Principles for the Selection of UCD Methods and their Implementation in the Usability Planner Tool

First Author Name (Blank if Blind Review)
Affiliation (Blank if Blind Review)
Address (Blank if Blind Review)
e-mail address (Blank if Blind Review)
Optional phone number (Blank if Blind Review)

Second Author Name (Blank if Blind Review)
Affiliation (Blank if Blind Review)
Address (Blank if Blind Review)
e-mail address (Blank if Blind Review)
Optional phone number (Blank if Blind Review)

ABSTRACT

Despite a growing interest in usability techniques in overall systems development, and the existence of standards aiming to guide the selection of UCD (User-Centered Design) methods, the diversity of usability methods available for application, and the lack of easy to use rules for selecting and applying them can make selecting and planning UCD activities a hard task with little or no support available. This paper describes proposed principles for selecting methods and their implementation in the Usability Planner tool. The principles are: a) identify the stages of design and development that would benefit most from use of UCD methods, b) identify the types of UCD activity that would provide the greatest benefit at each stage, c) identify the potential alternative methods that could be used to carry out each selected UCD activity, d) identify the practical constraints that should be considered when selecting from the potential methods, e) identify how each constraint influences selection of each potential method, and thus f) recommend which of the potential methods are likely to be most appropriate at each stage. An example is given of how the tool could be used.

Author Keywords
UCD method selection, development process, business benefits, business case, risk management.

ACM Classification Keywords
D.2.9 Software engineering management.

General Terms
Design, Human Factors, Management.

INTRODUCTION

UCD approaches and methods for systems development are used to help build systems with a good usability level. While there is general consensus on the overall UCD activities to undertake in development for dealing with usability, there is a great diversity of methods. Each author in the HCI (Human-Computer Interaction) field proposes his/her own version of a method and links them in a particular methodology.

On the other hand, in systems development at large, there is an increasing interest in usability as a system quality attribute [3]. This might be related to the increasingly wide use of systems by the general public, and the importance of graphical user interfaces [12], but also because UCD methods are gaining a wider acceptance as the way to obtain a higher quality systems. According to a survey of usability professionals [27], these methods will likely achieve wider user and greater impact.

Nevertheless, for the development organization aiming at good usability in their systems products, there is still a difficulty in choosing and applying usability methods. According to [24], "[UCD] methods are still underused and difficult to understand by software development teams and organizations". Developers with an interest in usability issues find the HCI field difficult to deal with in terms of usability methods and their integration in the overall systems development process, due to this lack of easy to apply knowledge on UCD method selection and planning in the development process.

When resource planning is discussed in a development project, usability professionals involved in a development team, or possibly developers self-taught in usability issues, may raise the need for using UCD activities. But these usability advocates experience difficulties in justifying investment in UCD methods in the development process in cost-benefit terms, or in avoidance of risks terms. This is particularly important in limited budget scenarios.

This paper discusses the process of selecting usability methods, and the design of a tool to support such a process.

OBJECTIVES

Our approach for the tool design considered a set of different contexts of use, representing the characteristics of the intended users and their needs, in relation to the task of selecting which UCD methods to use in systems development, and establishing the business case for their inclusion as part of the development activities.

The tool should support the needs of a variety of possible user profiles. The main user profiles considered are the following:

CHI 2012, May 5-10, 2012, Austin, TX, USA.

- Self-trained developer: Software developers with some but no extensive knowledge on HCI and UCD methods, will be mostly interested in knowing which methods to apply taking account of the specific constraints to be considered. The special case of a limited budget constraint will probably be of interest to this kind of user.
- Trainee/HCI student: Students will need to develop their skills in planning UCD methods according to particular constraints, and they would also like an overview of the alternative methods that can be used.
- Junior HCI consultant: A usability professional will be mostly interested in planning usage of UCD methods in development, taking account of possible constraints. These users will be more focused on UCD-related constraints, assuming that their responsibilities in a development project will be mainly circumscribed to pure UCD activities, and less dedicated to practices that fit into the overall development process.
- Expert HCI consultant: These senior professionals will have interest in method planning issues, with any kind of constraints, but they will also have a great interest in cost justifying usability activities, and aligning any possible UCD method planning with business priorities. Limited budget scenarios, being quite common will be specifically of interest to these users, as well.

As an example, the scenario of use described in Table 1 focuses on a Junior HCI consultant, and illustrates the possible difficulties in UCD method planning that could occur in a development project.

