UXQB
International Usability and UX Qualification Board

CPUX-UR Curriculum

Certified Professional for Usability and User Experience – User Requirements Engineering

Version 3.2.2 EN: 14 March 2023

Published by: UXQB e. V.

Contact: info@uxqb.org

www.uxqb.org

Edited by UXQB Working Group “CPUX-UR”

Table of contents

Preliminary notes

This document describes the content covered by the certification examination for the certificate “Certified Professional for Usability and User Experience – Advanced Level User Requirements Engineering” (CPUX-UR).

The curriculum and the defined terms contain all the topics and technical terms that might come up in the theoretical exam as well as in the tasks of the practical exam.

The certification examination will only include terms and topics that are described in the curriculum and glossary. The goal of the examination is to evaluate:

- What applicants know and what they can recall;
- Whether applicants understand the terms and topics;
- Whether applicants can apply the knowledge in practice.

This document covers the human-centred design activities “Understand and specify the context of use” and “Specify the user requirements” of the human-centred design process as defined in ISO 9241-210 “Human-centred design process for interactive systems” (see figure 1).

This document is primarily based on the following resources:

1. CPUX-F Curriculum (available at www.uxqb.org);
2. ISO 9241-210 “Human-centred design process for interactive systems”;
3. ISO/IEC 25063 “Common Industry Format (CIF) for usability: Context of use description”;
4. ISO/IEC 25064 “Common Industry Format (CIF) for usability: User needs report”;
5. ISO 25065 “Common Industry Format (CIF) for usability: User requirements specification”.

Further resources are listed in Appendix 3: References & Index

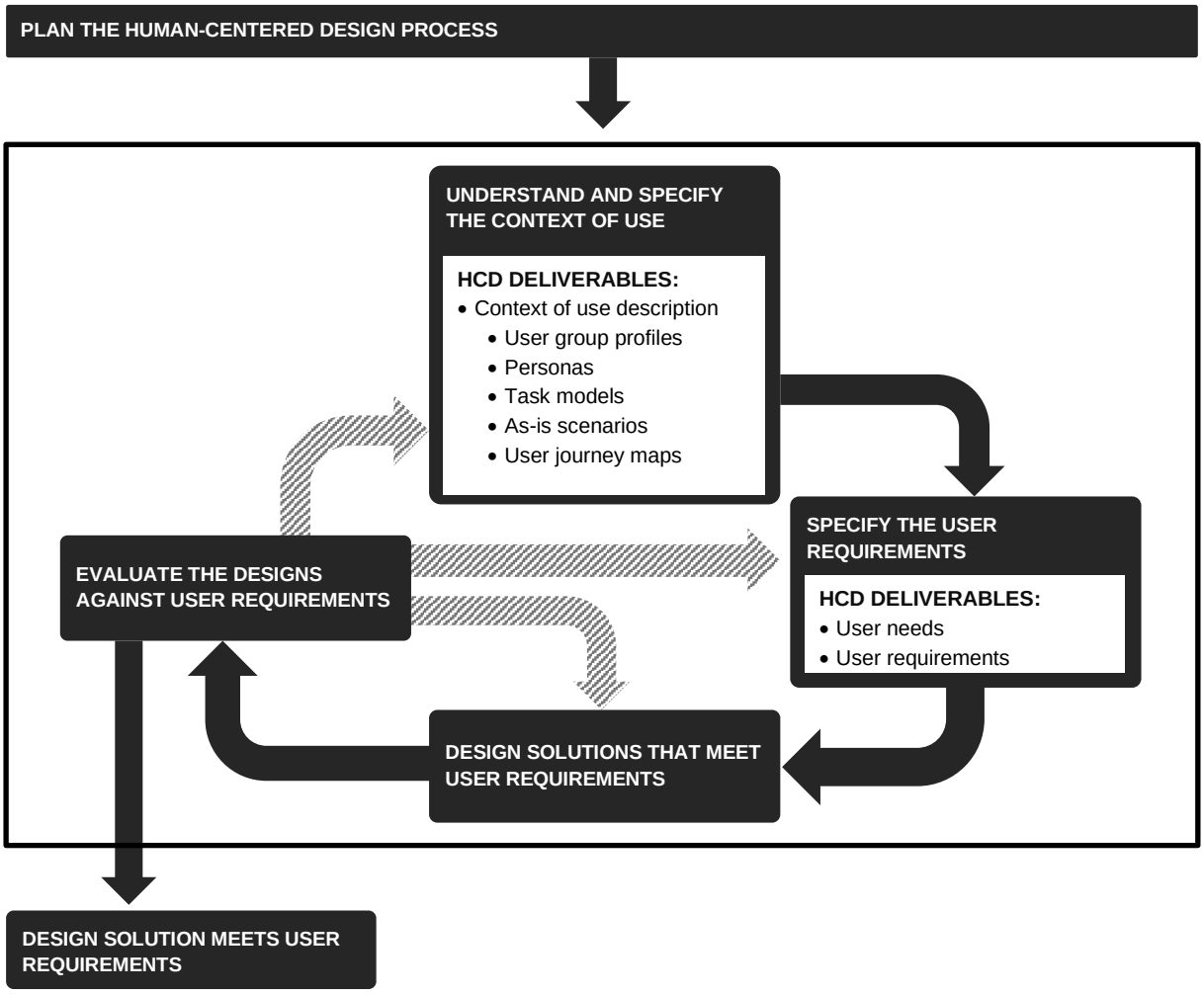

Figure 1 User Requirements Engineering (enlarged boxes) within the Human-centred Design Process and related HCD Deliverables

Structure of this document

The curriculum consists of six chapters. (Chapter 1 to Chapter 6)

- Every chapter of the curriculum contains of one or more learning units (for example, 1.1 to 1.4)
- Each learning unit contains:
  - A summary of the content of the learning unit;
  - The learning objectives of the learning unit (for example, 1.1.a … 1.1.i);
  - The learning content along with the defined terms structured in subclauses (for example, 1.1.1 … 1.1.4).

The learning content mainly states what is to be taught in a preparatory course for the certification examination CPUX-UR.

What does the CPUX-UR certificate confirm?

The certificate *UXQB® Certified Professional for Usability and User Experience – Advanced Level User Requirements Engineering (CPUX-UR, theoretical part)* confirms that the holder:

- has demonstrated their understanding of the specific terms, concepts and relevant methods of context of use analysis, as well as the elicitation, structuring and prioritization of user requirements for interactive systems;
- has demonstrated the competency in user requirements engineering by completing a theoretical certification exam.

The certificate *UXQB® Certified Professional for Usability and User Experience – Advanced Level User Requirements Engineering (CPUX-UR)* confirms that the holder:

- has demonstrated their understanding of the specific terms and concepts of context of use analysis, as well as the elicitation, structuring and prioritization of user requirements for interactive systems;
- can apply relevant methods of context of use analysis, as well as the elicitation, structuring and prioritization of user requirements for interactive systems;
- has demonstrated the competency in user requirements engineering by completing both a theoretical and practical certification exam.

The content and learning units of chapters 1 to 6 are taught in preparatory courses for CPUX-UR. The main focus is on understanding the terms and concepts from the curriculum for CPUX-F and applying them in practice as well as knowing, understanding, and being able to apply additional terms and concepts. Appendix 1: Model seminar contains a timetable for a typical seminar, which can be used by training providers as a guide to developing their own seminars.

Learning objectives and proficiency levels

Each learning objective is assigned to a proficiency level (K1, K2, K3). Proficiency level K3 includes level K2; level K2 includes level K1.

Table 1 states ways of demonstrating expertise for each proficiency level. The verbs used in this context are the same as those used to describe the learning objectives.

Table 1 proficiency levels for the CPUX certification model

Proficiency level	Ways of demonstrating expertise for each level
K1 (Know)	List, identify, recognise, name
K2 (Understand)	Analyse, apply, perform, justify, describe, assess, depict, devise, develop, complete, explain, exemplify, evaluate, formulate, identify, interpret, draw conclusions, transfer, differentiate, compare, understand, suggest, summarise.
K3 (apply in practice)	K3.1 Being able to plan. K3.2 Being able to perform. K3.3 Being able to analyse. K3.4 Being able to document. K3.5 Being able to communicate and pass on.

“Know” (K1) means being familiar with the basic terms and concepts in the area of usability and user experience.

“Understand” (K2) means

- Being able to recognise relationships within and between concepts. For instance, the relationship between user needs and user requirements.
- Being able to identify a suitable approach for a specific activity concerning a given problem, such as, the approach for recognising and formulating user needs from context of use information.

“Apply in practice” (K3) means being able to successfully apply what is known and understood in specific use cases.

The foundation level (CPUX-F) focuses on K1. The advanced levels (CPUX-UR, CPUX-DS, CPUX-UT) focus on K2 and K3.

The proficiency level K1 constitutes the basis for the theoretical exam at the foundation level (CPUX-F). The proficiency level K2 constitutes the basis for the theoretical exam in the advanced level (CPUX-UR, CPUX-DS, CPUX-UT).

The practical exam in the advanced level CPUX-UR focuses on the proficiency levels K3.1, K3.3, and K3.4.

To clarify: The proficiency of “being able to perform” (K3.2) is essential for being successful as a usability engineer (for example, conducting interviews and observations). Nonetheless, for the practical exam in CPUX-UR, the emphasis is on “being able to analyse”, as this proficiency is especially important for deriving, structuring, and prioritising user requirements. The corresponding learning content is not covered in other certification procedures.

Acknowledgments

This document was created by the following persons:

- Thomas Geis (Editor)
- Nicole Gerloff
- John Goodall
- Mascha Lampert
- Knut Polkehn (Co-Editor)
- Matthias Reisemann

Overview of learning units, main terms/concepts, methods, and deliverables

This overview serves as the central theme for the CPUX-UR curriculum. It shows the content of each learning unit in a compact format (most important terms, methods, and deliverables).

Table 2 Overview of learning units, main terms/concepts, methods and deliverables

Chapter		Learning unit		Important terms and concepts	Methods in this learning unit	Deliverables in this learning unit
1	Introduction to context of use analysis	1.1	Differentiate between requirements and solutions	• Requirement • Solution • Request	None; basic knowledge	None; basic knowledge
		1.2	Differentiate between stakeholder requirements and system requirements	• Stakeholder requirement • System Requirement • Legal or regulatory requirements • Market requirement • Organisational requirement • Domain-specific requirement	None; basic knowledge	None; basic knowledge
		1.3	The context of use as the basis for user requirements	• User • Task • Goal • Resources • Task model • Task model for design • Task object • Social environment • Physical environment • Technical environment	None; basic knowledge	None; basic knowledge

Chapter		Learning unit		Important terms and concepts	Methods in this learning unit	Deliverables in this learning unit
		1.4	User requirements as a separate category within the stakeholder requirements	User requirement	None; basic knowledge	None; basic knowledge
2	Plan context of use analyses	2.1	Identify the rationale and goals for the context of use analysis	• Context of use analysis • Different rationales • Different objectives	None; basic knowledge	None; basic knowledge
		2.2	Determine the approach for context of use analysis	• Context of use analysis • Classic context of use analysis • Model-based context of use analysis	None; basic knowledge	None; basic knowledge
3	Gather and document context of use information	3.1	Select and recruit users for gathering context of use information	• User vs. user group	Grouping of users based on relevant commonalities in the context of use	• User group profile • Recruitment screener
		3.2	Prepare and gather context of use information	• Contextual interview • Observation • Focus group	• Contextual interviews • Observations • Focus groups	• User group profile • Recruitment screener • Interview checklist
		3.3	Create context of use descriptions based on context of use information	• Context of use description	None; different documentation formats are presented	Context of use description in different formats: • As-is scenario • Persona • Further formats (see Chapter 3.3.2)

Chapter		Learning unit		Important terms and concepts	Methods in this learning unit	Deliverables in this learning unit

4	Identify user needs from context of use information	4.1	Systematically identify and formulate user needs	• User need
				  • Resource need
				  • Informational need
				  • Competency need
					Identify necessary prerequisites and objectives in context of use information	Evaluated context of use description — Part 1 of 2: User needs

5	Derive and structure user requirements from user needs	5.1	Systematically transform user needs into user requirements	• User requirement	Identify necessary interactions with the interactive system (recognise, select, input)	Evaluated context of use description — Part 2 of 2: User needs and derived user requirements
		5.2	Appropriately structure user requirements	• Task • Subtask • Task model for design	Structure user requirements based on tasks and subtasks	• Task model for each task • Structured list of user requirements
6	Consolidate user requirements	6.1	Consolidate and prioritise user requirements with users	• Importance (from the user's view) • Kano scheme	Consolidation workshop with users	Structured list of user requirements — with prioritisation from the user's view
		6.2	Determine implementation priority for user requirements together with project members	• Implementation priority • Product roadmap		Structured list of user requirements — in order of priority

1 Introduction to context of use analysis

1.1 Differentiate between requirements and solutions

In **human-centred design**, **interactive systems** are designed with a focus on the users of interactive systems and on other humans in the users' social environment who may be affected by the interactive system. The **quality** of the interactive system comprises **human-centred quality** and **technology-centred quality**. Human-centred quality results from meeting requirements for usability, accessibility, user experience and **avoidance of harm from use**. **Risks** for users and other humans can originate from the design, including use errors that can lead to negative consequences on health, safety, finances, or the environment.

The **context of use** is the basis for identifying **user needs**, from which we derive **requirements** which help achieve human-centred quality. While requirements are always based on user needs and specify the capabilities of the interactive system, **solutions** are attributes of interactive systems that are implemented to meet requirements. Requirements help in identifying suitable solutions and discarding unsuitable ones, to reach a consensus on proposed solutions that are based on facts rather than opinions. The **solution space** must fit the **problem space**. **Requests** from users and other stakeholders, for example **user wants** need to be analysed to identify underlying user needs, to enable valid user requirements to be specified. This ensures the **immunisation trap** is avoided, preventing baseless requirements and as a consequence, a suboptimal interactive system. The solution space for the interactive system is determined by context-specific user requirements, established design recommendations and specific **user interface guidelines**.

Learning Objectives	
1.1.a	Know about the two basic types of quality for an interactive system (human-centred quality and technology-centred quality).
1.1.b	Know and be able to differentiate between 1) Context of use, 2) User needs 3) Requirements and, 4) Solutions.
1.1.c	Know the difference between requirement and solution.
1.1.d	Know the difference between request and requirement.
1.1.e	Know the difference between problem space and solution space.
1.1.f	Understand that requirements must be based on user needs in the context of use.
1.1.g	Understand the difference between requirements and user interface guidelines.
1.1.h	Be able to determine whether a textual statement contains a requirement or a solution.
1.1.i	Know about the immunisation trap when formulating requirements.

1.1.1 Human-centred quality and technology-centred quality

Different perspectives must be considered when designing an interactive system to ensure quality. Most importantly, technology-centred quality must be considered alongside human-centred quality. Human-centred design of interactive systems focuses on achieving human-centred quality.

Human-centred design

CPUX-F Definition

An approach to design that aims to make interactive systems more usable by focusing on the use of the interactive system and applying human factors, ergonomics and usability knowledge and methods.

The concept "human-centred design" is used instead of "user-centred design" to emphasise the need to consider additional stakeholders who may not be users. Nevertheless, feedback from users through usability evaluation is the most important source of information in human-centred design.

The objective of human-centred design is to develop interactive systems that have appropriate human-centred quality for their intended users, whilst considering the social environment of others who are or who can be affected by the interactive system.

Interactive system

CPUX-F Definition

A combination of hardware, software and services that users interact with in order to achieve specific goals.

This includes, where appropriate, packaging, user documentation, online help, support, and training.

The term “quality” plays an important role in developing interactive systems.

Quality

The degree to which the interactive system meets requirements.

The term quality covers the fulfilment of requirements for usability, accessibility, user experience, and avoidance of harm from use (human-centred quality) as well as the fulfilment of system requirements (technology-centred quality).

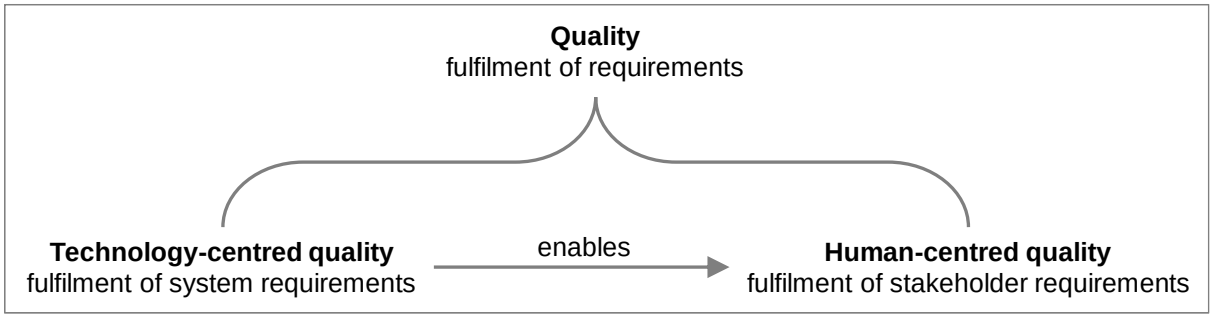

Figure 2 Quality as the fulfilment of System Requirements and Stakeholder Requirements

Technology-centred quality is the outcome of implementing system requirements.

