User Experience Certified Ambassador (UXCA) Syllabus

Version: 1.0
Released: September 2022

Copyright Notice

Thank You to the Main Authors

- Joi Arnaiz and Sjoerd Walinga

Thank You to the Co-Authors

- Jenna Charlton

Thank You to the Review Committee

- Alexis Jesús Herrera Colmenares, Anne Vroegop, Arjan Brands, Daniel van der Zwan, Deanna Sim, Gustavo Terrera, Ignacio Rocca, Jayapradeep Jiothis, Jeff Pierce, Jennie Jariel, José Díaz, Juan Pablo Rios Alvarez, Julie Gardiner, Kyle Alexander Siemens, Lisandra Armas Águila, Manuel da Silva Joaquim, Márcia Araújo Coelho, Miguel Angel de León Trejo, Nadia Soledad Cavalleri, Nicholas Terpolilli, Rik Marselis, Rogier Ammerlaan, Sandra Sanchez-Gordon (PhD.), Thomas Cagley, Vanessa Islas Padilla, Viepul Kocher, Wim Decoutere

Revision History

Version	Date	Remarks
0.01	December 2021	Initial Beta release – only for Review Committee
1.00	September 2022	First Public Release

Table of Content

Business Outcomes

Business objects (BOs) are a brief statement of what you are expected to have learned after the training.

BO-1	Understand the important role that user experience (UX) has in the creation of (user-centered) software.
BO-2	Understand the aspects of UX and its relation to customer experience (CX).
BO-3	Understand how to apply different methods to help your customers experience functional software that contribute to your business goal.
BO-4	Learn how to analyze user data in order to generate effective user personas.
BO-5	Learn how to apply UX to better evaluate software and generate assessments to improve software quality overall.
BO-6	Understand your users' needs and expectations so that you can create software that satisfies them.
BO-7	Learn methods for finding usability issues early (in the production process) and benefit from lower development costs and time.
BO-8	Learn how to plan, conduct, and analyze usability tests in all stages of software development to achieve key insights and valuable feedback.
BO-9	Understand how to apply different test techniques to identify bottlenecks and opportunities to (continuously) improve your software.
BO-10	Understand how to convert observations into findings and conclusions and summarize them in a usability report.
BO-11	Understand how to communicate insights to team members and stakeholders to make well-considered design choices to create successful software.

Learning Objectives/Cognitive Levels of Knowledge

Learning objectives (LOs) are brief statements that describe what you are expected to know after studying each chapter. The LOs are defined based on Bloom's modified taxonomy as follows:

Definitions	**K1 Remembering**	**K2 Understanding**	**K3 Applying**
Bloom's definition	Exhibit memory of previously learned material by recalling facts, terms, basic concepts, and answers.	Demonstrate understanding of facts and ideas by organizing, comparing, translating, interpreting, giving descriptions, and stating main ideas.	Solve problems to new situations by applying acquired knowledge, facts, techniques, and rules in a different way.
Verbs (examples)	Remember	Summarize	Implement

	Recall Choose Define Find Match Relate Select	Generalize Classify Compare Contrast Demonstrate Interpret Rephrase	Execute Use Apply Plan Select

For more details of Bloom's taxonomy please, refer to **[BT1]** and **[BT2]** in References.

Hands-on Objectives

Hands-on objectives (HOs) are brief statements that describe what you are expected to perform or execute to understand the practical aspect of what you learn. The HOs are defined as follows:

- HO-0: Live view of an exercise or recorded video.
- HO-1: Guided exercise. The trainees follow the sequence of steps performed by the trainer.
- HO-2: Exercise with hints. Exercise to be performed by the trainee, utilizing hints provided by the trainer.
- HO-3: Unguided exercises without hints.

Prerequisites

Mandatory

- None

Recommended

- To be done, if required

General Notes

To maintain the flow in the text of this document and courseware, the authors may refer to:

- "**Software,**" in some places where "Product, Service and/or System" is intended.
- "**Participant**" can refer to a subject matter expert or a representative user.
- **"a11y"** is the shorthand for "accessibility." The "11" represents the count of letters between the letter a and the letter y.

This syllabus is divided into three chapters:

- Chapter 1 covers the main fundamental aspects surrounding UX.
- Chapter 2 takes the information gained in chapter 1 and focusses on the importance of usability reviews and analyzing the results.
- Chapter 3 builds upon the information gained in the first two chapters and focusses on usability testing from the creation until the finalization and communication of findings in a usability report.

Chapter 1: Introduction to UX, Usability, Accessibility

Keywords

A/B Testing, Accessibility(a11y), Archetypes, Cognitive Walkthrough, Conformance Levels, Customer Experience (CX), Expert Review, Formative and Summative Evaluations, Formative and Summative Research, Heuristics, Heuristic Evaluation, Human-Centered Design, Human–System Interaction, Inclusive Design, Informal Review, ISO Standards (9241-210), Low and High-Fidelity Prototypes, Organizational Risks, Product Risks, Risk-based Thinking, Risk matrix, Subjective and Objective Insights, Universal Design, Usability, Usability Reviews, Usability Test, Usability Testing (Moderated and Unmoderated), User Experience (UX), User Experience Honeycomb, User Personas (Ad-hoc Persona, Buyer Persona, Disabled Persona, Proto Persona, etc.), Web Content Accessibility Guidelines (WCAG)

Learning Objectives for this Chapter:

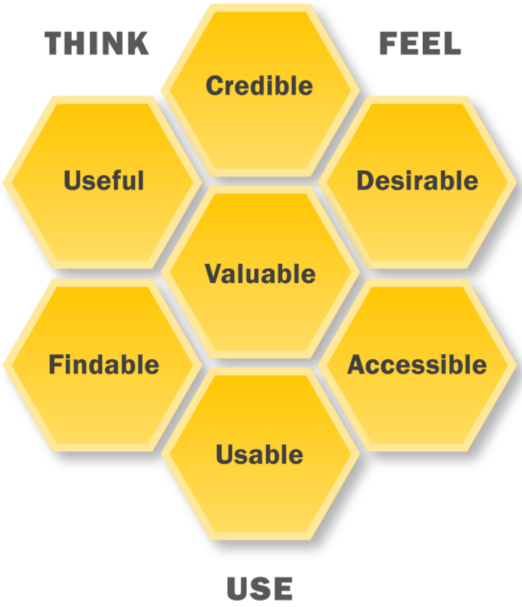
LO1.1	K1	Remember the main aspects we should have in mind while thinking about user experience (UX) and the value we provide to our users.
LO1.2	K2	Understand the general content and applicability of UX, usability, and accessibility (a11y) ISO standards.
LO1.3	K2	Understand the intention, implications, and applicability of WCAG and other regional requirements.
LO1.4	K1	Identify the guiding principles of WCAG.
LO1.5	K2	Understand the creation and implementation of user personas.
LO1.6	K1	Learn UX from the perspective of disabled personas.
LO1.7	K1	Remember the diversity and intersections of disability.
LO1.8	K2	Understand the difference between quantitative and qualitative insights, as well as the differences between subjective and objective observations.
LO1.9	K1	Remember the importance of having qualitative information to achieve better problem solving.
LO1.10	K1	Remember the importance of having a clear objective before choosing the method of gathering information.
LO1.11	K1	Remember why it is crucial to be objective while analyzing information.
LO1.12	K1	Remember the most common methods of obtaining insights and when to use them.
LO1.13	K1	Recall what you can test in the different project stages.
LO1.14	K2	Understand how to select the most appropriate method(s) for verifying and validating UX, usability, and accessibility in a given project stage.
LO1.15	K1	Remember the difference between formative and summative evaluation.
LO1.16	K2	Understand how risk-based thinking can be applied to a given project
LO1.17	K3	Apply useful, usable, findable, credible, accessible, desirable, and valuable concepts as questions to evaluate UX.
LO1.18	K2	Understand how design influences usability, and vice versa.
LO1.19	K1	Remember basic principles of universal design and inclusive design.
LO1.20	K2	Understand typical risks in UX, usability, and accessibility.

1.1 Overview of User Experience

LO1.1	K1	Remember the main aspects we should have in mind while thinking about user experience and the value we provide to our users.

User experience (UX) plays a vital role in the adoption of and loyalty to applications and websites. When a user faces an application or website, they should enjoy a useful, usable, desirable, findable, accessible, credible, and valuable experience. If these criteria are not met, the user will have a bad experience and will avoid using the application or website since these deficiencies are often associated with poor software quality.

This "user experience honeycomb" explains the various facets of user experience design. It was created by Peter Morville in 2004 and optimized by Katerina Karagianni in 2018 by grouping the 7 facets based on how the user interacts with a product (use, think, feel).



The 7 concepts are defined as follows:

Concept	Definition
Credible	The company and its software need to be trustworthy.
Desirable	The visual aesthetics of software need to be attractive and should encourage interaction. Design must support ease of use and there must be a balance between the graphic design and functionality.
Accessible	The software should be designed so that users with disabilities have the same user experience as others. This includes people who have visual, motor, auditory, speech, or cognitive disabilities.
Useful	The software needs to fulfill a need and serve a purpose. If the software is not useful or does not fulfill the user's wants or needs, then there is no real purpose for the product itself.
Findable	Information needs to be findable and easy to navigate. The navigational structure should make it clear where the user is, what the user can do or find where they are located, and where the user can navigate.
Usable	Software should be designed so that it is familiar and easy to understand and use. Any learning curves the user must go through should be as short and painless as possible.
Valuable	The software needs to provide value to the user, making life more efficient, effective, and/or pleasurable.

