The Internet of Things Prototyping Platform Under the Design Thinking Methodology

Prof. Victor Taratukhin, SAP America

Victor Taratukhin received his Ph.D. in Engineering Design in 1998 and Ph.D. in Computing Sciences and Engineering in 2002. Victor was a Lecturer in Decision Engineering and Module Leader (IT for Product Realization) at Cranfield University, UK (2001-2004), SAP University Alliances Program Director (2004-2012). He is Next-Gen Network Global Projects and Regional Director for Silicon Valley and US West at SAP America, Inc., Managing Director, Competence Center ERP at European Research Center for Information Systems (ERCIS), University of Muenster, Germany (2012-present) and was Visiting Professor at Stanford Center for Design Research (CDR) (2015-2016)

Teaching/Industrial Supervisor at Stanford (2015-2016)

ME 310f: The Essential Elements of New Product Development: Business and Industry Perspectives

Ms. Yulia Yadgarova, Georgia Institute of Technology

Yulia is visiting researcher in Georgia Institute of Technology, also a PhD candidate in the Department of Computer automation systems at Bauman Moscow Technical University. Consultant at SAP CIS Moscow. Research interests include Industrial Internet of Things and Multi-Agent systems.

Prof. Joerg Becker, University of Muenster, ERCIS

Jörg got his PhD and his habilitation at the University of the Saarland, Saarbruecken, Germany. Since 1990 he has been full professor for Information Systems and Information Management at the University of Muenster, Germany. His research fields are Process Management and Process Modelling, especially Reference Models, Data Management, especially Models for Management Information Systems, Retail Information Systems, and eGovernmant. From 2008 to 2016, he served as Prorector for Strategic Planning and Quality Assurance and CIO at the University of Muenster. Since 2004 he has been Managing Director of the European Reserch Center for Information Systems ERCIS, a network of 25 research institutions in the filed of Information systems.
The Internet of Things prototyping platform under the Design Thinking methodology.

Yulia Yadgarova, Research Scholar, Georgia Institute of Technology, yyadgarova3@gatech.edu
Dr. Dr. Victor Taratukhin, Professor University of Münster, victor.taratukhin@sap.com
Dr. Dr. Jörg Becker, Professor, University of Münster, becker@ercis.uni-muenster.de

Abstract.

The paper describes the current gap in the Design Thinking [1] process with respect to creating the complex Internet of Things prototypes.

The study lasted the several years have proven the usability of Design Thinking process while developing user-centered and complex IT applications [2]. Although, for the Prototyping phase of the process was crucial for providing a solution. After completing several studies on development teams and analyzing prototypes we unleash that when teams worked with serious and complex technologies can struggle with providing high-quality and industry-approved solutions. As a result, this followed by lack of trust from industry vendors and frustration from team members.

Providing the Internet of Things solutions from one side means establishing user-centric approach and interaction between users and strict software architecture from the other side [3]. However, the Design Thinking Prototype phase does not cover the complying with quality requirements of the software products. This lead to the significant gap between developed prototype and proposed industry solution which can cause impediments in the implementation of the project.

On the other side, the Internet of Think field has been developed more than 10 years and several research studies have identified the critical points and requirements for the working projects [4]. These requirements were followed by the architecture patterns and tactics each of them is intended to improve the quality attributes of the project [5]. As a result, the sustainable IoT prototype should meet these attributes.

As a solution, the visual functional and software framework for IoT Prototyping phase was proposed. The framework undertakes the problem of lack of knowledge in design team and aims to improve quality of the prototypes. Evaluation of the achievable attributes with respect to current research works is mentioned. The testing and implementation of the reference framework currently are in testing process by the DT teams.

Introduction.

The Design Thinking methodology introduced in the 2005 in Hasso-Plattner Institute of Design at Stanford University in California. The initial goal of the introduction of the DT (Design Thinking) was to help engineering and computer science students to be innovative and find insights in terms of developing new projects and solutions. The structure of the design team and rules and stages of the process actually was developed the years ago and still under investigation on the School of Design Thinking at Stanford. As the course is strongly recommended for the engineering curriculum, since 2005 a lot of research information have been collected and analyzed. As brief explanation of the DT process, we can outline four main takeaways those make DT special and different from other IT methodologies to organize IT projects:
- **Diversity of the team.**

The diversity of the team implies that within the project space the background and experience of the team members should vary significantly to bring the collaboration and increase the possible solution space of the outcome.

