The brain's persepective on requirements engineering

Extended Abstract for the MaRK'11 Workshop

Markus Flückiger
Zühlke Engineering AG
Switzerland
mdf@zuehlke.com

Abstract—**Software development is knowledge work. The work of cognitive psychologist offers many insights in respect to knowledge work and these insights suggest an approach to requirements engineering where collaboration and communication plays a much more important role than the documentation of requirements. Such insights must again impact, how requirements knowledge is managed in an organization. This extended abstract promotes the idea to evaluate today's approaches from a brain's perspective.**

Knowledge Work, Managing Requirements Knowledge, Problem Solving, Cognition

I. Introduction

According to [1] knowledge work is work that, at least for the person involved, is so complex or new that the person's experience is not sufficient to complete it. The person must acquire, integrate and even develop knowledge to do so. The missing knowledge is a barrier. Software development is knowledge work. Given this, a few questions can arise:

- What is knowledge?
- How do humans make use of knowledge?
- What knowledge is relevant for software development?
- What does that mean for the requirements discipline?

II. The work of cognitive psychologists

Cognitive psychologists have gained many insights relevant for knowledge work. These researches studied for example the process of problem solving, i.e. how humans do overcome the barrier that stems from the missing knowledge. In [2], the basic problem solving process is presented as a closed feedback loop that is driven by the task's goal: As the barrier prevents us humans to immediately find a solution for the overall goal, we look for a sub-goal that promises a step into the right direction. We solve this sub-problem based on the knowledge stored in our long-term memory, and rate the found solution. In this process, we gain insight about the problem as well as the possible solutions. This insight is integrated in our long-term memory. The improved knowledge allows us now to form a different sub-goal we can pursue in the same manner, until the task's overall goal is met. In this process of solving and learning, we construct a representation of the problem.

Our working memory holds the information to reason about. However, even for a simple problem, our working memory does not have enough capacity to hold all the details needed. Our brain uses several strategies to deal with this constraint: (1) we build a knowledge structure with a higher complexity that allows us to reason based on highly condensed information; (2) our intuition automatically uses existing knowledge to deal with the many details; (3) we externalize some aspects of the problem and the solution and use the highly developed perception to access it. Here are a few observations worth mentioning:

- Problem solving is a highly iterative process where we gain insight about the problem and the solution from pursuing sub-goals.
- We solve a problem by developing problem specific knowledge structures in our brain compact enough for our working memory to handle the problem.
- We understand the problem while trying to solve it.
- Our intuition is a highly efficient problem solving mechanism that errs if based on inappropriate knowledge.

As a last observation in this section, knowledge can be described as information stored in the human brain.

III. Software development

Software development creates a piece of software, quite often a software product. The goal is to create a "good" product. Such a product offers the right functionality with the needed qualities at the appropriate time to the market. A new software product induces change into a complex and dynamic socio-technical system. It is one driver to bring such a system from today into tomorrow. This dynamic behavior alters the definition of "good". Thus "good" can - at best - be assessed when the change happens, i.e. the software is in use. The knowledge needed to understand this socio-technical system stems from the following four basic knowledge areas: (a) market, e.g. where a company can make money; (b) experience, e.g. how the product will influence and change the user's processes and experience; (c) subject area, e.g. specific business rules; and (d) technology, e.g. what technologies to choose and how to architect the software. A "good" software results from considering the relevant information from all four knowledge areas and find a solution that fits the constraints of all of them. Finding the definition of "good", which is a key task of the requirements engineering discipline (RE), is all but simple.

IV. Knowledge in Teams

Software products are usually developed by teams. The promise is twofold: (i) Software development is fast enough as work can be done in parallel; (ii) the created software is "good enough", as individuals can combine their individual knowledge and skills. As for individuals, the barrier stems from missing knowledge. However, it is not that much the individual team member's knowledge that decide upon the chosen solution, but the team's knowledge. In [2], there are two team cognitive structures that should be mentioned here. The first is the team mental model. This is knowledge the team members have in common. Such knowledge enables the members to form a common understanding and goal of the problem. It is needed e.g. so that all members can head into the same direction. Such knowledge may stem from education and other sources. The more important part, i.e. the common representation of the problem at hand, mostly results from communication and collaboration between team members during the project. The second cognitive structure is the transactive memory. It is the knowledge of who knows what. This knowledge enables the team e.g. to split up the work effectively. It again mostly results from communication between and collaboration of the team members. Here are a few observations:

- The representation needed to overcome the barrier is distributed over the team members.
- The quality of the team knowledge, i.e. common mental model and transactive memory, is decisive for the quality of the solution.
- The team knowledge is mainly established through communication and collaboration.

V. Applied to requirements engineering

If the above perspective is applied to RE, some common practices are questioned. Here are a few lines of thought:
When documenting e.g. a requirement, the writer externalizes knowledge. For everybody but the writer, this is just information. Readers integrate that information into their individual knowledge structure. In this process communication failures will occur. If the purpose for writing requirements is to ensure that someone else creates the thing the writer had in mind then communication failures are not an option. In non-trivial cases, an acceptable level of failures needs an iterative communication process. This process is best called teaching. There are many possible models of teaching one can adopt. Some positions hand the responsibility for learning from the teacher to the student; the teacher becomes the student's sparring partner and coach. Quite obviously, such a position has nothing in common with document driven approaches, where a requirements engineer writes a document for developers, testers and others to consume.

According to the general requirements engineering literature, good requirements are essential. Many data points, e.g. a specific behavior of one group of users, a detailed demand from a market segment, a security consideration and more have been processed, consolidated and abstracted into requirements, e.g. a completely dressed use case. This essence now stands for a huge body of knowledge from all the four knowledge areas. Consumers of requirements will integrate the information into their knowledge structure containing their individual data points. Many of the latter will not match the ones used to derive the requirements. Not only does this cause friction, as some requirements must look illogical from a different perspective. Even more so will consumers create another than the intended solution as their intuitive problem solving is based on different knowledge.

In quite many projects, requirements engineers are men or women in the middle between stakeholders and developers who consolidate the wishes and needs from the former into a model for the latter to consume. The problem solving cycle is broken up in several parts synchronized by the requirements engineers. This results in several teams with separate knowledge structures. Each team will come up with different, locally "good" solutions. Not only will the one "good" overall solution be missed, friction between the teams will also impact the productivity of all.

VI. Conclusions for Managaing RE knowledge

Given the definition of knowledge above, the target of managing requirements knowledge in an organization is probably to bring the information about requirements into the right heads. The promising approach may be to foster communication and collaboration in a way that the information about all four knowledge areas flows through the organization. It seems, that the following guidelines can apply to managing requirements knowledge:

- Prefer steering communication and collaboration within an organization and interfacing with that organization over creating comprehensive requirements documentation.
- Prefer establishing interdisciplinary teams that cover all four knowledge areas over transferring requirements knowledge.
- Prefer creating an iterative process where problem and solution are understood in conjunction over isolating RE activities.
- Prefer designing a learning experience over teaching externalized knowledge.
- Prefer rich, exemplary information over essential information only.

References

[1] S. Hermann, „Herausforderung Wissensarbeit. Ist die Personalwirtschaft darauf vorbereitet?" In: Personal, Nr. 54, pp 10-14, 2002.

[2] J. Dirbach, M. Flückiger, S. Lenz, "Software entwickeln mit Verstand". D-Punkt Verlag, 2011

[3] S.W. J. Kozlowski, D. R. Ilgen, „Enhancing the Effectiveness of Work Groups and Teams." In: Psychological Science in the Public Interest 2006, Vol. 7, no. 3, pp. 77–124.