

Design systems impact on implicit alignment work of UX designers and UI developers

Research study at Aalto University
Espoo, Finland

by

Varya Stepanova

December 2020

1. Introduction

Both User Centered Design (UCD) and Agile Software Development (ASD) are widely used methods in the production of user interfaces (UI). However, they are “frequently putting contradictory demands on people working with the respective processes” (Jones 2019). Trying to match different disciplines, the designers and developers are supposed to conduct alignment work which brings them to mutual understanding in the context of their work, revealing “design breakdowns” (Maudet et al. 2017) early enough.

Design systems are a yet developing concept of UI development, its definition is different depending on the style of organization. (Huang 2019) The researcher considered a design system as “a platform that gathers a continuously evolving set of reusable design and development elements of user interfaces that are in line with a company’s brand identity and UX design principles [with] the elements ... organized systematically and presented along with explanations of how each design choice was made” (Konaté, Aïcha 2018).

Starting to be a new standard in the software creation industry, the design systems play a significant role in the design and development part of this process and influence the collaboration work of a UX designer and a UI developer.

1.1. Problem statement

The academic literature contains a lot of evidence of a significant gap between the design and development processes in software creation. This gap can be explained by many reasons such as different mental models of the specialists, different or sometimes conflicting methods they are using to do their job, and the challenges they meet on the process and organisational level.

1.2. Motivation and scope of the research

Software creation is nowadays a relevant subject to most companies. As more and more (both external and internal) products are becoming digital, the importance of their efficient and effective building and maintenance grows. Providing well-functioning products is currently a responsibility not only from the business perspective but also from the social one. The UX and UI are the product aspects that actually meet the consumers. This is why the importance of their correct delivery cannot be underestimated. The researcher was motivated to find out the factors

helping to overcome the obstacles in delivering good design to the users via its correct development.

1.3. Research outline

The research was meant to view the designer and developer collaboration process through the lens of “design systems”. Preliminary research and the working experience of the researcher promised that there might be changes in the flow of alignment work if design systems are involved.

Research objectives

To understand the impact of the design systems on the alignment work of designer and developer.

Research question

What do the design systems change in the alignment work of designer and developer?

2. Theoretical background

The current research concerns the existing scientific knowledge reflected in the academic literature. To ensure the research relevance and accordance with the already introduced concepts, the researcher conducted an extensive literature review. This section reflects the conceptual picture of the domain represented by the most recent scientific sources.

2.1. Business problem

Many researchers admit the gap between design and development on all the three project-level, process-level, and person-level.

Project-level

On the project level, there is often “the gap between design artifacts and their subsequent implementation” (Maudet et al. 2017) meaning that the released product differs from the initial design idea. Maudet et al. highlighted that among the projects they researched “none of the initial implementations were exactly as the original design” (Maudet et al. 2017).

Process-level

From the working process perspective, even though designers and developers are working on the same project, they are often separated on the organisational level. Developers form project-related teams while designers belong to a design department and contribute to several projects. (Caballero, Moreno, and Seffah 2017; Jones and Thoma 2019) Such separation becomes a business problem since that “causes inefficiency and is perceived to adversely effect quality of output” (Jones 2019). As a result, “current practices induce unnecessary rework and cause discrepancies between the original design and the implementation” (Maudet et al. 2017). The working process suffers from a significant delay in making decisions “because people seem reluctant to make decisions on things” (Jones and Thoma 2019).

Personal-level

On the personal side, the researchers admit the designers’ and developers’ frustration about the working process and outcomes. (Jones 2019) Designers do not get work satisfaction because “the UX work was not central enough in the development process” (Jones 2019). Due to the different angles in work (which are explained in more detail later), the “holistic aspects of the design fail to be implemented correctly causing frustration with the UX designers” (Jones 2019). Jones and Thoma say that both designers and developers “did not feel part of the same team” (Jones and Thoma 2019) and that indeed causes motivational and efficiency problems. Integrating the designers’ and developers’ work “has proven difficult, often leading to friction between them” (Maudet et al. 2017). This issue does not help to keep the team spirit and definitely negatively affects the results.

2.2. Mental models of designers and developers and their reflection in the working process

The conceptual differences between UCD and ASD raise a lot of challenges for the designers and developers when working on the same project. The UCD approach is “designing for the whole user experience” (Jones and Thoma 2019) while by ASD “work is divided up into chunks and smaller releases” (Jones and Thoma 2019). This difference makes it difficult to combine two approaches and also nourishes different mentality of designers and developers. Working as a designer and working as a developer requires different skills and “focus on different aspects of the design process” (Maudet et al. 2017).

Designers and developers demonstrate different ways of thinking and perceiving the working context. The literature mentioned that “designers usually take a more holistic view on the interaction design and information architecture of a website or product” (Jones, Thoma, and Newell 2016) while the developers tend to modularize the products they are building. For the designers, the iterative approach of ASD “may be averse to their way of thinking and working” (Jones et al. 2016). From the development perspective, ASD approach requires to work chunk by chunk but this is “making it difficult to consider the holistic design at all times” (Jones and Thoma 2019).

Designers

Holistic view (or “a big picture”)

The literature describes the work of designers as “very holistic” (Jones 2019) and focused on “a page (URL) level” (Jones 2019).

Designers tend to focus on a big picture and, for example, “omit important edge cases from their design documents” (Maudet et al. 2017). Often that emerges only on the development phase and creates a design breakdown which leads to either re-creating designs by the designers in a hurry or to the developers providing their changes which are not aligned with the designers.

Flat mock-ups

In particular, the big-picture approach results in flat mock-ups as the artifacts to represent the design vision. Such mock-ups communicate the “worked-out graphic elements (fonts, headers, etc.)” (Jones 2019) but do not take into account the “necessary details about the interaction layer in-browser” (Jones 2019). Flat drawings “were effective in communicating the overall idea, but left too many unanswered questions for correct implementation” (Maudet et al. 2017).

The “page-level” mentality also creates obstacles for “taking into account the dynamic nature of the data” (Maudet et al. 2017).

Lack of technical skills

Often mentioned that designers may lack technical skills and experience that makes it hard or impossible to foresee design breakdowns for future designs. (Jones and Thoma 2019; Maudet et al. 2017; Walny et al. 2020) This “leads to uncertainty about design feasibility and can also trigger more dramatic revisions during development” (Walny et al. 2020). Similar is highlighted by Jones in his thesis where he mentions that “designers do not know enough about the

technical limitations [...] [and] developers perceived inefficient and redundant work efforts at times during the ASD process” (Jones 2019).

Developers

Focus on details

Unlike designers, the developers tend to focus on separate interface elements rather than on the page level. (Jones 2019)

Modularisation

Working with flat mock-ups which meant to communicate the design idea, a developer “would [...] laboriously un-pick various aspects of the design to understand how the implementation in the browser would either look or behave during different circumstances” (Jones 2019). In other words, a developer modularises the holistic design idea to align it with their domain.