Jordan is a junior HCI consultant working in the user experience group at Satware, a software development company, and he has been assigned to the project Visiguide, that will develop a system to guide visitors to an institution through the different exhibits and rooms. He has been requested to advise which UCD activities and methods should be included in the project development process with a very restricted budget. He is under time pressure to deliver quick results, and he regards the project specifications delivered by the customer as not precisely defined.

He believes that considering usability is essential in order to develop a guide that will be effective for the visitors.

He is interested in selecting cost-effective usability methods that best fit the particularities of the Visiguide project. He thinks that most benefit will be gained by concentrating UCD resources on the early stages of the project.

He needs some guidance in which UCD methods would be most appropriate.

Table 1. Scenario of use for a junior HCI consultant.

APPROACHES TO METHOD SELECTION

What is a method?

There are several interpretations of what is meant by a UCD method, based on different ways of differentiating and categorizing methods in UCD.

1. Methods can be categorized by their intrinsic properties, for example as in ISO TR 16982 [17] (Table 2). This categorization distinguishes the practicalities of carrying out different types of methods. With this breakdown, the different types of methods can be used in different combinations for different purposes.

Type of method	**Description**
Observation of users	Systematic collection of information about behavior and performance
Performance-related measurements	Collection of quantifiable performance measurements
Critical incidents analysis	Systematic collection of specific events (positive or negative)
Questionnaires	Questionnaires to gather users' opinions about the user interface
Interviews	Interviews involving face-to-face interaction
Thinking aloud	Thinking aloud while using a system under test
Collaborative design and evaluation	Collaboration of different stakeholders in design or evaluation
Creativity methods	Elicitation of new features, usually extracted from group interactions
Document-based methods	Professional judgment based on examination of system documentation
Model-based approaches	Use of models to predict the users' performance
Expert evaluation	Evaluation based upon the expertise of the usability specialist
Automated evaluation	Automated diagnose the deficiencies compared to predefined rules.

Table 2. Types of methods in ISO TR 16982.

2. Methods can be categorized by whether users are involved, and whether a complete system, prototype or other representation, product or service of interest is available to be analyzed, designed or evaluated (Table 3). This is the approach taken in ISO WD 9241-230 (the current draft of the proposed revision of ISO TR 16982).

   This provides a useful broad distinction between methods, based on the practical issues of whether users are involved, and whether the method involves using an existing or prototype system. The proposed 14 categories of methods are more closely related to real use than the method types in ISO TR 16982.

	Methods directly involving users	Methods not directly involving users
Methods requiring a system	• Observing real use • User-based testing	• Inspection-based evaluation • Automated evaluation • Usage data analysis • Interface and interaction design
Methods that can be used without a system	• Observing users in their context • Interviews • Questionnaires • Collaborative methods • User self-reporting	• Assessing system-related information • Model-based approaches • Analytical methods

Table 3. Types of methods in ISO WD 9241-230.

3. Methods can be categorized by the type of user centered design activity, as in ISO 9241-210:
   - Understand and specify the context of use
   - Specify the user requirements
   - Produce design solutions
   - Evaluate

   However, these of activities are intended to be applied throughout the lifecycle, so for example, an evaluation method early in development will be very different to an evaluation method used before release.

4. Methods can be categorized by lifecycle stage, which has a close relationship to what is actually done during systems development. As this is the most natural classification for a systems developer, it is the one used in the tool. The disadvantage is that as in some cases essentially the same method can be used at different lifecycle stages, the same method can appear under different headings.

Another difference when categorizing methods is the level of specificity of each type of method. Many types of methods can be categorized at increasingly fine levels of detail, as for example shown for evaluation methods in Figure 1.

Another difficulty is that methods can be composed of sub-methods. For example, usability testing could include observation of users, performance-related measurements, questionnaires, interviews and thinking aloud. There can also be many variations in the elements that are used to create a method (for example whether the method involves users, developers, and/or usability experts; whether it is based on inspection, tasks and/or scenarios; the type of analysis, etc.). Particular permutations may be given a specific name (e.g. cognitive walkthrough), which may subsequently be used in modified forms, some of which are given a new name (e.g. simplified cognitive walkthrough).

