

AGILE COMMITMENTS: DEALING WITH BUSINESS EXPECTATIONS RISKS IN AGILE DEVELOPMENT

Mauricio Concha, Marcello Visconti and Hernán Astudillo
Departamento de Informática, Universidad Técnica Federico Santa María
Av. España 1680, Valparaíso, Chile

Keywords: Agile Development, Commitment Management, Risk Management.

Abstract: Agile methods have been proposed to increase customer satisfaction and deliver business value early, yet usually don't focus on progress visibility other than software deliverables. However, many customers demand risk visibility over the main aspects that define their expectations: functionality (scope), budget, time-to-market, and product quality. This article proposes an *agile commitment framework* based on structured definition and follow-up of commitments among customers and developers. The framework uses commitment management to improving risk management by enhancing visibility of business expectation risks, by providing a negotiation baseline among customers and developers, and by allowing mitigating action when appropriate. Finally, we summarize several case studies run to evaluate the proposed framework in academic and industrial settings.

1 INTRODUCTION

There are clear trends in software business: user expectations for quality are increasing, software cycle times are shorter, systems are becoming more complex and more integrated, new business opportunities require exploration of new product concepts, and agile development approaches have become very common in industry. Since software is a strategic element to support the business process within organizations, software alignment to business goals is an important aspect to manage.

Customer business expectations lead to the development of software, and those expectations are defined at the beginning by customers in terms of: functionality (scope), time-to-market, budget, and product quality. These are the aspects the customer is interested in, and if some of them are missed, it will lead to an unsuccessful project.

Agile methods are oriented to customer satisfaction and to deliver business value early, but if flexibility and adaptability are not managed during the project, agile methods may not assure achievement of all business expectations. Therefore, it is necessary to introduce a risk-based approach to improve risk management in agile projects.

This article is organized as follows: section 2 briefly describes agile development; section 3

highlights business risks in agile projects; section 4 presents commitment management; section 5 introduces the *agile commitment framework*; section 6 describes how to monitor and obtain visibility with the framework; section 7 shows early results obtained from using the framework; and section 8 presents future work and conclusions.

2 AGILE DEVELOPMENT

In software companies the strongest motivation to switch to agile development is failure in previous projects. This prompts the organizations to look for and try new, potentially more effective methods for software development.

Agile methods (Alleman and Henderson, 2003), such as Extreme Programming and Scrum, have been offered as a way to meet the needs of rapid external changes in dynamic market situations, lower defect rates, and reduce development times.

Agile methods are *adaptive* rather than *predictive*, unlike traditional methods where most of the software process is planned in certain detail for a large time frame. The plan-driven approach works well only if change is limited, and the application domain and software technologies are well understood by the development team.

The problem of change is exacerbated (Kontio et al., 2004) by long development cycles that yield code that may be well written but does not meet user expectations.

Agile methods were developed to adapt and thrive on frequent changes. The rapidly changing Internet-based economy demands flexibility and speed from software developers, something not usually associated with plan-driven development. The main motivations to use agile methods, as expressed in the “agile principles” (Agile Manifesto, 2001), are: increment of customer satisfaction; flexibility and change acceptance in requirements; frequent delivery of valuable working software to customers; and collaboration of business people and developers.

Many development teams have successfully used agile development to build quality software, but often these projects have failed to effectively contribute to overall company success. This failure is due to the fact that most company’s strategic planning processes have not been aligned to take advantage of the flexibility and adaptability of agile development (Rand and Eckfeldt, 2004).

3 BUSINESS EXPECTATIONS RISKS IN AGILE DEVELOPMENT

In commercial environments, business goals are usually managed through a formal contract signed between developers and customers. In particular, companies that outsource development are used to working under contract in order to manage the customer-developer relationship.

However, defining a traditional contract can be difficult in agile methods, because it would require to determine and fix system details early on during the development process. Instead, agile methods aim to achieve business goals using iterative development and close communication between customer and developer.

Indeed, a key factor for the appropriateness of agile methods is contractual obligation; if the requirements for work to be performed are part of a legal contract, an agile method may be inappropriate since requirements are malleable (Cram and Bohner, 2005).

