707.009
Foundations of Knowledge Management
„Organizational Knowledge Repositories“

Markus Strohmaier

Univ. Ass. / Assistant Professor
Knowledge Management Institute
Graz University of Technology, Austria

e-mail: markus.strohmaier@tugraz.at
web: http://www.kmi.tugraz.at/staff/markus
Overview

Agenda

• Experience Factories
• Case Based Reasoning

In the context of Software Engineering
Overview

Previously:
• Knowledge Organization
• Broad Knowledge Bases
• Knowledge Acquisition

Last week:
• Knowledge Transfer

Today:
• Organizational Knowledge Repositories
### Schools of KM
[Earl 2001]

Table 1. Schools of Knowledge Management

<table>
<thead>
<tr>
<th>SCHOOL</th>
<th>TECHNOCRATIC</th>
<th>ECONOMIC</th>
<th>BEHAVIORAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE</td>
<td>SYSTEMS</td>
<td>CARTOGRAPHIC</td>
<td>ENGINEERING</td>
</tr>
<tr>
<td>FOCUS</td>
<td>Maps</td>
<td>Processes</td>
<td>Income</td>
</tr>
<tr>
<td>AIM</td>
<td>Knowledge Directories</td>
<td>Knowledge Flows</td>
<td>Knowledge Assets</td>
</tr>
<tr>
<td>UNIT</td>
<td>Enterprise</td>
<td>Activity</td>
<td>Know-how</td>
</tr>
<tr>
<td>EXAMPLE</td>
<td>Bain &amp; Co AT&amp;T</td>
<td>HP Frito-Lay</td>
<td>Dow Chemical IBM</td>
</tr>
<tr>
<td>CRITICAL SUCCESS FACTORS</td>
<td>Culture/Incentives to share Knowledge Networks to Connect People</td>
<td>Knowledge Learning and Information Unrestricted Distribution</td>
<td>Specialist Teams Institutionalized Process</td>
</tr>
<tr>
<td>PRINCIPAL IT CONTRIBUTION</td>
<td>Profiles and Directories on Internets</td>
<td>Shared Databases</td>
<td>Intellectual Asset Register and Processing System</td>
</tr>
<tr>
<td>&quot;PHILOSOPHY&quot;</td>
<td>Connectivity</td>
<td>Capability</td>
<td>Commercialization</td>
</tr>
</tbody>
</table>
KM as risk prevention in Software Engineering

Addressing risks such as

- Loss of knowledge due to attrition
- Lack of knowledge and an overly long time to acquire it due to steep learning curves
- People repeating mistakes and performing rework
- Individuals who own key knowledge become unavailable

What kind of risks can you identify from a KM perspective?
Types of Knowledge Reuse Situations
[Markus 2001]

Four distinct types:

- **Shared work producers**
  - who produce knowledge they later reuse

- **Shared work practitioners**
  - who reuse each other’s knowledge contributions

- **Expertise-seeking novices**
  - who seek advise from experts

- **Secondary knowledge miners**
  - who seek to answer new questions or develop new knowledge
Knowledge Transfer in Software Engineering

Knowledge about …

Shared Work Producers

Knowledge Transfer

Shared Work Practitioners

Knowledge about …

Software Developer(s)

Knowledge Repositories

Software Developer(s)

Knowledge Producer Knowledge Intermediary Knowledge Consumer

Markus Strohmaier 2007
Knowledge Transfer Participants

M.L. Markus, Toward a Theory of Knowledge Reuse: Types of Knowledge Reuse Situations and Factors in Reuse Success

Shared Work Producers:
• „Prosumers“
• Producers of knowledge for their own later reuse

Shared Work Practitioners:
• Producers of knowledge for each others use
What kind of knowledge is relevant in a Software Engineering context?

Knowledge about:

- Software architecture
- Functional and Non-Functional Requirements
- Design rationale
- Design Trade-offs
- Stakeholders and Stakeholder expectations
- Lessons learned
- …
Knowledge Repositories

M.L. Markus, Toward a Theory of Knowledge Reuse: Types of Knowledge Reuse Situations and Factors in Reuse Success

Types of Knowledge Repositories:

– Repositories of documents
– Repositories of data
  ➔ Fundamental differences for retrieval

– Repositories that store external knowledge (e.g. Customers, Competition)
– Repositories that store internal knowledge (e.g. Meeting transcripts, mails)

