# Task Allocation Between UX Specialists and Developers in Agile Software Development Projects

Kati Kuusinen



# Task Allocation between UX Specialists and Developers in Agile Software Development Projects

Kati Kuusinen
Tampere University of Technology, Tampere Finland
kati.kuusinen@tut.fi

**Abstract.** Synchronizing efforts between developers and user experience (UX) specialists is one of the major challenges in agile UX work. In this paper, we report results of a study conducted over a release cycle of six agile software development projects in five companies, considering the task allocation and cooperation in the team. Team members (N = 31), including product owners, UX specialists, and developers, reported weekly on the UX-related tasks they had contributed to and whether the UX specialist had participated. We identified three forms of cooperation: minimal, product owner–UX specialist, and developer–UX specialist. Our study suggests that for projects operating in the minimal cooperation mode, the collaboration concentrates on the user interface (UI) design, while other aspects of UX work are downplayed. At the same time, many UX-related tasks were successfully handled by developers alone. Therefore, to support UX work integration, we suggest a task-oriented integration approach for projects with minimal UX resources.

**Keywords:** User experience (UX); Agile development; human-centered design (HCD), UX design work, Agile UX, human–computer interaction (HCI)

## 1 Introduction

In agile software development, small, cross-functional teams produce software in short, incremental iterations. The team should include all of the necessary expertise to enhance the team's efficiency and communication [14]. As agile methods [14] do not define the role of a UX specialist (UXS), it is often unclear how the UXS fits in the project team. Companies still struggle to integrate the UXS role into agile development practices [26]. Problems related to the amount and timing of UX work, as well as to the synchronization of tasks between developers and UXSs are common [26]. In addition, UX resources are often scarce in agile software projects [8, 31], which naturally reduces the amount of UX-related work that is possible. In many cases, development teams have to cope with only limited help from a UXS or, in extreme cases, with no UX support at all [25]. There have been attempts to improve team collaboration and to redistribute the workload of overburdened UXSs by creating the means to include developers in UX-related activities [5, 25, 29].

Regardless of the level of contribution by UXSs, certain activities that have a strong impact on UX, such as developing a UI, need to be conducted during development.

Current recommendations for agile UX development suggest that work should be divided into activities that are conducted during agile development and those that are conducted prior to development iterations as upfront design work [7]. UX-related work is typically separated into its own stream directed by independent UXSs [6, 28]. However, it has been reported that organizations following these practices struggle with issues, such as balancing the amount of upfront design work and managing communication between developers and UXSs [7, 26]. Separating the UXS's activities seems to exclude the UXS from the core project team and affect the timing of feedback cycles between disciplines. Both of these issues blur the project vision, hindering within-project communication and endangering the realization of UX design [4, 15, 23].

In this paper, we concentrate on the cooperation and task allocation between developers and UXSs. Our goal is to clarify which UX-related tasks can or even should be handled by developers and which require the special competence of a UXS. We contribute to the understanding of how UX work is conducted in agile projects, explaining participating roles, and task and collaboration frequencies. Thus, we present a framework of collaboration between UXSs, developers, and product owners (PO), and offer process implications and suggestions to help improve collaboration in different collaboration setups. We report results from a longitudinal multiple-case study in which we studied the UX work-related task allocation and cooperation between the team members (N = 31), including developers, POs, and UXSs.

The rest of the paper is structured as follows: Section 2 presents related work. Section 3 discusses the research methodology we used to conduct the follow-up study and introduces the project contexts studied. Section 4 presents results from the weekly survey period. Section 5 presents results from a retrospective survey in which the participants shared their experiences of the ways of working in each project. Section 6 discusses the limitations of the research. Section 7 discusses the results and implications of the identified cooperation types. Section 8 concludes the paper.

## 2 Related Work

Agile software engineering refers to a collection of methodologies that share fundamental similarities in being lightweight and flexible [32]. Agile methodologies embrace change by reducing its cost throughout the project. Means to reduce the cost of change include prioritizing work based on business value, utilizing small incremental iterations and short feedback cycles, being cooperative and open in communication, and delivering software early and continuously [1, 14]. In addition, agile methodologies seek efficiency through a less hierarchical management structure, close cooperation with the customer, and the use of self-organizing, cross-functional teams [1, 14].

We consider UX in this paper to be *a person's perception of the value that results from the use or anticipated use of software in a certain context of use*. This definition is adapted from [13, 16]. By the term *agile UX work*, we refer to agile software development [14] that puts emphasis on developing software that the user values. There are differences between academic UX research and industrial UX development in terms of the conception of UX; whereas UX research concentrates mostly on hedonic aspects and emotions, companies concentrate more on functionality and usability issues [30]. Since our research considers UX work in the industry, the emphasis is on usability and functionality.

An agile team should include all of the expertise necessary to construct running software that satisfies user needs [14]. Including people from different disciplines makes communication generally more challenging [12], but separating UXSs into their own teams, instead of including them in agile development teams, may easily lead to degraded communication. UXSs become seen as outsiders, and developers tend not to take ownership of UX issues. When UXSs and developers are separated, teams encounter problems with timing and the implementability of the design [10, 20]. Hodgetts [15] considered it vitally important for UX practitioners to see themselves as part of a project team and to conduct their tasks according to that perception. Lee [24] stated that UXSs need to be active participants in order to be embedded in agile teams. Isomursu et al. [17] concluded that UXSs' responsibilities should be in line with the expectations of development teams.