Looking to modularise the interfaces, the developers tend to rely more on the text descriptions rather than on their visual representations even though such descriptions “did not clearly convey the look and feel of the interaction to the designers” (Maudet et al. 2017).

Lack of early feedback

The literature often points out that developers’ satisfaction with the work process correlates with the intensity of collaboration with the designers. (Jones and Thoma 2019) In particular, developers value to be co-located with the designers as this makes it easier to get the missing information when a design breakdown occurs during the implementation phase. This signals the frequency of the design breakdowns and their heavy influence on the work process, which in turn means testifies lack of co-creation or even providing feedback early enough.

Designers and developers working together

Overall, the papers mention regarding designers and developers representation in the current ASD process “the two disciplines were disjointed in their work and processes, which leads to inefficiencies” (Jones 2019).

The outlined differences in working practices of designers and developers make it challenging to build a multidisciplinary team that follows a specific workflow. Thus, the designers’ “holistic view on the interaction design and information architecture” (Jones et al. 2016) which was highlighted by many papers conflicts with “modularization and iteratively adding features” (Jones et al.

2016) — the attributes of the ASD process reflected in developers' ways of working. The design work is often still not fully embedded in the iterative ASD process but follows the outdated "waterfall" practices (Jones and Thoma 2019).

On the other hand, the component-level focus in the development results in the fact that "more holistic aspects of the design failed to be implemented correctly" (Jones 2019).

The literature mentions the positive effect of cross-functional work resulting in adopting practices from specialists of the other area. Thus, Maudet et al. tell about a designer who started to practice a modular approach for her designs "inspired by the developers' way of working" (Maudet et al. 2017).

2.3. Typical workflow

Describing the current workflow in multidisciplinary UX/UI teams, the researchers characterised the current ASD process and "not iterative enough" (Jones 2019). The disciplines of UX designer and UI developers are "disjointed in their work and processes, which leads to inefficiencies" (Jones 2019).

Trying to overcome these differences, both designers and developers experience a need to commit extra work. Thus, the previous studies mentioned that the developers spend a "surprising" amount of time to recreate the design documents (Maudet et al. 2017).

The literature mentions that "much of the interaction time between these roles was used to re-align individual work progress to ensure a common understanding of the project aims" (Jones et al. 2016).

At the same time, there is a common understanding that UX/UI projects lack early communication between designers and developers, and that finds support in the scientific literature (Jones and Thoma 2019).

Dual-track agile

Looking for a general solution responding to the problem of miscommunicated design and development, many companies nowadays are using a *dual-track agile workflow*. This workflow has many visualisations and representations by both researchers and practitioners. The one illustrated in Figure 1 shows how dual-track agile merges the design thinking and ASD processes. The method proposes a shared prototyping phase which is meant "to transform the ideas into a tangible form to experience and interact with the proposed solutions" (Yalcinkaya 2017). This approach complements the results of research by Jones and Thoma who mentioned

“co-creation and prototyping” as one of the factors for “successful collaboration between designers and developers” (Jones and Thoma 2019).

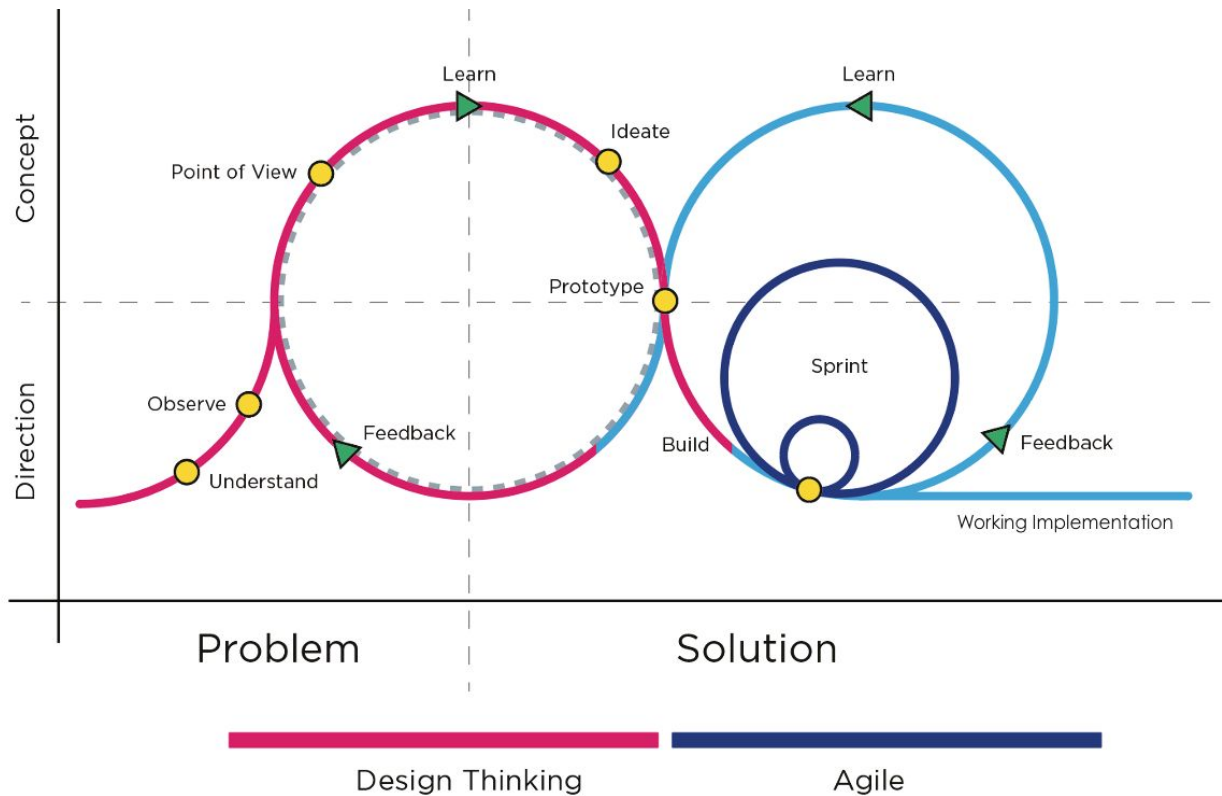


Figure 1. Applied design thinking and agile development methodology. (Yalcinkaya 2017)

Design breakdowns

Existing studies introduce a concept of design breakdown as “an impediment that must be fixed before the design can be implemented” (Maudet et al. 2017). Design breakdowns can be caused by several reasons such as 1) missing information, 2) edge cases, and 3) technical constraints. Each of those leads to a situation that design cannot be implemented as it was planned. Describing the methods to overcome design breakdowns, most of the literature mentions co-creation sessions as the most effective approach (Jones 2019; Jones and Thoma 2019; Maudet et al. 2017).

