«Knowledge Engineering applied to Software Engineering – past, present and future»

Trial lecture, November 30, 2010
Geir Kjetil Hanssen
Knowledge Engineering $\rightarrow$ Knowledge Management (KM)

- “A method that simplifies the process of sharing, distributing, creating, capturing and understanding of a company’s knowledge” ¹)
- “To make the enterprise act as intelligently as possible to secure its viability and overall success” ²)
- Involves a large number of multi-disciplinary, ambiguous, overlapping topics, disciplines, practices, theories etc.

Software Engineering

• “Software engineering is the discipline of specifying, designing, developing, deploying, and maintaining software systems.”¹
• "Software engineering is concerned with theories, methods and tools which are needed to develop software ... it is not constrained by materials governed by physical laws or by manufacturing processes” ²

• (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.
• (2) The study of approaches as in (1) ³

• SE is
  – complex, human knowledge work
  – often subject to failures, delays an cost overruns
  – a volatile industry serving a volatile market

• Even the product (software) is operationalized knowledge (domain, business processes, organization, technology, HCI, psychology, collaboration, communication, design, data storage, security etc.)

² Sommerville, I., Software Engineering. 8 ed. 2006: Addison Wesley.
Knowledge

- “Justified true belief” (Plato)
- “A dynamic human process of justifying personal belief towards the truth” (I. Nonaka)
- Explicit – Tacit
- Levels of refinement
  - Data
  - Information
  - Knowledge
- Knowledge evolution cycle
  - Originate/create knowledge (from practice such as learning, problem solving, innovation, creativity, and importation from outside sources)
  - Capture/acquire knowledge (in explicit forms)
  - Transform/organize knowledge (as written material, knowledge bases etc.)
  - Deploy/access knowledge (through education, training programs, automated knowledge-based systems, or expert networks)
  - Apply knowledge (in practice)

---

Organizational learning

- Experiential learning

- Communities of practice
  - "groups of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly."

- Single-loop and double-loop learning
  - concrete experience (feeling)
  - reflection & observation (watching)
  - abstraction & generalization (thinking)
  - test hypothesis in new situation (doing)

  - concrete + active
  - abstract + active
  - concrete + reflective
  - abstract + reflective

  - governing values —> actions —> consequence
  - expected error
KM timeline

- 10000bc: spoken language – explicit knowledge
- 2800bc: papyrus/paper – stored knowledge
- 1450: printing – distributed knowledge
- 1605: the Jacquard loom – automated knowledge
- 1801: "Advancement of learning" by Francis Bacon
- 50's: computers
- 60's: knowledge workers
- 70's: ARPANET
- 80's: computers everywhere
- 90's: "The knowledge creating company" by Takeuchi & Nonaka
- 00's: web-based tools

- 10000bc: spoken language
- 2800bc: papyrus/paper
- 1450: printing
- 1605: Jacquard loom
- 1801: "Advancement of learning" by Francis Bacon
- 50's: computers
- 60's: knowledge workers
- 70's: ARPANET
- 80's: computers everywhere
- 90's: "The knowledge creating company" by Takeuchi & Nonaka
- 00's: web-based tools
KM strategies

• Push – Pull ¹)
  – Push: using tools to identify and provide knowledge to potential users.
  – Pull: users themselves have to use repositories and other tools to identify relevant knowledge.

• Codification – Personalization ²)
  – Codification: codified knowledge is the main factor for knowledge exchange.
  – Personalization: personal interaction is the main factor for knowledge exchange.


Why KM in SE?

- The main asset in SE is knowledge
  - Knowledge on domain, business processes, organization, technology, HCI, psychology, collaboration, communication, design, data storage, security etc.
  - Personnel turnover represents a constant leak of knowledge, where tacit knowledge is hard to replace
  - The challenge is to keep and make use of the organizations knowledge
- KM challenges are critical to software engineering 1)
  - SE is knowledge work
  - Continuing development with maintenance is a KM challenge
  - Software and system architecture can be extremely complex
  - Rapid turnover of staff is a challenge (knowledge leak)
  - SE knowledge is an unusually complex combination of expertise
  - The organizations and the team cultures must be aligned with KM actions
- “…knowledge and its interorganizational management, as well as individual and organizational capability building, are becoming crucial factors for gaining and sustaining competitive advantages.” 2)
- “If organizations can manage the learning process better, then they can become more efficient.” 3)
- SE issues that can benefit from KE 4)
  - Planning, monitoring, and quality control based on project experience
  - Quality and process improvement of learning software organizations
  - Decision making
  - Automation

Knowledge management model ¹,²)

**Strategy**

*Goals and a way to achieve them*

**Processes**

*Methods to manage tacit and explicit knowledge*

**Tools**

*Infrastructure for explicit knowledge*

- Produce SW faster, with less cost and with higher quality
- Improve work situation
- Postmortem reviews
- Experience reports
- Communities of practice
- Intranet, wikis etc.
- Experience factory
- Data warehouse/mining
- Case-based reasoning

KM in SE examples
Postmortem analysis \(^1,\)\(^2\)

**Typical agenda for a postmortem analysis**

- **Introduction**
  Clarify the agenda and the expectations
  Who are we, what are we going to do and why?

- **Brainstorming/KJ**
  What went well in the project?
  Presentation of experience using stickers
  Structuring and prioritization of experiences

- **Root cause analysis**
  Why were these aspects a success?
  Organizing and prioritizing of causes using fishbone diagrams

- **Brainstorming/KJ**
  What did not go so well in the project?
  Presentation of experience using stickers
  Structuring and prioritization of experiences

- **Root cause analysis**
  Why were these aspects not a success?
  Organizing and prioritizing of causes using fishbone diagrams

- **Summing up**
  Summing up of the seminar and further plans
  Priority of actions

---


\(^2\) Dingsøyr, T. and F.O. Bjørnson, eds. *Experience transfer and knowledge management in SMEs developing and delivering software products.* 2010, EXTRA Consortium.

