



# Customer Configuration Updating

ICT Entrepreneurship

Prof. Dr. Sjaak Brinkkemper

Dr. Slinger Jansen

*Aim of this presentation: “To improve the understanding of ICT Entrepreneurship students of the CCU process and to show that this is a problem they should be concerned with now to enable a better future.”*

Utrecht University

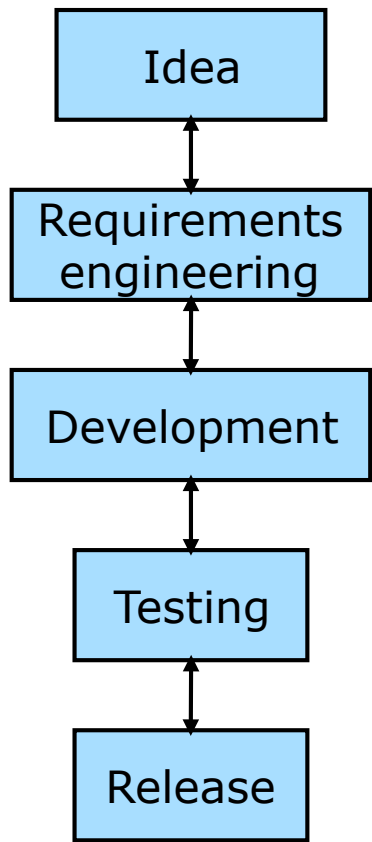
## Context of Customer Configuration Updating



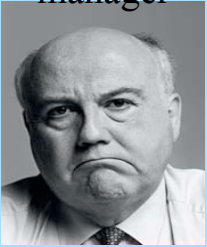
- It's about product software: To date product software is a **packaged configuration of software components** or a software-based service, with auxiliary material, which is released for and traded in a specific market [1].
- So where is CCU in the product development picture?

[1] L. Xu and S. Brinkkemper, "Concepts of product software: Paving the road for urgently needed research," in *First International Workshop on Philosophical Foundations of Information Systems Engineering*. LNCS, Springer-Verlag, 2005





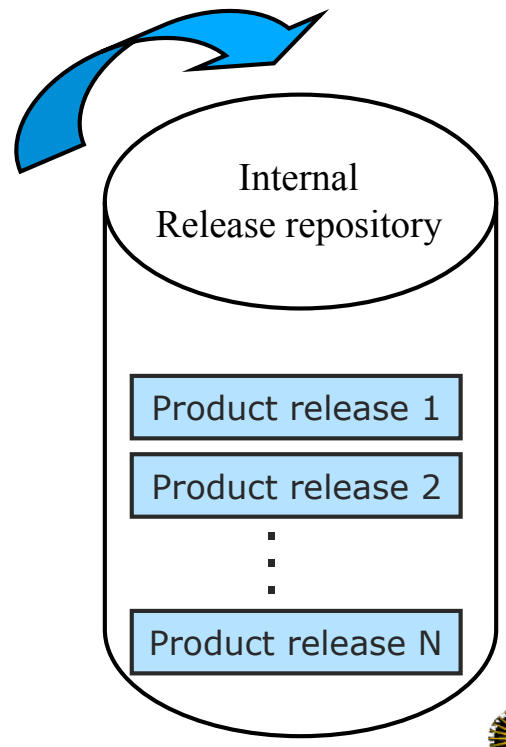
Unhappy business manager

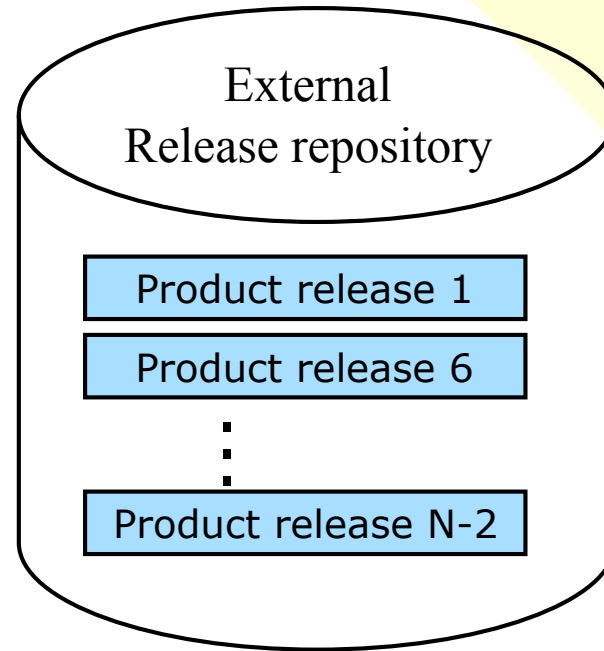
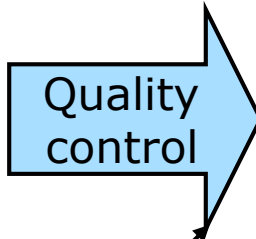
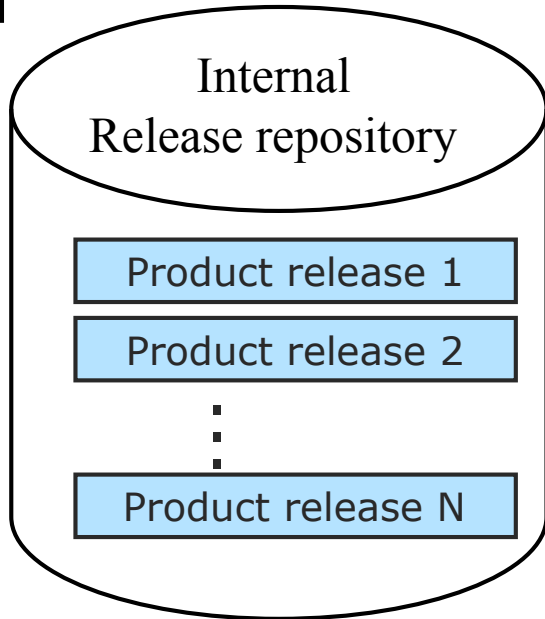
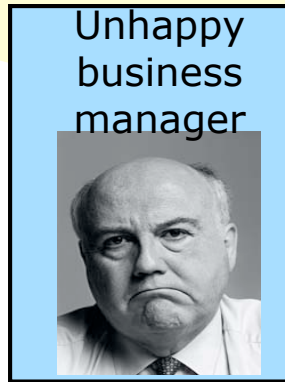
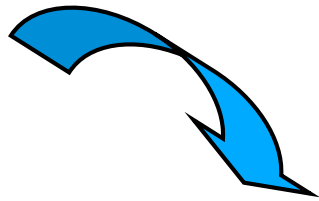
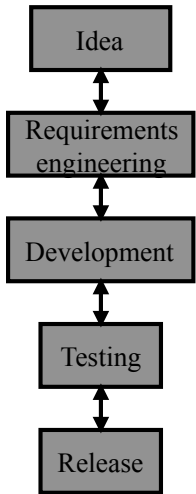


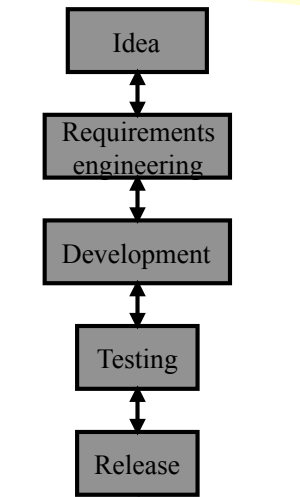
Happy Developer!