The challenge of increasing risks visibility when using agile methods is how to define early in a project the rules and conditions that will determine its development, without hampering the iterative and exploratory nature of agile methods. A customer

may be sure that agility can help, but still needs to manage and mitigate the risk of failing the business expectations (about functionality, time-to-market, budget, and software quality). And this means that developers need to understand, and commit to, the project’s business goals, not just to mere deliverables.

Some work has been already devoted to defining business goals for agile projects and to enhance risk-visibility between customers and developers: Agile Contracts (Beck and Cleal, 1999), Agile Procurement (Jamieson et al., 2005), and Risk-Driven Method for XP Release Planning (Li et al., 2006). However, we will argue that a more promising starting point is *commitments management* (Kontio et al., 1998).

4 COMMITMENTS MANAGEMENT

Software development is always a challenging undertaking and it requires high commitment from individuals who participate in it. Software development often involves new technology, challenging or unknown requirements, and tight schedules – making it particularly prone to several types of risk (Kontio et al., 2004)(Hartmann and Dymond, 2006).

The term *commitment* describes goals, forms of cooperation, responsibilities, decisions, and so on, that stakeholders agree upon in a project; commitments scope may include all critical aspects in the project.

Commitments management (Kontio et al., 1998) is an approach that uses commitments among customers and developers to define a list of agreements as baseline for the project, with the goal of mitigating the risk of losing sight of the original project motivations. The commitment specification defines all agreements and establishes a common view of the project among stakeholders

Commitments management (as part of project management) is the specification, formalization, and follow-up of commitments during the whole project, with the purpose of aligning the final product with the business strategy and goals that motivate the software project.

Software contracts and plans provide partial specification of the commitment agreed for the project. Project plans define budget and timeline, specify the software process and the risk management, but don’t consider the business underlying motivations and how to manage the problems. Thus, contracts and project plans are incomplete commitment specifications.

The commitment management process has been characterized in the following process areas (Kontio et al., 1998):

- **Business motivation.** Why is the Project being developed?
- **Project goals definition.** What is delivered and accomplished, when and for how much?
- **Process specification.** How is the Project developed?
- **Risk management.** What are the risks and what do we do?

In most projects it is not realistic to expect that all commitment specifications topics can be defined exhaustively, and usually there is neither time nor information to do this. Instead, focus should be on topics that are most relevant and can be specified. In the earliest stages of a project, the overall level of risk is often the most critical situation attribute. Thus, it should have the highest influence on commitment specification.

Currently, commitment specification is mainly based on the participant intuition. Given the importance of commitment management and specification, intuition alone may not be enough. As in many other areas of software engineering, practical guidelines and methods should be developed to support critical areas of commitment management.

The relationship between customers and developers requires consistent practices so partners develop confidence that commitments will be honored, even if individuals change. This in turn requires creative agreements that do not try to cover every eventuality, but instead provide ways to deal with unpredictable future events in a manner that both sides will perceive as fair and equitable (Schuh, 2005).

5 AGILE COMMITMENTS FRAMEWORK

We propose an *agile commitments framework* that adapts commitment management to manage risk on business expectation for *agile methods*. The specific objectives of this framework are to:

- Define and specify commitments among participants.
- Define and agree on the underlying business motivations.
- Manage and control the agreed-upon commitments during the whole project.

- Improve risk management through risk visibility on the business expectation about functionality (scope), quality, budget, and time-to-market.
- Provide a negotiation baseline for customers and developers.

The agile commitments framework has two components (see Figure 1): a **conceptual schema framework**, which defines the framework itself and describes its structure; and the **instantiation guide for project level**, to be used by managers to implement agile commitments in specific projects.

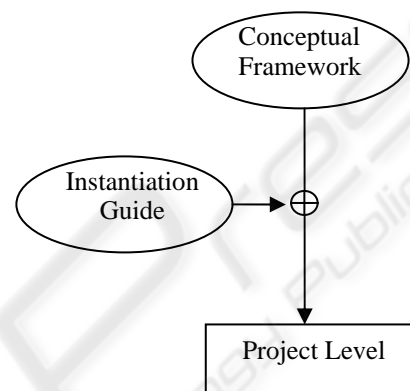


Figure 1: Agile Commitments Framework.