UX resources are still often scarce in agile software projects [8, 31]. In many cases, development teams have to cope with only limited help from a UXS or, in extreme cases, with no UX support at all [20, 25]. Ferre [9] introduces usability techniques developers can utilize in their work. Ungar [29] has included developers in design work together with users and UXSs in the *Design Studio* method, and Leszek et al. [25] has supported developers through the *Office Hours* concept in which each development team can consult a UXS periodically for an hour or two at a time. Bornø et al. [5] described experiences from a one-day UI *Redesign Workshop* for developers and UXSs.

Little is known about the actual daily work and task allocation between developers and UXSs. The majority of agile UX research has concentrated on the development process or on endeavors to modify user-centered design (UCD) practices so that they will be better suited for agile development [6]. A systematic review listed the practices mentioned most often in agile UCD work [7]. These practices included conducting little upfront design work, ensuring close collaboration between UXSs and developers, designing iteration ahead of development, prototyping, and conducting user tests [7]. Joshi et al. [18] found the following activities most important for achieving usability goals: user studies, UI design, usability evaluation and development support. Boivie et al. [3] determined that usability designers allocate roughly 25–50% of their time to analysis, 20–60% to design, and 10–50% to evaluation activities. The contribution of UXSs varies from single activities to full participation throughout the project [3]. In projects where there is no UXS involved, separate usability evaluation activities are more common [3]. Moreover, projects tend to favor informal feedback gathering methods, such as

asking users' opinion, instead of conducting formal user tests [3, 22]. Most feedback gathering activities occur in the early stages before implementation [22]. Ferreira et al. [10] reported an observation study of a team with a separate UXS who fed the team with ready-made designs. The team coordinated its work by (i) inspecting the design and identifying mismatches with what had already been implemented, (ii) interpreting the design in terms of working software, and (iii) prioritizing the design and determining the parts that had already been implemented [10].

To conclude, the UXSs' role is often undefined in development projects [3, 17]. It is unknown what a UXS's daily work consists of in agile projects and how developers can contribute to UX. Moreover, UX resources are often scarce in companies [8, 20, 25, 31], and therefore it is important to utilize such resources wisely. This paper contributes to the understanding of the focal tasks of UXSs and how developers and POs can participate in UX work.

## 3 Methods and Participants

Our research goal was to increase the understanding of UX-related work in agile enterprise software development. We piloted the study in two projects, one of them reported in [19]. The results of the pilot study indicated that there seemed to be differences in both the emphasis of tasks and the degree of communication between projects in the same company, which could impact the satisfaction of team members. We expected to gain interpretive results from this study. Moreover, we aimed at comparing the ways of working to team members' experiences of the project and their conception of the success of the project and its outcome. Our research questions were as follows:

1. How do UXSs, POs, and developers contribute to UX?
2. Which tasks should be collaborative and which can be handled by a certain role?
3. Which practices do team members consider good or bad, and why?

Our research consisted of surveying agile projects about their ways of working in terms of UX-related work. We defined "UX-related work" in the survey as follows:

> *Any work that contributes to understanding or defining user needs, designing and developing to meet the user needs, and evaluating or ensuring that the needs are being met. The work can be conducted by any project member; we do not limit the definition of UX work to a certain role (e.g. UX Specialist).*

We surveyed members of each participating team using a weekly web questionnaire over a release cycle. We sent individual survey invitations before Friday morning in the earliest time zone of participants' locations and allowed response time until Monday evening in the latest time zone. We used three slightly different surveys, one intended for each of the following groups:

1. Developers and POs of teams that included a UXS.

2. UXSs.
3. Developers and POs of teams that did not have a UXS.

For the Group 1 survey, participants reported weekly which UX-related tasks they had participated in and whether the UXS had been involved in the task. UXSs (Group 2) reported tasks that they had contributed to and also the roles of persons who had participated in the activity, if applicable. Participants in Group 3 reported tasks that they had contributed to. In addition, they reported on an imaginary setting where a UXS would have been involved in the team. They reported the tasks that the UXS would have contributed to, if any. They also reported if they had needed a UXS during the week and the reason why or why not.

Respondents selected the tasks from a pre-defined list (Table 1), which included two open fields for tasks that were not listed. This list was based on our earlier research on important UX-related tasks conducted in two companies [20, 21]. In that study, 116 people in various roles related to software development responded to an open-ended question asking: *What are the most important tasks of UX Specialists?* We analyzed the responses utilizing the affinity wall method described in [2]. In addition, we evaluated and iterated the resulting task list with a group of UXSs and software architects in order to cover all of the focal tasks.

**Table 1.** List of UX-related tasks in the weekly survey.

| Task | Description |
|---|---|
| Created concepts | Designing and sketching early ideas |
| Clarified user requirements | Gathering and interpreting user requirements |
| Clarified user definitions or target user groups | Defining who the users and user groups are |
| Planning user data gathering | Planning the user participation, studies, and tests |
| Conducted user study | Studying users, their behavior, and contexts |
| Conducted user testing | Evaluating the system, prototype, concept, or idea by testing it on users |
| Created UI designs | Designing user interaction or flow, making graphic design etc. Design can be of any fidelity |
| Reviewed UI designs | Inspecting and evaluating UI design feasibility |
| Created architecture designs | Designing the software structure and deciding on the fundamentals of the system |
| Created or groomed product backlog | Deciding of the scope of the outcome and the order of implementation (Scrum practice) |
| Planned a feature | Planning new or modified features for the release |
| Shared understanding of the UI | Discussing the UI design and decisions related to it, |

| Task | Description |
|---|---|
| design | communicating the design idea to stakeholders |
| Shared understanding of the technology | Discussing technical feasibility, its limitations, and possibilities |
| Implemented UI | Implementing the user interaction and interface |
| Determined how to implement UI design details | Deciding on nuances and details related to the UI design, e.g. when implementing on different platforms |
| Made changes in the UI design | Modifying the UI design e.g. based on feedback or review |
| Reviewed implementation | Checking that the implementation corresponds to the design |
| Had a demo session | Demonstrating the software to customers, users or stakeholders (Scrum practice) |

In addition to the weekly surveys, each participant filled in a retrospective survey about their experiences and their evaluation of the ways of working in terms of the UX-related work in the project. Through open-ended questions, the participants reflected on the good and bad practices and lessons learned from the ways of carrying out UX-related work in the project. At the beginning of the study period, we also interviewed one team member from each participating project who had a good understanding of the working methods of the project. We asked the interviewees about the ways of working in the project and about the company in general in order to understand the work context.