Technology-centred quality

The degree to which the interactive system meets system requirements.

Technology-centred quality consists of the following quality dimensions:

- Functional appropriateness;
- Technical performance;
- Interoperability;
- Reliability;
- Security;
- Maintainability;
- Portability.

Along with human-centred quality, technology-centred quality is an important aspect of overall quality.

While technology-centred quality is often taken for granted by users, human-centred quality is actively perceived by users when interacting with the interactive system.

> For example, the user of a washing machine unconsciously assumes that their laundry will be cleaned (enabled by technology-centred quality).

Human-centred quality

CPUX-F Definition
The extent to which requirements for usability, user experience, accessibility and avoidance of harm from use are met.

Avoidance of harm from use is defined as follows:

Avoidance of harm from use

CPUX-F Definition
The extent to which negative consequences to health, safety, finances or the environment that result from use of the interactive system are minimised.

Avoidance of harm from use is considered when specifying human-centred quality objectives for an interactive system. See also the examples given for avoidance of harm from use under “human-centred quality objectives”.

If avoidance of harm from use is not considered, users may be exposed to unacceptable risks.

Risk

Combination of the probability of occurrence of harm and the severity of that harm.

Harm is physical injury or damage to the health of people, property, or the environment. It is important to distinguish between the context of use, user needs, requirements, and solutions in a systematic way.

To achieve human-centred quality in the design of an interactive system, the context of use must be understood and specified.

Context of use

CPUX-F Definition
A combination of users, goals and tasks, resources and environments.

It specifically describes the components of the context of use that are relevant for the use of the interactive system.

Context of use information is factual information collected in the context of use. The results from observations and contextual interviews are described in context of use descriptions.

The part of the context of use that is used as a basis for the design of an interactive system is called the context of use for design.

Context of use descriptions are the source for identifying user needs. Thus, user needs can be traced back to the context of use.

User need

CPUX-F Definition
A prerequisite identified as necessary for a user, or a user group, to achieve a goal, implied or stated within a specific context of use.

User needs serve as a helpful intermediate step in the transformation of the context of use information into comprehensive user requirements.

Each component of the context of use (users, goals, tasks, resources, environments) can be a source for user needs.

User needs

- are independent of any proposed solution for each need. In other words: A user need must not refer to, for example, “the system” or “the website”.
- are identified through various approaches, including interviews with users, observations, user surveys, usability evaluations, expert analysis, etc. They often represent gaps or discrepancies between what is and what should be.
- can be differentiated as resource needs, informational needs, and competency needs.
- are transformed into user requirements and other requirements.

Requirement

CPUX-F Definition
A condition or capability that must be met or possessed by an interactive system to satisfy an agreement, standard, specification or other formally imposed documents.

Requirements help with the design of suitable solutions by dictating what should be achievable with the interactive system.

Solution

One or more related product characteristics that are specified or implemented and are supposed to fulfil one or more requirements.

1.1.2 Differentiate between requirements and requests

In contextual interviews, users may make suggestions for functions or for the design of the user interface.

Request

A user want or a stipulation stated by one or more stakeholders for the interactive system.

Requests are often stated as a proposal for a solution without stating the underlying user need. They are often descriptions of “features” that a user wants for a product.

User want

A desire regarding the attributes of an interactive system, stated by a user.

Features requested by individuals are not necessarily solutions suitable for all users.

Example 1:

- Request:
  After sending an email, the email program must automatically display the folder “Sent objects” instead of the email just sent.
- Potential user requirement:
  After sending an email with the system, the user must be able to recognise whether or not the email has been sent.

Example 2:

- Request:
  The dishwasher must continue to beep from the moment the washing programme has finished until the dishwasher has been emptied.
- Potential user requirement:
  With the system, the user must be able to recognise that everything in the dishwasher is clean before adding a dirty dish.

Requests

- for changing an existing system or prototype by any stakeholder, including management, are often called “change requests”.
- are not requirements, however, they can be analysed for implied user needs and derivable user requirements, just as in any other context of use information.
- should not be mistaken as requirements until they have been assessed for underlying context of use and unsatisfied user need(s).
- point to context-specific user requirements. User interface guidelines in turn are rules that apply across many contexts of use.

User interface guideline
CPUX-F Definition
A low-level, specific rule or recommendation for the design and implementation of a user interface that leaves little room for interpretation, allowing everyone to implement it consistently.

1.1.3 The benefits of taking requirements as the basis for the development of possible solutions

The context of use within which users complete their tasks with given resources in their given environment is the problem space.

Problem space
Collective term for the current context of use and the user needs within the context of use.

User needs are part of the problem space. User needs that are unmet in the current context of use lead to requirements on the interactive system for future releases.

User needs that resulted in requirements that were implemented in previous releases and which are now met, must be taken into account in future releases to ensure that existing user needs continue to be met.

Solutions to requirements that have already been implemented and which change over time must still meet the implemented requirements.

User requirements are part of the solution space from the user’s perspective, as they are formulated in relation to the use of the interactive system.

Solution space

Collective term for the user requirements and future solutions that meet the user needs in the context of use for an interactive system.

The solution space can be narrowed down based on requirements in order to identify appropriate solutions.

Sometimes, requirements are factually derived from the solution rather than from the user needs within the context of use. This methodological mistake is called the immunisation trap.

Immunisation trap

An unconsciously chosen approach for specifying requirements where, instead of being derived from user needs in the context of use and being independent of any solution, requirements represent known or imagined solutions.

Requirements that were specified in this way are called immunised requirements.

Example 1 of immunised requirements:

- Immunised: With the system, the user must be able to call up the pricing system at the ticket machine.
- Not immunised: With the system, the user must be able to recognise the cost of the trip based on the selected destination.

Example 2 of immunised requirements:

- Immunised: With the system, the user must be able to select one of the heating levels 1, 2, 3, 4, or 5.
- Not immunised: With the system, the user must be able to select their preferred room temperature.

Deriving requirements based specifically on context of use information avoids the immunisation trap. This allows the comparison of alternative solutions that meet the requirements.

1.2 Differentiate between stakeholder requirements and system requirements

Requirements can generally be categorised into **stakeholder requirements** and **system requirements**. **Stakeholders** of an interactive system comprise **users** of the interactive system and all other humans who have an interest in it and/or are affected by it. Users can be categorised into **primary users**, **secondary users,** and **indirect users**. The **sponsor** of the interactive system is also a stakeholder.

While stakeholder requirements specify capabilities of the interactive system from the perspective of stakeholders, system requirements specify capabilities of the technical system to meet stakeholder requirements. Stakeholder requirements are the basis for system requirements. Stakeholder requirements can be categorised into **legal or regulatory requirements**, **market requirements**, **organisational requirements**, **domain-specific requirements** and user requirements. While the process of **verification** ensures that all requirements – stakeholder requirements and system requirements – have been implemented, the process of **validation** ensures that stakeholder requirements have been implemented effectively, from the perspective of stakeholders. To reach the necessary human-centred quality of an interactive system, all relevant requirements have to be fully derived, implemented, and verified.

While this curriculum focuses specifically on user requirements, the **CPRE** curriculum (**IREB** e.V.) focuses on requirements in general and has no specific focus on user requirements.

Learning Objectives	
1.2.a	Know about and be able to differentiate between types of stakeholder requirements.
1.2.b	Know the difference between stakeholder requirements and system requirements.
1.2.c	Understand that stakeholder requirements are the basis for system requirements.
1.2.d	Understand that user requirements make up their own category within stakeholder requirements.
1.2.e	Know the meaning of verification and validation and the difference between the two.
1.2.f	Know the difference between usability engineering and requirements engineering.
1.2.g	Know about IREB and CPRE.

1.2.1 Classify stakeholder requirements

People with different roles and different interests are involved in the design of the interactive system.

Stakeholder
CPUX-F Definition
An individual or organisation with an active interest in an interactive system.

All users are stakeholders.

User
CPUX-F Definition
A person who interacts with an interactive system, or who uses the output of the system.

Stakeholders may or may not be users. Stakeholders are not considered to be users if they are affected by an interactive system but do not interact with the interactive system or use its output.

Examples of stakeholders who are not users:

- Managers of users;
- People affected by the noise produced by the interactive system;
- Marketers affected by the impact of the output on the brand name.

There are three types of users:

- Primary users
- Secondary users
- Indirect users

Primary and secondary users interact with the interactive system. Primary users perform tasks that the interactive system is primarily intended to support.

Primary user
CPUX-F Definition
A user who uses the interactive system for its intended purpose.

In contrast to primary users, secondary users perform support tasks.

Secondary user
CPUX-F Definition
A user who carries out support tasks with the interactive system, for example to maintain it or to train primary users.

Examples of secondary users:

- System administrator
- Trainer
- Service technician

While primary and secondary users interact directly with the system, indirect users use the output of the interactive system.

Indirect user
CPUX-F Definition
A user who uses the output of the interactive system, but who does not interact directly with the interactive system.

Example: A bank customer who receives a paper or electronic statement, or visits a branch is an indirect user of the output produced by the bank computer system that is used by the bank employees.

Sponsors are also stakeholders.

Sponsor

Stakeholder who provides the budget for the development project.

The sponsor decides on the budget for human-centred design activities, such as gathering and documenting context of use information or specifying user requirements.

Each stakeholder group imposes requirements on the interactive system.

Stakeholder requirement

What the interactive system should be capable of from the point of view of the stakeholders.

Stakeholder requirements can be classified as follows:

- Legal or regulatory requirements: the stakeholder is the “Legislator”;
- Market requirements: the stakeholder is the “Purchaser or purchase decision maker”;
- Organisational requirements: the stakeholder is the “Management of the organisation”;
- Domain-specific requirements: the stakeholders are indirect users, for instance, a user of the work products that were created by the primary or secondary user using the interactive system;
- User requirements
  - Qualitative: Stakeholders are the primary and secondary users;
  - Quantitative: All stakeholders.

Stakeholder requirements from one category can lead to further stakeholder requirements in other categories.

Legal or regulatory requirements prescribe rules of conduct for organisations and individuals.

Legal or regulatory requirement

A stakeholder requirement enforced by law or by a regulatory document (for instance, a harmonised ISO standard, a specification from a regulatory authority) that needs to be respected when designing an interactive system.

Examples:

- An employer must regularly check if working conditions pose a threat to health (Working Conditions Act).
- The principles of ergonomics are to be applied especially for the processing of information by humans (German ordinance for work with visual display units).
- The manufacturer must identify known or foreseeable hazards and hazardous situations, which could affect patients, users or others, related to use of the medical device (IEC 62366-1:2016).

A market requirement is a requirement that is relevant to the procurement decision of those who purchase interactive systems either for a specific user group or the users themselves.

Market requirement

CPUX-F Definition

A requirement for an interactive system based on marketing policy aimed at maximising business opportunities, sales and use.

Market requirements are often referred to as customer requirements.

For example: “The website must be at least as usable as those of the two strongest competitors”.

Organisational requirements formulate rules that users must follow when working within organisational processes.

Organisational requirement
CPUX-F Definition
An organisational rule that users have to follow when conducting their tasks.

Examples:

- The radiological assistant must observe the patient while adjusting the patient table.
- A sales representative must ask for written approval from the sales director for quotations that exceed €100,000.

In practice, sometimes the term “business requirement” is used.

- Business requirements provide an organisation with the rationale for developing or purchasing an interactive system. In contrast, organisational requirements are relevant for the operation of the interactive system and must be implemented within the project.

Domain-specific requirements specify the attributes of a correct and complete output of a task.

Domain-specific requirement
A requirement that defines the attributes of a correct and complete output of a task.

Examples of requirements related to the outputs of a task:

- An invoice must always state the customer’s order number.
- A newspaper article must always contain a title and an author.
- The cost of an insurance premium renewal must be calculated based on the rules of the insurance tariff.

1.2.2 Differentiate between stakeholder requirements and system requirements

System requirements are derived from stakeholder requirements and are the basis for the technical implementation of the interactive system.

System requirement
A requirement specifying what the interactive system must be technically capable of to meet one or more of the stakeholder requirements.

System requirements are always derived from stakeholder requirements. System requirements are not technical solutions. They specify the capabilities of the technical solution. In contrast, user requirements are the basis for the required interaction between user and interactive system (without prescribing the required technology).

Example of a system requirement that can be derived from a user requirement:

- User requirement:
  “With the system, the user must be able to recognise customers for whom a longer period of time than average has elapsed since their last order.”
- Resulting system requirement:
  “The system must calculate the time since last order for each customer and identify whether the average time has been exceeded.”

1.2.3 Verification versus validation

Verification determines if all requested stakeholder requirements and system requirements have been implemented as specified; validation determines that stakeholder requirements have been implemented appropriately from the stakeholders' perspective.

Verification

The process of determining if all requirements have been transferred into corresponding product characteristics.

Verification can be applied to confirm that all stakeholder requirements have been addressed in the design.

Example:

An inspection-based usability evaluation is a verification if it explicitly checks for the fulfilment of inspection criteria, such as user requirements and/or design rules. For example, the correct application of a style guide.

Validation

The process of determining whether all stakeholder requirements have been implemented appropriately from the perspective of all stakeholders.

Validation can be applied to confirm that solutions appropriately meet stakeholder requirements.

Example:

A usability test explicitly checks whether user requirements have been implemented appropriately and that the solution is usable from the perspective of users.

1.2.4 IREB and CPRE

There are international boards related to requirements engineering. UXQB publishes the curriculum for CPUX-UR. IREB publishes the curriculum for the "Certified Professional for requirements engineering" (CPRE).

IREB

A non-profit organisation with the name "International Requirements Engineering Board e.V." (IREB e.V.), publishes the curriculum for the "Certified Professional for Requirements Engineering" (CPRE).

CPRE

The "Certified Professional for Requirements Engineering" certificate from IREB e.V.

The "Certified Professional for Requirements Engineering" (CPRE) certificate focuses on requirements in general, whereas the CPUX-UR certificate addresses deriving user requirements in particular.

The main target group for the CPRE certificate are those involved in defining appropriate technical solutions.

The main target group for the CPUX-UR certificate are those involved in defining appropriate human-system interactions.

1.3 The context of use as the basis for user requirements

The context of use comprises of the users of the interactive system, their **goals** and **tasks**, the **resources** they use, and the **environment**(s) they are in while completing tasks and achieving goals. All of these components of the context of use can contain user needs that lead to user requirements.

It is the basis for the **context of use for design**. The context of use for design is a subset of the context of use, since typically not all user groups and not all tasks in the context of use are supported by the system.

Each task that a user completes can be subdivided into **subtasks**. Subdividing tasks results in a **task model**. Each task has a task model. The task model of the current context of use describes how users complete tasks before the new or revised interactive system is available. The **task model for design** is an adaption of the task model of the current context of use. Typically, it differs from the task model of the current context of use in that it takes into account the fact that tasks can be optimised in light of identified user needs, derived user requirements and newly available technology.

When users perform tasks, they create, modify, or simply inspect **task objects**. When solutions are being designed, the task objects of the current context of use can be fully or partially represented in the user interface of the new interactive system and may include additional information.

Learning Objectives	
1.3.a	Be able to precisely differentiate between users, goals and tasks, resources, and social and physical environment.
1.3.b	Be able to formulate goals as intended outcomes.
1.3.c	Be able to distinguish between task, task model, and task object • Know about the term “task model” and its importance for the description of tasks. • Know about the construct “task object” as the object for the completion of tasks. • Understand how task objects are represented in the user interface.
1.3.d	Be able to describe tasks as a “task model” with the components contextual precondition(s), subtasks and intended outcome(s) (contextual postcondition(s)).
1.3.e	Understand that the context of use for design is typically a subset of the given context of use and that it is used as a basis for deriving user requirements.

1.3.1 The four components of the context of use according to ISO 9241-11

The four components of the context of use according to ISO 9241-11 are:

- Users;
- Tasks and goals;
- Resources;
- Environment.

The interactive system itself is not part of the context of use. It is shaped around the context of use.

Users perform tasks to achieve goals.

Task
CPUX-F Definition
A set of activities undertaken in order to achieve a specific goal.

In general, tasks can be subdivided into subtask. These subtasks are supported by the future interactive system. For each subtask, the user requirements specify the interaction required with the system, in terms of what the user must be able to recognise, select, or input.

Subtasks either result from the initial task itself (“task-induced”) or from the specific implementation chosen for the interactive system (“system-induced”). System-induced subtasks are typically necessary but they do not contribute to the goal of the task.

Examples of tasks and subtasks:

- Renting a car is a task;
- Cancelling a reservation is a task;
- Compare available cars is a (task-induced) subtask;
- Deciding on a specific car range is a (task-induced) subtask;
- Registering on a car rental website is a (system-induced) subtask;
- Logging in is a (system-induced) subtask.

Goal
CPUX-F Definition
The intended outcome.

Intended outcomes exist for tasks, and for subtasks within a task, as subgoals.

Intended outcomes are observable conditions after a task has been completed.

Examples of goals in the form of an observable conditions:

- The bank customer has successfully initiated a money transfer.
- The driver has parked their car in the parking space without damage.
- The voter has placed the completed ballot slip into the ballot box.

Users perform their tasks in a specific environment.

Environment
CPUX-F Definition
The physical, social and technical conditions in which a user interacts with an interactive system.
It describes all of the circumstances or conditions under which a user performs their tasks.