No customers yet





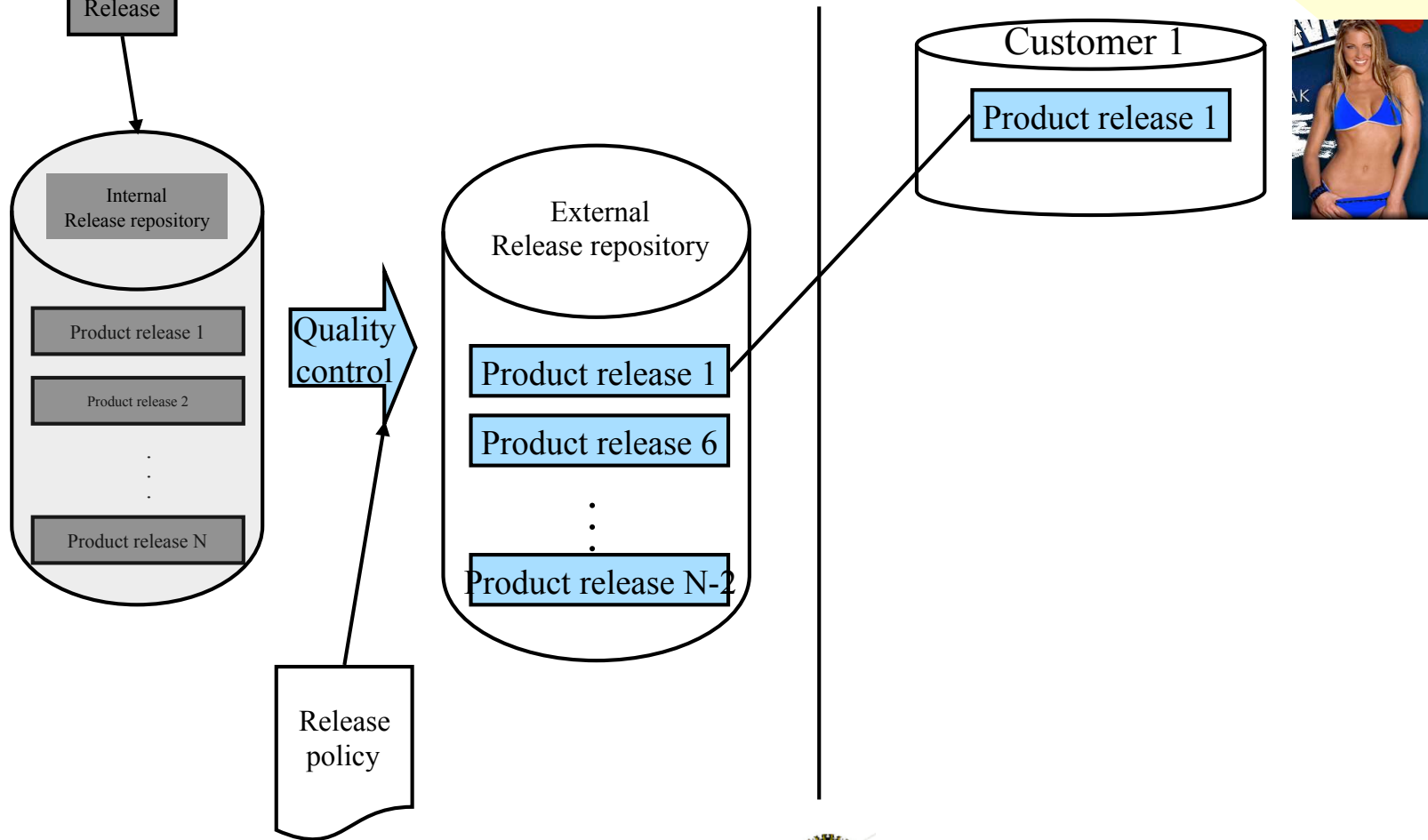


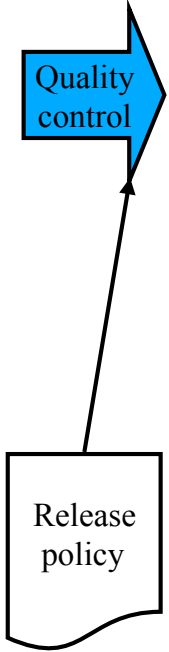
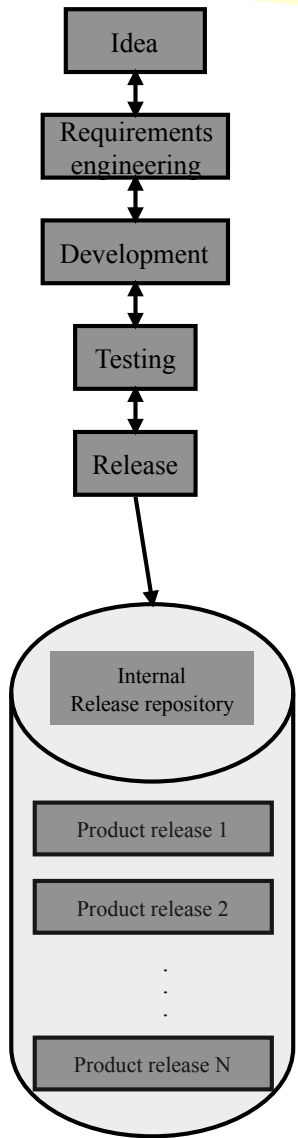
Mildly happy business manager

Very happy support engineer

Happy Developer!

One happy customer



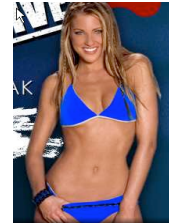
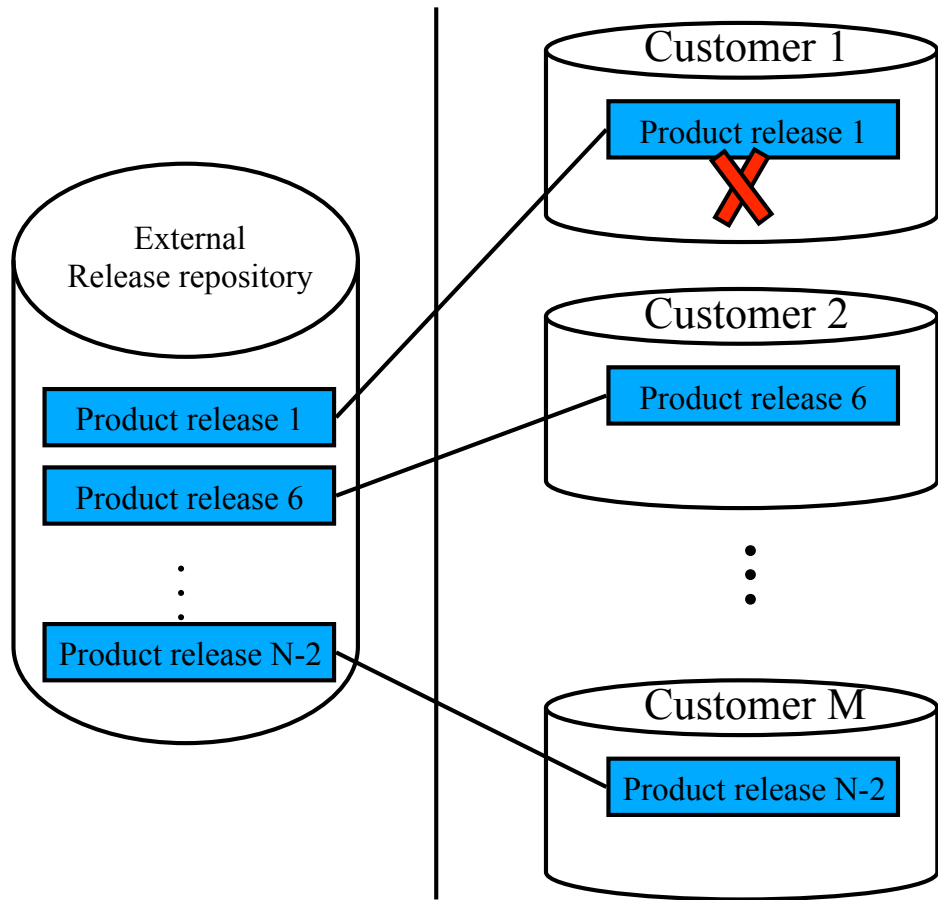


Very happy business manager

Unhappy support engineer

Unhappy Developer!

Some happy some unhappy customers



What to do next?



