



**LAUREA**

UNIVERSITY OF APPLIED SCIENCES

*Together we are stronger*

# A design handoff guide for software development

Muhammad Imran

2018 Laurea



Laurea University of Applied Sciences

## **A design handoff guide for software development**

Muhammad Imran  
Degree Programme in Business In-  
formation Technology  
Bachelor's Thesis  
December, 2018

Muhammad Imran

### A design handoff guide for software development

2018

Pages

45

---

In order to build a consistent and aesthetically pleasing user interface, communication between developers and designers is vital. This thesis project utilizes action research to show how a design handoff checklist can improve the communication between developers and designers and therefore the efficiency of the entire development process. This required the creation of a design hand off checklist that was then implemented to the development team to see whether it made the development process more efficient. The experiment was conducted in collaboration with a Finnish software company developing a survey tool. Due to confidentiality the company name is not mentioned in the thesis.

Because the handoff process is a matter of experience, this thesis project used an action research approach in order to experiment with the design handoff checklist. Research was conducted through observation of the handoff process in the company both before and after the implementation of the handoff checklist along with interviews with both parties. The interviews conducted before implementation helped shape the final checklist supported by a literature survey of practices utilized elsewhere. The interviews that followed the implementation of the checklist on the other hand measured the experience of both designers and developers in order to assess whether the checklist succeeded in improving the development process.

The design handoff checklist was extremely successful. Interview results showed that both designers and developers benefited from the use of the checklist. The development process became more efficient as communication was improved and repetition of work was decreased. 89% of the development team believed that the checklist made their work easier. The designers complained about the extra work caused by the careful planning of the handoff document, but 50% of the design team still agreed that with this extra effort, work didn't need to be repeated and designs came out as originally intended.

The thesis contains six parts. First, the company and ongoing project along with the thesis objectives are laid out in the introduction. The second chapter gives an overview of theoretical background in the matter of design handoff. The third chapter explains the research methods utilized in the thesis project and the fourth chapter presents how the checklist came together based on the literature survey and research within the company. The fifth chapter then presents the results of the experiment by analyzing interviews conducted after the implementation of the design handoff checklist. Finally, the sixth chapter concludes the thesis.

Keywords: Design handoff checklist, Collaboration, User interface, Design Specifications

## Table of Contents

1	Introduction .....	7
1.1	Thesis background .....	7
1.2	Thesis objectives.....	7
1.3	Thesis structure .....	7
1.4	Introduction to company and project.....	8
1.5	Limitations .....	8
2	Theoretical background .....	9
2.1	The importance of design handoff .....	9
2.2	What is design handoff or design specifications? .....	9
2.2.1	A design handoff checklist makes communication faster and smoother .....	10
2.2.2	Keeps designs up to date .....	10
2.3	Successful IT companies' approach to design handoff .....	10
2.3.1	Understanding a Design system .....	10
2.3.2	Understanding Style guides .....	12
2.3.3	Design systems, style guides, what's the difference?.....	13
2.4	Design handoff and product development process .....	14
2.5	Understanding developers and designers.....	14
2.6	How to improve collaboration between designers and developers .	15
2.6.1	Start collaborating from the start .....	15
2.6.2	Constantly keep communicating with each other during design process.....	15
2.6.3	Designing with developers in mind .....	16
2.6.4	Learn basic programming languages like HTML and CSS.....	16
3	Research method .....	16
3.1	Why this method of research?.....	18
4	Implementation and verification of design handoff guide .....	18
4.1	User stories .....	18
4.2	Site maps.....	19
4.3	Documenting variables, constants and properties .....	20

4.4	Copy.....	20
4.5	Naming conventions .....	22
4.6	Interactions flow .....	23
4.7	Prototypes.....	24
4.8	User interface components.....	24
4.9	States for components .....	26
4.10	Grids .....	27
	4.10.1 Keeps your content organised .....	28
	4.10.2 Enhances visual hierarchy .....	28
4.11	Margins and paddings.....	29
4.12	Assets .....	30
4.13	Project management software.....	30
5	Results.....	31
6	Conclusion .....	33
7	References .....	36
8	Figures.....	39
9	Appendices.....	40

## ABBREVIATIONS

UX= user experience

UI= user interface

D.R.Y principle= Don't repeat yourself principle

## 1 Introduction

This thesis covers the topic of how to make the process of software development more efficient by improving communication between developers and designers through proper documentation of the design handoff. The author of this thesis set out to accomplish this by creating a handoff checklist and evaluating the effects of the checklist after implementation. Company and the product name will not be revealed due to competitive advantage issues (this was requested by the company). At the moment the company is refining its product development process to have better collaboration between developers and designers. There is no proper way of documenting design handoffs that's why a design handoff checklist is required. The company is also building a design system and a clear design handoff checklist is therefore called for.

### 1.1 Thesis background

The author of this thesis is working as a user interface designer (UI designer) and at the same time a fourth-year student of degree in business information technology.

A UI designer's responsibility is to convert both company and user needs into beautiful user interfaces and create great user experiences. One very important part of this job is to communicate designs to developers. Designers provide developers with documents that explain how to execute their designs. This is called the design handoff. Poorly documented designs slow down the development process. This is where the company needed help in order to come up with a design handoff checklist which would help make the development process more fluent.

### 1.2 Thesis objectives

The main objective of this thesis is to create a design handoff checklist which will help in improving the communication between developers and designers and then evaluate whether the checklist succeeded in doing so. The secondary objective of this thesis is to help the company in speeding up its software development process.

### 1.3 Thesis structure

This thesis has six parts. The first part is about introducing the problem, limitations and company. The second is about theoretical background and analysing current solutions in the field. This chapter explains what the design handoff is, how it fits in the development process and what design specifications are. The chapter then goes on to offering suggestions on how to improve collaboration between developers and designers. The third is about research methods. The fourth chapter will explain how the checklist came together based on observation of the company along with methods uncovered during literary research. Chapter five will look at

the results and effects of the checklist. The sixth and final chapter will conclude the thesis bringing together the outcome of the research and the effects of the checklist after implementation.

#### 1.4 Introduction to company and project

The company used in this thesis is a Finnish software company. Currently it is developing an online survey software which will help its users to create beautiful and intelligent surveys. The research will be focusing on the company's development team that consists of seven developers and two designers. The communication between these two actors will be a key aspect observed in this thesis. The software the company is currently developing has three layers: Super admin, which is only accessed by staff members of the company and not by customers; Product X, which is for customer use where they can create their surveys and analyse results; and the final layer is the Player, that is for end-users. Due to these different layers the software is quite advance and that's why it is perfect for this thesis because it has multiple design handoffs happening within it.

#### 1.5 Limitations

Research in the topic of design handoff is quite limited and scattered. This reflects the issue with the handoff process itself. What becomes apparent when researching design handoff is that there is no universal way of preparing the handoff document, instead each company has their own way of doing it. This thesis is bringing together some of the most common solutions to this issue and aims to make sense of the wide array of options available. Documenting design specifications is a very demanding task and it requires time, dedication and hard work. Each designer has their own way of handing off designs to developers. Some use online tools like InVision, Avocode, Zeplin etc and others manually prepare all the documents for example PSDs, Sketch files and style guides for developers. Big companies such as Apple, Microsoft, IBM and Google have their own Design systems which help their product development teams build consistent and beautiful interfaces.

There are multiple blogs and articles written by designers and developers, such as Kevin Tomasso (2017), about how to design with your developers in mind, but it can be challenging to find enough research literature to study. Procedures vary between development-teams a lot, which makes it a difficult subject to cover. Some teams have developers who are experienced enough that they don't need extra instructions on how to execute a design. Some work very close to each other and can ask if there are any doubts. Taking this into account it is difficult to put together one definitive design handoff checklist. This may affect the credibility of the thesis.



## 2 Theoretical background

We are living in a world where everything is being digitalized. Developers and designers are building products which run on multiple screens and platforms. The relationship between designers and developers is very important in order to develop a great product. Designs should be documented and communicated properly within a product development team.

Often when developers are handed a design, they encounter a situation where they don't quite understand what is required and therefore might not pay enough attention to all the details. Mostly developers stay out of the design concept or idea phase and don't get involved in finalizing designs. Sometimes designs are handed over to developers without having the big picture explained to them. Both parties are in fault here, but the consequence is always extra work for both. The lack of a proper handoff document creates a lot of confusion and development process slows down. This of course isn't beneficial to any company.

### 2.1 The importance of design handoff

Before getting to know, what a design handoff is, it's important to know why it's needed in the first place. Take any industry there are companies competing with each other to be the best. According to Jovanovic, J (2010) companies who meet user needs best and provide best user experiences win in the end. Nowadays companies are in a rush to launch products fast and rather fail fast than waste time on an idea that doesn't work. This allows companies to learn fast from their failures but the difference between failing fast and just failing should be recognized.

When you enter a software market and introduce your product it should not only function, but its design should be purposeful, and it should present your brand's identity. In the authors experience a well ironed software will definitely get more attention from the users and if it serves the purpose then it will succeed. It's good to have software that functions but having a good design is what will make users fall in love with your product.

This is why a well thought out design handoff checklist is very important. It helps in having both developers and designer on the same page. By following it, developers will build better software and there will be no duct-taping before launch. A design handoff checklist works just like a food recipe, it contains all the ingredients and instructions on how to build the user interface.

### 2.2 What is design handoff or design specifications?

According to Babich (2017) development teams spend 50% of their time doing avoidable work, due to the lack of proper communication. Communication is very critical during the development process so both designers and developers should continue to communicate with each other throughout the entire development process.

Design handoff or design specifications is a document which contains all the layers of information necessary to building a software. It's like a manual which has information about user interface details for example colours, fonts and margins. It also contains information about user flows, interaction flows, component behaviours and functionality. Developers use this document to build the software according to design needs.

The handoff document helps developers stay focused on what they should be building. It also helps designer to stay focused on visuals, experiences and interactions instead of wasting time on revisioning the specifications. Babich (2017).

#### 2.2.1 A design handoff checklist makes communication faster and smoother

Sometimes when developers don't have enough information about UIs, for example if some components or screens or artboards have multiple states, they can easily check from the handoff document instead of checking with designers. This helps in saving a lot of time.