We define UX as the process by which people interact with software. UX considers the user's personal experience with a given interaction, taking place mostly in the digital world (websites, applications, and software). On the other hand, it is important to differentiate UX from **customer experience (CX)**. CX encompasses all the interactions people have with a brand, and includes all the brand's possible by-products, considering multiple contact channels—not only digital ones.

It is essential to know the difference between UX and **usability** and their relationship. Usability is the measure or ease in which specific users use a product to achieve specific objectives with **effectiveness**, **efficiency**, and **satisfaction** within a particular context of use. The three objectives are defined as:

Objective	Definition
Effectiveness	The accuracy and completeness to which users achieve specified goals. It is about whether users can complete a task and if so, to what extent.
Efficiency	The resources used in relation to the results achieved. It is about how fast users can complete a given task, and with what level of effort.
Satisfaction	The user's physical, cognitive, and emotional response to the use of the software.

1.2 ISO Standards for UX, Usability and Accessibility

LO1.2	K2	Understand the general content and applicability of UX, usability, and accessibility (a11y) ISO standards.

The International Organization for Standardization (ISO) publishes standards for just about everything imaginable in the world, including for UX, usability and accessibility. These standards are internationally composed and agreed upon by experts to make software better, and the standards are aligned to make companies, governments, and other organizations more efficient **[R1].**

1.2.1 ISO standard for UX and usability

UX and usability are both part of the **ISO 9241-210** "Ergonomics of human-system interaction." This ISO standard provides requirements and recommendations for **human-centered design** principles and activities throughout the life cycle of computer-based interactive systems. It is concerned with ways in which interactive systems can enhance **human–system interaction**. More information about this ISO can be found on the official website: https://www.iso.org/standard/63500.html.

1.2.2 ISO Standard for Accessibility

Accessibility, also known as a11y, is also part of the ISO 9241-210 "Ergonomics of human-system interaction," but is only partially defined. Therefore, it is recommended to use the **Web Content Accessibility Guidelines (WCAG)** to meet accessibility standards. These guidelines are designed to make software more accessible to everyone. Since the WCAG guidelines are the best known and because they are internationally accepted, these guidelines are explained in more detail in chapter 1.3.

If your company is highly ISO-centric, you may also want to consider the ISO standards ISO/IEC GUIDE 71, ISO/IEC 30071, and/or ISO/IEC 40500 that are specified for accessibility.

1.3 WCAG

LO1.3	K2	Understand the intention, implications, and applicability of WCAG and other regional requirements.
LO1.4	K1	Identify the guiding principles of WCAG.

The WCAG is the formal, internationally accepted set of guidelines to which every application and website is required to conform. WCAG is constantly evolving, with updates and revisions released periodically to refine and expand the guidelines. As of this writing, the current WCAG version is 2.1, with 2.2 scheduled to be published sometime during 2022.

WCAG conformance is measured in **conformance levels**; these levels are **A, AA, and AAA**. For each guideline, there is an established and measurable **success criteria** required to meet the desired conformance level. Most organizations strive to achieve AA conformance level. Note that not all guidelines have all conformance levels possible; for instance, guideline 1.1 (Text Alternatives) only has level A conformance.

Conformance Level	Level of Requirement	Explanation
A	Must	This is the minimum level that **must** be achieved to be compliant to WCAG standards.
AA	Should	This improves accessibility and attempts **should** be made to achieve this level of success.
AAA	May	This is the highest level of conformance and, while not required, will improve the experience for disabled users.

WCAG is divided into **four principles** that we identify using the acronym **POUR**. These principles represent what is required to make applications and websites usable to the widest possible range of users and with **assistive technology**.

Principle	Explanation
Perceivable	Information and interface must be presented so that a user can perceive it by at least one of their senses.
Operable	User interface, components, and navigation must be operable by a method a user can employ.
Understandable	All information on the application or website, as well as the operation of user interface, must be understandable to users of all ability levels.
Robust	Content must be robust enough so that it can be reliably interpreted by a wide range of assistive technologies and user agents.

Note that WCAG on its own is not an enforceable guideline; individual countries and regions enforce WCAG guidelines via their own laws and regulations. For instance, in the USA, WCAG is enforced through a series of laws and federal guidelines including the Americans with Disabilities Act. It is important to research the locally specific enforcement of WCAG and other requirements that may be present for digital accessibility in your region.

WCAG is a set of guidelines for the development of accessible applications and websites, but the guidelines alone do not ensure you have built an application that is fully accessible. It is possible to follow every guideline and still have an inaccessible application or website because of a lack of usability. Later in the syllabus, we will discuss options for testing and labs to measure the level of usability and accessibility for your website or application; be sure to make use of these options so that guidelines are met with their intentions in mind **[R2].**

1.4 User Personas and Archetypes

LO1.5	K2	Understand the creation and implementation of user personas.

1.4.1 User Personas

"User personas" is a useful tool that allows us to create a story around the end users of the system and empathize with these users. They are a representation of your most realistic user based on user and market research and what is already known about them. In other words, it is the target audience.

Alan Cooper, the father of the term "user persona" **[R3],** proposes creating user personas based on information from real users or your target audience. User personas can be generated from existing information sources or from data gathered via interviews done in person or remotely. If your software does not have real users yet, you can conduct interviews with your target audience to investigate your assumptions and then create user personas.

When creating a user persona, you can start by giving it a name and an identity, placing it within a specific context, and considering their needs. This way, you will be able to consider what type of content adds value for your users and helps them address those specific needs. It is crucial to understand how to connect with the users and understand their challenges, questions, and pain points in order to fully understand their responsibilities, concerns, and potential frustrations. This process will help you deliver a unique and meaningful experience.

Among the data or characteristics that need to be defined for our user persona, we can consider:

A. Personal and/or professional data
B. Objectives
C. Challenges and problems
D. Solutions to those problems
    - How do we solve the user's problem?

Note that a user persona is not the same as a "**buyer persona**"; in many cases they represent different perspectives. A user persona provides insight into how a user operates a system, while a buyer persona provides insight into how a potential customer makes decisions for a purchase. This means that the goals and needs of a buyer persona will differ from those of a user persona.

1.4.2 Archetypes

According to Carl Jung, psychoanalyst and creator of the term, "**archetypes**" are constructed based on our ancestors' experiences and memories. This implies that we do not live and develop isolated from the rest of society, but rather that the cultural context influences us intimately, transmitting thought patterns and a specific set of behaviors through which we experience reality.

If we focus on the individual level, we can see archetypes as observations of emotional and behavioral patterns that shape an individual's way of processing sensations, images, and perceptions as a meaningful whole. When creating user personas in the design process, archetypes can help us connect emotionally with the user. Models like the archetypes proposed by Carl Jung can be inspirational in understanding why different users experience a product in different ways.

1.4.3 Ad-hoc Persona versus Proto Persona

Ad-hoc Persona	Proto Persona
Ad-hoc persona is an elaboration you and your team members create with your knowledge of users. One of the most important things about this document is that it should match the data you have on your users.	A proto persona is a description of the target users based on the assumptions of stakeholders. You can draw a quadrant and fill it with the user information you feel is necessary for the problem you are trying to solve.

1.5 Accessibility Personas

LO1.6	K1	Learn UX from the perspective of disabled personas.
LO1.7	K1	Remember the diversity and intersections of disability.

Personas for accessibility are similar to user personas with one primary difference: they are created for and categorized based on users with disabilities. To create meaningful personas for accessibility, it is important to learn about the disabled community and collect information from disabled users to identify how their requirements differ from those of non-disabled users.

There are three primary categories of disability: physical, visual/auditory, and cognitive/neurological. Many disabled people may have challenges from more than one of these categories. For instance, a user that has Multiple Sclerosis may have restricted mobility, experience brain fog, and have compromised vision.

The following table contains some examples or the three primary categories:

Physical	Visual/Auditory	Cognitive/Neurological
Limb Difference	Blind	Autism Spectrum
Tremors	Deaf	ADD/ADHD
Paralyzed	Hard of Hearing	Mental Illness

This is only a small sample of some cases as an example. An exhaustive list can be found here: https://www.w3.org/WAI/people-use-web/abilities-barriers/ **[R4]**

It is important to include the disabled community in the creation of personas for accessibility. This can be accomplished by interviewing disabled users, using surveys, and observing the ways disabled users interact with your application in labs. However you choose to develop your accessibility personas, it is critical that you remember that nothing should be created for disabled users without including disabled users and special interest groups and communities in the conversation.

1.6 Gathering Information

LO1.8	K2	Understand the difference between quantitative and qualitative insights, as well as the differences between subjective and objective observations.
LO1.9	K1	Remember the importance of having qualitative information to achieve better problem solving.

Improving usability can be done by conducting **formative** and **summative** research and it can yield **subjective** and **objective** insights.

1.6.1 Formative Research

Formative research focuses on determining which aspects of the design work well and which do not. It is usually conducted during an early stage of product development (for example, the design and prototyping stages) to test expectations, to generate ideas, and to help form the product's shape and design. The goal is to identify and understand design and usability issues before the product enters development.

Formative research is more qualitative in nature. It lets you see how users actually experience the design, see where and why they get stuck, and hear what they say when using the "think-out-loud" method (in which participants are asked to continuously verbalize their thoughts while they use the system).

Qualitative research helps to collect qualitative data through the direct observation and study of participants. **Qualitative data** yields an understanding of the motivations, thoughts, and attitudes of people and the behaviors that result. This type of research is key to uncovering the "why" behind actions and developing a deep understanding of a topic or problem.

