



A Case Study on Combining Agile and User-Centered Design

Yekaterina Pakhtusova, Swati Megha, and Nursultan Askarbekuly^(✉)

Innopolis University, Innopolis, Tatarstan 420500, Russia
{y.pakhtusova,s.megha,n.askarbekuly}@innopolis.university

Abstract. Agile is a dominating software developing technique throughout the world. It promotes high development speed, work in short iterations, and delivering working software as soon as possible.

User-Centered Design is a more traditional approach that puts users and their needs first. It advocates spending more time understanding your users, their needs and wants, while the implementation process can be delayed and last longer. Both approaches try to deliver the best possible software for the users but utilize different means. This paper examines through a case study how Agile and User-Centered Design can be combined and whether this combination brings additional value. A development method is then suggested, combining the two philosophies.

Keywords: Agile · User centered design · Scrum · Product development

1 Introduction

User Centered Design (UCD) is a technique for developing software products, that focuses on users and their needs in each phase of the design process [12]. Agile Software Development (ASD) is a collection of approaches and practices in software development, that aims to maximize the value provided to the customer through early and continuous delivery of working software [15]. These approaches share some similarities [7], like

- They rely on an iterative development process, building on empirical information from previous cycles
- Both techniques place an emphasis on the end-user, encouraging participation throughout the development process
- Both approaches emphasize the importance of team coherence (one of the aspects of UCD is that the whole team should constantly think about the user during the development process)

Both of these methods have a goal of delivering quality software to stakeholders, but there are fundamental differences [7]:

Supported by Innopolis University.

- UCD suggests maintaining specific design products to support interaction with developers, while agile methods look for minimal documentation
- ASD methods are largely against an up-front period of investigation instead of writing code, whereas UCD inspires the team to understand their users as much as possible before the product implementation begins

Several attempts have been made to connect ASD and UCD, but most of them inevitably have trade-offs. This paper focuses on one of such methods called Big Upfront Design (BUD), which supposes that designers must be given sufficient amount of time to find out the basic needs of their users before any code is written.

1.1 Objective

The main goal of the research is to show how to apply Big Upfront Design method of connecting Agile and User Centered Design. Within the study, the research team has implemented an online courses aggregator service called “EduHub” and attempted to apply a combination of Agile and UCD practices in accordance with the BUD method. The study analyses the BUD approach and suggests potential improvements.

1.2 Outline

In Sect. 2 the author describes existing approaches for combining ASD and UCD. In Sect. 3 the methods used for this research are listed. Section 4 is about how UCD and ASD were combined in practice by the author and development team who worked on online course aggregator service. Section 5 lists the results of the research and evaluation of overall work. In Sect. 6 some conclusions about the project are presented, which might be useful for people who conduct similar research work.

2 Related Work

This section describes User Centered Design, Agile Software Development, and existing techniques used for combining them. The author considers 5 most well-known combination techniques and lists its advantages and drawbacks.

2.1 User Centered Design

UCD History. The concept of User Centered Design has been around for a long time [1]. The design of some everyday things sometimes is bad and leaves users frustrated because of the inability to complete a certain task. But when things are designed good and can be used intuitively, we can say that designers used the principles of UCD, that is - constructed things for the users.

But the term UCD originated only in 1986 - from the book “User-Centered System Design: New Perspectives on Human-Computer Interaction” written by

Norman and Draper [19]. Norman continued to develop UCD concepts further in his book “The Psychology Of Everyday Things” published in 1988 [18]. There he suggested 7 principles for facilitating designer’s tasks.

Norman’s work emphasized the need to thoroughly investigate the necessities and aspirations of the users and the expected uses of the product. The obligation to involve actual users, often in the real-life environment, was a natural evolution in the field of User Centered Design.

UCD. UCD is considered to be one of the most popular frameworks for developing user interfaces. It is based on the active participation of users to improve the understanding of user and task requirements, and the iteration of design and evaluation phases [4, 16].