Nowadays many teams collaborate remotely and sometimes it can be very difficult to contact designers due to, for example time zone differences. Design handoff can be very useful in solving this issue if it is stored on some google drive or cloud service. This way developers can check all the specifications remotely.

#### 2.2.2 Keeps designs up to date

Sometimes with time or during the development process, teams change their design to be up to date. When this happens, it is important that the changes are updated everywhere and not only in one place. If the designs are not kept up to date everywhere, it can lead to confusion, miscommunication and extra work that could have been avoided. It is therefore important to keep the entire development team onboard at all times.

### 2.3 Successful IT companies' approach to design handoff

Larger companies with more funds have their own design systems and style guides. This helps these companies build consistent, beautiful and great user experiences for their customers. Big companies tend to have a lot of experience and funds at their disposal and therefore have had the opportunity to spend more time in refining their processes. This is why following their footsteps is a good start. Design systems and style guides will now be introduced briefly.

#### 2.3.1 Understanding a Design system

For the last couple of years design systems have been getting very popular among small and big IT companies. The reason why design systems are getting popular is because they can fill the collaboration gap and speed up the development process more comprehensively com-

pared to what traditional methods have been offering. A design system is a collection of reusable components which are built by following some clear principals and standards. They allow you to build multiple applications by assembling together the same components. IT-companies are expected to build software at a fast pace but it's not easy because building software is a very complex process, usually made more difficult by having multiple teams working on the same product. This can lead to the outcome becoming very inconsistent. Design systems help teams build more consistent experiences more efficiently, which leads directly to financial profits for the company. Figure 1 demonstrates nicely how investing in a design system can be costly to begin with but saves funds in the long run.

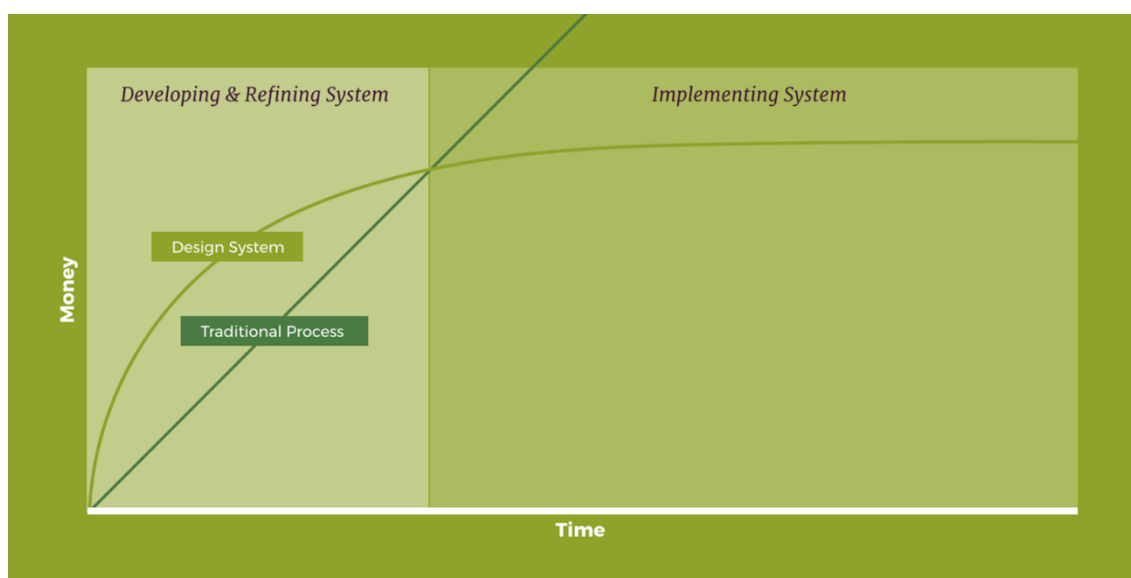


Figure 1 What is a design system? (Clark 2017)

Courtney Clark makes a point in her article about design systems working like a LEGO set. In a basic LEGO set you normally have a brick, stud, plate, tile etc. All these pieces are also available in different colours and shapes. With these pieces you can make multiple combinations because you can reuse same pieces multiple times and build something new every time. When we build a design system, we build bricks and tiles in the shape of components and patterns which we can reuse to build multiple products. A design system is a series of components and patterns that can be reused to build multiple products. It also allows companies to manage design at scale (Clark C. 2017).

According to Nathan Curtis (2016): “a design system is not a project. It’s a product serving multiple products.” This compliments Clarks metaphor of the Lego set.

Key benefits of implementing a design system:

- Efficiency

Design systems save a lot of time and money because instead of building new components from scratch you can reuse existing components by tweaking them slightly as needed for each new product. This way teams can shift products more efficiently.

- Consistency

All components in a design system follow the same design rules and principles everywhere. When you reuse these components in different products they still behave in the same way, which eliminates any kind of inconsistencies in a product. This means that whenever users interact with your products their experience feels the same.

- Scalability

When you can build products more efficiently, you can easily manage the scale. Design systems are continuously maintained and kept up to date. With changing design trends, components get old, and old features get outdated. Because design systems are continuously kept up to date you can grow your design system and keep all your products up to date (Clark C. 2017).

### 2.3.2 Understanding Style guides

Dorin (2017) defines style guides as a set of standards and principals which should be followed by every designer and developer to improve the digital presence of a product. Before design systems got popular everyone was relying on style guides. In fact, many companies are still using style guides. Style guides are a mix of visual guidelines and design principles. They also contain information about colour palettes, typography, brand, icons, page layouts, grids and assets. Style guides can be helpful for specific projects but have limitations when you need a set of rules for multiple projects simultaneously. In the case of more versatile projects design systems are preferable. Style guides:

- Improve collaboration in a team

Style guides present common rules for building an application, which helps co-operation between designers and developers.

- Keep company's brand identity alive

A style guide ensures you maintain design that follows brand guidelines and ensures visual consistency.

- Helps new employees in familiarising with the company

Provides a ready set of rules for new employees to follow, which makes it easier to acclimate to company procedures. (Dorin (2017))

### 2.3.3 Design systems, style guides, what's the difference?

Rutherford claims in his article, *Design Systems vs. Pattern Libraries vs. Style Guides - What's the Difference?* (n.d): "At the broadest level, a design system is a living entity containing the common linguistics, principles, and tools to help teams build products coherently." Style guides on the other hand are subset of design system and they are more static. This is presented in figure 2.

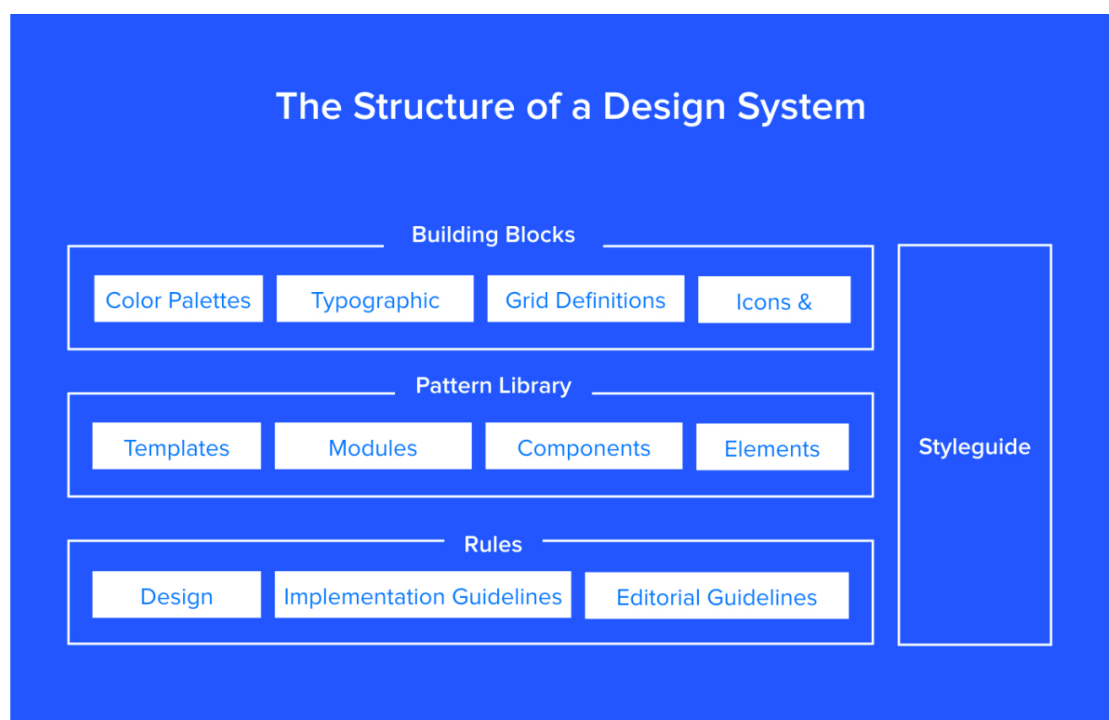


Figure 2 *Design Systems vs. Pattern Libraries vs. Style Guides - What's the Difference?* (Rutherford n.d.)

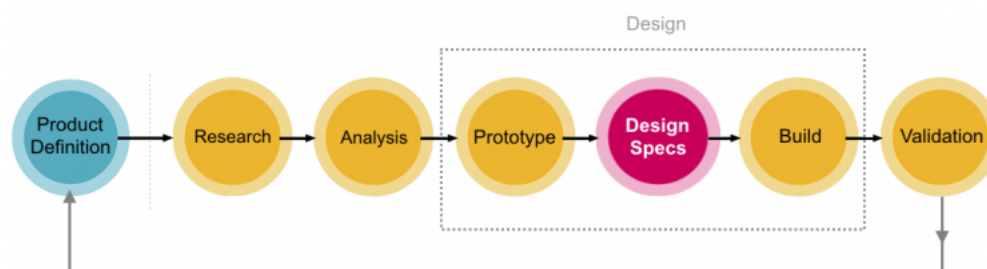
Whether you use a design system, or a style guide they will both help in improving collaboration within a team and help a company build products more efficiently. Before choosing one over the other, you should assess the company's needs. If you are building a small business website, then adopting a style guide approach is more suitable. But if you are a bigger organization planning to build multiple products then it's better to have a design system, because it allows you to construct different products while ensuring representation of brand identity. In the case of the company used in this thesis a design system was required because the ongoing project was complicated spanning over a longer period of time.

