Two Gaps instead of One. The Interplay between User Experience Engineering and Interaction Design

Timo Jokela
Helsinki University
timo.jokela@helsinki.fi

ABSTRACT

The paper identifies three main phases of designing user interfaces: user experience (UX) engineering, interaction design and software development. The paper focuses on the interplay between UX engineering and interaction design. It is argued that the gap between these two phases in inherently challenging. To overcome the gap, designers' knowledge on the results of UX activities and general UX guidance is critical. Incentives on good UX can be one means for good results.

Keywords

user experience, interaction design, software design

INTRODUCTION

The workshop theme is the challenge in making the interplay between user experience and software design successful.

This paper argues that there actually are three main phases, instead of two: user experience (UX) engineering[1], *interaction design*, and software design.

Interaction design produces the visual and logical design of the system from the user's point of view (user interface), i.e. the design of:

- individual user interface elements, such as dialogs, menus, tables, etc.
- the structure of the user interface, such as the identification, definition and design of the windows
- visual appearance
- also supporting material such as user manuals

Software development is used to implement interaction design. In other words, interaction design, as such, is not software development.

Thereby, this paper argues that main phases of designing interaction solutions are:

1. UX engineering: provides user driven data to guide interaction design
2. Interaction design: driven by UX engineering activities and UX design guidelines and standards, one creates interaction design solutions (ISO/IEC 2010)
3. Software design: interaction solutions are implemented through software.

Thereby, there are two separate challenges of interplay:

- the interplay between UX and interaction design
- the interplay between interaction design and software design

These different kinds of phases are illustrated in the 'JFunnel' lifecycle model: UX guided interaction design, Figure 1. In the figure, the three activities are marked with different colors:

- orange: UX activities
- light grey: interaction design
- dark grey: software implementation

In this paper, we focus on the interplay challenge between the two first sets of activities: UX engineering (orange) and interaction design (light grey).

UX ENGINEERING AND INTERACTION DESIGN

The distinction between UX activities and interaction design is made because the two kinds of activities have a fundamental difference:

- UX activities produce various kinds of user data, such as user descriptions, UX requirements, and results from evaluations (but does not produce interaction design solutions)
- Interaction design produces the actual user interaction design solutions: GUI designs, user documentation, and so on.

In other words, the role of UX activities is to provide user data to the interaction design activity; the role of the interaction design activity is to transform this data into design solutions that represent good UX. This is illustrated also with the arrows from other activities to the interaction design activity (Figure 1).

This separation is also useful because those who carry out UX activities (UX specialists) are often not the same people as those who produce design solutions (interaction designers). Moreover, the model illustrates that it is not

[1] The term UX engineering is analogous with the term 'usability engineering'. It covers not only UX evaluation but also understanding users, their goals, tasks and environments of use, and UX requirements determination.

enough to introduce UX activities - UX activities have no value unless the results are taken into account in the interaction design activity.

The UX Input for Interaction Design

The UX activities can be categorized into two groups:

- UX *requirement* activities that provide data based on analysis of users without any prototype or model of the system to be developed (activities 1…4)
- UX *evaluation* activities that provide feedback on a prototype or a model of the system under development (activities 6,7)

The UX requirement activities (1…4) provide data such as

- user group descriptions
- user goals, tasks and work descriptions
- UX targets: strategic targets are at business level; operational targets are at user level; UX measures

The UX evalutation activities (6,7) provide

- qualitative findings about which design solutions work and which solutions do not work from the viewpoint of UX
- results to which extent the UX requirements are met

In addition, the generic UX guidelines provide principles and design guidance at generic level. Examples of such guidelines are ones from the ISO 9241 series [1].

These inputs can be categorized as follows:

- UX activities 1-4 provide input for how to design systems to support users' tasks and work
- Generic guidelines provide input for how to design details of interactions
- UX activities 6 and 7 provide feedback to both of these levels of design: how the user interface support users' work; and how well the details are designed