The activities which UCD is composed of [11] (see Fig. 1):

- Define the users’ context
- Specify the user requirements
- Design solutions to meet the user requirements
- Evaluate the designs against requirements

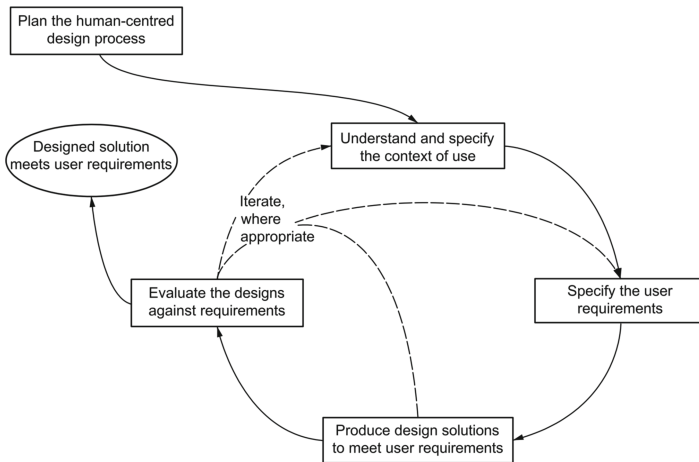


Fig. 1. UCD process as described by the standard ISO 13407

The main principles of UCD are listed below [11]:

- The design is based upon an explicit understanding of users, tasks, and environments
- Users are involved throughout the design and the development
- The design is driven and refined by user-centered evaluation
- The process is iterative

- The design addresses the whole user experience
- The design team includes multidisciplinary skills and perspectives

As a proof of best internationally endorsed practice, UCD processes are also defined in ISO documents, including ISO 13407 and the associated technical report, ISO TR 18529.

To illustrate the effectiveness of UCD, several studies have been conducted. The results show that 72% of the respondents reported UCD methods had made a significant impact on product development in their organizations. The majority said UCD methods had improved the usefulness and usability of products developed in their organizations, 79% and 82% respectively[16].

2.2 Agile Software Development

ASD is a software engineering philosophy that promotes [2]:

- individuals and interactions over processes and tools
- working software over comprehensive documentation
- customer collaboration over contract negotiation
- responding to change over following a plan

Nowadays, Agile methodology shows that it has indisputable advantages over traditional development techniques:

- Only 9% of Agile projects failed in 2015 in comparison with 29% failure for Waterfall projects [25] (these figures might differ in reality due to some arguable occasions regarding CHAOS Report [13])
- 71% of surveyed companies reduced project costs as a reason for adopting Agile [8]

These facts are correlated with ASD popularity:

- 71% of surveyed companies admitted using Agile approaches sometimes, often, or always [21]
- Agile projects are 28% more successful than traditional projects [20].

2.3 Combining Both

For more than a decade, software companies tend to integrate Agile Software Development methods and User Centered Design [22]. Several practitioners integrated UCD in their Agile process with varying degrees of success. This paper will consider the five most successful approaches to integration ASD and UCD.

To organize the work between developers and usability experts, Sy [24] proposes a method called “*Parallel tracks*” that requires developers and user experience designers to work in parallel tracks after the planning iteration also called iteration “0”. It gives the opportunity for usability experts to have enough time to gather users’ data, analyze that data and to propose design solutions. However, adopting this approach may cause the lack of communication which could lead to

misunderstanding and resentment between designers and developers, since there can emerge a situation when designers and developers would work on completely different tasks due to such desynchronization.

Another similar approach was proposed by Armitage [3], but it concerns only the organization of designers’ work. “*Design work done on parallel levels*” suggests the design work is done on three parallel levels from unit to global level:

- Provide detailed designs for the requirement developed in the current or next iteration.
- Redesign software developed in previous releases (a release is a set of several iterations).
- Provide overall product vision, to keep a global coherence throughout the project and developed software.

However, it is not clear if the evaluation of the design done in one iteration is conducted by users.

Deuff et al. [10] present another proposition of process for UCD in Agile called “*Sequence of an iterative design phase and an iterative development phase*” that gives a good place to an upfront designing. In classical Agile methodology usability experts have to switch between too many tasks (gather the necessary data, define the design, test) because they do not have time to prepare the data beforehand. To resolve this issue they propose to cut the project in 3 phases: Design, Development and Final test. Due to that reason, we cannot state that the process is iterative, as each phase proceeds after another.