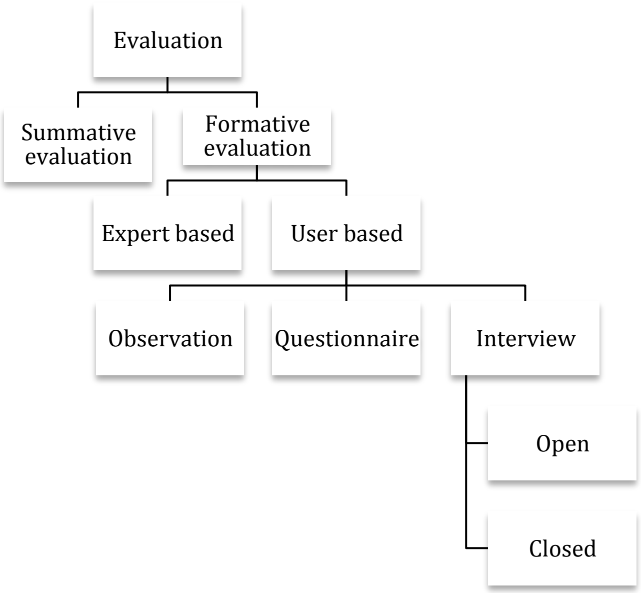


Figure 1. Decomposition of evaluation method categories.

Previous tools

Although there is a wide literature on the reliability and value of different usability methods, little has been published on how to select the most appropriate methods, although some collections of methods do state the advantages and disadvantages of each, e.g. [22], [20].

One of the first attempts was the INTUIT expert system developed by the HUFIT project [11] to recommend which user centered design methods should be used.

More recently UsabilityNet has provided a tool for filtering methods based on three criteria: limited time and resources, no access to users and/or limited skills/expertise [4].

The most sophisticated approach so far is the ISO TR 16982 standard that suggests criteria for selecting types of methods and indicates which types of methods are recommended or not recommended for each criterion. This includes comprehensive criteria, but is very difficult to apply because it involves reconciling numerous tables, and as mentioned above, the recommendations relate to types of methods that are categorized by their intrinsic properties.

All the previous approaches are based on applying constraints to already identified sets of methods or types of methods, to prioritize those that are likely to be most appropriate. However this assumes that the user of the tool already knows which potential methods might be needed in their situation. A more comprehensive tool should first help the user identify the potential methods that may be appropriate to use at different stages in a project.

Basis for method selection

The reason for using usability methods is to make specific contributions to user-centered design. As Wixon [28] says, “the goal is to produce, in the quickest time, a successful product that meets specifications with the fewest resources, while minimizing risk. […] In the world of usability work on real products embedded in a corporate and business framework, we must focus on factors of success, such as how effectively the method introduces usability improvements into the product."

Experienced usability professionals who are familiar with a range of methods will select the method that they believe will produce the most useful results taking account of practical constraints (such as those listed in Figure 3). If no method that they have used before appears appropriate, they may investigate using a different method, based on existing knowledge, or information obtained from other sources.

Principles for designing a tool to support method selection

A tool to support the process of selecting usability methods therefore needs to help the user of the tool make the following decisions:

a) To identify priorities:
   1. Which stages of design and development would benefit most from use of UCD activities?
   2. What types of UCD activity at each stage would provide the greatest benefit?

b) To identify potential methods:
   3. Which potential alternative methods could be used to carry out each selected UCD activity?

c) To select from the potential methods:
   4. What practical constraints should be considered when selecting from the potential methods?
   5. How does each constraint influence selection of each potential method?
   6. Which of the potential methods is most appropriate having applied all the relevant constraints to all the potential methods?

These steps are explained in more detail in the following section.

PROCESS FOR METHOD SELECTION

1. Which stages of design and development would benefit most from use of UCD methods?

To make this decision, the user needs to consider at which stages of design and development in their project the use of usability methods could give the greatest benefit.

The first step is to be aware of how improved usability could provide benefits and/or reduce risks in a particular project. [5] gives examples of the cost-benefits that that can be obtained from improved usability (Table 4, column 1).

However, as suggested in [6], use of user centered design methods is also an effective means of risk mitigation. Each of the potential benefits can also be interpreted as avoidance of potential risks (Table 4, column 2). In many situations, project managers will be more concerned with managing risk than the promise of potential economic benefits.

