

The Role of Boundary Crossing for Knowledge Advancement in Product Development

Anders Mørch, Kathrine Nygård
InterMedia, University of Oslo, Norway
anders.morch@intermedia.uio.no, k.a.nygard@uv.uio.no,

Outline

- Identifying information society abilities for CSCCL
- Case: software product development
- Methodology
 - Design-based research in workplace settings
- Boundary crossing
 - Product development with user participation
 - Researchers' intervention at a workplace
- Summary and conclusions

Information society abilities

- Knowledge practices laboratory (KP-Lab) aims at
 - Studying emerging knowledge practices for the 21st century
 - Developing tools for transforming existing practices into knowledge practices
 - Understanding the role of boundary crossing for knowledge work
- Two practices we present in this talk
 - Joint artifact development
 - Multidisciplinary team work

Brief intro to case

- Company is a small software house in Norway with two main offices
 - Western office: Development and support
 - Eastern office: Sales and support
- Produces project management software for the oil, gas and (more recently) building construction industry
- Software development is characterized by frequent customer interaction and two different in-house cultures identified by the two main offices

Research questions

- *Adaptive product evolution*: Analyzing the trajectory of the company's project management software with focus on developer-user relations;
- *Mentoring*: How new employees are trained in a customer context through mentoring, and how this exemplifies novice-expert and horizontal boundary-crossing dimensions of knowledge advancement;
- *Knowledge advancement*: How shared knowledge is produced, accessed, used, shared, and reproduced in the company and across its multiple work contexts.

Methods

- Empirical
 - Observation in two offices and a customer site
 - Video-recordings of meetings
 - Interviews with company employees and customers
- Design
 - Design workshops and scenario building (participatory design)
 - Extreme programming techniques (software engineering)

Participatory Change Processes

- A blend of three existing methodologies:
 1. Design-Based Research (DBR)
 - Design experiments (Brown, 1992; Collins, 1992)
 2. Developmental Work Research (DWR)
 - Application of Activity theory into Change lab tools (Engeström, 2005)
 3. Participatory Design (PD) (Namioka & Schuler, 1993)
 - Mutual learning
 - Design techniques
 - Design workshop
 - Mockups and scenarios

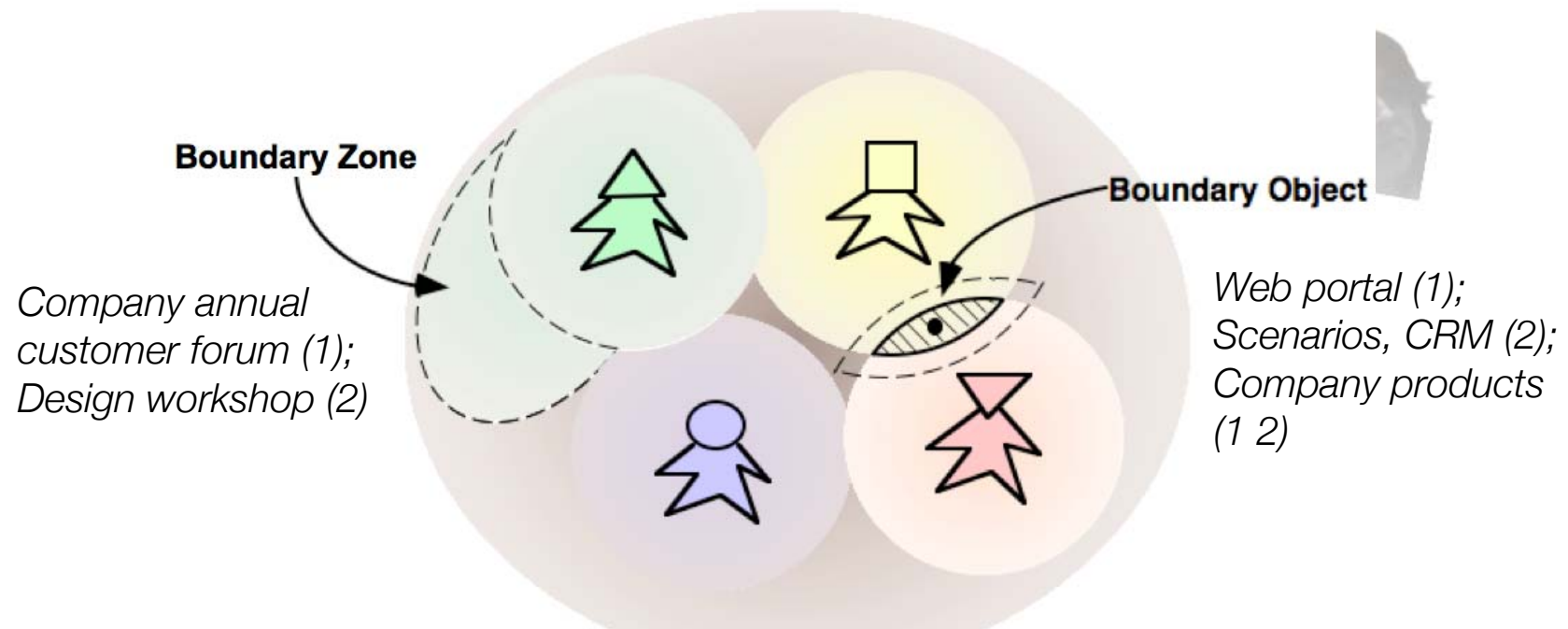
Why another methodology

- DBR for situations where the context is complex
 - Open-ended objects: 1) business needs, 2) research needs
 - Contributing to lasting changes of existing practice
- DWR for situations where the artifact is complex
 - Identify common artifact for joint development, many stakeholders
 - Development of new ICT tools (if needed)
- PD for situations that require usable and useful ICT artifacts
 - Researcher-developer interaction
 - Artifacts: web portal (to integrate with company CRM)
 - Developer-user (customers of company) interaction
 - Artifact: Company project planning tool