## The Start-up problem

- Problems
  - Supporting multiple versions in the field
  - No way to conveniently update, a consultant needs to be sent over
- Solutions
  - Send over simple file patches
  - Buy a third-party installer and update service that can apply customly created updates
  - Build your own updater
- This example describes the situation for most product software vendors at some point in time





## So what's the area of study?

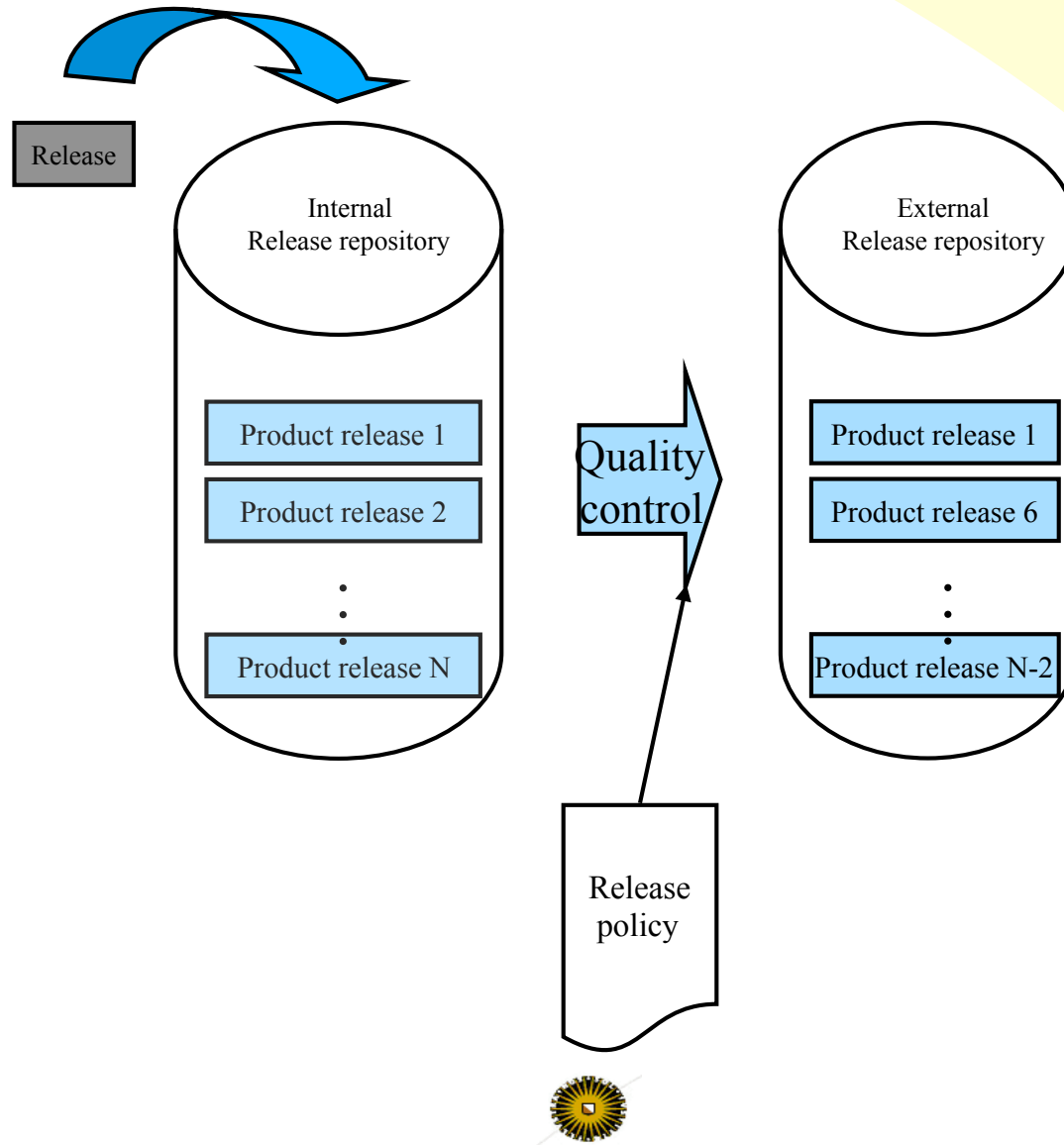
- We are looking at the processes of
  - Release
  - Package
  - Delivery
  - Deployment
  - Usage
  - Activation
  - Licensing
- These are all part of software product lifecycle management
- **Definition:** The customer configuration updating is defined as the combination of the vendor side **release** process, the product or update **delivery** process, the customer side **deployment** process, and the **activation** process. [2]
- Who are the stakeholders?

[2] Slinger Jansen and Sjaak Brinkkemper: *Definition and Validation of the Key Process Areas of Release, Delivery and Deployment of Product Software Vendors: turning the ugly duckling into a swan*, Technical Report UU-CS-2005-041, Institute of Information and Computing Sciences, Utrecht University, Utrecht, the Netherlands, September 2005





# Release process: Visual representation

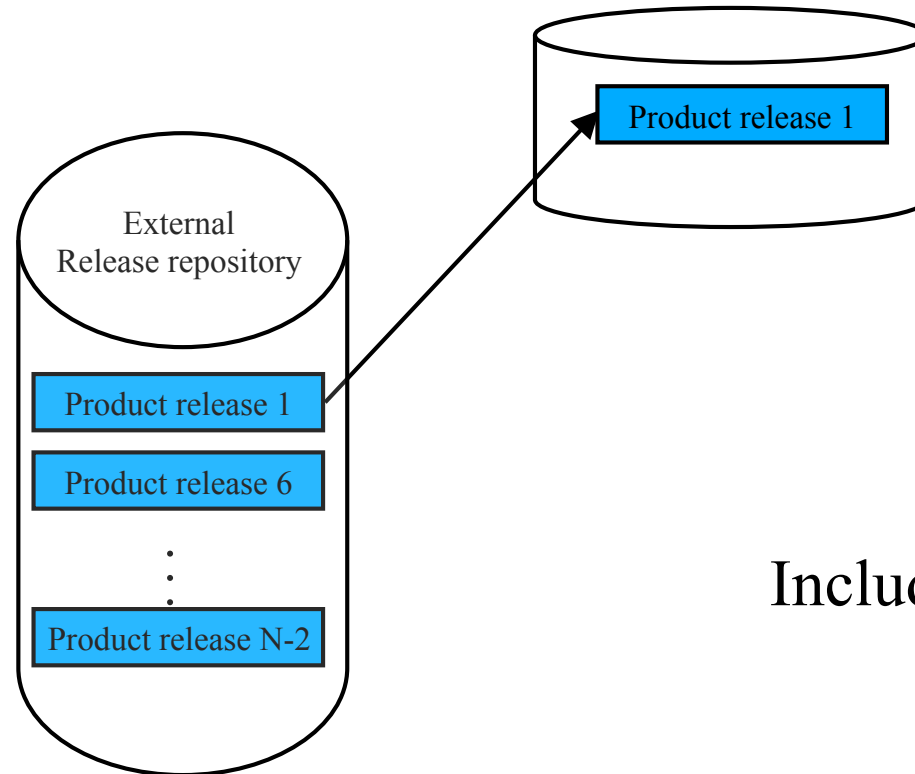


# Delivery and Deployment: Visual representation



Product Software Vendor

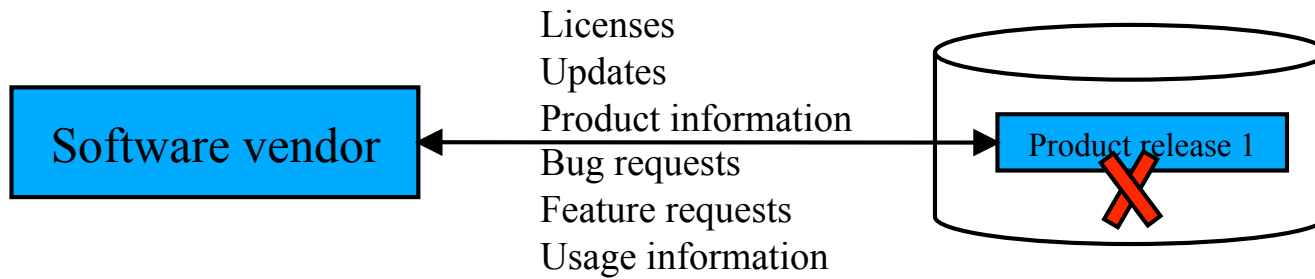
Customer



Includes updates!