---

[NTNU – Trondheim]
Norwegian University of Science and Technology

www.ntnu.edu
KM in SE research

- Findings from a systematic review of the research literature on KM in SE 1)
- (1) What are the major knowledge management concepts that have been investigated in software engineering?
  - The majority of studies of knowledge management in software engineering relate to technocratic and behavioral aspects of knowledge management.
  - The studies that report on concepts within the fields of technocratic and behavioral aspects have very little overlap.
  - There are few studies relating to economic, spatial and cartographic approaches to knowledge management.
- (2) What are the major findings on knowledge management in software engineering?
  - The major finding, which is repeated over several papers and across several schools is the need to not focus exclusively on explicit knowledge but also on tacit knowledge.

(a) The effect of knowledge integration on Software development performance 2)
KM critique

• “Information graveyards” ¹,²)
• “If a group of people don't already share knowledge, don't already have plenty of contact, don't already understand what insights and information will be useful to each other, information technology is not likely to create it.” ²) p.104
• Overfocus on technology and tools ³)
  – Codifiability of knowledge is overstated
  – Overemphasizing the utility of new information systems
  – Unjustified assumptions about the willingness of employees to use such IT systems
  – The codification of tacit knowledge into formal systems may generate its own pathology

### Some cases of KM in SE 1)

<table>
<thead>
<tr>
<th>Study</th>
<th>Focus</th>
<th>Lessons learned</th>
</tr>
</thead>
</table>
| Liebowitz, *A Look at NASA Goddard Space Flight Center’s Knowledge Management Initiatives* | • Many initiatives, e.g. NASA Lessons Learned Information System (explicit) and knowledge sharing forums (tacit/CoP).  
• KM Working group and KM Officer | • Knowledge repositories needs to be “push-enabled”  
• You need an integrated KM architecture to cut across the functional silos.  
• KM processes must be embedded within the daily activities of the employee for KM to be successful. |
| Ramasubramanian and Jagadeesan, *Knowledge Management at Infosys* | • Dedicated KM infrastructure function (8 full-time  
• Knowledge repositories (K-shop and asset DB) + People Knowledge Map) | • Monetary incentives ensures repository content and quality  
• Reduced the defect-level by 40%  
• Increased productivity by 3% |
| Schneider et al., *Experience in Implementing a Learning Software Organization* | • DaimlerChrysler’s implementation of experience factory  
• A separate organizational entity supporting the complete software lifecycle | • Learning implies broad and deep change of culture and work-style at all levels  
• Capitalizing on learning involves the risk of obsolete knowledge  
• Need both knowledge and relevant context information |
| Ramesh, *Process Knowledge Management with Traceability* | • Studied best practices in traceability in 30 system development organizations.  
• Traceability: the ability to follow the life of an object developed during software engineering from its creation to its use. | • Traceability positively affects construction, storage, transfer and use of knowledge. |
| Wei et al., *Design and Evaluation of a Knowledge Management System* | • Developed and evaluated a system that supports organization-wide knowledge creation, update, sharing, and reuse.  
• HW-manufacturer in Taiwan. | • The turnaround time for knowledge improved over the previous manual process.  
• Enhanced productivity (reduced production interrupts).  
• Reduced the frequency of delayed replies. |

---

1) All studies are found in the May/June 2002 issue of IEEE Software (Special issue on Knowledge Management in Software Engineering).
# KM generations – past, present and future

<table>
<thead>
<tr>
<th></th>
<th>1st generation (past)</th>
<th>2nd generation (present)</th>
<th>3rd generation (future)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newell, 2005 1)</td>
<td>based on the view of knowledge as possession</td>
<td>based on the view of knowledge as based in practice</td>
<td></td>
</tr>
<tr>
<td>Firestone &amp; McElroy, 2003 2)</td>
<td>focus on knowledge sharing</td>
<td>based on knowledge creation</td>
<td></td>
</tr>
<tr>
<td>Vorakulpipat &amp; Rezgui, 2008 3)</td>
<td>focused on information sharing, information repositories, and intellectual capital accounting</td>
<td>focused on the concepts of tacit knowledge, social learning, situated and embedded knowledge, and communities of practice</td>
<td>“For the future of KM, it is necessary to explore and emphasize the impact of KM on people, organizations, and society in terms of value creation”</td>
</tr>
<tr>
<td>Tuomi, 2002 4)</td>
<td>Computer supported storing and sharing of knowledge</td>
<td>KM specialists and computer supported human sensemaking</td>
<td>“..understand better the cultural basis of knowing and social activity..”, “..the ways in which social and organizational learning lead to new social practices.”</td>
</tr>
</tbody>
</table>

**Summary**

- collect, structure and share explicit knowledge
- practice based learning, knowledge creation
- KM as an integrated part of organizations, society and processes

---

Thinking ahead

• More emphasis on work processes oriented towards knowledge development and sharing than explicit systems for knowledge management
  • feedback-driven development
  • self-managed teams (in direct contact with externals)
  • collaboration with externals (customers, suppliers, third-parties)
  • building skills/craftsmanship (dojos’s, kata’s)
  • keep the good people (e.g. Google)

• Openness and transparency
  • knowledge transparency, internally and externally
  • open processes/networked organizations/ecosystems
  • open knowledge (opensource, wikipedia/google)

• Technology-support
  • advanced indexing/search
  • relieve knowledge workers, e.g. code complexity analysis
  • supporting communication and collaboration
  • automation of work, more emphasis on knowledge creation and exchange
  • manage the complexity of software, increase maintainability