2015 15th International Conference on Computational Science and Its Applications

Trends in the use of Design Thinking for Embedded Systems

Rodolfo Araújo
Project Manager
Phoebus Technology
Joao Pessoa, Paraiba, Brazil
rodolfo.cunha@phoebus.com.br

Eudisley Anjos
DSC - Federal University of Paraiba
and CISUC - University of Coimbra
Joao Pessoa, Paraiba, Brazil
eudisley@ci.ufpb.br

Danielle Rousy Silva
DSC - Center for Informatics,
Federal University of Paraiba
Joao Pessoa, Paraiba, Brazil
Danielle@ci.ufpb.br

***Abstract* — The increasing complexity of embedded systems turned the requirements management in an even more dificulty task. In addition, there is a demand for innovative, customized and trendy systems. Design Thinking can minimize these problems throughout the development of Embedded Systems. The use of Design Thinking aims to understand the real users needs and generate innovative solutions for new systems. It can improve the requirement elicitation process and increase the quality of the final system. This paper approaches new trends in the use of design thinking during the whole development process of innovative embedded technologies.**

Keywords—Embedded Systems; Design Thinking; System Requirement; Software Engineering.

I. INTRODUCTION

Embedded Systems are constantly present in our life. Since a simple conversation using smartphones to our big trips in aircrafts, we can find embedded systems. The use of these technologies are still increasing and being relevant for several different areas. According to a research guided by BBC [2], embedded systems market is expected to have an annual growth rate of 5.4% over the next five years, going from 152,4 billion in 2014 to 198,5 billion dollars in 2019. Even though this market growth is known, the development of embedded systems is complex and requires innovative features and high quality. To provide high complex systems, the use of requirements engineering methods is crucial to preserve the budget and achieve the expected quality [1].

However, this growth also brings to market a higher competitive and demanding for quality solutions. Thus, Embedded Systems can benefits from Requirements Engineering, since it is recognized as crucial for development of softwares with quality [7]. It has many methodologies and techniques to support system development that focus basically on the customer real needs. But, in Embedded Systems context, specifically, where the products have a close association with innovation [9], the Requirements Engineering process could be improved by adopting Design Thinking [3].

Design Thinking is a human-centered process that aims the creation of innovative solutions. It covers the elicitation of client real needs instead of just system requirements. Therefore, it provides a way to produce in a short period of time, rapid and simple prototypes which eventually evolve to innovative products.

Innovation is recognized as a part of embedded systems [9] and there is a continuous demand for it. More, there is a constant requirement for agility, quality and low costs, despite of the complexity involved throughout the development. In this case, the adoption of Design Thinking is appropriate since it helps to solve the difficulties found during the embedded systems development [3].

This paper approaches the use of Design Thinking during conception and development of innovative embedded technologies. The main objective is to identify potential trends and benefits in the use of DT and how it could be advantageous to embedded systems. For this, this research performs an analysis of the requirements elicitation techniques for embedded systems and how DT may be used in this field.

II. EMBEDDED SYSTEMS

The term "Embedded Systems" is hard to be defined. Since it refers to a segment of computer systems that are constantly evolving. However, frequently the Embedded System definition is associated to a computer system composed by both hardware and software, built to execute a specific function [16].

There are some characteristics that differs Embedded Systems to others computer systems: flexibility and complexity. The first refers to the large number of situations in which Embedded Systems are used and we can find them, by instance, in: coffee machines, TV's, microwaves and mobile phones. Actually, it is possible to find hundreds of embedded devices at our homes [11]. But, Embedded systems are also presented in bigger and more complex products, like vehicles, aircrafts, rockets and medical systems. These systems are highly complex, since they have a high number of features that must be executed at the same time. Besides it is critical, because a failure can cause disaster results [16]. For example, in a vehicle automatic parking function the system needs to manage the steering wheel, accelerator pedal, the breakers e the sensors. Also, the Embedded Systems are crucial part in what is being considered the next technological revolution: the Internet of Things [5]. The term "Internet of Things" describes the

978-1-4673-7367-8/15 $31.00 © 2015 IEEE
DOI 10.1109/ICCSA.2015.25

possibility of control and communication between physical objects by Internet. Thus, all these things are connected and might change information by themselves.