5.1 Conceptual Schema Framework

Agile commitments are based on the commitment specification for software projects proposed by Kontio (Kontio et al., 1998), and are divided in 4 process areas, each one with specific goals (see Table 1).

The issues that reduce the uncertainty of a new project using agile methods are defined with the elements of the framework. Commitments can be negotiated at the beginning and during the project, and provide a common and agreed upon outlook to customers and developers.

The framework structure is based on the “continuous representation” from CMMI (Software Engineering Institute - CMU, 2006), and defines specific goals in every process area. The four process areas are described in the following sections.

5.1.1 Business Motivation

The purpose of the business motivation area is to define *why the project is being developed* and what *business goals and expectations* are settled by the customer. The specific goals are:

- **Strategic directions and intentions:** Specify for the project the directions, intentions, and

business strategy. Therefore, the business expectations are defined at the higher level. Thus, it is possible to agree upon a common perspective about the project.

- **Business value goals:** Establish the business value goals for the software that will be developed, the expected final product, and the expected economic effect in business.
- **Time-to-market:** Specify the business opportunity defined for the project. If there are either a deadline or time-to market defined for the software. It is important to agree on the opportunity cost that will exist if the project is delayed.

Table 1: Conceptual Schema Framework.

Process Areas	Specifics Goals
Business motivation	Strategic directions and intentions
	Business goals
	Time-to-market
Project Goal	Deliverables and Iterations (value added)
	Schedule and times
	Cost and budget
	Quality
Agile Process Specification	Project management
	Agile process definition (standard or framework)
	Conflict resolution procedures
	Change control procedures
Project Risk Management	Shared assumptions for the project
	Risk Analysis and identification
	Scope of risk management
	Accepted Risks
	Risk responsibility assignment

5.1.2 Project Goal

The purpose of the project goal area is to define *what will be delivered and accomplished*, when and for how much. The specific goals are:

- **Deliverables and iterations:** Estimate the number of iterations, and the time for every cycle. Define the scope of all deliverables at each iteration and agree on business value added for every intermediate product.
- **Schedule and times:** Define the schedule and global times for the project, considering the number of iterations and when the software is required. This goal has key importance in project management, and it is related to the “time-to-market” business goal.
- **Cost and budget:** Establish an agreement over the total cost estimation for the project, considering the iterations and resources needed.

This must be a realistic agreement among the participants.

- **Quality:** Define how the quality in the project will be assessed, this allows the definition of acceptance criteria for the deliverables.

5.1.3 Agile Process Specification

The purpose of the agile process specification is to define *how the project will be developed* and what methodology will be used. The specific goals are:

- **Project management:** Define how the project management will be carried out, which depends on the agile method selected for the project. If the project is part of a larger one, it is required to define how the project integration will work.
- **Agile process definition** (standards or frameworks): Establish which agile method will be used, whether some framework will be followed, and what level of compliance with standards is defined for the project.
- **Conflict resolution procedures:** Establish how the problems will be addressed during the project, and who is going to make the important decisions. This goal must be aligned with the business goals of the project.
- **Change control procedures:** Define the process for change evaluation and incorporation, considering that the agile project must be open to changes.

5.1.4 Risk Management

The purpose of the risk management process area is to *identify potential problems* before they occur, and agree upon activities o mitigate adverse impacts on achieving objectives. A continuous risk management approach is applied to effectively anticipate and mitigate the risks that have critical impact on the project. The specific goals are:

- **Shared assumptions for the project:** Define explicitly the assumptions for the project, and how those must be known and agreed upon for every participant in the project. For example, technology, feasibility, external factors, etc.
- **Risk analysis and identification:** Identify and analyze risks in the project to determine their relative importance and potential impact.
- **Scope of risk management:** Define the scope of risk management in the project, and its importance during development.
- **Accepted risks:** Specify the list of risks that will be mitigated and what activities will be planned and executed in order to reduce the impact and occurrence probability.

- **Risk responsibility assignment:** Define the responsibilities in risk management, and who will be in charge of each planned action

5.2 Instantiation Guide

The instantiation guide defines the steps to follow to implement agile commitments in a specific project. During the instantiation activities, customers and developers consider the project characteristics and the particular business expectations.