Cost-benefit	Associated risk reduction
Reducing development costs	Risk of increased development costs
Increased product sales	Risk of reduced product sales
Increasing traffic and sales on web sites	Risk of reduced traffic and sales
Increased productivity of users	Risk of reduced productivity
Reduced support and maintenance costs	Risk of increased support and maintenance costs

Table 4. Benefits and associated risks

Although these decisions cannot be automated, the tool has been designed to prompt the user to review these issues to help the user decide at which stages of design and development the use of usability methods could give the greatest benefit.

2. What types of UCD activity at each stage would provide the greatest benefit?

To make this decision, the user needs to be aware of all the types of UCD activity that could potentially be beneficial. ISO TS 18152 provides an exhaustive catalogue that can be used to support this decision, but the list of 122 activities can be somewhat daunting.

To make them easier to understand, these activities need to be categorized into manageable groups. One popular high-level categorization is in terms of the purpose of the activity such as the categories in ISO 9241-210 mentioned above, but for planning purposes, more specific categories are needed that relate to different stages of design and development.

IEC 62508 [14] has categorized the ISO TS 18152 activities by the project stages. The subset of these currently used in the tool are shown in Figure 2.

Alternative categorization could be used for users with different backgrounds, for example the categories in [23] for project managers, or in [9] for software developers.

3. Which potential alternative methods could be used to carry out each selected UCD activity?

An initial collection of potential methods have been identified that can be used to achieve the groups of activities in Figure 2. The methods were derived from the list in IEC 62508, the Usability Body of Knowledge [25], and the literature survey in [10]. These are methods with a track record of use in industry, which can be categorized by lifecycle stage.

1. Concept
    - Envisioning opportunities
    - System scoping
2. Planning
3. Understanding needs
    - Context of use
    - Tasks
    - Usability needs
    - Design options
4. Requirements
    - Context requirements
    - User requirements
5. Analyse requirements
6. Design/development
    - High level design
    - Prototyping
    - Usability evaluation

Figure 2. Project stage categories for UCD methods

4. What practical constraints should be considered when selecting from the potential methods?

The most comprehensive list of potential constraints on the use of usability methods is that in ISO TR 16982. For use in the tool, the wording of the constraints has been simplified, some additional constraints added (*Efficiency or accuracy is important,* and *General purpose: used in many different contexts*) and the resulting list has been categorized as shown in Figure 3.

It is also intended to introduce application domain constraints (e.g. web, consumer applications and professional applications).

5. How does each constraint influence selection of each potential method?

ISO TR 16982 relates each constraint to broad categories of methods categorized by their intrinsic properties. For the tool a table has been generated relating each constraint more specifically to each method in the tool. For each combination of method and constraint, the value can be "Unsuitable", or a weight on a four point scale: Less suitable, Neutral, More suitable or Very suitable.

The initial weights were created based on the judgment of the first author. It is intended that these are a starting point and will be refined based on feedback so that the weights are adjusted until the tool produces results that correspond as closely as possible with the judgment of experts. In practice the initial weights have produced results that have seemed plausible to people who have tried the tool in a demonstration environment, and for scenarios such as the one in the example below.

6. Which of the potential methods is most appropriate having applied all the relevant constraints to all the potential methods?

When no constraints apply, the potential methods for each group of activities are assumed to be equally suitable. As constraints are selected, Unsuitable methods will be eliminated, and for the remaining potentially suitable methods, weight is added for More suitable or Very suitable methods, and subtracted for Less suitable methods, thus producing a list of methods rated for suitability.

The tool can link to Internet resources such as the Usability Body of Knowledge [25] so that the user can find out more about each method.

Project constraints

- Need quick results
- Very restricted budget
- Usability important
- Uncertain specification

User constraints

- Difficult to involve users
- No access to users
- Some users have disabilities
- Mostly first time users

Task constraints

- Complex task
- Many tasks
- Safety or business critical system
- Organisational changes needed

Product constraints

- Efficiency or accuracy is important
- Adaptation of an existing system
- A well understood product
- Customisable product