Examples: Physical conditions

- Outdoors with street lighting at night;
- An open office with a high level of noise;
- A car on the road;
- At home.

Examples: Social conditions

- Other passengers on the bus;
- Colleagues in the office;
- Family members in the car.

Example: Technical conditions

- Connectivity to the internet;
- Access to electricity by means of power sockets;

- Access to public broadcast in the event of an environmental disaster.

Additionally, the social conditions include organisational conditions.

Example: Organisational conditions

- Hierarchy within an organisation;
- Corporate culture and habits.

Users make use of resources while performing tasks in the given environment.

Resource
CPUX-F Definition
All means required to achieve an intended outcome in the context of use.

Typical examples of resources are time, financial budget, physical and mental effort, hardware, software, and materials.

Resources can be divided into:

- Reusable resources, such as equipment, information, support, and;
- Exhaustible resources that run out, such as available time, physical and mental effort, financial resources, and materials.

Equipment includes hardware, software, and physical objects that are used by the user in combination with the interactive system to perform a specific task. The interactive system itself is not part of the equipment.

1.3.2 Differentiate and describe user groups

User groups are formed when primary, secondary, and indirect users are identified as stakeholders of the interactive system.

Examples:

- The surgeon is the primary user of a medical device designed to assist them during surgery.
- A trainer from the manufacturer of the surgical system is a secondary user of the system when training the surgeon.
- The patient being treated by the user of the medical device is an indirect user of the surgical system.

User groups can also be identified by considering differences in the components of the context of use: user characteristics, attitudes and experience of users, their tasks and goals, the physical and social conditions, and the resources used when completing tasks.

Examples:

- Different user characteristics: Users who prefer buying tickets online; users who prefer using ticket machines; users who prefer buying from the ticket counter.
- Different task characteristics: Users who commute to the same location each day; users who take business trips to different destinations each day.
- Different environmental characteristics: Users who travel second class; users who travel first class.
- Different resources: Users who have little time to reach the destination; users who have plenty of time to travel.

1.3.3 Identify and describe tasks and subtasks

A user's task can be divided into subtasks.

Subtask
CPUX-F Definition
A step undertaken to complete a task.

Subtasks are supported by the interactive system so that the user is more effective and efficient in the completion of the task. Not all subtasks can be supported by the interactive system. Subtasks are described independently of the interaction with the interactive system.

Subtasks are described as activities or decisions that have to be taken as part of the task. They do not describe the interaction with the interactive system. Subtasks are supported by the interactive system, in terms of information, choices and/or inputs

Example:

- Subtask: "Decide on a means of transportation for a specific destination".
- Supporting interactions provided by the interactive system to help users complete the subtask:
  - Overview of the buses, trains, taxis that are available, with departure time, arrival time, and costs
  - The option to select one available means of transport

Task model
CPUX-F Definition
A description of a task consisting of the reason for starting the task, the goal it supports, and the subtasks that have to be carried out in order to complete it.

Subtasks can be identified in the current context of use.

The purpose of a task model is to describe the subtasks within a task in a meaningful order, including the contextual precondition(s) and intended outcome(s). Intended outcomes are also called contextual postconditions.

A task model of the current context of use describes tasks and subtasks as currently performed ("as is"). In turn, the task model for design describes how the task is to be performed when supported by a new or revised interactive system.

The lifecycle of a task typically consists of the following phases:

- Plan;
- Prepare;
- Perform;
- Evaluate result;
- Communicate result.

Within these phases, the subtasks take place. Simple tasks might follow only a subset of these phases.

A task is always triggered by one or more contextual preconditions and ends with one or more contextual postconditions.

Excessive structuring of tasks into many subtasks with several hierarchical levels, leads to unnecessary complexity. Therefore, task models should be assessed for whether:

- Two tasks have been unintentionally merged into one (formulating the pre- and postconditions of the task at hand is a good test for this).
- Each subtask has been formulated as an activity or decision

- A user requirement has been unintentionally formulated as a subtask.

  Example: Task model of the current context of use

Task:	Clean the occupant's windows.
Contextual precondition:	The occupant perceives their windows to be so dirty that they dare not invite friends to visit.
Intended outcome: (Contextual postcondition)	The occupant perceives their windows to be clean.

Subtasks by phases of the task lifecycle:

Plan
- Decide when the windows are to be cleaned.
- Decide who should clean the windows.

Prepare
- Make water, detergent, a cloth, and a stepladder available.
- Clear the windowsill.

Perform
- Apply detergent to the window that needs cleaning.
- Remove the dirt from the window.

Evaluate result
- Check each window for the absence of dirt.
- Check each window for the absence of streaks.

Communicate the result
- Show the clean windows to the next person entering the house.

Task models of the current context of use typically need to be adapted to task models for design after considering how they will be supported by the interactive system.

Task model for design

An adaption of the task model from the context of use analysis is adapted for one of the following reasons:
- The context of use for design has been constrained (for example, only one specific user group will be supported).
- There are fewer subtasks due to automation, or additional subtasks due to constraints on the implementation (for example, a specific technology that imposes subtasks on the user).
- Users' tasks were simplified in early design.

Adapting task models can include deleting, adding, or changing subtasks within each supported task.

Example: Task model for design

Task:	Clean the occupant's windows.
Contextual precondition:	The occupant perceives their windows to be so dirty that they dare not invite friends to visit.
Intended outcome: (Contextual postcondition)	The occupant perceives their windows to be clean.

Subtasks by phases of the task lifecycle:

Plan
- Decide when the windows are to be cleaned.
- Decide who should clean the windows.

Prepare:
- Fill the cleaning robot with detergent.
- Position the cleaning robot.

Perform
- Instruct the cleaning robot to clean the window.
- Observe the cleaning process.

Evaluate result
- Check each window for the absence of dirt.
- Check each window for freedom from streaks.

Communicate the result
- Show the clean windows to the next person entering the house.

1.3.4 Identify task objects in the current context of use

After a task or subtask has been performed, three different forms of observable postconditions can be discerned:
- The user has created a new task object.
- The user has modified an existing task object.
- The user has generated new knowledge using an existing task object

Examples:
- The accountant has created a new invoice (task object) for a customer.
- The accountant has corrected the invoice (task object) based on a customer complaint.
- The customer has inspected the invoice (task object) and found a mistake within the invoice.

Task object

The objects that are created, modified, or inspected by a person to achieve the intended outcome(s) of a task.

Task objects can be identified in context of use descriptions.

To perform tasks efficiently with the interactive system, users need to be able to identify task objects. For this reason, task objects and their attributes must be represented in the user interface.

Task objects can be fully or partially represented in the user interface of the interactive system and may include additional information.

Examples:

- When executing the task “issue an invoice”, a person creates the task object “invoice”.
- When executing the task “clean the window”, a person modifies the task object “window”. Certain characteristics of the window have changed after executing the task. Before cleaning, it was dirty, afterwards, it appears clean.
- When executing the task “read the daily newspaper”, the person studies the task object “news article”.

It is important to understand the difference between an “as is” task object, in the context of use, and the “to be” task object, represented in the user interface, as the basis for design decisions.

User requirements are the basis for adapting “as is” task objects into task objects that fit the “to be” design. However, “as is” task objects may be insufficient when performing tasks with the interactive system. To ensure effectiveness, efficiency, and user satisfaction when using the interactive system, user requirements are used to specify the task objects in the user interface.

Examples of task objects represented in the user interface:

- A list of invoices (task object) in an invoicing system with the additional information, “unpaid”.
- A customer order in an order management system with necessary information, for example, “not yet confirmed”.
- A letter to a customer, in a word processing system with the address prepopulated.
- The boarding pass in an airline app shows that the flight is delayed.

1.3.5 Context of use for design

The context of use for design is usually a subset of the current context of use. Project constraints often only allow for the design of certain aspects of the context of use.

Context of use for design

A subset of the context of use, used as a basis for the design of an interactive system.

Examples of context of use information not considered during design:

- Physical attributes of males versus females are not considered when designing the interactive system.
- “Silver surfers” as an existing user group for a mobile app are not considered in its design.
- The use of the interactive system “on the road” is not considered in its design.

1.4 User requirements as a separate category within stakeholder requirements

User requirements are one category of stakeholder requirements. There are two types of user requirements. **Qualitative user requirements** refer to what users will be able to recognise, select, or input while using the interactive system to complete one or more tasks. Qualitative user requirements are derived from user needs identified in the context of use. **Quantitative user requirements** are required levels of usability, accessibility, user experience, and avoidance of harm from use that the interactive system must meet. Quantitative user requirements are typically specified by stakeholders as acceptance criteria. For example, in an insurance company, the head of the claims department might set a quantitative user requirement on the system, specifying that all users must be able to complete an insurance claim within 15 minutes.

The direct source for a user requirement is always one or more user needs. Other stakeholder requirements can indirectly lead to user requirements. User requirements can lead to other stakeholder requirements.

Not every user requirement contributes equally to the usability of an interactive system. While the implementation of a specific user requirement might be critical to the usability of the interactive system, the implementation of another user requirement may only affect usability in a small way.

Learning Objectives	
1.4.a	Understand that the usability of an interactive system can significantly depend on whether or not a single user requirement is implemented.
1.4.b	Understand that qualitative user requirements are the basis for designing the interaction between the user and the interactive system.
1.4.c	Understand how user requirements can not only lead to other stakeholder requirements but can also be derived from other stakeholder requirements.

1.4.1 Distinguish between qualitative and quantitative user requirements

User requirements are the basis for the development of the user interface.

A basic distinction can be made between:

- Qualitative user requirements which are the basis for the interaction between the user and the interactive system, and
- Quantitative user requirements as acceptance criteria for the interactive system.

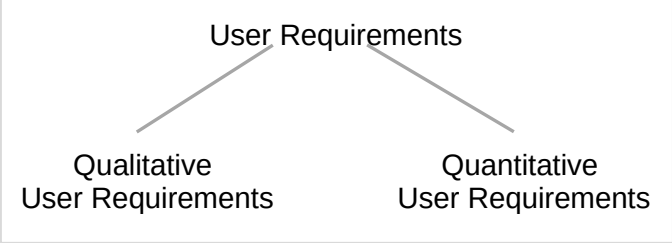

Figure 3 Types of User Requirements

1.4.2 Qualitative user requirements

Qualitative user requirements are not features; they provide the basis for features. It must be possible to trace each qualitative user requirement back to the underlying user need and corresponding context of use information. This is known as “Traceability”. Compare this with chapter 5.1.2 (Transform user needs into one or more user requirements).

Qualitative user requirement
CPUX-F Definition
A statement of what users must be able to recognise, select or input as part of conducting a task with the interactive system to meet identified user needs in a specified context of use.

Qualitative user requirements are specified with the aid of the following elements:

- User group;
- Type of usage (recognise, select, input);
- Optional special condition(s) of the context of use for which the user requirement is relevant.

The direct source for qualitative user requirements is always one or more user needs.
Indirect sources are:

- Human-centred quality objectives;
- Legal or regulatory requirements;
- Market requirements;
- Organisational requirements;
- Domain-specific requirements
- Requests;
- Identified or anticipated problems with usage;
- Interaction principles and heuristics that are used for the specification of user requirements.

Syntax rules for stating user requirements and a technique for deriving qualitative user requirements from user needs can be found in chapter 5.1.

Examples of reasonable qualitative user requirements:

- “With the system, the user must be able to recognise which cars are available within a specific price range.”
- “With the system, the user must be able to select a car with automatic transmission.”
- “With the system, the user must be able to enter (input) the rental period.”

Examples of incorrectly formulated qualitative user requirements:

- “The user interface must be usable and support all user tasks (this is too general).”
- “The user interface must have a big, red “rent this car” button (this describes a solution).”

Examples of a solution that represents the implementation of a user requirement:

- User requirement:
  “With the system, the user must be able to recognise that the washing cycle has finished.”
- Solution:
  The washing machine beeps three times and the word “End” is illuminated.

Examples of positive and negative consequences if user requirements are or are not implemented:

- User requirement:

"With the system, the user must be able to recognise that they will not receive their money until they have retrieved their cash card."

- Negative consequence the user requirement is not implemented:
  The user might leave their card in the ATM. In the worst-case scenario, their card is then stolen.
- Positive consequence if the user requirement is implemented:
  The user is not able to leave their card in the cash machine.

Examples of organisational requirements and domain-specific requirements that result from user requirements:

- User requirement:
  "With the system, the user must be able to input the patient's identification number."
- Organisational requirement:
  "The nurse who creates the patient records must assign an identification number to each new patient."
- Domain-specific requirement:
  "A patient identification number must always be part of a patient record."

1.4.3 Quantitative user requirements

Quantitative user requirements are acceptance criteria for the effectiveness, efficiency, and user satisfaction of the interactive system, for example, whether users can solve a particular task with the system in an acceptable time or with a specified maximum number of use errors.

Quantitative user requirement
CPUX-F Definition
A required level of human-centred quality to meet identified user needs expressed in terms of measures of usability, accessibility, user experience and avoidance of harm from use in a specified context of use.

The elements of a quantitative user requirement are:

- The stakeholder(s) who issued the requirement (for example, employer or legislator).
- The user group(s) who have to fulfil the requirement during use.
- The number or percentage of users who have to fulfil the requirement.
- The object of use (task object in the user interface, tool).
- Quantities (for example, maximum available time, error rate, precision).
- Any special condition(s) of the context of use under which the use takes place.

Sources of quantitative user requirements are:

- Quantitative user needs in the context of use (for example, "The patient must close their eyes within 60 seconds to ensure there is enough moisture on the surface of their eyes.")
- Stakeholders who define acceptance criteria for the usability of an interactive system (for example, for a usability test).

  Examples:

  - "80% of users who have used the car rental website at least twice must be able to rent an economy size car within 5 minutes from Frankfurt Airport (Germany) for two days starting tomorrow at 09.00 A.M."

Quantitative user requirements should always be verified against the user needs that provide the rationale for their existence, to avoid arbitrary levels of usability.

2 Plan context of use analyses

2.1 Identify the rationale and goals for the context of use analysis

The context of use is analysed to pinpoint context of use information which can help to identify user needs that are poorly understood. These can then be used as a basis for the specification of user requirements. There are different situations in which a context of use analysis takes place. Context of use analyses can be used to develop a new interactive system or to improve an existing interactive system.

As a starting point for planning the extent and focus of a context of use analysis, **human-centred quality objectives** should be set by the stakeholders of the system. These are project objectives from the perspective of users of the interactive system. In addition to business objectives, they describe what must be achieved for the users of the interactive system with regard to usability, accessibility, user experience, and avoidance of harm from use.

Learning Objectives	
2.1.a	Know the different reasons for carrying out context of use analyses.
2.1.b	Be able to define human-centred quality objectives for the design of the interactive system (in addition to general project objectives).

2.1.1 Typical situations in which context of use analyses are carried out

Typical situations in which a context of use analysis is carried out within a project for designing an interactive system are:

- A new interactive system is to be developed for which there is no existing version.
- The uptake or intensity of use of a given interactive system needs to increase.
- The efficiency of an interactive system for a specific user group needs to be improved.
- The functionality of an existing interactive system is to be extended (an increase of effectiveness, for example, for a new user group, or for a new task for a currently supported user group)
- Relevant user requirements are to be identified for the execution of a usability evaluation of an interactive system.

2.1.2 Human-centred quality objectives within a project

In addition to project objectives, human-centred quality objectives should also be identified at the beginning of a project.

Human-centred quality objective
CPUX-F Definition
An intended outcome of the development of an interactive system for the user, relating to usability, accessibility, user experience, or avoidance of harm from use.

Human-centred quality objectives

- are general statements of outcomes to be achieved for users of the system; they are often provided by the interactive system's sponsor, and are defined from the perspective of its future users.
- are objectives of stakeholders regarding the human-centred quality of the interactive system to be created or revised. They are not yet user requirements.
- are specified during the planning of human-centred design activities. The user requirements engineer acts as a moderator, as they will have specialised knowledge about this type of specification. At the beginning of the project, explicitly specifying human-centred quality objectives helps adequately in setting the focus and scope of the context of use.
- can also be formulated as verifiable quantitative user requirements so that they can serve as acceptance criteria for the project.

Examples:

Human-centred quality objectives for usability:

- After purchasing a railway ticket, the traveller will be able to purchase an identical ticket at a later date without having to specify the ticket information again.
- At the first attempt, users must be able to use the web-based health record without help.

Human-centred quality objectives for accessibility:

- Blind users must be able to recognise and understand the content of the website.
- Wheelchair users must be able to buy tickets at all stations without assistance.
- Visually impaired users must be able to find all products in the store without assistance.

Human-centred quality objectives for user experience:

- Before using the expenses system, users must get the impression that they will save time by using it.
- Users must have the feeling of complete privacy when using the electronic voting booth.
- Users must feel very confident that they are not producing avoidable waste (for example capsules) when using an automatic coffee machine.

Human-centred quality objectives for avoidance of harm from use:

- When using a system for creating prescriptions, the user must not be able to prescribe incompatible drugs.
- People interested in buying a coffee machine must avoid producing hazardous waste (for example, capsules).
- Users must be able to recover data deleted from the system.

2.2 Determine the approach for context of use analysis

Context of use analysis is the process of planning, gathering, and documenting authentic **context of use information**, identifying the user needs in this information, and deriving the user requirements from those user needs.