We analyzed the weekly data using descriptive statistics, including mean, standard deviation, count of occurrences, and ratios of different cases. We calculated counts of occurrences for each task role-wise. We compared task frequencies and ratios between collaborative and non-collaborative occurrences of reported tasks. We analyzed the retrospective survey responses using a qualitative content analysis method.

### 3.1 Participants

We selected projects using the following criteria:

- The project utilized agile methods. The basic criterion was that the PO considered the project agile; we did not want to interfere with the actual execution of the project.
- The project had a release cycle of six months or less.
- The outcome was enterprise software that would be used by a person.
- The outcome had a graphical UI that required design work. In particular, we excluded pure backend development projects and projects tailoring third-party systems.
- UX design work was ongoing or starting soon.
- Project members were willing to participate.

The participant population consisted of project team members. Table 2 presents characteristics of participating companies and projects, and Table 3 describes the participants. The PO and developer roles were defined similarly to Scrum [27], wherein the PO is responsible for maximizing the business value of the project, creating and maintaining the feature list (product backlog), and acting as a link between the developers and stakeholders; developers form a self-organizing team that is responsible for delivering working software in an iterative and incremental manner. At the beginning of each iteration, the development team selects those tasks from the product backlog that they will commit to delivering by the end of the iteration [27]. UXSs' responsibilities are inherited from human-centered design, as defined in [16]. Those include understanding and specifying the context of use, specifying the user and organizational requirements, producing design solutions to meet those requirements, and evaluating and redesigning until those requirements are met [16]. In all the projects, UXSs' roles were defined vaguely. Originally, the following tasks were assigned to UXSs. In P2 and P3 UXSs were to ensure the product viability and to create the UI design. In contrast, in P1 and P4, the UXS was to create the UI design while the PO was responsible for the product vision. In P5, the UXS was working as a consultant guiding the team while the PO was responsible for the vision.

**Table 2.** Description of participating companies and project teams. P4 and P5 were conducted at the same company. Legend: PO = product owner, UXS = UX specialist, W = amount of studied weeks.

| Project | Company description | Team size and location | W |
|---|---|---|---|
| P1 | An engineering and technology company with around 20 000 employees worldwide. Utilized both waterfall and Scrum practices. Several small distributed UX teams and UXSs. | 11 of which 8 developers located in Russia, 1 PO and 1 part-time UXS co-located in Finland. 2 participating sites. | 24 |
| P2 | An IT service company with 100–500 employees in Finland. UXSs working in project teams. | 6 of which 4 developers, 1 PO, 1 UXS, all co-located in Finland. 1 participating site. | 17 |
| P3 | An IT service company with 100–500 employees in Europe. Utilized Scrum. A centralized UX team in one site and distributed specialists in others. | 5 of which 2 developers, 1 PO, 1 UXS, all co-located in Finland. 1 participating site. | 7 |
| P4 | An IT service company with around 20 000 employees worldwide. The company mainly utilized customer-defined processes. It had a centralized UX team on one site and numerous distributed UXSs on several sites | 8 of which 5 developers and a scrum master in China, and 1 PO and 1 UXS in Finland. 2 participating sites. | 8 |
| P5 | | 5 of which 3 developers in Finland at location A, 1 PO in Finland at | 7 |

| Project | Company description | Team size and location | W |
|---|---|---|---|
| | (Projects P4 and P5 were conducted at this company). | location B, and 1 UXS in Latvia. 3 participating sites. | |
| P6 | A mobile technology company with 100–500 employees worldwide. Utilized agile practices and customer processes. A centralized UX team. | 3 of which 1 developer and 1 PO in Finland co-located, the second developer in Estonia. Possibility to consult a UXS at another location in Finland. 3 participating sites | 13 |

Of all respondents, 45.2% were from Finland, 25.8% from Russia, 22.6% from China, 3.2% from Estonia, and 3.2% from Latvia. In total, there were 38 team members working for the projects, of which 31 responded to our survey for a response rate of 81.6%. Some of the non-respondents were backend developers who felt that the study did not concern them because of their minimal contribution to end-user UX. Altogether we collected 237 weekly sheets from the projects.

In P1, two developers had conducted some courses in HCI while the PO and the rest of the developers had had a short training in HCI. In P2, the PO and developers had conducted some courses in HCI. In P3, one developer had majored in HCI and he also had ten years of experience in UX design work. The PO and UXS of P3 had conducted some studies in HCI. Developers of P4 had no training in HCI. A developer in P5 and P6 had conducted one course in HCI. The rest in those teams had no training in HCI. UXS of P2 had majored and UXS of P1 had minored in HCI. The rest of the UXSs had conducted some courses in HCI.