1.6.2 Summative Research

Summative research describes how well a design performs. It is usually performed later in the software development process, such as just before, during, or after development to measure the usability of a component or the whole software. Summative research focuses on obtaining measurements of the effectiveness and efficiency of software, as well as user satisfaction. It can be used to evaluate a design based on usability requirements so that the acceptability of the design can be established from the user's point of view.

Summative research is quantitative in nature. It can act as a final validation after usability issues have been identified and addressed.

Quantitative research is used to collect and analyze numerical data, identify patterns, make predictions, and generalize findings about a target audience or a topic.

Brief explanation of the differences between formative and summative research:

- Formative research methods are used to discover new information about user goals and motivations and better understand their behavior.
- Summative research methods are used for assessing a specific software solution to ensure it is easy to use and works as intended.

Using both formative and summative research during the design process ensures the software you are building addresses real needs, solves existing pain points, and does both in the most viable way.

1.6.3 Subjective Insights

Subjective Insights are judgements based on personal opinions, feelings, and/or points of view. This information comes from the user's attitude and is based on what they ***say***.
For example: *"The participant experiences the website as credible."* (you can **hear** the user **saying** …)
Subjective information can include emotional and factual data.

- Emotional data is a judgment formed about something.
  - For example: *"I think this website is credible because it looks professional."*
  - The word "professional" is not measurable and can be different for everyone.
- Factual data is something that is known or proved to be true.
  - For example: *"I think this website is credible because my purchase is insured and affiliated with quality marks."*
  - The presence of the insurance and the quality marks can be checked, and are therefore measurable. These factors are the same for everyone.

1.6.4 Objective Insights

Objective Insights are facts based on measurements and observations. This information comes from user behavior and is what the user ***does***.
For example: *"After a pop-up appeared, the participant tried to click it away by clicking next to the window. When that failed, the participant clicked the 'close' icon in the window."* (you can **see** the user **doing** …)

This is a continuous learning process that uncovers new insights to help you create value for your customers.

- Subjective insights come from what users *say*
- Objective insights come from what users *do*

Objective insights will tell you what is happening, while subjective insights help to provide the reason why it is happening. User attitudes and user behavior are often quite different; what a user says is often different from what a user does. Therefore, it is (strongly) recommended to combine different research methods.

1.7 Defining the Objective and Scope

LO1.10	K1	Remember the importance of having a clear objective first before choosing the method of gathering information.
LO1.11	K1	Remember why it is crucial to be objective while analyzing information.

Just as it is important to have a clear test **objective** when creating functional test cases, it is equally important to have a clear test objective when creating and executing a usability test. The stated objective helps to identify the appropriate **scope** of the test and focus on the things that matter, ensuring relevant and useful insights. The objective determines the **methodology** that will be used for the previously mentioned usability test and helps to maintain objectivity when analyzing observations.

1.8 Introduction of Usability Reviews and Usability Testing

LO1.12	K1	Remember the most common methods of obtaining insights and when to use them.

Conducting usability reviews and usability testing are the two most common methods of evaluating software usability.

1.8.1 Usability Reviews

Usability reviews are a structured usability inspection method of examining the usability of software by evaluating it against a set of recognized usability best practice principles. The reviews are generally performed on interfaces such as screenshots, prototypes, live websites, apps, audio recordings, etc.

The purpose of a usability review is to inspect software and identify potential usability problems and deviations from established criteria. They are usually performed by one or more usability experts, UX professionals, design team members, testers and/or other subject matter experts. Useful link for more information: https://www.nngroup.com/articles/ux-expert-reviews/ **[R5]**.

Broadly speaking, there are two different types of usability reviews: heuristic-based reviews and scenario-based reviews.

- In a heuristic-based review, you evaluate an interface against a set of usability best practice principles (e.g., heuristics, standards, guidelines, etc.) such as making a clear distinction between mandatory and optional input fields in forms.
- With a scenario-based review you evaluate software against defined user scenarios such as finding information about a specific product or requesting a quote.

These two review types can be applied independently but are most effective when combined.

The most common usability review methods are:

- Heuristic evaluation
- Expert review
- Cognitive walkthrough
- Informal review

The following table provides information about the four commonly used usability review methods:

Review type	Formal or Informal*	Mainly based upon...
Heuristic evaluation	Mainly done formally	Best practice principles (e.g., heuristics, standards, and usability guidelines)
Expert review	Mainly done informally	Opinions, personal experience, and common sense
Cognitive walkthrough	Can be done formally or informally	Scenarios and/or series of tasks
Informal review	Mainly done informally	Feedback from participants

** Note that “Formal” and “Informal” refer to an execution with or without (respectively) planning and/or following guidelines and rules*

Other common usability review methods, which will not be discussed in this course, are: pluralistic walkthrough, heuristic walkthrough, persona-based inspection, feature inspection, standards inspection, and formal usability inspection.

1.8.1.1 Heuristic Evaluation

A heuristic evaluation is an inspection method that is often employed to evaluate the usability of the software. This type of evaluation involves one or multiple evaluators and is performed by using accepted fundamental usability principles: the heuristics. Software should at least meet these principles in order to be labeled “user-friendly.”

A heuristic evaluation has a low barrier of entry and can be performed by anyone, regardless of their level of experience or knowledge. All the evaluator has to do is go through a list of heuristics and identify where the software does not follow those principles.
Heuristics that can be used when inspecting software are covered in more detail in chapter 2.

1.8.1.2 Expert Review

An **expert review** - carried out by one or more UX experts - is a great method of analyzing a product or service’s usability. Expert reviews are the best option when the examiner has a deep knowledge of usability best practices and a large amount of experience conducting usability research and is not involved in creating the design to be reviewed.

Carrying out an expert review helps to identify areas needing attention and catches “obvious” issues that should be fixed before conducting a usability test. Expert reviews can be done at any stage of the product development process. However, conducting them during the initial phase, when the user interface is still a draft, is recommended as the design could be easily modified at this stage.

1.8.1.3 Cognitive Walkthrough

A **cognitive walkthrough** is comparable to a heuristic evaluation. The difference is that a heuristic evaluation focuses on the entire software (or a part of it), while a cognitive walkthrough focuses on scenarios and tasks, mainly focusing on the paths and steps that users take to achieve their goals.

Prior to conducting a cognitive walkthrough, the evaluator must first identify which goals users have and what they want to achieve with the software.

The evaluator then goes through each step, identifying problems users might encounter as they learn to use the interface.

1.8.1.4 Informal Review

As the word informal suggests, an **informal review** does not require experts apart from the facilitator or an adherence to guidelines and rules. Conducting an informal review is attractive since it is an inexpensive way of getting more feedback quickly into the design process.

The steps are very simple: find a room and invite your team or, if you can, a real user. You can invite as many people as you like. Then you should guide them through the designs and if possible, highlight some of the issues that were found in your previous usability test. Naturally, the participants would start talking about what they see. If not, you can ask probing questions to encourage them to answer.

1.8.2 Usability Testing

Usability testing is a method of evaluating the degree to which software can be used by specified users. During a usability test, participants are asked to perform a set of tasks. These are tasks that people should be able to perform with the software in a specified context of use.

Broadly speaking, there are two different approaches for usability testing: moderated and unmoderated testing.

- **Moderated usability testing** requires the active participation of a facilitator (or “moderator”). This facilitator guides the test participant through the testing process. It can be done either in person (e.g., in a lab environment) or remotely (e.g., with software that supports screen sharing).
- **Unmoderated usability testing** is conducted by test participants in their own environment without the presence of a facilitator. This approach is primarily done with online (automated) usability testing tools.
- These approaches are explained in more detail in chapter 3.

The most common usability testing methods are:

- In-lab usability testing (with or without eye-tracking technology)
- Card sorting
- A/B testing

The following table shows how most common usability testing methods are implemented:

Method	Validation with	Type (in nature)	Key characteristics	When?
In-lab Usability Testing	Users	Qualitative	Users are observed while they perform a set of tasks with the software to reveal bottlenecks or uncover opportunities for improvement.	In all project stages
Card Sorting	Users	Open card sort: Qualitative Closed card sort: Quantitative	Users are observed while they organize topics into categories that make sense to them. In an open card sort, participants need to give these categories a category name that makes sense to them. In a closed card sort, these category names are already defined. Card sorting helps to understand how users structure information and helps to design an information architecture that matches user expectations.	In all project stages
A/B Testing	Real time traffic	Quantitative	Users are randomly assigned to different groups. Each group is shown a different variant of the user interface to reveal with which version the highest number of users are able to achieve their goal. Users are not aware they are participating.	Only possible when the website is live*

A condition for A/B testing is that you must have at least 1000 unique sessions per week for statistical outcomes. You can calculate the exact number of unique sessions that you need on this website: https://abtestguide.com/abtestsize/* **[R6]

Other common usability testing methods, which will not be discussed in this course, are: preference testing, tree testing, mouse tracking, form analytics, and multivariate testing.

The goal of usability testing is to reveal areas of confusion and bottlenecks and uncover opportunities to improve the overall user experience regarding the effectiveness and efficiency of the software, and user satisfaction (see chapter 1.1). Usability tests are usually performed by a facilitator with solid experience in the field of usability testing and user research.

1.8.3 What to Review and/or Test in each Project Stage

LO1.13	K1	Recall what you can test in the different project stages.
LO1.14	K2	Understand how to select the most appropriate method(s) for verifying and validating user experience, usability and accessibility in a given project stage.

Usability reviews and usability testing can be done in different project stages. They are often conducted repeatedly, from early design until a product's release.

