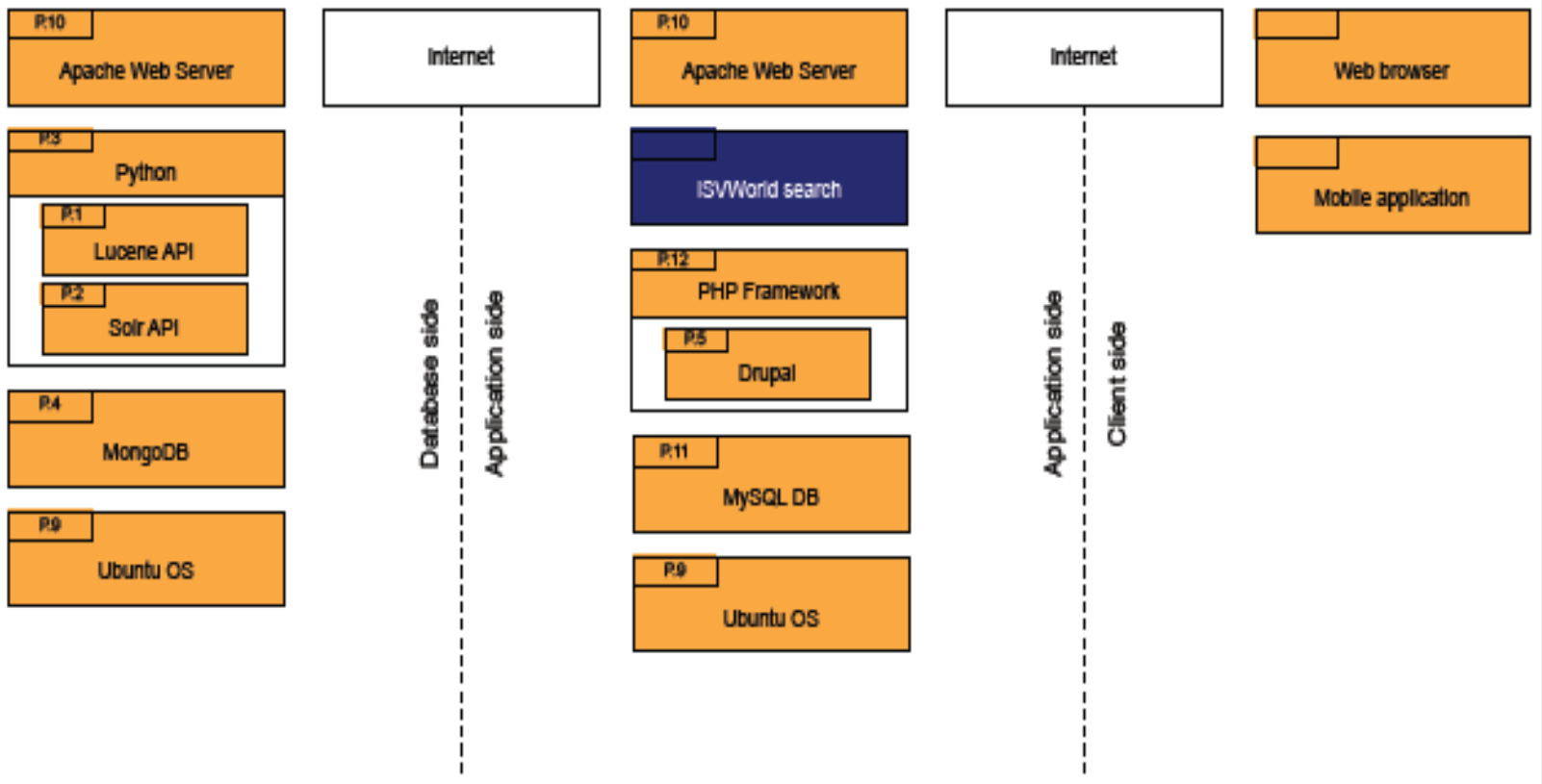
Intention mining in user behaviour



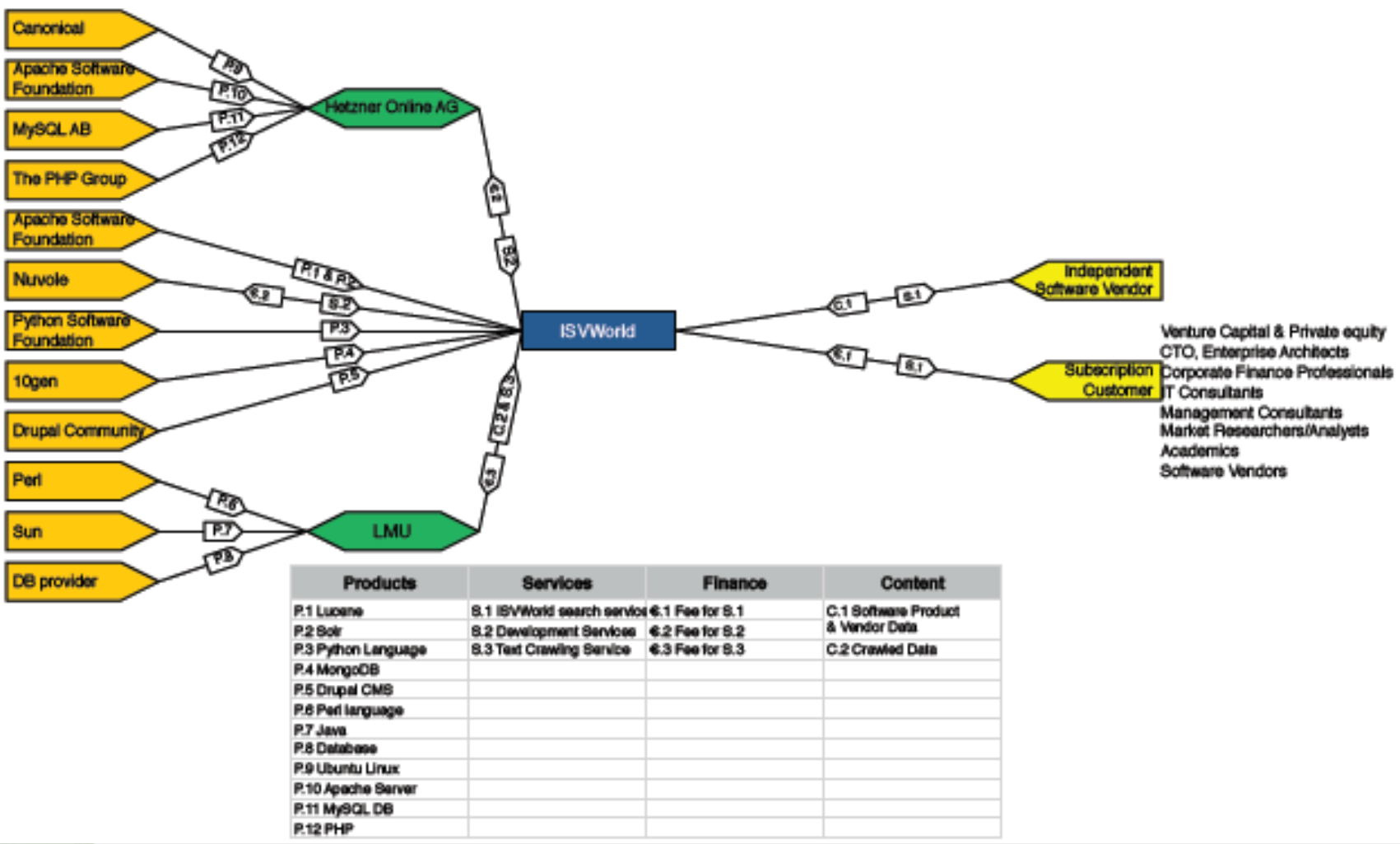
Epure et al., CAiSE 2014

Software Ecosystems modeling

Product Stack: dependency on third party software

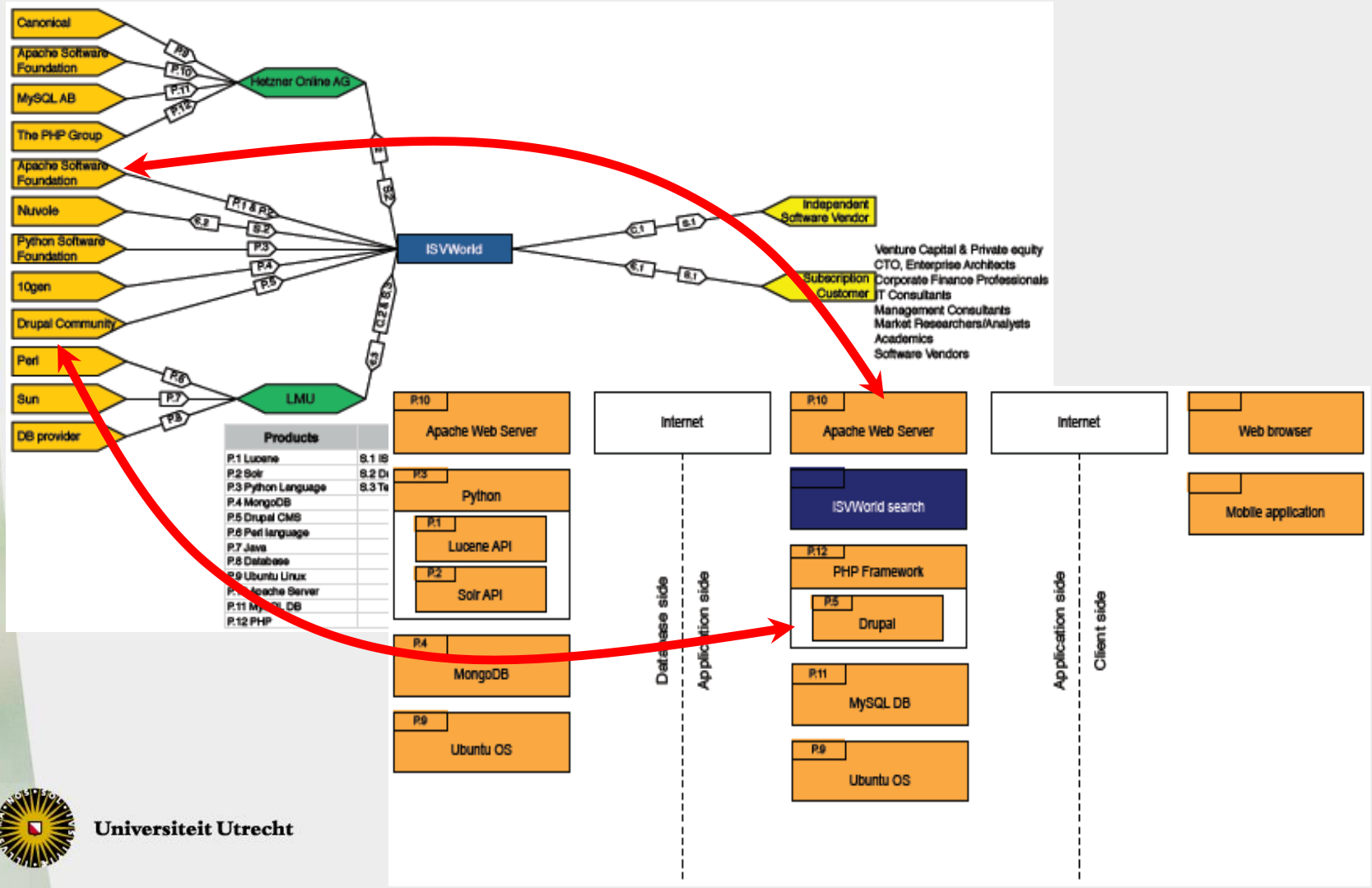


Software Supply Chain