- **The problem space exploring.**

One of the main features of Design Thinking process is the strong exploration of the problem space which brings the deep dive into the user’s problem definition and helps the project team to understand the scenario. The core outcome and the core strength of this process is fully intuitive way to explore the problem through the visualizing the use and seeing the problem from the customer/user point of view.

- **The solution space exploring.**

The solution space exploring includes the wide investigation of possible problem-solving scenarios which are directly connected with the problem space exploring and may also influence on the initial problem understanding. This is strongly related to the prototyping phase which means the physical representation of the ideas and attempts to make initial version of the solution.

- **The iterative process between process and solution spaces.**

The revising and continuous iteration between both spaces helps to bring new insights and at the end improve the understanding of the problem and quality of the prototype in terms of functionality. The iterations between solution and problem definition is necessary to deep investigation and resemble the process of creating new ideas in the Design Theory approach [2]

All of those four features of the DT methodology are initial formalization of the process of insight creation. The system approach for the innovation process is unusual in terms of the such collaboration prospective and not only relevant to the technical insights. Actually, the point of diversity commands creation eliminates the predominant technical view [1] which is based mostly on the technical experience and technology capabilities. This is one of the main problems in the engineering/IT domain.

On the other side, to bring new feasible ideas, it is better to take a look at the possible technical problems and explorations which can be crucial in terms of successful reveal the innovation.

**Design Thinking technology and steps.**

The kernel of the Design Thinking methodology is three-activity process [1] and includes phases of innovation, execute and synthesize:

![Fig. 1 - Unified Innovation Process Model for Engineering Designers and Managers](image-url)
From the system analyses point of view the Design Thinking approach can be described with the Functional Diagram (IDEF0) as presented below (Fig. 2)

![Diagram of Design Thinking methodology (functional approach)](image)

Fig. 2 – Design Thinking methodology (functional approach)

**The technological gap in DT projects.**

In the current research project, we mainly focused on the process of the Prototype creation and the feasibility of the obtained solutions in the world. Exploring the wide variety of the creative design concepts and investigation of different prototyping strategies [6], we came to the conclusion that at the prototyping phase the most valuable is to provide a lot of iterations rather than focusing on the one approach. This means, that during the realization it could be less possible to developers and project team to provide feasible and sustainable architecture within the strict time constraints. This probably leads to the less valuable insight process and reduce the number of possible success solutions. At the meanwhile, focusing only on the possible solution space with not paying attention on the realization will bring a problem of drive the potential idea to the market due to the technical incompleteness and defects.

This is the most valuable gap we have found during the process of investigation of the design teams work during the several workshops held in lead US and Russian universities. The goal of introducing the possible additional step in the process is to eliminate this technical gap while keeping the freedom of the iteration process and reducing the difficulties in the technical implementation of the projects. The big and general restriction of our research is avoiding the boundaries in the Design Thinking process itself and not to try restricting the process of prototyping by exploring the technical limitations.

The place of our approach is at the end of the Design Thinking steps and it’s appropriate if the Design Team come to the conclusion of the possibility of the IoT architecture. The IoT field was chosen due to the complexity and encapsulating both operation technology (OT, engineering) and information technology (IT) which is one of the most difficult combinations for the technical architectures. To explain the possible architectures and our approach below
we have highlighted the types and features of IoT projects also with IoT architectures which could be chosen. Based on the classification and approach we have developed a methodology of the choosing the most appropriate IoT architecture and included this methodology as one of the steps of the DT process (Fig. 2).

**Internet of Things technology classification and architectures.**

The Internet of Things as a fact was firstly introduced in 1999 by Kevin Ashton, the founder of the MIT Auto-ID center [7]. The last years technology becomes more popular in the wide variety of fields among science and technology. Although the term IoT (Internet of Things) itself does not bring the strong understanding of the technology. The literature review highlights two main sides of the IoT as a technology. The first – IoT is a network, second – IoT is a network of Things which implies any physical objects of our world. Those objects are the part of the technology, connected via network and have unique identifiers. In each moment it’s possible to read the state of the object and sometimes control this object through the network. The interesting thing is that those objects can operate without human which means the autonomous behavior of the world. The cooperation between objects brings synergy and probably is one of the main advantages of the technology.