Constantine and Lockwood [9] propose another approach, which is the integration of Usage-Centered Design and Agile. *Usage-Centered Design* is more focused on roles than on users and on usage scenarios also knew as task cases. Roles and tasks are identified by stakeholders. The process is composed of iterations that are all composed of the succeeding steps (see Fig. 2):

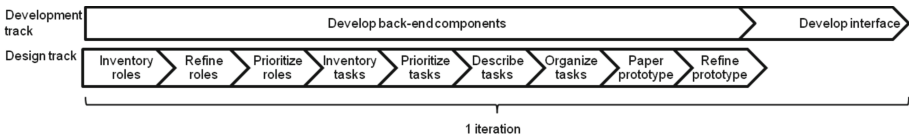


Fig. 2. One iteration of Usage-Centered Design adapted to Agile methods

During this time developers create the back-end components. When the prototype is refined, they develop the interface. The disadvantage is that the evaluation of designs against requirements is not covered in this approach, which goes against the UCD principles.

Big Upfront Design. The fifth approach for combining Agile and User Centered Design is called “*Big Upfront Design (BUD)*”.

Agile methods do not encourage big upfront design [5, 17]. In reality, an analysis that is conducted by the product owner before the start of the development process is necessary to compose the product backlog, but no best practice is defined to help the product owner to complete this task. That is why some usability experts propose to conduct a big analysis upfront. Other practitioners are against this practice and prefer to use the iteration “0” to conduct a short analysis and then go deeper throughout the project if the better analysis is needed.

Supporters of BUD

Chamberlain [7] in his work insists on a big upfront design before any development: “UCD practitioners must be given ample time in order to discover the basic needs of their users before any code gets released into the shared coding environment.” A lot of work has to be done during one iteration in any Agile methodology so that usability experts do not always have time to ask questions or to take a global view and ensure the homogeneity and consistency of the solution.

For Brown [6], in long projects sometimes it is necessary to devote more time for analysis in order to gather the necessary data.

Opponents of BUD

On the contrary, Armitage [3] insists that it is too risky and time and money-consuming to design deeply in advance and it is totally against Agile practices which encourage “trial and error to reduce the risk of building the wrong thing”. A big upfront design might reduce the quality of the software and its design [3].

Another problem is the difficulty to accept changes later when a big upfront design was done, which goes against the Agile values “Responding to change over following a plan” and “Working software over comprehensive documentation” [3, 14, 17].

Conclusion

For Brown [6], it is nonsense that no upfront design is allowed in Agile-UX. In fact, Agile developers work with a high-level plan also called a roadmap. It is also necessary for usability experts to develop a roadmap in the form of a diagram or a sketch. This way the team has time to build global vision while not spending too much time on a design phase that never ends. This is essential to identify future possible technical difficulties.

BUD covers only three of the four UCD activities: understanding and specifying the context of use, specifying the user and organizational requirements, and producing design solutions. There are no specific recommendations for evaluation of a design against requirements, so there is no guarantee that big upfront design will meet the users’ requirements.

BUD does not ensure users’ involvement, evaluation, iterativeness, and multidisciplinary even if they are recommended to be used for a better design. It becomes obvious that the goal of this approach is to implement the first UCD principle: understanding of users, tasks, and context.

2.4 Research Purpose

This research is aimed at applying Big Upfront Design (BUD) on a real project with high user involvement connected with online education. During the research process, the author will try to improve existing BUD practice by ensuring that all the principles of User Centered Design are respected. By applying BUD on the project, it will be possible to see its advantages, disadvantages, and potential improvements to the approach.

3 Methodology

This section provides a detailed overview of the methodologies used by the author for the combination of ASD and UCD.

3.1 Context

To understand how to combine Big Upfront Design and Agile Software Development, the author applied both of these techniques to the development of a real project. The project is an online service for aggregation of Massive Open Online Courses (MOOCs) called EduHub. EduHub provides an opportunity to search for MOOCs from several popular course providers (e.g. Coursera, Udemy, Udacity, etc.) in one place, with the ability to filter and sort them according to some criteria defined by the user. The project was selected due to the high popularity of online education (according to [23] more than 101 million users study online).

3.2 Agile Software Development

The team that is working on EduHub project consists of three people - two software developers and one UI/UX designer (the author). To work effectively, the team selected Agile Software Development as a methodology for creating software. But, for the purpose of genericity and context of the project the team does not stick to some specific framework (e.g. Scrum, Crystal, XP, etc.), but rather selected a set of practices adopted by popular frameworks [8], such as:

- Backlogs - the team maintains an ordered list of product requirements that will be delivered in future sprints. The team has separated product and sprint backlogs.
- User stories - description of a feature of the product written from the perspective of a user.
- Iterative and Incremental Development - development of a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system.
- Scrum events: planning, review, and retrospective

- planning - at the beginning of a sprint, the team agrees on the sprint goal, a short description of what they are forecasting to deliver at the end of the sprint
- review - at the end of a sprint, the team hold a review, where the members review the work that was completed and the planned work that was not completed
- retrospective - during this event the team reflects on the previous sprint and discusses what can be improved in the current development process
- Planning poker - the team members make estimations of the difficulty of the tasks and then discuss it
- Velocity tracking - a technique for forecasting the team's performance and planning of the future sprints
- Pair programming - two coders work collectively at one computer. One writes code while the other inspects each line of code as it is typed in. The two programmers shift roles frequently.
- Cross-functional team - a team consisting of people with different functional expertise working toward a common goal.