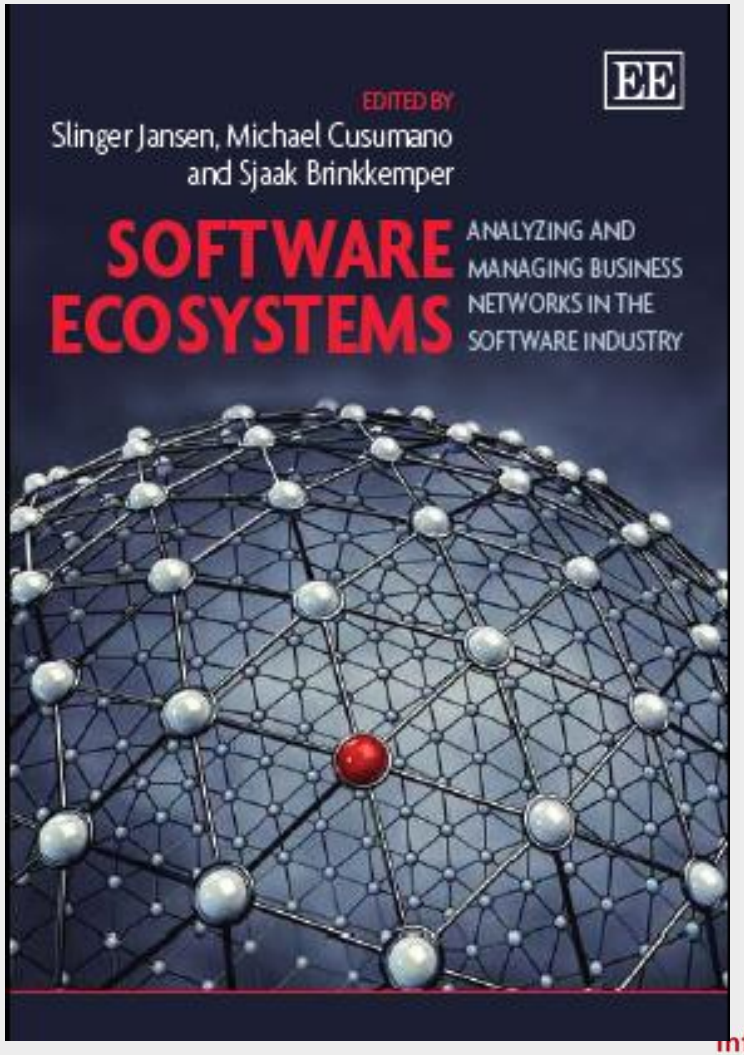


Products	Services	Finance	Content
P.1 Lucene	S.1 ISVWorld search service	€.1 Fee for S.1	C.1 Software Product & Vendor Data
P.2 Solr	S.2 Development Services	€.2 Fee for S.2	
P.3 Python Language	S.3 Text Crawling Service	€.3 Fee for S.3	C.2 Crawled Data
