Joël Maximilian Mai

11118561

Media informatics

Exposé for a bachelor thesis

Merging Human-computer interaction with agile software development

Contents

Tags

Motivation and problems to solve

The Waterfall process comes with risks for companies. To implement such a process correctly, all requirements for the finished product have to be sorted out at the beginning of the project, therefore requiring a lot of time at the start, leaving little time to react to changes. But it is almost standard for software projects to have changing requirements during the development phase, which results in undesirable products for the end-user or unsatisfied clients [1, p. 14]. The cost of change compounds with time [1, p. 22].

Out of these problems arise a few requirements to the product, that has to be met always. The product has to be desirable to the end-user. It has to be technically feasible. And to project has to deliver business value fast.

To solve these problems, developers and designers came up with solutions, such as working agile or having iterative design phases. Starting with agile, the most frequent problem with agile comes with implementing an agile framework in the company. All members of the project and the client have to commit to the agile manifesto. But most clients cannot leave someone at the agency to stay in contact and collaborate with the teams. Therefore a big share of Web Agencies kept parts of the original waterfall phases in their processes [2]. But because of this management decision, developers and designers cannot fully work agile, and therefore only the negative impacts of working the agile show, such as messy tickets, volatility errors, and opaque workflows. This needs additional tweaking such as the backlog [3].

Improvements to Development or Design phases kept their focus on each separate phase in dictating the other [1, p. 22]. For example, agile frameworks usually need designers to work in smaller increments [4], with not enough time to research, which results in concepts that do not grab the whole aspect of all requirements or rely on their assumptions [5]. This results often in less collaboration between the two teams [1, p. iv][6, p. 115]. Both teams work after one another instead of parallel or intertwined [7, p. 9]. Therefore Designers are driven by the deadlines set by the project manager for their concepts and not collaborating closely with software architects leads to unviable solutions [1, p. 4]. But Developers build working software, which is no longer valuable to the end-user because they put technical feasibility over the set boundaries by the UX team. The lack of focus put on the end-user in agile development is a major concern and the biggest weakness [8, p. 312].

The Human-centered Design approach solves many of the problems above, but also comes with problems. First of all, it is important, that the Design includes all humans in the required research. Concentrating only on the end-user will leave gaps in the requirements that can lead to an undesirable product by the client. More often than not, the Researcher writes User Stories wrong [9]. They already have a solution in mind, leaving out the possibility of a better solution. Usually, the User is kept in mind during the

development of the concept, by Empathy Maps or other Tools. But once the concept is finished and given to the development team, those end-user requirements are forgotten and replaced by technical restraints, leading again to an undesirable product [1, p. 19, p. 33].

Once a product reaches the Quality Assurance Department, other problems show up, like that the final deployed product only receives irregular Updates. Neither agile frameworks nor human-centered design keeps in mind, that requirements after deployment can change, and therefore the view on sustainable products has to change too [1, p. 18, pp. 30-31].

It is needed to differentiate between starting a new project and maintaining a product. When starting a new project, software architects are not included during the concept phases, which leads to no feasible solution. Designers usually waste time fetching out documentation and designs, while Developers have to wait until the design is set. When maintaining a product, Designers and Developers need to collaborate more closely. They also have to be in the meetings as Consultants to discuss feasible solutions with the client to develop a satisfying product. The business also has to listen and measure the success of the product in the live environment.

HCI and Agile share the values of user-focus and iterative cycles, which leads to the conclusion, that combining them and solving the main problems, should lead to a workflow, that solves all key problems [10]. When trying to fix all of the problems stated above, we still need to keep in mind the time, budget, and scope of the projects. Therefore a subgoal should be to reduce waste and unused documentation.

Status Quo

Derived Thesis

Procedure

A new project is started

A product needs an update

A Bug is found in a product

A Client wants a new feature

A product is transferred over to the agency

Project Plan

August 23, 2022	Submit Exposé
August 30, 2022	Register thesis?
November 28, 2022	Thesis submission
December 1, 2022	Colloquium

Acronyms

HCD
Human-Centered Design

HCI
Human-Computer Interaction

UI
User Interface

UX
User Experience

Glossary

Roles of SCRUM

Client
Clients are the Contractors of a product. They are also known as business owner/ product owner.

Customer
Customers are using the product. They are also known as shoppers or users.

Project Manager
The project manager has the goal to ensuring that there are clearly defined project objectives and parameters and that the project team meets these objectives.

Product Manager
The role of a product manager is to ensure product success throughout the entire product life cycle, during and beyond the initial project.

Agile Integrator
The agile integrator has the task to implement agile mehtods into the organisation and processes. He is also known as SCRUM Master. He is responsible for creating the best possible working conditions for the team and supports them in organizing themselves.

Desinger
Can be a Frontend Developer, UX Designer, UI Designer or else. Has the focus on the layout of the product.

Architect
Can be a Requirement Engineer, Software Engineer, Software Architect, UX Designer or else. Has the focus on the structure and dependencies of the product

Developer
Can be a Frontend, Backend Devleoper, Software Architect or Software Engineer or else. Has the focus on the technical constraints of the product.

Tester
Can be partially automated. But every piece of software that is critial to the system has to be tested, by someone or something. Usually Quality Assurance Department writes tests and implements these testing infrastructure.

Definitions

Requirement Engineer
The role requirement engineer is almost completly integrated into the product owner. But due to the product owner being the client, the requirement engineer is a intelligent tool to use to refine client side requirements. They help understanding the clients needs and translating them for the team members

UI Designer

UX Designer

Software Architects
Software Engineer

Software Developer

Frontend Developer

Backend Developer

Continuous Learning

Phases of a Product

Continuous Evolution

Continuous Exploration

Continuous Integration

Continuous Deployment

Continous Relasing
Release on demand

References

[1] L. Ratcliffe and M. McNeill, *Agile experience design: A digital designer's guide to agile, lean, and continuous*. New Riders, 2011.

[2] A. Windolph, "10 spannende einblicke zum stand des projektmanagements 2022 [statistik] - projekte leicht gemacht." `https://projekte-leicht-gemacht.de/blog/projektmanagement/einblicke-projektmanagement-2022`, January 2022. (Accessed on 07/12/2022).

[3] Y. Nakao, M. Moriguchi, and H. Noda, "Using agile software development methods to support human-centered design," 2014.

[4] K. Drozd, "The art of thinking big and working small — atlassian." `https://www.atlassian.com/agile/agile-conversations/think-big-work-small`. (Accessed on 07/18/2022).

[5] D. Bluestone, "How to combine user-centered design and agile development :: Uxmatters." `https://www.uxmatters.com/mt/archives/2015/12/how-to-combine-user-centered-design-and-agile-development.php`, December 2015. (Accessed on 07/14/2022).

[6] M. Minge and A. Föhl, *Bringing It Together: Three Approaches to Combine Agile Software Development and Human-Centered Design*, pp. 21–27. Faculdade de Informatica, PUCRS, Porto Alegre, Brazil, 01 2019.

[7] P. Forbrig and M. Herczeg, "Managing the agile process of human-centred design and software development," 2015.

[8] C. S. A. Peixoto, "Human-computer interface expert system for agile methods," in *Proceedings of the ITI 2009 31st International Conference on Information Technology Interfaces*, pp. 311–316, 2009.

[9] S. James Robertson, "user story considered harmful," 2016. (Accessed on 06/21/2022).

[10] A. Elssamadisy, "Human computer interaction (hci) and agile compatibility." `https://www.infoq.com/news/2007/06/hci_agile/`, June 2007. (Accessed on 07/14/2022).