

CI Stakeholder Focus Workshops

This document summarizes some of the stakeholder focus workshops that we conducted in November 2012.

Goal of the focus workshops was to better understand how stakeholders see strengths and limitations of CIS within their context and to identify recurring concerns within their domain.

Each domain workshop had a duration of at least 3 hours. The participants included one researcher from the research team and 3 – 5 stakeholders with different backgrounds.

Procedure

The a focus workshop consisted of three phases:

1. Introduction: A researcher from our team presented a 20 minutes overview presentation of what CIS are, where they are currently used, and concluded with a live walkthrough in an example system.

2. Work Session: Then we started in an open workshop session where the following four areas of interest should be explored and discussed.

1. Current Limitations
 - a. What are current limitations / big challenges in your domain / company where you think CI could be useful?
2. Concerns
 - a. When you think now of CIS, what concerns would you have?
3. Brainstorming of Application Scenarios
 - a. Let's think about application scenarios in your domain / company where a CIS could be useful.
4. Risks
 - a. What risks would you associate with CIS?

The researcher from our team moderated the workshop and took notes. Depending on the engagement of the participants, some areas where explored deeper than others.

3. Debrief: Debrief of participants. Collection of feedback for improvement of the workshop. If interested, scheduled follow-up meeting.

Workshop Protocol #1

Participants have been anonymized. Certain sections and figures have been removed due confidentiality / non-disclosure agreements.

Date / Time:	2012-11-12 , 13:00 – 17:30
Location:	TU Wien, CDL Flex
Domain:	Software Development Tools
Participants (roles only)	<ol style="list-style-type: none">1. CEO (SW Tool Provider, Austria)2. Project Manager (SW Tool Provider, Austria)3. Business Development Consultant (Independent, Austria)4. Professor (Software Engineering, TU Wien)5. CI Team – Researcher

Notes

Current Limitations:

- Preserving of project-independent technical knowledge.
 - Being less dependent on individual “gurus” in project teams.
- Keep knowledge software components up-to-date (gap between documented and as-is status of components).

Concerns

- What can users do with the system?
 - Who is the user?
 - Software Developers
 - Quality Assurance Managers
 - Project Managers
 - Testers
 - CRUD operations on elements
 - Rating / Reviewing
- What do users get from the systems?
 - Recommendations
 - Notifications
 - Reports / digiest of ongoing activity.
 -
- How to get existing data into the system?
 - Companies have repositories of past projects:
 - Mining repositories to fill with existing material?
- Who is responsible for keeping the system running?
 - Idea “Librarian” role, who is in charge of administrating the system.

Brainstorming of Application Scenarios:

- “Facebook for Engineers”
- Sharing of code.
 - Users can share code elements with each other.
 - What attributes would such an element have?
 - Code
 - Description
 - Name
 - Author
 - Rating (5star Amazon-like)
 - Related elements.
 - ? dependencies
 - What analytics data would be important:
 - Can it also be used via web / tablet?
- Sharing of test cases.
- Sharing of configurations and specifications for model-drive software development.

Risks

- Developers could get distracted from their “real” work, by focusing more on the CI system, instead on getting the product done.
- Developers already use many tool (bugtracker, wiki, ...). There is only space of a limited set of tools that developers can use at once.
- Companies don’t want to use public platforms (Wikipedia-, github-like) for their internal development.
 - System needs to be deployable within organization
 - Virtual machine, private cloud
- Work laws: Any tracking of employees work behavior may be forbidden in certain organization due to agreements with unions.
 - What / how many organizations are affected by these laws?
 - Is this an Austrian-specific rule? (Germany as well?)

Workshop Protocol #2

Participants have been anonymized. Certain sections and figures have been removed due confidentiality / non-disclosure agreements.

Date / Time:	2012-11-18 , 10:00 – 14:00
Location:	TU Wien, CDL Flex
Domain:	Enterprise Knowledge Management
Participants (roles only)	1. CEO (SemanticWeb-/SocialWeb Service Provider , Austria) 2. CTO (SemanticWeb-/SocialWeb Service Provider , Austria) 3. Senior Post-Doc (Semantic Web, TU Wien) 4. CI Team – Researcher

Notes

Current Limitations:

- Systematic integration of knowledge from multiple domains.
- Certain knowledge can only be provided by users themselves.
- How to keep system active?
 - At the beginning some users are active, but then more and more users stop using the platform until practically no one uses it anymore.
 - critical success factor.
 - Already a problem when the system is enrolled within organizations:
 - System is accepted and adopted by employees
 - System is rejected by employees
 - -> also failure of management within org?
- Enterprise 2.0 software is complex to realize.
 - Replacing established communication (emails) with social networks is less effective than though
 - Hype is down: after failure stories, companies are more cautious with social software in their organization

Concerns

- What kind of data do users contribute to the system?
 - How to systematically explore what data is useful before building the system?
- How to keep systems active?
 - Incentive mechanisms
 - System generated recommendations
 - What kind of recommendations are useful?
 - See existing working in recommender system research.
- How to keep users engaged and motivated?

- Gamification
 - Achievements
 - Collecting points
 - Ranking list and leaderboards
- Look at how existing systems do it:
 - Facebook, Twitter, Pinterest, LinkedIn
- ? Micropayments (Amazon Mechanical Turk, Crowdfunder).

Brainstorming of Application Scenarios:

<<Removed upon request of participants, due to strong connection with their product.>>

Risks

- Privacy: If data is not handled properly, other employees/managers can use tracking data against the employees.
 - If happens: would immediately kill the platform, as employees would be afraid of using it.
- Realization Complexity and needless bureaucracy.
 - Project team is small (3-5 people)
 - Create a system design that can be realized and documented by small project teams.
 - Architecture documentation: currently done on-the-fly
 - Some AD elements remain outdated, as there are not enough resources to constantly keep them up-to-date.
 - How does a sufficient CI architecture look like?
 - Getting the system bigger is not interesting at the start.
-