P.4 MongoDB			
P.5 Drupal CMS			
P.6 Perl language			
P.7 Java			
P.8 Database			
P.9 Ubuntu Linux			
P.10 Apache Server			
P.11 MySQL DB			
P.12 PHP			

Managing SECO dependencies



Book on Software Ecosystems

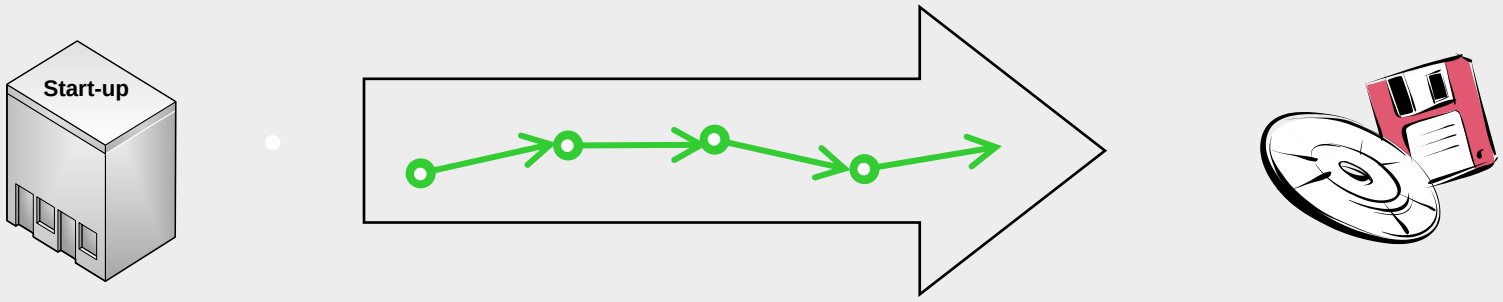


Published March 2013
See Amazon

Outline