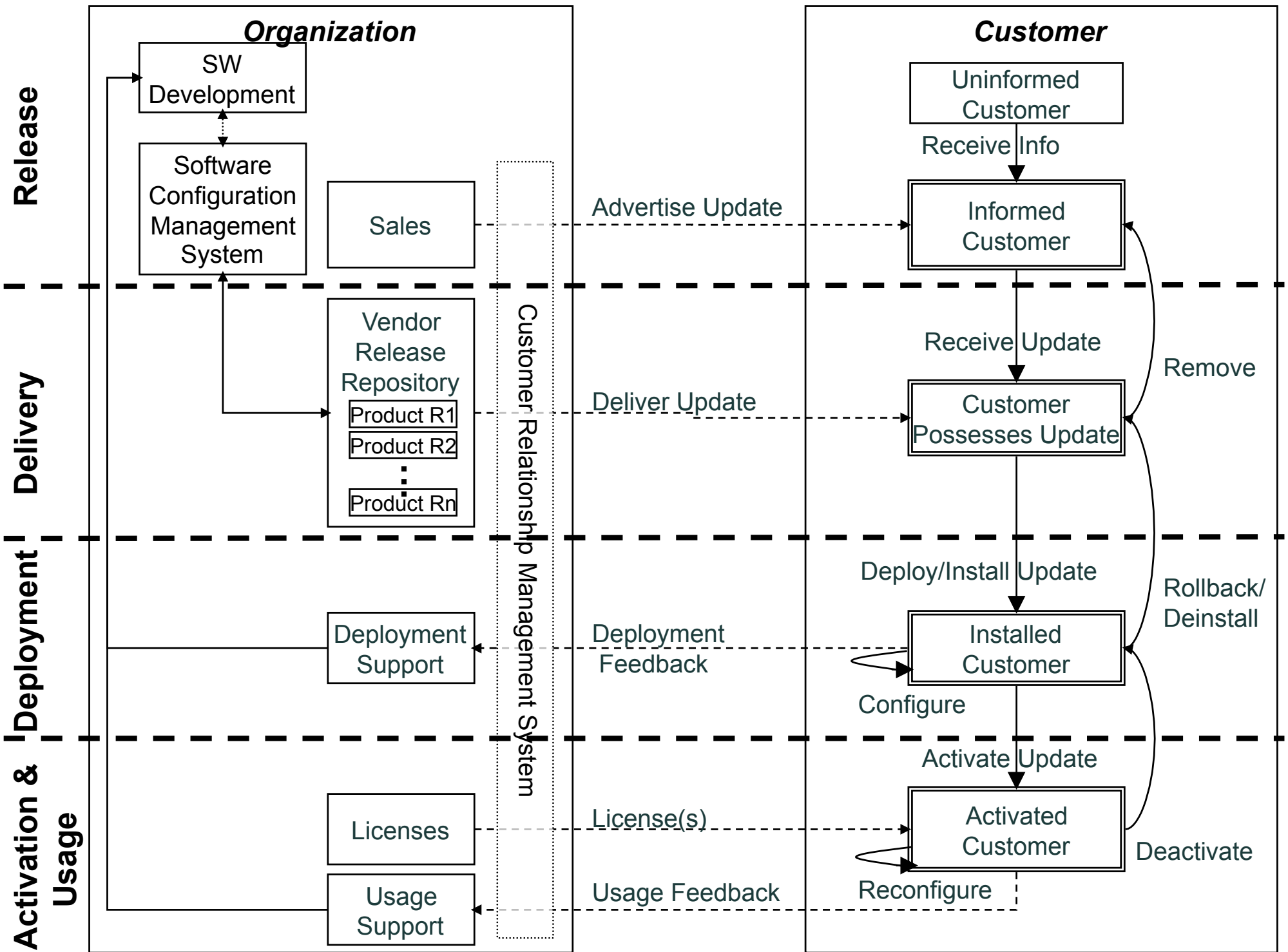


# Delivery and Deployment: Visual representation



Includes updates!





## Release Management



- Why do release management?
  - Quality control (developers have a tendency to do late modifications)
  - Complexity control (if you're only supporting a limited number of configurations, life is much easier)
  
- Subquestion: Why do software configuration management?
  - Development cooperation (multiple developers working on one product)
  - Development management (change tracking, automatic testing, etc.)
  
- So what solutions are found in the field?
  - Visual Source Safe (MS)
  - CVS (by far most common, but has renaming issues)
  - Dedicated PDM Systems (see Exact Case)
  - Subversion (up and becoming the defacto standard)



## Packaging



- The process of assembling **all artifacts** (digital and physical) to be delivered to a customer such as
  - Manuals
  - Languages (for internationalization)
  - executables and binaries
  - CDs/DVDs
  - Promotional material
  - License key
  - ... and much much more
- Many companies use simple checklists, but larger ones use proprietary or conventional **Product Data Management** (PDM) systems



## Software Delivery



- Facets of delivery are
  - Network **infrastructure**
  - Delivery format (files, patches, component packages, etc)
  - Package format (setup.exe, MSI, rar, NIX, etc)
  - Physical delivery format (CD, Floppy, DVD, USB stick, etc)
  
- Some interesting examples
  - Client server architectures
  - BitTorrent
  - USB **licensing** software
  - Java-based deployment tools



## Deployment



- Deployment steps
  - Install, de-install, (re-)configure, rollback, update
- Deployment format
  - Component
  - Patch
  - Plug-in
  - setup.exe
- Deployment constraints
  - Operating system
  - Other components
  - Hardware (diskspace, CPU, extra devices)
  - Earlier versions of the component
  - Customisations
- Novel techniques
  - Automatic constraint resolving
  - Run-time updates



## Activation and Licensing



- Types of **activation technologies**
  - Hardware locks
  - Software locks
  - On-line activation
  - etc.
- Types of **licenses and payment**
  - Pay per module
  - Pay per use
  - Pay per bit (data storage)
  - Pay per record entry
  - Pay per transaction
  - Pay per user
  - Pay per floating user
  - etc.
  
- Which of these is interesting for your product?



“Should we care that our customers sometimes buy something they don’t need?” –*an unnamed business manager*



## Usage

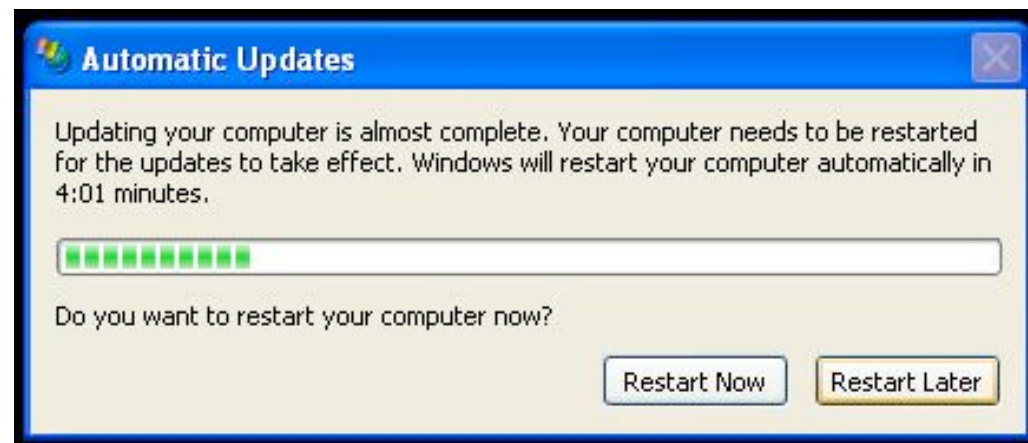


- During usage and configuration a number of things can improve the customer's experience
  - Customer data can be **analyzed** for product improvement
  - Customer **usage data** can be used for hotspot analysis
  - Customer **error reports** ("do you wish to send a feedback report to MS?")
  - In-product questions and bug reports (similar to screen shots)
  - **Purchase** button in product (I need this functionality NOW)
  - ...



So that's all there is?

