Journal of Universal Knowledge Management, vol. 1, no. 2 (2006), 130-136
submitted: 15/5/06, accepted: 15/6/06, appeared: 28/6/06 © J.UKM

Learning from own Experience and Advices in Literature

Andrea Herrmann
(University of Heidelberg, Germany
herrmann@informatik.uni-heidelberg.de)

Abstract: "A fool learns from his own mistakes, a wise person learns from the mistakes of others." The requirements engineering literature is full of Best Practices and other advices. But how to see the forest instead of the trees? This work proposes an approach for using own experience and experience of others in an integrated way for learning and reuse of process knowledge in RE.

Keywords: knowledge management, requirements engineering, process, best practices
Categories: D.2.1, D.2.9, K.6.3, I.2.6

1 Introduction

Requirements engineering (RE) is an activity taking place early in a software project, and its product is the requirements document which is the basis and input for later activities. Therefore, it is not surprising that among the most important factors of project success or failure, RE issues appear in high proportion, for instance among the "CHAOS ten", i.e. the ten major project success factors, identified by the CHAOS study of the Standish Group [Standish, 00]. As Glass [Glass, 02] and other authors emphasize, in software engineering the same errors are made again and again. Learning in RE therefore is important. Learning in RE can mean the reuse of requirements or can refer to the RE process. We here treat the RE process.

"Learning from success" means to search for Best Practices or "critical success factors" which significantly raised the probability of success of real projects. Equivalently, „learning from failure" means to identify risk factors which significantly raised the probability of failure of projects.

But listing critical success factors and risk factors is not sufficient for learning from experiences, as projects are different and take place in various environments. Such context information must not be lost. Some practices contribute strongly to project success in some cases and less in others, e.g. depending on the type of project risk [Couillard, 95], on the technological uncertainty [Shenhar, 96] or other context characteristica. Some critical success factors only have a positive effect in combination with others [Ebert, 05], [MacCormack, 03].

The importance of complete stories for memorizing, communicating and reusing experience was emphasized by Schank and Abelson [Schank, 95]. Case-based reasoning is the application of the story-telling principles on machine learning [Aamodt, 94]. Learning happens by adapting old stories to new situations. This old story can be one´s own project plan which is improved by incrementally adopting advices which were successful in similar projects, or by copying another´s project story adapting it to one´s own need. According to Schank and Abelson, the similarity of stories is judged by their goals, plans and themes.

We here propose an approach for combining data about own projects and documented project experience of others for better learning in RE. These two sources provide us a different quality of data. From our own experience, we can gather detailed data from projects which take place in our own working environment. But the number of projects is limited and after one, two years, project details are forgotten. The anecdotes which remain represent no objective image of our own past projects. We also get an intuitive feeling of what is important. On the other hand, literature contains data about much more projects, but project details are not given or confidential, and the context of the projects may be totally different from our own. Both types of sources provide for Best Practices which have proven their effectiveness by statistics on quantitative, objective project data, but also contain subjective conclusions and convictions. In the following we use the general term "advice" and take account of all types of advices.

When during practical work professionals do activity planning or problem-solving, they have evaluation criteria and a process in their minds for transforming such different information into expectations and decisions. The goal of this work is to make such an implicit process explicit. The stakeholders of this approach are software project team members, but they can also be supported by a knowledge management specialist.

The following concepts and process can be used for supporting the whole software engineering process, but we here are only interested in the RE activities.

2 Model for Project Data and Experiences from Literature

Our approach presented here is based on a project model which can be used for gathering project data as well as for classifying experiences from literature. Both types of data are then integrated and evaluated using practitioners´ experience. These data must in the same time tell a whole story without losing significant details and be indexed in a conceptual model for identifying doublets and for judging the similarity of stories, comparable to how it is done in case-based reasoning.

2.1 Project concept model

Our conceptual model of a project will not be presented in detail here, only its main concepts. This model was meant to be as simple as possible. Therefore, we compared several project models and identified the common elements. Our terminology is based on the PM BOK [PMI, 03]. In our model, the main elements of a project are activities and deliverables. Each activity is described by the resources (e.g. human resources, methods, tools) used by it, constraints (e.g. constraints on resources), sub-activities and principles (=rules about how to do the activity). Project deliverables are characterized by their content, properties and sub-goals (e.g. wanted properties of the deliverable). Also data concerning the whole project (input and output) are gathered: the goals and constraints of the project, its total resources, risks, and results.

2.2 Gathering project descriptions

The descriptions of own projects are gathered by a lessons learned analysis. This project (activities performed, deliverables produced, which resources were used etc.) is totally known, and also its context. In a first step, the project is modelled graphically and its completeness is verified by a review. In a second step, the project data are indexed according to the project model and entered into the project database. Our model has been applied to half a dozen case studies so far and was able to model the project story, including all information which was necessary for identifying origins of project problems and consequences of applied advices.