3.3 User Centered Design

The author uses traditional activities of which UCD consists. These include iterations of the steps:

1. Define the users' context
2. Specify the users' requirements
3. Produce design solutions to meet user requirements
4. Evaluate the designs against requirements

As it was already mentioned in Sect. 2, in Agile there is no best practice for the product owner how to define the product backlog, but this task should be done before the start of the Agile development process. To support the product owner for this task, some usability experts propose to conduct a big analysis upfront. So, the team decided to utilize BUD to analyze what needs to be done.

During Big Upfront Design phase of the design process, the main goal is to capture users' needs, usability goals, the context of use and design criteria. This time is also used to define users or to build personas.

To implement the second step of UCD (specify the users' requirements), several user research methods were utilized:

- Online survey - to understand, who our possible users are, how often they visit resources connected to online education, what difficulties they have while searching for a course, and to get information from as many people as possible, the team conducted an online survey using Google Forms.
- Personas - is a technique of creation of a representative user based on available data and user interviews. Basing on the results of the aforementioned survey, the author created the three most common personas of potential users.

- Heuristic evaluation - is a method for evaluating a website against a list of established guidelines. The team evaluated 5 popular online course providers to analyze how they provide service to users.
- Use Cases - the author created a description of how users use a particular feature EduHub, including the steps users take to accomplish each task.
- Interviews - the author conducted one-on-one discussions with users to get detailed information about a user’s attitudes, desires, and experiences.
- Contextual interviews - they enable the observation of users in their natural environment, giving a better understanding of the way users work. For this study, the author observed how people searched for online courses using existing solutions.
- Prototyping - to allows the team to explore ideas before implementing them, the author created mock-ups of the site.

To ensure that the design conforms to the requirements (step 4 of UCD), the team conducted usability testing to identify user frustrations and problems with the site through one-to-one sessions where a user was asked to perform some tasks on the site and their reaction was recorded.

4 Implementation

This section outlines the phases through which the project underwent.

The basic outline of the team’s working process can be seen on the following Fig. 3:

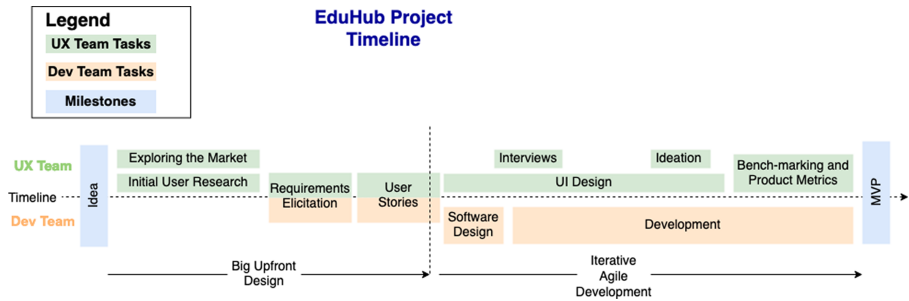


Fig. 3. Project timeline up to the MVP stage

There were two main phases of the development process:

4.1 Big Upfront Design Phase

After the team selected the idea of EduHub project, the author started to conduct market and user research.

During this stage, an online survey was conducted in order to identify who are our potential customers and what are their pains that they encounter during the search for MOOCs. Also, the data about online education and users from the open sources were used to make initial conclusions. The main point of this research is the creation of personas - description of potential customers and their needs:

- Student
They study at university/school and search for additional materials that can help them to understand their studies. They usually search for free courses as they do not possess a lot of spare money. They expect that an online aggregator will be fast and with a wide choice of courses.
- Professional
It is a person who usually already has a higher education and work experience. They work in rapidly developing industries where new technologies are emerging every so often, so they have to keep up with changes or they will lose their qualifications. They are ready to pay good money for courses. The most important feature of the site is information content - in order to compare available options and choose the best one.
- Lifelong learner
Any person who constantly improves their knowledge and expands their horizons can be called a lifelong learner. They like to learn new things just for their curiosity and thirst for knowledge. This makes it harder to highlight some features that are common for lifelong learners as they are varying in age, gender, income and education level, etc. In site, the most important things are a diverse choice of courses and user-friendly interface.

