



Advanced CASE Technology & Language Systems

ACTL Systems Ltd.
217/5 Jaffa St. P.O.B. 8129
91081 Jerusalem, ISRAEL
Tel. 972-2-5376459
Fax. 972-2-5370425



Architecting in Agile Projects

Abstract

This article outlines some ideas why and how to architect in Agile projects.

Agile teams often set the focus on sprint based delivery of working code and user stories. "Conventional teams" often precede development with an intensive architecting and design phase. Does architecting and designing belong to the past or can architecting and working Agile be combined?

Introduction

Many of our customers in the defense industry are used to waterfall driven process. This waterfall based process is often driven by contractual constraints and the perceived need of the system engineers to mitigate risks and manage large scale development through upfront design of the system architecture. But reality shows that the architecture and designs often change when implementation starts thus rendering the architecture and the design obsolete and not up-to-date. Because of these changes, documents such as SRS and SDD must be re-written at the end of the project to reflect the actual requirements and design. How can this work be prevented?

Defining Architecture and Design

Before moving on, let me define what I mean with architecture and design.

The architecture is the result of the design, meaning that design as a process, where as architecture is the design artifact. Since design is an ongoing activity, the architecture is constantly evolving. So in my view there is no such thing as a finished architecture as long as the system is not completed. Wait a minute! I just said that design is an ongoing activity? Why? The answer is simple. In Agile projects one is supposed to constantly refactor! Also TDD urges you to always have the best and nicest code that you can have at each point of time.

But what is architecture? Well, there are many definitions of what architecture is. For me, Architecture is a simplified description of the static as well as the dynamic aspects of the system. It serves for team communication, discussion and reference when extending the system. So if you want to talk about the architecture how does the architecture look like? Well, our answer is, that Architecture is a model expressed using UML/SysML.

But the model is only a means to communicate and not a goal or end in itself! Therefore the model must not describe the complete solution. It needs only to cover the aspects required for discussing the current large scale structure and mechanisms, for assessing the impact of current decisions and in and to assure that the various system parts are clearly separated, their interfaces well defined and that together the parts contribute to the current scope of system functionality.

I am putting emphasis on the word current because at each point of time the scope and focus is different. The more advanced you are in your project, the more issues and details come up.



Advanced CASE Technology & Language Systems



Agile Modelers architect, Agile developers produce code!

In our view Agile teams can either be agile developers or agile modelers. What is the difference?

Agile developers take user stories and develop them in sprints. The teams are supposed to refactor, but they often don't do it because of time pressure. This leads ultimately to spaghetti code/architecture and the velocity will eventually drop dramatically. (These are our best customers, we call them SCRUM victims :-))

In large projects, Agile modelers perform model based design prior to development in each sprint. The model is not only a sketch on a white board, since white boards are of transient nature. Taking picture to achieve "persistence" is not sufficient because they can not be manipulated. Therefore a model using a modeling tool is required. When you use a modeling tool, you always have an up-to-date "documentation", hence an architecture. So in each sprint you have an architecture description of the sprint scope. But the architecture should not only focus on the current sprint but also take into consideration stories that will most probably be tackled in the next 2 to 3 sprints.

I said "take into consideration" and not model! This is an important difference. Since taking into consideration means only to think about them and not actually find a solution for them!!!

The resulting model is then generated to code and the code is realized and tested prior to the presentation and retrospect.

So agile modeling teams are constantly designing resulting in an ever evolving architecture. The team acts as architect!!! There is no need for an architect role.

None the less we think that one team member should take the architect role. As we see it the architect's role is to cut discussions short and make decisions as well as to monitor to some extend the need for architecture change during development and scope extension in each sprint.

Dani Mannes

danim@actl.co.il

www.actl.co.il