Both design systems and style guides are a good solution to avoid misunderstanding within a team because they help everyone to follow one language. Design systems are a better approach to solve design handoff problem because their modular approach to building UI components or blocks prevent inconsistencies.

#### 2.4 Design handoff and product development process

To understand better why design handoff is important, the product development process and how the design handoff fits in it will be introduced. According to Babich (2017), design handoff is introduced at the end of the design phase right before the designs are handed over to developers. At this point expectations for a product according to user needs are already met. It is very important at this stage that multiple layers of information are communicated properly to developers before they start implementing the design.

For example, let's say you are building an online software, design specs should have information about software screens, mock-ups, interactions, grid system, colours, typography, copy, assets, task checklist, margins and measurements. Figure 3 demonstrates how the design handoff, which is represented by the pink circle, is located in the development process.



*Figure 3 Design Specifications—Speeding Up the Design to Development Workflow and Improving Productivity (Babich 2017)*

#### 2.5 Understanding developers and designers

According to Leverenz (2015), both designers and developers are critical thinkers. Designers don't just try to make things look pretty, their goal is to solve problems and make the user's life easy. Their aim is to build user interfaces, patterns and standards to create great user experiences. Designers try their best to design user interfaces which are easy enough for any users. Developers turn these user interfaces into code. Their purpose is to make things work. Developers pay more attention to the technical side of a software. Given below are matters developers specifically pay attention to:

- Software should be responsive
- It should work on different browsers

- Page loading should be fast
- How many requests are being made in the software and can we reduce the number of requests
- Is the code minified and optimized?
- Are images optimized

Developers will make sure all the functionalities are there, but they might not focus as much on making sure that the margins and paddings are right. Perfecting the styles is not something developers do. Sometimes there are some developers who are also designers. These developers with the ability to design will make sure the styles are correct, but it is not very common to have a developer who is also a designer.

## 2.6 How to improve collaboration between designers and developers

Babich (2017) says neither designers nor developers can see the whole picture on their own. They have to collaborate with each other to build a great software. Design is all about working together, sharing ideas and going through multiple iterations to finalize a good solution. By involving developers in the design process, developers can help designers in finalizing their ideas by making sure that the designs are technically possible. If both parties are involved, they can help each other in building great user experiences.

### 2.6.1 Start collaborating from the start

According to Aapaoja, Haapasalo and Söderström (2013) if everyone involved in a project starts collaborating from the start then the process of iteration can be reduced. By having everyone on the same page from the beginning teams can build products more efficiently. When everyone is aware of the global goals of a product and what is expected from them then everyone stays focused all the time.

That's why it is very important that designers set some goals and communicate all the requirement to developers from the start, so they know what is expected from them. This way teams can avoid last minute fixes.

### 2.6.2 Constantly keep communicating with each other during design process

To build better communication both developers and designers should start communicating from the start instead of only during the design handoff process. After testing out some concepts with users it's important to share these thoughts with developers so they can understand why some features are selected and what users want. By sharing the background and findings developers can understand the bigger picture.

Constant communication helps designers explain to developers what exactly they are trying to achieve, and it will help developers understand how to implement the designs. By having open communication designers can save a lot of time because when they share their ideas with developers at an early stage, they ensure that they aren't designing something technically impossible. More experienced developers can also have very good ideas which can be useful in speeding up the development process. In other words, having open and constant communication is beneficial to everyone.

### 2.6.3 Designing with developers in mind

It is very common for designers to blame developers if the product does not look like the designs. This was also noted by the author while observing the development. There are always issues like margins and padding are not right, wrong hover effects or wrong font. This can be avoided, if designers spend as much time in documenting the designs as they spend in perfecting all those pixels and margins. By considering the developers and providing them with a well-documented design handoff, you can improve the quality of a product.

According to Tomasso (2017) by understanding how your development team functions and what processes they use to implement your designs you can create better designs. When there is harmony and trust within a development team, better products will result.

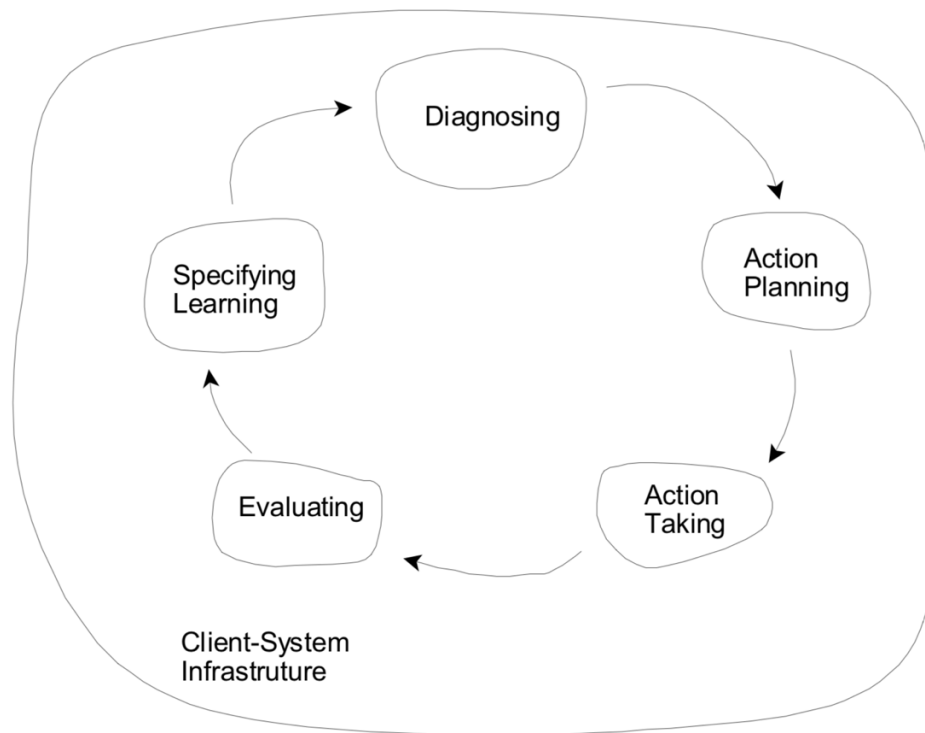
### 2.6.4 Learn basic programming languages like HTML and CSS

Nowadays every job advert for a designer has a requirement that you should have knowledge of basic programming languages like HTML, CSS and JavaScript. It's not compulsory to have this knowledge but if you learn these languages this will make you a better designer. You will be able to think like a developer and design better solutions.

## 3 Research method

This thesis set out to prove that a carefully constructed design handoff checklist can improve the efficiency of the development process in a specific company. For the purpose of measuring this, the thesis utilises the action research approach. Action research follows a cycle of diagnosing, experimenting, evaluating and learning which enables the researcher to participate in the experiment and assess human behaviour. Figure 4 demonstrates this cycle (Baskerville R. 1999). In the case of this thesis the author constructed a design hand off checklist together with the development team and then observed as the checklist was implemented into the development process. Due to the nature of this thesis being one that measures experience the methods chosen to study the matter of the design-handoff are mostly qualitative in nature. Therefore, action research was the ideal approach to this topic.





*Figure 4 The Action Research Cycle (Baskerville 1999)*

The first stage of action research is diagnosing. As Richard Baskerville puts it: “Diagnosing corresponds to the identification of the primary problems that are the underlying causes of the organization’s desire for change” (Baskerville R. 1999). In order to diagnose the issue at hand the company was first observed over a period of one month to see how design handoff functioned and what was required for the hand off checklist.

The second stage of the action research consists of action planning. The purpose of this stage is effectively to make a plan for resolving the issues discovered in the diagnosing stage (Baskerville R. 1999). For this stage interviews were conducted with the developers and designers on how they perceive the design handoff. The results were then paired with a thorough literary research that led to the creation of the design handoff checklist.

Once the checklist was ready it was implemented in the action taking stage, which is the third stage of the action research approach (Baskerville R. 1999). After implementation of the checklist the developers and designers were observed for a further two months to evaluate how the checklist affected the handoff in everyday work between developers and designers. In addition to observation, interviews were conducted also after the implementation of the handoff guide to assess both the developer and designer perspective on their experiences of the handoff with the checklist.

The next stage after action taking is evaluation (Baskerville R. 1999). At this point the results of observation and the second round of interviews were evaluated. Which leads us to the final stage of action research: specifying learning. At this stage the results of the research are considered and if successful the design handoff checklist can be added permanently as a tool used by the development team. If the experiment proved unsuccessful, this would be the point to start a new cycle of diagnosing what was wrong with the first version of the checklist.

### 3.1 Why this method of research?

While the purpose was to measure changes in efficiency and fluency a fully quantitative approach would not have yielded the wanted results because the projects designers and developers work on vary greatly in size and complexity even within this particular project that was observed. This means that time spent on communication between developers and designers during handoff is affected by project size and therefore the data of time spent would not give reliable research results on changes in efficiency.

## 4 Implementation and verification of design handoff guide

According to Mohammed, B. (2016) designers spend a lot of time and care when they are designing for users. It would be nice if they spent as much time and care when preparing the design handoff documents. Developers don't just skim through design handoff documents they actually dig deep and read through all the designs. By preparing easy to follow instructions and explaining everything step by step you not only make developers happy you also speed up the development process.