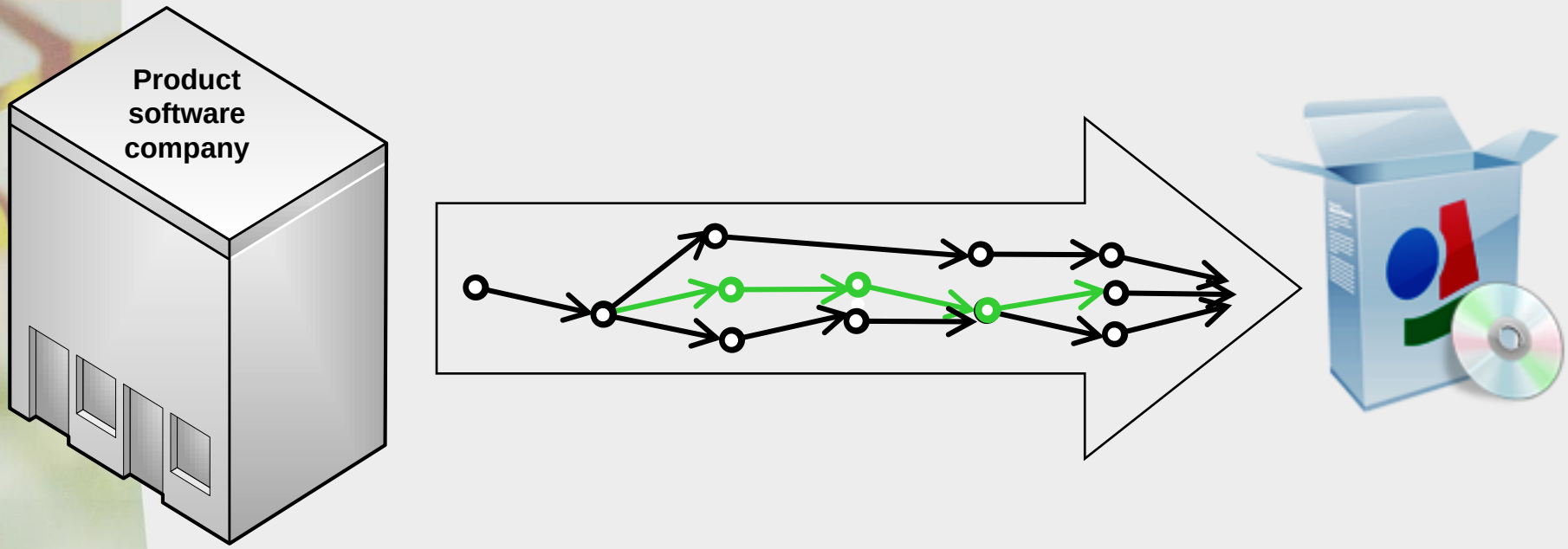
- Software products
  - Software types
  - Economic context
  - Research framework
- Software production
  - Software product management,
  - Software ecosystems
  - Incremental method evolution
- Challenges for the research community
- Invitation and discussion