As a general principle, a context of use analysis begins with gathering empirical information, such as facts about users, their goals and tasks, the resources they use, and the environments they are in. **Empirical information** is the basis for **constructed information**. User requirements and personas are examples of constructed information as they are derived from interviewing or observing representatives of one user group of the interactive system. In projects, context of use analysis is often referred to as **user research**. Individuals performing user research may be known as user researchers or user experience researchers. User research, however, encompasses all activities in a project or organisation that involve direct interaction with users. This includes context of use analysis as well as collaboration with users in design and usability testing. Therefore, user research has a wider scope than context of use analysis.

It is important to set the correct focus for a context of use analysis. The focus of an analysis of a context of use may be broad or detailed, depending on the context of use information the team already has. If the project team has little knowledge about the context of use, or team members disagree on aspects of the context of use, a **classic context of use analysis** is recommended and no assumptions about the context of use are made. If the project team has comprehensive knowledge and/or information about the context of use, a **model-based context of use analysis** is recommended, where known information on task models and user requirements is structured by the project team and focussed questions on unknown information are formulated that are then used for focussed interviews, observations, and focus groups.

Model-based context of use analysis fits well with the approaches **Lean UX** and **Design Thinking**.

Learning Objectives	
2.2.a	Understand that preliminary assumptions about the context of use are necessary as the starting point for collecting empirical information, that is, facts from the context of use.
2.2.b	Be able to apply different approaches for context of use analysis (classic versus model-based).
2.2.c	Understand that context of use analysis does not conflict with activities associated with “Design Thinking” or “Lean UX”, but that these approaches are also based on a context of use analysis, just as almost any human- and user-centred approach is.

2.2.1 Context of use analysis

During the process of performing a context of use analysis, context of use information is collected and documented.

Context of use information

Part of a context of use description containing user needs.

Every user need is based on at least one specific piece of context of use information. A single piece of context of use information does not replace a (coherent) context of use description.

Example:

In an observation at Munich airport, one finding was that the majority of passengers arriving at the airport used the underground train to leave the airport.

This piece of context of use information is one piece of a context of use description containing the user need: “After arrival, a passenger needs to have a valid ticket for the underground train so that they can reach their final destination, lawfully.”

The objective of the context of use analysis is to obtain empirical information about users, their goals and tasks, the environment in which they perform their tasks, and the resources they use.

Empirical information

Factual information that has been collected in the context of use through contextual interviews or observations.

Empirical information is the basis for constructed information.

Constructed information

Information that is derived from empirical information.

Examples of constructed information include personas, requirements and prototypes.

Information that cannot be traced back to empirical information is called an “assumption”. For example, it could be an assumption that bank employees would also like to sell insurance to their customers. A context of use analysis could reveal that this user group is afraid to sell insurance to bank customers.

Context of use analysis

The process of planning, gathering, and documenting authentic context of use information, identifying the user needs contained within them, and specifying user requirements

Possible approaches for context of use analysis are “classic context of use analysis” and “model-based context of use analysis”.

The term “context of use analysis” is also referred to as user research, whereas user research also includes evaluation with users.

User research

User research incorporates all activities during analysis, design, and usability evaluation of interactive systems where information is gathered together with users about:

- The context of use;
- Wants, likes and dislikes, and;
- Usability problems during the use of interactive systems.

Those performing context of use analysis must know about the meaning of the term user research so as to be able to communicate with people using the term in projects in a professional way. Context of use analysis and deriving of user requirements are key components of user research. Another key component of user research is usability testing.

2.2.2 Establish consensus about the approach in the project

In development projects, a consensus must be reached on the approach to context of use analysis within that project. To reach a consensus on the approach, the following steps can be taken:

1. Set human-centred quality objectives together with all project members.
2. Gather known information and assumptions about the context of use in a systematic manner.
3. Identify knowledge gaps and uncertainties about the context of use.
4. Devise appropriate research questions for context of use analysis.
5. Identify participants for surveys or interviews.
6. Decide on the approach to be taken (classic versus model-based).
7. Plan resources and ensure their availability.

2.2.3 The classic approach versus the model-based approach

The approaches to classic and model-based context of use analysis differ; both have advantages and disadvantages. The project team must decide on the approach to be taken.

Classic context of use analysis

A systematic approach to context of use analysis where context of use information is initially collected without making any assumptions. User needs are then identified, and user requirements are derived from the user needs.

Initial situations for the classical approach to context of use analysis:

- There is little or no information or disagreement about the components of the context of use;
- The customer’s questions to be addressed in the context of use analysis are general;
- There is a high expectation of discovery of new insights from the context of use analysis;
- The innovation potential that is achievable as a result of further developing the interactive system is largely unknown.

Procedure of the classic approach to context of use analysis:

1. reach consensus with the sponsor and other project stakeholders on selecting this as the approach for the project;
2. specify user group profiles for each user group;
3. determine the questions to be addressed;
4. perform contextual interviews, observations and/or focus groups;
5. document, analyse, and structure gathered context of use data.

Sometimes the project team is - or believes they are - sufficiently informed about the context of use. In such cases, a model-based context of use analysis is a good choice to engage the project team.

Model-based context of use analysis

An approach to context of use analysis where open questions about the context of use are collected based on:

- Known context of use information;
- Given task models;
- Known user requirements and use scenarios and/or;
- Existing prototypes.

Empirical information about the context of use is then gathered in a focused manner with the help of contextual interviews and/or observations.

Initial situations for a model-based approach to context of use analysis:

- The customer has information about the users, tasks, resources and environments
- The project members doubt that empirical data gathering with users will yield relevant insights;
- There is extensive experience in the use of an existing interactive system;
- A specific focus has been set for the context of use analysis;
- The context of use analysis is expected to yield very specific insights;
- The questions to be addressed in the context of use analysis are very specific and build on each other;
- The customer broadly seems to be aware of the user requirements.

Procedure for the model-based approach to context of use analysis:

1. identify assumptions and knowledge about the context of use;
2. specify user group profiles for each user group;
3. construct task models for each task to be supported;
4. anticipate user requirements for each task and its subtasks;
5. identify research questions to be answered in further context of use analysis, regarding uncertainties, gaps, and contradictions in the knowledge about the context of use for the project;
6. perform contextual interviews and/or observations and/or focus groups, if needed;
7. document, analyse, and structure the gathered context of use data.

In contrast to classic context of use analysis, model-based context of use analysis uses interview checklists that have more specific questions, and assumes to an extent that both task models and user requirements are known.

The more empirical context of use information is made available to the project team, the more sense it makes economically to do a model-based context of use analysis. See also Lean UX. The less empirical context of use information is available, the riskier the model-based context of use analysis approach is, especially when taking the immunisation trap into account.

Context of use information can be made available to project members via different forms of documentation of the context of use, as discussed in chapter 3.3.

2.2.4 Relationship to “Lean UX” and “Design thinking”

Context of use analysis is one approach within human-centred design. Other currently discussed approaches are “Lean UX” and “Design thinking”.

Lean UX

An approach to human-centred design that integrates principles and methods for usability and user experience into agile development, thereby achieving economic advantages.

Lean UX is based on a combination of different approaches: agile development, design thinking, and lean start-up.

Anyone performing context of use analysis must know about Lean UX in order to be able to communicate professionally with project team members.

A model-based context of use analysis is a suitable starting point for Lean UX. Instead of doing extensive user research upfront – as in the classic context of use analysis – hypotheses to be validated are derived from known context of use information and/or interviews with sponsors, the organisation, and the development team. Later, when first prototypes are available, these hypotheses are validated or questioned in contextual interviews, observations, or focus groups.

Lean UX is based on the idea of combining lean start-up, agile development, and design thinking.

Lean start-up suggests that everything is a hypothesis in the beginning and consequently needs to be validated. The team learns through experiments about the user and the market. Failure is part of the learning process. Not every hypothesis is validated and not every experiment provides the desired results.

Agile development processes form the basis of Lean UX. The iterative approach in teams and the realisation of small, well-defined packages makes it possible for small and quick tests to be regularly carried out. The results from tests are directly used in the next iteration (“sprint”) for further development.

In addition to Lean start-up and agile development, design thinking also serves as a basis for Lean UX. Through its solution-orientated approach, based on a deep understanding of the problem space, design thinking helps project members incorporate users' perspectives into the development process.

Design thinking

A systematic approach to finding solutions for complex problems of all areas of life, focussing on human values before considering technological or business constraints. It is a tool to tackle the unknown and as such is used to create products, services, or process innovations.

Anyone performing context of use analysis must know about design thinking in order to be able to communicate professionally with project team members.

Comprehensive and in-depth understanding of the problem space, and non-restrictive, creative exploration of solution ideas, are at the core of design thinking. Ideas for solutions are then adapted to the problem context.

This approach emphasises three important components:

1. People: Multi-disciplinary, collaborative teams that act fast using their collective intelligence, and create their own effective work process thereby achieving unique results.

2. Places: The best environment for ideas to unfold is in an open and flexible work environment with variable project rooms, movable tables and walls, a lot of space for visualisations and a great variety of materials for illustrating ideas, thoughts and work products.
3. Process: The team navigates through the solution space based on an open “error culture”, following an iterative, multilevel design process (understand, observe, define point of view, generate ideas, prototype, test).

Design thinking is a specific approach for working out a creative solution. However, user requirements are only implicitly considered in this approach; they are not explicitly formulated or documented for later use.

3 Gather and document context of use information

3.1 Select and recruit users for gathering context of use information

To gather valid empirical information about the context of use, it is important to interview and/or observe people who truly belong to the intended user group(s) for the interactive system. **User group profiles** specify attributes of users within each intended **user group**. If the interactive system is being developed for a specific organisation where users can be accessed directly, the user group profile is sufficient for recruiting users for contextual interviews, observations, and/or focus groups. If the interactive system is a commercial, off-the-shelf product or software for use across organisations, users that are typically unknown to the project team must be recruited. In these cases, it is necessary to create a questionnaire based on user group profiles that allows for verification of potential participants for contextual interviews, observations, and/or focus groups, to ensure that any participants belong to the intended user group(s). This questionnaire is called a **recruitment screener**.

Learning Objectives	
3.1.a	Be able to develop user group profiles as a basis for recruitment screeners.
3.1.b	Be able to develop recruitment screeners for selecting users for contextual interviews, observations, and/or focus groups.
3.1.c	Know how to recruit users for contextual interviews, observations, and/or focus groups.

3.1.1 Identify and document user group profiles

Valid context of use information can only be gathered by interviewing and/or observing existing or potential users of the interactive system. For this purpose, the user groups for the interactive system must be known.

User group
CPUX-F Definition
A group of users with the same or similar personal characteristics and contexts of use related to the interactive system.

Characteristics of a user group are documented in user group profiles, which can be used to recruit users for participation in a context of use analysis or an evaluation of the interactive system.

User group profile
CPUX-F Definition
A generalised description of a user group.

User group profiles describe the relevant characteristics, capabilities, and attitudes of intended users for the interactive system that is being designed or evaluated. There is no set format or content for a user group profile. User groups can be differentiated by characteristics, capabilities, attitudes, the tasks they perform, the resources they use when performing tasks, and the physical, social and technical environments they are in.

3.1.2 Create recruitment screeners based on user group profiles

Based on the user group profile, a recruitment screener can be created to ensure that participants for contextual interviews, observations or focus groups fit into the user group profile.

Recruitment screener

A set of criteria derived from the user group profile that participants in a context of use analysis must meet, and a set of questions that allow verification that the criteria are met.

A recruitment screener is used during recruiting to determine whether candidates fit the user group profile.

3.1.3 Recruit users

Depending on the project situation, the recruitment of users will proceed differently.

a) Procedure for custom development projects for a single organisation:
   1. Identify the user group profiles, with experts from within the organisation.
   2. Select users with the help of their managers.

b) Procedure for development projects that address multiple organisations within a target market:
   1. Create user group profiles for each user group.
   2. Create a recruitment screener for each user group.
   3. Identify potential users in potential client organisations (if necessary, together with a recruitment agency).
   4. Verify potential users using the recruitment screener.
   5. Recruit and invite verified users to attend contextual interviews, observations, and/or focus groups.

c) Procedure for development projects for users as direct purchasers of the interactive system:
   1. Create user group profiles.
   2. Create a recruitment screener for each user group.
   3. Identify potential users (if necessary, together with a recruitment agency).
   4. Verify potential users using the recruitment screener.
   5. Recruit and invite verified users to attend contextual interviews, observations, and/or focus groups.

3.2 Prepare and gather context of use information

Empirical context of use information is gathered using **contextual interviews**, **observations**, or **focus groups**. Typically, a **moderator** and a **note-taker** work as a team to gather data. Regardless of the chosen method, the moderator should apply the **master-apprentice model**.

Contextual interviews are used to gain a broad understanding of the components of a context of use and their interrelationship. Observations are used to gain an understanding of specific aspects of task execution. Focus groups are typically used for specific topics within the context of use, where a deep understanding is required from the perspective of different user groups. Whenever possible, **interviews** should take place in the physical environment where the participant normally performs the tasks that will be supported by the interactive system. This is known as a contextual interview. The moderator uses an **interview checklist,** which contains the questions that need to be covered. Questions are **open** and **neutral** to maximise the validity of the empirical information. **Closed questions** should be used sparingly and **leading questions** should be avoided altogether.

During an observation, users are observed in the physical environment where they perform the tasks; observations are documented. The **observer** does not interrupt, except if they need to ask a clarifying question.

A specific type of observation is a **cultural probe**, where users supply evidence of their own situations as part of their context of use. This evidence includes photos, notes, sketches, or other “probes”. This method can be applied if direct observation is not possible due to safety constraints, economic constraints, or if longer periods need to be observed. Another specific type of observation is the **diary study**, where users supply context of use information by filling in a standardised questionnaire at defined time intervals, for example, once per day, over a defined period.

During a focus group, a group of representatives from one or more user groups is led through a discussion by a moderator, based on predefined questions that stimulate the participants to give information on specific topics within the context of use.

When conducting interviews or facilitating focus groups, it is important to be aware that people describe their context of use based on their personal experience or **mental model**, i.e., their perception of themselves, others, the environment, and the things they interact with.

In interviews and focus groups in particular, users articulate user wants in addition to context of use information. User wants are typically not context of use information, user needs, or user requirements. User wants are typically desired solutions. User wants can lead to user requirements and must always be investigated for the underlying context of use in case any user needs can be identified.

Learning Objectives	
3.2.a	Know which methods exist for gathering context of use information.
3.2.b	Know how to decide which method or combination of methods to use.
3.2.c	Know what to pay attention to when preparing and conducting contextual interviews.
3.2.d	Know what to pay attention to when preparing and conducting observations.
3.2.e	Know what to pay attention to when preparing and conducting focus groups .
3.2.f	Know that data from interviews and observations are transitory unless logged or documented.

3.2.1 Gather qualitative information

When gathering empirical qualitative information, interviewers, observers, and moderators should adopt a certain attitude towards the participant, as described by the master-apprentice model.

Master-apprentice model

CPUX-F Definition
A technique for a successful interview: The interviewer treats the user as the master while the interviewer is the apprentice. The goal of the master-apprentice model is to understand users' goals and tasks in detail by learning them as an apprentice would.

When gathering qualitative data, one should

- differentiate between context of use information, user needs, user wants, and ideas for the future;
- ask for details of or observe frequently and infrequently recurring situations, tasks, and intended outcomes from the user's perspective;
- ask for details of or observe further information on the context of use, such as users, tasks and goals, social and physical environmental conditions, and resources.

3.2.2 Typically used methods for gathering context of use information

Typically used methods for the gathering of context of use information are:

- Contextual interviews;
- Observations;
- Focus groups.

The criteria for selecting or combining methods depends on the questions to be addressed in the context of use analysis:

- If all components of the context of use are of interest from the subjective perspective of users, a contextual interview is the appropriate method.
- If detailed information about the task execution needs to be collected objectively in the context of use, an observation should be chosen.
- If a deep understanding of the problem space is required for a specific topic and different perspectives of users need to be taken into account, a focus group is appropriate.

User self-reporting is a specific case of observation. A cultural probe as one kind of self-reporting is particularly useful for capturing information when user observation is impractical.

Cultural probe

A specific case of observation where the user observes themself and collects "probes" (traces) of their activity in the context of use, which they then provide for analysis, for example, photos, notes and sketches.

Another way of user self-reporting is the diary study.

Diary study

A specific case of observation in which participants record specific events, feelings, behaviours and interactions into a mobile app or paper diary to provide insight into the user experience of an interactive system.

Diary studies are particularly suitable for observing user behaviour, in the context of use, across a period of time. They are particularly useful when behaviours happen sporadically or

are unplanned, when actions take place over time, or when typical observation may influence user behaviour. They can also help in understanding what motivates users to act. Participants may be asked to record entries at specific intervals (twice a week; once a day), when contacted by the moderator and asked to record an entry, or when something happens (when the children return

from school; when the weather changes).When performing context of use analysis, a moderator who is neutral to the project, should interview or observe participants or perform focus groups.

Moderator
CPUX-F Definition
A neutral person who conducts a usability test session or a focus group session.
The interviewer in a contextual interview can be considered the moderator.

The term “facilitator” is a frequently used synonym for moderator.

Often, the moderator is accompanied by a note-taker who focuses on capturing user statements or observation data.