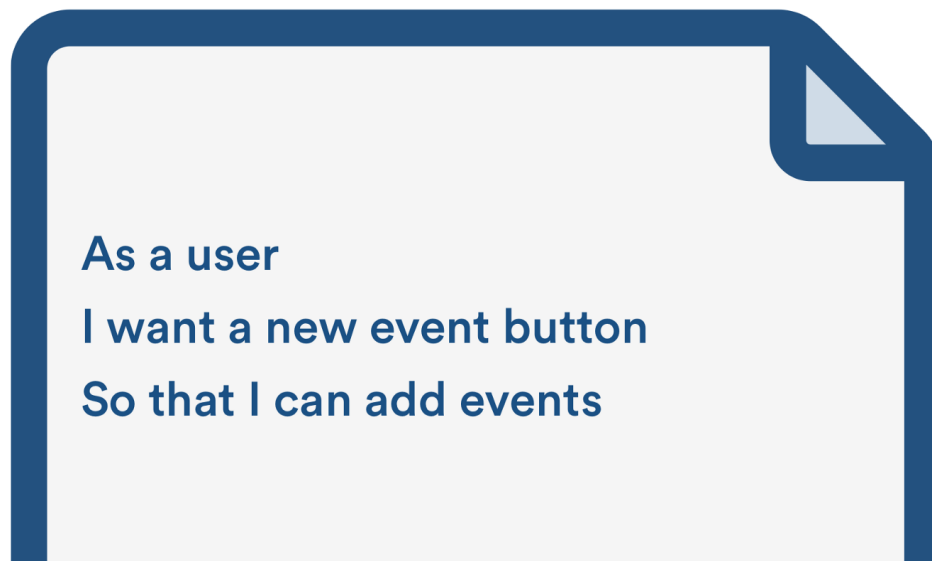
The design handoff checklist was put together largely based on theoretical research on articles on experiences in other companies worldwide. Some aspects of the checklist such as the correct use of copy were included also based on observation of the needs of the company.

Based on interviews with developers, designers, theoretical background and observation of the example company, it was decided that the features introduced in this chapter should be included in the design handoff checklist. The following steps have been taken at the company to improve communication between developers. This process can still be improved but it has been very effective so far.

### 4.1 User stories

The author noticed that developers at the company did not understand why they needed to build a particular feature and they thought they were wasting time on small things. Therefore, user stories were added in the handoff checklist. Once developers read the user stories, they understood why they had to build a particular feature.

User stories are a simple way to tell your developers what you want to build. These stories help them understand the background of a feature. Nowadays companies are following agile software development method and user stories can be very useful when your product is evolving with time and as user needs change (Mansour, 2017). Figure 4 demonstrates a typical user story, where a feature is required based on user needs.



*Figure 5 User stories (Mansour 2017)*

#### 4.2 Site maps

Developers did not have clear site maps at the company, so in course of research it was decided that a clear site map should be built. Site maps not only improved search engine optimization, but they also help developers in building the structure for applications. A site map is the simplest way to show your team how the applications will look like and how users will navigate through it.

According to Govella (2017) a Site map or Information Architecture is a simple way to show a web applications structure. It focuses on three very important things.

- How contents are organized
- How contents are labelled
- How user can navigate through content

Figure 5 demonstrates how a sitemap can help to visualize the structure of an application:

Figure 1: A simple sitemap

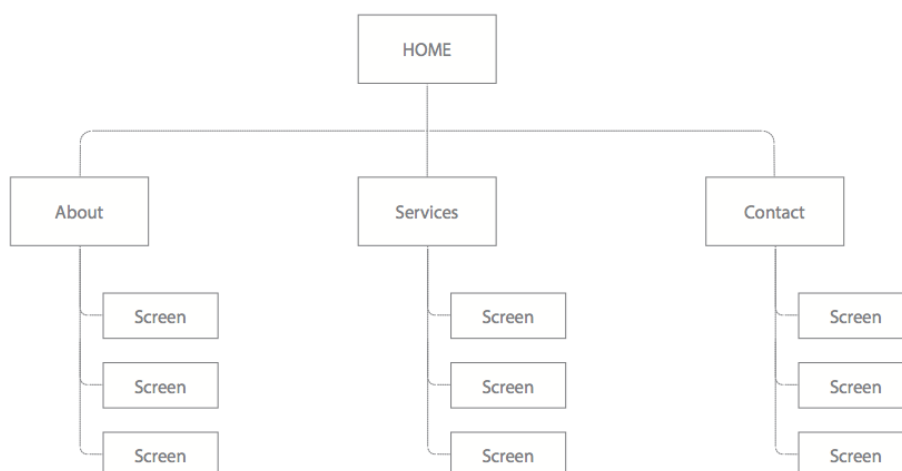


Figure 6 The secret lives of sitemaps (Govella 2016)

#### 4.3 Documenting variables, constants and properties

While building an advanced software there are many things that need to be documented. Variables, constants and properties are very important, and they are used everywhere in the software, so it is very good practice to store them all in a document, in order for the developers to access them easily. Having them saved in one place will help with consistency and facilitate possible later changes. You can also add list of Event logs and URLs in the same documents.

The company faced a problem that in some input fields users could type only 40 characters and in some 120 characters. When the design files were checked it became clear that they had different values for input fields in different designs. This costed more time to fix the issue. To avoid this issue in the future, common variables and constants were put in one document. Designers could then reference these variables or constants in their designs, and developers could check the document for the value of a particular variable or constant. The same variables could then be used in multiple designs.

#### 4.4 Copy

Copy is a document that specifies what kind of tone and voice should be used in a software. It contains all the text that should be used in the UI. For example, what is feedback message, alert message and placeholder text etc. Sometimes it's not easy to show all the text in design so it's good practice to have a Copy document which contains all the text for UI.

The company did not have a Copy document, so it was requested from the product owner to provide a Copy document because he had a better idea about what kind of tone and voice should be used in the software. Once the Copy document was acquired it was organised according to Figure 6 and shared with developers. This helped developers in finding the right Copy at one place

According to Mohammed, B (2017), it is sometimes very difficult to show the entire Copy in the user interface mock-ups. It is good practice to save all of the Copy in one document which is shared with developers. Share your Copy in three columns as demonstrated in figure 6.

First column: describe the type of Copy. It will help developers to quickly parse through the page. For example, you can group your Copy by page names (Home, About, Contact etc).

Second column: explain the heuristics behind the Copy. What are best practices according to heuristics. For instance, using user-friendly language.

Third column: write down the exact message that should be shown to the user in the user interface.

Type	Heuristic	Message
<b>Homepage</b>		
Banner	————— ————	—————
Alert	—————	—————
Alert	—————	————— ————
User Profile	——— ————	——— ————
Error	—————	——— ————
<b>Cart</b>		
Empty State	—————	—————
Promoted	——— ————	—————
Alert	——— ————	——— ————
Error	————— ————	—————

Figure 7 A guide to successful design handoffs (Mohammed 2017)

#### 4.5 Naming conventions

In the first round of interviews it became apparent that developers did not understand which mock-up belong to each other. They said it was hard to categorise mock-ups because they had random names. This is why consistent naming became a part of the design checklist. Now at the company designers have been naming screens according to page names as suggested by Mohammed (2017) which has helped developers in finding the designs easily.

Mock-ups or screens contain all your design details, they should be shared with your developers. It's a simple task but there are a few points which are very important to keep in mind before sharing your mock-ups.

Naming screens can impact the flow of co-operation between designers and developers. Sometimes designers add versioning while naming the screens which is not a good habit because versions as names don't explain the purpose of the mock-up or screen. Screen names should be self-explanatory. It's also important to use consistent casing while naming your screens, you can use camel casing or lower casing, but it should be consistent. Developers will appreciate it if you use consistent casing while naming your screens, because they use consistent casing while building the software. If naming is done inconsistently the developers will have to rename, which causes unnecessary work.

When designers work on a project they go through many iterations before they finalise a design. Sometimes there are multiple concepts in the same file. Developers don't appreciate receiving a file with multiple versions, because it is confusing and makes work more difficult. It will be very hard for them to figure out which is the final design. So, it's best to weed out all the unnecessary screen or artboard and keep only the final one in it.

Figure 7 shows multiple versions of a design. "Home1.png" should not be included in the file passed on to the developers. Instead only the final "home new2.png" should be included and this should be named simply home.png as demonstrated on the left side of the image.

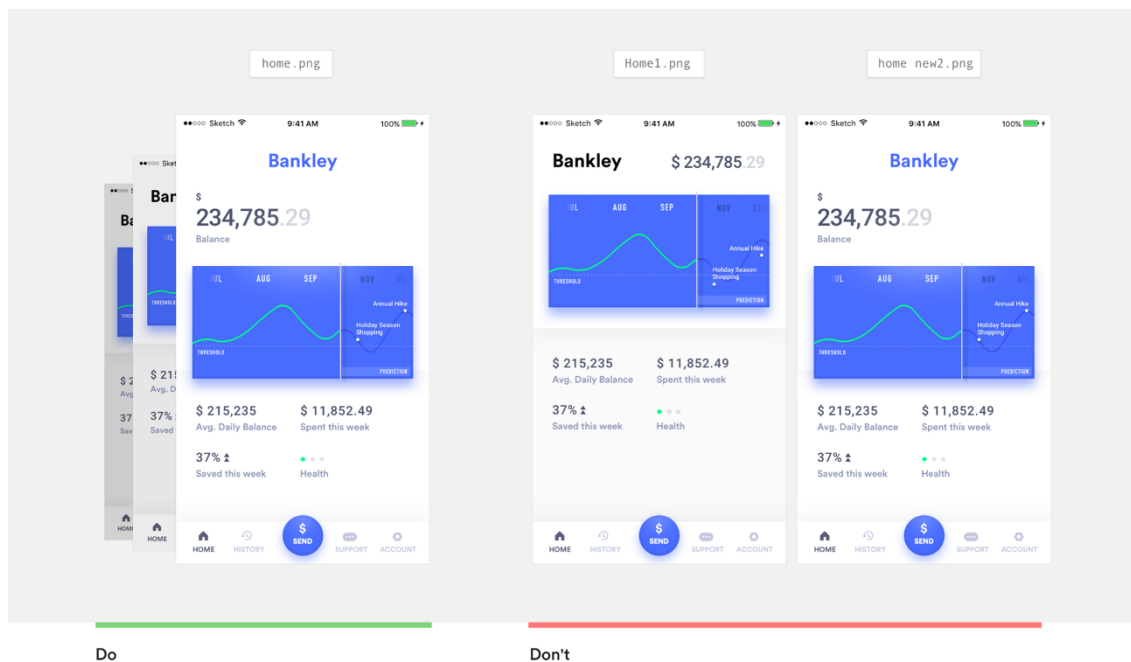


Figure 8 A guide to successful design handoffs (Mohammed 2017)

#### 4.6 Interactions flow

The developers mentioned in the interviews that the design team was always missing some states, like what the user sees while a screen is loading, in their designs. Design handoff checklist has already solved that issue because now the designers make sure all the states are added in designs, so developers don't have to check with the design team again.

