Preserving and reusing architectural design decisions
van der Ven, Jan

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2019

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.
Chapter 9

Conclusions

“The limits of my language means the limits of my world.”
- Ludwig Wittgenstein

9.1 Research Questions

This thesis consists of work on architectural knowledge vaporisation. We have addressed three different themes around this topic: artifacts, process and reuse. We started the research with exploring scenarios that involved architectural design decisions. The first research question we addressed was:

*RQ1: What are the industry needs for managing architecture decisions?*

This research question was addressed in Chapter 2, where the industrial needs for architectural decisions were assessed. The data was extracted from interviews with 14 employees from four different companies. We created an overview of the relevant stakeholders and identified 27 distinct use cases where architectural knowledge was needed. Based on this work, we can conclude the following.

First, most of the industry needs for architectural decisions come from architects or architecture reviewers. Others do not seem to be very interested in or aware of architectural knowledge. Second, in 12 of the 27 identified use cases, the architectural decision was explicitly mentioned. Other use cases involved the consequences of the decisions (e.g. UC5. Check Correctness), or process related issues (e.g. UC4. Perform a review for a specific concern). Architectural decisions can be seen as the most important aspect of architectural knowledge.

The use cases were described as interactions on a fictive Knowledge Grid. In this thesis, we have shown that functionality of such a Knowledge Grid can be implemented in real tooling. One tool was developed as proof of concept to address the second research question:

*RQ2: How can tacit knowledge about architectural decisions be preserved for later use?*

In Chapter 3, we related the processes from rationale management to that of software architecture design, creating a connection of rationale to (software) artifacts. We discovered that tacit knowledge from rationale management was missing in most software architecture artifacts (the Choice + Rationale and the Alternatives). We introduced a model for architectural decisions that included this underrepresented tacit knowledge. This model connects the rationale of the decision
to software artifacts like architectural documentation or source code. We showed how to manage alternatives in the code itself to demonstrate the preservation of tacit knowledge in currently-used artifacts.

The developed model was used in the work of Chapter 4, where we showed that it is possible to assist architects and reviewers in preserving tacit knowledge on architectural decisions. The developed tool enabled architects and reviewers to enrich the documentation they were using with formal architectural knowledge, in this case architectural decisions. This work showed that reviewers get a better understanding of the architecture when using this tool. In the next research question, the focus was on the decision process:

RQ3: How does the architecture decision process influence the decision results?

With the introduction of the Triple-A model in Chapter 5, we were able to see characteristics from decisions in the decision process. We showed that it is possible to characterize teams by who makes the decision, how the decision is documented and what the periodicity of the decision is. In the industrial cases from this chapter, we discovered interesting relationships between the factors from the Triple-A framework and the success of the decision process. Concerning the documentation, it is important that documents are kept short and to the point (The illusion of documentation and complexity instead of simplicity). Concerning the decision-maker, hands-on experience is essential to make up-to-date decisions, while the architect should never be the single point of failure. Last, architecting should be a continuous process to cope with changing needs.

In the work presented in Chapter 6, we conducted a survey to extract the most eminent factors that influence the architecture decision process. We challenged some beliefs from the architecture as well as the agile community. On the time axis, decisions do not become better (in terms of RoI or development speed) when you reserve more time to make them, and this factor does not influence the quality of the product significantly. On the who-axis, we busted the belief that ivory-tower architects create worse architectures. There was no indication that the role of the person was relevant for the result of the project. The only relevant indicator was the amount of development and architecting experience; more experience created higher quality products. Concerning documentation, using less documentation speeds up the development process, while the quality of the process does not seem to be affected.

In Chapter 7, the success factors of the lean startup movement have been studied to see what can be learned for the architectural decision process. This research is based on interviews with architects and founders at startups, to explore how they made decisions. We showed that the decision processes of business and software architecture are aligning. There is a trend towards making decisions as short-running experiments. As unknowns increase, it is necessary to conduct experiments on your assumptions as quickly as possible. There is no strict line between business and architecture in this. The most important aspect that can help making better decisions is the availability of data. This is exactly the subject of the last research question.

RQ4: How can architecture decision makers reuse decision data?
The need for data to base decisions on is explored in Chapter 8. For this work, we have focused on a specific architectural decision: the decision to use or change a component in a software system. We showed that it is possible to mine data on these decisions from the history of open source Ruby projects. In addition, we have closed the gap to the tacit knowledge behind the decisions. We have seen that decision-makers can access this knowledge by contacting the authors of the commits. The authors responded with help for the decision-makers. The data on the occurrence of decisions in the past can be used to reuse decisions, which enables a faster decision process.