Implicit joint alignment work

The literature describes the co-creation process of designers and developers as “implicit joint alignment work” (Brown, Lindgaard, and Biddle 2012). Brown et al. studied the alignment meetings and offered a model of implicit joining alignment work, illustrated in Figure 2.

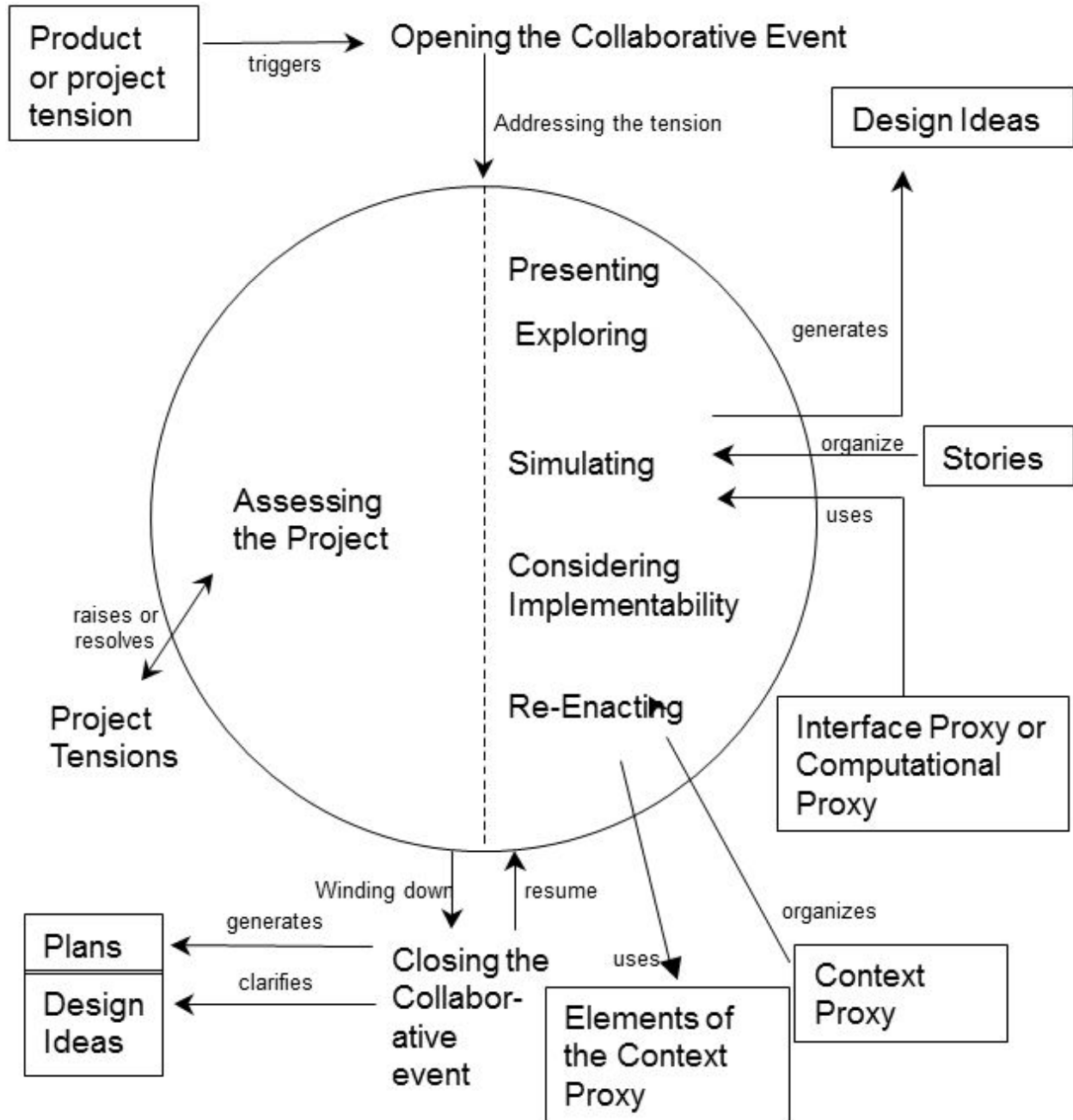


Figure 2. A model of implicit joint alignment work involving designers and developers depicting the concrete nature of commonly occurring collaborative events. (Brown et al. 2012)

2.4. Possible improvements to the workflow

Even though the mentioned literature provides some suggestions to improve the current workflow of mutual UX design and UI development process, they don't look to be extensive and game-changing. Most of the suggestions operate in the area of co-locating the designers and

developers (Jones 2019; Jones and Thoma 2019), “providing space and opportunity for collaborative meetings” (Brown et al. 2012). Despite being challenging for a remote work situation, these suggestions also do not offer any change to the core of the working process. Searching for factors deeply affecting the workflow must be a large space for potential constructive research.

3. Design of the research

The current research was designed to fulfill the gap in the existing knowledge about the implicit alignment work of UX designers and UI developers. Admitting the fact that this gap is significantly large, the research only concentrates on one aspect — the role of design systems (as an artefact and the methodological approach) in the alignment work. The research is meant to deliver a contribution to both existing knowledge and a practical aspect of UI software creation.

3.1. Data collection

The research data is obtained from intensive interviews with the professionals in the related area. The choice to go with intensive interviewing is motivated by the knowledge that it “permits an in-depth exploration of a particular topic”, which makes it “a useful method for interpretive inquiry” (Charmaz 2006). The research participants represent both groups — UX designers and UI developers. Because the design systems are only becoming an industry standard, it is very challenging to get enough participants from one company only. However, the shapes of the projects where design systems are currently involved and ways of working in the teams of these projects are quite standardized. This makes it possible to interview professionals from different companies and even countries.

To respond to the final goal of the research, the interviews were designed to be semi-structured meaning that the exact flow of the discussion should adjust to the context. The decision to go with interviews of this kind was motivated by the knowledge that more structured interviews tend to produce more quantitative than qualitative data (DiCicco-Bloom and Crabtree 2006). Less structured interviews, at the same time, seem to be less informal and more comfortable for the interviewees.

The structured part of semi-structured interviews is kept by following the interview guideline. The interview guideline provided separated question tracks for UX designers and UI developers but

kept both paths to be responding to each other. This approach ensured more relevant content with topic intersections when both designers and developers share their opinion about the same topics but from different perspectives. The baseline questions in the interview guideline were grouped into three categories: 1) Personal and context, 2) Alignment work of designers and developers, 3) The role of design systems. The full content of the interview guideline can be found in Appendix 1.

3.2. Data analysis

Grounded theory

The obtained data is coded and processed with the methods suggested by the *grounded theory*. The process of building the theory following these methods is illustrated in Figure 3. In the process of *codifying* the rich data and categorizing the codes into *themes*, emerges the explanatory model based on the revealed connections between the found out themes. Aligned with the current knowledge base obtained from the preliminary literature review, this explanatory model forms a theory.