The following table is an overview of what you can review and/or test:

Project stage	What?	How?	Why?
Before starting	● Current software ● Competitor’s site	Usability review	Gain insights from an **expert perspective** if elements are **designed and developed** well. **Learn from bottlenecks** in fundamentals, based on best practice principles, to **prevent** these problems from occurring during the design and development of the new software.
		Usability test	Gain insights, from a **user’s perspective,** into **well-functioning** elements, **reveal areas of confusion and bottlenecks**, and/or **uncover opportunities** to **improve or solve** these problems in the new software.
During design stage	Low fidelity prototype *(see chapter 1.8.4)*	Usability review	Gain insights, from an **expert perspective**, into whether elements are **designed** well. **Identify (potential) bottlenecks** in fundamentals, based on best practice principles, to **prevent** these problems in the **high-fidelity prototype** of the new software.
		Usability test	Gain insights, from a **user’s perspective**, into the degree of **findability of information**, and/or user **expectations** for **functionalities**.
	High fidelity prototype *(see chapter 1.8.4)*	Usability review	Gain insights, from an **expert perspective**, into whether elements are **designed** well. **Identify (potential) bottlenecks** in fundamentals, based on best practice principles, to **prevent** these problems during the **development** of the new software.
		Usability test	Gain insights, from a **user’s perspective**, into the degree of **findability of information**, and/or user **expectations** for **elaborated functionalities**, and (first) **impression of the design**.
During development stage	● User stories and tasks ● Pre-launch	Usability review	Gain insights, from an **expert perspective**, into whether elements are **developed** well. **Identify (potential) bottlenecks** in fundamentals, based on best practice principles, to **prevent** these problems in the **release** of the software.
		Usability test	Gain insights, from a **user’s perspective**, into **well-functioning** elements, **reveal areas of confusion and bottlenecks**, and/or **uncover opportunities** to **prevent** these problems in the **release** of the software.
After release	Newly launched software	Usability review	Gain insights, from an **expert perspective**, into whether elements are **designed and developed** well. **Reveal (potential) areas of confusion and bottlenecks**, and/or **uncover opportunities** to **improve or solve** these problems in a new version.
		Usability test	Gain insights, from a **user’s perspective**, into **well-functioning** elements, **reveal areas of confusion and bottlenecks**, and/or **uncover opportunities** to **improve or solve** these problems in a new version.

1.8.4 Low and High-Fidelity Prototypes

In the context of digital solutions, a prototype is a draft version of software. It is a simulation of the final design and the interaction between the user and the interface. Prototyping allows for the testing of a simulation of the entire software or just a single interaction, as well as for (further) elicitation of requirements. These processes make it possible to solve problems before time and money is invested in development.

Prototypes can be developed with varying degrees of detail and at different levels of functionality. This level of detail and functionality is described as the “fidelity”. In other words, the fidelity of the prototype refers to how closely it matches the look, feel and functionality of the final software.

The fidelity of a prototype can vary in areas of:

- Interactivity
- Visuals
- Content

Within these areas, a prototype can fall anywhere in the range between these two extremes:

- Low-Fidelity: mostly used to convey ideas and to show alignment with the stakeholders.
- High-Fidelity: mostly used to demonstrate what the actual experience will be like and is close to the final design.

The following table below explains what high and low fidelity means in each of these areas:

	Low-Fidelity Prototype	**High-Fidelity Prototype**
	Interactivity	
Clickable links and menus	No: targets do not work.	Yes: many or all are clickable.
Automatic response to user actions	Yes: links in the prototype are made to work via a prototyping tool (e.g., InVision, PowerPoint).	No: screens are presented to the user in real time by a person playing “the computer.”
Paper or digital	Often paper-based	Often computer-based (digital)
	Visuals	
Realistic visual hierarchy, priority of screen elements, and screen size	No: only some or none of the visual attributes of the final live system are captured (e.g., a black-and-white sketch or wireframe, schematic representation of images and graphics, single sheet of paper for several screenfuls of information). Spacing and element prioritization may or may not be preserved.	Yes: graphics, spacing, and layout look like a live system would look (even if the prototype is presented on paper).
	Content	
Content	No: the prototype includes only a summary of the content or a stand-in for product images.	Yes: the prototype includes all the content that would appear in the final design (e.g., full articles, product description text, and images).

Source: https://www.nngroup.com/articles/ux-prototype-hi-lo-fidelity/ **[R7]**

1.8.5 Usability Reviews versus Usability Testing

Both **usability reviews** and **usability testing** are useful for identifying usability problems that can arise in an interface during the design and development process.

Since both methods help to detect and resolve bottlenecks early, it is recommended to apply both methods throughout the entire process. However, this is not always possible due to factors such as lack of time and/or money.

You may find yourself in a situation where you must choose between the two methods.

The following table below shows the main differences between usability reviews and usability testing in case it is necessary to make a choice:

	Usability Review	Usability Testing
Insights are based upon …	the knowledge and experience of a subject matter expert.	the behavior of a target user.
The judgment is …	primarily subjective.	primarily objective.
The execution is …	done by going through a set of usability best practice principles and/or defined user scenarios.	done by performing a set of tasks.
The observation level is …	a holistic approach to the software.	a screen-by-screen approach to the software.
The investment is …	relatively low, because it requires only a few activities and there is no budget required for participants.	often higher than a usability review, because it requires multiple activities like working out scenarios, setting up a test environment and recruiting participants (which also leads to a longer lead time), and in most cases there is a budget required for participants.

1.9 Formative and Summative Usability Evaluation

LO1.15	K1	Remember the difference between formative and summative evaluation (K1)

Formative and summative usability evaluations have different goals, and there are differences in how and when you would carry out each type of evaluation.

- **Formative evaluations** tell you what aspects of your designs work or do not work and why. These are frequently used at the beginning of the design process to promote the iteration to a better product. There are many types of formative evaluation methods, such as heuristic evaluation, thinking-aloud testing, usability walkthrough, and cognitive walkthrough. For these methods, you do not need a large number of users: five is more than enough **[R8]**. Learning about what does and does not work in a design enables you to identify features that are well designed and those that need improvement.

- **Summative evaluation** is normally carried out when you have a complete design or a shipped software. Users perform tasks with your software and grade the experience on a predefined scale against predefined criteria such as satisfaction, ease of use, or aesthetics, among other categories. The criteria and scale should be determined by the person that defines the strategy. This evaluation will tell you how usable or satisfying the experience is and can be performed at various points during the product life cycle after the software is launched. However, be aware that to achieve reliable numbers, you need a large representative sample **[R9]**. This takes more time to get the results if your software is not live, since it takes longer to reach a larger number of people. Summative evaluations are not helpful if you want to quickly learn what you need to do to improve the software, but they are useful if you want to see, overall, how your product compares to its competitors or previous design iterations.

1.10 Risk-based Thinking

LO1.16	K2	Understand how risk-based thinking can be applied to a given project.

An organization chooses to conduct usability testing to reduce the **risks** associated with poor usability. Typically, applications with poor usability do not see wide adoption by the potential user community. To reduce this risk, it is critical that we utilize **risk-based thinking** and actively look to identify potential usability problems.

After risk identification it is necessary to develop a system or **scale** to weigh risks, focusing on the **likelihood** and **impact** of a potential usability problem. There is no hard and fast rule for how to measure risk; some teams use a numeric scale, some use a heat map, and others choose to develop a scale using emojis or other visuals. Generally, after risks have been identified and measured, they are tracked using a **risk matrix**. This matrix displays all identified risks with likelihood and impact scores and any supporting documentation created by those involved in risk identification, mitigation, changing, or disappearing. A risk matrix is a living document that will change over time as risks are identified, mitigated, changed, or eliminated. It is important to remember that risk identification and mitigation is not a one-time task; risk is a continuous conversation and the conversation does not end until the application is retired.

Risk-based thinking helps to decide what kind of usability reviewing and/or testing is needed to cover a risk. The effort to be spent on testing is determined based on the risk level. With high risks, it is recommended to do much testing; with low risks, it is recommended to do little testing.

1.11 Design Influences on Usability

LO1.18	K2	Understand how design influences usability, and vice versa.
LO1.19	K1	Remember basic principles of universal design and inclusive design.

A common misconception about UX design is that its sole purpose is to create visually attractive and aesthetically pleasing applications and websites. However, UX design involves much more than just a pretty presentation. The layout, color choices, spacing, font, and more can have a significant impact on the user's experience with the software and their ability to use it. **Universal design** is expressed in seven principles to help make software as it exists in the physical world more usable and pleasing to as many people as possible.

The seven principles of universal design were developed in 1997 at North Carolina State University by a group of professionals from various disciplines and industries, with the intention of creating guidance for the design of environments, products, and communications.

The seven principles of universal design are broken down in the following table:

Principle	Description	Guidelines
Equitable Use	The design is useful and marketable to people with diverse abilities.	1a. Provide the same means of use for all users: identical whenever possible; equivalent when not. 1b. Avoid segregating or stigmatizing any users. 1c. Provisions for privacy, security, and safety should be equally available to all users. 1d. Make the design appealing to all users.
Flexibility in Use	The design accommodates a wide range of individual preferences and abilities.	2a. Provide choice in methods of use. 2b. Accommodate right- and left-handed access and use. 2c. Facilitate the user's accuracy and precision. 2d. Provide adaptability to the user's pace.
Simple and Intuitive Use	Use of the design is easy to understand regardless of the user's experience, knowledge, language skills, or current concentration level.	3a. Eliminate unnecessary complexity. 3b. Be consistent with user expectations and intuition. 3c. Accommodate a wide range of literacy and language skills. 3d. Arrange information according to its importance. 3e. Provide effective prompting and feedback during and after task completion.
Perceptible Information	The design communicates necessary information effectively to the user, regardless of ambient conditions or the user's sensory abilities.	4a. Use different modes (pictorial, verbal, tactile) for redundant presentation of essential information. 4b. Provide adequate contrast between essential information and its surroundings. 4c. Maximize "legibility" of essential information. 4d. Differentiate elements in ways that can be described (i.e., make it easy to give instructions or directions). 4e. Provide compatibility with a variety of techniques or devices used by people with sensory limitations.
Tolerance for Error	The design minimizes hazards and the adverse consequences of accidental or unintended actions.	5a. Arrange elements to minimize hazards and errors: the most used elements should be most accessible and hazardous elements should be eliminated, isolated, or shielded. 5b. Provide warnings about hazards and errors. 5c. Provide fail-safe features. 5d. Discourage unconscious action in tasks that require vigilance.
Low Physical Effort	The design can be used efficiently and comfortably and with a minimum of fatigue.	6a. Allow users to maintain a neutral body position. 6b. Use reasonable operating forces. 6c. Minimize repetitive actions. 6d. Minimize sustained physical effort.

