Paper presented at the 20th International Conference on Software Engineering, April 19-25, 1998, Kyoto, JAPAN

Towards Better Software Projects and Contracts: Commitment Specifications in Software Development Projects

http://wwwseg.cs.hut.fi/

Jyrki Kontio[*]
Nokia Telecommunications
Information Networking Systems
P.O.Box 315
FIN-00045 NOKIA GROUP, Finland
E-Mail: jyrki.kontio@ntc.nokia.com

Olli Pitkänen[+]
Helsinki University of Technology
TAI Research Centre and
Lab. of Information Processing Science
FIN-02015 HUT, Finland
E-Mail: olli.pitkanen@cs.hut.fi

Reijo Sulonen
Helsinki University of Technology
TAI Research Centre and
Lab. of Information Processing Science
FIN-02015 HUT, Finland
E-Mail: reijo.sulonen@cs.hut.fi

ABSTRACT
Any software development project requires commitments from its participants. These commitments can include money, resources, deadlines, and specified functionality for the end product. We have developed a framework and a set of guidelines to support the specification of such commitments. We have evaluated the framework empirically in a series of case studies. The studies indicated that the framework addresses commitment specification issues that are normally not covered in project contracts and that the specification framework was considered beneficial by project representatives.

Keywords
Commitment, risk management, contract, goal

1 INTRODUCTION

A software project is a joint undertaking by two or more participants. As a minimum, users and developers are such participants, sometimes within a single organization, sometimes representing different companies. Cooperation to develop software requires commitment from all participants: users need to commit to providing specifications and feedback, as well as financial compensation; developers must provide resources, technical skills and commitment to schedule. Often these commitments are made early in the project when there is limited information available on many of the details that ideally should be specified. In this paper we present an approach for defining these commitments at the beginning of development work.

We use the term *commitment* to refer to goals, forms of cooperation and responsibilities that the participants agree upon in a project. In practice, a written project contract or a project plan is the most obvious manifestation of these commitments. We use the term *contract* to refer to the set of agreed commitments regardless of whether there exists a legal, written contract between parties, or the contract is an informal – even verbal – agreement on the commitments.

Commitment management has not been addressed from the perspective described in this paper directly earlier in the software engineering literature. It has been discussed from the point of view of personnel management, i.e., obtaining and maintaining individual engineers personal commitment to a project and the importance of "management commitment" is widely accepted for any project or endeavor [7]. However, the issues related to formalizing these commitments and negotiating about them has been rarely addressed.

Clearly, software contracts and project plans provide the basic templates for specifying project commitments. Some standards and examples of what project plans and contracts should contain exist, but they do not cover all aspects of commitment specification. Project plan standards mainly address schedule and cost goals, define the process used and require some basic form of risk management to be defined, but they do not address underlying motivation and problem management in projects [3]. Furthermore, they do not address the commitment specification *process* at all.

Some work in the area of risk management addresses commitment management indirectly. The SEI team risk management method [8] assumes that stakeholders can reach a consensus and all aspects of risks are discussed in an open forum. However, this approach does not account

---

[*] Jyrki Kontio is also affiliated with Helsinki Univ. of Tech.
[+] Olli Pitkänen is also affiliated with Opplex Oy Law Office.

for stakeholder conflicts, hidden agendas and uniquely different perspectives by stakeholders. Also, the SEI approach does not address the contractual aspects of risk management nor the negotiation process associated with it.

Barry Boehm's Win-Win model of risk management [1,2] takes into account different stakeholder perspectives and the negotiation process involved. While Boehm's win-win approach is the most widely known approach that addresses stakeholder expectations, it is limited to goal definition and there are no formal guidelines how commitments should be specified.

The Riskit method [4,5] also explicitly recognizes stakeholders and their potentially different expectations. The Riskit method contains an approach to trace risks from stakeholders and their goals to risks that are analyzed. Again, also the Riskit method fails to address the commitment specification content and process in detail.

Despite these advances, the current state-of-practice in industry is largely unaware of the need to specify and manage commitments. Most contracts and project plans are based on straightforward application of example templates and little attention is given to what commitments in each situation should be defined. Practitioners have little more than their own experience and intuition to support the drafting of better contracts.

This paper proposes a framework for selecting a strategy to specify these commitments and proposes guidelines to support the implementation of the selected strategy into the commitments that are made.

2 COMMITMENT SPECIFICATION

In this section we present a set of topics that can be defined in software project commitment specification and a model for selecting which topics should be addressed in different situations.

2.1 Underlying motives

In short, our commitment specification framework can be thought of as a series of questions that characterize the project:

- *Why* is the project being done?
- *What* is delivered and accomplished, when and for how much?
- *How* is the project done?
- *What if* something goes wrong?

The *underlying motives* explain why the project is being done. It refers to motivation and higher level business goals that parties have for participating in the project. Examples of such motives are "to provide better customer service to obtain competitive advantage", "increase profitability", and "reduce costs".