According to a report released by IC Insights [18], that presents a projection of Internet of things scenario, besides the growth of the market be an important factor; the expansion on diversity of systems utilization has to be taken into consideration. The data presented in Figure 1 shows that applications of embedded systems, as Industrial Internet and Connected Cities, will lose room for more popular segments as Wearable Systems, Connected Vehicles Systems and Connected Homes. Thus, embedded systems will be even more present in people everyday life.

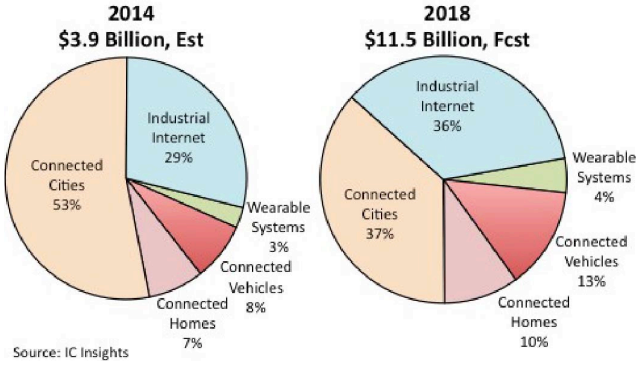

Figure 1. IoT Semiconductor Sales by System Segment.

However, the expansion of embedded systems segment brings new challenges that need to be overcome. For example, the increase of connectivity trough the Internet opens up security issues, where new threats might emerge [10]. Therefore, this is one of the mainly remaining challenges, since the classic approaches of security are not, in most of cases, appropriate to embedded systems [5].

Another relevant problem about embedded systems is their complexity. With a growing demand of system that aggregate several functionalities in a single product, and the need of execute them simultaneously, make this kind of computational system highly complex. There are still others opened challenges as Nano-scale Technology, Device Interoperability or Power consuming [17].

Hence, the effective use of Requirements Engineering techniques is an important step to solve the problems mentioned and to obtain high quality embedded systems designs [4]. This shows the potential that exists on investing in a more optimized software and hardware project for embedded systems.

III. Requirements Engineering on Embedded Systems

Requirements engineering is a branch of Software Engineering responsible for managing the system requirements. A Requirement Engineering process well executed is recognized as a key factor to quality software development inside the budget and schedule [7]. Although being one of the most important areas of Software Engineering [1], there are a few researches on Requirements Engineering for Embedded Systems.

Real time systems with a high number of functionalities are inherently complex. Adding this to constraints like hardware size, power consumption and processing power, for instance, makes these systems even more difficulty to develop. Due these facts, the adoption of Requirements Engineering is crucial for the success of embedded systems [4].

The first fact is that Requirement Engineering is not only about writing requirements, but also, discovering and understanding the real problem and the real needs of a client [15]. This is important, cause it avoids the development of low quality software that does not solve the real problem. On high complex software this is even more crucial, because of the high number of functionalities. Hence, requirement engineering methods are important in these cases, like the creative technique of "five whys". It is a technique where you ask "why" five times until you found the real cause for a specific problem.

A second aspect is that requirements engineering provides many approaches in order to identify and manage software constraints or non-functional requirements. This covers a recurrent challenge faced by embedded systems: the constraints [12]. Although the advanced researches on this topic, there are still some common constraints as small hardware size, power management or limited memory.

Nevertheless, traditional requirements engineering still lacks support for creativity and innovation in development process. Innovation is an important characteristic of embedded systems and should be a main concern during the conception of these systems. Thus, Design Thinking is an effective alternative to the traditional requirements engineering techniques and can benefits embedded systems quality and innovation.