Designers spend a lot of time fine tuning and tweaking the designs and it can be frustrating for them to see if the end product does not match or behave according to designs. It's very easy to blame developers that they did not do their job right, but designers should be more critical about themselves because they could have prepared the documentation properly before sharing it with developers.

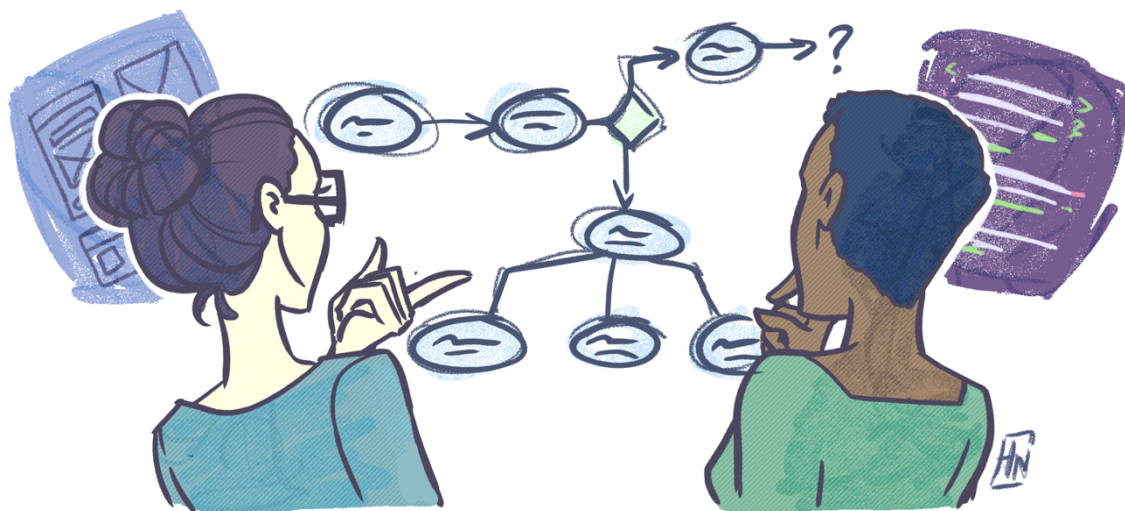


Figure 9 An introduction to interaction flow (Nguyen 2017)

Sharing just static screens is not enough, there are multiple states and steps involved to complete a task in a software. If these steps are not explained properly to developers, then they might change the flow or navigations of the software and this will affect user experience which can be costly. It is therefore very important to document these user interactions, for example what happens when a user presses some button, where does it take the user, does it take to another page or does it open a dialog, what happens next and so on. These are questions that need to be answered in interaction flow. It's important to document Task flows and Wire flows so that developers can build the software the way the designer has pictured it.

#### 4.7 Prototypes

According to Simpson (2017), prototypes are the best way to present your ideas to a team. Just like a picture can paint a thousand words a prototype can paint a thousand user stories. It is good to build a prototype if you are building an advanced software. Developers sometimes tend to ignore reading long design related explanations. By having a prototype, you can make the developers work easier.

In interviews with developers some developers complained that it's hard to read through long handoff documents. When a feature is very advanced, and designs have multiple states then it's very helpful to have a prototype.

#### 4.8 User interface components

According to Kamelsky (2017), a UI kit contains all the elements used in a software. It helps designers and developers to build consistent user interfaces. If you are working on a bigger project, then UI kits are very helpful in maintenance process. While building a UI kit follow



D.R.Y principle<sup>1</sup>, this is short for Don't Repeat Yourself. If you are using an element in two places in your software, then put one in the UI kit. By doing so no matter how many times that element is used in your software it only needs to be built once and can then be used again wherever necessary. If you don't have a UI kit, you will end up having the same component with different properties and developers will have to build it multiple times. This will take up more time and slow down the development process.

According to Kamelsky, here is a list of UI elements which are mostly used in 90% of existing software:

- Buttons
- Fields
- Lists
- Drop-down list
- Tables
- Images
- Labels
- Menus
- Carousel
- Checkboxes
- Radio groups
- Headers H1 - H6
- Paragraphs
- Spinners and progress bars
- Notifications
- Grids

There were many inconsistent UI elements at the company, for example there was around 20 different types of buttons, when only seven types of buttons were needed with different states. Menus, dropdowns and text styles were also inconsistent, so a design system was built with the help of the design team and all the UI elements were added in it, so the same ones could be used everywhere. Now developers only check the design system for latest UI components and use them everywhere in the product. This has saved a lot of time because now the team is more consistent, and developers are not wasting time on building extra inconsistent UI elements.

---

<sup>1</sup> D.R.Y principle= Don't repeat yourself principle

#### 4.9 States for components

Handing over a bunch of components to developers is not enough. Each component behaves in a certain way when the users interact with it, so it is very important to show all the states a component has. If you don't communicate these states, the developers might miss the states or add a totally different behaviour to a component than was originally intended.

According to Speelman (2015), by adding stateful design thinking in your design process you develop sympathy for users which helps you in designing better user experiences.

According to Speelman (2015), here are some of the most important states:

- Resting / Default
- Hover
- Focus
- Clicked on
- Empty
- Complete
- Full
- On / Off
- Switched
- Visited

Figure 9 demonstrates some of these states (*Lyakhov 2017*)

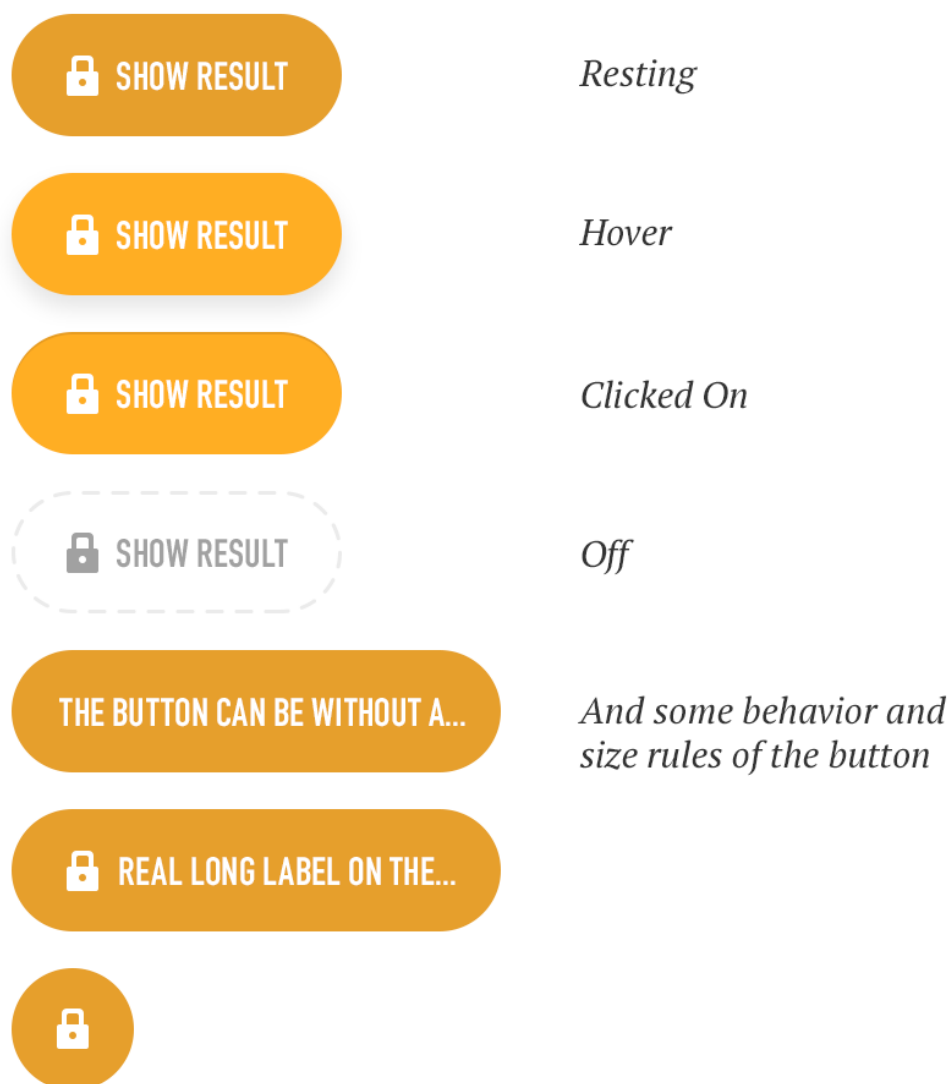


Figure 10 10 reasons to have design checklists and UI kits (Lyakhov 2017)

The old handoff documents in the company were missing some of the states of the UI elements. For example, buttons had only default state and were missing states like what happens when user hovers on a button and what happens when user clicks on it. This was a major issue because user testing uncovered that some users were confused about button functionalities. Therefore, the checklist shows now whenever a UI element is built, designers should add all important states for that element.

#### 4.10 Grids

According to Babich (2017), it doesn't matter if you are designing a magazine, newspaper, web application or a mobile application. Every designer needs some kind of grid to structure

their designs. A grid acts as invisible glue that holds your designs together. Figure 10 shows the grid behind a desktop screen.

Here are some of the reasons why grids are very important:

#### 4.10.1 Keeps your content organised

One of the most important reasons why a grid is used is because it provides consistency. By using a grid, you can place your content more neatly which gives your page layout a cleaner look and feel.

#### 4.10.2 Enhances visual hierarchy

According to Everette (2013) having a page layout that communicates the purpose of the page is very important because it helps users find what they are looking for as quickly as possible. Most designers don't pay attention to enhancing the visual hierarchy because they just concentrate on getting everything to fit in a layout. If a user ends up on a cluttered layout then it is likely they will give up and leave your application.