Size and Space for Approach and Use	Appropriate size and space are provided for approach, reach, manipulation, and use regardless of the user's body size, posture, or mobility.	7a. Provide a clear line of sight to important elements for any seated or standing user. 7b. Make reach to all components comfortable for any seated or standing user. 7c. Accommodate variations in hand and grip size. 7d. Provide adequate space for the use of assistive devices or personal assistance.

Source: https://www.udll.com/media-room/articles/the-seven-principles-of-universal-design/

More information about universal design can be found here: http://universaldesign.ie/What-is-Universal-Design/The-7-Principles/ **[R10]**.

Inclusive design is focused on ensuring that we include as many users as possible in our design and development decisions. The focus of inclusive design is ensuring that we are meeting the requirements of WCAG and the needs of our disabled and less skilled users from the inception of building our applications. Often, accessibility and usability are treated as afterthoughts when developing software; inclusive design makes disabled and less skilled users central in the development of software and helps to prevent the need for remediation efforts at the end of the project.

1.12 Risks in UX, Usability, and Accessibility

LO1.20	K2	Understand typical risks in UX, usability, and accessibility.

During a project, it is important to be aware that various product risks can be lurking in your usability, accessibility, and user experience. These risks can strongly influence the overall quality of software; therefore, it is essential to take caution when making design and development-based decisions. Therefore, it is highly recommended that risk assessments be performed and assessed regularly when developing software. These risk assessments reveal potential consequences for the organization if the software has flaws in usability, accessibility and/or UX.

It is crucial to view product risks from the perspective of the product itself, its' users, and the affects on the organization behind it. The user experience honeycomb (see section 1.1) provides the seven categories to which the product risks can be aligned.

1.12.1 Categorization of Product Risks in Relation to the Product Itself

There are seven categories of product risk, and the following points show typical examples for each (in relation to the product):

Credible:

- Users share their negative experience with other potential clients, which diminishes consumer confidence and brand equity.
- Users find that the behavior of the application is inconsistent.
- Users notice inaccurate information, which leads to distrust of the software.

Desirable:

- Users do not use or purchase the software.
- Users are dissatisfied with the software and complain.
- Users who have purchased the software and become frustrated want a refund or cancel their subscription.

Accessible:
- Users feel unsuccessful using the software.
- Users are not able to use devices of different sizes and configurations.

Useful:
- An employer notes that staff cannot work effectively and efficiently with the software and switches to another supplier.
- Users find the scope of the software is too narrow to solve their problem.

Findable:
- Users dislike that the search results are not targeted.
- Users experience the button labels as misleading, resulting in the finding of irrelevant information.

Usable:
- Users will look for a (better) alternative that is easier to use or works more efficiently.
- Users use the software incorrectly without realizing it.
- Users call customer service (unnecessarily) for help because they do not understand how to use the software.

Valuable:
- Users stop using the software before reaching their goal.
- Users dislike using the software and prefer to use it as little as possible.
- Users feel that the cost outweighs the benefits that the software provides.

1.12.2 Categorization of Product Risks in Relation to Its' Users

Here are again the seven categories of product risk, and the following points show typical examples for each (in relation to the user):

Credible:
- The users have the feeling that they cannot rely on the software.
- The users become suspicious and do not feel comfortable using the software.
- If the software does not guide users intuitively, they become insecure or even afraid of losing data.
- The software is not widely known.

Desirable:
- The users consider the software to be slow.
- The users consider the software to be annoying.
- The users consider the software to be unpleasant and/or unsatisfactory.

Accessible:
- More money is spent on customer support and training than necessary.
- People with color blindness and/or low vision have certain issues using the software or experience difficulties such as overlooking elements due to too little contrast.
- Users with certain disabilities are unable to use the software (which violates regulations in some regions and countries).
- The software is not compatible with other software or hardware used by people with disabilities.

Useful:
- The software does not meet the needs of potential users.
- The users lack required functionalities.
- The users consider the software as complicated and not intuitive enough.

Findable:
- The users are not able to orient themselves within the software.

- The users are not able to find the information they are looking for.

Usable:

- The users must concentrate more than necessary when performing a task.
- The users stumble upon bottlenecks when they perform a task.
- The users experience frustration when using the software.
- The software confuses the users.
- The users become afraid of making mistakes.

Valuable:

- The software does not feel like a product that the user needs or that it offers any added value.
- The users perceive the software as a burden rather than a pleasure.

1.12.3 Categorization of Product Risks in Relation to the Organization Behind it.

This is especially essential for companies that exist only in the software, for example SaaS companies.

If users have a negative experiences, then there is a strong likelihood for example, that the software will be used rarely (or not at all), the software will generate a low level of revenue (or none at all), the satisfaction of the app will be low (poor ratings), few people will commit to the software (loss in consumer confidence), the software will get a high rate of abandonment or churn.

Chapter 2: Usability Reviews

Keywords

Accessibility Heuristics, David Travis's Heuristics, Formal and Informal Reports, Heuristic Analysis, Nielson's Heuristics, Usability Report, Usability Reviews

Learning Objectives for this Chapter:

LO2.1	K2	Understand the difference between a heuristic and a rule.
LO2.2	K2	Understand which are the possible usability problems a software could have.
LO2.3	K2	Understand when and why to do a heuristic analysis.
LO2.4	K2	Understand Nielsen's usability heuristics.
LO2.5	K2	Understand the output of the heuristic analysis and the limitations.
LO2.6	HO	Apply Nielsen's 10 heuristics to the formal review.
LO2.7	K1	Remember the purpose of accessibility heuristics.
LO2.8	K1	Remember mandatory topics for informal reviews.
LO2.9	K2	Understand how to carry out an evaluation according to the product you are testing.
LO2.10	K2	Understand what to include in a usability report

2.1 Introduction to Heuristics

LO2.1	K2	Understand the difference between a heuristic and a rule.
LO2.2	K2	Understand which are the possible usability problems a software could have.
LO2.3	K2	Understand when and why to do a heuristic analysis.
LO2.4	K2	Understand Nielsen's usability heuristics.
LO2.5	K2	Understand the output of the heuristic analysis and the limitations.
LO2.6	HO	Apply Nielsen's 10 heuristics to the formal review.

As explained in section 1.8.1, heuristics are accepted fundamental usability principles. Heuristics are often applied in usability reviews.

The 10 heuristics of Nielsen and Molich are the most well-known.

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

Visibility of system status
Match between system and the real world
User control and freedom
Consistency and standards
Error prevention
Recognition rather than recall
Flexibility and efficiency of use
Aesthetic and minimalist design
Help users recognize, diagnose, and recover from errors
Help and documentation

More in depth explanations of each of Nielsen's heuristic can be found here: https://www.nngroup.com/articles/ten-usability-heuristics/ **[R11]**.

Another helpful set of heuristics designed by David Travis. Travis's heuristics are context specific. These guidelines are purposefully expressed as positive statements, so that when you feed the results back to the product team you can identify some strengths of the design before you launch into the problems. The main category of these heuristics or guidelines are:

- Home page usability
- Task orientation
- Navigation and IA
- Forms and data entry
- Trust and credibility
- Writing and content quality
- Page layout and visual design
- Search usability
- Help, feedback, and error tolerance

Help, Feedback & Error Tolerance
Home Page Usability
Task Orientation
Navigation & IA
Forms & Data Entry
Trust & Credibility
Writing & Content Quality
Page Layout & Visual Design
Search Usability

Despite the complete list being dated, many points are still relevant. The entire list can be found here: https://www.userfocus.co.uk/resources/guidelines.html **[R12]**

2.2 Accessibility Heuristics

LO2.7	K1	Remember the purpose of accessibility heuristics.

Much like there are heuristics that help identify best practices in UX testing, there are heuristics that represent best practices in accessibility testing as well. For instance, "keyboard only" is a heuristic representing the MVP (Minimum Viable Product) in accessibility testing. Below is a table with some common accessibility heuristics.

Heuristic	Description	Justification
Keyboard Only	All page elements can be interacted with using keyboard only (no mouse or trackpad). The user never gets caught in a keyboard trap.	Many disabled users will need to interact via keyboard only. It is essential the user be able to access all elements on the page without use of a mouse or trackpad.
High Contrast	All elements on the page are visible when high contrast has been enabled.	Many users, for various reasons, use high contrast mode. It is critical that these users be able to see and access all on-screen elements when high contrast mode is enabled.
Tool	Run an accessibility scan using an audit tool or accessibility testing tool.	Using a tool will help you identify issues quickly to save time in manual testing. Keep in mind use of a tool should never replace completing a visual or manual inspection.

