Improving Solution Architecting Practices

Poort, E.R.

2012

Improving Solution Architecting Practices

Eltjo R. Poort
Ph.D. Thesis, 2012

Summary

As the presence of Information Technology increases, so grows the impact of the design decisions shaping the IT solutions that touch our lives. We feel this impact as we are amazed at the new possibilities offered by developments like the Internet, which all but redefined our social interaction experience within the time span of one generation. But the impact of IT design decisions is not always positive. We sometimes feel negative impact as small irritations, like our kids complaining whenever their favorite social media site modifies its functionality. Sometimes, however, things go really wrong, with far-reaching consequences.

Once in a while, a single wrong design decision makes its impact felt across an entire nation's political landscape, or even globally. In the past decade, the Netherlands alone has seen a number of such events, such as the troubled C2000 communication system for emergency workers, the fraud sensitive OV-chip public transportation payment card, and the Dutch highway tunnel safety systems that were fully opened to the public years after the original deadline due to software quality issues. In all three cases, is was not the functionality of the solutions that was wrong - it was the other, "non-functional" aspects. These aspects, like performance, security, safety and reliability are often called quality attributes. In the last decades it has become increasingly clearer that these quality attributes are mainly determined by the solution architecture, and hence Non-Functional Requirements (NFRs) should be driving solution architecture design.

A review in 2003 by Dalcher and Genus estimated the total financial cost of failed IT projects in the United States and the EU at 290 billion US dollars. More importantly, the above examples show a significant impact on our quality of life. Some are even life threatening. If we want to address these problems, we need a better understanding of the interaction between non-functional aspects and the architecture of IT solutions.

This thesis is the result of a journey to improve architecting practices in Logica, a large IT Services company. This journey started in 2003, when we identified a need to better understand the impact of Non-Functional Requirements on our solutions. The first part of the thesis reports research into this area.

The first research question is how to structure solutions to optimally address Non-Functional Requirements. This research has led to "Non-Functional Decomposition" (NFD), a new framework in which conflicting requirements can be used to optimize a solution's structure. NFD is a technique, based on the relationship between functional and non-functional requirements, that clarifies the mapping of requirements onto a solution architecture. Our new framework reveals rationale behind existing architectural patterns and tactics, and can be helpful in developing new patterns and tactics to deal with conflicting NFRs.

A common factor in many troubled IT projects is the traditional client/supplier situation, in which the solution requirements (the "what") are drawn up by a client, and the architectural design (the "how") by one or more suppliers. A key problem in situations like this is the quantification of Non-Functional Requirements: determining the numerical values that the solution's quality attributes should achieve. Our research shows that optimal quantification of NFRs is significantly impaired by tendering rules limiting the communication between client and supplier. Tendering rules may even force a client to select the supplier that has the *worst* understanding of the impact of Non-Functional Requirements. In economic terms, it makes sense to delay the quantification of NFRs until client and supplier have had sufficient time to assess and communicate the value and cost associated with these requirements. Our research points towards a number of potential solutions, one of which is better use of the Competitive Dialogue tendering model. To solve the key problem, however, requires a change in attitude: both parties need to have sufficient trust to share information regarding the impact of non-functional aspects on their side of the contract, and willingness to accept a fair share in the associated risk.

A survey among architects shows that, as long as the architect is aware of the importance of NFRs, they do not adversely affect project success, with one exception: highly business critical modifiability tends to be detrimental to project success, even when the architect is aware of it. IT projects where modifiability is relatively business critical perform significantly worse on average. Our conclusion is that modifiability deserves more attention than it is getting now, especially because in general it is quantified and verified considerably less than other NFRs. Architects should be careful when dealing with IT projects with a strong focus on modifiability. We advise to pay particular attention to aspects like managing customer expectations, because it seems that customer satisfaction especially is significantly lower on average in this type of IT projects.

Our journey to improve architecting strategies in Logica gained momentum and focus in 2006, when the company's Technical Board expressed the requirement for a standard approach towards architecting across the company. This requirement gave us a clear sense of direction, and the result was the establishment and implementation of a solution architecting approach: Risk- and Cost Driven Architecture (RCDA). The context feeding both this research and the RCDA approach was the function of "Technical Assurance", a technical conscience role fulfilled by the author from 2005 onwards. The extensive interaction we had over the years with hundreds of IT projects with various degrees of size and complexity, in multiple industry sectors, is one of the main data sources for the research presented here. Another important source is the international architecture community in the company. All of this has led to the second part of the thesis, which is about finding a good solution architecture approach, culminating in the presentation of RCDA.

The basis for RCDA lies in new insights into the nature of solution architecture. Over the years, we have started to view architecture as a risk- and cost management discipline. This view extends existing views of architecture as a higher level abstraction and as a set of design decisions. In comparison with these existing views, it helps architects better order their work, and it helps in better communicating about the architecture with stakeholders in business terms. RCDA is a collection of practices, embedded in a new framework to ease identification and integration of best fit practices in a particular

context. The practices were harvested from practitioners and literature, and enhanced by research. The results of a survey among architects trained in RCDA indicate that for the majority of trainees, the approach has significant positive impact on their solution architecting work. This is true for RCDA as a whole, for its principles, and for its individual practices. The positive effects appear to be much stronger if the architects are in a position of responsibility and authority, e.g. in a Lead Architect role.

Working towards the establishment of RCDA, we have performed separate research into two specific topics: the requirements an architecture process need to fulfill in order to comply with the CMMI process improvement standard, and the role of architecture knowledge sharing in IT projects. Our conclusions about CMMI are that the latest version 1.3 has significantly improved support for architecture, compared to version 1.1. CMMI can still be improved in the areas of architecture governance, facilitating the sales phase and learning from architectural choices. To research the role of architectureal knowledge sharing, we conducted another survey among architects. The survey shows that architects face many challenges sharing architectural knowledge in projects, especially in large projects. Most of the common challenges appear to be generally neutralized somehow, since they show no correlation with project success. The only challenges that *are* correlated with project success are the ones related to interpersonal relationships. We conclude that *dealing with emotions* is a crucial factor in how architectural knowledge sharing leads to successful projects.

There is one overarching conclusion to be drawn from all this. Solution architecture is a discipline in the Information *Technology* context. The key findings of our research, however, point to the importance of non-technical aspects such as trust, emotions, risks and costs, responsibility and authority. Hence, our final conclusion is that good solution architecting is not so much a technical problem, but rather a socioeconomic one.