As part of the framework, we have defined some document templates to support the instantiation process; they provide a standard tool to specify and control commitments for the project.

6 ACHIEVING RISK VISIBILITY IN A PROJECT

The execution of the commitment management framework is oriented to measure risk qualitatively; thus, the main problem is to decide which risk metrics should be gathered during the project.

Risks are defined as a possibility of loss or negative impact on a project. Risks can be evaluated during the project with a probability value and potential impact. In the context of this work, risk will be considered as the possibility and impact of not achieving the customer’s business expectations.

A *risk metric* is an objective measure associated with a risk factor to be mitigated, although its measurement may be problematic (Boehm and DeMarco, 1997). The risk factors addressed by the process framework are based on the business value expectations and goals: functionality, time-to-market, budget, and product quality.

A *risk factor* is a potential problem, characterized by the probability of occurrence and a potential loss (of life, money, property, reputation, and so on). In the previous factors, customers can define the potential loss, and therefore the problem becomes how to measure the probability of occurrence, and how this probability changes during the project execution.

For this specific process framework, the available approaches for assessing risk, and hence validating the process framework, are:

- **Initial Scenario:** At the start of the project, all business value goals (functionality, time-to-market, budget, and quality) must be established in terms of *qualitative metrics*, as well as *potential losses* incurred if a business value goal is not met.

- **Current Risk:** The perceived risk at the moment of the measure; it is a subjective assessment. It can be measured using the *perceived probability*, and it must be measured along the whole project execution.
- **Risk Incurred:** The *probability of failure* that the project faced but eventually avoided. Therefore, the problems did not occur because the mitigation efforts worked.
- **Final Scenario:** At the end of the project, it is possible to *compare* the initial business goals taken in the “Initial Scenario” with the final values obtained for business goals (functionality, time-to-market, budget, and quality).

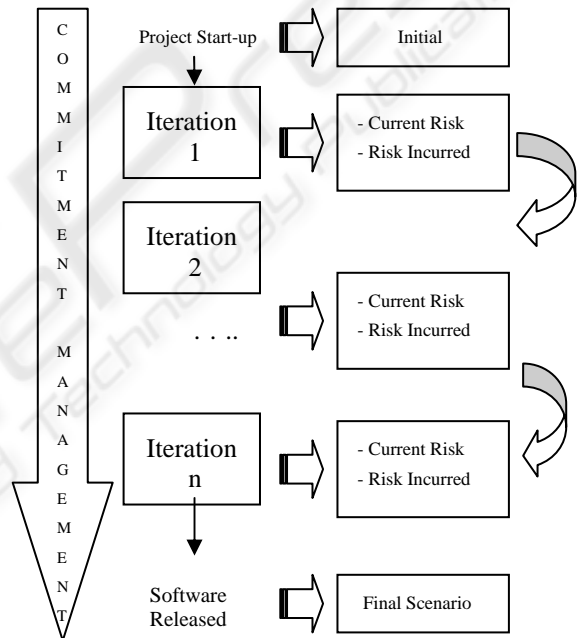


Figure 2: Continuous Risk Visibility.

Figure 2 shows the point in time where each assessment is carried out, according to the agile commitments framework.

At the end of each project, two important metrics can be obtained: the *total risk* incurred during the project for the business goals fulfillments, and the *variation in the final results* obtained for the business goals according the customer evaluation. Also, the customer is able to assess whether this framework was useful with the purpose of reducing risk exposure, and if the business goals have been met.

7 CASE STUDIES: EARLY RESULTS WITH THE AGILE COMMITMENT FRAMEWORK

The agile commitment framework has been evaluated for feasibility and effectiveness through several case studies at two different levels: at the **Conceptual Framework Level**, the process framework has been validated using *expert judgment*; and at the **Project Level**, the framework has been *instantiated and verified in real projects*, in two different case studies: an academic exercise, and a real project developed in an international maritime transportation company.

7.1 Conceptual Framework Level Evaluation

At the conceptual level, a case study was carried out by instantiating the framework in 54 different agile projects developed in software companies. This case study introduced the framework in real projects at their initial stages; it collected information about the instantiation process itself, the results obtained, and the evaluation by IT professional that used the framework, according to their expertise and expert judgment.