**Table 3** Description of participants. Legend: PO = product owner, IT = information technology, M = mean, SD = standard deviation. Age and experience are expressed in years. *We did not get demographic data from one UXS and three developers.

| Role | Developers (N = 19)* | POs (N = 6) | UX Specialists (N = 5)* |
|---|---|---|---|
| **Mean age** | M = 31.07 (SD 4.95) | M = 35.00 (SD 3.03) | M = 40.00 (SD 8.00) |
| **Educational background** | IT | IT | IT, society and culture, or industrial design |
| **Education in HCI** | None to major subject. The majority had some self-learning to some courses | Some self-learning to some courses | Some courses to major subject |
| **Development experience** | 0–20, M = 8.46 (SD = 4.68) | 2–9, M = 7.00 (SD = 2.53) | 0–20, M = 8.75 (SD = 9.84) |
| **UX design work experience** | 0–10, M = 2.46 (SD = 3.57) | 0–1 M = 0.25 (SD = 0.27) | 5–20, M = 11.25 (SD = 7.09) |
| **Project management experience** | 0–5, M = 1.07 (SD = 1.43) | 0–6, M = 4.50 (SD = 2.26) | 0–5, M = 2.00 (SD = 2.45) |
| **Agile work experience** | 0–8, M = 4.00 (SD = 2.04) | 0–7, M = 4.92 (SD = 2.46) | 5–9, M = 6.00 (SD = 2.00) |

# 4 Results of the Weekly Survey

In this section, we present our results in terms of the description of ways of working in the projects, task allocation and the frequency of reported tasks, and the work roles that cooperated with UXSs in their work.

## 4.1 Task Allocation and Frequency as Reported by Developers and POs

In the Group 1 survey (developers and POs of teams that included a UXS, i.e. projects P1–P5; P6 is not included since it did not have a UXS), the most often reported tasks were related to clarifying user requirements, feature planning, and creating and reviewing UI designs (Table 4). Those tasks in which cooperation between UXSs and

other team members was reported most often included conducting user study and discussions related to the UI design or technological issues, and carrying out other UX-related activity such as holding meetings (kickoff or status meetings), creating user documentation, and implementing test cases. UXSs participated least often in architecture design creation, the creation and grooming of product backlog, and implementation reviews.

Others collaborated most often with UXSs when conducting user studies (100% of reported occurrences were conducted in cooperation with the UXS), sharing understanding of technical issues (88.9%) or the UI design (86.4%), creating concepts (50.0%), reviewing (41.9%) or creating the UI design (45.1%), and clarifying user requirements (41.2%).

**Table 4.** Occurrences of UX-related tasks as developers and POs of P1–P5 reported. N = 23.

| **Task** | **Total reported times** | **Times without UXS** | **Times with UXS** | **Ratio with UXS / total** |
|---|---|---|---|---|
| Clarified user requirements | 68 | 40 | 28 | 0.412 |
| Reviewed UI designs | 57 | 29 | 28 | 0.491 |
| Planned a feature | 54 | 39 | 15 | 0.278 |
| Created UI designs | 51 | 28 | 23 | 0.451 |
| Had a demo session | 49 | 38 | 11 | 0.224 |
| Made changes to the design | 46 | 37 | 9 | 0.196 |
| Reviewed the implementation | 42 | 30 | 12 | 0.286 |
| Created or groomed product backlog | 40 | 24 | 16 | 0.400 |
| Created architecture designs | 39 | 36 | 3 | *0.077* |
| Determined how to implement UI design details | 31 | 22 | 9 | 0.290 |
| Shared understanding of UI design | 26 | 13 | 13 | 0.500 |
| Created concepts | 23 | 14 | 9 | 0.391 |
| Clarified end user definitions or target user groups | 22 | 3 | 19 | *0.864* |
| Shared understanding of technical issues | 18 | 2 | 16 | *0.889* |
| Planned user studies or user tests | 11 | 8 | 3 | 0.273 |
| Other UX-related activity | 6 | 1 | 5 | 0.833 |
| Conducted a user study | 2 | 0 | 2 | *1.000* |
| Conducted user testing | 2 | 2 | 0 | 0.000 |
| **TOTAL** | **587** | **366** | **221** | |

### 4.2 Collaboration Types

We identified three collaboration types in the study. These collaboration types are not intended to be discrete; projects can have traits from several of them. Instead, the collaboration type indicates the emphasis of communication in the project. In P1 and

P3, collaboration in general was less than in P2 and P5 (Table 5). In P4, the majority of reported collaboration occurred between the PO and UXS. This was also the case for P6, which did not have a designated UXS in the team. In projects with the minimal collaboration type, others collaborated with the UXS most often when conducting a user study (100.0% of the cases) and when sharing understanding about the UX design. By contrast, in the Developer–UXS type of collaboration, the rest of the team collaborated with the UXS most often during the following tasks: having discussions over the implementation of design details, sharing understanding of UI design, and creating concepts and UI design. Thus, we conclude that when there was less collaboration, it concentrated on the UI design and possible user studies. When the amount of collaboration increased, the team could also discuss the actual implementation and concepts behind UX design decisions. In effect, the quality of implementation in terms of the UI design was ensured beforehand through discussions over design details. Developers reported repeat occurrences of clarifying user requirements without the UXS in both the minimal and PO–UXS collaboration modes. This most likely indicates a form of rework; developers did not have direct contact with users, and the UXS clarified user requirements for themselves. Including developers in the clarification work might have saved developers time and increased their understanding of user requirements.

