A Requirement Engineering Approach to User-Centered Design

Bin Xiao, Kai Su*, Xiaoguang Su
Department of Management Engineering and Equipment Economics, Naval University of Engineering, Wuhan, Hubei, 430033, China
*Corresponding author’s e-mail: keppelsue@163.com.

Abstract. This paper introduces a user-centered design method in software engineering practice. This method can develop software more effectively based on the design results of UCD, and make these results be tracked and tested in the future. Another feature of this method is to make UCD as schedulable and manageable as other software development activities. This paper introduces a requirement analysis framework based on three types of usability requirements, which reflect the results of UCD activities in software development. These types of requirements come from the description of UCD in ISO 13407 model.

1. Introduction
Usability is an important attribute of a software product. It means the degree of effectiveness, efficiency and user satisfaction that a specific user can achieve when using a software product in a specific environment. Many usability engineering (UE) and test methods are based on these three standards. In software design, how to associate the above three types of usability elements, how to express them in the UE process, and how to evaluate them are the problems we need to solve.

This paper designs a requirements engineering (RE) framework, which introduces three types of usability related requirements: Usability, Workflow and Design. This UE process can be applied to more complex software engineering (SE) processes. Its goal is to unify the framework with usability engineering practices and existing re methods in software engineering disciplines.

1.1. Requirement Engineering
In most cases, the functional and non-functional elements of the software system are specified by the software requirements specification (SRS). The functional requirements specify all the functional services that the system should provide to users, and what the system should do to the user's operation. Non-functional requirements define the constraints on the services and functions provided by the system. In most cases, the product quality of the software system is reflected by non-functional requirements. Usability is one typical non-functional requirements.

In addition, correctness, clarity, integrity, consistency and verifiability are also important quality attributes. Furthermore, each requirement described in the SRS is appropriately related to these attributes. When the requirements provided by SRS for system development are verifiable and testable, it can bring great benefits to software system development. However, it is very difficult to get testable and precise requirements. For example, architecture requirements in system requirements are difficult to be verified and tested in the design phase.
In addition, in order to ensure the availability related requirements can be verified and tested in the whole process of software system development, it is necessary to introduce quality attributes. The Usability community has achieved some results to provide guidance for the development of Usability related requirements [1].

1.2. User-Centered Design Process

Some scholars have been carrying out research on user-centered design (UCD) (e.g. [2], [3]). After analysis, we found that these studies are based on the user-centered design process. The user-centered design process is a supplement to the existing design method, which can be applied in the system design process.

The process consists of 4 base activities:
• To specify the context of use,
• To summarize the user requirements,
• To generate design specifications and
• To evaluate the designs according to the requirements.

During the development process, the above four activities are carried out in an iterative manner. Until the system meets the requirement specification, the process will be finished. Another feature of this process model is that it needs to iterate and test the requirements during the user-centered design process.

In addition, the requirements model and prototype need to be evaluated throughout the design process. At the same time as iterative improvement of system design, the requirements need to be further refined and improved. In the UCD process, not only new requirements will be generated, but also more specific requirements will be needed because the system design becomes more specific and precise.

After the completion of UCD, its results shall be the input of the next design phase and provide verification criteria for it.

1.3. Evaluation

As mentioned earlier, UCD is an iterative process model. In the iteration cycle of the design process, all kinds of evaluation are needed. For example, the intermediate results of the design should be evaluated at an early stage to identify and improve design defects as early as possible. Based on this feature, it is recommended to apply assessment activities to the development cycle, rather than just at the end of the development process. The application of evaluation in the whole process of system development life cycle can effectively ensure that the final product can meet the expectations of users, software designers, and developers as well as user units.

Evaluation activities are mainly aimed at two types of objectives, each of which has its own evaluation and verification methods. First of all, we should check and evaluate the form and content of the software documents formed in the development stage, and obtain the key information related to the subsequent process. The examination and evaluation of documents can take the form of interviews with domain experts, users and managers. Furthermore, it is necessary to review the documents to determine their consistency with the previous work.

Next, the intermediate results of the system development should be evaluated to determine whether it meets the requirements of the requirements specification. For UCD evaluation work, this work can be implemented in the way of expert evaluation or usability verification. It should be noted that the accuracy and measurability of requirements are the premises of evaluation. These two process evaluation activities have their own characteristics, so it is necessary to take corresponding evaluation methods.