More information about this can be found here: https://karlgroves.com/2013/09/05/the-6-simplest-web-accessibility-tests-anyone-can-do **[R13]**

2.3 Usability Report

LO2.8	K1	Remember mandatory topics for informal reviews.
LO2.9	K2	Understand how to carry out an evaluation according to the product you are testing.
LO2.10	K2	Understand what to include in a usability report

You have 2 types of reports, **formal** and **informal**.

Formal reports should at least include:

- **Background summary:** include a summary of what was tested, where and when the test was performed, equipment needed, what was done during the test, and a brief description of the problems found and what worked well.
- **Goals:** add the purpose of the test, why it was done, and what you wanted to validate or find.
- **Methodology:** include the process that was carried out so that others can later recreate the test. Explain, step by step, all the different activities you did and why.
- **(Test) results:** add an analysis of the (test) results: the tasks that were performed and which ones had the highest and lowest completion rates. Include tables or visuals that will help you understand the results. You can also add revealing comments from the participants.
- **Findings and recommendations:** list all findings and recommendations, providing all information necessary on a case-by-case basis. Each finding or recommendation should include a description of the situation as specific as possible. Most usability reports focus on issues, but they are also useful for reporting positive results. You may not be able to implement all the recommendations. Software development requires a series of trade-offs to balance the schedule, budget, team, and specific features or marketing requests. Set priorities based on solving the most severe and global problems first to improve the user experience.

On the other hand, **informal reports** can be whatever you feel would help you communicate better. It could be a presentation, an email, a video, a board, a poster, etc. You just need to include the information about findings and recommendations. Although this type of report can be fast and get straight to the point, sometimes the lack of other information which gives context could make it difficult for other team members to fully understand the reasons behind some decisions. Also, it could be hard to return to an informal report for insights after some time.

Chapter 3: Usability Testing

Keywords

Accessibility Lab, Accessibility Risks, Contextual Inquiry, Facilitator (or Moderator), Guerilla Testing, Question-Asking Techniques (Boomerang, Colombo, Echo), Note Taker (or Scribe), Observations, Risk-based Thinking, Usability Conclusions, Usability Findings, Usability Risks, Usability Test Plan, Usability Test Report, Usability Test Script, Usability Test Study, Usability Testing Approach, Usability Testing Types (Moderated and Unmoderated), UX Risks

Learning Objectives for this Chapter:

LO3.1	K2	Understand what usability testing is.
LO3.2	HO	Apply appropriate selection method(s) for verifying and validating UX, usability, and accessibility in a given project stage.
LO3.3	HO	Apply risk-based thinking to a given project and identify UX, usability, and/or accessibility risks.
LO3.4	HO	Identify best practices for an accessibility lab given scenarios.
LO3.5	K3	Apply the principal steps in a usability testing approach.
LO3.6	K3	Apply role-based responsibilities to unique roles in a usability test.
LO3.7	K1	Remember a role based on a description.
LO3.8	K2	Understand the content of a usability test plan and the most appropriate usability test scenario or a given study.
LO3.9	K3	Apply mandatory topics for a usability test plan.
LO3.10	K3	Apply a simple usability test task to a given project and put together the key activities for preparing a usability test.
LO3.11	K3	Apply usability test techniques to identify key activities in a usability test session.
LO3.12	K2	Understand questions that are appropriate to use during a usability test study.
LO3.13	K3	Apply a given list of observations to select useful findings, define conclusions, and define classifications.
LO3.14	K2	Understand the relationship between findings and conclusions.
LO3.15	K2	Understand the classifications and ratings for usability findings.
LO3.16	HO	Review a usability test report for a given project.
LO3.17	HO	Apply observations to select useful findings and use these findings to define conclusions and classifications.

3.1 Introduction to Usability Testing

LO3.1	K2	Understand what usability testing is.

Usability testing is a research method of evaluating how software is used by specific users. The goal is to measure how usable a design, flow, or real software can be with users who represent your target audience.

The tests can vary from remote to in-person, moderated to unmoderated. They also depend on the type of information you need (qualitative or quantitative). Designing different tests helps you get a complete picture of user behavior, and details on interaction with the software. Usability testing helps to identify issues or needs you would not have otherwise noticed.

Usability testing can help to:

- … eliminate assumptions and get real data about user experience.
- … identify specific issues with flows that are essential for your product.
- … achieve a better understanding of user needs.
- … save time by understanding what to prioritize, so you can go straight into delivering the experience your users need.
- … gain new insights. During user tests, most of the users interviewed talk about their experiences and opinions, and in many cases, they will mention concerns about the product and give solutions to specific problems.

3.2 Usability Testing Types and Approaches

LO3.2	HO	Apply appropriate selection method(s) for verifying and validating UX, usability, and accessibility in a given project stage.
LO3.3	HO	Apply risk-based thinking to a given project and identify UX, usability, and/or accessibility risks.
LO3.4	HO	Identify best practices for an accessibility lab given scenarios.

There are the two main types of usability testing:

- **Moderated**: Participants are observed and/or interacted with while they complete the tasks for the test. Moderated testing is best suited for complex tests that do not have a structured sequence of steps or tests for which more interaction and questioning would be beneficial.
- **Unmoderated**: Participants independently complete testing without interaction from the test facilitator. Unmoderated testing is more beneficial when you have a particular set of questions about how people use a user interface for straightforward tasks. Usually, these tests are tool-dependent, so you would need a user testing tool to run them.

Options for moderated and unmoderated testing:

- **In-person**: Consider in-person testing anytime you have a user physically present. In-person testing can be beneficial as a facilitator can observe and record the participant’s body language, gestures, and non-verbal cues. Methods of in-person testing include contextual interviews and eye tracking, among others.
- **Remotely**: Remote testing includes any tasks that take place without a facilitator present in the same room as the participant. Remote testing is ideal when your budget is relatively small, or when the target audience is (physically) located in another country.

Remote and in-person tasks can be moderated or unmoderated. Remote unmoderated usability testing is so fast and easy that some teams make it their only method for evaluating their experiences. Still, it would be beneficial to be open to other alternatives as well. For example, remote moderated usability tests are inexpensive and can provide more information because there would be an opportunity for the facilitator to ask questions.

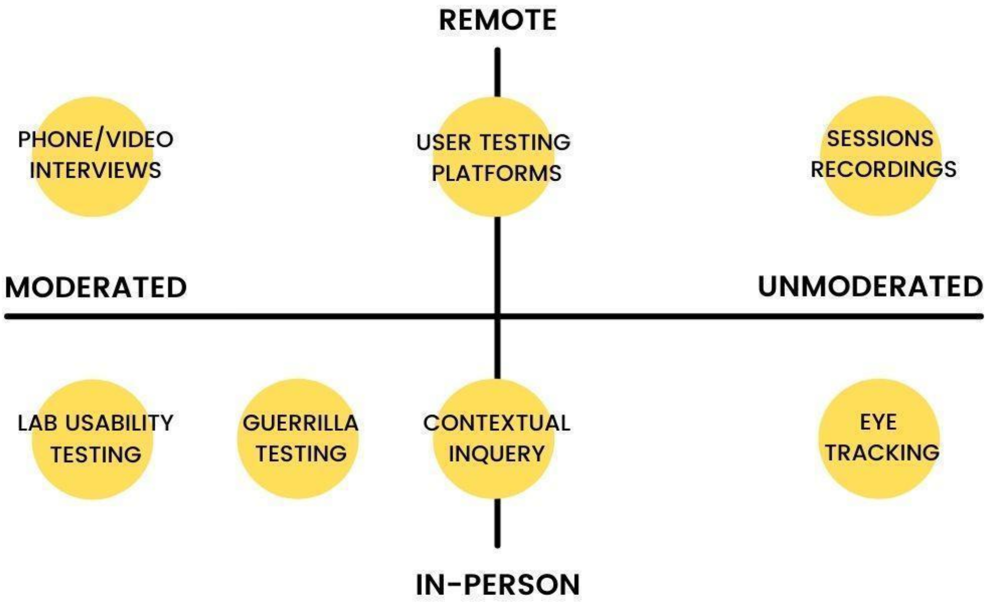
Remote unmoderated testing usually involves use of one of the many user testing platforms. These platforms support the creation of some tasks and collection of data. Many teams find these platforms beneficial because they are faster, easier, and cheaper than traditional in-person testing. Additionally, the tool tends to replace the traditional role of the facilitator, saving time for the research team.

The absence of a traditional facilitator can lead to issues that may go unnoticed. For example, when communicating their thoughts, a participant might just say “I don’t like that” instead of explaining why. When a facilitator is present, they can encourage the participant to explain why they dislike something by asking follow-up questions.

Another example of what may occur in unmoderated testing is participant misunderstanding of instructions. Without a facilitator, nobody is there in real time to correct misunderstandings or answer questions that the participant might have about what they are being asked to do. An advantage of moderated testing is that the facilitator can react to various situations that arise, with the aim of maintaining quality research.

Compared to in-person moderated usability testing, remote testing saves time in terms of travel for both researchers and participants and can be more comfortable for the participants. It is also easier to organize moderated usability testing remotely when recruiting candidates with very specific user profiles (i.e., people with a very specific job). It might be easier for the participants to find an hour where they can be at their computer in a quiet location versus traveling into a prepared physical testing facility or office.

This chart shows the relationship between the different methods for moderated, unmoderated, in person, and remote tests.



Guerilla testing
Guerrilla testing means going where users are and testing the prototype or system in its environment. The participants are chosen randomly, and they are asked to perform a quick usability test, often in exchange for a small gift.

Contextual inquiry
Contextual inquiry is more like an observation method that helps obtain information about the user experience. A facilitator goes to the user's place, observes, and take notes. This method helps get rich data about users and their workspaces, personal preferences, and habits.