These underlying motivations serve two purposes. First, they help communicating the real interests and motives of participants so that project goals and project execution can be done and interpreted correctly. In a sense, the underlying motives describe the "spirit" of the contract, while other aspects of the contract define the "letter" of the contract. The second reason for documenting underlying motives explicitly is to provide interpretation rules in case the project contract is in dispute: a mediator (e.g., a judge) can use the underlying motives information to reach a compromise that is based on the recognition of motives and interests of parties. However, in some cases it may not be possible to document all underlying motives due to their confidentiality. Also, if these underlying motives are described in a project contract, great care should be taken so that they will not be more binding than intended. Yet, we suggest that all the stakeholders will gain, if the motivations are discussed and documented as openly as possible.

Project goals define what is delivered and what is accomplished by the project, when the work takes place and finishes, and how much resources and money is spent. In other words, project goals include project output (product specification), quality goals, schedule, cost, intellectual property rights, and any other type of attribute

Commitment specification topic	Aspects included
Underlying motives *Why is the project being done?*	Business goals Strategic directions and intentions Corporate vision and mission and their interpretations to the project.
Project goals *What is delivered and accomplished, when and for how much?*	Project output Schedule Cost Other goals and constraints
Process Specification *How is the project done?*	Administrative procedures Development process requirements Change control procedures Conflict resolution procedures Process standards or frameworks Risk management process definition
Risk and problem management *What if something goes wrong?*	Assumptions • Shared assumptions for the project Risk management • Scope of risk management • Accepted risks • Risk responsibility allocation Problem management • Breach criteria • Consequences

Table 1: Commitment specification topics

that is associated with the project (such as reuse or competence development). We use the term goal to refer to any of them, i.e., a goal is a general statement of purpose, direction or a concrete objective. Table 1 lists topics that are included in project goals.

Process specification defines how the goals should be achieved, how the parties cooperate and how the software is to be developed. This can include administrative procedures, such as how often meetings are held and how records are kept form these meetings, or specification of the development process itself, e.g. "a user interface prototype is delivered during the first phase of the project". Process specification can also make references to life cycle models or process frameworks and standards, such as ISO 9000-3 or SEI's Capability Maturity Model.

The *risk and problem management* topic covers proactive responsibilities to identify and avoid potential problems, as well as defining a priori what to do if some problems occur. Projects *assumptions* document the often unstated, implicit beliefs or assumptions that are shared by both parties, such as e.g., technology, feasibility, and external events. By making key assumptions explicit the parties can ensure that they share an understanding of these assumptions and their commitment to the project is based on awareness of these assumptions.

Risk management includes the definition of responsibilities for risk management and allocation of responsibilities for some risks, definition of risks that are accepted (i.e., shared and knowingly taken), and cost allocation principles for risk management.

We use the term *problem management* to refer to procedures and clauses that define what will happen if something goes wrong in the project. Examples of problem management clauses in contracts are compensations, penalty fees, and indemnities.

We have listed a summary of the commitment management topics in Table 1. The set of issues presented here and in Table 1 form the commitment specification template, a checklist that can be used when defining commitments in a project.

2.2 Situation Attributes Influencing the Commitment Specification

In most projects it is unrealistic to expect that all commitment specification topics can be defined exhaustively. Usually there is neither enough time nor information to do this. Instead, one should focus on topics that (i) are most relevant and (ii) can be specified.

There are many potential situation attributes that influence what commitment specification topics should be defined. They include size of the project, number of partners, whether the project is in-house development, life cycle model used, level of reuse, use of new technologies, etc. [6]. However, at an early stage in a project the overall level of risk is often the most critical situation attribute. Thus it should have the most influence on commitment specification.

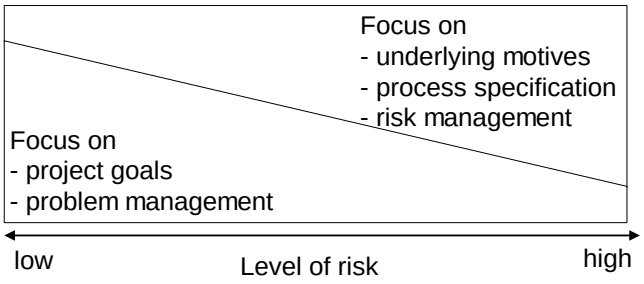

Figure 1: Commitment specification topics and risk

Level of risk can be thought of as a subjective estimate of overall risk level in a project compared to other projects. Theoretically, it can be defined as the sum of expected utility losses of all risks in a project, although this can rarely be calculated in practice [4].

It may be futile to make fixed commitments to goals in a high-risk project as these risks may make goals unreachable. A better strategy in such a situation is to focus on *how* the project is done: frequent meetings, change management procedures, reporting, information exchange, etc. As Figure 1 illustrates, project with low level of risk can commit to goals more firmly. High-risk projects can have less emphasis on project goal definition and, instead, motives, process, and risk and problem management are relatively more important.