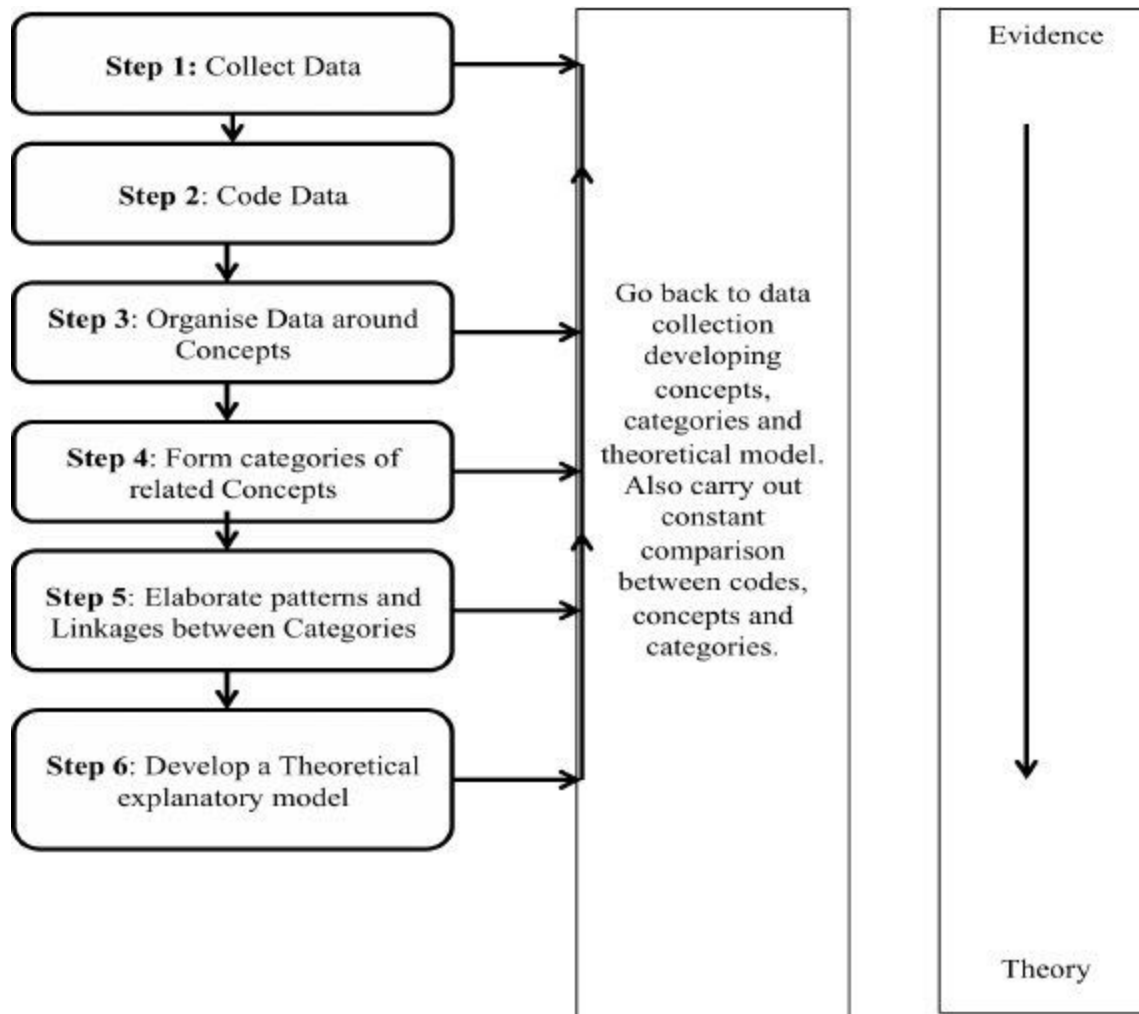


Figure 3. Grounded Theory Data Analysis Steps. (O'Hagan and O'Connor 2015)

Practically, the codifying process was conducted in two steps. In the pre-phase, the recorded interviews were transcribed and cleaned from nonrelevant (mostly colloquial) content pieces. In step 1, the researcher selected inspiring quotes and assigned them to codes using a *line-by-line coding* approach. One code could be assigned to several quotes across the whole interview database. The frequency of the code appearance signaled the importance of the topic. In step 2, the researcher categorised the codes into themes based on their conceptual similarity. Since the data was obtained from people representing different organizations and different professional areas, the researcher was looking for a method that allowed to process all the interviews together but at the same time keep the data separate based on their source. To achieve this controversial requirement, the categorisation was conducted on a virtual whiteboard using the *color-coding technique*.

Color-coded theme grouping

The codes of each interview were exported into a virtual whiteboard as sticky notes. The codes of each participant were assigned a unique color but the designers always got light-colored sticky notes with black text on them while the developers always got a dark-colored sticky note with white text on it.

The codes were grouped into themes based on the conceptual similarity of the codes or belonging to the same topic. The themes and the codes they got containing were visually presented on the titled whiteboard sections.



Figure 4. Color-coded codes grouped into themes.

The color-coded grouping made it possible to process the data of all the participants together but keep its connection to who was the source of this data and which side (design or development) he or she represented.

In case the theme mostly emerged from the data of a single one interviewee, it should be visible as only one color of sticky notes with codes would be presented. However, in the current research, such a situation never occurred that confirms the assumption about the high relevance of the data obtained from so different interviewees.

Assigning light-colored notes to designers and dark-colored notes to developers made it visible if any theme emerges in the data from one side only. That was itself a very valuable theme characteristic as the research was focused on finding collaboration and communication issues between these two groups.

Additionally, the visual representation of the themes made it noticeable if any theme was getting too large. In this case, the researcher took additional iteration for categorising such a theme into smaller concepts.

3.3. Building the theoretical framework

Building the theoretical framework onto the emerged themes was conducted in three phases. Phase 1 was meant to narrow down the scope by omitting not relevant themes. As the interviews were very fruitful to share about all the aspects of design and development processes, focusing on the relevant themes was needed to keep the research on track towards finding the impact which was made specifically by the design systems. In phase 2, the researcher organised the selected themes into a system based on relationships between them. Since the grouping work was conducted with a virtual whiteboard and sticky notes, the representation of the relationships was also formed there naturally providing its visual version. The result of phase 2 was a schematic description of both positive and negative design system impact elements. Lately, the researcher translated this representation into a scheme presented in the “Results of research”. During phase 3, the researcher was looking for an application of the emerged impact elements to the existing theoretical knowledge. The visual model of the impact made it possible to place its elements onto visual representations of the processes where design systems are used as artefacts (or “working tools”). These combined visual models represent the role of different impact elements as well as illustrate the role of design systems to the whole product creation process.

4. Results of research

There were 10 30-minute interviews with UX designers and UI developers. For further analysis, there were selected 6 the most substantive and relevant interviews: 3 UX designers and 3 UI developers.