**Table 5.** Percentage of collaborative occurrences in minimal type projects (P1 and P3) and in Developer–UXS type projects (P2 and P5). Percentage of all reported tasks for which P6 members would have wanted the contribution of a UXS.

| **Task** | **P1 and P3 With UXS** | | **P2 and P5 With UXS** | | **P6 UXS** |
|---|---|---|---|---|---|
| | **M** | **SD** | **M** | **SD** | **Would** |
| Created concepts | 20.8% | 0.21 | **90.00%** | 0.10 | 33.3% |
| Clarified user requirements | 29.7% | 0.05 | 61.9% | 0.05 | 40.0% |
| Clarified user definitions or target user groups | 16.7% | 0.17 | 50.0% | 0.00 | **57.1%** |
| Planned user studies or user tests | 25.0% | 0.00 | 33.3% | 0.00 | 25.0% |
| Conducted a user study | **100.0%** | 0.00 | | | **100.0%** |
| Conducted user testing | *0.00%* | 0.00 | | | 28.6% |
| Created UI designs | 26.7% | 0.07 | **90.0%** | 0.10 | **66.7%** |
| Reviewed UI designs | 37.5% | 0.13 | **82.3%** | 0.01 | 47.6% |
| Created architecture designs | *4.2%* | 0.04 | 9.1% | 0.00 | 0.00 |
| Created or groomed product backlog | 8.7% | 0.09 | 32.1% | 0.01 | 0.00 |
| Planned a feature | 20.0% | 0.00 | 22.5% | 0.23 | 14.3% |
| Shared understanding of the UI design | **83.3%** | 0.17 | **96.2%** | 0.04 | |
| Determined how to implement UI design details | *9.4%* | 0.13 | **100.0%** | 0.00 | 12.5% |
| Made changes to the UI design | 34.6% | 0.09 | 73.3% | 0.07 | 47.6% |
| Reviewed the implementation | 16.9% | 0.15 | 68.2% | 0.32 | **57.1%** |
| Had a demo session | 20.8% | 0.08 | 35.0% | 0.00 | 28.6% |

When there was no designated UXS involved, the team desired help primarily in conducting user studies, creating UX designs, clarifying target users, and reviewing the implementation. These might be the areas where problems become visible when working without a UXS. Of the 13 observed weeks, the P6 team reported 8 weeks that they would have needed the contribution of a UXS. The external UXS contributed for two weeks (weeks 2 and 10). The reasons mentioned most often for requiring the contribution of a UXS were the following: (i) to make better design decisions in terms of the user flow and UI (7 mentions in 6 weeks), and (ii) to get feedback on the current design of the user flow and UI (5 mentions in 5 weeks). The reasons why contribution from a UXS was not needed were the following: (i) we did not do or plan anything in the project this week that would affect the user experience, and (ii) we already had the needed competence within the project team (for both, 6 mentions in 5 weeks).

### 4.3 Roles that Cooperated with UX Specialists per Task

UXSs conducted the majority (62.8%) of all reported tasks in collaboration with others. They collaborated most with POs; 51.5% of all reported occurrences of collaboration included the PO role. Developers were involved in 34.9% of all reported collaborative tasks. The proportion for customers or users was 26.6%, and for other design-related roles it was 10.1% (including other UXSs, usability experts, and graphic designers). POs most often collaborated when creating architecture design (100.0% of reported collaborative occurrences related to this task), clarifying user requirements (84.0%), holding demo sessions (81.8%), and creating concepts (78.9%). UXSs collaborated the most with developers during demo sessions, when discussing the UI design and when determining how to implement design details. Developers did not participate in conducting user studies or tests, clarifying end user definitions or target user groups, or creating architecture design together with UXSs.

UXSs reported the only collaborative occurrences of the following: conducting user testing, holding a demo session, and creating architecture designs. Other collaborative tasks included clarifying user requirements (85.3% of reported task occurrences were collaborative) and planning user studies (84.7%). UXSs worked alone most often when implementing UI (58.6%), making changes to the design (50.7%), and creating UI designs (46.9%). Table 6 describes the proportional values per task of UXSs working alone and in cooperation with different roles. Cooperation with each role is represented as a proportion of the total number of reported cooperative occurrences of each task.

There were remarkable differences in cooperation between the projects. The UXS was in continuous collaboration with customers and users in P2, P3, and P4, whereas the customer or user was never mentioned in P1 and P5. The PO was responsible for collaboration with users and customers in both of those projects. In P5, the PO arranged two short sessions with users in order to understand their needs and to evaluate the product concept. In P1, collaboration with users was rare, and the project mainly concentrated on the technical problem it was trying to solve. By contrast, P2 was the most collaborative project in terms of both the frequency of collaboration and

variety of roles with which the UXS continuously collaborated. In P4, the PO participated frequently in each task, except for conducting user studies and tests. The developers of P4 were involved primarily when the UXS explained the UI design to them, during demo sessions, and when making changes to the UI design. In P1, there was less reported cooperation, occurring almost solely between the PO and managers.

**Table 6.** Average proportional occurrences of working alone and with others as reported by UX specialists (N = 5). PO = (with) product owner, DEV = (with) developer, CUS = (with) customer, OUX = (with) other UX specialists, including graphic designers.

| Task | Alone | Together | PO | DEV | CUS | OUX |
|---|---|---|---|---|---|---|
| Conducted user testing | 0.000 | 1.000 | 0.000 | 0.000 | **1.000** | **1.000** |
| Created architecture designs | 0.000 | 1.000 | **1.000** | 0.000 | 0.000 | 0.000 |
| Had a demo session | 0.000 | 1.000 | **0.818** | **1.000** | **0.818** | 0.000 |
| Clarified user requirements | 0.147 | 0.853 | **0.840** | 0.427 | **0.588** | 0.000 |
| Planned user studies or tests | 0.153 | 0.847 | 0.316 | 0.000 | 0.211 | **0.789** |
| Shared UI design understanding | 0.225 | 0.775 | 0.599 | **0.588** | 0.347 | **0.080** |
| Reviewed UI designs | 0.236 | 0.764 | 0.534 | 0.226 | 0.141 | 0.254 |
| Created concepts | 0.307 | 0.693 | **0.789** | 0.386 | 0.235 | 0.000 |
| Clarified user definitions or target user groups | 0.324 | 0.676 | 0.479 | 0.000 | 0.521 | 0.000 |
| Created or groomed product backlog | 0.329 | 0.671 | 0.571 | 0.265 | 0.204 | 0.000 |
| Reviewed the implementation | 0.333 | 0.667 | 0.300 | 0.340 | 0.180 | 0.000 |
| Conducted a user study | 0.420 | 0.580 | 0.000 | 0.000 | 0.538 | 0.000 |
| Determined how to implement design details | 0.428 | 0.572 | 0.000 | 0.182 | 0.000 | 0.000 |
| Planned a feature | 0.453 | 0.547 | 0.703 | **0.618** | 0.322 | 0.223 |
| Created UI designs | 0.469 | 0.531 | 0.435 | 0.383 | 0.094 | 0.081 |
| Made changes to the UI design | 0.493 | 0.507 | 0.307 | 0.396 | 0.247 | 0.074 |
| Implemented UI | 0.586 | 0.414 | 0.222 | 0.000 | 0.000 | 0.000 |