Context constraints

- General purpose: used in many different contexts

Expertise constraints

- No usability expertise available

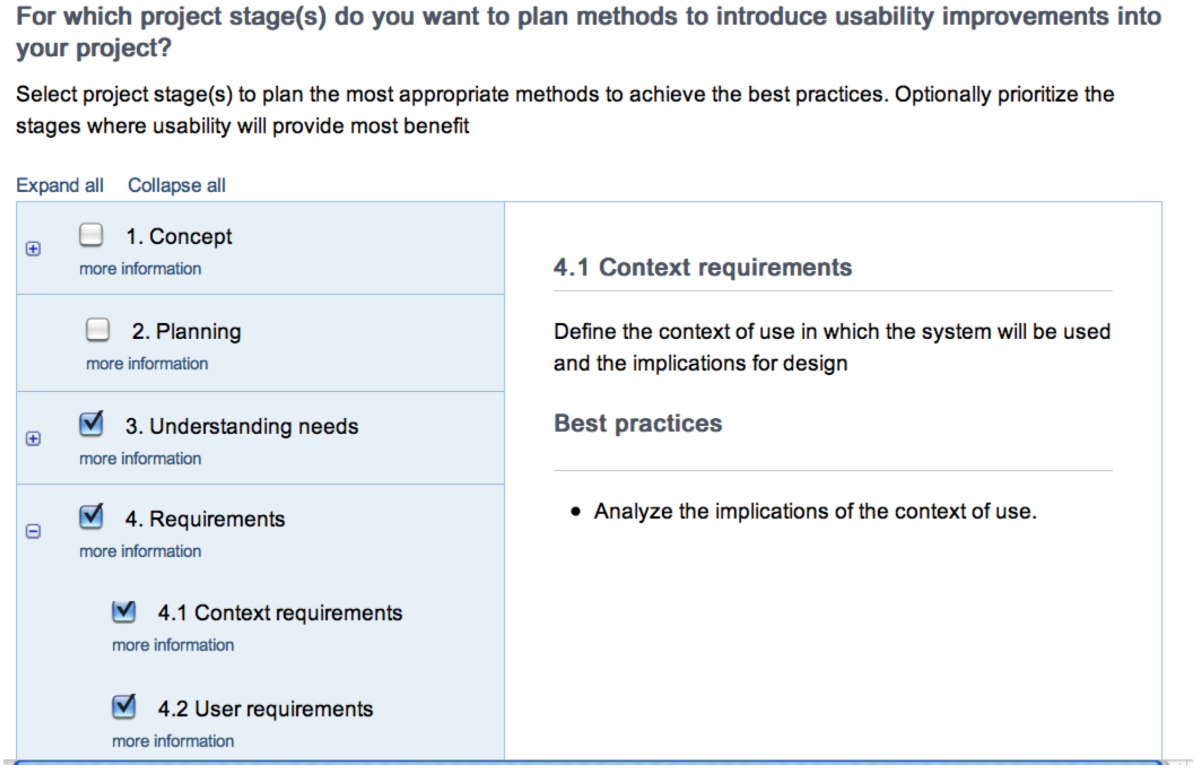
Figure 3. Constraints

DEVELOPMENT OF THE USABILITY PLANNER TOOL

The Usability Planner tool has been developed iteratively to meet the needs of the types of user identified above, starting with rough paper mock-ups, then wire frames created in Balsamiq [1], followed by online Balsamiq prototypes, before creating a fully functional web application. Informal usability testing using think aloud was conducted through numerous iterations with usability professionals and academics at meetings and conferences.

Some of the lessons learned were:

- Early prototypes had seven screens (Home, Establish business case, Specify stages for which methods are needed, Review the benefits of HCD, Specify constraints, Prioritize methods, and Review), but through iterative evaluation, this has been simplified to four screens (Home, Project stages, Methods, and Your plan).



For which project stage(s) do you want to plan methods to introduce usability improvements into your project?

Select project stage(s) to plan the most appropriate methods to achieve the best practices. Optionally prioritize the stages where usability will provide most benefit