- The process is well defined
  - The problems are well defined
  - Customer satisfaction levels are clear
  - Possible solutions to common problems are well defined
  - Many different frameworks exist
- 
- So what's the big deal?! Why did Sjaak hire me?!



The problem is more complex...

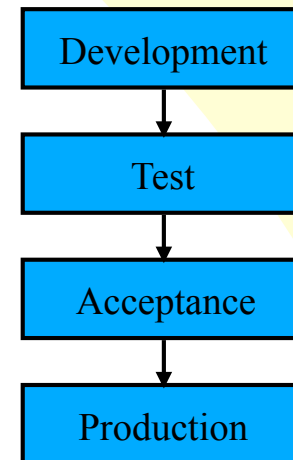
- Different variable releases
- Different features per revision
- Changing data models
- Customisations
- Different operating systems
- Different dependencies per product module
- Different customers to serve
- etc.





## What are customers?

- Original view of customer **too simplistic**, customers are
  - Users
  - System managers
  - Business managers
  - Security personnel
  - Other **components**
- Different ways of handling are required, for example:
  - **Cordys**
  - ES DOS customers
  - Itil DTAP
  - etc.

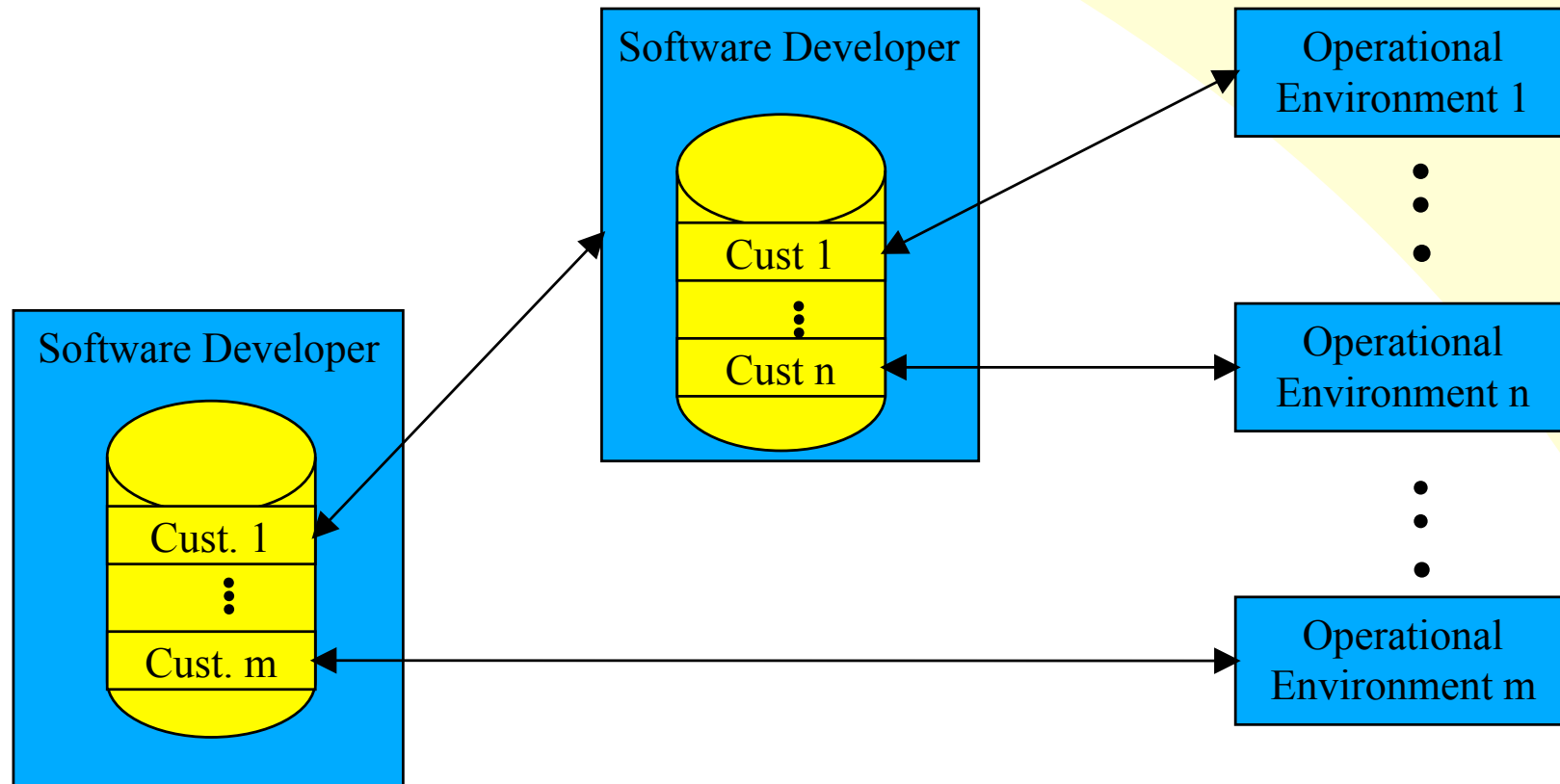


“Our customers are always fighting amongst themselves. Developers need the newest tools, business managers want the newest features, security experts want the latest patches with proven technology, and system managers just want less work.”





## Possibly part of a Software Supply Network



[Slinger Jansen](#) and [Wilfried Rijsemus](#): Balancing Total Cost of Ownership and Cost of Maintenance Within a Software Supply Network , proceedings of the IEEE International Conference on Software Maintenance ([ICSM2006](#), [Industrial track](#)), Philadelphia, PA, USA, September, 2006.



## And what about re-use?

- Integration of products and services
- All with different versions (and possibly differently versioned protocols)
- Operative question: How to manage such relationships?
  - Who gets a new version?
  - When?
  - What quality?
- What types of SLA's are established?