IV. Design Thinking on Embedded Systems

Design Thinking is a creativity technique that aims to generate innovative solutions. The starting point of Design Thinking process is understand the perspective of users and try and discover their insights [6]. Therefore, Embedded Systems being so closely associated with innovation aspect, can improve the traditional Requirements Engineering process by adopting Design Thinking. Mainly because it helps to define requirements more effectively, which is a common embedded system problem [3]. Also, Design Thinking provides a rapid prototyping method, an fundamental aspect in Embedded Systems design [14]. Prototypes emerge early in Design thinking process, this happen because while in regular Requirements Engineering processes early failures are constantly avoided, in Design Thinking they are encouraged. In fact, this is the part of learning from errors that Design Thinking uses to improve the quality of generated solutions.

There are still several potential improvements that Design Thinking adoption can provide to Embedded System Development. They are described below, while the main phases of Design Thiking are presented.

The process of Design Thinking consists of 4 phases: Research (empathize and define), Ideate, Prototype and Testing. There is not necessarily an order, they can be executed simultaneously and many times. The Figure 2 shows how this process is commonly known.

Figure 2. Design Thinking Process

Following there is a description of each phase of design thinking process approaching the use in embedded system development.

A. Research (Empathize and Define)

At this stage, we are supposed to create a solution to a problem that a particular person or people have. For that, the main objective is understand the context in which the solution will exists and how that solution will interact with other things within that environment. Also, it is fundamental to empathize with the user. This means that having their perspective would be possible to understand accurately their real needs and, consequently, focus on creating an efficient solution that really solve their problem.

Although regular requirements elicitation also addresses these concepts of understanding user real needs, the design thinking approach propose some additional concepts. While requirements elicitation focus on the software aspect of a system, Design Thinking sees the system as a whole, as a product and does not ignores any part of it. In addition, Design Thinking also presents the context aspect. It means that throughout the development, besides the concern with the product itself, should exist a special attention to where the product will be utilized. So it wonders some of the characteristics of the environment, such as: the product will stay indoor or outside? The terrain is going to be plain or irregular?

Would Design Thinking feature of seeing the system as a whole, instead of considering just the software aspect, benefits embedded systems in some way? Being a set of hardware and software, embedded systems needs this macro view that design thinking offers? Another relevant question that should be considered is about the context of the product. Design Thinking proposes a lot of effort thinking about the environment where the product will be placed. Therefore, information about temperature, space or the existence of other near interactive products, for instance, would bring some improvement for embedded systems development and help to deliver more quality products?

We believe this phase of Design Thinking could offer additional mechanisms to create a more innovative and efficient embedded systems identifying system requirements that would be hard to list by the more traditional requirements engineering methods.

B. Ideate

After understanding the context and user real needs, it is time to create possible solutions to the particular problem the system is suppose to solve. Ideation phase put together the perspective of all stakeholders, and initially there is one condition: there are not restrictions about the generated ideas. The main characteristics in this process are creativity and imagination. Still, design thinking suggests that collaboration is the key in this phase, where it is better having five people working on an idea for one day, instead of one people working five days on an idea. This way it is possible to avoid a higher amount of thoughts intersections and increase the perspective variety. At the end of Ideation, the single selected solution goes through a process of refinement. Thus, although Ideation requires many ideas and creativity, it also needs rationality and a critical thinking about the viability and constraints related to the final solution.

While in traditional requirements engineering process there is techniques like brainstorming, which can generate various ideas, design thinking uses the ideation process. Although they are very similar, there are some differences between them. Brainstorming is used to generate ideas about a defined problem. Ideation, however, is more visionary and can be used to find solutions for problems not yet defined. Another difference is that ideation allows seeing the same thing with different perspective. So the same problem could be rethinked and redesigned many times, allowing the generation of many different approaches to archive the same result.

With these points listed, we need to analyze if embedded systems could have benefits from the use of Ideation process. Would the rethink and redesign proposed by Ideation process, be able to create more innovative embedded systems? Also, we need to ask if the possibility of finding solution for a problem that does not exist yet, would be useful for embedded systems design.