The agile commitments framework was instantiated for the projects in the case study only for the “Initial Scenario”, and this framework was not used in the other phases of the development, because it was not considered as relevant for this level of conceptual evaluation.

The case study included two groups of projects, classified according to the professional level of the participants: the first group had 48 graduate-level IT professionals and the second one had 35 IT professionals with undergraduate level software engineering studies. The main conclusions from this case study (Concha, Visconti and Astudillo, 2007) are summarized in Table 2. In general terms, the framework has been successfully used for the initial phase of these projects being applied correctly in all agile methods used in the case study, also it was possible to confirm the framework as a platform to define commercial conditions between customer and developers, and finally there were no negative observations in the conceptual evaluation.

In this case study, we did not verify with the participant companies how every project finished because that was out of scope of this proposal, nor did we verify if agile development had been correctly applied. The only result we were interested in from this case study was the instantiation of the

framework for the “Initial Scenario”, and the expert opinions.

Table 2: Conceptual Level Conclusions.

Perspective	Description of Results
Agile Development	The framework is applied correctly in all agile methods used in the case study. In total, 14 Scrum, 15 XP, 3 AUP, 1 FDD, 21 incremental & iterative ad-hoc development.
Risk Management	IT professionals (experts) confirm as valid using the commitment management in order to support risk management for business expectations issues.
Negotiation	For outsourced development projects, it is possible to confirm the framework as a platform to define commercial conditions between customer and developers, providing a useful negotiation baseline.

7.2 Project Level Academic Evaluation

In the first case study, the framework was instantiated in 8 academic projects, each of them 3 month long, and developed as part of a “Software Production Workshop” for professional engineering seniors. The workshop theme was Semantic Web development using agile methods, specifically Feature Driven Development (FDD).

Table 3: Project Level Conclusions (case study 2: Semantic Web and FDD projects).

Perspective	Description of Results
Agile Development	The framework is applied to FDD with no observations.
Risk Management	It allowed risk visibility during the whole project to instructor and students.
Negotiation	It allowed a baseline for agreements and commitments between instructor and students.
Project Management	The framework leads the commitment management; therefore it is carried out consistently during the project development.

The objective of this case study was to use the agile commitments framework in the projects and to learn about the risk visibility during the projects. The framework was used in the complete project life cycle.

At the workshop end, 6 out of 8 projects had instantiated the framework correctly. Consistently, the 2 projects that were unable to instantiate the framework had a poor overall workshop evaluation. The results of this case study from the instructor (stakeholder) viewpoint are summarized in Table 3.

7.3 Project Level Industry Evaluation

The second case study at project level corresponds to a real development project in industry. The proposed method was applied to a Web-based application project in a shipping company. The agile method used was FDD. The 2-month project was developed under outsourcing, and only the domain experts belonged to the company. During the project, and using the agile commitments framework, the project manager was able to obtain the risk visibility required. This risk visibility allows the stakeholders to make decisions in order to mitigate the risks, particularly in terms of quality and scope (functionality) expected for the final software product. Those elements were critical in the customer business expectations for this particular project.

Table 4: Project Level Conclusions (case study 3: industry project).

Perspective	Description of Results
Agile development	The framework is applied to FDD with no observations.
Risk Management	The framework is a useful tool to assess the risk exposure level in the different project stages; this allows the managers the timely implementation of the corresponding mitigation strategies. The subjective risk assessment proposed by the framework could be improved through objective measuring.
Negotiation	The framework enables a common understanding between stakeholders and developers about the business goals of the project, providing a good baseline to negotiate commercial conditions. Also, it allows the customer to control the project during its iterations.
Project Management	The framework supports the implementation of commitment management in an agile project, contributing to control the progress of the project based on the commitment fulfillment, and providing the required risk visibility on the business expectations risks.

In the commercial aspects, the framework enables a common understanding between stakeholders and developers about the business goals of the project, providing a good baseline to negotiate commercial conditions.

The results for this industry case study from the stakeholder viewpoint are summarized in Table 4.