Knowledge Pull

Supplier Interaction

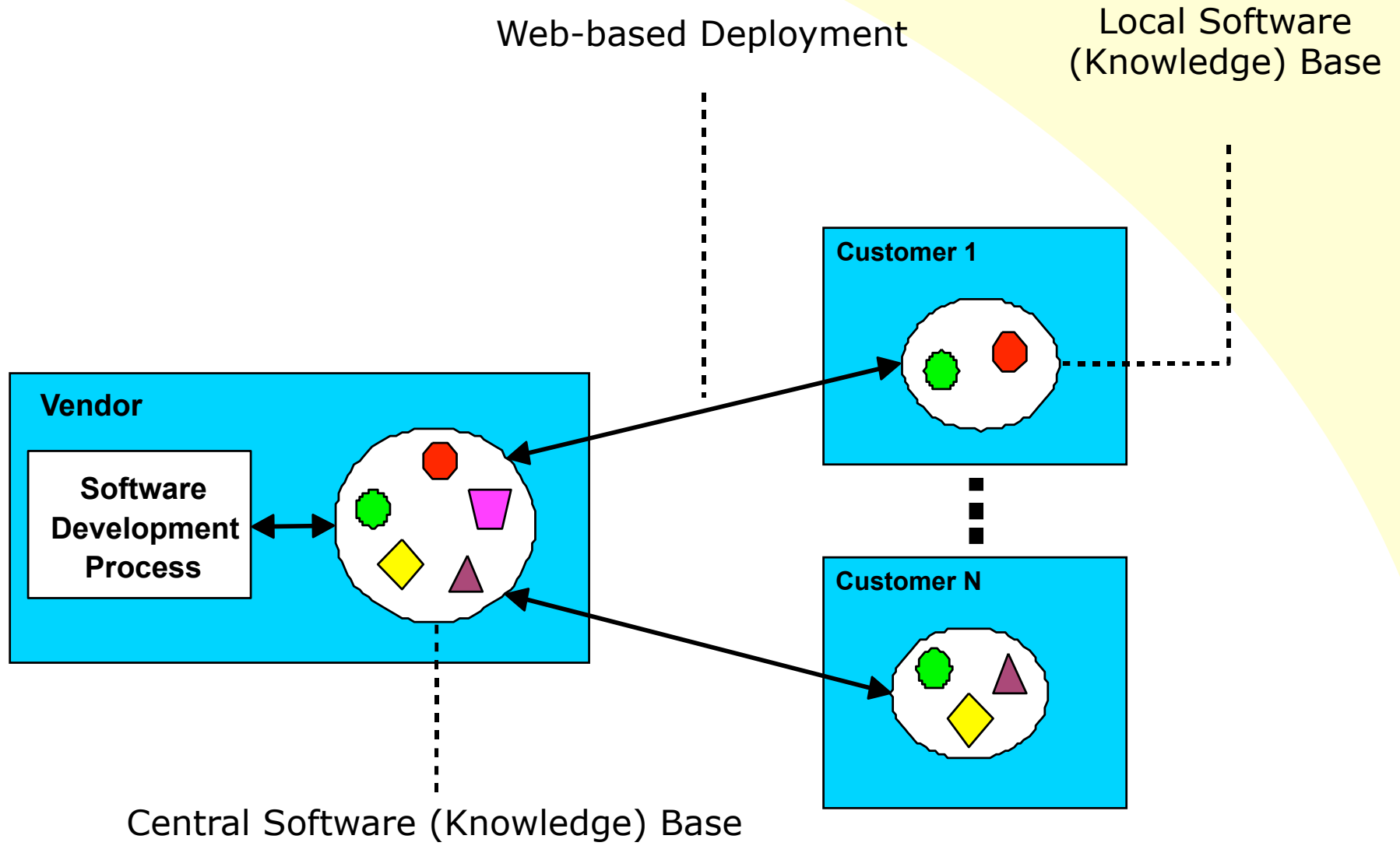
Case	ERPVendor				CADVendor					
	MS	OTS	WC	PDFC	CR	SS	MS	IC	AC	DWG
Component Provider										
Access to source code		Y	Y					Y		
Access to early releases	Y	Y		Y				Y	Y	
Access to release planning	Y	Y		Y		Y		Y	Y	
Access to online dev portal	Y	Y				Y		Y	Y	Y
Manual feedback forwarding	Y	Y	Y	Y		Y		Y	Y	Y
Cooperative Development		Y						Y		
Contact with sales department	Y	Y		Y		Y	Y		Y	
Contact with developers		Y				Y		Y	Y	Y
CEO level contact		Y						Y		Y
Contact with Helpdesk	Y	Y	Y	Y	Y	Y		Y	Y	Y
Company Size	79,000	20	20	10 <sup>1</sup>	6,200	1,100	79,000	40	5,169	40 <sup>1</sup>
Relationship Type	(x)	(xi)	(xi)	(xi)	(xi)	(xi)	(xi)	(x)	(x)	(xi)
Sees its customers as	(i)	(iii)	(iii)	(iii)	(iii)	(iii)	(iii)	(i)	(i)	(iii)

## The solution to all our problems (alright, my PhD)



- A distributed software knowledge base that stores all information on
  - Customers
    - Customizations
    - Configurations
    - Licenses
    - Other components
    - Error reports
    - etc.
  - The Vendor's products
    - Required components
    - Available patches
    - Solved bugs
    - Possible configurations
    - New products and releases
    - etc.



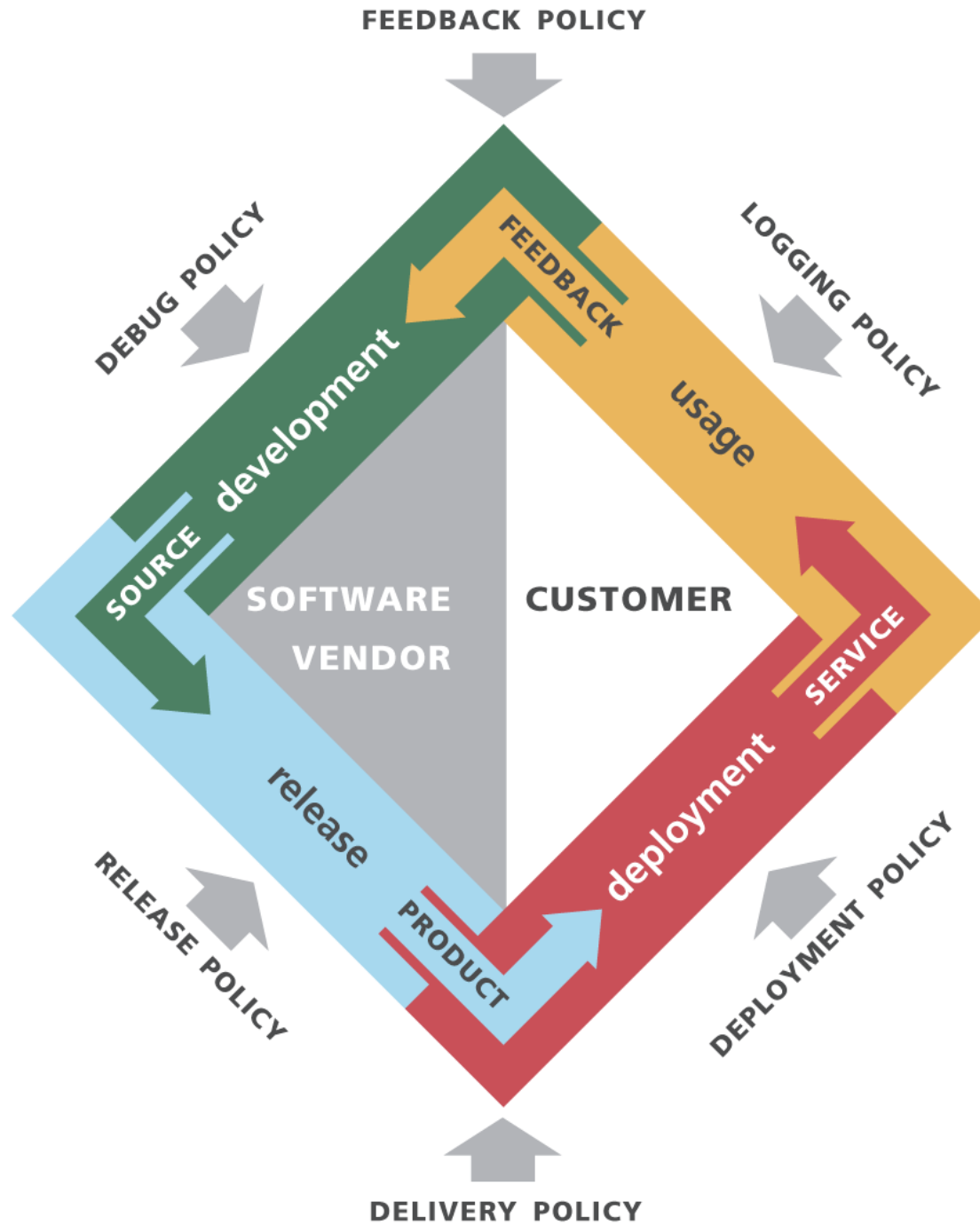




## Vision: Continuous Customer Configuration Updating

- **Continuously** release software after every change to a release repository
- **Continuously** (be able to) release to customers
- **Continuously** (be able to) deploy an update (run-time?)
- **Continuously** change your software
  
- You should at least strive for continuous release!