3 EMPIRICAL STUDIES

We have evaluated the proposed framework empirically by using two complementary evaluation approaches. The feasibility of our approach was studied by applying the framework to a set of seven existing software project contracts, and comparing whether all commitment specification topics were covered. This was done as postmortem analysis and the sample set of contracts was selected from the customer base of a software law consulting firm. All projects had been defined and negotiated without our involvement. The cases represented a wide range of organizations, projects and forms of cooperation.

In the feasibility study we analyzed the project plans and contracts on these projects and coded their contents according to commitment specification topics. The analysis of the results showed that all projects focused on defining their goals, although there was a large variance in the level of detail and the type of goals that were included. The process specification was also well addressed by most

projects. Risk and problem management topic was addressed in detail by three projects and even the remaining projects had addressed it in some detail.

The underlying motivation topic was only addressed by three projects and remained rather abstract when it was defined. Under risk and problem management topic, assumptions were only defined in two projects. Risk management process and identification of risks was only addressed by one project, while one other project had documented some risk mitigating actions or points.

Overall, it seems that underlying motives and risk and problem management are the areas that are most often left unspecified. Although our sample size is too small to draw any general conclusions, it seems that the level of risk does not correlate with whether these issues are covered or not.

The second study attempted to further evaluate the feasibility of the approach and its impact on projects. We chose to perform postmortem what-if scenario analyses with the representatives of two projects. The goal of this study was to gain initial feedback on practitioners opinions on the usefulness or potential benefits of the approach. Both projects estimated that the further specification of goals, risk management and problem management would have taken approximately two to three working days of effort. In one of the projects this was actually done and resulted in concrete changes in the project: the new, revised goals were issued as the official goals of the project, the updated risk management process was implemented and used, and the problem management procedures were revised.

The project representatives concluded that further specifications could and should have been made in the beginning of the project. They recommended that the commitment specification process will be followed in the next projects that are initiated.

In both cases we inquired the project representatives what motivated them to specify the topics they had specified initially in the project. They seemed to have followed a simple rule of thumb: fill in the required slots in the project or contract template if the information is easily available. In other words, it seems that specification templates and information availability strongly influence the content of commitment specification. Neither aspect necessarily correlates with what *should* be defined in the beginning of a project. As our model suggests, project risk may determine what topics should be defined more thoroughly. A single specification template or information availability may be a poor guide in this specification process.

4 CONCLUSIONS

The state-of-the-art and practice in commitment management today is mainly based in practitioners intuition. Given the importance of commitment management and specification, intuition alone may not be enough. As in many other areas in software engineering, practical guidelines and methods should be developed to support critical areas of commitment management.

In this paper we have presented a commitment specification approach. Through our analysis of common guidelines and standards and analysis of seven cases we suggest that the framework advocates a slightly different approach to commitment specification from what the state-of-practice in industry is. The initial empirical data suggest that the more focused specification of commitments is necessary and beneficial. The approach is simple and inexpensive to use: by raising awareness and introducing the commitment specification template most project managers can improve the commitment specifications within hours of effort.

Clearly, commitment management and our proposed approach for it require additional studies and further development. However, we believe that the approach presented in this paper is a concrete first step towards more explicit and better commitment specifications. More details on this framework is available on a separate report [6].

ACKNOWLEDGEMENTS

We would like to thank the participating organizations for their cooperation and willingness to share their experiences and data.

REFERENCES

1. B.W. Boehm and Bose P., A Collaborative Spiral Software Process Model Based on Theory W. Proceedings of the 3rd International Conference on the Software Process. IEEE Computer Society. Washington, DC.
2. B.W. Boehm and Ross R, Theory W Software Project Management: Principles and Examples *IEEE Transactions on Software Engineering*, vol. pp. 902-916, 1989.
3. IEEE. *IEEE Standard for Software Project Management Plans, Std 1058.1-1987*, New York: IEEE, 1987.
4. J. Kontio, The Riskit Method for Software Risk Management, version 1.00, CS-TR-3782, 1997. Computer Science Technical Reports. University of Maryland. College Park, MD.
5. J. Kontio and V.R. Basili, Empirical Evaluation of a Risk Management Method 1997. Proceedings of the SEI Conference on Risk Management. Software Engineering Institute. Pittsburgh, PA.
6. J. Kontio and O. Pitkänen, Commitment Management in Software Projects and Contracts 1998. Technical Report B-138/98, ISBN 951-22-3959-0. Helsinki University of Technology. Espoo.
7. P.L. McDoniel, J. Palko, and T.P. Cronan, Information

systems development: issues affecting success *J.Comput.Inf.Syst.*, vol. 34, pp. 50-62, 1993.

8. G. Pandelios, Software Risk Evaluation and Team Risk Management 1996. Tutorial Presentations at the 1996 SEPG Conference. Software Engineering Institute. Pittsburgh, PA.