2.3 Classifying experiences from literature in an advice map

In literature, one finds others´ (often intuitive and subjective) project experiences or statistical evaluations of project data. They are of different quality and reliability, as was described in the introduction. It makes no sense to try to reconstruct the details of a case study project and its context from the fragmentary data contained in its publication. But usually, the project experience in literature is summarized in form of critical success factors or advices. They tell which activities to do how or which deliverables to produce how, if you want to be successful.

Therefore, we capture others´ experience in the form of advices. It is necessary to categorize them, for identifying doublets and for attributing them to activities and deliverables of real projects when they are needed. For grouping similar deliverables and activities, we distinguish the following product groups:

- Project Definition, Contracting & Planning
- Project Management, Reporting & Controlling
- Requirements Engineering (RE)
- Design
- Realisation
- QM & Testing

Within RE, the following activities and deliverables were defined:

RE Activity	**RE Deliverable**
Requirements Elicitation	Requirements Document
Requirements Documentation	System Test Cases
Requirements Analysis	Requirements Document, Analysed Version
Requirements Validation	Requirements Document, Validated Version
Requ. Management Definition	Requirements Management Policies
Requirements Management	

This classification is the basis for the construction of an advice map where each activity and deliverable is attributed advices concerning their resources, properties, principles, etc. So far, we have analyzed ten sources about RE process and 19 concerning project management (as many advices there apply to the management of all project activities). We extracted 120 RE advices from them, which were grouped according to the project model. Applicable on RE are also the 13 advices concerning

communication and 36 concerning human resources management. This collection certainly is not yet complete, as further advices are still found. As an example, we here present the advices concerning the activity “Requirements Management”:
Resources:

- change control board
- change request process
- Requirements Engineer available after the requirements phase

Sub-activities:

- document management
- control efforts that arise from changes
- use a central issue list
- keep requirements documents up-to-date
- do version management & document changes
- keep requirements document consistent with other documents (plans, design, product)
- maintain a traceability manual
- process improvement

Principle:

- keep requirements volatility low
- control requirements creep
- enforce cut-off dates for requirements for each release

It is important to add some meta information to the advices which is usually contained in the literature sources:

- Do the data refer to one or several projects? How many?
- Is the advice deduced from a statistical analysis of objective project data or based on subjective impressions?
- If it is a subjective impression: From how many persons does the advice stem?
- With respect to which success criteria has this advice been successful? (Remark: They can not be covered totally by the classical three “meet budget, schedule, and specification”.)
- In which context did this advice help improve the satisfaction of these success criteria? (The context can be characterized by: project duration, team size, company size, customer-specific or market-driven project, internal or external project, type of project risk, technological uncertainty, business sector. These are project characteristics which in empirical studies have been shown to make a difference concerning the effectiveness of advices.)

This information helps to judge the reliability and applicability of the advices to your own context. In the next section, these meta data are complemented by subjective evaluations of the advices.

2.4 Adding own experience to the advice map

The project team´s experience is not fully comprised by the project database, also their overall experience of what is successful should be used.

One’s own advices can be added to the advice map, either after statistical evaluation of the project data, or as a subjective advice.

We propose that each team members also adds the following evaluations to each advice:

- classification into “basic, intermediate and advanced” [Sommerville, 05]
- number of projects where it has been applied by themselves
- ratio of projects where it has been applied successfully (success rate)
- effect of the advice when used or when not used (not effect on overall project success criteria, but for instance on which property of which deliverable)
- When the advice has not been applied: Why not?
- When the advice did not have the wanted effect: Why not?
- Relationships to other advices (counteracts, supports, not applicable together with, replaces)

These evaluations help to further judge the reliability and applicability of the advices.

2.5 How to use this knowledge

The knowledge contained in the evaluated advice map and the project database are then used for planning RE activities (by the project manager or the requirements engineer), and during problem-solving or lessons learned analyses for identifying potential of improvement for critical activities and deliverables (by project managers, requirements engineers or managers responsible for process improvement). The advice map certainly is not the model of the ideal project, as some advices contradict each others, and some might be irrelevant for some specific context. It is a classified catalogue of advices.

For using this knowledge, we propose the following algorithm:

1. Statistical evaluations of the data of projects which took place within a similar context tell which activities, deliverables or resources have been critical in the past, i.e. were sources of project problems. Such general information is important for focusing attention when planning a new project. When searching for problem solutions during a project or identifying potential of improvement during a lessons learned analysis, the problematic elements of this project are determined within its project model.
2. Choose from the advices map those pieces of advice which refer to these critical project elements.
3. Check whether these advices are valid for the specific context and success criteria.
4. Check the success rate, effects etc. (section 2.4) and the meta data (section 2.3) of these advices and decide whether you want to apply them.

After having applied an advice more or less successfully, the experiences with it should be added to the database in the form of a new advice evaluation (section 2.4).