## 5 Results of the Retrospective Survey

This section presents results of the retrospective survey in which the participants evaluated the project and their ways of working. We present the results in terms of the identified collaboration types.

### 5.1 Minimal Cooperation Type Projects

After the weekly study period, we asked team members to share their opinions and experiences regarding the ways of working during the studied release cycle. POs evaluated the impact of UX-related issues on project success. The PO of P1 considered that having a UX engineer (instead of a design-oriented UXS) working in close collaboration with the developers and basically guiding the development work would have been a more suitable approach for the project. He also stated that the UX design was often late "*so most of the time developers just did it 'like before'*." They also needed to do some UI-related rework, as no one, including the users, understood "*the functionality that end users actually wanted*." They might have benefited from a rapid prototyping approach with repeat user evaluations, as it is often difficult for users to understand their own needs beforehand.

In P3, the PO explained that they had learned to be more aware of the UX budget; they had used too much too early, which had led to a situation whereby the UXS could not participate in the project as much as she should have in the later stages. The PO of P3 stated: "*We had a good intention to emphasize UX in the project and we did quite comprehensive designs early in the project. In some areas we did a bit of overdesign and at some point UX got ramped down due to tight budget*." The PO summarized their lessons learned as follows:

- *"Create designs on as-needed basis as the project proceeds."*
- *"Avoid overdesign because things tend to change during projects."*
- *"UX is a continuous process, not one big push at the beginning of the project."*
- *"Time spent on UX design pays off even if budget is tight."*

To conclude, it seems that the POs of projects with the minimal collaboration type would have preferred the Developer–UXS collaboration type for their projects.

A developer of P3 stated: "*In the future I would do almost all the design together between developers and UX designers to avoid communication issues, as well as to avoid doing unnecessary work. Also, not burning the entire UX budget so early in the project to be able to review and improve UI-related implementations also later in the project.*" Also the UXS agreed: "*I would design more in co-operation with the developers (something I do today)*."

### 5.2 Developer–UXS Cooperation Type Projects

The P2 team was pleased with the UX work in general. They especially appreciated that the UXS was part of the team from the beginning: "*We got* [a UXS] *right from the start of the project. The person has been in the project all the time and not just conducting quick UX fixes*" (PO, P2). The PO also shared that they had been able to decrease project costs by negotiating with the customer about reasonable customer requirements, because the UXS had studied the actual user needs.

The P2 team reported frequent cooperation with users. The UXS had conducted user studies earlier and validated design decisions with users. Still, both the UXS and

developers would have wanted even stronger user involvement. Developers in P2 wanted either to meet end users or to get better reasoning for design decisions: "*Reasoning behind design decisions could have been communicated better to developers, as the developers were in no direct contact to end users*" (developer, P2). Another developer mentioned that not being able to meet users decreased their motivation, as the developers did not really know for whom they were developing the software.

Developers in P5 were also pleased with the close collaboration with the UXS. One of them desired the "*shifting discussion from technical level to user level*." Even though they had a user participating in the process, one developer wished for "*more iteration with end user*." On the other hand, the PO of P5 complained that he did not have clear view of the UX work, since the UXS mostly collaborated with the developers. Also, the PO of P2 mentioned that "[UXS's] *tasks should be similarly visible on the* [*kanban* (a tool to visualize workflow)] *board as the other developers' tasks are*." He added that breaking down tasks into chunks is important, because tasks described on too general a level can decrease developmental efficiency.

### 5.3 PO–UXS Cooperation Type Projects

The P6 team was only able to consult an external UXS occasionally. The PO of P6 reported that even the minor contribution of a UXS significantly improved the developed software. He expressed that he would never conduct a project again without the contribution of a UXS, adding: "*I would definitely emphasize regular short UX meetings, 1-1.5 hours weekly. That would actually keep a small project more on track and to help the project be more successful. And it also prevents from rework. The whole team should participate in these UX meetings*."

The UXS in P4 was pleased that UX issues were considered at the project level. The PO of P4 was able to learn how to conduct some UX work by himself. However, the developers were less pleased with the situation in which they had, in practice, no chance to impact design decisions, as ready-made designs were communicated to them to be implemented as-is. The PO felt that the developers were inexperienced, and, because of that, he thought that they would not have been able to challenge the UI design. Most of the developers also did not speak English very well.

## 6 Limitations

We studied only projects developing enterprise systems. All of the projects were Finland-based, although 54.8% of the respondents were from other countries. We utilized theoretical sampling instead of random sampling for selecting the participant projects. The participants were aware of what was being measured, which could have made the study prone to performance bias. Moreover, prompting the participants to think about the UX work in their project weekly might have made participants more aware of UX work and its importance.