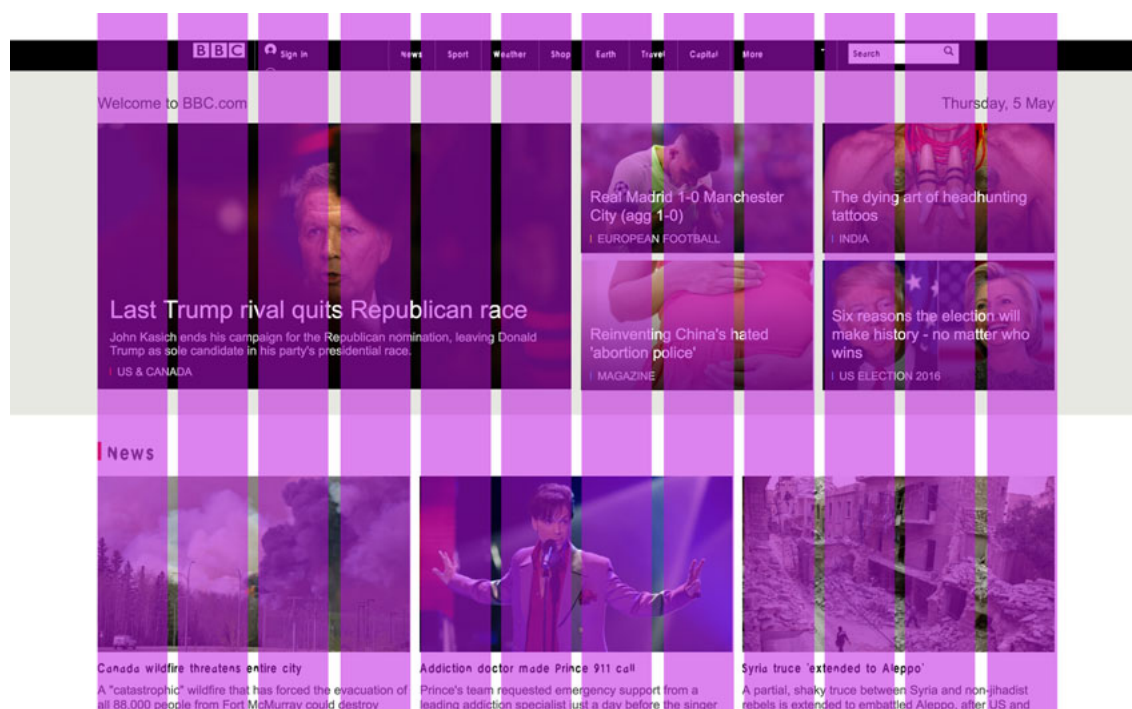
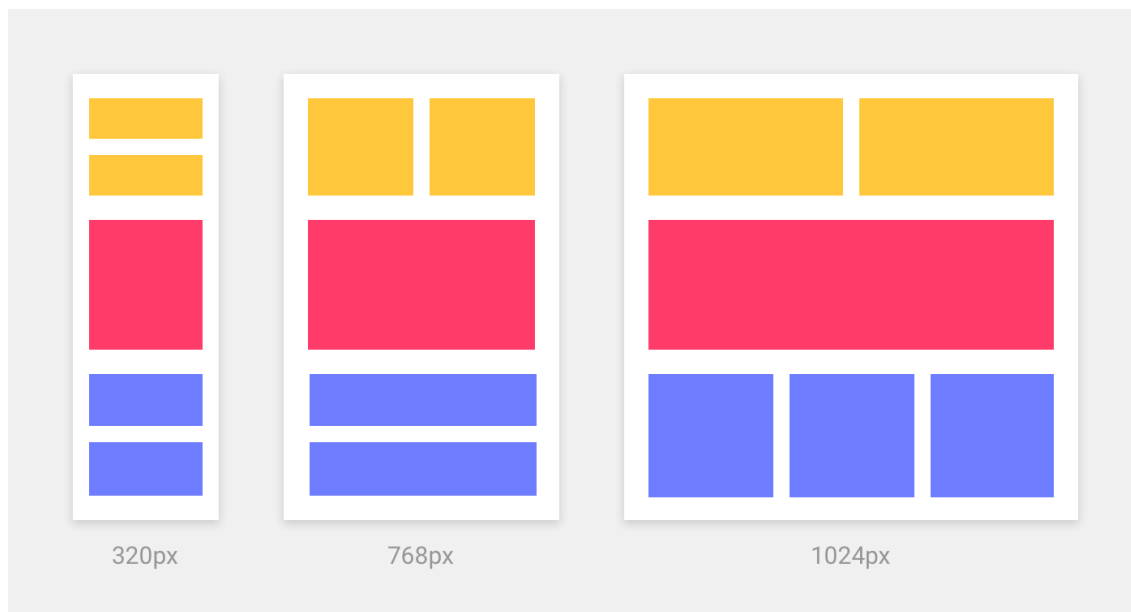


Figure 11 A comprehensive introduction to grids in web design (Wassermann 2016)

Grids are the best solution in laying out your elements according to their importance. They also help in designing according to different screen sizes. Nowadays designers have to design for multiple screen sizes and grids are the perfect solution for that.



*Figure 12 Handoffs guide for pixel perfect design part 2 (Kamelsky 2017)*

This is also why sharing your grids with developers in the start is very important. By discussing the grids in advance designers will have the confidence to design according to multiple screen sizes. Figure 11 demonstrates the grid layout for three different screen sizes: mobile, tablet and desktop.

The company's product page layouts were inconsistent at the beginning of the research process. This caused the software to look broken. At some places the development team was using 2 column layouts and at other 1 column layouts. In order to address this issue, the author studied the entire software and categorised the pages. As a result, three categories were formed: pages that use 1 column, 2 columns and 3 columns. Then these layouts were added in the design system and used everywhere. This helped in having more consistent page layouts.

#### 4.11 Margins and paddings

Once your designs are ready it is very important to redline everything. Redlining is the process when designers define what the margins and paddings between elements are. It is a very tedious and time-consuming process, but it helps developers in implementing the designs more accurately. Some designers prefer to redline their own UI designs but with the tools available nowadays it seems a waste of time (Mohammed, B 2017).

There are so many online tools which can help developers inspect your designs and get all the specs themselves. All you have to do is properly organise layers and groups in your Sketch, PSD file and these tools will do the rest. Some of these tools even help in creating design guidelines and code.

When designers redline their designs, developers don't have an excuse to not execute the design properly. Designers can then question developers if they find any discrepancies in the build.

Name	Platform	Free	Subscription	Purchase	Programs	Specs	Automatic	Assets	Measure	Variables	Guides	Comments	Sharing	Integrations
Avocode	Apple, Windows, Android		\$7		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Apple, Windows, Android	Integrations
InVision	Cloud		\$15		Yes	Yes			Yes			Yes	Cloud	
Marketch	Apple	Yes			Yes	Yes			Yes				Cloud	
Markly	Apple			\$50	Yes	Yes			Yes				Cloud	
Red Pen	Cloud		\$20									Yes	Cloud	
Sketch Measure	Apple	Yes			Yes	Yes	Yes	Yes	Yes	Yes			Cloud	
Specctr	Apple, Windows	Yes	\$9		Yes	Yes			Yes			Yes	Cloud	
Sympil	Cloud	Yes	\$15		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Cloud	Integrations
UX Pin	Cloud		\$20		Yes	Yes			Yes			Yes	Cloud	
Zeplin	Apple, Windows	Yes	\$15		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Cloud	Integrations

Figure 13 Various tools with their design handoff capabilities (Mohammed 2017)

In interviews some developers complained that sometimes designs don't have consistent margins and paddings. This issue was solved by suggesting to the design team that they use online tools which auto generate margins and paddings. They have been using InVision, but you can use any online tool that serves your needs best. Figure 13 shows some of the other tools available.

#### 4.12 Assets

Assets are icons, images and illustrations which you use in your UI designs. Make sure all your assets are exportable when you hand off your design files. Many times, designers forget to share the assets, and this causes delay in the development process because developers have to wait for assets. It is best to make sure that your assets are exportable when you share your files via InVision, Avocode, Zeplin etc or create an assets folder on google drive and share it with developers (Mohammed, B 2017).

Developers also complained about not getting all the assets in design handoffs. To solve this problem, a google drive folder was made which had all the assets used in software. This helped developers find all assets in one place.

#### 4.13 Project management software

There are many project management software's and you can pick any that you wish but at this company they have been using Trello. Trello is a great project management tool, it is based on Kanban system. One of the things that makes Trello such a useful tool is that there can be multiple boards for different teams. For example, design team can have their own board and development team can have their own. Each board has multiple columns and they are named according to development process. These could include for example, software features, backlog features, in development features and ready for testing features. Columns

have a list of cards and these cards are based on user stories. A card can be assigned to different team members and they can have deadlines. Cards are moved from left to right based on their status. Once a card or task is ready then it's moved to the next column in the process.