4.1. Emerged themes

The gathered data contained a lot of information related to the design and development process on the general level, including the impact of organization structure and characteristics of broken or well-managed co-creation processes, this research concentrates on the role of design systems.

4.1.1. Problem confirmation

The focus of the research was the role of design systems to the process of designer and developer collaboration. Keeping this focus, the researcher excluded the themes which were not relevant to the research question. However, the two themes which were not related to the design system confirm and explain the core differences between designers' and developers' working approach and the nature of design breakdowns occurring in the working process. In this regard, these themes were also considered relevant to the research topic.

Theme "Holistic vs modular"

Included codes: thinking in unique screens, need for decomposition, screen changes result into changes in components, screen thinking leads to inconsistency.

The developers highlighted that one of the main working problems for them was that "designers are thinking in screens, they are not used to a systematic approach and every screen is unique for them". For a developer, it results in a need to "decompose" the designs on every iteration and translate the change to the components. The developers claim that this process leads to UI inconsistencies. At the same time, a designer who is already practicing a systematic approach in her designs noted that it was "no problem at all" meaning that such a way of working is very efficient. Thus, the research data confirms the differences in mental models of designers and developers noted in the literature (Jones 2019; Maudet et al. 2017).

Theme "The nature of design breakdowns"

Both designers and developers described the design breakdowns emerged in their work confirming the picture given in the literature (Maudet et al. 2017). All the interviewees mentioned that design breakdowns will always be and there is no technical or methodological way to decrease them down to zero. As one of the interviewee said, "often, the technical constraints cannot be seen in advance". However, there is a general understanding that "the bigger the task the more questions will arise". That bridges to the modular approach practised in design systems and foresees its positive influence on the amount and deepeness of the design breakdowns.

4.1.2. Characteristics of design systems

Theme “Ready-made components”

Included codes: ready-made components, reusing the components, UI work already done, easy-to-manage small chunks.

All the research participants admitted that design systems provide the project teams with ready-made components to be re-used in the projects on both design and development phases.

The components were described as “small chunks which are easier to process”, e.g. one designer noted that “this really gives us the flexibility to reevaluate the priority”.

The designers shared that ready-made components decrease the need in low-level UI work as it has already been done on the design system level.

Both designers and developers mentioned that without shared components the developers would need to create everything from scratch.

Theme “Increasing productivity of the team”

Included codes: helping productivity, easy-to-evaluate small chunks, hard-to-manage bigger chunks, less breakdowns, decreasing low UI work.

Everyone admitted that using a design system increases productivity. Designers mentioned that having a design system, they are mostly “concentrating on user flows” and have the possibility to pay more time to the UX rather than UI part of their job. Developers say that using predefined components takes nearly zero time while developing them from scratch would take days or even weeks.

Not only re-usage improves the productivity but

Theme “Improving the quality of the projects”

Included codes: increasing the UI consistency, concentrating on user flows.

Both designers and developers highlighted that using a design system improves the UI consistency across the projects. Designers were saying that “low level UI work is done” that gives much more time “to do the actual job”.

Theme “Easing communication in the team”

Included codes: providing extra information, shared language.

Describing how their in-team communication has changed, the designers mentioned that “there is no handover anymore” as their design system itself provides extra design information about features to be implemented. Developers were saying about developing a shared language for the different parts of the project or UI elements in particular. This language was named as a factor to smooth later in-team communication.

Theme “Providing too much restrictions”

Included codes: dictating technical solutions, outdated solutions behind the design system, strict policy takes developers’ time, writing code in a specific manner, ought to use the design system.

Some developers shared their frustrations about their design system limiting them too much. In particular, they mentioned that “design dictates how to develop” or that they “have to write code in a specific manner which would not be a choice if there is no design system”. If a design system is obligatory to be used in the company, it may require more time to develop the project. However, there should be admitted that such problems were mentioned in relation to the outdated technical solutions behind a specific design system.

Theme “Requiring investments”

Included codes: early cooperation requires resources, need for alignment on changing the components, design system meetings, design system ambassador, responsibility to give feedback, need for documentation, complex long-lasting project, personal connection, contribution requires alignment with other teams, need to learn the system.

The research participants had shared the opinion that using a design system should be considered as investment which takes resources now but pays later. Design systems were given characteristics of “complex long-lasting projects” which often “require a separate team”. The participants mentioned the time-consuming working process around design systems including “weekly design systems meetings” and separate communication channels and processes to organise the contributions into a design system. One participant described the role of a *design system ambassador* — a “dedicated representative from a design system team which helps project teams to use their design system in a correct manner”.

One of the interviewees raised a need of changing the company working culture when using design systems. In particular, he mentioned that in design systems regard, “a developer has responsibility to give up-to-date feedback” however “not all the developers understand this need”. Providing such a change on the company culture would itself require a lot of time, resources and organisational work.

Theme “Decreasing the quality of the projects”

Included codes: lack of components, need for documentation, outdated documentation, low quality of components, design system bugs affect the projects, changes affect the projects, bad design system cannot evolve.

In some cases, design systems can decrease the quality of the projects they are being used in. The interviewees highlighted that since the design system’s code is working inside all the other projects, “every change to a design system affects the projects”. This fact is especially sensitive if a design system has some bugs.

Theme “Complicating the process”

Included codes: UI kit differs from the components, what is the source of truth, outdated design system decreases interest to the project, contribution takes time, limits decrease involvement into the projects, frustration about the system being obligatory.

As an additional element of the company project structure, a design system usually complicates the development process. A design system should be well-managed, otherwise there might be situations when different parts of it are unsynchronised (“the UI kit used by the designers differs from the technical implementation of the component”) and “it is not clear what is a source of truth”. One of the developers shared frustration about their outdated design system that “decreases interest in the project”, and it was mentioned that “contribution to the design system takes time”. Depending on the overall quality of a design system, an obligation to use it can lead to a frustration about the organisation in general.

Theme “Gaining shared knowledge”

Included codes: team shares same data domain (components), decreasing design breakdowns, components are the reference, designers gain project knowledge, more structure to the design process, developers are more happy.

The most significant insight out of all the interviews was that practicing a design system approach, the team members operate in the same data domain and constantly gain shared knowledge about their own project. This shared knowledge was pronounced as a factor which “creates a feeling what can be implementable and what cannot”, meaning that shared knowledge decreases the amount of arised design breakdowns. Designers mentioned that their design process had become more structured and clear. Developers shared that they are more happy about their work when they are using a proper design system.

4.2. Design systems impact scheme

Together, the themes emerged in the research form the framework which visually represents both the positive and negative impact of using a design system in the project work. Presented in Figure 5, the *design system impact scheme* organises the emerged themes as a dual-track keeps some themes opposite to the others. The central line illustrates a not trivial process of keeping the balance between positive and negative impact elements. This balance should ensure the long life of the project during which the team gains shared knowledge by using the design system as an artifact and co-existing in the same data domain.