The results of the use phase can be validated through usability testing or expert evaluation review reviews, such as personas or other models that describe work, context, and focus users. A sampling survey is another kind of verification method which can get the feedback of users on the system development process. When evaluating the workflow described in the requirements, the suggested
methods include: comparing the workflow actually used by users with the work in the design documents, listening to users' feedback, and responding to the deviation in time.

The key point of evaluation work is: evaluation work is a process throughout the development cycle, which needs to be implemented in all stages of development. Clear and measurable criteria are the premise of effective evaluation in all stages of system development. In conclusion, in order to evaluate UCD results accurately and effectively, corresponding requirements should be generated in the three stages of the software development process mentioned above.

2. A User-Centered Requirement Approach

According to the previous analysis, we propose a new framework based on usability, workflow, and design. In the use stage, the analysis mainly focuses on the expected users, their work and tasks, their psychological model, the concept of system use, physical environment. From these analysis results, we further summarize and extract the user requirements for system availability, including effectiveness, efficiency and stability. These requirements can be used to evaluate the intermediate results of system development, and can also be used as evaluation criteria of UCD results in other stages.

The user requirements phase mainly deals with the task processing or workflow that interacts with the user. In the context of using data in the system, we can get specific task performance model from users, optimize workflow, and generate an improved task performance model. Through this stage of processing, we can obtain the system interaction requirements for users in the work scenario of task processing or workflow execution. These requirements have the characteristics of effectiveness, efficiency, and availability, which can be verified by evaluation. Typical proofs include comparing the optimization state of the workflow. At the same time, these requirements also serve as the input of the follow-up system development process and provide verification criteria for process evaluation. Interface layout, human-computer interaction, information structure, and other system attributes are defined in the product design solution stage. Some of these system attributes belong to the underlying model of the system, which can not be verified directly in the form of user experience, but some of them can be verified.

Therefore, when translating these attributes designs into solutions, the following requirements need to be considered:

• Conceptual/Structural/Framework type requirements.
• Visual requirements
• Interaction requirements

The usability requirements generated in the context of the usage phase are the main basis for user interface (UI) requirements assessment. The main evaluation contents include the consistency of interface layout, human-computer interaction design and user demand model, the effectiveness of function, system operation efficiency and function availability. In addition, we need to evaluate the consistency of workflow and system requirements, including the correctness, usability and ease of use of workflow. Figure 1 depicts a user-related requirements framework around which all user-related requirements processes and validations are conducted.

![Usability-related requirements Framework](image-url)
It should be noted that from the perspective of complete system design, these three types of availability related requirements do not cover all system descriptions. However, in order to get the correct system development results, it is necessary to iteratively conduct user requirements specification and evaluation in the system development process.

2.1. Usability Requirements
These requirements include the system characteristics and quality requirements criteria that the system development results must meet. It is based on three usability dimensions: effectiveness, efficiency and user satisfaction. Usability requirements are based on the context of user functions, business improvement requirements and professional specifications in specific fields. Cooper [2] proposed that the source of usability requirements can be system context model and user characteristics description, which explained the task prototype and user requirements necessary for system development.

Although the availability requirements belong to the non-functional requirements of the system, most of the non-functional requirements will have a greater impact on the functional requirements because they will involve the facts of system usage and user interaction.

Ways to verify usability include interviews and reviews. At the end of the system development cycle, the validity can be verified by usability test, expert review or survey, and questionnaire. When the testing department conducts requirements testing, it can combine usability requirements with other non-functional requirements.

2.2. Workflow Requirements
In the Specify Requirement phase of system development, in order to accurately reflect the changes and improvements in the design and development process, it is necessary to continuously identify the current system status according to user scenarios, and continuously optimize the workflow and system requirements. Once new user requirements or related workflows are identified, new workflow descriptions are needed to document these requirements. These requirements show how the system can fulfill the user's tasks or work objectives. The workflow requirements encapsulate the information processed in the workflow and the work steps of the workflow to illustrate the functional behavior of the system, without caring about the specific technical details of its implementation. The main focus of workflow requirements is to implement an improved workflow based on user scenarios and ensure that users can complete their work correctly.

In workflow requirements, use case or scenario description and other similar forms are usually used to describe the interaction and information flow in user scenarios. The role of the scenario is to describe the specification of the system interaction process. A use case is a general scenario that describes the interaction with a specific user interface. The use case shows the way to realize the interaction between the user and the function in the system. In the use case description, the requirements of all users involved in the interaction in the use case should be considered at the same time to ensure that the interests of all users are met. Although use cases cannot cover all requirements, they are the basis of system requirements [4].