Expand all   Collapse all

⊞ □ 1. Concept
more information

□ 2. Planning
more information

⊞ ☑ 3. Understanding needs
more information

⊟ ☑ 4. Requirements
more information

☑ 4.1 Context requirements
more information

☑ 4.2 User requirements
more information

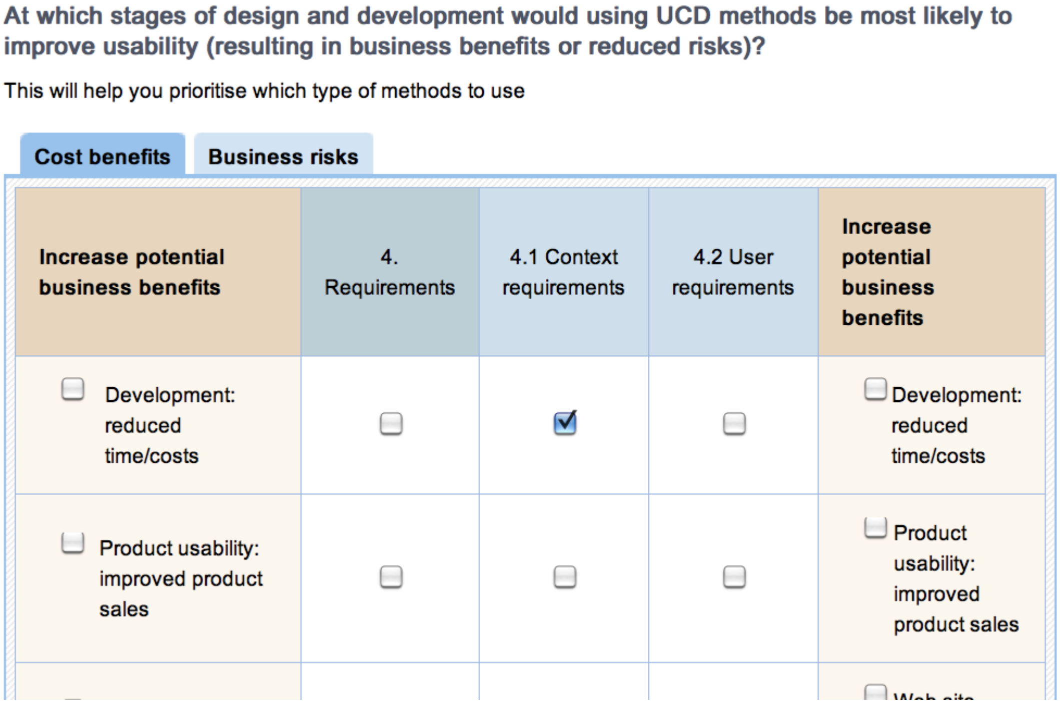
4.1 Context requirements

Define the context of use in which the system will be used and the implications for design

Best practices

- Analyze the implications of the context of use.

Figure 4. Project stages



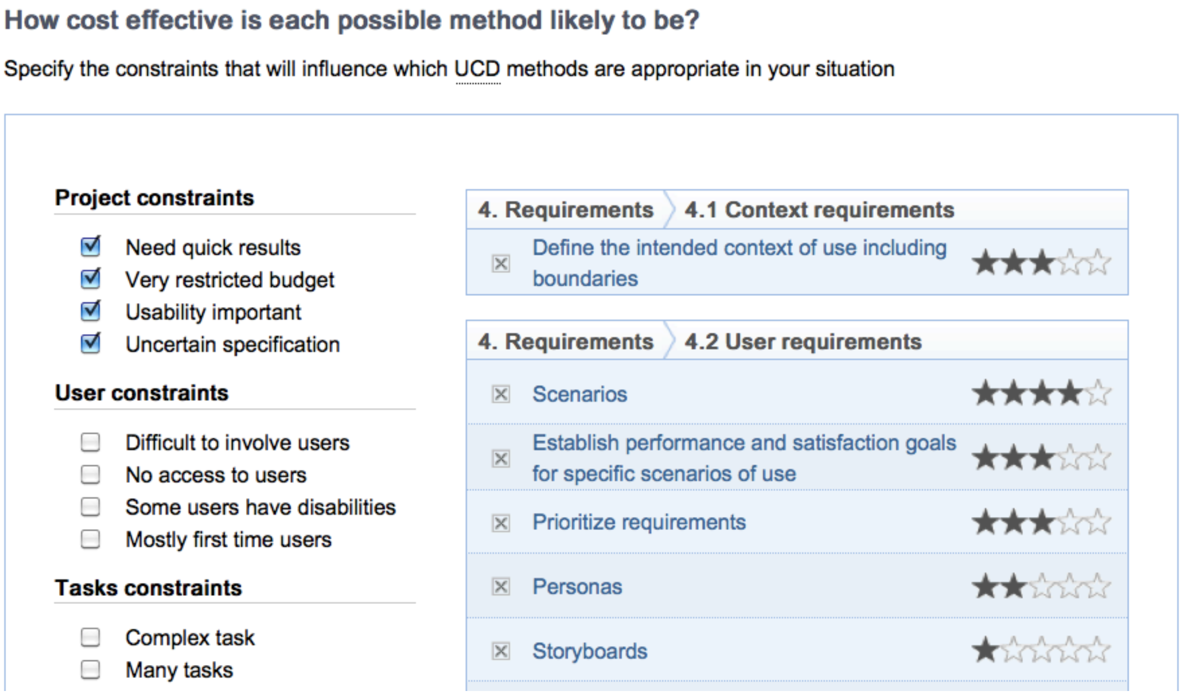
At which stages of design and development would using UCD methods be most likely to improve usability (resulting in business benefits or reduced risks)?