9.2 Contributions

In this thesis, we have shown that knowledge on architectural decisions can be found and used in all forms: tacit, documented and formalized. We described how to model tacit knowledge and add it to architectural artifacts. In addition, we have shown that formal data on architectural decisions can be mined from open source projects, and that is is possible to access the underlying tacit knowledge by contacting the decision-makers.

Table 9.1 summarizes the contribution of this thesis per chapter. In the first column, the chapter number is given, the second column describes the main problem addressed. As described in the introduction of this thesis, the third column describes what IT artifacts [86] were used to address the problem, and the fourth column describes the type of artifact. The last column describes the evaluation of the artifacts.

As contribution of this thesis to the research community, first we presented a metamodel for architectural decisions, which can be used when formalizing architectural decisions in tooling. We show the feasibility of the metamodel by describing how we based our tools in it. With these tools, it is possible to experiment with design decisions; they can be preserved during the documentation process or recovered later on based on version history. We showed that the developed tools that were based on this metamodel can provide data for reusing decisions.

As a second contribution, we showed that there are no easy parameters to predict the success of architectural decisions. We showed that many beliefs on making architectural decisions do not hold. The only relevant indicator that predicts slightly better decisions is the experience of the decision-makers. We showed the similarities between architecture design and new product development, where decisions are based on data that is acquired from short-running experiments. The contribution of this thesis is that the experiment-based decision process can be adopter in architecture design to make better decisions, or fail more rapidly.

In addition, we have introduced a way of conducting experimental research based on practices from new product development; an A/B experiment. We have used this method to see if there were differences in responses to our emails, when we phrased the email differently. This method is widely used in industry to measure the effectiveness of software, but to the knowledge of the author of this thesis, this methodology is not used in software architecture research yet. Therefore, an additional contribution of this thesis is that A/B testing can be used to generate data for research.
Table 9.1: Contributions per chapter

<table>
<thead>
<tr>
<th>Ch</th>
<th>Problem</th>
<th>IT Artifact</th>
<th>Type of Artifact</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>How are architectural decisions used?</td>
<td>A Use Case model</td>
<td>Construct</td>
<td>Industrial assessment</td>
</tr>
<tr>
<td>3</td>
<td>How can architectural decisions be better preserved and understood?</td>
<td>A model for Design Decisions and rationale</td>
<td>Model and Instantiation</td>
<td>Example implementation</td>
</tr>
<tr>
<td>4</td>
<td>How can architectural decisions be integrated with the documentation and implementation?</td>
<td>A tool that assist in managing architectural decisions</td>
<td>Method and Instantiation</td>
<td>Quasi-controlled experiment</td>
</tr>
<tr>
<td>5</td>
<td>How to predict the efficiency of an architectural decision?</td>
<td>A framework for classifying architectural decisions</td>
<td>Model</td>
<td>Industrial cases</td>
</tr>
<tr>
<td>6</td>
<td>Can we make better decisions by changing the decision process?</td>
<td>guidelines for decision making</td>
<td>Method</td>
<td>Survey</td>
</tr>
<tr>
<td>7</td>
<td>How can we make architectural decisions faster?</td>
<td>Guideline for the decision process</td>
<td>Method</td>
<td>Interviews</td>
</tr>
<tr>
<td>8</td>
<td>Can we reuse architectural decisions?</td>
<td>Implementation for mining and exploring architectural decisions</td>
<td>Instantiation</td>
<td>Expert review and email experiment</td>
</tr>
</tbody>
</table>
9.3 Future Work

In this thesis, we have seen a shift from opinion, rationale based decision-making to data-driven experimental decision-making. The term VUCA (Volatile, Uncertain, Complex en Ambiguous)\(^1\) is often used for the current fast-changing world. In this world, there are many unknowns and assumptions and ideas need to be validated very rapidly, based on data. We believe the work in this thesis is the first step in this direction for software architecture, where we showed how decision support can be constructed based on available data. As the amount of available data is increasing, we expect that assistance for decisions based on this kind of data will increase, in research as well as in industry. Specifically, Chapter 8 focused on component selection decisions. It would be interesting future work to see if other types of architectural decisions can be mined from available data. In addition, extending this current work to other programming languages would be a logical choice.

A challenging tension exists after this thesis concerning software architecture research. On the one hand, keeping and maintaining architectural decisions is important for project success (especially in the long run). On the other hand, we see that the adaptation of tools that manage this is very low in industry. Main reasons given are the effort needed to use the tooling [38]. In addition, speed at which decisions must be made increases. So, the challenge remains to create non-intrusive tooling that extracts relevant architectural knowledge fast to address knowledge vaporisation.

---

\(^1\)https://en.wikipedia.org/wiki/Volatility,_uncertainty,_complexity_and_ambiguity