– Repositories containing general knowledge (e.g. scientific knowledge)
– Repositories containing specific knowledge (e.g. context sensitive)

– Repositories containing declarative / procedural / rationale / analytic knowledge
Knowledge Repositories in Software Engineering

Related Concepts:
- Lessons Learned database
- Lessons to Learn database
- Project close-out / post-mortem repositories
- Experience factory
- Knowledge base
- Etc.
Knowledge Repositories in Software Engineering


Goals of Knowledge Repositories:

- Reduce software defects
- Sharing knowledge about local policies and practices
  - E.g. through Document management
- Capturing knowledge and knowing who knows what
  - E.g. through competence management and expert identification
- Software process improvement
  - Decrease time and cost and increase quality
  - E.g. through software reuse
- Making better decisions
- Accessing domain knowledge
- Supporting learning and feedback

What are different purposes knowledge repositories can serve?
Skills Management

Interviews mit Vorgesetzten zu Stellenanforderungen der Mitarbeiter

Erstellung eines SOLL Profils

Abstimmung der Profile mit Stelleninhabern
Einschätzung des IST Profils als Selbst- (x) und Fremdeinschätzung (x)

Abstimmung der unterschiedlichen Sichtweisen

<table>
<thead>
<tr>
<th>Fachliche Fähigkeiten</th>
<th>Beschreibung</th>
<th>Know-how für einfache Fragestellungen</th>
<th>Know-how für komplexe Probleme</th>
<th>Diagnosekompetenz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kenntnis der Basisanwendungen</td>
<td>Anwendungskenntnis aller Basisysteme und Basisprodukte</td>
<td>x</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>Kenntnis technischer Infrastruktur</td>
<td>Kenntnis technischer Infrastruktursysteme, wie Netzwerk, Router etc.</td>
<td>x</td>
<td>x</td>
<td>0</td>
</tr>
<tr>
<td>Systemknowhow</td>
<td>Entwicklervorkenntnisse zu Systemen und Produkten</td>
<td>0</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>Management, Methoden</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Innovationsfähigkeit, Lernfähigkeit, Reflexionsfähigkeit</td>
<td>Methodenentwicklung, Kontakte zu Kompetenzträgern ist ständig auf der Suche nach Verbesserungen; informiert sich regelmäßig über neue Entwicklungen im Fachgebiet und bringt diese Informationen am Arbeitsplatz ein</td>
<td>x</td>
<td>x</td>
<td>0</td>
</tr>
<tr>
<td>Linienorganisation</td>
<td>Verwalten von Budgets, Stunden, Zusammensetzung des Stundensatzes, BAB, Termine Ressourcenplanung</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Problemlösungsfähigkeit</td>
<td>Krisenmanagement, Prioritäten setzen, Analyse von Problemen, vernetztes Denken</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Projektdurchführung</td>
<td>Systemleitertensteuerung</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Personliche/Soziale Kompetenzen</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gesprächstechnik</td>
<td>Kann Kunden am Telefon gut führen und sachlich Problemlösung herausarbeiten</td>
<td>x</td>
<td>x</td>
<td>0</td>
</tr>
<tr>
<td>Mitarbeiterführung</td>
<td>Personalaufbau und -weiterbildung; findet Personal für den diagnostizierten Bedarf in seiner Abteilung; systematisiert die Aufgaben in seiner Abteilung und ordnet Mitarbeiter effektiv</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Markus Strohmaier 2007
Barriers to Knowledge Transfer

- **Issues with the Knowledge Repositories [7]:**
  - Lack of awareness,
  - Lack of trust,
  - Lack of time,
  - low information quality,
  - low usage,
  - Preparing entries is time consuming,
  - expensive maintenance,
  - context dependency
Three Potential Solutions
[Cabrera2002]

A look back to our last class:

1. Restructuring the payoff function
2. Increasing perceived efficacy of individual contributions
3. Establishing group identity and promoting personal responsibility

But organizational knowledge repositories are not necessarily discretionary databases
The Experience Factory

V. R. Basili and G. Caldiera and D.H. Rombach, Experience Factory
The Experience Factory


Improving the software process and product requires the continual accumulation of evaluated experiences (learning)
– in a form that can be effectively understood and modified (experience models)
– into a repository of integrated experience models (experience base) that can be accessed and modified
– to meet the needs of the current project (reuse).

