From software design patterns to technology transfer patterns

This article reflects the status at the time of its publication. The content may since have become outdated.

24/07/2019 In the past two to three years, I noticed the term “innovation management” rising up. As innovation is a crucial success factor especially for large companies worldwide and “management” means the direction, organization, and planning of all kinds of fields, the term implies that there indeed is a well-structured way to success.

But is there really? Practice teaches us the opposite almost every day, though. Especially in our work at the Porsche Digital Lab, where we explore new technologies and experiments to bring big data, AI and IoT into practice at Porsche — trial and error is an integral part of our daily work. As my colleague, Alissa Wilms said recently: We try out new ideas fastly, discard them if necessary and start over quickly.

How structure helps in situations of uncertainty

Taking a closer look at the field of software design, I realized that it is a complex undertaking, where strict processes have shown to fail in situations of uncertainty. Instead, it is common knowledge that the software engineering community documents well-proven solution concepts in the form of software design patterns. So why not apply this successful concept to other areas?

Based on the observations in my last article, I have started to work on applying the very same idea of documenting the body of knowledge that seems to exist on technology transfer. What I'm trying to do is define a simple but useful structure that can be used to solve common challenges of technology transfer in practice.

The structure of patterns in software design

Software design patterns are a "general, reusable solution to a commonly occurring problem within a given context in software design". Basically, they are instructions to get it right.
Amongst others, such a pattern comprises the following parts:

- **Pattern Name and Classification**
  A descriptive and unique name that helps in identifying and referring to the pattern.
- **Intent**
  A description of the goal behind the pattern and the reason for using it.
- **Motivation (Forces)**
  A scenario consisting of a problem and a context in which this pattern can be used.
- **Applicability**
  Situations in which this pattern is usable; the context for the pattern.
- **Consequences**
  A description of the results, side effects, and trade-offs caused by using the pattern.
- **Implementation**
  A description of an implementation of the pattern; the solution part of the pattern.
- **Known Uses**
  Examples of real usages of the pattern.

Pattern Name and Classification
A descriptive and unique name that helps in identifying and referring to the pattern.

Intent
A description of the goal behind the pattern and the reason for using it.

Motivation (Forces)
A scenario consisting of a problem and a context in which this pattern can be used.

Applicability
Situations in which this pattern is usable; the context for the pattern.

Consequences
A description of the results, side effects, and trade-offs caused by using the pattern.

Implementation
A description of an implementation of the pattern; the solution part of the pattern.

Known Uses
Examples of real usages of the pattern.

Applying software design patterns to technology transfer

For the first of 5 lessons I've learned about technology transfer that I gathered in my last article, an applied pattern could look like this:

- **Pattern Name**
  Thinking ideas through across different domains.
- **Intent**
  Addresses the challenges associated with too narrow views, namely: Lack of understanding of potentials, misunderstandings, the slowdown of adoption.
- **Motivation (Forces)**
  Experts are strong in their specific fields of knowledge. With this in mind, great value is often created by combining expertise from different domains and across organizational units. Software engineers, for example, are highly trained specifically in the fields of abstraction, system decomposition and problem-solving in order to find adequate solutions quickly. However, on the flipside, complimentary business aspects such as time-to-market, advertising, sales, and overall business value are often neglected.

Pattern Name
Thinking ideas through across different domains.

Intent
Addresses the challenges associated with too narrow views, namely: Lack of understanding of potentials, misunderstandings, the slowdown of adoption.

Motivation (Forces)
Experts are strong in their specific fields of knowledge. With this in mind, great value is often created by combining expertise from different domains and across organizational units. Software engineers, for example, are highly trained specifically in the fields of abstraction, system decomposition and problem-

solving in order to find adequate solutions quickly. However, on the flipside, complimentary business aspects such as time-to-market, advertising, sales, and overall business value are often neglected.

Indeed, an understanding of business and technology are equally important. There is an interplay between technology and business decisions, and it is paramount to understand the interdependence of the two spheres. Yet, the required skill set is hard to teach. Moreover, broad topics like artificial intelligence should be understood at the manager-level, both in terms of knowledge of application fields, and the ability to convey knowledge about complex subjects.

- **Applicability**
  This pattern shall be applied in scenarios in which complex technologies shall be leveraged, e.g., in digital products.
- **Implementation**
  The pattern can be implemented in different ways, e.g.: Individual development of employees, Job rotation throughout different business fields, Diverse teams.
- **Known Uses**
  Virtual Product Teams, Competence Centers.

I guess we can agree that challenges in innovation management and especially technology transfer are complex and not easy to tackle. Still, we can also observe that when specific challenges are brought up, often others from different units, organizations and companies can directly relate to them as they had similar experiences. At least I found this to be true in countless discussions with different CIOs and CEOs over the past years in my roles as a management consultant and director of the Porsche Digital Lab Berlin.

This little insight makes me believe that a community approach to the elicitation, discussion, and maintenance of technology transfer patterns would be highly beneficial; certainly for those feeling lost in the (digital) transformation. If we take this catalog of challenges and solutions as an international approach, it would be enriched with diverse perspectives — bringing us closer to connecting leaders and managers all over the world with a common pool of possible solutions for similar problems. Wouldn't this be great?

I am happy to hear your thoughts on this. Let's get in touch!

MEDIA ENQUIRIES

Dr. Mahdi Manesh

Director Porsche Digital Lab

Link Collection

Link to this article
https://newsroom.porsche.com/en/2019/digital/porsche-digital-lab-software-design-patterns-technology-transfer-patterns-18195.html

Media Package
https://pmdb.porsche.de/newsroomzips/ab796041-5aba-46e5-9134-a800c1bdb9a3.zip

External Links
https://medium.com/next-level-german-engineering

Dr. Ing. h.c. F. Porsche Aktiengesellschaft
Porscheplatz 1
70435 Stuttgart