Note-taker
CPUX-F Definition
A UX professional who makes notes of usability findings during a usability test session, focus group or interview.

Using a note-taker allows the moderator to fully concentrate on performing the contextual interview, observation, or focus group.

3.2.3 Contextual interviews

This section describes the typical process, typical mistakes, and recommendations for contextual interviews. Interviews and contextual interviews are conversations between a moderator and a user about the context of use. A contextual interview takes place in the physical environment where the users perform their tasks.

Interview
CPUX-F Definition
A data-gathering method where carefully selected individuals are asked questions to gain a deep understanding of the context of use.

The purpose of a (contextual) interview is to gain a deep understanding of the context of use (users, goals, tasks, environments, resources) to fully identify the user needs within the context of use. This understanding must be gained independently of any solution currently in use.

Contextual interview
CPUX-F Definition
An interview that takes place at the location where users usually perform tasks related to the interactive system.

The combination of “contextual interview" and “observation” is often referred to as “contextual inquiry”.

In interviews, open questions are preferable because they stimulate users to give comprehensive information rather than short answers to questions.

Open question
CPUX-F Definition
A question in an interview that does not give any indication of the expected format or content of the answer.

In contrast to open questions, closed questions do not encourage the participant to be talkative.

Closed question
CPUX-F Definition
An interview question that requires an answer from a predetermined set of alternatives.

Interviewers should avoid closed questions. They stop users from expanding on their personal experience and reveal little detailed information on the context of use.

Examples of closed questions:

- ”Have you ever rented a car?”
- Corresponding open question: “Please tell me about the last time you rented a car.”

Questions should be open and neutral.

Neutral question
CPUX-F Definition
A question in an interview that has no built-in assumptions, and no frame that excludes anything or directs the reply in a certain direction.

Examples of neutral and open interview questions:

- What happened when you …?
- What do you mean by that? (A typical follow-up question to gain a deeper understanding).
- What options do you have now?
- What should the home page of the new car rental website look like?

In contrast to neutral questions, leading questions influence users in the direction of their response.

Leading question
CPUX-F Definition
A question in an interview that signals a preference for certain possibilities or attempts to direct the reply in a certain direction.

Example of a leading question:
“Would you like to have pretty colours on the home page of the new car rental website?”

Corresponding neutral question:
“What should the home page of the new car rental website look like?”
Note that the neutral question does not even mention colour.

Interview questions should be:

- Open rather than closed and neutral;
- Leading questions are bad interviewing practice and must be avoided.

Typically, questions for interviews are prepared in advance and listed in an interview checklist. The interviewer uses the interview checklist to ensure that all relevant questions are asked.

Interview checklist

A written list of suitable questions and cues used by an interviewer during an interview to make sure that all relevant topics are covered.

In an interview, the interviewer usually conducts a briefing and then asks the user questions about typical procedures that are relevant to the planned interactive system. The identification of user needs and subsequent deriving of user requirements demands qualitative analysis of information that has been gathered.

The typical procedure for planning, preparing, conducting, and documenting interviews is:

1. Identify stakeholders who are relevant for acceptance and processing of contextual interview results (for example, product managers and sponsors).
2. Invite stakeholders to join in the preparation of contextual interviews.
3. Together with stakeholders, create a user group profile for each user group being surveyed.
4. Together with stakeholders, create questions for the interview checklist for each user group being surveyed.
5. If participants in the contextual interviews do not work for the project's organisation and consequently need to be recruited externally, create a recruitment screener to make sure that interviewees are actual representatives of an important user group.
6. Meet with the interviewee at the scheduled time at their workplace, for the planned duration. The duration may differ substantially depending on the topic, for example, a contextual interview about "boiling eggs" might take 15 minutes, whereas a contextual interview about "follow up on unpaid invoices" might take 90 minutes.
7. If possible, plan for two people to conduct the interview – an interviewer and a note-taker.

Prepare:

1. Tell the interviewee about the topic and the goal of the interview. The topic is the collection of authentic information about the context of use; the goal is the subsequent deriving of user requirements).
2. Assure interviewees and their employers that the interview documentation will be anonymised.

Perform:

1. Introduce yourself as interviewer and note-taker.
2. Conduct the contextual interview using the questions from the interview checklist.
3. If any statement of the interviewee is not understood, paraphrase what has been said and ask further questions:
   a) If the interviewee leaves the contextual level and starts proposing solutions, let the interviewee describe those solutions and then ask in what context that solution would help and why, allowing them to describe that context in detail.
   b) If the interviewee leaves the contextual level and starts talking about problems with an interactive system, let the interviewee describe those problems and then ask in what context these arise and why, allowing them to describe that context in detail.
4. Write down all the statements of the interviewee.

5. At the end of the interview, ask the interviewee if there are any further topics or statements they would like to address beyond what has been talked about and record any further statements.
6. End the interview; for the first interview, the interviewer and note-taker should talk about the interview to understand whether anything needs to be optimised before the next interview.

Document:

- Write down the statements of the interviewee as a coherent piece of text (as-is scenario) so that stakeholders get a comprehensive overview of the context of use for the interviewee without any need for further inquiry.
- The as-is scenario can be structured in line with the central questions of the contextual interview. This helps stakeholders to compare statements across interviews later on. The context of use description can also be structured without reference to the central questions.

Quality criteria for conducting contextual interviews are:

- The mindset of the interviewer:
  application of the master-apprentice model, level of curiosity, etc.
- Methodical competence:
  competent preparation (creating the interview checklist), execution with mastery of basic interviewing techniques (questioning techniques like active listening, paraphrasing (repeating what was said with one's own words), verbalising (reflecting on non-verbal or body language), as well as identification of underlying user wants in what is said (repeatedly asking: "why", "why", …).
- Domain-specific competence:
  a basic understanding of the respective domain, to be able to ask the right questions and be able to understand the answers.
- Execution of pilot interviews with feedback:
  training as well as testing the checklist.

A contextual interview falls into the category of "semi-structured" interviews because the interview will lead to a conversation that is based on a list of pre-defined questions.

If necessary, after performing an interview, the interview checklist can be revised based on the insights which might have led to additional questions, so that these can be addressed in subsequent interviews.

Typical mistakes when conducting contextual interviews are:

- Participants are invited who have not actually experienced the topic(s) but instead talk about their assumptions, which are then falsely recorded as valid knowledge or arguments.
- The interviewer sticks religiously to the interview checklist (the content and order of questions) instead of adapting to what is learned during the interview.
- The interviewer gets lost in the conversation and important questions are not addressed.
- The interviewer switches between levels during the interview (context of use, user needs, user requirements, solutions) when paraphrasing.
- The interviewer repeats literally what is said ("parroting") instead of verifying if the statements of the interviewee were understood.
- The interviewer adds constructed contextual information.
- The interviewer asks leading or closed questions, or does not follow the master-apprentice model.

3.2.4 Observations

This section describes variants of observations, the typical process of observation, typical mistakes, and recommendations for conducting observations.

Observation
CPUX-F Definition
A method for gathering contextual information relating to user needs in which a UX professional watches users who carry out tasks using the interactive system.

An observation should take place in a context that is as natural as possible, for example, at a user's workplace. The observer does not interfere, except if they need to occasionally ask a clarifying question.

If no interactive system is used, existing manual procedures should be observed.

There are several variants of an observation:

- Structured versus unstructured
  A structured observation is directed and planned. Fixed observation schemes and categories are used to reduce the subjectivity of observers. Unstructured, “open” observation is undirected, unplanned, subjective, and open to anything that occurs. It is therefore well-suited for forming hypotheses.
- Participatory versus non-participatory
  When conducting a participatory observation, the observer participates in the execution of the activities to be observed. When conducting a non-participatory observation, the observer is completely passive regarding the activities that are observed. Note: A participatory observation where the observer interacts with the observed environment by himself is also called “ethnographic study”.
- Open versus undercover
  When conducting an open observation, the observer is identified as such. Other people involved in the observation know the reason for the observer’s presence. In an undercover observation, the observer disguises their identity – for example, they claim to have a different role or interest in the activity – to influence the observed situation as little as possible.
- In the field versus in the lab
  A field observation happens in the natural surroundings of the people being observed. When conducting an observation in a lab, the lab is modified according to the observation objectives and resembles a situation that is as realistic as possible for people being observed, and supports natural behaviour as far as possible.
- The specific case of self-observation (user self-reporting, see also cultural probe)
  The user observes themself by keeping a diary (diary study) and collecting “probes” (traces) of their activity in the context of use which they then provide for analysis (for example, photos, notes, sketches).

A typical procedure for planning, preparing, performing, and documenting an observation is:

Planning:

1. Identify stakeholders who are relevant for accepting and processing observation results (for example, product managers and sponsors).
2. Invite stakeholders to help prepare the observation.
3. Together with stakeholders, create a user group profile for each user group being observed.
4. For each user group, clarify which tasks are to be observed and which questions

stakeholders have in mind.
5. If the people being observed do not work for the project's organisation and consequently need to be recruited externally, create a recruitment screener to make sure that observed people are actual representatives of an important user group.
6. If possible, plan for two people per observation – an observer and a note-taker.

Prepare:
1. Meet the participant being observed at the scheduled time and for the planned duration (90 to 120 minutes) where the observation is taking place.
2. Tell the participant about the topic and goal of the observation. The topic is collecting authentic information about the context of use and the process of tasks; the goal is the subsequent deriving of user requirements.
3. Let participants describe:
    a) Which tasks they typically perform;
    b) With whom they cooperate in the context of those tasks, and;
    c) What the outcome of each task is.
4. Name the tasks to be observed.
5. Assure participants and their employers that any documentation of the observation will be anonymised.
6. Obtain written consent from participants for capturing observed information.

Perform:
1. Introduce yourself as observer and note-taker.
2. Ask the participant to state the name of the task they are about to perform.
3. Observe the participant during the execution of the task.
4. Note in brief all observations and statements so that a coherent text can be created afterwards.
5. Ask questions for clarification whenever necessary.
6. End the observation.
7. Thank the participant for their cooperation and support.

Document:
- Write down observations and statements as a coherent text (as-is scenario) so that stakeholders get a comprehensive overview of the context of use for a participant without any further inquiry. Take photos as illustrations for text.

Quality criteria for performing observations are:
- Observations must be objective, reliable, and valid, and thereby, independent of the observation process, the observer as a person, and any other effects.
    - Objective: the results of the observation are independent of the observer, meaning that two observers come to the same result.
    - Reliable: observation reliably provides the same results as long as nothing changes in what is being observed.
    - Valid: observation measures exactly what it is supposed to measure.
- Observations must be documented as efficiently as possible. For example, by using previously created recording sheets to capture activities quickly and automatically, maybe through several observers at once or with support from a technical system.
- Observations should be performed by at least two people (whenever possible): A moderator and a note-taker.
- Observers must prepare together for observation, to familiarise themselves with the task at hand and to ensure that the observation is objective, reliable, valid, and efficient.

Typical mistakes and problems in performing observations:

- Observations can get distorted as a result of previous knowledge and assumptions about the activities being observed.
- Observations can get mixed up during evaluation and interpretation.
- Premature evaluation can occur if abstraction of assumed findings is done too early.
- The more familiar the observer becomes with the activities being observed, the more their concentration and reliability of observation decreases.

Observations should be conducted by briefed observers.

Observer

CPUX-F Definition
A person who watches users in an observation, usability test session or focus group.

In context of use analyses, users that discuss or carry out tasks that are related to the interactive system are observed.

Observers must not interfere with the observed activity (except in participatory observation). However, observers may be actively involved in the analysis of the results.

3.2.5 Focus groups

This section describes the typical process, typical mistakes, and recommendations for performing focus groups.

Focus group

CPUX-F Definition
A moderated discussion of predetermined subjects and questions between members of one or more user groups.

The results of a focus group must be viewed as context of use information that needs to be analysed afterwards to establish user needs and derivable user requirements.

Conducting focus groups is especially helpful if:

- A deep understanding of the problem space is required for a specific topic, thereby, taking different perspectives into account;
- Ideas need to be developed, or;
- Requirements for the solution need to be worked out to consider different points of view.

The moderated discussion focusing on one specific topic or question results in thorough processing by participants, so can unearth valuable insights and knowledge.

A focus group is a rather expensive qualitative method that requires not only good preparation and professional execution but also a detailed recording of notes, and time-consuming analysis, for example, the development of affinity diagrams.

The typical procedure of planning, preparing, conducting, and documenting focus groups is as follows:

Planning

1. Identify stakeholders who are relevant for accepting and processing focus group results (for example, product managers and sponsors).
2. Invite stakeholders to help prepare the focus group.
3. Clearly formulate the object of interest (for example, a product idea or a topic) and goal to be achieved in the focus group (for example, “What does the actual context of user group(s) look like with regard to the product idea?” or “Understand all that is included in

the topic ‘heating’.”).
4. Together with stakeholders, create a user group profile for each user group involved.
5. For each user group involved, clarify which open questions are best suited as a starting point for discussion.
6. If participants in the focus group do not work for the project’s organisation and consequently need to be recruited externally, create a recruitment screener to make sure that participants are actual representatives of an important user group.
7. Invite participants to attend at the scheduled time and planned duration (90 to 240 minutes).
8. If possible, plan for two people to facilitate – a moderator and a note-taker.

Prepare
- Create an agenda based on the objectives for the focus group and the topic(s) in question.

Perform
1. Introduce yourself as moderator and note-taker.
2. Explain to participants the goal of the focus group and the approach taken.
3. Ensure participants and their employers that any documentation of the focus group will be anonymised.
4. Clarify the code of behaviour for interaction between participants and with the moderator, for example, raising your hand before speaking.
5. Present the topic to be discussed.
6. Ask questions according to the moderation guidelines.
7. Stimulate discussion among participants.
8. When participants make a statement, summarise it in a short complete sentence and review it with participants.
9. Write down every statement of every participant.
10. When all questions have been discussed, thank all participants and end the focus group.

Document
1. Assign each statement to a topic or question of the focus group.
2. Formulate recommendations for action based on the statements.
3. Include statements in context of use descriptions if reasonable.

Quality criteria for performing focus groups:
- Adhere to a structured process using well-chosen methods for stimulating discussion among participants and for collecting and discussing their thoughts and ideas.
- Ensure a reasonable number of participants (typically 5 to 8) enables and fosters a lively discussion.
- Ensure a balanced composition to the group, for example, participation of those with operational expertise versus laypeople, or a lack of diversity in the topics and questions for discussion.
- Wherever possible, ensure observations are performed by at least two UX professionals – a moderator and a note-taker.
- Demonstrate moderation expertise – organise the group without giving your own input. Handle group phenomena, such as individuals dominating or leading, or the tendency for groupthink (answers focused on minimising conflict and reaching consensus).

Typical mistakes regarding the procedure of conducting focus groups:
- Insufficient expertise in moderation (see also the quality criteria), especially:
  - Previous knowledge and assumptions of the moderator about the topics and questions being discussed influence how the discussion is stimulated; the discussion

may no longer be about the opinions of the participants, and/or logging of statements from participants might be distorted.
    - Insufficient control of dominating participants who are leading or shouting down other participants.
- Recruiting participants who cannot discuss the topic(s) from their own experience or conscious non-experience, but instead talk about their assumptions, which are then falsely recorded as valid knowledge or arguments.
- The exploration phase for each topic is too short, leading to early abstractions (assumed interrelations) and results or premature decisions without considering every aspect of the topic.

3.2.6 Collect and document context of use information

Contextual interviews, observations and focus groups provide insight into the user's mental model.

Mental model
CPUX-F Definition
The concept people have built of themselves, others, the environment and how things with which they interact work.

Mental model is also popularly defined as a person's thought process about how something works in the real world.

People form mental models through experience with interactive systems, training and instruction. The mental model of an interactive system is formed largely by interpreting its perceived actions and visible structure. Expectations resulting from the use of other or similar systems are also of importance in the formation of mental models.

If a user's mental model of an interactive system is incomplete or contradictory, then the user will not be able to easily use the interactive system. Disclosed mental models are important sources of information for identifying user needs.

Example:

The users of an office dishwasher always assume that someone else has emptied the dishwasher. As a result, they put their dirty dishes in the dishwasher without first checking to see if the dishwasher has been emptied.

- User need: Office workers need to know if the dishes in the office dishwasher are clean or dirty to prevent clean and dirty dishes from getting mixed up.

Mental models can be captured in contextual interviews by using the "teaching back" technique: The interviewee is asked to explain ("teach") how they go about performing tasks with the help of the interactive system, how they operate the system, and how the system itself operates. Teaching back and paraphrasing are techniques that can be applied when following the master-apprentice model during a contextual interview.

During a contextual interview, users can express user wants. Interviewers should question these wants instead of jumping to hasty conclusions or constructing interpretations. User wants can be collected during context of use analysis. For solutions to be accepted, it is important that user wants are taken into account. They should be documented in the form of solution proposals, together with the user requirements for which the user wants are possible solutions. However, user wants must not be the starting point for developing a solution.

User wants are explicitly stated by users in contextual interviews, whereas user needs are

typically implicit and only found by the user requirements engineer through analysis of context of use information and problem descriptions.

User wants may relate to psychological needs that are relevant for product design; thus, psychological needs can be identified by investigating user wants in contextual interviews.

Psychological needs are listed in Table 3 below.