3 Summary and Further Work

Learning from others´ and own experience is usually done intuitively. This work proposes an approach which supports this learning in the field of process knowledge in RE. A project model is used for structuring both project experience from literature and own project experience. We so far classified 120 RE advices and data from half a dozen case study projects. The model was able to represent all data. Further advices and project data will be added for improving the comprehensiveness of the database.

The algorithm for identifying suitable advices as described in section 2.5 is a heuristics for manual selection. It probably can be improved by more sophisticated criteria like a similarity measure like it is used in case-based reasoning. So far, we defined no such measure for projects. Context variables and success criteria certainly are important factors of a similarity measure.

So far, we worked with several simple tools (graphics, word and table processing) for making first experiences in case studies in academic environment, but as now the needed data and the process are clear, the next step will be the implementation of a tool support and then the step to supporting practical work.

An important issue has not been treated in this work but will be essential for the application of the approach and tool by practitioners as part of their daily work: Does the benefit outweigh the effort? The process must be light-weight and the tool efficiently deliver the relevant answers to practical questions during project planning, problem-solving and lessons learned analyses. We think an authorization concept will also be important which guarantees that data from single projects are confidential and only overall (anonymous) project statistics and the advices is accessible to the other users.

Experiences from projects which introduced knowledge management systems in organizations show that information and training about the system, clear and light-weight processes, user and management commitment and fast results were major challenges [Komi-Sirvio, 02], [Rus, 02], [Schneider, 02]. Similar experiences were reported from introduction of new methods and by process improvement projects which also changed the way people work. Our approach can provide for fast results as the database initially is already filled with advice from literature. We especially like the idea that the knowledge gathering is done by a specialist [Komi-Sirvio, 02], as such a knowledge management specialist can help to satisfy the project team´s need of story-telling for communicating and reusing experience. This specialist´s tasks can be to coach the lessons learned workshop for gathering a coherent project story in a graphical form and to code this experience in the conceptual model for input into the project database (section 2.2), to add further literature advice to the advice map (section 2.3), to propose new advice on the basis of project statistics (section 2.4), and to search the database for potentially useful advice (section 2.5) for answering practical questions. The project team´s tasks are merely to tell the project story to the specialist during the lessons learned analysis (section 2.2), to evaluate advice (section 2.4) and to ask the specialist for advice (section 2.5) (or searching the database themselves, if they prefer). We believe that human communication and story-telling are important for knowledge management, and the database will play the role of a collective memory.

References

[Aamodt, 94] A. Aamodt, E. Plaza, Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches, AI Communications, vol. 17, no. 1, 1994

[Couillard, 95] J. Couillard, The Role of Project Risk in Determining Project Management Approach, Project Management Journal, Dec 1995, p.3

[Ebert, 05] C. Ebert, J. De Man J, Requirements Uncertainty - Influencing Factors and Concrete Improvements, in Proc. ICSE, 2005, pp.553-560

[Glass, 02] R. Glass, Project Retrospectives, and Why They Never Happen, IEEE Software, vol. 19, no. 5, 2002, pp.111-112

[Komi-Sirvio, 02] S. Komi-Sirvio, A. Mantyniemi, V. Seppanen, Toward a practical solution for capturing knowledge for software projects, IEEE Software, vol. 19, no. 3, 2002, pp.60-62

[MacCormack, 03] A. MacCormack, C.F. Kemerer, M. Cusumano, B. Crandall, Trade-offs between Productivity and Quality in Selecting Software Development Practices, IEEE Software, vol. 20, no. 5, 2003, pp.78-85

[PMI, 03] Project Management Institute (PMI), PMBOK Guide - A Guide to the Project Management Body of Knowledge, German translation, 2003.

[Rus, 02] I. Rus, M. Lindvall, Knowledge management in software engineering, IEEE Software, vol. 19, no. 3, 2002, pp.26-38

[Schank, 95] R.C. Schank, R.P. Abelson, Knowledge and Memory: The Real Story, In: Robert S. Wyer, Jr (ed), Knowledge and Memory: The Real Story, Hillsdale, NJ. Lawrence Erlbaum Associates, 1995, pp.1-85

[Schneider, 02] K. Schneider, J.-P. von Hunnius, V.R. Basili, Experience in Implementing a Learning Software Organization, IEEE Software, vol. 19, no. 3, 2002, pp.46-49

[Shenhar, 96] A. J. Shenhar, J. J. Renier, R. M. Wideman, Project Management: From Genesis to Content to Classification, Operations Research and Management Science (INFORMS), Washington, DC, May 1996, http://www.maxwideman.com/papers/genesis/genesis.pdf (last visited: april 2006)

[Sommerville, 05] I. Sommerville, J. Ransom, An Empirical Study of Industrial Requirements Engineering Process Assessment and Improvement, ACM Transactions on Software Engineering and Methodology, 2005, vol. 13, no.1, pp.85-117

[Standish, 00] Standish Group, CHAOS Report 2000, http://www.standishgroup.com /sample_research/PDFpages/extreme_chaos.pdf (last visited: april 2006)