After preliminary market analysis, the author started to analyze existing sites that provide access to online courses (e.g. Coursera, Udemy, Udacity, etc.) to detect their good features to adopt them in EduHub project and avoid inconveniences that are present on some sites.

Up to this point only the UX team (the author) was involved. Then the development team joined.

Together with the development team, the UX team elicited the initial requirements in order to construct user stories later. These composed user stories were taken as initial direction for developing the system.

Using the requirements, the development team was able to determine the stack of technologies that will be used during the coding stage.

Overall, BUD phase took about one month to complete. During this time, the general roadmap of the project was constructed, the development and UX teams got an understanding of the future work, and initial requirements were gathered, which was enough to start the development.

4.2 Iterative Agile Development Phase

After completing BUD phase, the author started to conduct interviews with real people who resembled some type of personas that were mentioned above.

During the interviews, the author asked questions regarding their pains during the search for MOOCs and what features they would expect from a service that provides access to all the possible online courses at one place. The most common requirements were:

- wide range of available courses
- ease of use
- availability on PC and mobile devices
- ability to sort and filter courses
- ability to read and write feedback about courses

This was a way to validate our requirements and user stories that were constructed previously. Also, it gave ground for further ideation that took place later on.

Simultaneously with the interviews, the UX team started working on UI design production. Firstly, the author produced paper sketches of the future interface for the sake of simplicity and speed. During paper prototyping, the planned design underwent some changes, which was not critical because of the choice of a prototyping tool.

The design was produced with the help of Figma - a free tool for creating digital prototypes. The prototypes that were created include:

- search for online courses
- sorting and filtering results
- viewing course information (details, syllabus, feedback)

The development team worked in parallel, preparing software architecture and design of the system. By the time the software design activities were completed, the UX team already had the first batch of UI sketches ready for use.

The author transferred materials needed for implementation of the design in code to the developers to start coding. The author then continued to work on the design simultaneously with programming work of developers.

As the team worked using Agile practices, every week at the end of a sprint there was a team meeting, where the team discussed how the sprint went, what was done and not done, aroused questions, and plans for the next sprint. The two teams discussed and assigned priorities to tasks together. The effort estimation for tasks in the product backlog was done by the development team. Based on the priorities and effort estimates tasks were selected tasks for the sprint.

During the development of Minimal Viable Product (MVP) by developers, the author created the design for the rest of the features of EduHub. The design team worked one step ahead of the developers to ensure that the latter would always have UI designs ready to work on. The MVP included the following features:

- possibility to create an account
- profile of the user with personal information and settings
- possibility to leave feedback for a course
- addition of a course to favourite courses list

- email subscription to a newsletter
- display of top-5 courses of the site

In the end of UI Design stage, the author conducted an ideation session in order to generate some ideas for additional features of the EduHub which could be implemented during the future iterations. This ideation session was important because it allowed to create a post-MVP vision for the product development.

At some stage the team realized the need of administrator panel to control the content of EduHub service. The team (as current stakeholders) discussed what features are needed to be available in the administrator panel. After collecting all the requirements, the author implemented the administrative service using the same principles as for the EduHub service - simplicity and focus on functionality.

When the UI design for the MVP product was finished, the UX team started to develop the product metrics. It was important to create the metrics before the MVP was ready to assess how the product meets user requirements when it would be ready. Having the metrics and results of testing, it is much easier to make improvements.

When the MVP was ready, the author started conducting usability testing to find out if the site is understandable, convenient, and most importantly meets the requirements of the users.

For the sake of usability testing, 20 people who are similar to our target audience were selected. 15 people were students and young specialists who search for online courses for their studying process, other 5 were people who just enjoy self-learning and use online courses to learn something new, not for their job.