The paradigm implies the logical separation of project development (performed by the Project Organization)
– from the systematic learning and packaging of reusable experiences (performed by the Experience Factory).
The Experience Factory
V. R. Basili and G. Caldiera and D.H. Rombach, Experience Factory

– The Experience Factory is a logical and/or physical organization that supports project developments by analyzing and synthesizing all kinds of experience, acting as a repository for such experience, and supplying that experience to various projects on demand (Figure 2).

– It packages experience by building informal, formal or schematized, and productized models and measures of various software processes, products, and other forms of knowledge via people, documents, and automated support.
The Experience Factory

V. R. Basili and G. Caldiera and D.H. Rombach, Experience Factory

- **Experience Factories (EF)** focus on the facilitation of Knowledge Transfer between Software Developers
- Experience Base
  - “Packages Experiences”
- Goals
  - Knowledge Transfer
  - Knowledge Reuse
Experience Factories (EF) focus on the facilitation of Knowledge Transfer between Software Developers

- **Experience Base**
  - “Packages Experiences”

- **Goals**
  - Knowledge Transfer
  - Knowledge Reuse

Examples: Know-Center Processes, Post-Mortem Process and LL Documents
Knowledge Flow Theory
[Nissen 2004]

Classification of different types of knowledge flows along 3 dimensions

- Explicitness
  - Tacit / Explicit

- Reach
  - Individual, Group, Organization, Interorganization

- Life Cycle
  - Evolve, Apply, Distribute, Formalize, Organize, Create, ...

Formalization:
Let \( a = a_1e + a_2r + a_3l \)

within the coordinate system \( e = \) expliciteness, \( r = \) reach and \( l = \) lifecycle

then the goal of e.g. an experience factory can be expressed as the vector \( AB \) with \( A = (\text{implicit, group, share}) \) and \( B = (\text{explicit, group, share}) \).
Case Based Reasoning [Aamodt 1994]

– A problem solving paradigm
– Utilizing the specific knowledge of previous experiences

– A new problem is solved by
  • Finding a similar past case and
  • Reusing it in the new problem situation

– An incremental approach to learning –
– experience is retained each time a problem has been solved
Case Based Reasoning Applications

Example: Salesforce – Customer support by call centre agents


What type of knowledge repository is this system?
Case Based Reasoning [Aamodt 1994]

In CBR terminology

- A **case** denotes a problem situation
- A **past case** denotes a previously experienced situation, which has been captured and learned in such a way that it can be reused in solving future problems (also previous case, stored case, retained case)
- A **new case** (or unsolved case) is a description of a new problem to be solved

Case-based reasoning is a cyclic and integrated process of solving a problem, learning from this experience, solving a new problem, etc
Case Based Reasoning [Aamodt 1994]

Learning in CBR occurs as a natural by-product of problem solving
When a problem is solved, the experience is retained in order to solve similar problems in the future
When an attempt to solve a problem fails, the reason for the failure is identified and remembered in order to avoid the same mistake in the future

CBR favours learning from experience, i.e. Learning from concrete problems
Case Based Reasoning [Aamodt 1994]

Central tasks are

1. Identify the current problem situation
2. Find a past case similar to the new one
3. Use that case to suggest a solution
4. Evaluate the proposed solution
5. Update the system by learning from this experience
Case Based Reasoning [Aamodt 1994]

A descriptive Framework for CBR systems

The CBR Cycle

1. RETRIEVE most similar case
2. REUSE the knowledge in that case
3. REVISE the proposed solution
4. RETAIN relevant parts of this experience

Fig. 1. The CBR Cycle
Case Based Reasoning [Aamodt 1994]

Example: Customer Support System
Case Based Reasoning [Aamodt 1994]

Core problems addressed by CBR researchers:

1. Knowledge representation (How to represent cases?)
2. Retrieval methods (How to identify relevant cases?)
3. Reuse methods (How to translate knowledge from old to new case?)
4. Revise methods (How to improve cases?)
5. Retain methods (How to increase and improve the case base?)

A very broad field of current research:
See, for example, http://www.iccbr.org/iccbr07/
Any questions?

Coming up: A guest lecture by T. Ley

Tomorrow!

<table>
<thead>
<tr>
<th>10</th>
<th>2.12.2009</th>
<th>Cognitive Psychology Theories for Knowledge Management (slides)</th>
</tr>
</thead>
</table>

In this class, we will discuss some fundamental psychological concepts in the context of knowledge management, including for example Knowledge Space Theory.

Guest Lecture: Tobias Ley, Know-Center and Karl-Franzens-University Graz