Table 3 Psychological needs

Psychological Need	Description
Autonomy	Feeling that you are the cause of your own actions rather than feeling that external forces or pressures are the cause of your action. For example, deciding to leave the office as planned instead of waiting for an update to finish.
Competence	Feeling that you are very capable and effective in your actions rather than feeling incompetent or ineffective. For example, being able to use known shortcuts to perform a task in one step instead of following a multi-step wizard to do the same.
Relatedness	Feeling that you have regular and intimate contact with the people who care about you rather than feeling lonely and uncared for. For example, being able to see that colleagues start their work in the morning, even if they are not in the same office.
Popularity	Feeling that you are liked, respected, and have an influence over others rather than feeling like a person whose advice or opinion nobody is interested in. For example, being able to see that colleagues frequently use a document that you worked on for a long time.
Stimulation	Feeling that you get plenty of enjoyment and pleasure rather than feeling bored and under stimulated. For example, a visual representation of a growing tree symbolises the progress of your work. An unbalanced treetop signals missing content; the colour of the leaves, their relevance.
Security	Feeling safe and in control of your life rather than feeling uncertain and threatened by your circumstances. For example, knowing that all necessary steps related to specific demands have been taken instead of being nervous while waiting for feedback.
Meaning	Feeling that you are developing in line with your potential and making life meaningful rather than feeling stagnant, and that life does not have much meaning. For example, if a website donates to reforestation with money collected through searches, the user immediately sees how many trees will be planted thanks to their search activity, or the activity of others.

3.3 Create context of use descriptions based on context of use information

A **context of use description** serves different purposes, one of which is to inform stakeholders, and establish awareness and empathy for the problem space of users of the interactive system; another purpose is to identify user needs and derive user requirements for the interactive system.

There are various forms of context of use descriptions. At the beginning of a project, a summary context of use description is drafted as a starting point, listing the different user groups, the tasks they perform to achieve goals, the resources they use, and their environment(s).

Summary context of use descriptions

- are typically a mixture of known context of use information and assumptions made by the team.
- contain summary information across user group profiles.
- are typically together with user group profiles adapted based on insights from contextual interviews, observations, and/or focus groups.

Detailed context of use descriptions

- document empirical facts collected from contextual interviews, observations and/or focus groups
- are either narrative context of use descriptions or model-like context of use descriptions.

Common forms of narrative context of use descriptions include **personas** and **as-is scenarios**. While personas mainly serve the purpose of creating empathy for a user group, as-is scenarios are suitable for identifying user needs and systemically deriving user requirements. As-is scenarios reflect the context of use from the perspective of one user group, comprehensively describing task and goals, the interrelationship between tasks, resources, and the social and physical environment, in the form of narrative text.

The main difference between an as-is scenario and a use scenario is that as-is scenarios focus on describing the components of the current context of use, while use scenarios focus on the intended situation of use of the interactive system under development.

Therefore, a use scenario is not a context of use description because it describes the intended use situation with the interactive system under development. It is one description of the context of use for design and thus a first textual prototype of the system under development.

Model-like context of use descriptions provide structural representations of the context of use. Common forms include **user journey maps**, **affinity diagrams**, and **goal catalogues**. While model-like context of use descriptions support the efficient communication of aspects of the context of use among the project team, as-is scenarios tell the full story and are the primary source for identifying user needs.

Learning Objectives	
3.3.a	Know about the types of context of use descriptions
3.3.b	Know that sometimes, user group profiles have to be adjusted based on information from contextual interviews, observations and/or focus groups that have been conducted
3.3.c	Understand the role of personas in comparison with user group profiles
3.3.d	Be able to document context of use information in a way that minimizes the need for further enquiries and allows other team members and stakeholder to gain a deep understanding of the context of use

3.3.1 Context of use descriptions

Interviews, observations, and focus groups typically yield a large amount of context of use information. Context of use descriptions serve to make context of use information tangible, thus facilitating communication with project stakeholders.

Context of use description

CPUX-F Definition

A set of deliverables describing the users, goals, tasks, resources and environments identified through the analysis of observations, contextual interviews, focus groups and user surveys.

A

There are two forms of context of use description: summary and detailed.

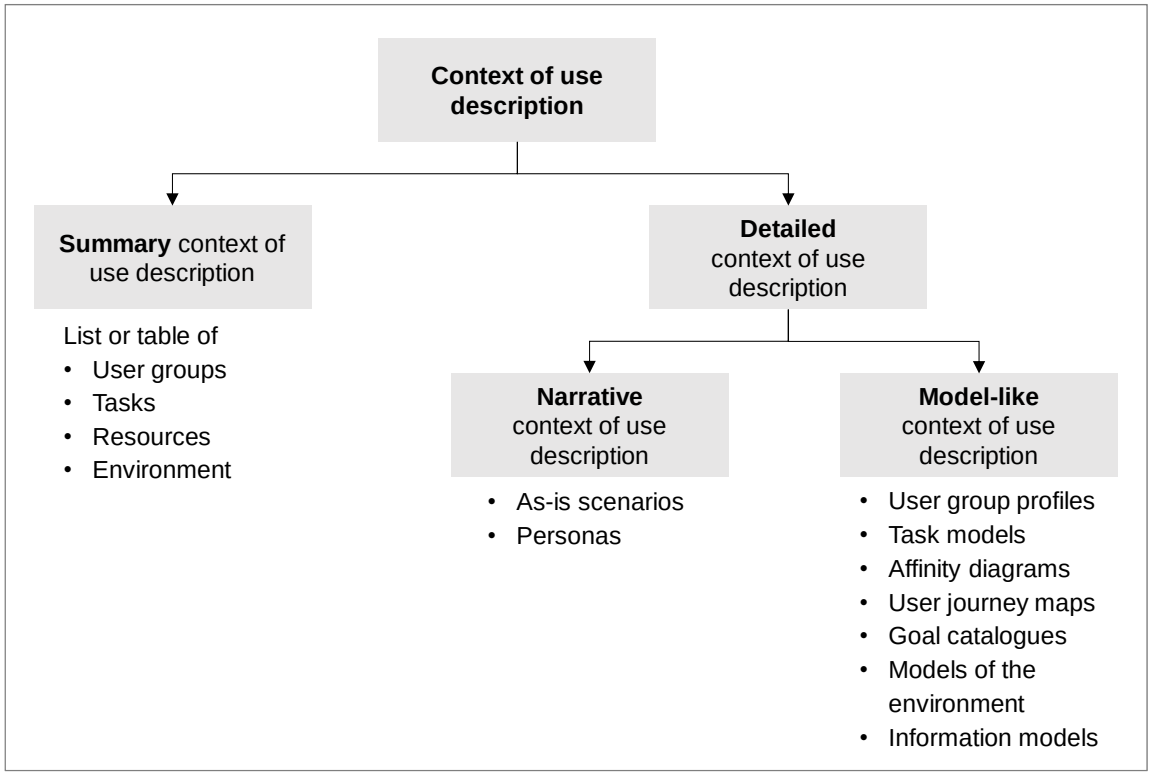

Figure 4 Types of context of use descriptions

Summary context of use descriptions contain keywords for user groups of an interactive system, their tasks and goals, resources, and physical, social, and technical environments. Typically, at the beginning of a project, summary context of use description contain what is

already known about users and their tasks (before any context of use analysis). For a model-based context of use analysis, this is usually the starting point.

Summary context of use descriptions are especially helpful in planning a context of use analysis. However, they are not substitutes for detailed context of use descriptions. Once the context of use has been analysed, summary descriptions can be adapted and used to communicate a summary of the results to the project stakeholders.

A detailed context of use description provides a description of users, their goals and tasks, resources, and physical, social, and technical environments. Detailed context of use descriptions are typically documented in narrative form; model-like descriptions are derived from these.

Common approaches to documenting contexts of use are as follows:

1. Narrative descriptions
- As-is scenarios describe how users behave when performing their tasks
- Personas are personalised illustrations of users within a user group, often representing the users in as-is-scenarios

2. Model-like descriptions
- User group profiles describe important characteristics of user groups
- Task models and task objects describe the structure of tasks and subtasks as well as the object of the task execution
- User journey maps describe the expectations, behaviours, and emotions of users across all touchpoints of the interactive system or organisation
- Affinity diagrams describe the semantic structures seen by users
- Goal catalogues describe the different goals relevant to different user groups

Other model-like descriptions, not detailed in this document

- Models of the environment (social, physical, and technical) describe important characteristics of all attributes of the environment.
- Information models describe the informational structure of each task model and task object.

Detailed context of use descriptions provide a comprehensive overview of the context of use for anyone not involved in contextual interviews and observations. Detailed descriptions enable structured identification of user needs and deriving of user requirements.

Model-like descriptions are the most appropriate approach for communicating context of use information to team members and stakeholders or for identifying missing or incomplete context of use information.

3.3.2 Narrative notations for context of use descriptions

The term “narrative” refers to documented context of use information that explicitly tells a story, based in reality, in the form of coherent text.

As-is scenario

CPUX-F Definition
A narrative text description of how a user currently completes one or more tasks in the current context of use.

As-is scenarios describe the current context of use and serve as a basis for identifying user needs and deriving user requirements.

As-is scenarios mainly describe the interplay between the components of a context of use.

By describing the behaviour of users, as-is scenarios reveal problems in the given context that reduce the efficiency of task performance.

Example:

John Miller is a business traveller who takes several flights in the course of a week. He prefers to take his car to the airport, but he misses a flight every now and then, and regrets not having taken a taxi or train to the airport. Today it happened again: “Again and again I just underestimate the time needed to navigate those long car queues at the entrance to the car park, and the huge walking distance to the gate.”

While the as-is scenario describes the current context of use and all its components (users, tasks, goals, resources, and environment (social, physical, and technical)) in narrative style, a persona description represents a user group’s attributes, attitudes, and goals, in narrative form.

Persona
CPUX-F Definition
A description of a fictitious but realistic user and what they intend to do when using an interactive system.

Personas are not real; they are representations of real users based on empirical data from observations or interviews.

A persona description

- typically has a name and age, some background information, and goals and desires that illustrate the user group.
- should include information regarding the persona’s knowledge about and interest in the subject matter of the interactive system.
- often includes a photo.
- provides a “face” to users, to give all project members a good idea of who the users of the interactive system will be, what characteristics they have, what motivates them and what personal goals they have.

There are two types of persona: primary and secondary. Primary personas represent the main target user group, whereas secondary personas provide insights into further goals or characteristics that are relevant for deriving user requirements but that would overload the descriptions of primary personas. For example, a secondary persona for an alarm clock is a light sleeper, while the primary persona often sleeps through their alarm.

Personas based on assumptions are called proto-personas; they are used in Lean UX . They can be used as a starting point for a model-based analysis of context of use. “Anti-personas” can be used to represent users for whom the interactive system has explicitly not been designed. For example, a safe-cracker is an anti-persona for a bank safe, and their characteristics would lead to requirements for the bank safe. Someone who routinely avoids paying for parking spaces is an anti-persona for a chargeable parking space, and their characteristics might lead to requirements for the interactive system used to access the parking space.

Quality criteria of the development of personas:

- A persona is created by those who analysed the context of use.
- A persona description does not represent a real person but is equivalent to the description of a real person.
- A persona combines characteristics of real users of a single user group.
- A persona description contains all important characteristics and the most important

personal goals.
- A persona must be based on empirical data and must not be purely imaginary.
- Proto-personas must be clearly labelled as such.
- Each user group should have between two and five (maximum) personas. It can be difficult to work with more than five personas.
- As they are representations of real users, personas must be checked regularly and modified when necessary, as the context of use can change over time.
- The responsibility for the creation, quality, and maintenance of personas lies with the product team, even if the personas were initially created by others (such as an agency). All of the context of use information used to create and maintain personas should remain available to the product team.

3.3.3 Model-like notations for context of use descriptions

“Model-like” refers to documented context of use information where the structure is explicitly visible, that is, in the form of lists, tables, diagrams, and other forms of structured visualisations.

Summary context of use descriptions, user group profiles, and task models are model-like context of use descriptions that have been introduced earlier in this document.

User journey maps can also be used as a means of communicating the given context of use in a compact form to stakeholders.

User journey map

CPUX-F Definition
A linear depiction of a user’s interaction with the interactive system and the organisation behind it covering encounters that influence the user experience.

User journey maps extend beyond the actual use, for example, from discovery of the product to purchase, to subsequent usage of the product. All touchpoints are considered, including the user’s first contact with the interactive system, for example, “How I heard about this new interactive service”) to direct task-oriented interaction, to communicating about the system's assumed user experience with others, for example. writing a report to colleagues about users' experience with the new system. User journey maps do not replace as-is scenarios or use scenarios.

Affinity diagrams are suitable for structuring information and gaining insights on various levels, when large amounts of qualitative context of use information need to be analysed.

Affinity diagram

A structured format for documenting results from the analysis of qualitative data.
- Ideas or statements belonging together are sorted into groups in a bottom-up fashion, with each group assigned a heading.
- The grouping and subsequent naming of related groups make up the typical structure.

Ideas or statements are first noted on cards (“affinity notes”) and then hierarchically grouped in several iterations. Each group of items or each group of groups gets an appropriate heading. Affinity diagrams are typically created via collaborative efforts, and regrouping is allowed. The structure of an affinity diagram (levels: goals, topics, relationship between items) can be better visualised using cards of different colours.

In studying context of use information, one learns a lot about users’ goals. These can be

structured and listed in a goal catalogue.

Goal catalogue

A tabular display of all identified goals that are relevant for user behaviour in the context of use. Each goal associated with a valid user group or persona must be included. The goal catalogue helps verify whether relevant goals have been identified and described for a user group and identify any comprehension problems or gaps.

4 Identify user needs in context of use information

4.1 Systematically identify and formulate user needs

User needs are contained within context of use descriptions although in most cases they are not explicitly stated. Analysis of a context of use description reveals user needs, providing the basis for subsequent user requirements. Identified user needs are not necessarily satisfied in the current context of use, so they provide a rationale for the design of the interactive system.

There are three types of user needs:

- **Informational needs** state the specific information needed for a person to complete a task and achieve one or more intended outcomes.
- **Resource needs** state the specific resource that a person needs to complete a task and achieve one or more intended outcomes.
- **Competency needs** state a specific skill or competence that a person needs to complete a task and achieve one or more intended outcomes.

It is essential that consistent phrasing is used when stating user needs, to ensure that they are clear and can be communicated to others.

User needs always include a detailed description of the intended outcome and the prerequisites needed (information, resources, or competence) to achieve that outcome.

Context of use information must be analysed to identify the intended outcome and the prerequisites required to achieve it, to systematically identify user needs.

Learning Objectives	
4.1.a	Know about and be able to differentiate between types of user needs
4.1.b	Know about and be able to apply quality criteria when stating user needs
4.1.c	Be able to precisely recognise user needs in context of use information
4.1.d	Be able to formulate user needs in a way that enables requirements to be derived from them
4.1.e	Understand the difference between user wants and user needs

4.1.1 Types of user needs

User needs can be categorised as informational needs, resource needs, or competency needs.

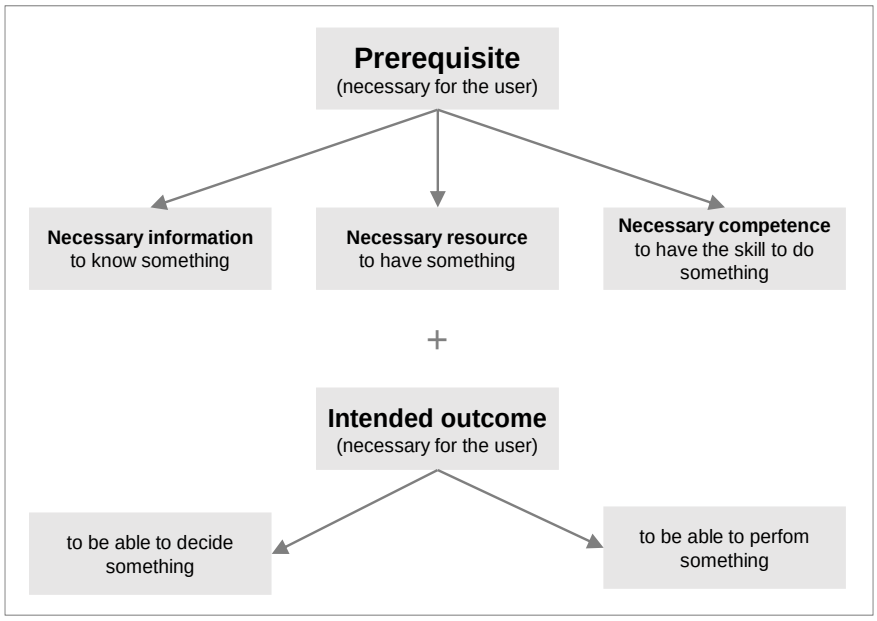

Figure 5 Types of user needs

Informational need

A user need that states the specific information needed to complete a task.

Informational needs typically lead to user requirements for interactive systems because the interactive system is able to provide that information to the user in the user interface. Informational needs are often implicit and only become apparent when analysing resource needs.

Examples:

- The patient needs to know when their scheduled appointment will in fact begin before arriving at the doctor's practice (information) in order to make good use of the time remaining (intended outcome).
- The prospective taxi passenger needs to know how much the trip is likely to cost before getting into the taxi (information) in order to decide whether to take the taxi or use a different form of transport (intended outcome).

