

---

# Agile Contracts

## Position Paper for “Agile Contracts” Workshop at OOPSLA-2003

**31-August-2003**

*Michael Hirsch  
Zühlke Engineering AG  
Wiesenstrasse 10a  
CH-8952 Schlieren, Switzerland*

*Hirsch.Michael@acm.org  
Phone +41 1 733 66 11 (Office)  
Phone +41 52 301 30 46 (Home)*

---

## 1. Background

Zühlke Engineering AG, the company I am working for since 10 years, is an independent systems development contractor with offices in Zurich (Switzerland), Frankfurt (Germany) and London (UK). The company was founded in Switzerland in 1968 and today employs about 240 employees. The main business of Zühlke is development of custom software and hardware systems. About 50% of our customers are in manufacturing industries, 30% in financial services, and the rest is in tele-communications, transportation and life sciences.

I am head of software project management, which means responsibility for (a) hiring and training software project managers and (b) selecting and improving the company's software development process. In addition to that I work as project manager on inhouse development projects, as software project management consultant for our customers, and as trainer for courses on software processes and software project management.

Since most of our business is custom development, adequate contracts are of utmost importance to us. The majority of projects we do are technically and organizationally very challenging. This is because our customers tend to build less complex systems themselves and outsource only what they consider too risky to do themselves. Very often requirements are vague and unstable when we are approached. Even so, customers invariably want to know (at least) the approximate cost of developing their system. We found that agile development practices work very well under these circumstances, both for software and for combined hardware and software systems.

## 2. Development Processes and Contracts

Structure and contents of contracts depend strongly on the development process used. Since 1998 we are using a lightweight and agile implementation of the Rational Unified Process (RUP) for all software development work at Zühlke<sup>1</sup>. I have attached a description our development process to this position paper (title: “The Zühlke Software Development Process”).

---

<sup>1</sup> I am presenting a tutorial on how to use RUP for agile software development at OOPSLA 2003 (tutorial #54)

In order to reduce risk for the customer and for ourselves, we are breaking down a project into at least two different contracts:

- One contract for the Inception phase (the “Inception” contract)
- One contract for the Elaboration, Construction and Transition phase (the “Development” contract).

On large and / or very risky projects, we are using three or more contracts, for example, one for each phase of the project.

### **3. Contract Types**

#### **3.1 Inception Contract**

The purpose of the Inception Contract is to scope the project and come up with a first estimate of total project cost. This contract is normally for time and materials. The results are:

- A prioritized list of system features
- A description of the most important use cases
- A risk analysis
- An outline of the proposed system and software architecture
- An estimate of total project effort and cost

The results delivered under this contract are used by the customer to decide if the project is continued or not. The total effort for the inception contract is normally between 5% and 15% of the total effort for the project. The effort and cost estimation done under this contract consists of two scenarios, a “best case” and “worst case” scenario.

#### **3.2 Development Contracts**

For system development (i.e. for the Elaboration, Construction and Transition phases) there are different approaches in terms of contracts. However, all types of contracts require:

- Iterative and incremental development of the system with frequent intermediate releases which are delivered to the customer
- Customer participation in iteration planning
- Customer feedback for each intermediate release
- Stable iterations, i.e. the goals of an iteration in progress are not changed. Changing project direction occurs when the next iteration is planned.

A development contract specifies start and end dates of planned iterations, as well as the major goals for each planned iteration. There is no assignment of features and use cases to iterations in the contract yet. Delivery dates for intermediate releases are specified in the contract as well.

The types of development contracts we are using are:

- One times and material contract for Elaboration, Construction and Transition.  
This contract type puts the customer into control over the direction of the project. The customer has the ultimate say of what goes into an iteration and when the project is finished. The advantage of this contract for the customer is that he can basically change the direction of the project at any time. The disadvantage is that development cost is open ended.

- One fixed price contract for Elaboration, Construction and Transition.  
With this contract type, we work on a “build to budget” basis. The cost the customer is willing to invest in development is fixed. What varies is the set of features built. The list of features created under the Inception contract is prioritized into “must have”, “important” and “nice to have” features. We commit to build all “must have” and — for example — 50% of the “important” features. Which “important” features we build is our decision. This way the customer gets the assurance of a fixed feature set (the “must have” features) for a fixed price, and we still have some freedom of adjusting the feature set built (the “important” features) depending on how the project progresses. Changes to the agreed set of “must have” features are possible at any time under a mutually agreed and straightforward change request process. Change requests are handled the traditional way, i.e. we prepare a brief offer for one or more change requests, which is used by the customer to decide if the change request is worth the additional cost and time. Change requests are always implemented in future iterations, never in a running iteration. The reason is not to jeopardize the delivery date of the iteration in progress.
- One contract for each iteration.  
We tried this approach once, with unsatisfactory results. The problem of this approach is that it almost always takes the customers organization too long to sign a contract in time for the next iteration. The result is gaps of a few days to a few weeks between iterations, with all problems this causes (e.g. assignment of development staff etc.). Also, the overhead of preparing, negotiating and signing multiple contracts is just too big for small and fast paced projects.

We found that new customers typically prefer fixed price development contracts, whereas customers who already know us and our performance prefer time and materials development contracts, because it gives them more freedom in changing requirements. Ultimately it is a question of how much a customer trusts his supplier. Because iterative development requires frequent delivery of intermediate results, a supplier cannot hide the true state of a project for very long. In our experience it takes two to three iterations with good and on time results to win the trust of a new customer.

## 4. Experiences

Since the introduction of “agile RUP” at Zühlke in 1998 we have completed about 30 projects with this process. With one exception, all projects were successful<sup>2</sup>. Since most customers are not yet very familiar with agile development practices, it is essential that the responsibilities of the customer are clearly stated in the contract and equally clearly communicated to the customer. Normally it takes about two to three iterations for a customer who is new to agile development to get used to it. However, once they are familiar with it, they would never want to go back to traditional fixed price, fixed time and fixed functionality contracts with big bang delivery at the end of the project.

---

<sup>2</sup> The exception was a project where the customer had absolutely no time to participate in iteration planning and for providing feedback to intermediate releases.