The software startup



Start with a simple process

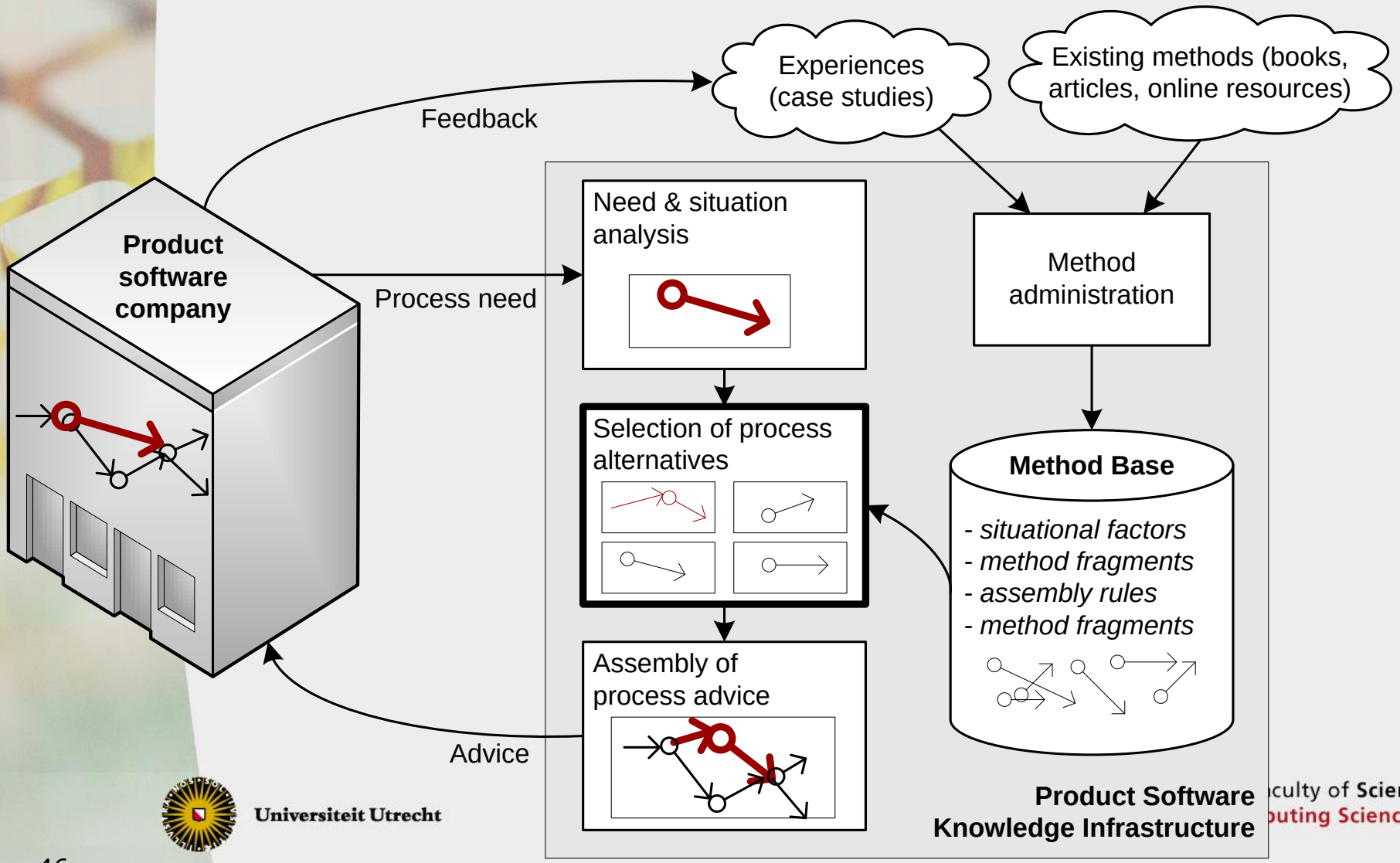
Incremental Process Capability



Gradual more extensive processen are added dependent on:

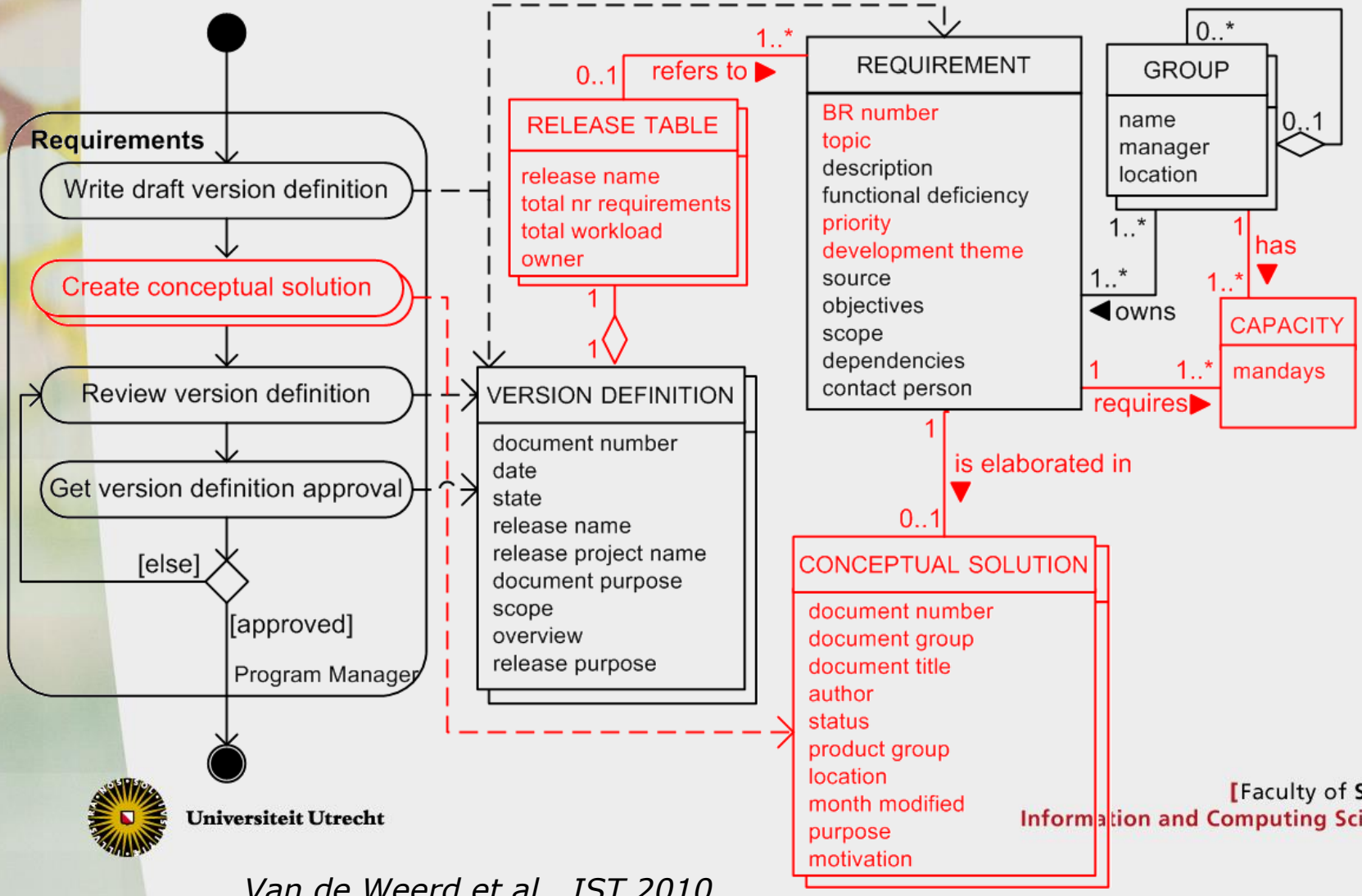
- product
- development culture
- internal maturity
- external stakeholders