C. Prototype

In design thinking, the prototype process is about experimenting ideas. Once a possible solution is created, it should be immediately developed. This allows a dynamic and agile way of experimenting the solutions. Prototyping is also a form of analysis of the solution, since it is possible to observe when a solution is too complicated or too simple, too big or small, for example.

While in traditional requirements engineering approach, the focus of prototyping is primarily to show the current stage of development to the client and understand, by their feedback, if the solution being developed is really according to their needs, the design thinking prototype process goes beyond that. In addition to the traditional aspects of prototyping, the DT prototyping process explicitly embraces the concept of early failure. For DT, the failure is one of the more efficient ways to obtain a quality product. The idea is to fail early, correct the

errors and build again. That approach allows the opportunity of testing different and innovative ideas in a security environment until a prototype succeed.

Noticing the difference between the prototyping in traditional requirement elicitation and the prototyping in design thinking, how the design thinking approach would be more effective for embedded systems? Would early prototyping of ideas be more efficient for embedded systems than a later prototyping? Would the higher number of failures in early stages suggested by design thinking prototype help to improve the quality of embedded systems?

We believe that, being complex systems, embedded systems could benefits from the high number of failures at the early stages. Every failure will force the designers of the systems to rethink their solution, and repeating this process constantly at early stages, will avoid the development of a product that don't meet the client real needs. In our vision, having the concept of failure as a process to obtain quality at the end of product development, is a innovative form of creating quality products in embedded systems context

D. Testing

Testing is the last part of design thinking process. At this point, the solution is tested in a real operating environment to verify its performance through metrics. This is done with a minimum viable product. That is, the simplest functional version of a product in which it is possible to have a feedback. The cycle of testing should be done every time a new prototype is created. Thus, testing is a constant part of design thinking, since there are a higher number o prototypes created during the development of a product. Also, it is important to notice that the whole design thinking process should be iterated many times, until the desired product be according to requirements. The testing phase, either in traditional requirements engineering or design thinking are basically the same. Except the fact embedded systems testing are more complex. For instance, to execute a test in embedded system may be necessary to run a entire machine to test a single new functionality of a product.

Therefore, with the increase of bigger and more complex systems, in which innovation and quality are essential, without ignores the schedule and budget, Design Thinking offers alternative ways that could provide more effective and innovative solutions to overcome these aspects. Also, it is important to notice that, in some cases, there is no need of a radical change between traditional requirements engineering techniques and Design Thinking. Merging both could create a great combination. Although, this integration should be more adequate in a context where a system does not requires a higher level of innovation.

To validate the questions raised during this section, we are going to conduct a field research over the next months. Two teams with the same bugdet, schedule and restrictions will develop an innovative embedded system product. While one team will use the traditional requirements engineering process, the other will use Design Thinking. Once this process is done, we will be able to verify, through metrics, if the team that have used Design Thinking obtained any benefits over the other.

V. Conclusions and Future works

Embedded Systems segment is growing likewise the demand for innovative, complex and high quality products. To keep themselves in the competitive market, the companies must find a way to handle these requirements. So, Design Thinking seems an efficient way to archive this.

In future innovative scenarios where everything is connected via Internet, known as Internet of Things (IoT), or where the nanotechnology will be, in fact, a reality. Design Thinking is one of the most appropriated methodologies to elicitate and manages the systems requirements. There is two main points to prove this: first, it is a methodology that gives more support for innovative thinking. While in conventional requirements engineering process may there are some creative techniques, the Design Thinking process embrace creativity and innovation and makes it one of the more important aspects of the process. The second and most important point is that traditional Requirements Engineering techniques focus more on the software aspect of a system. Design Thinking sees Embedded System as a whole. This means that all characteristics of the product are analyzed. Besides hardware and software, there is a special attention to users and the context of the products.

The aim of this paper was to identify trends and benefits of adoption of design thinking during embedded system requirements elicitation. The study focused on finds and analyzes how design thinking approaches can help to improve the requirements engineering process. Considering, at the same time, the complexity and innovation closely associated to Embedded Systems.