Resource need

A user need that states the specific resource needed to complete a task.

Resource needs typically lead to organisational requirements to ensure that the resources are made available to the respective user group.

Resource needs help in the identification of informational needs, which in turn lead to user requirements for interactive systems.

Examples:

- The patient needs to have an *appointment* (resource) to ensure they receive treatment at a specific date and time (intended outcome).
- The prospective taxi passenger (user group) who needs a taxi at a moment's notice, needs to have a *taxi available in their vicinity* (resource) in order to reach their destination on time (intended outcome).

Competency need

A user need for a user group to possess a specific competency or skill

Competency needs typically lead to organisational requirements. The implementation of organisational requirements ensures that the required competency is available to the respective user group.

Examples:

- The baker needs to have the skill to produce a cheesecake (competency) to be able to sell it successfully (intended outcome).
- The ophthalmologist needs to have the skill of performing cataract surgery (competency) in order to achieve the intended treatment outcome for the patient (intended outcome).

4.1.2 Quality criteria of user needs

When identifying user needs, certain quality criteria must be met:

User needs

- must always be justified by the context of use.
- typically apply for all users of the same user group.
- are typically indisputable.
- always consist of a prerequisite that must be established, formulated as a condition, (either “know something” or “have something available” or “have skill to do something”), and an intended outcome enabled by the prerequisite (“be able to decide something” or “be able to do something”)

The interactive system that is yet to be developed or evaluated is not stated in the user need.

4.1.3 Rules for phrasing user needs

The following syntax rules help in the correct formulation of user needs.

Informational need:
The <user group> needs to know <information> in order to <decide> or <perform an activity>.

Examples:

- The patient (user group) needs to know when their scheduled appointment will in fact begin before arriving at the doctor’s practice (information) in order to make good use of the time remaining (intended outcome).
- The prospective taxi passenger needs to know how much the trip is likely to cost before getting into the taxi (information) in order to decide whether to take the taxi or use a different form of transport (intended outcome).

Resource need:
The <user group> needs to have <resource> in order to <decide> or <perform an activity>.

Examples:

- The patient (user group) needs to have a scheduled appointment (resource) to ensure they receive treatment at the agreed time (intended outcome).
- The prospective taxi passenger (user group) needs to have an available taxi within walking distance (resource), in order to reach their destination on time (intended outcome).

Competency need:
The <user group> needs to have <competency or skill> in order to <decide> or <perform an activity>.

Example:

- The doctor (user group) needs to have the skill to make the correct diagnosis (competency) in order to determine the right therapy (intended outcome).
- The taxi driver needs to have the skill to drive a taxi according to legal requirements (competency) in order to obtain a taxi licence (intended outcome).

The following examples do not follow the above phrasing rules or meet the quality criteria of user needs. The examples demonstrate frequently used phrases that are

- either too general (restating the task of the user)
- or point to a solution that does not consider the user need.

Example 1 of a poorly phrased user need:

The passenger needs to be able to buy a ticket from the ticket machine in order to avoid a penalty.

This example restates the user's task that the ticket machine needs to support and is therefore too general.

An example of how the above statement can be appropriately rephrased:

The passenger needs to know which train ticket is valid for a specific destination in order to use the train without getting into any trouble if their ticket is inspected.

Example 2 of a poorly phrased user need:

The car driver needs a way of measuring the pressure of each tyre in order to drive safely.

The example points to a solution rather than stating the information or resource needed.

An example of how the above statement can be appropriately rephrased:

The car driver needs to know if the pressure of each tire is too high or too low in order to adjust it.

4.1.4 Systematically identify user needs in context of use information

The following two-step procedure is suitable for identifying user needs within context of use information:

1. Identify each intended outcome in the context of use information.
2. Identify the prerequisites that are needed for each intended outcome.

Every piece of context of use information can contain one or more user needs. Therefore, it is recommended that context of use descriptions are processed sentence by sentence, and user needs are identified by applying the following guiding questions:

- What information is needed in the context of use to reach the intended outcome?
- What resource is needed in the context of use to reach the intended outcome?
- What competency or skill is needed in the context of use to reach the intended outcome?

Table 4 provides examples of context of use information and how user needs can be identified.

Table 4 Context of use information with identified user needs

Context of use information (Extract from a context of use description)	**Identified user needs** (examples)
Patients often have to wait for treatment beyond the agreed time of their appointment. Being forced to wait is very annoying for patients, especially when they have to sit in the waiting room for up to 90 minutes without any clear indication of how long it will be until it is their turn.	Example of an informational need: The patient (user group) needs to know when their scheduled appointment will in fact begin before arriving at the doctor's practice (information) in order to make good use of the time remaining (intended outcome).
Patients often make appointments well in advance, because good doctors are not available at short notice.	Example of a resource need: The patient (user group) needs to have an appointment (resource) to ensure they receive treatment at the agreed time (intended outcome).
General practitioners have a large variety of diseases to deal with. However, they make diagnoses at fast speeds, and patients rely on their prescribed treatments.	Example of a competency need: The doctor (user group) needs to have the skill to make the correct diagnosis (competency) in order to determine the right treatment path (intended outcome).

It is recommended that each contextual interview is documented as a separate as-is scenario. Many user needs re-occur across interviews and observations with users from the same user group.

- If, for example, the as-is scenarios from five contextual interviews from one user group are analysed, the overlap of identified user needs across documented contextual interviews can exceed 50%.
- Therefore, it makes sense to reference a previously documented user need when it comes up again.

The above approach can also be applied for summarised as-is scenarios across individual contextual interviews or observations. Larger sets of user needs should be sorted based on the tasks to be supported by the interactive system.

In order to be able to cross-reference and backtrack context of use information, it is recommended that unique identifiers are used for each user need.

Example for a user need with a unique identifier:

- UN7 – The doctor (user group) needs to make the correct diagnosis (skill) in order to determine the right treatment path (intended outcome).
- "UN" classifies this statement as a user need. "7" was the next consecutive number when documenting user needs.

Other naming conventions can be used based on their suitability.

Further user needs can be derived from organisational requirements, as long as they relate to users.

Example of an organisational requirement:
If a quote exceeds €100,000, the salesperson must ask the sales manager to approve the quote.

Example of a user need resulting from an organisational requirement for the user group “salesperson”:
The salesperson needs to know the quote value limit above which they must submit the quote to the sales manager for approval, to be in line with the rules of the organisation.

User wants are often formulated as a desired solution for an interactive system but they are not necessarily based on actual user needs from the context of use. When conducting contextual interviews, probing the user further can clarify whether a request is based on a user need from the context of use, or if it only reflects the individual preferences of the user and cannot be generalised for product design.

5 Derive and structure user requirements from user needs

5.1 Systematically transform user needs into user requirements

User needs are the single direct source of user requirements; each user requirement can be traced back to one or more user needs. Thus, user needs justify user requirements.

To derive qualitative user requirements from user needs, each user need must be analysed according to three guiding questions:

- What information will the user be able to recognise using the interactive system, to satisfy their need?
- What choices will the user be able to make with the help of the interactive system, to satisfy their need?
- What will the user be able to input using the interactive system, to satisfy their need?

It is essential that consistent phrasing is used when stating user requirements, to ensure that they are clear and can be communicated to others.

The minimum content that a qualitative and quantitative user requirement must contain is described in clause 5.1.1.

When deriving user requirements from user needs, conflicting user needs may arise; compromises must be reached to resolve such conflicts. These compromises can be detailed in user requirements or in solutions derived from user requirements.

Learning Objectives	
5.1.a	Know about and be able to apply quality criteria to user requirements
5.1.b	Be able to apply the syntax rule for a qualitative user requirement
5.1.c	Be able to derive qualitative user requirements from user needs
5.1.d	Be able to formulate quantitative user requirements
5.1.e	Be able to resolve conflicts between user needs from different sources by consensual user requirements and/or solutions

5.1.1 Quality criteria of user requirements

Quality criteria that apply to both qualitative and quantitative user requirements are as follows:

- User requirements must always be formulated with regard to the use of a solution rather than as an actual solution.
- User requirements must always refer to one or more user groups.

Specific quality criteria for qualitative user requirements are as follows:

- Qualitative user requirements must always be justified by user needs from the context of use.
- Qualitative user requirements must always consist of a requirement regarding the use of an interactive system (input, select, recognise), if necessary under certain conditions (and, if appropriate, with reference to a user need).

Quality criteria for quantitative user requirements are as follows:

- Quantitative user requirements must always be justified by user needs in the context of use or by stakeholder requirements.
- Quantitative user requirements must be suitable as acceptance criteria for the human-centred quality of the interactive system.

5.1.2 Transform user needs into one or more user requirements

Qualitative user requirements can be derived from any user need, by using the following central questions:

- What does the user need to be able to recognise in the interactive system, to satisfy their need?
- What does the user need to be able to select in the interactive system, to satisfy their need?
- What does the user need to be able to input to the interactive system, to satisfy their need?

Table 5 provides examples of how user needs can be transformed into user requirements.

Table 5 Context of use descriptions with identified user needs and derived user requirements

Context of use information (Extract from a context of use description)	Identified user needs (examples)	Derived user requirements
Patients often have to wait for treatment beyond the agreed time of their appointment. Being forced to wait is very annoying for patients, especially when they have to sit in the waiting room for up to 90 minutes without any clear indication of how long it will be until it is their turn.	Example of an informational need: UN1: The patient (user group) needs to know when their scheduled appointment will in fact begin (information) in order to make good use of the time remaining (intended outcome).	UR1: With the system, the user must be able to recognise that a scheduled appointment will be delayed. UR2: With the system, the user must be able to recognise for how long the scheduled appointment will be delayed.
Patients often make appointments well in advance, because good doctors are not available at short notice.	Example of a resource need: UN2: The patient (user group) needs to have an appointment (resource) to ensure they receive treatment at the agreed time (intended outcome).	UR1: With the system, the user must be able to recognise which dates and times are available for appointments. UR2: With the system, the user must be able to select a date and time for the appointment.

5.1.3 Rules for phrasing qualitative user requirements

The following syntax rule can be used for stating qualitative user requirements.

Syntax rule:

With the system, the user must be able to recognise / select / input / <information or a resource> under <optional condition>.

Examples:

- Dish washer: With the system, the user must be able to recognise that all dishes have been cleaned (information), before adding dirty dishes to the dishwasher (condition).
- Fridge: With the system, the user must be able to select the height of each storage compartment (resource) when placing a precooked dishn the fridge using only one hand (condition).

In order to be able to cross-reference and backtrack to user needs and context of use information, it is recommended that unique identifiers are used for each user requirement, as previously recommended for user needs.

Example of a user requirement with a unique identifier:

- UR1 - With the system, the user must be able to recognise what time their appointment is.
- “UR” classifies this statement as a user requirement. “1” was the next consecutive number for user requirements relating to that user need.
- See also Table 5. Other naming conventions can be used as appropriate.

5.1.4 Rules for phrasing quantitative user requirements

Quantitative user requirements can be formulated by taking the following components into consideration:

- The stakeholder who issued the requirement (for example, employer or legislator)
- The user group(s) who must fulfil the requirement during use
- The number or percentage of users who must fulfil the requirement during use
- The units of measurement (for example, task time, error rate, precision)
- Special condition(s) of the context of use

  Example: 95% of intranet users must be able to access the travel expense form from their regular place of work, within 10 seconds, regardless of the application currently running.

5.1.5 Resolve conflicting user needs

It might be difficult or impossible to satisfy different goals at the same time, because they are mutually exclusive.

Such conflicts can appear for user needs of the same user group.

Example:

- On returning home, the homeowner (user group) needs the room temperature to be sufficiently high (resource), in order to feel comfortable (intended outcome).
- The homeowner (user group) needs to have minimised power consumption (resource), in order to keep energy costs within available financial limits (intended outcome).

Conflicts in user needs can also appear for user needs across different user groups.

Example:

- The emergency room patient (user group) must receive immediate treatment (resource), without requiring an appointment, so that life-saving measures can be taken (intended outcome).
- The patient (user group) needs an appointment (resource) to ensure they receive treatment at the agreed time (intended outcome).

If there is no solution to conflicting user needs at the requirements stage, requirements should be formulated for all user needs, and a compromise within the solution space ,based on alternative solution drafts, should be sought. User requirements derived from identified user needs must be documented, even if the user needs cannot be satisfied simultaneously. It is necessary to evaluate the consequences of considering or ignoring a specific user requirement, in order to decide which of the seemingly conflicting user requirements should be further considered.

Example of an organisational solution for a previously mentioned user need:
Certain medical staff are reserved (organisationally) for treating emergency patients, which means that they are not available to attend scheduled appointments with patients. This ensures emergency room patients can be treated immediately.

5.2 Appropriately structure user requirements

Once user requirements have been derived from user needs, they are structured along tasks and subtasks. Tasks and subtasks are supported effectively and efficiently, based on the implementation of user requirements. The task model of the current context of use is likely to change, in light of derived user requirements. Some subtasks might be completed automatically by the system, whereas other subtasks might be new, based on the technology utilised. The task model for design is made of the adjusted structure of each task and its subtasks.

A set of user requirements, structured along tasks and subtasks, helps the project team to focus on the tasks of users that have to be supported by the interactive system, rather than on an unstructured list of features without a task focus.

Learning Objectives	
5.2.a	Be able to distinguish between a task model of the context of use and a task model for design
5.2.b	Be able to assign user requirements to tasks and subtasks that need to be supported by the interactive system

5.2.1 Develop the task model for design for each task to be supported

Identify tasks and subtasks to be supported, based on:

- the phases of the task lifecycle (plan, prepare, perform, evaluate result, communicate result), taking into account the task model of the current context of use, as explained in 1.3.3
- the derived user requirements.

Example: Task model for design

Task:	Clean the occupant's windows.
Contextual precondition(s):	The occupant perceives their windows to be so dirty that they dare not invite friends to visit.
Intended outcome(s) (Contextual postcondition(s))	The occupant perceives their windows to be clean.

Subtasks by phases of the task lifecycle:

Plan

- Decide when the windows are to be cleaned.
- Decide who should clean the windows.

Prepare:

- Fill the cleaning robot with water and detergent.
- Position the cleaning robot.

Perform

- Observe the cleaning process.

Evaluate result

- Check each window for the absence of dirt.
- Check each window for freedom from streaks.

Communicate the result

- Show the clean windows to the next person entering the house.

5.2.2 Structure user requirements according to tasks and subtasks to be supported

First, assign each user requirement to the task that will be supported by the implementation of the corresponding user requirement. Next, assign each user requirement to the specific subtask within the task that will be supported by the implementation of the user requirement.

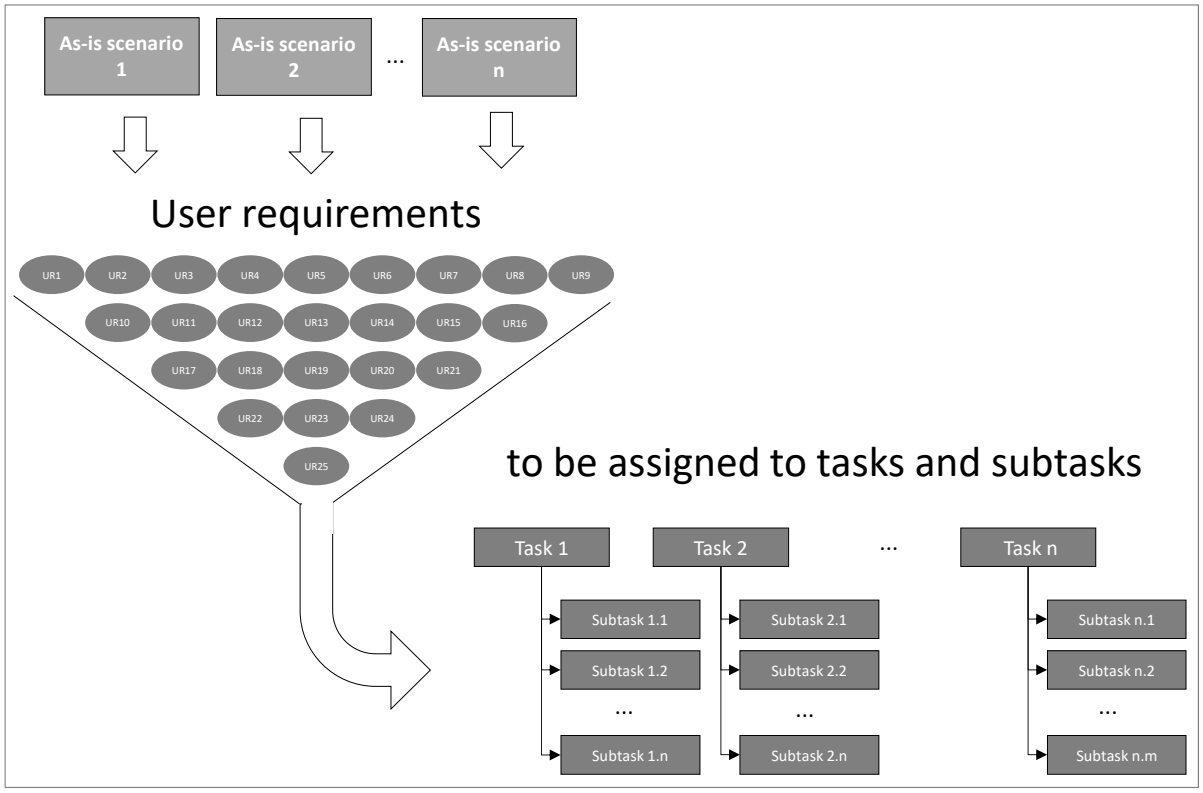

Figure 6 Structuring of user requirements according to subtasks

Example: Task model for design with user requirements assigned

Task	Clean the occupant's windows.
Contextual precondition(s)	The occupant perceives their windows to be so dirty that they dare not invite friends to visit.
Intended outcome(s) (Contextual postcondition(s))	The occupant perceives their windows to be clean.