3.3 Steps and Key Activities Required for a Usability Test

LO3.5	K3	Apply the principal steps in a usability testing approach.

To conduct a usability test, you should follow the following nine steps and key activities:

1. Define the scope of the test
   - Define what you want to test.
     - Ask yourself questions about the design/product.
     - What aspects of it do you want to test?
     - Define what to test (scope).
   - Decide how to conduct your test.
     - Is it moderated or unmoderated?
     - Is it done remotely or in-person?
     - Do you have a budget? (More info about this can be found here: https://www.nngroup.com/articles/remote-usability-testing-costs/) **[R14]**
2. Design the task(s)
   - Prioritize the most critical flows and divide them into tasks; no more than 5 per participant.
   - Define precise tasks with feasible goals.
   - Create real scenarios where participants can try to use the product as they normally do.
3. Prepare the test
   - Make sure you have all necessary materials for the test.
   - Prepare the devices or software you will use.
   - Create document for notes and/or observation.
   - Find a suitable location.
4. Conduct a pilot or test study
   - It will help you determine:
     - The order in which to present the tasks to the participants.
     - Any problems the test could have, e.g., something missing.
5. Recruit the participants
   - Set up the test in a quiet and distraction-free place.
   - Recruit participants.
   - Arrange the date, time, location, and incentives with each of the participants.
6. Run the session(s)
   - Observe and follow your test plan.

  - Explain to the participants what you expect from them and what you are going to do while they do the test.
  - Ask clarifying questions (i.e., ask why they decided to do something or click somewhere).
  - Ensure you give the participant the time to explain their thoughts and listen closely to what they say (it could inspire a follow-up question).
  - Be aware of any issues. Notice if participants fail to see part of the screen, go in a different direction, or misunderstand the instructions.
  - Recording the session is recommended. If you do so, please do not forget to ask participants for their consent.
7. Get insights
   - Review session video and notes and analyze data.
8. Create a usability report
   - Create a formal or informal report. Choose the one that fits your timing and what you want to communicate.
9. Present findings and recommendations
   - All stakeholders should be included.

3.4 Potential Roles in a Usability Test

LO3.6	K3	Apply role-based responsibilities to unique roles in a usability test.
LO3.7	K1	Remember a role based on a description.

Each role has specific responsibilities, and the success of the usability test depends on proper planning and everyone carrying out the responsibilities of their role successfully. The most common roles in usability testing are **facilitator** or **moderator**, **note-taker** or **scribe**, and **participant**; there should be no overlap in roles. The facilitator should not also take notes: it could introduce a bias, or the facilitator could miss important details due to the other responsibilities. Additionally, the note-taker should not also be facilitating as they could miss important details.

Role	Description	Responsibilities
Facilitator (or Moderator)	Guides participants through test process, asks questions, gives instructions, and is careful not to bias results or behave in a way that would threaten the validity of the data collected.	Assigns tasks for participants to complete
		Answers questions for participants
		Watches participant behavior and interaction with software during session
		Asks participants follow-up questions based on observation
Note-taker (or Scribe)	Takes notes and records observations made during the session. The scribe should ensure notes recorded during session do not threaten validity of results or include a bias.	Records participant reactions during session
		Records observation data shared by facilitator
Participant	User of application being tested or	Complete tasks assigned by the facilitator

	user of similar technology with the correct knowledge base selected to take part in the study.	Think out loud and narrate their actions and thought process as they complete tasks
		Provide meaningful reactions and feedback about product and session

3.5 Usability Test Plan

LO3.8	K2	Understand the content of a usability test plan and the most appropriate usability test scenario or a given study.
LO3.9	K3	Apply mandatory topics for a usability test plan.
LO3.10	K3	Apply a simple usability test task to a given project and put together the key activities for preparing a usability test.
LO3.11	K3	Apply usability test techniques to identify key activities in a usability test session.

Preparing a usability test starts with writing a plan. The usability test plan is a document that informs stakeholders and/or clients about the usability test to be conducted, and that guides them through the process. One of the purposes of this document is to gain the approval of the stakeholders. If modifications to the test plan are necessary, adjustments can be made prior to the test until it meets the expectations of the stakeholders.

A usability test plan contains information about:

- Objective(s) of the usability test
- Characteristics of the usability test: name, version, scope, name of facilitator(s) and/or note-taker(s), and the material to be tested
- Planned date(s) and time(s) for the participants
- Location: physical or remote
- Applied processes and/or techniques
- Number of test participants who will take part
- Profile of the participants
- Estimated length of each session
- Resources required to carry out the usability test
- The deliverable: how the usability findings will be communicated

The usability test plan must be concise and to the point, so it is easy to read and remember. Additional details about the usability test tasks are provided in the usability test script.

3.5.1 The Usability Test Script

The usability test script is a document that helps the moderator guide the session. It can be part of your test plan, but for most stakeholders, it is generally more desirable to put this in a separate document so that the content is more relevant to the reader.

The usability test script contains information about:

- Activities and instructions for preparing the usability test and welcoming participants (set the stage for the test)
- Briefing instructions (inform the participant what to expect)
- Pre-session questions (gain beneficial background information on behaviors that are useful in guiding the participant through the session)
- Usability test tasks (present the task to the participant)
- Post-session questions (gain valuable feedback from the participants about the software)

3.5.2. A Good Test Task

A good test task can be conceptualized as a “job-to-be-done” (abbreviated as JTBD). A JTBD describes the thought process of the user when they want to get something done, for example: making a purchase, contacting someone, subscribing to a newsletter, etc. **[R15]**.

A JTBD consists of a “*situation*,” “*motivation*,” and an “*expected outcome*”:

- The *situation* focuses on the **when**
- The *motivation* focuses on the **want**
- The *expected outcome* focuses on the **can**

That makes a JTBD look like: **When** ______, I **want** to ______ so I **can** ______

In a test task:

- The '*when*' describes the **scenario**
- The '*I want to*' describes the **task**
- The '*so I can*' describes the **goal**

Example JTBD's and test tasks:

- JTBD: **When** I want to know what I have left to spend this month, **I want to** log in to my bank account **so I can** see my balance.
- Test task: Imagine that you want to know what you have left to spend this month ***(scenario)***. You want to log in to your bank account to check your balance ***(task)***. Find out what you have left to spend ***(goal)***.
  - The goal is a success if the user finds the balance.

- JTBD: **When** I have a question about my subscription, **I want to** contact customer service **so I can** decide if I need to do something.
- Test task: Imagine that you have a question about your subscription ***(scenario)***. You want to talk to someone about this, because you don't know what the consequences are ***(task)***. Contact customer service ***(goal)***.
  - The **goal** is a success if the user can contact customer service and is therefore able to decide if there is an action required.

- JTBD: **When** I am interested in a course, **I want to** be kept informed about special offers **so I can** decide in the future if I want to take the course.
- Test task: Imagine that you are interested in a course ***(scenario)***. You want to stay informed about special offers so you can benefit from a regular discount on the course ***(task)***. Sign up for the newsletter to receive information about future discounts ***(goal)***.

- The **goal** is a success if the user signs up for the newsletter.

More information about this topic can be found here:
https://marketing.intercomcdn.com/assets/jtbd/v1/Intercom_on_Jobs-to-be-Done.pdf **[R15]**

3.5.3 Running the Usability Test Session

When conducting the session, the facilitator and note-taker should follow a set procedure with each participant.

This four-step procedure can be customized and still help to standardize the experience for each participant:

1. **Introductions and warm-up**
    - Make sure your participant is comfortable with the setup (chair, desk height, mouse placement, etc.); if you are doing it remotely, check that the audio on both sides is working properly.
    - If you are recording the session, ask the participant for their agreement; if you are running the session in-person, ask the participant to sign a printed form.
    - Be friendly and empathetic: try to make them feel relaxed in the testing environment.
2. **Collect pre-testing data**
    - During your conversation, collect general information about your participant.
3. **Go into the first task**
    - Lead the participant into the first testing task. Usability tests usually have 3 or 4 scenarios to go through.
4. **Follow up questions and wrap-up**
    - Leave some time at the end of the session to ask any follow-up questions and collect the participant’s final feedback. Be sure to thank them for their help.

3.5.4 Appropriate Questions for a Usability Test Study

LO3.12	K2	Understand questions that are appropriate to use during a usability test study.

There are some common mistakes when facilitating usability test studies, including approaching them more like conversations instead of observations, interrupting, asking too many probing questions, or offering help when the participant is stuck with a task. On the other hand, there could be some facilitators afraid to talk during the session, who might not know how to react when the participant asks a question.

The most important thing to consider while facilitating a usability testing session is how to ask your questions in a way that is not leading or biased. A leading or biased question manipulates (consciously or unconsciously) participants into responding in a certain way, modifying your test results. Also, avoid closed questions, which result in a "yes/no" answer.

The following are examples of questions you should not ask during a usability session:

- Do you think this is a nice illustration?
- How can we make this experience simpler?

- Was the language clear and straightforward?

Using words like "simple" and "nice" can bias participants towards specific ideas, so avoid adjectives. Phrase questions in a way that invites participants to answer openly.

The following questions are phrased more effectively:

- What do you think about this illustration?
- What is your general opinion about the experience?
- What do you think about the language used in the product?

The following are some **question-asking techniques** you can use to not overwhelm or leading the participant.

Echo technique

Used to help participants clarify what they mean when they say something incoherent or unclear. What you do is follow up with a question that uses the exact word or words the participant said. For example, imagine the participant says, “this is how we usually do things....” The facilitator can then say, "how do you usually do things?” Adding an interrogatory tone to the exact words the participant says helps eliminate judgment and invites them to say more.