The interviewees were given 14 tasks:

1. look through top-5 courses on the main page
2. sign up for a newsletter on the main page
3. search for a course from the home page of a site
4. when results of the search are shown, filter the search to display only free courses
5. select a course that a user likes and go to its details
6. look through the description, syllabus, and feedback
7. go to the site-provider of the course to learn it
8. create an account on the site
9. search for a new course and add it to favourite courses
10. leave feedback for this course
11. go to your profile and look at your favourite courses. Unmark a course from favourites
12. go to settings and add your personal information
13. reset your password
14. log out from your account

While the subjects were performing the tasks, the author observed their behaviour and actions. All the respondents were able to fulfill the tasks correctly and under acceptable time. This can be explained by the fact that all the interviewees were active Internet users and the site utilizes habitual UX design patterns that most of the Internet users have already encountered.

5 Evaluation and Discussion

This section describes the results of the work done and some insights that were received during the project.

5.1 Results

In this section the author enumerates the problems encountered during the project and their possible solutions and the insight that can be useful for the similar projects.

At the initial stage of the project, the UX team worked alone. The reason for it was that user and market research had to be conducted to verify the initial hypothesis before any design and development effort could take place. The hypothesis was that there is a need for an online course aggregator, i.e. online learners are having difficulty with searching for online courses due to a big number of learning platforms and course options. This is the essence of BUD, and the exact circumstance for it to be applied.

During the initial research phase, the UX team faced a problem that there was not enough secondary information (from some other sources) about online education situation in Russia, so the team had to collect it as primary information (from own research). Due to this situation, it took more time for the initial research than it was planned.

The development team was more or less idle during the initial research stage. Since EduHub is an academic project, the development team took advantage of this period to explore the solution space in terms of the state-of-art architectural designs that could be applied during the project. In industrial circumstances, it is a question whether the development team should even be hired/formed/onboarded before the initial user and market research stage produces a conclusive result in regards to hypothesis validation.

Requirements elicitation and documentation has been done through collaboration between the development and UX teams. Starting from this stage the development team could be heavily involved with the project without idle periods.

It is healthy to have both teams collaborating during the production of functional and quality requirements (product backlog) since the development team can assess and estimate the effort needed for developing the functionalities, while the UX team can deal with the prioritization of backlog items.

It should be recognized that BUD in its essence is a sequential approach (and sometimes looks like a Waterfall development model), but this is not a bad idea for kick-starting the project. Once the initial set of requirements is in place, the Agile stage begins, which means that Agile practices can prevail within the development process from this point on.

The gathered and documented requirements were just the initial set, and there was no intention to freeze the requirement and refuse change. They were just a fulcrum that gave a direction for further actions. And the UX team took all the advantages of the agile principle of welcoming changes in the project.

Having the initial set of requirements ready, the development team took their time to make basic architectural choices fitting the project's context and to decide on the appropriate technological stack. The UX team used this time to produce initial UI sketches so that the development team would have the design to implement in the code. This is relevant in the case of software products with graphical user interface. In a similar manner, the UX team was timely supplying the necessary designs to the development team to avoid delays and bottlenecks.

Once the UI designs require less time and effort from the UX team, the designers have to be creative to keep themselves busy until the MVP arrives. The UX team worked on the vision of the future features of EduHub (Ideation session, additional user interviews), as well as developed the metrics for assessment of MVP.

MVP allowed the UX team to conduct further Usability Testing, collect analytics and test the hypotheses that were created during Ideation.

6 Conclusions and Future Work

This section presents the overall conclusions about this research and some plans for the future work.

6.1 Conclusion

In this paper, we have implemented an online course aggregator called EduHub. Two teams were working on that project - UX team and development team. To guide and synchronize a workflow between two teams UCD and ASD methodologies were used. To glue UCD and ASD together, the author proposed to use BUD technique. But the original description of BUD did not fully satisfy UCD process activities and the author had to improve and complement the original BUD method.

To ensure compliance with the fourth step of UCD (evaluation of design against requirements) the author proposes to use usability testing after completing MVP. After that, usability testing can be conducted after any newly added feature.

Overall, we can say that the study is successful. The author with the team achieved the desired results and fulfilled the goal of the project.

We can conclude that application of Big Upfront Design technique for combining UCD and Agile brought beneficial results and allowed to follow the guidelines of both methodologies.

The only disadvantage (though it actually can be an advantage) is that one has to spend some additional time at the beginning of a project for a BUD stage to do preliminary research. This time which UX team spends can be used for searching of developers or deciding on technology stack for the project.