The results are based on self-reports from the team members, and, as such, the results are not as reliable as with observation studies. However, since observing geographically distributed projects over a release cycle is impossible in practice, we selected this research methodology despite its limitations. Participants reported their tasks using a pre-defined task list. Some tasks may not have been reported, since they were not included in the list; however, it was possible for participants to add other tasks. The list was based on earlier research on the most important UX tasks and was therefore not exhaustive. We did not define *collaboration* in the survey, which could have led to differences between participants in their perception of collaboration.

Not all of the project members responded the survey, which could have introduced possible attrition bias. However, the response rate was excellent at 81.6%. Also, not every participant reported every week. Projects did not provide us with actual work-hour data. Thus, we asked the participants to report absences on the weekly survey in order to be able to distinguish between non-response and absence. However, there are non-responses in within-person data, which may also have introduced attrition bias. Because the UXSs and other team members reported independently on the same instances of collaboration, we were able to cross-check the data, and no significant outliers were found.

## 7 Discussion

To summarize our main findings, we observed the following three types of UX-related cooperation in the projects:

1. Minimal cooperation (P1, P3)
   - Either one single person responsible for UX work, mainly without project members' contributions, or several people conducting it separately
2. Close cooperation between the UXS and the PO (P4, P6)
   - PO and UXS conduct UX work together and ready-made designs are communicated to developers
3. Close cooperation between the UXS and developers (P2, P5)
   - Developers and UXS conduct the majority of UX work together; PO plays a smaller role in UX tasks

Of these, number 3 – the Developer–UXS collaboration type – was most favored among participants. However, implementing this mode of operation requires that both the UXS and the developers be experienced and willing to work together. The UXS should be able to allocate design tasks, which is challenging [26]. Moreover, it is important to have a PO who understands and appreciates UX work. By contrast, number 2 – the PO–UXS cooperation type – might be better for more inexperienced teams and for projects where the project scope is less clear. The rationale is that in this mode of operation, the PO always holds the reins, and there is a clear chain of command throughout the development. To our understanding, number 1, the minimal cooperation type, should usually be avoided because it introduces more problems

than benefits. This was observed in our study and also previously in [11, 20]. Table 7 discusses the benefits and challenges related to each cooperation type in more detail.

We found that clarifying user requirements, feature planning, and reviewing and discussing UX designs were often collaborative activities. Developers conducted these activities irrespective of UXSs' contributions – either between developers or with the UXS. We interpret this as being because those are the core activities needed to be able to implement purposeful software. Thus, when there is insufficient communication between the UXS and developers, developers need to form their own understanding of the user need and UI design solutions. Moreover, their understanding might be significantly different from what the UXS designed. Thus, clarifying user needs, feature planning, and discussing UX design should be collaborative activities.

**Table 7.** Benefits and challenges in the identified types of cooperation

| Type | Benefits | Challenges |
|---|---|---|
| **Minimal** | UXS can concentrate on UX tasks without disturbance [10]. | Synchronizing work between UXS and developers. Maintaining the big picture of the project. Can lead to unfit design and double work. Spending the UX budget too early. |
| **PO–UX** | UX issues at project level. Helps in maintaining the big picture of the project. | High overhead cost especially in distributed projects. Developers have less impact on the design. |
| **Dev–UX** | Developers have access to reasoning behind design decisions. Discussion about the design is easier. Enables making smarter compromises between design and technical limitations. | Tendency to allocate no or too little time for planning and user studies, risk of piecemeal work and of compromising UX too much for technical reasons. |

Developers and POs emphasized the importance of having a UXS as part of the team from early on and throughout the project. The PO of P5 mentioned that he would utilize a UXS even for small projects with tight budgets, since he saw how it had improved quality and efficiency due to the decreased amount of rework needed. The PO of P3 stated, *"Time spent on UX design pays off even if the budget is tight."* Also, developers from P1, P2, P3, and P5 mentioned that working with a UXS was an advantage in the project. Participants felt that the early work devoted to understanding user needs and thus the scope of the developed system had clarified the project vision and reduced uncertainty throughout the project. In addition, UXSs and developers should cooperate in order to create implementable UI designs: UXSs learn to take technical limitations into account, and developers learn basic UI design principles. Cooperation can also decrease the amount of rework when it is clearer for both the UXS and the developers which parts have already been implemented and how they impact on future designs. In addition, understanding the explicit user needs

helps developers to design the implementation details in terms of the UI. Considering the PO role, the PO of P5 successfully gathered user feedback and organized workshop sessions in order to gain understanding of user needs. On the other hand, in P1 (in which the PO was also responsible for user communication), the user need remained unclear throughout the project. To conclude, it is beneficial to share the responsibilities of UX-related work among team members. However, it should be acknowledged that the PO and developers need to have a certain level of expertise to be able to successfully execute these tasks.

## 8 Summary and Conclusions

This paper reported on results of a longitudinal multiple-case study in which we studied six agile projects in five companies over a release cycle. Participants reported weekly the UX-related activities that they had participated in and whether the UXS had been involved. Developers and POs contributed the most to clarifying user requirements, reviewing UI designs, and feature planning. UXSs contributed most often in creating and reviewing UI design and changing the design. When there was no designated UXS involved, the team desired help primarily in conducting user studies, creating UX designs, clarifying target user groups, and reviewing the implementation.

We identified three descriptive types of cooperation, namely minimal, PO–UXS, and Developer–UXS cooperation. In projects with the minimal cooperation type, the UXS works mainly apart from the other team members, while with the PO–UXS and Developer–UXS cooperation types, the UXS works mostly with the PO and developers, respectively. The Developer–UXS cooperation was the most desirable cooperation type among the participant projects.