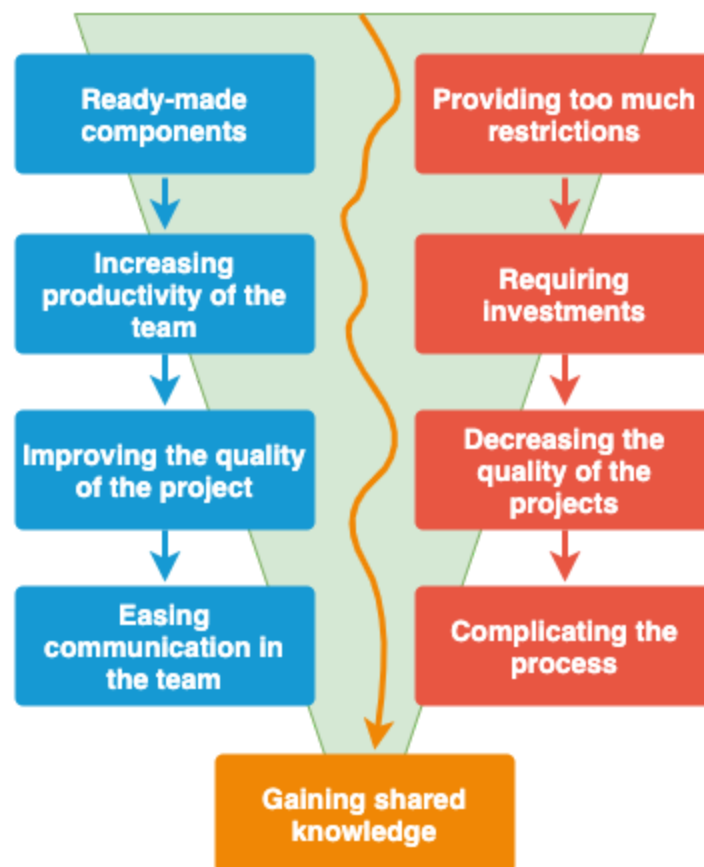


Figure 5. Design system impact scheme. Visual representation of the emerged themes.

4.3. Contribution to the knowledge

4.3.1. Design systems impact on the implicit joint alignment work

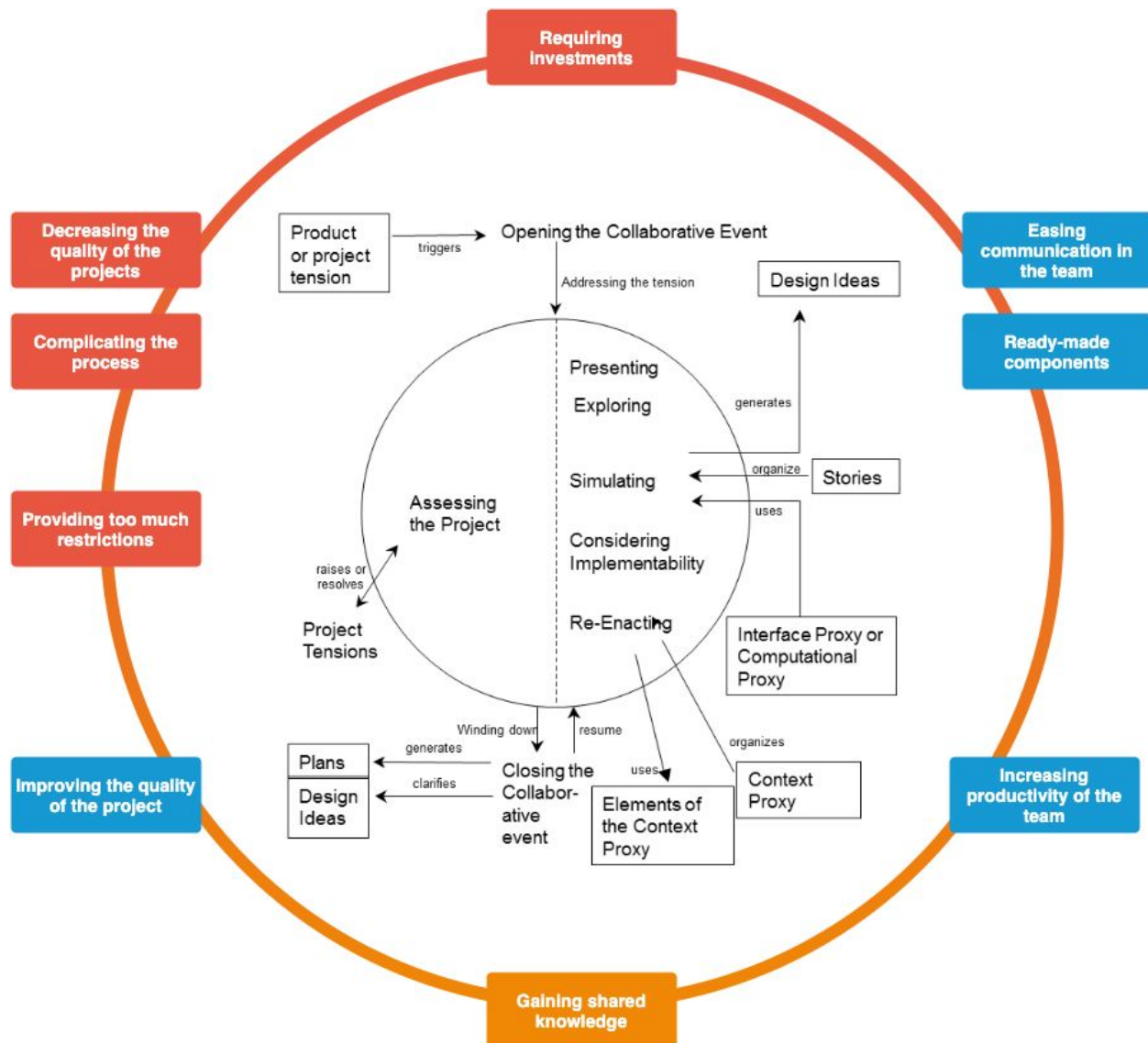


Figure 6. Design system impact into implicit joint alignment work.

Applied to the model of an implicit joining alignment work, the emerged themes illustrate the impact of the design systems on this process. Thus, negative impact elements create more project tensions in both pre-phase and in the process of alignment work. At the same time, the

positive impact elements facilitate the co-creation and speed-up exploring, simulating, and considering implementability by decreasing the need for low-level UI work. The fact that building and maintaining the design system requires additional time and resources is illustrated as a general process characteristic along with the main positive outcome — gaining the shared knowledge. Visual representation of such an application can be seen in Figure 6.