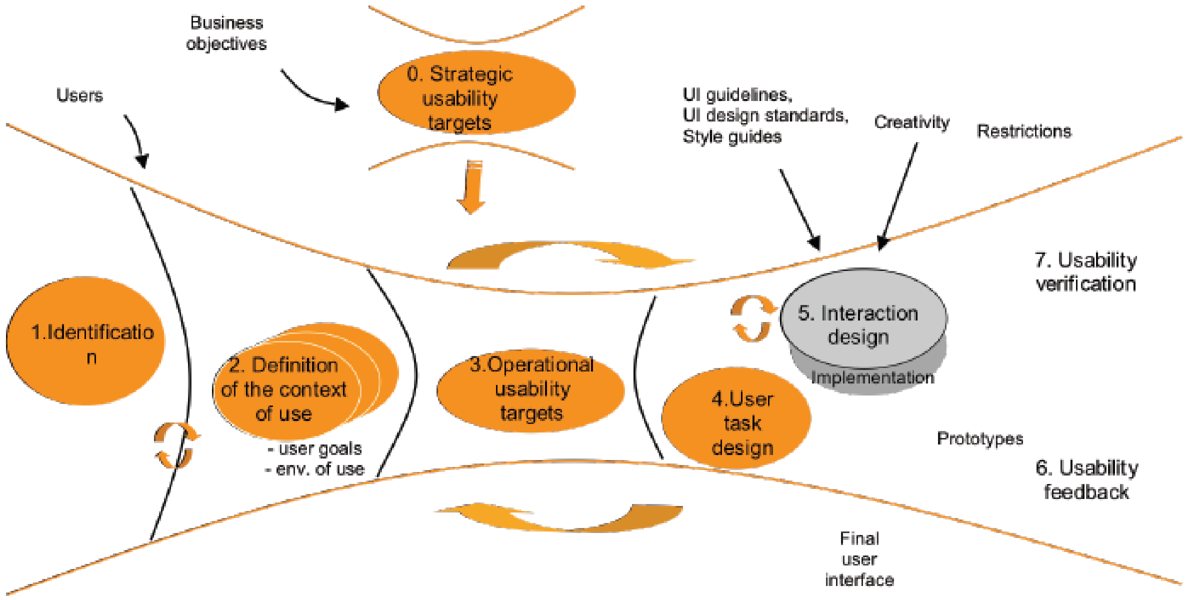

Figure 1. The JFunnel user experience life-cycle model [2]. The different types of activities are marked with different colors: UX engineering activities are orange, interaction design light grey, and software design (implementation) dark grey.

The Challenge between UX and Interaction Design

One can see that there is an obvious challenge in the interplay between UX activities 0…4 and interaction design: user definitions, user goals and task descriptions, and UX targets do not provide concrete guidance for interaction design. They provide only requirements, and it is the task of designers to transform this data into UX designs solutions.

For example, consider a UX requirement "The system should be learnable without user training". This kind of requirement represents a gap between UX and interaction design. It is a requirement for interaction design but does not provide any guidance for how to achieve it. It is a design challenge to produce design solutions that would meet the requirement.

UX evaluation activities 6, 7 provide clear results what works and what does not work. But the results do not as such tell what a better interaction design solution is.

UX guidelines provide more concrete guidance, but still they are more requirements than solutions. For example, "speak user's language" does not say what the understandable terms of a specific system are. One concrete category of UX guidelines, however, exists: interaction design patterns (e.g. [5]); although focus currently more on usability than UX).

The Interplay between UX and Interaction Design

The conclusion is that the interplay between UX and interaction design is an inherent challenge. The extent to which interaction designs truly represent good UX depends very much on the interaction designers: to which extent they are willing and able to transform the input of the UX activities and generic UX guidelines into interaction design solutions.

Thereby designers' knowledge on factors affecting UX is extremely important: results from the different UX activities and contents of general UX guidelines. One solution that has worked in designing good usability [3] – give incentives to designers – may also have a positive impact for designing good UX.

DISCUSSION

This paper focuses on the gap between UX engineering and interaction design, and argues that there is a gap between these two activities. UX requirements may be tough, and it may be a challenge to produce design solutions that meet such requirements.

Would there be a direct gap between UX and software design?

One should understand that there does not exist a single 'right' interaction design solution: the UX requirements may be achieved with different kinds of design solutions. Thereby one should aim at producing such interaction design solutions that are easy to implement with software. In other words, one should aim for interaction design solutions that (1) meet the UX requirements and (2) are easy to implement with software. Probably this is achievable in most cases. But if not, then we have a gap between UX engineering and software design.

REFERENCES

[1] ISO/IEC (2010). 9241-210 Human-Centred Design Processes for Interactive Systems. ISO/IEC 9241-210: 2010 (E).

[2] Jokela, T. (2008). A Two-Level Approach for Determining Measurable Usability Targets. 5th COST294-MAUSE Open Workshop on Valid Useful User Experience Measurement (VUUM), Reykjavik, Island.

[3] Jokela, T., J. Koivumaa, et al. (2006). "Quantitative Usability Requirements in the Development of the User Interface of a Mobile Phone. A Case Study." Personal and Ubiquitous Computing 10(6): 345-355.

[4] Microsoft (2010). Windows User Experience Interaction Guidelines.

[5] Tidwell, J. (2006). Designing Interfaces: Patterns for Effective Interaction Design, O'Reilly.