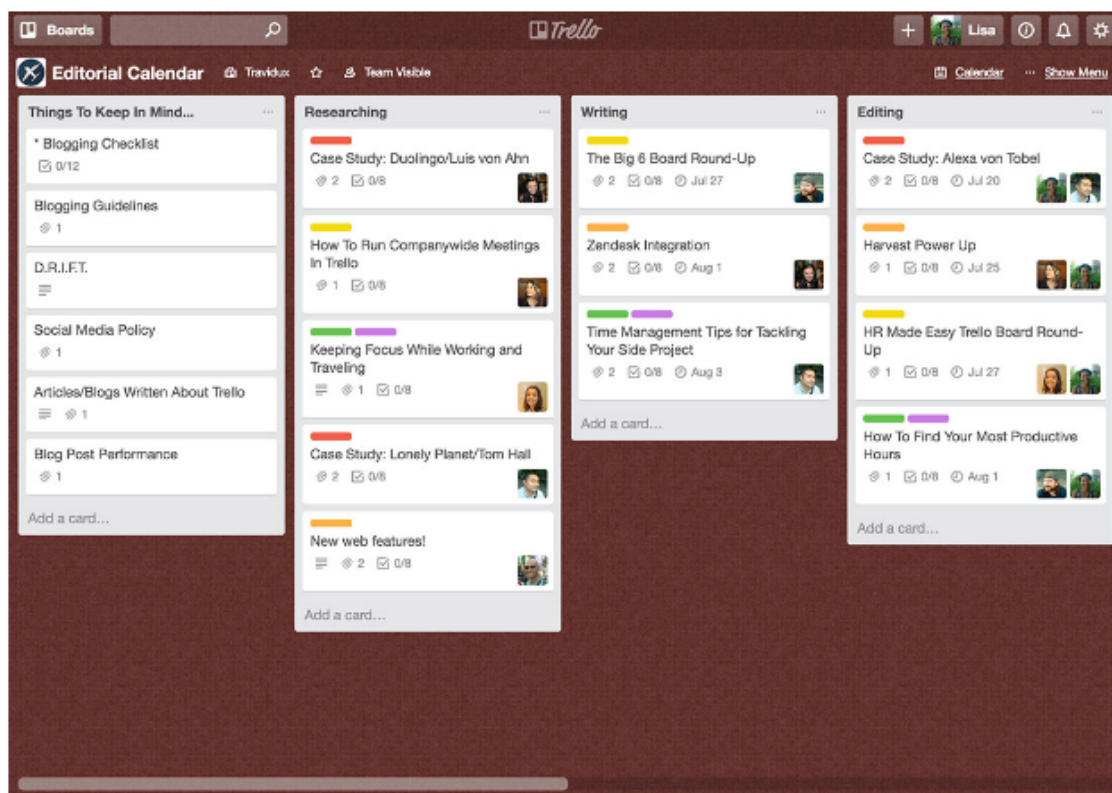


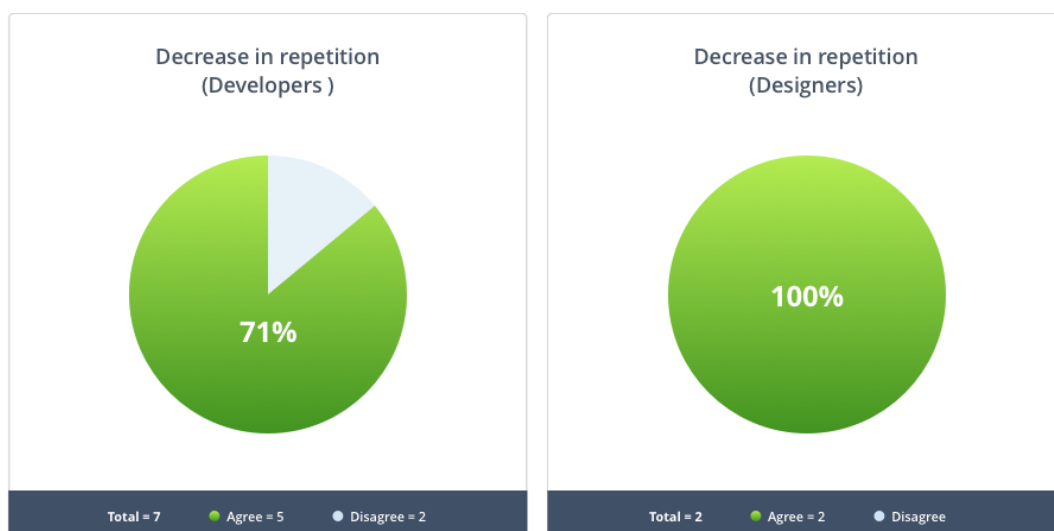
Figure 14 Trello vs Asana - The best project management app for 2018 (Zivkovic 2018)

Trello is a great solution for developers and designers to collaborate with. As each card is based on a user story or requirement, designers can attach all the task related design handoff documents to it. This helps developers in finding all important documents at one place. Developers and designers can also chat on these cards which is great because they can have task specific conversations. In short using a project management tool is a great way to keep everyone on the same page.

## 5 Results

After implementation of the checklist, the handoff process was observed, and an interview was conducted with both developers and designers. The results were even more positive than expected. There was a significant decrease in miscommunication and repetition of work wasn't happening the way it had before. In the second interview, the same four questions were asked from both sides. The first question was: *do you feel like you are doing the same*

*work over and over again, or have you noticed a decrease in repetition after the checklist?* The general consensus was that the amount of unnecessary repetition of work had decreased. Five out of seven developers felt that repetition had decreased along with both of the designers. The remaining two developers felt like they were still having to repeat some work but were unsure whether the amount of repetition had decreased. These particular developers were working on a more complicated feature at the time, which explains the additional communication compared to the rest of the team. The results of this question are represented in figure 15.



*Figure 15 do you feel like you are doing the same work over and over again, or have you noticed a decrease in repetition after the checklist?*

The second question: *Do you feel like you're communicating more or less after the checklist?* received a unanimous vote that communication between the two sides had improved and become smoother. Especially the use of Trello to manage projects was considered a game-changer in communication

The third question of the interview was: *Have you noticed something missing in the checklist?* Six out of the seven developers and both designers felt that the checklist had all necessary features, but one developer brought up the idea that the checklist could also contain components with ready code for developers. While this is a good idea, the company lacks the resources to achieve this at the moment. This could however, be something to develop in the future.

The final question divided opinions somewhat between developers and designers. The question was: *How has the checklist affected your everyday work?* All the developers agreed that



the checklist had made their work easier and more efficient. It meant that all necessary information was included in the handoff and they had a clear idea about what was expected of them. The designers on the other hand both brought up the same point that the checklist has increased their workload because preparing the handoff requires now more detail and therefore time. This was to be expected because the checklist forces the designers to include everything necessary in the handoff document, whereas earlier designers could only include what they thought was necessary. However, in the big picture this means that no valuable information is omitted and the development process as a whole is smoother when developers don't have to keep coming back to designers for the missing information. The other designer also admitted that in the long run a carefully prepared handoff document does help designers as well because the designs were now executed the way they were intended and didn't keep coming back for fixing. The results of this question can be observed in figure 16.



*Figure 16 The checklist has improved everyday work*

Based on the results from observation and the data from the second interview the design handoff checklist has been a success in the company and has benefited both designers and developers in making the entire development process more fluent.

## 6 Conclusion

Nowadays users don't have much patience in figuring out how to navigate through software's, because alternatives always exist. If users don't get what they want within seconds, they will likely leave your software and find an alternative. The main reason behind having an inconsistent software is an unorganised development team. When teams don't follow a process thoroughly, inconsistencies appear.

This became apparent during the first observation period and so this thesis set out to improve the development process with a design handoff checklist. Observation revealed that the design handoff needed a lot of improvements. Design handoff was missing a proper structure and designs were handed over with some areas unexplained. Designers had the opinion that developers were not paying enough attention to details, but the results of the checklist proved that it was the handoff that needed improvements.

There were a few challenges encountered while conducting research. Because observation was used as a research method, the objectivity of any findings had to be carefully justified. In order to avoid being subjective, interviews of developers and designers were added both before and after implementation so as to have more than one perspective on the success or failure of the experiment.

Another complication encountered was the ratio of designers to developers at the company. With only two designers opposed to seven developers the interview results gave more variety on the developer side while with designers there was only two viewpoints on the matter. The lack of interviewees unfortunately gives room for error in data.

During the initial observation period, it was noted how developers kept coming back to the designers for specifications over the designs. There was also noticeable frustration on the designer's side when the final products didn't match the original designs. Designers tended to blame the developers for this, but through interviews and observation it became obvious that the designers needed to put more effort into the handoff document for the developers to be able to execute the designs as originally intended.

Initial interviews and literary research provided the following features to be included in the design checklist: user stories; site maps; documenting variables and constants, URLs, properties etc; copy; mock-ups; interaction flows; prototypes; UI kit; states for components; grids; margins and paddings; and assets. All of these came together with a project management software for which Trello was chosen.

After implementation of the checklist there was a noticeable change in the entire handoff process. All of the developers were happy with the new system that made their work easier, because all the necessary information was provided, and they didn't have to repeat work the way they used to. A majority of five out of seven developers believed they were repeating work less whereas the other two were unsure whether the amount had changed. Therefore, there was clearly a decrease in unnecessary repetition and communication had improved between the two sides. In the final interviews the developers unanimously expressed their contentment with the new system.

Designers on the other hand were unsurprisingly not overly excited about the extra work in preparing the handoff document as the checklist increases the workload for designers. The handoff document had to be now more carefully prepared and a lot of information was included for all designs that wasn't previously there. However, in the long run even the designers acquiesced that putting the work in at the beginning was beneficial for both parties, and they no longer had to see designs completed incorrectly. All in all, eight out of the nine-person development team believed that their everyday work had got more efficient and misunderstandings that led to repeated work had decreased. That's 89% of the team, which means the handoff checklist has been successful.

The research results come to show how important it is to set clear goals and guidelines from the start and get the entire development team on the same page. There is no such thing as over communicating when it comes to developers and designers. Both parties should constantly share ideas and trust each other.

The design handoff checklist doesn't have to look exactly the same for other companies. Still some form of checklist is necessary for all development teams in order to keep the development process efficient. Each team has their own weaknesses and strengths, some might need more detailed documentation, and some less so, therefore the checklist should also reflect the team.

To conclude it is the responsibility of both parties to ensure that communication is working. The designers need to put the effort in to follow the handoff checklist when preparing their designs and the developers need to make sure they know what was intended by a design before executing it. It may not be possible to create a universal handoff checklist that will work for all development teams, but each team should have their own roadmap to follow when working together. This checklist was based on the needs of the members in the development team and the results were better than expected at the beginning of the experiment. A handoff document or checklist can at best times ensure that time and resources aren't wasted in avoidable corrections, which leaves the development team free to spend more time on building beautiful user interfaces.

## 7 References

### Printed sources

Christian W. Dawson, 2011. *Projects in Computing and Information Systems: a guide for students*.

Everette N. McKay, 2013. *UI is communication: how to design intuitive, user-centered interfaces by focusing on effective communication*.