4.3.2. Design systems impact on the dual-track agile

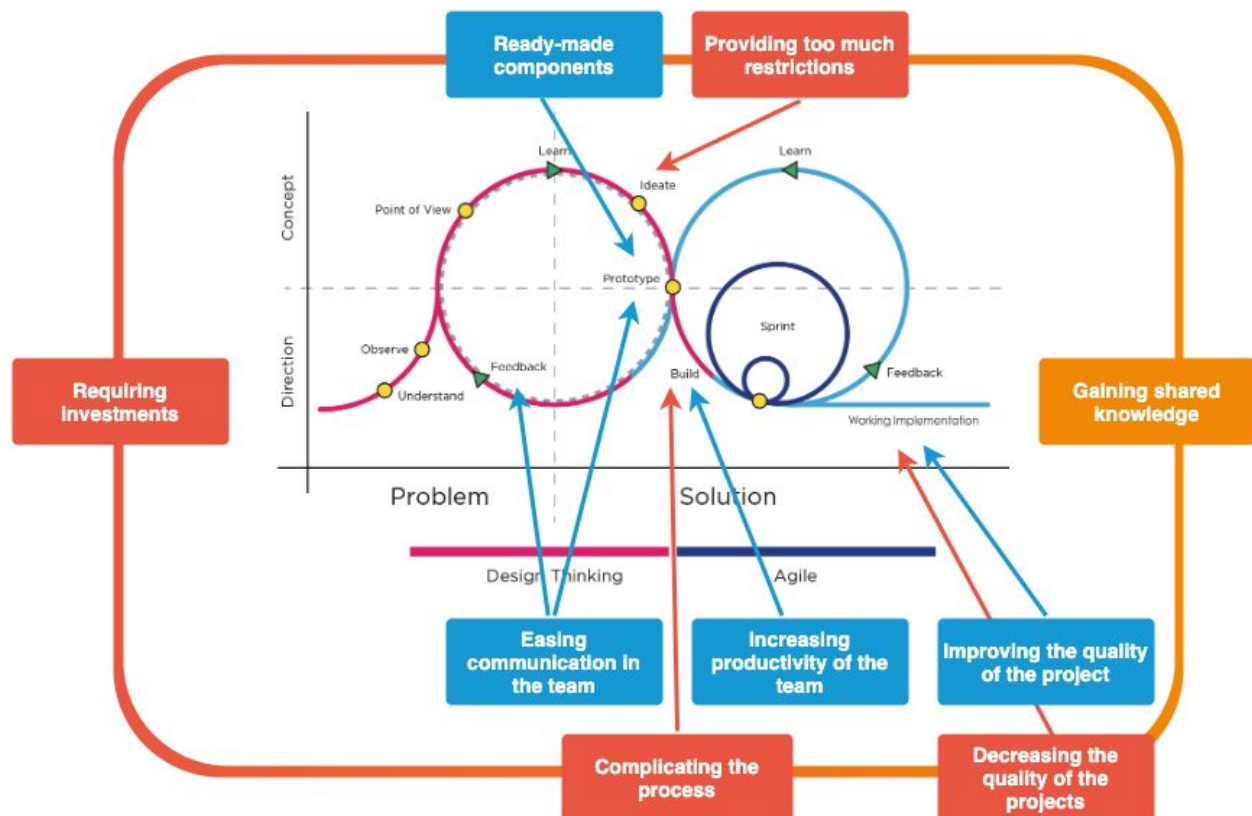


Figure 7. Design system impact to dual-track agile process.

Dual-track agile process experiences both positive and negative impact elements of the design systems. Figure 7 represents the design system's impact on a dual-track agile process by applying the emerged themes to a visual representation of the dual-track agile process positioning the impact elements following their relation to the process elements. Using a design system as an artefact should speed up the co-creative prototyping but may provide too heavy restrictions that, in turn, limit creativity. However, in-team communications made easier should positively affect the co-creating process as well as improve the quality of feedback. The fact that having a design system is a business investment again stays as a general process

characteristic along with gaining shared knowledge. These two elements stay permanent throughout all the development cycle iterations, or throughout the whole project lifecycle.

4.4. Discussion

4.4.1. Research findings

During the current research, the business problem outlined in the Theoretical background section was confirmed. The research participants witnessed that The workflows of UX designer and UI developer are very much unsynchronised and that affects all the three levels of work: project- and process- related, and personal.

The research was focused on the impact which design systems bring to the alignment work of designer and developer. Accordingly, the main finding of the research is the Design systems impact scheme ([4.2. Design systems impact scheme](#)). The scheme illustrates that as a methodological approach, design systems can both positively and negatively affect the software creation process and its results. Looking for a balance between the positive and negative tracks should be the main focus of the specialists practicing design systems in their work. The design systems impact scheme highlights that besides being a helpful artifact and tool for the working process, design systems also contribute in the long run helping the multidisciplinary team gain a shared understanding of the project. In turn, this shared understanding simplifies the design and development process in the following iterations decreasing the number of design breakdowns on the go.

The results of the research contribute to the existing knowledge by fulfilling the known software creation workflows with the design systems' contribution to them. The visualisation ([4.3.1. Design systems impact on the implicit joint alignment work](#)) illustrates how impact elements of the design systems affect different aspects of the alignment work. While creating additional project tensions and requiring time and resource investments, a design system helps as the following:

- A design system eases communication in the team and takes the “thinking” phase of the co-creation process smoother. With improved communications, design ideas emerge easier. The acceptance of good design ideas by the other team members gets more probable. At the same time, if the presented design idea needs improvements, the feedback about it arises more willingly. Overall, the phase of generating design ideas and validating them turns to be more fruitful with increased quality of its output.

- A design system offers ready-made components to form future designs out of them. Using these components decreases the amount of low-level UI work and speeds up the prototyping process. In the alignment work session, the ready-made components accelerate the simulating phase. The next phase, when the multidisciplinary team considers implementability is mitigated by the fact that a lot of pre-work has been done on the design-system level.
- A design system increases the productivity of the team positively affecting the software creating process in general and the alignment sessions in particular.
- Highlighted by the research participants and confirmed with the analysis, practicing the design systems approach improves the quality of the project from both the design and technical perspective. In the interviews, the designers and developers were mentioning that “design systems give me to do the actual job” reflecting that the low-level UI work and guarding consistency is not the final purpose of what they are doing but a secondary task which often takes too much time.

Scaled up to the picture of the whole software creation process ([4.3.2. Design systems impact on the dual-track agile](#)) design systems impact elements affect there as well.

- The “design thinking” subtrack experiences positive tensions of eased communication and the fact of having ready-made components that designers could use for creating future designs.
- However, it must be admitted (and was mentioned in the research data) that a design system with a strict usage policy could negatively affect creativity and that could result in a lower quality of proposed design solutions.
- In the “agile” subtrack, a design system improves the overall quality of the project. This is a positive factor for a holistic user experience and for the maintainability of the project. On the business level, this improvement results in potentially higher revenue for the project and lower maintenance costs.
- The mutual track part when the design thinking process meets the agile software creation process is also experiencing the influence of design systems impact elements. Eased communication and improved productivity positively affect the alignment workflow and it was described above.
- The negative impact can be generalised to the statement that a design system is a resource-taking investment that should pay back on all the future iterations of the software creation process but require time and funding in the beginning.