Boundary crossing

- Experts operate in and move between multiple activity systems , in particular during
 1. Adaptive product development
 - Developers develop new features and adapt existing products to new customers and niches in the market
 - Customers (domain expert users) propose improvement requests
 2. Researchers' intervention
 - Design-based research in the workplace (PCP)
 - Consultancy services to improve existing practices for managing knowledge in the company (CRM system)
- Concepts for studying boundary crossing

Boundary zones and adjoining objects



“Where heterogeneous expertise meets,” adapted from Fischer (2001)

BZ: Neutral ground in-between activity systems, allowing goals, outcomes and structures of different stakeholders to be reflected (Konkola, 2001; Edwards, 2007)

BO: Shared artifacts across activity systems that may also be part of an AS (Star, 1989)

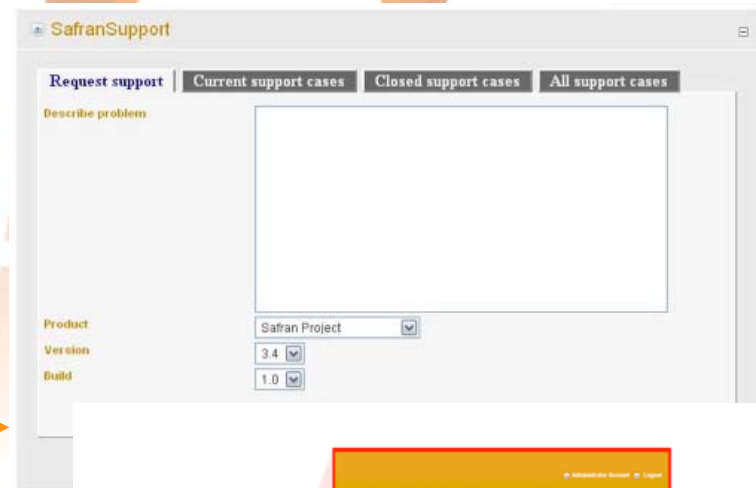
Interview excerpt

- *Ps3* They send in a request for a new functionality or improved functionality, and when we are developing, then we look (.) we assess it, does it make sense? Will it have a positive impact on other customers? Will it have negative effects for anyone? And if there are others requesting the same feature (...) we look into it on a larger scale? (...).
- *I2* Can you tell more about improvement-requests?
- *Ps3* When we go through improvement-requests we check the relevant areas of functionality and consider refactor areas where there are many requests. Then we have to (evaluate) whether they understand it ((the new functionality)), or whether we need to add more to it, and all in all we see this in relation to the rest of the product...

Design artifacts

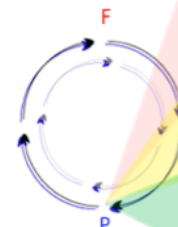
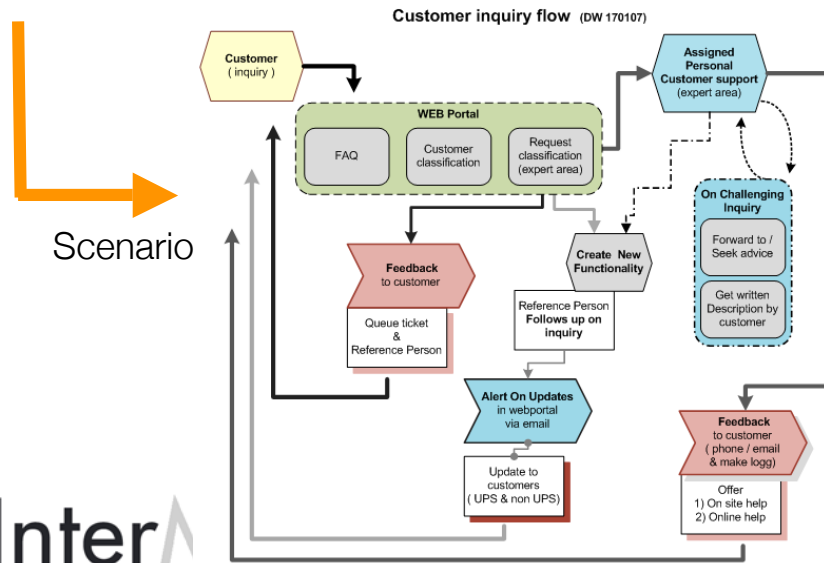


Design workshop



1st prototype

Iterations



Summary and conclusions

- We have studied a workplace setting and identified two key knowledge practices
 - Joint artifact development (collaborative design)
 - Multidisciplinary team work (complex communication)
- Using concepts from a sociocultural approach
 - activity system, boundary object, boundary crossing
- Using techniques from socio-technical systems design (integrating PD and SE)
 - design workshops, scenarios, prototyping, extreme programming, evolving artifacts