In order to evaluate whether the description of the workflow is accurate and whether the workflow does improve the actual task, developers can review and verify the workflow requirements together with users. When the system development is in the later stage, the requirements can be used to verify whether the system interaction mode meets the workflow requirements, and evaluate the workflow implementation of the relevant prototype or system. For example, you can use the main scenario in a use case as an evaluation scenario in a usability test.

The description of system function and the requirement of workflow help to clarify the content of system development. We can organize tests based on workflow description in workflow requirements to verify the validity of system prototype and final product established in the design solution stage.

2.3. UI Requirements
Availability requirements and workflow requirements are combined into a set of conceptual models and Solution elements during the product design solution phase. These models and solutions usually describe the logical model and physical properties of the system. For example, the appearance, behavior, and interaction of widgets, UI elements, and interactive interfaces. Generally, they also provide instructions for the status prompt system response mode and preset behavior displayed by the system.

UI requirements are generated based on user feedback on the system prototype, intermediate products in the current development stage, UI specifications, system style descriptions and other models.

UI requirements’ concerns are distributed at different levels of interaction between the system and users, including the following types:

- Definition of the overall logic structure of the user interface.
- System interface display and action response requirements.
- User system interaction requirements.
- Interaction requirements across multiple user interface elements.
- Requirements of message responding to user behavior.

In order to ensure that the implementation of the system conforms to the analysis results in the early stage, it is necessary to evaluate the user interface requirements based on the availability requirements and workflow requirements. The purpose of the assessment is to ensure that the user interface requirements and the appearance and behavior response of the system meet the user's expectations. At the same time, the UI requirements are also used to evaluate whether the solution implemented by the coding system conforms to the design of the solution.

In the development process, developers can use UI requirements as an input of system design, and test departments can also use UI requirements to ensure the proper implementation of UI. Developers can choose the most appropriate implementation of a given workflow to enhance or extend workflow requirements.

Domain experts also use UI requirements to verify that the system conforms to its development specifications. They have practical value in supporting system specification and verification. In addition, for the end-user, the UI requirements are transparent, that is, the user does not need to care about the specific and individual UI features in the system, but only needs to confirm that the system meets the overall experience expectations of the system.

3. Summaries and Outlook

We propose a usability based user center requirements framework, which combines requirements engineering practice with usability engineering. Different from the common software development process model, this method puts forward the requirement analysis and evaluation model based on three types of requirements. Different from the common software development process model, this method proposes three requirements analysis and evaluation models based on usability, workflow and user interface. The framework extends the concept of usability requirements by introducing workflow and user interface requirements. Based on the requirements of workflow and user interface, the requirements of UCD can be described more accurately in the later stage of system development. This paper describes the three types of requirements, describes the relationship between them, and gives their application and reuse details.

By describing these three types of requirements separately, we can ensure that the user-centered requirements have an iterative and traceable hierarchy. In addition, this method advocates selecting the best verification and evaluation method based on different types of requirements. For example, usability testing is used to verify usability requirements, and expert evaluation is used to evaluate user interface requirements. The research of Freyman et al. provides guidance on how to choose an appropriate evaluation method [5]. According to these research results, developers can choose appropriate evaluation methods in the system development life cycle according to the types of requirements.
Another advantage of the requirements framework is that it provides better flexibility. Based on this framework, different types of user requirements can be obtained more effectively, so as to better meet user satisfaction with the system, and effectively reduce the cost of system testing and verification.

The approach and requirements framework proposed in this paper also need to be tested in the actual development and application. The follow-up work includes the means of integrating the new requirement type into the existing requirement engineering method, the evaluation and implementation method of the new requirement type, and the further improvement and refinement of the required framework.

References
[1] CISU-R. (2014)Common Industry Specification for Usability-Requirements, National Institute of Standards and Technology.
[2] Cooper and R. Riemann. (2016) User Interface Design. Wiley, Hoboken.
[3] D. J. Mayhew. (2015)The Usability Engineering Lifecycle – A practitioner’s Handbook to User Interface Design. Morgan Kaufmann, Burlington.
[4] Cockburn. (2016)Writing Effective Use Cases. Addison-Wesley, Boston.
[5] M. Freymann, K. Nebe. (2017)Selecting Appropriate Evaluation Methods for Different UCD Outcomes. In: Human-Computer Interaction International Conference 2007, Beijing. pp.102-107.