## Research approach

- How to prove proper knowledge management can improve
  - Customer experience?
  - Ability to handle **large amounts** of customers (growth potential)?
  - Shorten maintenance cycle?
- Find ways to **measure** these
- Show that companies (by doing case studies and a national questionnaire) are already putting the **SKB into practice** (Exact, Cordys)
- Improve on the idea by introducing **my own tools**
  - Delivery tool called PHEME
  - Product deployment tool that deals with features





## Case Study Approach

- **Exploratory** case studies at **product software companies**
- Interview sessions with personnel
- Two interviews per interviewee (1h each)
  - 1. Exploratory
  - 2. Cross checking observations
- Case study **protocol** that has been applied to six cases to date
- Case study **knowledgebase**
- Peer **reviewing** of process and knowledge acquisition



## Planon

- A software vendor that builds “building and property management software”
- 160 employees -> 900 customers
- Four countries
- Currently trying to build a J2EE version of their Delphi product



Overview


CC Updating

Issues


Solution

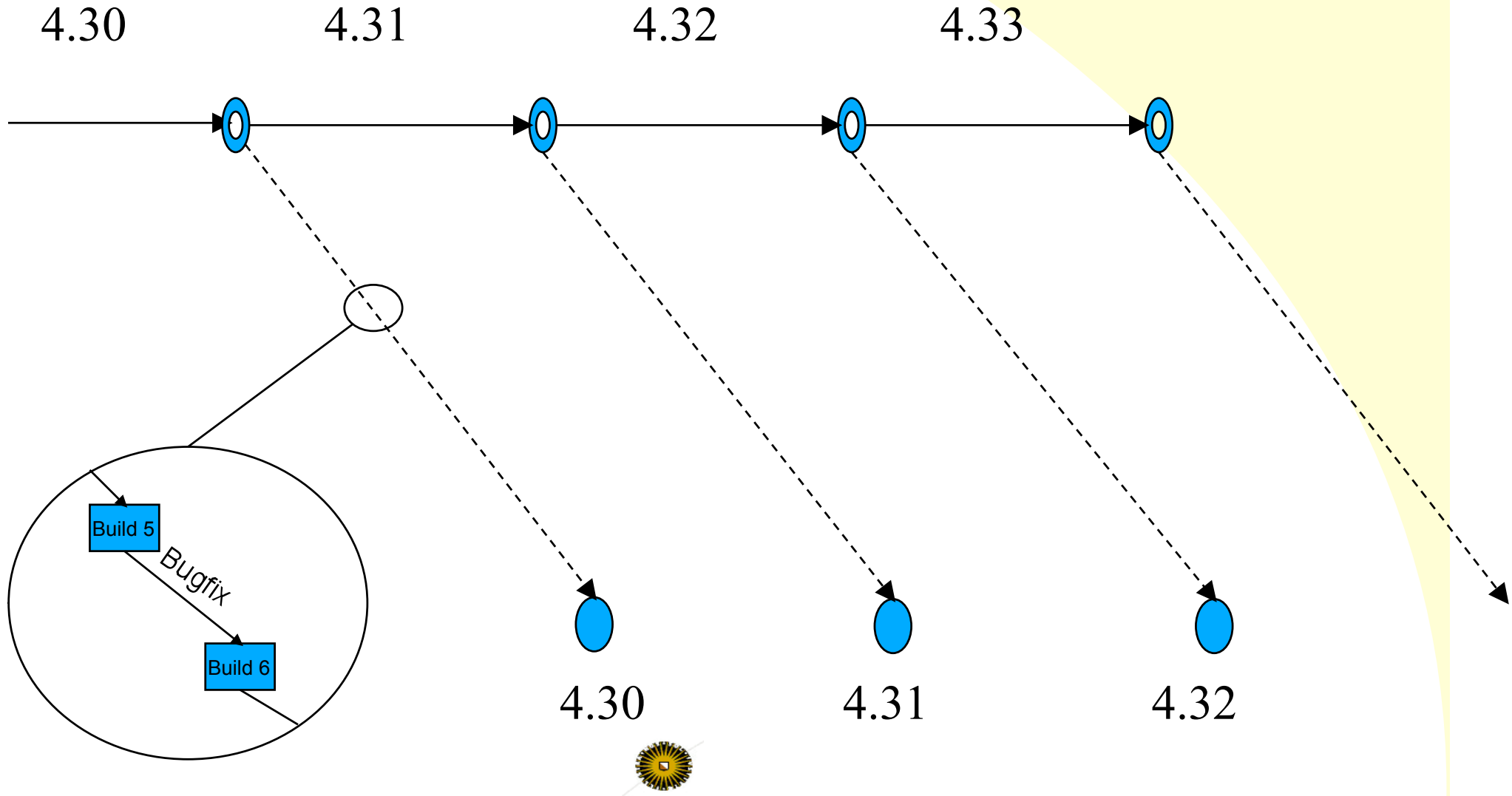
Cases

Conclusions

 Branch, release and freeze datamodel



 Stop supporting



## Process Step

## Tool used

## Responsible party



Contract is entered into PCIS	PCIS	Sales
License file and .ini file are generated for Release Wizard	PSL	Order bureau
Release Wizard packages files	Release Wizard	Order bureau
Burn license file and package onto CD	Release Wizard	Order bureau
Deploy CD to customer server	PAsetup	Consultant
Customer clients install application from server	Pclient	Customer
Customer clients use application and server	Planon.exe	Customer

## Lessons Learnt from Planon



- **KISS** – “Keep it Simple Stupid!”
- Being a good **toolbuilder** doesn’t necessarily improve the processes
  - Horror story of bad internal product management
- Well defined **release structure**
- (Transitioning to J2EE from another development platform cannot guarantee success)

“What if your developer walks under a tram?”



## Exact Software (1)



- **ERP Software** manufacturer
- Large range of products
- Subjects of case:
  - e-Portal product
  - Deployment tool
- ES has **160.000 customers**
  - Small to medium enterprise market
  - Aiming for larger customers
- 2025 employees
- Development takes place (mainly) in **three locations**:
  - United States
  - The Netherlands
  - Malaysia

“All of a sudden our build servers had no more time left to build... That’s when we realized we needed a new way of thinking about the build process, and we relocated our builders the next day.”



## Exact Software (2)



- Serves **large amount** of customers by
  - Explicit management of **CCU process**
  - CRM and CCU **integration**
  - Explicit license management
- Oversimplification has lead to **absence of SCM**
- **Proprietary tools** for CCU support



## e-Synergy

- Sold to customers
- Used **internally** as well
- Combines:
  - Documents (SCM)
  - Financials
  - Logistics (PDM)
  - Project
  - CRM
  - HRM
  - Workflow
- Used internally for **CRM/PDM/SCM**