This will help you prioritise which type of methods to use

Cost benefits   Business risks

Increase potential business benefits	4. Requirements	4.1 Context requirements	4.2 User requirements	Increase potential business benefits
□ Development: reduced time/costs	□	☑	□	□ Development: reduced time/costs
□ Product usability: improved product sales	□	□	□	□ Product usability: improved product sales

Figure 5. Cost benefits



How cost effective is each possible method likely to be?

Specify the constraints that will influence which UCD methods are appropriate in your situation

Project constraints

☑ Need quick results
☑ Very restricted budget
☑ Usability important
☑ Uncertain specification

User constraints

☐ Difficult to involve users
☐ No access to users
☐ Some users have disabilities
☐ Mostly first time users

Tasks constraints

☐ Complex task
☐ Many tasks

4. Requirements › 4.1 Context requirements

☒ Define the intended context of use including boundaries ★★★☆☆

4. Requirements › 4.2 User requirements

☒ Scenarios ★★★★☆
☒ Establish performance and satisfaction goals for specific scenarios of use ★★★☆☆
☒ Prioritize requirements ★★★☆☆
☒ Personas ★★☆☆☆
☒ Storyboards ★☆☆☆☆

Figure 6. Specify constraints

- In order to make the purpose of constraints easier to understand, the prioritized list of methods changes dynamically on the same screen as the constraints are adjusted.
- Many people without management responsibility did not find the assessment of benefits and risks relevant: they just wanted assistance in selecting appropriate methods. For this reason, the prioritization of stages has been made a secondary, optional feature.

EXAMPLE OF TOOL USE

This is an example of the recommendations made by the current implementation of the tool, when applied to the scenario in Table 1.

In this scenario, Jordan has decided that he needs to use methods early in development. Considering the list of stages in Figure 2, as the concept has already been established, he selects the *Understanding Needs* and the *Requirements* stages (Figure 4). Requirements are subdivided into *Context requirements* and *User requirements*. He decides that a better understanding of *Context requirements* will contribute significantly to reducing the development time and costs (Figure 5).

From the constraints, he selects: Need quick results, Very restricted budget, Usability important, and Uncertain specification. Based on these constraints, the current version of the tool first recommends: *Success critical stakeholder identification, Context of use analysis,* and *Work context analysis*. Based on the description in the tool, Jordan decides that *Work context analysis* is not appropriate, but *Context of use analysis* sounds important, so he links to the Usability Body of Knowledge for more information. The tool also recommends *Early prototyping and usability evaluation* and *Heuristic/expert evaluation*, which he agrees with.

For *Requirements* the recommendations are: *Define the intended context of use including boundaries*, *Scenarios*, *Establish performance and satisfaction goals for specific scenarios of use*, and *Prioritize requirements* (Figure 6). He checks with the Usability Body of Knowledge for more information on *Defining the intended context of use*, and *Establishing performance and satisfaction goals*.

From this example it can be seen that the results provided by the tool are plausible, but further refinement could be obtained by introducing constraints related to the application area. For example for a purely consumer device, *Work context analysis* would not be appropriate.

DISCUSSION

The design of the tool is based on the hypothesis that the process that experienced usability professionals use to select appropriate methods is a combination of consideration of estimated cost benefits or risks and consideration of applicable constraints.

The tool and the principles on which it is based are intended to help the user consider all the relevant factors that should influence which UCD methods to use at particular project stages. The tool can assist by:

1. prompting users to think about prioritizing UCD activities based on cost-benefits and risks;
2. helping users consider the benefits that can be obtained from UCD activities at different lifecycle stages;

3. suggesting a range of possible methods that can be used to carry out specific UCD activities;
4. providing a checklist of constraints that may apply;
5. suggesting a prioritized list of methods for each project stage, based on the constraints;
6. enabling the user to explore how different constraints influence the appropriateness of particular methods.