On-line Method Engine



Van de Weerd et al., CAiSE 2007

Incremental Method Evolution



Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

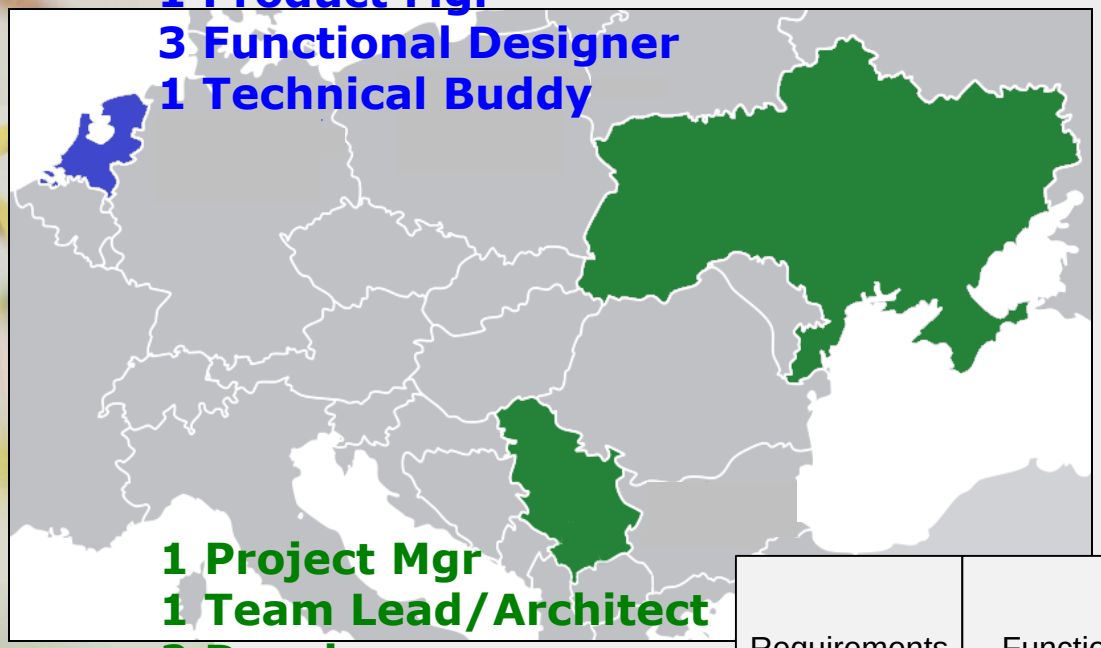
Van de Weerd et al., IST 2010

Formalization of Method Incrementation

- **Definition 3.1**
  The mapping method: $\mathsf{T} \rightarrow \mathsf{M}$, where
  m = method(t) means that the method $m \in \mathsf{M}$ is the valid method at time t.
- **Definition 3.6**
  A method increment is a method fragment
  $f \in \mathsf{F}$ such that $\exists i$ contains(f,method(t_i)) $\wedge$ ¬contains(f,method(t_{i-1}))
- **Definition 3.7** The mapping insert
  $\mathsf{F} \times \mathsf{M} \rightarrow \mathsf{M}$: 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

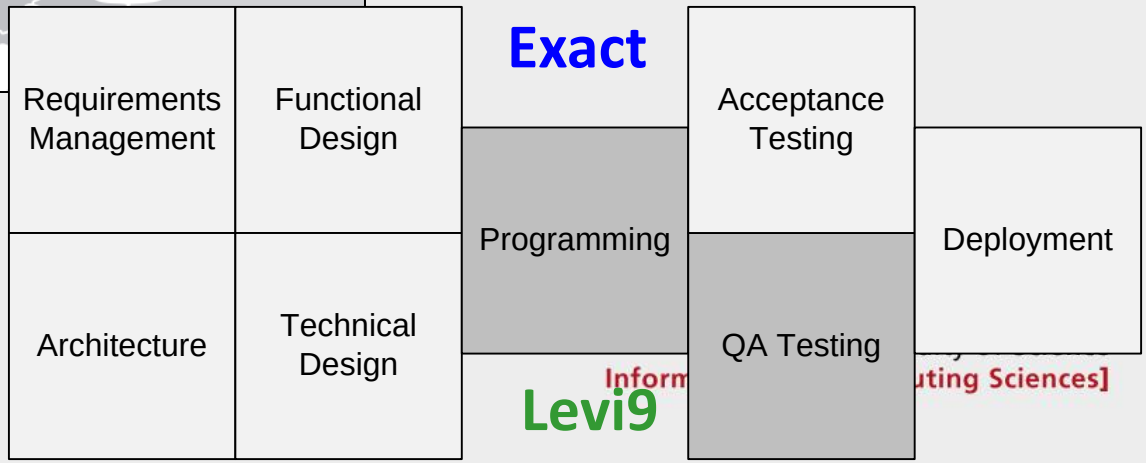
- **Rule 3.1** Insertion of concepts
  insert(c,m_i) = m_{i+1} $\Rightarrow$
  ¬concept(c,m_i) $\wedge$ concept(c,m_{i+1})
- **Example**
  insert(RELEASE TABLE,BaanIncr2) =
  BaanIncr3 $\Rightarrow$
  ¬concept(RELEASE TABLE,BaanIncr2) $\wedge$ concept(RELEASE TABLE,BaanIncr3)
  The release table was not a concept in BaanIncr2 and therefore inserted to get BaanIncr3