The IoT projects became most popular in 2010-2017 with appearance of wide-variety of end-devices and development of the data transfer technologies and protocols. This was followed by appearance of the hardware- and software- platforms for developing and prototyping of such solutions (Arduino starter kits, Raspberry, Smart Home systems...). Meanwhile among the simple solutions for the quick development the complex systems and tools start their adoption in the market – the systems that support Industry 4.0, Smart health and Smart City technologies. That was followed by creation of various classifications of the IoT solutions. One of the most popular classifications is Postscapes [8] which divides the whole range of IoT solutions into several categories according to the functionality (“Smart home”, “Smart transportation”, “Smart sales”, “Smart home”). The other huge classification of the IoT solutions which is overlap the previous and which is more relevant for our research, the systems are divided into “Horizontal” and “Vertical” solutions (Fig. 3)
Based on the classification above the architectures of the IoT solutions may vary significantly, and the choice of the right architecture is one of the most difficult challenges during the implementation of the IoT project. At the same time, the right choice of the architecture in the beginning of the software development and prototyping process can influence on the successful implementation of the product due to significant simplifying of the development process. From the architectural point of view, several research projects have derived seven major quality features of the IoT architecture which should be achieved for the successful implementation of the project [9]. Those quality attributes include:

- Interoperability
- Scalability
- Availability
- Information security
- Maintainability
- Efficiency
- Privacy

As an example of the software architecture styles which can be used for the architectural choice we have chosen the most documented in the literature [10, 11]: client-server, publish-subscribe, service bus, SOA, REST, microkernel, layered architecture.

Based on this variety of the architectures and the classification we have developed the classification which we use during the Design Thinking process to identify the right software architecture in the initial phases of the project. As the basis of classification, the following features were chosen:

1. The number of the devices in the potential system;
2. The frequency of adding new devices in the system;
3. The presence of several software/hardware vendors in the system;
4. The geographical distribution of the devices;
5. General functional field (Monitoring/Control);
6. Number of users in the system and number of devices per user;
7. Are the devices stationary/mobile;
8. Real-time requirement;
9. The amount of analytical calculations in the system.

Such features are valuable for the initial project evaluation due to their correlation with the quality attributes listed above. For example, the quantity of the devices influences the scalability of the system while presence of several software vendors directly increases the required interoperability of the system.

Based on the developed classification and the Design Thinking steps the methodology for defining the appropriate software architecture for the IoT projects on the Prototyping phase was developed. This methodology modifies the standard DT Prototyping step by adding several additional action items into the process of prototype creation. Meanwhile, to support such methodology the Questionnaire was developed and introduced on one of the DT steps.

The model for software architecture styles evaluation

To briefly outline the underlying model of our methodology, let’s dip dive into the definition of software architecture style. According to [13], software architectural styles define “a family of such systems in terms of a pattern of structural organization. More specifically, an architectural style determines the vocabulary of components and connectors that can be used in instances of that style, together with a set of constraints on how they can be combined. These can include topological constraints on architectural descriptions (e.g., no cycles). Other constraints—say, having to do with execution semantics—might also be part of the style definition.”
As a model for the architecture styles evaluation we use an approach based on the correlation between the architecture elements and possible obtained quality attributes listed above. For each software architecture style [10] we have defined the elements, constraints, and rules which are the inputs of the evaluation model (Fig. 4).

Fig. 4 – The model of software architecture style evaluation

For each possible architecture style given the available elements and external requirements we are able to build the system of equations which evaluates the quality attributes of the IoT solution:

\[
\begin{align*}
Q_1 &= F(E_{obj}^1 + E_{const}^1 + E_{rules}^1 + E_{design}^1) + \sum_{i=1}^{m} R_i \\
\vdots \\
Q_n &= F(E_{obj}^n + E_{const}^n + E_{rules}^n + E_{design}^n) + \sum_{i=1}^{m} R_i,
\end{align*}
\]

where \(Q_1 \ldots Q_n\) - quality attributes listed above, \(E_{obj}^k\) - Objects in terms of software architecture style, \(E_{const}^k\) - constraints, \(E_{rules}^k\) - semantic rules, \(E_{design}^k\) - system design requirements of the style, \(R_i\) - one of the external project requirements.

