Enterprise Information Systems for Product Software Vendors: 
bringing the vendor and customer closer together

SIKS Dutch/Belgian Enterprise Information Systems Day

Slinger Jansen
Visting: University College London
Home institution: Utrecht University

Programme

- Introduction to Product Software and Customer Configuration Updating
- CCU Model description
- An enterprise information system for CCU
- Our case study research
- Practical examples
- Agile community
- Software supply networks
- Discussion and conclusions

Context of Customer Configuration Updating

- Main research question: "Can product software release, delivery, and deployment be improved by explicitly managing software and customer knowledge within a software supplier's organization?"
- 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].


What's the problem?

- No definition of the CCU process
- No method to evaluate a software supplier’s CCU process
- No tools that suffice all the needs of a software supplier

CCU Model

- Models the interaction between software vendors and its customers for each process area
- Based on:
  - Carzania
  - Safe
  - Update model
- Used to evaluate CCU processes of nine (9) case studies in the product software industry

**Required Functions for a PSW EIS**

- **Development**
  - Project management (development, release management)
  - Workflow management (development)
  - Software configuration management (development, release management)
  - Bug track system (support, development, maintenance)

- **Customer relationship management**
  - (marketing, support system integration, PRIs, licensing, etc)

- **Product data management** (release management, variability, etc)

- **Deployment support** (software delivery and deployment, media creation)

- **Delivery support** (internal/external publishing portals, release information portals, etc)

- **Data gathering** (pay-per-use, commonly used features, bug reporting, etc)

**Release Management**

- Release planning (tools, workflow, project, internal communication, etc)
- Release publishing (tools, connection to website)

- The fabricated product (enterprise integration, exception handling in the supply chain, etc)

---

**Case Study Protocol**

- **Exploratory case studies at product software companies (3-8 weeks)**
- **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

---

**A method for evaluating CCU for any vendor**

- **The CCU evaluation method is based on SPICE, the Software Process Improvement and Capability Determination (much akin to CMM-i)**
- Provides a method to evaluate the processes of release, delivery, deployment, and activation and usage separately, or integrated
- This is done through questions such as "Is there a release plan with regards to your software product that is available to all staff?"

- **Method developed through six case studies at Dutch software vendors**
  - **Company size from 60 to 1500 employees**
  - **Between 100 and 160,000 customers**
  - **Same method (case study protocol) for each case**

---

**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

Software Configuration Management

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

Development

Quality Assurance

Internal customers

Pilot customers

Customers

Product Data Management

- Product composition (even some variability support)
- Workload management
  - Tasks 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
Maintenance

- **Vendor side:**
  - Integrated system thus supports
    - Version control
    - Software configuration management
    - Product composition
    - Manager deliverables
    - Release and delivery is can be done more effectively allowing ES to serve many customers

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

*Please mind:* All this is possible only because Exact Software builds products for a limited number of technologies, with a very stable and rigid development process.

Planon and the agile community

- Integrating different systems
  - Sourceforge (used to be Visual SourceSafe)
  - Bugzilla (used to be Planon CRM)
  - Planon CRM
  - Planon Licenser (proprietary (!))
  - Planon update packages (proprietary (!))
  - "Don’t build an information system unless you need it!"
  - Excel sheets work fine!
  - Reporting tools change constantly

- However, some functions will not change
  - Software configuration management
  - Product data management
  - Customer relationship management
  - Project and workflow management

- The agile community requires enterprise information systems too!

Pheme

- Pheme is a tool for software knowledge distribution to transport knowledge between software vendors and end-users in a software supply network:
  - Software and updates
  - Feedback reports (usage, error, configuration, etc)
  - Commercial information ("An update of component X has been released"
  - License keys
  - etc.

- So what does it look like?
  - A memory resident Pheme service
  - A protocol through which Pheme instances communicate, and a protocol through which software products communicate with Pheme
  - Example: Joomla configuration, deployment, and update automation

Pheme serves well in a Software Supply Network

- Software vendors organize in federations
  - Focus lies on customer interaction and knowledge sharing
  - However, no tools dealing with software supply networks
  - Information portals required for both customers and resellers/partners/etc.

- Integration of different development management methods with project management and workflow systems

- **Product software knowledge infrastructure**
  - Product software management workbench
  - Componentization of information systems (who needs feature X?)

- Development of release repositories and software delivery portals

- Component-Off-The-Shelf evaluation and acquisition systems

- Integrating product and enterprise information system

Conclusions and points for discussion

- **There are no enterprise information systems that enable product software vendors to manage product software**
  - Development
  - Release
  - Delivery
  - Deployment
  - Usage and activation

- Exact Software is most advanced, but
  - Not generalizable
  - No support for agile methodologies
  - Not componentized
  - No release planning tools (and many other features missing, such as COTS evaluation)

- **Agile community needs enterprise information systems too!**

Software Supply Networks research contributes, due to more federation between product software vendors (e.g., CDTS)
Takehome: There are no enterprise information systems that sufficiently support software developers and product software vendors in their development, release, delivery, and deployment processes, even though there is an outspoken need for such systems.

Please take the time to read:


S. Jansen, S. 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, proceedings of the IEEE International Conference on Software Maintenance (ICSM2006, Scientific track), Philadelphia, PA, USA, September, 2006. Accepted for publication.