## 9 References


1. Beck, K. et al. Agile Alliance. Manifesto for Agile Software, (2001). Available at http://agilemanifesto.org.
2. Beyer, H., & Holtzblatt, K. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann (1998)
3. Boivie, I., Gulliksen, J., & Göransson, B. The lonesome cowboy: A study of the usability designer role in systems development. *IwC, 18*(4), 601–634. (2006)
4. Budwig, M., Jeong, S., & Kelkar, K. When usability met agile: A case study. In *Proc of the 27th international conference extended abstracts on Human factors in computing systems* (CHI EA '09). ACM, New York, 3075–3084. (2009)
5. Bornø, N., Billestrup, J., Andersen, J. L., Stage, & J. Bruun, A. Redesign workshop: involving software developers actively in usability engineering. *Proc. NordiCHI* 2014. ACM, 1113–1118.

6. Brhel, M., Meth, H., Maedche, A., & Werder, C. Exploring principles of user-centered agile software development: A literature review. *Information and Software Technology, 61*, 163–181 (2015).
7. da Silva, T., Martin, A., Maurer, F., & Silveira, M. User-centered design and Agile methods: a systematic review. Proc. *Agile Methods in Software Development (Agile 2011)*.
8. da Silva, T., Selbach Silveira, M., & Maurer, F. Ten lessons learned from integrating interaction design and agile development. *Proc. Agile Conference (AGILE), 2013*, 42–49, IEEE (2013)
9. Ferre, X. Integration of Usability Techniques into the Software Development Process. Proc. *Workshop on Bridging the gaps between software engineering and human-computer intreraction* at *International Conference on Software Engineering (ICSE'03)* (2003)
10. Ferreira, J., Sharp, H., & Robinson, H. User experience design and agile development: managing cooperation through articulation work. *Software: Practice and Experience, 41*(9), 963–974. (2011).
11. Ferreira, J., Sharp, H., & Robinson, H. Values and assumptions shaping agile development and user experience design in practice. *Proc. XP* 2010, LNBIP 48, 178–183
12. Gulliksen, J. Bringing the social perspective: User centred design. In HCI (1) pp. 1327–1331. (1999)
13. Hassenzahl, M., & Tractinsky, N. User experience - a research agenda. *BIT, 25*(2), 91–97. (2006).
14. Highsmith, J., & Cockburn, A. Agile software development: The business of innovation. *Computer, 34*(9), 120–127. (2001).
15. Hodgetts, P. Experiences integrating sophisticated UX design into agile process. *Proc. AGILE* Conference *(2005)*
16. ISO 9241-210:2010. Ergonomics of human-system interaction. Part 210: Human-centered design for interactive systems (2010)
17. Isomursu, M., Sirotkin, A., Voltti, P., & Halonen, M. User experience design goes agile in lean transformation -- A Case Study. Proc *of Agile Conference (AGILE)*, 1–10. (2012)
18. Joshi, A, Sarda, N. Evaluating Relative Contributions of Various HCI Activities to Usability. Proc. *HCSE'10*, 166-181. (2010)
19. Kuusinen, K. The impact of user experience work on cloud software development. *Communications of Cloud Software, 2*(1) (2013)
20. Kuusinen, K., Mikkonen, T., & Pakarinen, S. Agile user experience development in a large software organization: good expertise but limited impact. *Proc. Human-Centered Software Engineering* HCSE'12. Springer Berlin Heidelberg. 94–111. (2012)
21. Kuusinen, K., & Väänänen-Vainio-Mattila, K. How to make agile UX work more efficient: management and sales perspectives. *Proc. 7th Nordic Conference on Human-Computer Interaction: Making Sense through Design (NordiCHI '12).* ACM, 139–148, (2012)
22. Lárusdóttir, M. K., Cajander, Å., & Gulliksen, J. Informal feedback rather than performance measurements–user-centred evaluation in Scrum projects. *Behaviour & Information Technology, 2013*, 1–18 (2013)
23. Lárusdóttir, M. K., Cajander, Å., & Gulliksen, J. The big picture of UX is missing in Scrum projects. *Proc. of International Workshop on the Interplay between User Experience (UX) Evaluation and System Development (I-UxSED).* 49–54. (2012)

24.Lee, J. C. Embracing agile development of usable software systems. *Proc. Conference on Human Factors in Computing Systems CHI*, 1767–1770. ACM. (2006)
25.Leszek, A., & Courage, C. The Doctor is ”In” – Using the Office Hours concept to make limited resources most effective. Proc. *AGILE Conference*, 196–201, (2008)
26.Salah, D., Paige, R., & Cairns, P. A systematic literature review on agile development processes and user centred design integration. In *Proc. of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE'14).* ACM,. A. 5, 10p (2014)
27.Schwaber, K. *Agile project management with Scrum (Microsoft professional)*, 1st. ed., Microsoft Press, (2004)
28.Sy, Desirée. Adapting usability investigations for Agile user-centered design. *Journal of Usability Studies 2*(3), 112–132 (2007)
29.Ungar, J. The Design Studio: Interface design for agile teams. *Proc. AGILE Conference*, 519–524. IEEE Computer Society, (2008)
30.Väänänen-Vainio-Mattila, K., Roto, V., & Hassenzahl, M. Towards practical user experience evaluation methods. EL-C. In *Meaningful Measures: Valid Useful User Experience Measurement* (VUUM): 19–22. (2008).
31.Wale-Kolade A. Y. Integrating usability work into a large inter-organisational agile development project: Tactics developed by usability designers. *Journal of Systems and Software 100*, 54–66. (2015)
32. Williams, L., & Cockburn, A. Agile software development: It's about feedback and change. *Computer, 36*(6), 39–43. (2003).