Both case studies allowed us to receive feedback from the customer side on two evaluation levels: 1) the conceptual level, where the framework has been assessed by IT professionals considered as experts in the area because of their expertise in project management; and 2) the project level, where the framework has been instantiated and used in real projects during the full life cycle of a number of academic projects as well as an industry project.

8 CONCLUSIONS AND FUTURE WORK

Agile methods can be aligned to business goals using commitments management as a complementary activity, to mitigate risk to business value expectations. In this article, we have defined an approach that can be used regardless the agile method implemented in the organization. The proposed solution corresponds to the integration between agile methods and a commitments management technique.

Commitments management does not modify the essence of agile methods; it only supports them with complementary practices. We can see at least four benefits from using the proposed agile commitments framework: 1) the framework is well-defined and generalized for any agile method; 2) the framework allows customers and developers to develop a negotiation baseline, as an effective and *agile* alternative to contracts; 3) the framework improves risk management through risk visibility on the business expectation elements: functionality (scope), quality, budget, and time-to-market; and 4) the framework provides a risk-driven decision support tool to the customer during the whole development process.

Concerning potential improvements to the agile commitment framework, an issue to consider is to avoid the subjective risk assessment during the project; the framework could suggest some objective way to measure risk using, for example, a value based technique. Another issue for further discussion is to determine if the framework is suitable for project with short iterations; in such cases there is not enough time to implement the commitment management.

Ongoing work focuses on extending the agile commitments framework by defining an intermediate instantiation level, to allow managers to define process instances for an organizational domain, which can then be instantiated for specific projects in a project portfolio under the same business condition.

Schuh, P., 2005. *Integrating Agile Development in the Real World*. Programming Series, Charles River Media.

Software Engineering Institute (SEI) - CMU, 2006. *CMMI for Development, Version 1.2. Technical Report CMU/SEI-2006-TR-008*.

REFERENCES

- Agile Manifesto*, 2001. Retrieved April 1, 2007 from <http://www.agilemanifesto.org/>.
- Alleman, G., Henderson, M., 2003. "Agile Development Work in a Government Contracting Environment." In *Procs. Agile Development Conference (ADC'03)*.
- Anderson, D., 2004. *Agile Management for software engineering*. The Coad Series.
- Beck, K., Cleal, D., 1999. "Optional Scope Contracts." *Tech Report*. Retrieved from www.xprogramming.com/ftp/Optional+scope+contracts.pdf
- Boehm, B., DeMarco, T., 1997. "Software Risk Management." *IEEE Software*, May/June 1997.
- Concha, M., Visconti, M., Astudillo, H., 2007. Agile Commitments: Managing Business Expectation Risks in Agile Development Projects, In *Technical Report 2007/03, January 2007*. Departamento de Informática, Universidad Técnica Federico Santa María.
- Cram, M., Bohner, S., 2005. "The Impact of Agile Methods on Software Project Management." In *Procs. International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS'05)*.
- Hartmann, D., Dymond, R., 2006. "Appropriate Agile Measurement: Using Metrics and Diagnostics to Deliver Business Value." In *Procs. of AGILE 2006 Conference (AGILE'06)*.
- Jamieson, D., Vinsen, K., Callender, G., 2005. "Agile Procurement: New Acquisition Approach to Agile Software Development." In *EUROMICRO -SEAA'05*.
- Kontio, J., Hoglund, M., Ryden, J., Abrahamsson, P., 2004. "Managing Commitments and Risks: Challenges in Distributed Agile Development." In *Procs. 26th International Conference on Software Engineering (ICSE'04)*.
- Kontio, J., Pitkanen, O., Sulonen, R., 1998. "Towards Better Software Projects and Contracts: Commitment Specifications in Software Development Projects." In *Procs. International Conference on Software Engineering (ICSE'98)*.
- Larman, C., 2004. *Agile and Iterative development: A manager's guide*. Addison Wesley.
- Li, M., Huang, M., Shu, F., Li, J., 2006. "A Risk-Driven Method for XP Release Planning." In *ICSE'06, May 20-28, 2006, Shanghai, China*.
- Rand, C., Eckfeldt, B., 2004. Aligning Strategic Planning with Agile Development: Extending Agile Thinking to Business Improvement. In *Procs. Agile Development Conference (ADC'04)*.