4.4.2. Application of results

Understanding the impact of design systems into the process of alignment work of UX designer and UI developer should have the following application:

1. Help further organizing multidisciplinary work in the UI production field.
2. Outline the means of design systems in this process.
3. Show which attributes of design systems are most helpful.
4. Provide context for future creating of the complementary tools for the design systems, documenting designs, and UI elements.

4.4.3. Limitations of the research

The research was only focused on the design systems' impact on the joining implicit alignment work of a UX designer and UI developer in a multidisciplinary team. However, such a complex and not formalised process must experience a lot of other positive and negative tensions on all levels (organizational, professional, social, etc.).

The research data set could be more relevant if there are interviewees from the same large organisation but from different teams that are using a design system or not. However, it must be admitted that finding such a representative group is a very challenging task itself.

4.4.4. Future studies

Raising the current research question and the topic of design systems, in general, opens a lot of space for future studies.

1. Considering the processes of design and development separately, there can be more detailed impact findings.
2. As mentioned earlier, the design systems are a yet-evolving concept and it has different understanding in different organisations and sometimes even across the same organisation. Studying the specific shapes of design systems could bring additional insights into the perspective of alignment work and in general.
3. Alignment workflow is a nontrivial and not structured process that must experience an effect of many other factors but design systems. Another direction for future studies could be exploring the other factors and studying how they work in the process.

5. Conslution

This paper describes the research which raised the question “What do the design systems change in the alignment work of designer and developer?”. As it was shown, design and development are often two separate processes due to many reasons. At the same time, they must be aligned with each other as both contribute to the core of software creation. At the moment, the industry experiences a lot of challenges due to the difficulties of this alignment process.

The design systems as a methodological approach and artefacts play a significant role in the work alignment process. This research results in the structured and categorised representation of design systems' impact on the working process. The illustration of the impact is detailed to specific elements. Such a representation demonstrates how these elements have a (mostly positive) effect on different aspects of the software creation process. In a nutshell, the teams which are practicing the design systems, ease the multidisciplinary communication but they have to be careful not to complicate their process too much. The overall quality of the project increases if the design system policy is flexible. A design system requires time and resource investments but in the long run, it improves the productivity of the team (on both design and development sides). The ready-made components offered by a design system speed up the product creation process as well as generating new design ideas and meeting the customer and business needs.

References

- Brown, Judith M., Gitte Lindgaard, and Robert Biddle. 2012. "Joint Implicit Alignment Work of Interaction Designers and Software Developers." P. 693 in *Proceedings of the 7th Nordic Conference on Human-Computer Interaction Making Sense Through Design - NordiCHI '12*. Copenhagen, Denmark: ACM Press.
- Caballero, Leydi, Ana M. Moreno, and Ahmed Seffah. 2017. "How Agile Developers Integrate User-Centered Design into Their Processes: A Literature Review." *International Journal of Software Engineering and Knowledge Engineering* 26(8):1175–1201.
- Charmaz, Kathy. 2006. *Constructing Grounded Theory*. London ; Thousand Oaks, Calif: Sage Publications.
- DiCicco-Bloom, Barbara, and Benjamin F. Crabtree. 2006. "The Qualitative Research Interview." *Medical Education* 40(4):314–21. doi: 10.1111/j.1365-2929.2006.02418.x.
- Huang, Yun-Hsuan. 2019. "Developing a Design System for an E-Commerce Website." Master thesis.
- Jones, Alexander J. 2019. "Factors for Successful Agile Collaboration between UX Designers and Software Developers in a Complex Organisation." Doctoral dissertation, University of Chester, United Kingdom.
- Jones, Alexander, and Volker Thoma. 2019. "Determinants for Successful Agile Collaboration between UX Designers and Software Developers in a Complex Organisation." *International Journal of Human-Computer Interaction* 35(20):1914–35. doi: 10.1080/10447318.2019.1587856.
- Jones, Alexander, Volker Thoma, and Gemma Newell. 2016. "Collaboration Constraints for Designers and Developers in an Agile Environment."
- Konaté, Aïcha. 2018. "Design Systems at Work." Master thesis.
- Maudet, Nolwenn, Germán Leiva, Michel Beaudouin-Lafon, and Wendy Mackay. 2017. "Design Breakdowns: Designer-Developer Gaps in Representing and Interpreting Interactive Systems." Pp. 630–41 in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. Portland Oregon USA: ACM.
- O'Hagan, Ann Osborne, and Rory V. O'Connor. 2015. "Towards an Understanding of Game Software Development Processes: A Case Study."
- Walny, Jagoda, Christian Frisson, Mieka West, Doris Kosminsky, Soren Knudsen, Sheelagh Carpendale, and Wesley Willett. 2020. "Data Changes Everything: Challenges and Opportunities in Data Visualization Design Handoff." *IEEE Transactions on Visualization and Computer Graphics* 26(1):12–22. doi: 10.1109/TVCG.2019.2934538.
- Yalcinkaya, Mehmet. 2017. "Understanding the Technical and Cognitive Challenges, and Closing the Gaps in Architectural, Engineering, Construction-Facility Management Standards." Doctoral dissertation, Aalto University, Helsinki.

Appendix 1. Interview guideline

designers	developers
Personal and context	
<ul style="list-style-type: none"> Describe your work profile. What is your role in the company and/or at the current project? 	
Alignment work of designers and developers	
<ul style="list-style-type: none"> What is the shape of your team? What meetings do you have during the day? What tools do you use for them? Please describe how you start working on a new thing (new page, new interface). Please describe the process of communicating your design to a developer. What tools do you use? What do you align the new designs with? Website/mockups/design system? What happens if developers would like you to re-do something? How do you keep the brand and UI consistent? What are the problems you faced when communicating with developers? Please draw a scheme of alignment work with a designer. 	<ul style="list-style-type: none"> What is the shape of your team? What meetings do you have during the day? What tools do you use for them? Please describe how you start working on a new feature. How does a designer communicate their designs to you? What tools are involved? Do you need to translate the design to your own domain before implementing? What happens if you face a breakdown (something cannot be implemented, etc)? How do you keep UI consistency? What are the problems you faced when communicating with Please draw a scheme of alignment work with a designer.
The role of design systems	
<ul style="list-style-type: none"> Does your company have a design system? What exactly is it? What is a single source of truth - design or development of a design system? What was the reason why you started to build a design system? What was the difference in this design communicating process in the times when your company did not have a design system? What and how do you do if you need to change the existing patterns in your design system? Do you have pain points regarding your design system? What is your vision of the design system in your company? What changes it needs? 	