Boomerang technique

Named after the tool that indigenous Australians used for hunting, and the idea is that it returns to the person who initially threw it. For usability testing purposes, we use it to return a participant’s question. Imagine a participant asks, “How should I use this?” The facilitator might then ask the participant, “how do YOU think it should be used?” This technique will help to go deeper into the participant’s assumptions about the product.

Colombo technique

A TV series character, Detective Lieutenant Colombo, knew how to ask the right question or make the right comment at the right time. The idea is to try to adopt elements of Colombo’s persona and ask partial questions, especially when you cannot think of a non-leading question. For example, you might point to a specific spot on the screen and say, “That button ...,” only saying the part you think is safe to say and wait until the user interrupts you and finalizes your phrase.

3.6 Observations and Usability Findings

LO3.13	K3	Apply a given list of observations to select useful findings, define conclusions, and define classifications.
LO3.14	K2	Understand the relationship between findings and conclusions.
LO3.15	K2	Understand the classifications and ratings for usability findings.

3.6.1. Observations, Insights, Findings, and Conclusions

Insights arise from **observations**. Insights lead to **findings**. Findings lead to **conclusions**.

3.6.2. Taking Notes on Observations

There is no generally accepted shorthand for notetaking during a usability test. Note-taking can be done freehand, but there are also structured ways to do this, such as:

- The rainbow spreadsheet **[R16]**: this technique takes its name from the different colors used to represent the study participants. It is a spreadsheet in which the entire team documents, simultaneously, the data collected during a study. It serves as the centerpiece for lessons learned from a study and could later turn into the final report.
- Note(-taking) cards: these are five stacks of cards. Each stack represents a type of observation, for example:
    - The user can/cannot ..., because …
    - The user wishes ..., because …
    - It's unclear for the user that ..., because …
    - The user expects ..., because …
    - The user likes/dislikes …, because …

  Each observation is noted on a card from one of these stacks. Afterwards, these cards can easily be clustered together.
- An annotation board: this can be as simple as a sheet of paper that is divided into blocks. Each block represents a topic that is relevant for the user research. Some examples of topics are "findability," "usability," "credibility," etc. Each observation is noted under the corresponding topic. Upon completion, all observations are clustered by theme.
- Screenshot note-taking: in this case, copies of screenshots are printed and notes are noted directly on the printed screenshots.

The most important task for the note-taker(s) is to reflect events that influence (positively or negatively) effectiveness, efficiency, and satisfaction and that are related to the goals of the study.

Note-taking can be done on paper, digitally, by recording the user's voice, or by taking a video in which you can have more visual information about gestures and non-verbal communication. All of these methods have advantages and disadvantages that should be considered when making an appropriate choice.

Helpful hints to consider:

- If the facilitator is also the note-taker (which is not recommended), a screen between the facilitator and the participant can come across as distant. With paper, this is not the case.
- Taking notes from behind a screen comes across as less transparent. For example, the participant may think that the facilitator might be busy with other activities (and therefore not listening with full attention). Paper feels more transparent because the participant can see what the facilitator is doing and is then sure that the facilitator is not busy with other distractions.
- Writing down an observation takes more time than typing.
- If taking notes on a keyboard, the noise can be annoying and may constantly remind the participants that their actions are observed and recorded.
- Collaborating with other note-takers is easier on a computer because an observation document can be shared among them.

3.6.3. Translating Observations into Findings and Conclusions

In order to write conclusions, it is required to first define the findings.

Defining findings:

An observation is useful when it can be translated into a usability finding. A usability finding can describe:

- … a usability problem.
- … a positive finding: something that users liked, experienced as pleasant, or clearly helped the users to achieve their goal.
- … a suggestion: an opportunity created by a good idea from the participant.
- … a new user requirement resulting from the analysis of usability findings.

It is important that findings are based on observations so that personal opinions are avoided.

Findings should be classified and have a severity rating that indicates the impact of what was observed on the user experience and how critical the issues are.

Findings classification:

Classification	Description
Usability problem	A bottleneck with a low, medium, or high severity rating (as mentioned in the table below)
Positive finding	An approach that works well and can be recommended
Suggestion	A possible solution to a finding, a recommendation, or a comment based on feedback from a participant
Functional problem	Bug/defect/failure/fault

Typical severity ratings:

Severity rating	Description
Low priority	A small problem which somewhat hinders the user in achieving their goals within the software
Medium priority	A major problem which hinders the user in achieving their goals within the software
High priority	A critical problem which seriously hinders the user during use or blocks the user from achieving their goals within the software

There are variables that affect the severity rating of findings. These variables are:

Variables	Description
Impact	How strongly it affects the user and the user's environment when the usability problem occurs

Persistence	How quickly users learn to avoid the usability problem
Frequency	How often the usability problem occurs

	Low priority	Medium priority	High priority
Impact	It has little impact on the user and the user's environment.	It has medium impact on the user and the user's environment.	It has large impact on the user and the user's environment.
Persistence	Users learn to avoid the usability problem very quickly.	Users learn to avoid the usability problem with difficulties.	Users didn't learn to avoid the usability problems.
Frequency	The usability problem rarely occurs.	The usability problem occurs very often.	The usability problem occurs all the time.

Writing conclusions:
When findings have been defined, a list of areas for improvement can be drawn up and conclusions can be written to make a summary statement about the effectiveness and efficiency of the software, as well as user satisfaction.

More information about how to write conclusions can be found here: https://www.interaction-design.org/literature/article/how-to-write-the-perfect-conclusion-to-your-ux-case-study **[R17]**

3.7 Reporting and Communicating Usability Findings and Conclusions

LO3.16	HO	Review a usability test report for a given project.
LO3.17	HO	Apply observations to select useful findings and use these findings to define conclusions and classifications.

3.7.1. Report Usability Findings

After a usability test has been conducted, all findings have been defined, and conclusions have been written, a **usability test report** follows.

A **usability test report** is a document that shares insights with stakeholders such as the product team and/or executive management. It should contain a section that describes key characteristics of the usability test (also mentioned in the usability test plan in chapter 3.7 "The Usability Test Plan"), the most important findings from the usability test, the conclusion(s), and associated recommendations for improvement of the software product.

Each finding should include the following information:

- a clear description of the finding;
- the classification and severity rating of the finding (see chapter 3.10.3 "Translate observations into findings and conclusions");

- an indication of the number of participants who stumbled upon the problem, for example "some," "most," or "all" participants;
- potential recommendation(s) for improvement (if possible); and
- optionally (if possible) relevant quotes from participants related to the finding and/or supporting screenshots that demonstrates the finding.

3.7.2. Communication of Usability Findings

Once the usability test report has been worked out, stakeholders should be informed of the results.

The following techniques can be used for this process:

1. Deliver a presentation of (main) usability findings and recommendations with subsequent discussion.
2. Distribute the usability test report with planned subsequent discussion.
3. Create and distribute a video summary.
4. Get key stakeholders involved during the usability test and have informal discussions about usability findings between usability test sessions.
5. Organize workshops with stakeholders.
6. Put usability findings on a product backlog (with traceability of the submitter, should a further explanation be required).

Make sure communication is always two-way. Never simply present or distribute usability findings. Give stakeholders the chance to comment on the usability findings before they are distributed to others. Discussing usability findings from their perspective can prevent misunderstandings.

Avoid opinions during discussions as they can easily lead to an "opinion war." Some stakeholders see "their" software product as an extension of themselves and might take it personally when someone finds flaws in it.

Keep in mind that if certain stakeholders have no (or little) knowledge of usability, they may need to be convinced to take the findings from a usability test seriously and act on them.

References

Reference	Source
1.2 [R1]	https://www.iso.org/standard/77520.html
1.3 [R2]	W3C Web Accessibility Initiative. (2020, October 17). *Web Content Accessibility Guidelines (WCAG) Overview*. Web Accessibility Initiative (WAI). https://www.w3.org/WAI/standards-guidelines/wcag/
1.4 [R3]	https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them
1.5 [R4]	https://www.w3.org/WAI/people-use-web/abilities-barriers/
1.8.1 [R5]	https://www.nngroup.com/articles/ux-expert-reviews/
1.8.2 [R6]	https://abtestguide.com/abtestsize/
1.8.4 [R7]	https://www.nngroup.com/articles/ux-prototype-hi-lo-fidelity/
1.9 [R8]	https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/
1.9 [R9]	https://www.nngroup.com/articles/how-many-test-users/
1.11 [R10]	http://universaldesign.ie/What-is-Universal-Design/The-7-Principles/
2.1 [R11]	https://www.nngroup.com/articles/ten-usability-heuristics/
2.1 [R12]	https://www.userfocus.co.uk/resources/guidelines.html
2.2 [R13]	https://karlgroves.com/2013/09/05/the-6-simplest-web-accessibility-tests-anyone-can-do
3.4 [R14]	https://www.nngroup.com/articles/remote-usability-testing-costs/
3.5.2 [R15]	https://marketing.intercomcdn.com/assets/jtbd/v1/Intercom_on_Jobs-to-be-Done.pdf
3.7.2 [R16]	https://marketing.intercomcdn.com/assets/jtbd/v1/Intercom_on_Jobs-to-be-Done.pdf
3.6.2 [R16]	https://docs.google.com/spreadsheets/d/1bPg6op9Sk46lFVGaET-fruE0qz-ctNQsxbZKF-5lpn4/edit#gid=0
3.6.3 [R17]	https://www.interaction-design.org/literature/article/how-to-write-the-perfect-conclusion-to-your-ux-case-study