Collaboration patterns



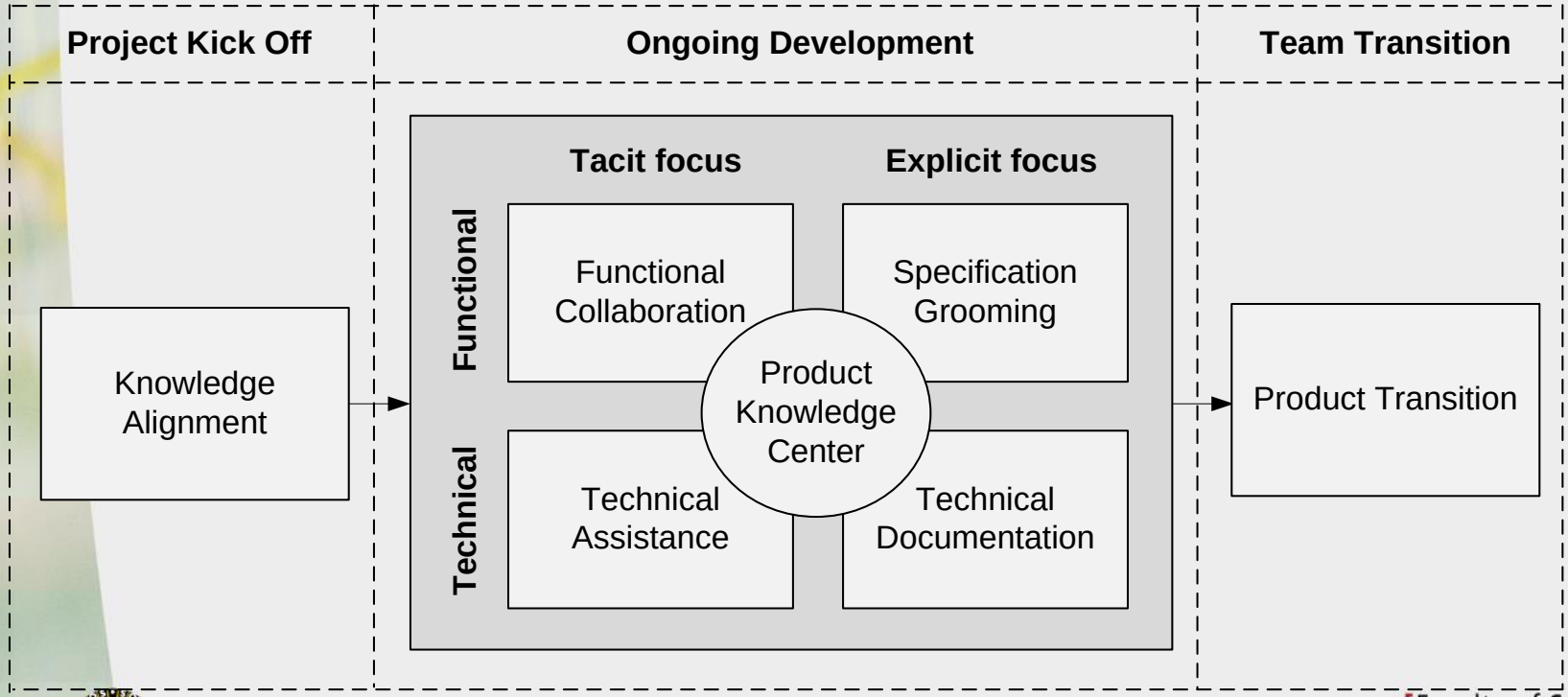
1 Sub-Team Lead/Architect
3 Developers
1 QA Manager
1 Tester

Kristjansson et al., JKE 2011



Universiteit Utrecht

Product Knowledge Center



Outline

- Software products
  - Software types
  - Economic context
  - Research framework
- Software production
  - Software product management,
  - Software ecosystems
  - Incremental method evolution
- Challenges for the research community
- Invitation and discussion

Research differences

Software Engineering

1. One customer
2. Any domain
3. Different projects
4. Pool of workers
5. Changing collaboration patterns
6. Main players: project mgr, systems architect, requirements engineer
7. . . .

Software Production

1. One market
2. Same domain: domain experts
3. Release projects
4. Fixed organization
5. Fixed collaboration
6. Main players: CTO, software architect, product manager
7. . . .

Research differences

Software Engineering	Software Production
1. One customer	1. **One** market
2. Any domain	2. **Same** domain: domain experts
3. Different projects	3. Release projects
4. Pool of workers	4. **Fixed** organization
5. Changing collaboration patterns	5. **Fixed** collaboration
6. Main players: project mgr, systems architect, requirements engineer	6. Main players: CTO, software architect, product manager
7. . . .	7. . . .

Universitei

Great opportunities for experimental SE research !!

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?

Invitation and Discussion

SINC

Software Industry Conference, 12 March 2015

See http://www.softwareindustryconference.com/

Call: 4 slots with academic presentations

Questions?

- Papers see: www.dblp.org
- Email: S.Brinkkemper@uu.nl
- Contact me in case you are interested in joining a national or European project in Software Production
- Upcoming: Vacancies for assistant professors