6.2 Future Work

In the future, the author plans to conduct more diversified usability testing, because due to the tight schedule and academic purpose of the research the author had to test students and employees of their university. Since interviewees were from one city and initially predisposed to active Internet usage, the results can be biased.

As a future development of the project, the author and the developers plan to introduce additional features which were extracted from user interviews and created during an Ideation session. One of such features is the introduction of study plans - roadmaps containing a list of courses needed to learn a profession.

References

1. Abras, C., Maloney-Krichmar, D., Preece, J.: User-centered design (2004)
2. Agile Alliance: Agile manifesto (2001). <http://www.agilemanifesto.org>
3. Armitage, J.: Are agile methods good for design? *Interactions* **11**(1), 14–23 (2007)
4. Askarbekuly, N., Solovyov, A., Lukyanchikova, E., Pimenov, D., Mazzara, M.: Building an educational product: constructive alignment and requirements engineering. In: Ahrum, T.Z., Karwowski, W., Kalra, J. (eds.) *AHFE 2021. LNNS*, vol. 271, pp. 358–365. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-80624-8_44
5. Blomkvist, S.: Towards a model for bridging agile development and user-centered design. In: Seffah, A., Gulliksen, J., Desmarais, M.C. (eds.) *Human-Centered Software Engineering—Integrating Usability in the Software Development Lifecycle. Human-Computer Interaction Series*, vol. 8, pp. 219–244. Springer, Dordrecht (2005). https://doi.org/10.1007/1-4020-4113-6_12
6. Brown, D.D.: Five agile UX myths. *J. Usability Stud.* **8**(3), 55–60 (2013)
7. Chamberlain, S., Sharp, H., Maiden, N.: Towards a framework for integrating agile development and user-centred design. In: Abrahamsson, P., Marchesi, M., Succi, G. (eds.) *XP 2006. LNCS*, vol. 4044, pp. 143–153. Springer, Heidelberg (2006). https://doi.org/10.1007/11774129_15
8. CollabNet VersionOne: The 13th annual state of agile report (2019). <https://www.stateofagile.com>
9. Constantine, L., Lockwood, L.: Process agility and software usability: toward lightweight usage-centered design. *Inf. Age* **8**(8), 1–10 (2002)
10. Deuff, D., Cosquer, M., Foucault, B.: Méthode centrée utilisateurs et développement agile: une perspective & « gagnant-gagnant » au service des projets de R&D. In: *Conference Internationale Francophone sur l'Interaction Homme-Machine*, pp. 189–196, September 2010
11. ISO DIS: 9241-210:2010 Ergonomics of human system interaction - Part 210: Human-centred design for interactive systems (2009)
12. Endsley, M.R., Jones, D.G.: *Design for Situation Awareness. An Approach to User Centered Design*. CRC Press, Boca Raton (2004)
13. Eveleens, J., Verhoef, C.: The rise and fall of the chaos report figures. *IEEE Softw.* **27**, 30–36 (2010)
14. Ferreira, J., Noble, J., Biddle, R.: Up-front interaction design in agile development. In: Concas, G., Damiani, E., Scotto, M., Succi, G. (eds.) *XP 2007. LNCS*, vol. 4536, pp. 9–16. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73101-6_2

15. Highsmith, J.: Agile Software Development Ecosystems. Addison-Wesley, Boston (2002)
16. Mao, J.Y., Vredenburg, K., Smith, P.W., Carey, T.: The state of user-centered design practice. *Commun. ACM* **48**(3), 105–109 (2005)
17. Nodder, C., Nielsen, J.: Agile Usability: Best Practices for User Experience on Agile Development Projects. Nielsen Norman Group, California (2010)
18. Norman, D.A.: The Psychology of Everyday Things. Basic Books, New York (1988)
19. Norman, D.A., Draper, S.W.: User-Centered System Design: New Perspectives on Human-Computer Interaction. Lawrence Earlbaum Associates, Hillsdale (1986)
20. PricewaterhouseCoopers: Agile project delivery confidence (2017)
21. Project Management Institute: Success rates rise. Transforming the high cost of low performance (2017)
22. Schwartz, L.: Agile-user experience design: an agile and user-centered process? In: The Eighth International Conference on Software Engineering Advances, ICSEA 2013 (2013)
23. Shah, D.: By the numbers: MOOCs in 2018 (2018)
24. Sy, D.: Adapting usability investigations for agile user-centered design. *J. Usability Stud.* **2**(3), 112–132 (2007)
25. The Standish Group International: Chaos report (2015)