### Electronic sources

Aapaoja, A, & Haapasalo, H. & Söderström, P. 2013. Early Stakeholder Involvement in the Project Definition Phase: Case Renovation. Accessed 20 April 2018

<https://www.hindawi.com/journals/isrn/2013/953915/>

Babich, N. 2017. Building better UI designs layout grid. Accessed 20 May 2018

<https://www.smashingmagazine.com/2017/12/building-better-ui-designs-layout-grids/>

Babich, N. 2017 Design Specifications—Speeding Up the Design to Development Workflow and Improving Productivity. Accessed 6 March 2018

<https://medium.com/thinking-design/design-specifications-speeding-up-the-design-to-development-workflow-and-improving-productivity-4f6af97e64c>

Barashkov, A. 2017. Handoffs guide for pixel perfect design part 1. Accessed 20 May 2018

<https://medium.com/pixelpoint/handoffs-guide-for-pixel-perfect-design-part-i-8bbd95d8ffcd>

Baskerville, R. L. (1999). Investigating information systems with action research. *Communications of the AIS*, 2(3es), 4. Accessed 2<sup>nd</sup> November 2018

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.137.2608&rep=rep1&type=pdf>

Baskerville, R., & Myers, M. D. (2004). Special issue on action research in information systems: Making IS research relevant to practice: Foreword. *MIS quarterly*, 329-335. Accessed 2<sup>nd</sup> November 2018

<https://pdfs.semanticscholar.org/4fc0/33ba8ae6ae5ca24a1e1c5268551f1a635bee.pdf>

Clark, C. 2017. What is a Design System? Accessed 2 March 2018

<https://forumone.com/ideas/what-is-design-system>

Curtis, Nathan 02.2016 A Design System isn't a Project. It's a Product, Serving Products. Accessed 10 March 2018

<https://medium.com/eightshapes-llc/a-design-system-isn-t-a-project-it-s-a-product-serving-products-74dcffef935>

Dorin, A. 2017. How to create a style guide from scratch. Tips and tricks. Accessed 10 March 2018

<https://medium.muz.li/how-to-create-a-style-guide-from-scratch-tips-and-tricks-e00f25b423bf>

Govella, A. 2016. The secret lives of sitemaps. Part 1. Accessed 3 May 2018

<https://medium.com/why-what-what-s-meta-matters/the-secret-lives-of-sitemaps-be47bbecc886>

InVision. 2015. 5 ways to bridge the designer developer gap on responsive web projects. Accessed 25 March 2018

<https://medium.com/@InVisionApp/5-ways-to-bridge-the-designer-developer-gap-on-responsive-web-projects-5d7edf7e410b>

Jovanovic, J. February 25, 2010. User Interfaces In Business Web Application Design

<https://www.smashingmagazine.com/2010/02/designing-user-interfaces-for-business-web-applications/>

Kamelsky, V. 2017. Handoffs guide for pixel perfect design part 2. Accessed 10 May 2018

<https://medium.com/pixelpoint/handoffs-guide-for-pixel-perfect-design-part-ii-d91999742dd9>

Kamelsky, V. 2017. Handoffs guide for pixel perfect design part 3. Accessed 24 May 2018

<https://medium.com/pixelpoint/handoffs-guide-for-pixel-perfect-design-part-iii-3acc5a93d3a2>

Leverenz, A. 2015. Successful collaboration between designers and developers. Accessed 2 April 2018

<https://medium.com/@justalever/successful-collaboration-between-designers-and-developers-4d9ebdd6efe7>

Lyakhov, K. 2017. 10 reasons to have design checklists and UI kits. Accessed 20 May 2018

<https://medium.com/shakuro/10-reasons-to-have-design-checklists-ui-kits-64f219f90f3d>

Mansour, S. 2017. How we destroyed user stories. Accessed 21 May 2018

<https://medium.com/@sherifmansour/how-weve-destroyed-user-stories-8b36120645c6>

Mohammed, B. 2017. A guide to successful design design handoffs. Accessed 25 March 2018

<https://blog.marvelapp.com/guide-successful-design-handoffs/>

Nguyen, H. 2017. An introduction to interaction flows. Accessed 2 April 2018

<https://uxplanet.org/an-introduction-to-interaction-flows-a4f783402529>

Rutherford, Z. n.d. Design Systems vs. Pattern Libraries vs. Style guides - What's the difference? Accessed 27 February 2018

<https://www.uxpin.com/studio/blog/design-systems-vs-pattern-libraries-vs-style-guides-whats-difference/>

Simpson, A. 2017. Why prototyping is a must for designers. Accessed 7 May 2018

<https://medium.com/designing-atlassian/why-prototyping-is-a-must-for-designers-5ef98dfb3bdc>

Speelman, V. 2015. The nine states of design. Accessed 4 April 2018

<https://medium.com/swlh/the-nine-states-of-design-5bfe9b3d6d85>

Tomasso, K. n.d. Designing with your developers in mind

<https://www.invisionapp.com/ecourses/designing-for-developers>

Wassermann, E. 2016. A comprehensive introduction to grids in web design. Accessed 21 May 2018

<https://webdesign.tutsplus.com/articles/a-comprehensive-introduction-to-grids-in-web-design--cms-26521>

Zivkovic, M. 2018. Trello vs Asana - the best project management app for 2018? Accessed 24 May 2018

<https://blog.leadquizzes.com/trello-vs-asana>

## 8 Figures

<i>Figure 1 What is a design system? (Clark 2017)</i> .....	11
<i>Figure 2 Design Systems vs. Pattern Libraries vs. Style Guides - What's the Difference? (Rutherford n.d.)</i> .....	13
<i>Figure 3 Design Specifications—Speeding Up the Design to Development Workflow and Improving Productivity (Babich 2017)</i> .....	14
<i>Figure 4 The Action Research Cycle (Baskerville 1999)</i> .....	17
<i>Figure 5 User stories (Mansour 2017)</i> .....	19
<i>Figure 6 The secret lives of sitemaps (Govella 2016)</i> .....	20
<i>Figure 7 A guide to successful design handoffs (Mohammed 2017)</i> .....	21
<i>Figure 8 A guide to successful design handoffs (Mohammed 2017)</i> .....	23
<i>Figure 9 An introduction to interaction flow (Nguyen 2017)</i> .....	24
<i>Figure 10 10 reasons to have design checklists and UI kits (Lyakhov 2017)</i> .....	27
<i>Figure 11 A comprehensive introduction to grids in web design (Wassermann 2016)</i> .....	28
<i>Figure 12 Handoffs guide for pixel perfect design part 2 (Kamelsky 2017)</i> .....	29
<i>Figure 13 Various tools with their design handoff capabilities (Mohammed 2017)</i> .....	30
<i>Figure 14 Trello vs Asana - The best project management app for 2018 (Zivkovic 2018)</i> .....	31
<i>Figure 15 do you feel like you are doing the same work over and over again, or have you noticed a decrease in repetition after the checklist?</i> .....	32
<i>Figure 16 The checklist has improved everyday work</i> .....	33

## Appendices

Appendix 1: Interview templates .....	41
Appendix 1: Handoff checklist .....	41



## Appendix 1: Interview templates

**1<sup>st</sup> interview questions**

<b>What kind of issues have you encountered in the design handoff?</b>	
<b>Do you feel designs are easy to find?</b>	
<b>Do you feel designs are easy to understand?</b>	
<b>Is it clear what needs to be done?</b>	
<b>Are visual specifications clear and easy to follow?</b>	
<b>Are all required assets easy to find?</b>	
<b>How do you feel about current design handoff?</b>	

**2<sup>nd</sup> interview questions**

<b>Do you feel like you are doing the same work over and over again, or have you noticed a decrease in repetition after the checklist?</b>	
--	--

<b>Do you feel like you're communicating more or less after the checklist?</b>	
<b>How has the checklist affected your every-day work?</b>	
<b>Have you noticed something missing in the checklist?</b>	

## Appendix 2: Design handoff checklist

1	Discovery process	Yes	No
1.1	Get developers involved from the very beginning		
1.2	Invite developers or at least lead developer to be part of user interviews when possible		
1.3	Share user interview finding with developers		
1.4	Discuss user personas with developers in the beginning		
1.5	Consult technical restraints for designs in advance with developers and get their ideas		
2	Planning process	Yes	No
2.1	Discuss user stories with lead developer before the sprint kick-off		
2.2	Let developers know what's coming in the next sprint		
2.3	Check with developers if they will be using some frameworks before designing the UI		
2.4	Check browser support with developers		
3	Concept sharing (prototypes)	Yes	No
3.1	Share low fidelity prototypes with developers to check the feasibility of a feature and get their feedback after every iteration		
3.2	Design your mock-ups for extreme viewports to test out how they will look. For example, design for mobiles and large displays to test responsiveness		
3.3	Use rough contents in your mock-ups instead of lorem ipsum		
3.4	Get developers involved in user testing sessions and share finding with all developers		

3.5	Make sure your cover all interaction states in prototypes for example, error or success states and transitions between states.		
<b>4</b>	<b>User Interface design</b>	Yes	No
4.1	Name your artboards according to screen functionality, avoid random names like “Newest” and “Latest”. Share latest designs and deprecate old ones.		
4.2	Make sure the designs meet user requirements and user personas		
4.3	If you are using something twice in your UI then make it a pattern or a component (for example menus, buttons, cards etc.) to create consistency		
4.4	Let developers decide image formats and sizes		
4.5	Share your grid system with developers and design mock-ups for all breakpoints		
4.6	Use web-safe fonts and whole font values to preserve font integrity		
4.7	Make sure everything that you are using in your web application is yours and you have the copyrights for everything		
4.8	Make sure you have redlined all your mock-ups before sharing		
4.9	Add comments on your mock-ups so developers can understand them properly		
4.10	Delete all extra layers before sharing		
4.11	Group and name all layers according to UI modules (navbar, footer etc)		
4.12	Make sure all assets are exportable		
4.13	Go through all mock-ups, prototypes, interactions, states and related documents while handing off designs to developers		
4.14	Index your designs so that they are easy to find		
<b>5</b>	<b>Share all related documents</b>	Yes	No

5.1	Personas		
5.2	Story boards		
5.3	Interaction flows		
5.4	Mind maps		
5.5	Copy		
5.6	Checklist for developers what needs to be done		