The final decision of picking the appropriate style depends on the quality evaluation and type of the IoT solution. Evaluation of the architecture style is the linear regression with parameters strictly related to the correlation between the quality attribute and particular style. The parameters of the regression are:

\[
S_{tk} = \alpha_{0k} + \alpha_{1k} \cdot QA_{1k} + \alpha_{2k} \cdot QA_{2k} + \ldots + \alpha_{zk} \cdot QA_{zk},
\]

where \(\alpha_{tk}\) - coefficients that show the importance of the particular attribute for the type of the architecture, \(QA_{1k}\) - quality attributes of the software, listed above.
**Modified Prototyping phase.**

As the modification of the existing technology, we have identified and clarified the steps of the DT in terms of functional modelling. The initial model of the DT steps in terms of the IDEF0 modelling was presented on the Fig. 2 and identifies the main steps and inputs/outputs of the DT process as a system approach. After decomposition of the Step 4 (Prototyping phase of the DT) the various research studies have identified the four main steps of the Prototyping phase: Envisioning possibilities, creating a prototype to embody a possibility, Getting feedback about the prototype, and Reevaluating constraints [12]. However, the proposed approach with respect to IoT project specificity brings a slightly different approach to developing a prototype of the solution, which is identified on the Fig. 5.

The process of Prototyping starts from the identifying the type of the proposed solution. Among the set of new ideas and solutions one should choose the approximate type of the proposed prototype. In case of identifying IoT type, the Project team can use the modified methodology with steps outlined on the Fig. 5. The functional prototype creation phase means that the process of prototyping is divided onto two different steps: functional development (collecting requirements and identifying the set of functions and features) and technical project architecture definition.

After creation of the functional prototype and identifying the possible type of the solution with the classification Questionnaire, according to the answers given from the Project team we can identify the possible type of IoT architecture. The correspondence between type of the IoT solution and the possible architecture style quality attributes is defined through the Type Matrix listed in the Table 1.

![Fig. 5 – Prototyping phase of the project (developed methodology)](image-url)

As a result of the modified Prototyping phase the feasible software architecture with the optimal set of quality attributes will be obtained. This result can be used as an initial phase of the Agile development process following the Design Thinking phase, which is valuable outcome of the DT process.
The learning outcomes and results of the modified methodology.

During the process of implementation of the modified methodology we believe that the new project results will help students to learn specific technical domain in detail and help them obtain the engineering and IT-insights.

From the educational prospective, it was very important for students to create not only the prototype as an abstract part of the process, but feasible draft of the project that can be used in the further project work and development. While developing the modified methodology there were three main challenges we covered.

The first challenge was to modify the process of DT and bring some technical feasibility into the very vague process of developing insights. This is quite challenging task, but the developed methodology do not influence on the process of creating new ideas, so we can argue that some automation in the Design Thinking process can be valuable for the sake of feasibility of the prototypes.

The second challenge was to identify and develop a process of the IoT quality evaluation and identification of the possible architecture based on the required quality attributes. The IoT domain in contradiction with the general Software Quality domain is pretty new, so identifying and distinguishing the quality attributes with the existing software architectures is a difficult task. Finding the ques of this process can potentially lead to the automatic generation of the initial software architecture and minimisation of the programmer’s mistakes during the implementation of the projects.

Although these problems were new, given with such approach, the potential students will be aware of possible variants of the systems architecture, “bottlenecks” and limitations of the IoT technology and will be able to generate, develop and maintain initial solution in the future project work.

Conclusion

In the presented paper we have introduced the modified Design Thinking methodology for the engineering/computer science students which is improving a quality and feasibility of the developed prototypes. The presented methodology works mainly with IoT prototyping projects, but it is possible to extend such kind of methodology on the other types of IT/engineering projects.

We hope that improving the quality of the student prototypes as well as the DT prototypes in general is a step forward to quick and quality approach to develop the IT projects and reducing the cost and time for development.

<table>
<thead>
<tr>
<th>Type of the IoT/Quality attribute</th>
<th>Connected vehicles</th>
<th>Logistics</th>
<th>Predictive maintenance</th>
<th>I4.0</th>
<th>Smart City</th>
<th>Home Automation</th>
<th>Smart Health</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency of add</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Single vendor/Multiple vendors</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Static/Mobile</td>
<td>Mobile</td>
<td>Mobile</td>
<td>~</td>
<td>Static</td>
<td>~</td>
<td>Struct Static</td>
<td>Static</td>
</tr>
<tr>
<td>Geographically distributed</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Monitor/control</td>
<td>Monitor</td>
<td>Monitor</td>
<td>Monitor Control ~</td>
<td>~</td>
<td>~</td>
<td>Monitor</td>
<td>Low</td>
</tr>
<tr>
<td>Amount of users</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>High</td>
<td>~</td>
<td>No</td>
<td>Low</td>
</tr>
<tr>
<td>Real-time requirements</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>~</td>
<td>~</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Analytic requirements</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
<td>~</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>
Bibliography:

6. Steven P. Dow and Scott R. Klemmer The Efficacy of Prototyping Under Time Constraints