Subtasks and user requirements:

1. Decide when the windows are to be cleaned.
   UR1: With the system, the user must be able to recognise that the windows need to be cleaned.
2. Decide who should clean the windows.
3. Fill the cleaning robot with water and detergent.
   UR2: With the system, the user must be able to recognise that water and detergent must be topped up.
4. Position the cleaning robot.
   UR3: With the system, the user must be able to recognise at which location to start the cleaning process.
   UR4: With the system, the user must be able to select the initial location for cleaning.
5. Observe the cleaning process.
   UR5: With the system, the user must be able to recognise the time left until the end of the cleaning process.
   UR6: With the system, the user must be able to recognise that the cleaning process has been completed.
6. Check each window for the absence of dirt.
7. Check each window for freedom from streaks.
8. Show the clean windows to the next person entering the house.

6 Consolidate user requirements

6.1 Consolidate and prioritise user requirements with users

After user requirements have been derived and structured into tasks and subtasks, they should be reviewed with users to determine whether or not each user requirement is adequate, complete and relevant for users.

Not every user requirement has the same relevance to users and decisions regarding implementation priority should be based on the relevance of each user requirement.

Structured schemes help users to rank the relevance of each user requirement systematically. One such popular scheme is the **Kano model**.

It is helpful to conduct workshops with users of the system, to ensure user requirements are adequate and complete, and to rank the relevance of each user requirement. The verified and ranked set of user requirements is the basis for prioritising the implementation of user requirements, over a specific time period.

Learning Objectives	
6.1.a	Know how to determine whether or not specified user requirements are adequate, complete and relevant, together with users
6.1.b	Understand the need to use a structured scheme for prioritisation
6.1.c	Know about publicly available schemes for prioritisation

6.1.1 Correctness, completeness and relevance of user requirements

Consolidation workshops with users from different user groups should be conducted to analyse the correctness, completeness and relevance of specified user requirements, from users' perspective.

6.1.2 Using schemes to rank relevance of user requirements

Structured schemes can be used to rank the relevance of user requirements. Although structured schemes can be developed by UX professionals in the project team, the use of publicly available schemes, such as the Kano scheme, is recommended, as they are well-established and are often likely to achieve consensus.

Kano scheme

A classification model for prioritising qualitative user requirements or functions of an interactive system (that might be derived from these requirements), from the users' point of view.

In a Kano scheme, potential users categorise requirements or features, using the following categories:

1. That would be nice
2. That is something I require
3. That makes no difference to me
4. That is barely acceptable.
5. That would bother me very much.

For each user requirement, two questions are asked. First, the person is asked how they would feel if the requirement was implemented. Then they are asked how they would feel if

the requirement was not implemented.

The combination of both answers leads to one of the following requirement classifications:

- Must-be quality
  (these requirements are taken for granted by the users)
- One-dimensional quality
  (these requirements result in satisfaction when fulfilled and dissatisfaction when unfulfilled)
- Attractive quality
  (these requirements provide satisfaction when achieved fully, but do not cause dissatisfaction when unfulfilled)
- Indifferent quality
  (these requirements have no relevance to the users)
- Reverse quality
  (the implementation of these requirements results in dissatisfaction)

User requirement to be prioritised:
Before cooking eggs with the egg boiler, the user must be able to select how many eggs are to become hard-boiled, how many medium-boiled, and how many soft-boiled.

What would you say, if we would implement this?
☑ That would be very nice
☐ That is something I require
☐ I don't care about that
☐ That is barely acceptable
☐ That would bother me very much

What would you say, if we would not implement this?
☐ That would be very nice
☐ That is something I require
☐ I don't care about that
☐ That is barely acceptable
☑ That would bother me very much.

if we would not implement this: / *if we would implement this:*	That would be very nice	That is something I require	I don't care about that	That is barely acceptable	That would bother me very much
That would be very nice	-/-	Attractive quality	Attractive quality	Attractive quality	One-dimensional quality
That is something I require	Reverse quality	Indifferent quality	Indifferent quality	Indifferent quality	Must-be quality
I don't care about that	Reverse quality	Indifferent quality	Indifferent quality	Indifferent quality	Must-be quality
That is barely acceptable	Reverse quality	Indifferent quality	Indifferent quality	Indifferent quality	Must-be quality
That would bother me very much	Reverse quality	Reverse quality	Reverse quality	Reverse quality	-/-

Figure 7 Kano Scheme

The Kano scheme can be used in consolidation workshops with users. A typical procedure is as follows:

Plan

- Recruit between 5 and 15 representative users per user group
  (participants from previous contextual interviews and/or observations are good candidates, as they tend to be highly motivated to understand which requirements were derived from interviews and observations).
- Invite just one user group to each workshop, otherwise valid disagreements between user groups may interrupt the workshop.

Introduce

Explain that

- tasks were identified from contextual interviews and/or observations, and that they are used as the basis for the design of the interactive system.
- tasks have been divided into subtasks.
- for each subtask, what a user must be able to do with the interactive system – recognise, select, input – has been identified.
- the participants will now be asked to identify any incompleteness or inconsistencies in the constructed information.

Execute

- Read out (and explain, if necessary) each task and its subtasks.
- After communicating a task and its subtasks, collect feedback from the participants (Is anything unclear? Is anything missing? Is anything wrong?)
- For each task, read out (and explain, if necessary) the user requirements, using the subtasks as a guide.
- If anything is unclear to participants about a user requirement, encourage them to ask questions.
- Introduce the prioritisation scheme that will be used (for example, the Kano scheme); explain how the scheme is to be applied.
- After introducing all the user requirements for one task, let participants prioritise them. A good way of doing this is to have each participant hold up a card with a number for each category. The moderator or a note-taker counts all votes for each category of the scheme.
- Repeat the previous step for each task that is to be supported by the interactive system.

Close

- Explain that a statistical analysis of the results will not be performed as the number of participants is typically too small.
- If, for example, a requirement falls into multiple categories, then the votes are recorded for each category.
- Explain that prioritisation helps with decision-making about the timeline for implementation, in consultation with the project manager.

6.2 Determine implementation priority for user requirements together with project members

Once the set of user requirements for the interactive system has been verified with users, the project team must decide on the implementation priority of each user requirement. Besides cost estimates, the importance of each user requirement for users must be taken into account. It is critical to communicate the set of user requirements to stakeholders and project team members, to highlight their importance to users in terms of effectiveness, efficiency and satisfaction. User requirements that cannot be implemented immediately should be placed on a **product roadmap,** which specifies what will be implemented across releases at defined time periods.

Ensure that user requirements are prioritised by project members, in relation to when they should be implemented. Typically, not all user requirements can be implemented at the same time, and **prioritisation** can help avoid implementation conflicts.

Learning Objectives	
6.2.a	Be able to communicate user requirements to project stakeholders.
6.2.b	Be able to convey the importance of each user requirement to project stakeholders
6.2.c	Know that user requirements should be documented in a robust and retrievable way, for example by using a product roadmap
6.2.d	Be able to reach consensus about implementation priority for each user requirement together with stakeholders of the project

6.2.1 Implementation of user requirements to achieve human-centred quality

It is the responsibility of those who derive user requirements, to explain to stakeholders, how important the implementation of user requirements is for the human-centred quality of an interactive system. Comprehensive communication is vital in convincing stakeholders of the importance of implementing user requirements. It is also important to communicate the consequences of not implementing high-priority user requirements.

6.2.2 Prioritisation of requirements

Often, not all user requirements can be implemented at the same time, and some may not be implemented at all, due to the cost of implementation. Therefore, requirements must be prioritised.

Prioritisation

An activity which involves determining if and when a requirement is implemented.

Initial prioritisation should always be carried out with respect to human-centred quality objectives (that is, how relevant is the implementation of requirements with respect to users?), before considering other stakeholders. The Kano scheme is well-suited for prioritisation with respect to users.

A typical prioritisation scheme for implementing requirements is as follows:

- Priority 1: to be implemented in the current product version
- Priority 2: to be implemented in the next product version
- Priority 3: for future consideration
- Priority 4: out of scope (possibly in scope for other/new systems)

Requirements that are not going to be implemented in the current product version must be included in the product roadmap.

Product roadmap

A representation of how a product is supposed to develop across releases over defined periods of time, regarding its functionality.

Product roadmaps are used for communication between the disciplines of product management, requirements engineering and systems engineering, and management. This helps to build a common understanding of the long-term product plan and its associated activities.

Appendix 1: Model seminar

Concept of the seminar

- Participants must thoroughly familiarise themselves with the curriculum and the terms defined therein, the illustrative examples, and the example questions for the theoretical and practical exams, before the start of the seminar. Learning the terms and concepts – which is necessary for being able to pass the test – is not the main topic of the seminar. (Proficiency level K1 “Know”)
- The seminar mainly focuses on conveying the relationships between terms and concepts with respect to the learning objectives. (Proficiency level K2 “Understand”)
- In the seminar, participants practise the methods of user requirements engineering. (Proficiency level K3 “Apply in practice”)
- The practical exercises for learning units 4.1, 5.1 and 5.2 follow the same format as the practical exam, helping participants familiarise themselves with the format of the practical exam.

Timetable for the seminar

Day	Time	Learning unit number and title	Format
1	10:00-10:45	1.1 Differentiate between requirements and solutions	Interactive lecture (45 min.)
1	10:45-11:30	1.2 Differentiate between stakeholder requirements and system requirements	Interactive lecture (45 min.)
1	11:30-11:45	Break	
1	11:45-12:30	1.3 The context of use as the basis for user requirements	Interactive lecture (45 min.)
1	12:30-13:15	1.4 User requirements as a separate requirements category within the stakeholder requirements	Interactive lecture (45 min.)
1	13:15-14:15	Break	
1	14:15-15:00	1.4 User requirements as a separate requirements category within the stakeholder requirements	Exercise for 1.4 (45 min.)
1	15:00-15:30	2.1 Identify the rationale and goals for the context of use analysis	Interactive lecture (30 min.)
1	15:30-15:45	Break	
1	15:45-16:45	2.2 Determine the approach for context of use analysis	Interactive lecture (60 min.)
1	16:45-17:30	3.1 Select and recruit users for the gathering of context of use information	Interactive lecture (45 min.)
	End of day 1		
2	09:00-09:45	3.2 Prepare and gather context of use information	Interactive lecture (45 min.)
2	09:45-10:15	Develop questions for contextual interviews	Exercise 1 for 3.2 (30 min)

Day	Time	Learning unit number and title	Format
2	10:15-11:15	Perform a contextual interview	Exercise 2 for 3.2 (60 min)
2	11:15-11:30	Break	
2	11:30-12:15	3.3 Create context of use descriptions based on context of use information	Interactive lecture (45 min.)
2	12:15-12:35	4.1 Systematically identify and formulate user needs	Interactive lecture (20 min.)
2	12:35-13:00	5.1 Systematically transform user needs into user requirements	Interactive lecture (25 min.)
2	13:00-14:00	Break	
2	14:00-15:30	Systematically identify and formulate user needs, and systematically transform user needs into user requirements	Group exercise for 4.1 and 5.1 (90 min.)
2	15:30-15:45	Break	
2	15:45-16:15	Document a contextual interview in the form of an as-is scenario	Exercise 1 for 4.1
2	16:15-17:15	Systematically identify and formulate user needs	Exercise 2 for 4.1
	End of day 2		
	09:00-10:30	Systematically transform user needs into user requirements	Exercise for 5.1 (90 min.)
3	10:30-10:45	Break	
3	10:45-11:30	5.2 Appropriately structure user requirements	Interactive lecture (45 min.)
3	11:30-12:15	Appropriately structure user requirements	Exercise for 5.2 (45 min.)
3	12:15-13:15	Break	
3	13:15-14:00	6.1 Consolidate and prioritise user requirements with users	Interactive lecture (45 min.)
3	14:00-14:45	6.2 Determine implementation priority for user requirements together with project members	Interactive lecture (45 min.)
3	14:45-15:00	Break	
3	15:00-17:00	Preparation for the CPUX-UR exam	Exercise (120 min.)
3	End of day 3	-/-	

Appendix 2: Important changes to this document

Version	Changes
14 March 2023, version 3.2.2	Editorial mistakes corrected: • Figure 1 (English version only): German text replaced by English text. • Clause1.4.2: In bulleted list “Domain-specific requirements” added underneath “Organisational requirements” • Clause 3.2.3 and 3.2.4: In bulleted list underneath “Perform”, first bullet item “Introduce yourself as <interviewer / observer> and note-taker” added to be consistent with clause 3.2.5 • Page 70 (German version only): “Organisatorisches Erfordernis” replaced by “Organisatorische Anforderung”
21 January 2023, version 3.1.2	• Figure 1 User Requirements Engineering (enlarged boxes) within the Human-centred Design Process updated
20 December 2022, version 3.1.1	• On page 7, invalid reference to illustrative examples of work products removed
3 December 2022, version 3.1	• Definitions updated for consistency with new CPUX-F curriculum, version 4.01
31 July 2022, version 3.0	• Summaries added at the beginning of each learning unit • Definitions are included in the chapters, and not at the end of this document anymore • Editorial corrections • Index added
23 December 2021, version 2.0	• Chapter 7 "Cooperation between user requirements engineers and other roles / disciplines" removed • Redundant learning objectives removed • Document comprehensively editorially revised
7 June 2016, version 1.3	• 1st version

Appendix 3: References & Index

Beyer, H. and Holtzblatt, K. (1998). Contextual Design. Defining Customer-Centered Systems (Morgan Kaufmann Series in Interactive Technologies). San Francisco: Morgan Kaufmann.

Brown, T. and Katz, B. (2009). Change by design: How Design Thinking transforms organizations and inspires innovation. New York: HarperCollins.

Constantine, L. (2009). Model-driven Inquiry - A Streamlined Approach to Data Collection. User Experience, Volume 8, Issue 3, 3rd Quarter 2009.

Cooper, A. and Reimann, R. and Cronin, D. and Noessel, C. (2014). About Face: The Essentials of Interaction Design. 4th Edition. New York: John Wiley & Sons, 2014.

Deutsche Akkreditierungsstelle GmbH (DAkkS) (2010). Leitfaden Usability. [online] Available at https://www.dakks.de/files/Dokumentensuche/Dateien/71%20SD%202%20007%20A1_Leitfaden%20Usability%201.3_zur%C3%BCckgezogen%20am%2020.01.2020.pdf

Geis, T. and Dzida, W. and Redtenbacher, W. (2004). Specifying usability requirements and test criteria for interactive systems. Germany: Federal Institute for Occupational Safety and Health.

Geis, T. and Polkehn, K. (2018). Praxiswissen User Requirements. dpunkt Verlag

Dzida, W. and Freitag, R. (1998). Making use of scenarios for validating analysis and design. IEEE Transactions on Software Engineering, Volume 24, Issue 12.

Gothelf, J. and Seiden, J. (2012). Lean UX: Applying Lean Principles to Improve User Experience. O'Reilly and Associates.

Hackos, J. and Redish, J. (1998). User and Task Analysis for Interface Design. Chichester: Wiley.

Hartson, R. and Pyla, P. (2012). The UX Book: Process and guidelines for ensuring a quality user experience. Morgan Kaufmann.

International Organization for Standardization (ISO) (2010) ISO 9241-210: 2010. Ergonomics of human-system interaction - Human-centred design for interactive systems. International Organization for Standardization (ISO).

International Organization for Standardization (ISO) (2011) ISO/IEC 25010:2011. System and software quality models. International Organization for Standardization (ISO).

International Organization for Standardization (ISO) (2014) ISO/IEC 25063:2014. Common Industry Format (CIF) for usability: Context of use description. International Organization for Standardization (ISO).

International Organization for Standardization (ISO) (2013) ISO/IEC 25064:2013. Common Industry Format (CIF) for usability: User needs report. International Organization for Standardization (ISO).

International Organization for Standardization (ISO) (2013) ISO 25065:2019. Common Industry Format (CIF) for usability: User requirements specification. International Organization for Standardization (ISO).

Riedemann, C. and Freitag, R. (2009). Modeling usage: Techniques and Tools. IEEE Transactions on Software Engineering, Volume 26, Issue 2.

Rosson, M. and Carroll, J. M. (2002). Usability Engineering: Scenario-Based Development of Human-Computer Interaction (Morgan Kaufmann Series in Interactive Technologies).

Van der Veer, G. C. (1994). Mental Models of Computer Systems: Visual Languages in the Mind. In: Tauber, M. and Mahling, D. and Arefi, F., ed., Cognitive Aspects of Visual Languages and Visual Interfaces. Amsterdam London New York Tokyo: North-Holland, pp. 3 ff.

Young, I. (2008). Mental Models. Aligning Design Strategy with Human Behavior. New York: Rosenfeld Media.

Index