Overview

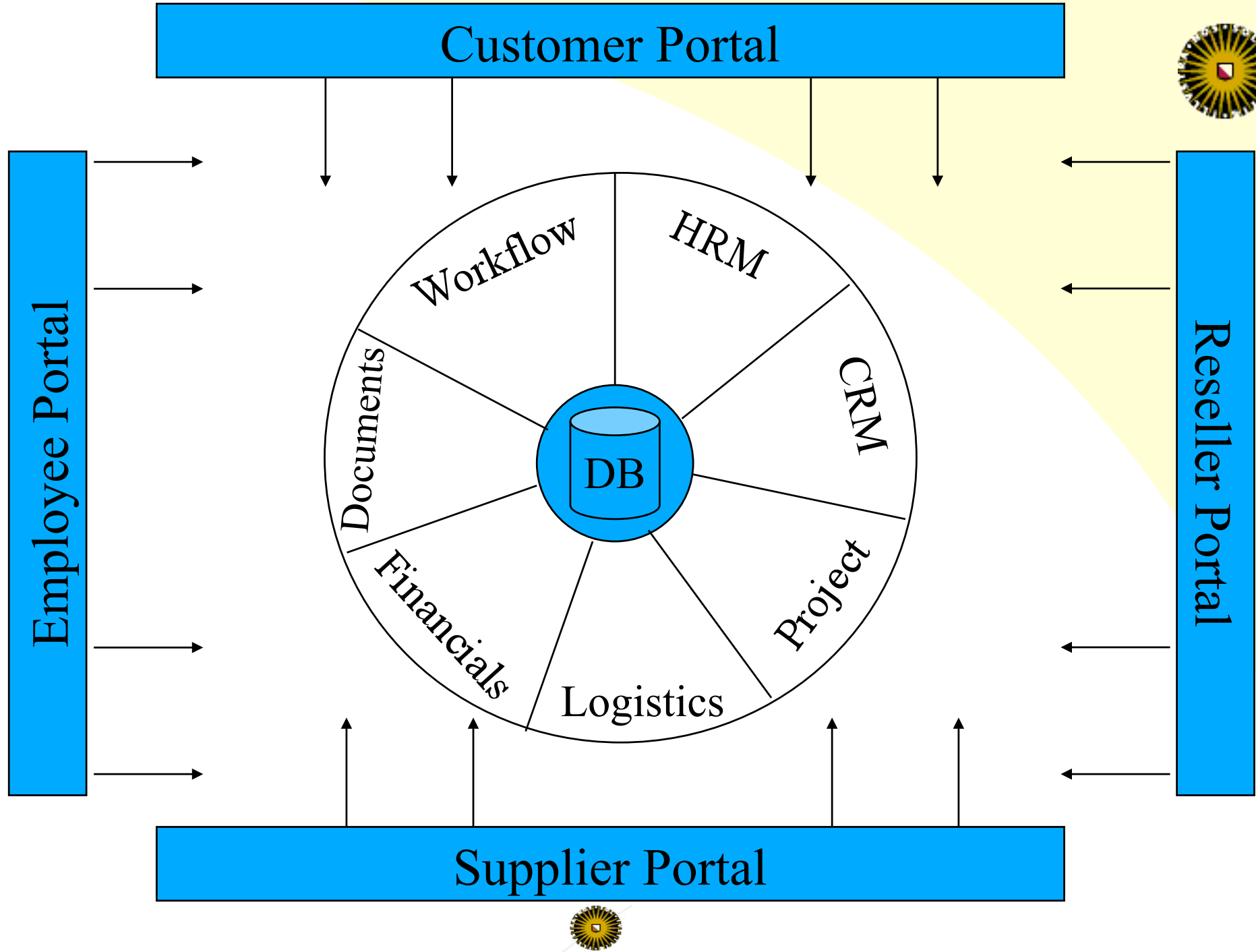
CC Updating

Issues

Solution

Cases

Conclusions



## Software Configuration Management



- **Version control** (...)
- Artefact management
- Maintenance process support
- Weekly releases
- **Promotion scheme**
- Eats own **dogfood**

Development

Repository D

Quality Assurance

Repository C

Internal customers

Repository B

Pilot customers

Repository A

Customers

Repository 0





## Product Data Management

- Product **composition**
- Workflow management
  - Projects are composed of tasks
  - Tasks are attached to items in the PDM
- Allow for different views
  - Sales view looks at **deliverables**
    - Executables
    - Manuals
    - Promotional material
  - Developer view looks at **product relationships and source items**



## Customer Relationship Management



- Architecture for **custom solutions**
- Customer can download and install updates
- **Automatic license renewal** on vendor side
- Automatic **post deployment user feedback** - Version number of new version is sent after update by the update tool
- Both vendor and customer side **license management** are explicit



Overview

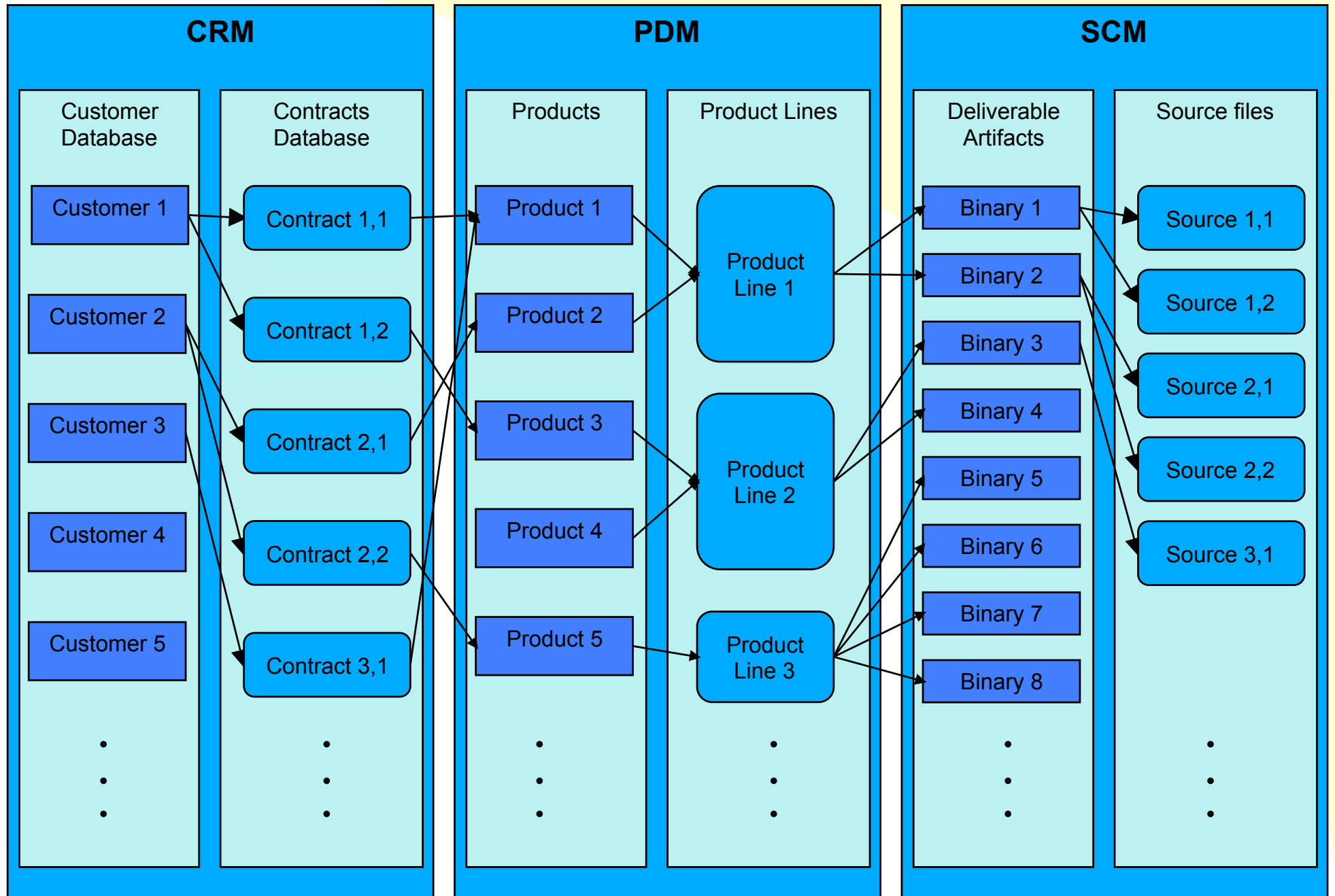
CC Updating

Issues

Solution

Cases

Conclusions



## Vendor side maintenance



- **Integrated system** thus supports
  - Version control
  - Software configuration management
  - Product composition
  - Manage deliverables
  - **Build problems** are solved (three timezones)
- Release and delivery is can be done **more effectively** allowing ES to serve **many (160k)** customers!



## Customer side maintenance



- On the customer side the integrated system thus supports
  - License **renewal**
  - **Customised** solutions
  - Updating process
  - Automatic **feedback**
- Customers **spend less time updating** and **encounter less problems**



## Lessons learnt



- Lessons
  - **KISS** works
  - Integrated support systems for maintenance improve process
  - Integrated maintenance of customer configurations and published releases improves **product quality**
  - Pilot customers improve product quality
  
- Possible due to
  - ES products are quite **stable**
  - ES customers are **tame**
  - ES products can be updated **through the web**

“People always think we just had a good idea, however, e-Synergy is the **ONLY** way in which we could still support 160k customers.”





## What will the future bring?

- More commercial and open source solutions for **software deployment** for products such as Wise Install, InstallShield, etc.
- More commercial and open source solutions for **network deployment** such as NetDeploy, Alt-Iris, etc.
- Continuous integration for testing and deployment
- More SKB related solutions such as release notes management tools and customer data tracking
- More customer feedback management
  - Release notes managers
  - Integrated call tracking systems
  - Integrated error tracking systems
- Different license models that focus especially on free products with large \$ maintenance contracts





[slinger@cs.uu.nl](mailto:slinger@cs.uu.nl)



So you want to be an Academic?

Cannot be fired



Full professor	Hoogleraar (Emeritus)	Sjaak Brinkkemper
Special professor	Bijzonder hoogleraar	Wim Scheper
Associate professor	Universitair hoofddocent	Ronald Batenburg
Assistant professor	Universitair docent	Slinger, Jurriaan
Postdoc	Postdoc	...
PhD student	Promovendus	Henk van der Schuur
Master student	Master student	You
Bachelor student	Bachelor student	...