This is an initial study that is part of a big project called SEED (Software Engineering for Embedded Systems). The conclusions raised here are essential to the next steps of research. Among them we can cite:

- Analyze a real embedded system project and understand the requirements, detecting lacks of design, which could be solved using Design Thinking.
- Design an embedded system using design thinking and traditional ways adopted by companies. Two teams will perform different design activities and in the end the results are compared.
- Development of a framework to compare the main characteristics of requirements elicitation process, focusing on which points are more important for embedded systems.
- Propose a model of DT for embedded system project and development

Besides these research topics, other studies not mentioned here are approaching software engineering and embedded systems using design thinking. Part of these studies are

currently being done by our team and the results has shown how necessary and promising is this field of research, for example a systematic mapping of traceability of requisites for embedded systems.

REFERENCES

[1] A. Chakraborty et al. "The Role of Requirement Engineering in Software Development Life Cycle, " in *Journal of Emerging Trends in Computing and Information Sciences.* vol 3. 2012.

[2] BBC Research (2014) Embedded Systems: Technologies and Markets. Avaliable from http://www.bccresearch.com/market-research/information-technology/embedded-systems-report-ift016e.html

[3] C. Vetterli, W. Brenner, F. Uebernickel, and C. Petrie, "From palaces to yurts: Why requirements engineering needs design thinking," *IEEE Internet Comput.*, vol. 17, pp. 91–94, 2013.

[4] E. Sikora, B. Tenbergen, and K. Pohl, "Industry needs and research directions in requirements engineering for embedded systems," *Requir. Eng.*, vol. 17, pp. 57–78, 2012.

[5] F. A. T. Abad, et al. "On-chip control flow integrity check for real time embedded systems, " in *Cyber-Physical Systems, Networks, and Applications (CPSNA) 2013 IEEE 1st International Conference on*, 2013. pp. 26-31

[6] H. Plattner, C. Meinel and L. Leifer. *Design Thinking Research: Building Innovators*, Springer Books, 2014, pp. 289.

[7] K. Wiegers and J. Beatty. *Software Requirements*. Redmond, WA: Microsoft Press, 2013, pp. 672.

[8] M. A. Feki, et al. "The Internet of Things: The Next Technological Revolution, " in *Computer, vol. 46*, 2013. pp. 24-25.

[9] M. Törngren, et al. *Vision and Goals: Towards a flourishing eco-system for industry and academia, excelling in embedded systems education, research and innovation*, 2010. pp. 22.

[10] M. M. Kermani, et al. "Emerging Frontiers in Embedded Security," in VLSI Design and 2013 12th International Conference on Embedded Systems (VLSID), 2013, pp. 203-208.

[11] M. Verhoef, et al. "Collaborative Development of Embedded Systems, " in *Collaborative Design for Embedded Systems: Co-modelling and Co-simulation.* Springer Books, 2014.

[12] M. Zeller et al. "Towards Self-Adaptation in Real-Time, Networked Systems: Efficient Solving of System Constraints for Automotive Embedded Systems, " in *Self-Adaptive and Self-Organizing Systems (SASO), 2011 Fifth IEEE International Conference on*, 2011, pp. 79-88.

[13] P. A. Laplante. *Requirements Engineering for Software and Systems*. Boca Raton, FL: CRC Press, 2014, pp 324.

[14] R. Wang. "The design of a rapid prototype plataform for ARM base Embedded System in", *Consumer Electronics, IEEE Transactions on., vol. 50, pp. 746-751, 2004.*

[15] S. Robertson and J. Robertson. *Mastering the Requirements Process: Getting Requirements Right.* Westford, MA: Pearson, 2013, pp. 149.

[16] T, Noergaard. *Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers*. Waltham, MA: Elsevier, 2013

[17] V. Seppo, Innovations in Embedded and Real-Time Systems Engineering for Communication. IGI Global, 2012.

[18] IC Insights: Top 13 Foundries Account for 91% of Total Foundry Sales in 2013

[19] H. Oshana & M. Kraeling, Software Engineering for Embedded Systems, 1st Edition, Newnes, 2013.