Analysis of scenarios such as the one above and informal feedback from users suggest that even with the initial weights, the tool is producing useful results.

It is the intention to iteratively refine the constraints and their relative weighting to as closely as possible replicate the decisions made by experienced usability professionals.

But there are some intrinsic limitations in a tool such as this.

Even after adding additional constraints and refinement of the weighting, it is not expected that the tool will always produce the same recommendations as an expert. One reason is that usability professionals typically customize methods to the specific project environments [21]. In this sense each "method" is really a family of potential methods that an expert can scale or adapt to different situations. So the expert may find ways of adapting methods in situations where they might not usually be appropriate.

Although a tool like this cannot be a substitute for expertise, it can help the users understand the basis for making decisions, and lead them to investigate the possibility of using methods that they may not have considered before.

The tool could be particularly useful in helping less experienced users such as students, trainees or a self-trained developers explore different methods and learn more about UCD.

The tool could help a junior professional plan use of UCD methods in a project, and understand the implications of different constraints. It could help a more senior professional consider the implications of cost benefits and risks, and produce a considered plan for use of methods that they could justify to project management.

The approach taken by the tool is based on the principles of value-centered systems engineering [8] as applied in the context of UCD [23]. Value based systems engineering promotes rational behavior aimed at satisfying the success-critical stakeholders in systems development. The challenge for a usability professional is that it is often difficult to gain acceptance for the use of UCD methods employing rational decision criteria, particularly when using agile processes and in less formal development environments. It is hoped that the results produced by the tool will both assist the user to make better-informed decisions and to make a more convincing business case for the use of particular methods.

CONCLUSIONS

Few usability professionals are familiar with ISO standards, and even for those who are, standards like ISO TS 18152 and ISO TR 16982 are hard to use. For example Jokela [19] reports that companies find standards like ISO TS 18152 too complex and too difficult to understand, and the British Computer Society Special Interest Group in Software Testing describes ISO TR 16982 as "somewhat academic" [2]. A tool can make the content of these standards much more useful.

The process of iteratively developing a prototype with user feedback helped refine what was initially a very complex process [7] into an increasingly simple procedure.

The next steps are to evaluate use of the tool in different environments, particularly in education and training, and when used by self-trained developers and junior staff, both to refine the design and content, and to obtain feedback on its usefulness.

The following future developments are under consideration.

- More detailed criteria could be added that would enable more specific versions or implementations of methods to be recommended. Research would be required to establish how much detail different types of users would find beneficial in the recommendations, to provide them with a basis for obtaining the additional information (and potential expertise) that would be needed to actually use the methods.
- Providing an estimate of the effort needed to carry out methods (which could use the figures in [5] as a starting point).
- Investigating whether the tool would benefit from customization for use in different environments.

The Usability Planner tool [26] is offered freely as a web tool to anyone needing support in the decision of which usability methods to apply in the development of interactive systems, according to the particular constraints for the project or organization. The code will be offered as an open source license (GNU General Public License, version 3 [13]).

REFERENCES

1. Balsamiq Mockups. balsamiq.com/products/mockups
2. BCS SIGiST. *Usability Guidelines* www.testingstandards.co.uk/usability_guidelines.htm (2006)
3. Bevan, N. Quality in use: meeting user needs for quality, *Journal of Systems and Software*, 49(1) (1999)
4. Bevan, N. UsabilityNet Methods for User Centred Design. *Human-Computer Interaction: Theory and Practice (Part 1), Volume 1 of the Proceedings of HCI International 2003*, 434-438. Lawrence Erlbaum.
5. Bevan, N. Cost benefits framework and case studies. In R.G. Bias, & D.J. Mayhew (eds), *Cost-Justifying*

Usability: An Update for the Internet Age. Morgan Kaufmann (2005)

6. Bevan, N. Reducing risk through Human Centred Design. *Proceedings of I-USED 2008*.
7. Bevan, N. Criteria for selecting methods in user-centred design. *Proceedings of I-USED 2009*, Uppsala, Sweden (2009)
8. Boehm, B, Jain, A. *An Initial Theory of Value-Based Software Engineering, in Value-Based Software Engineering*, S. Biffl, A. Aurum, B. Boehm, H. Erdogmus, and P. Gruenbacher, Eds.: Springer-Verlag. http://sunset.usc.edu/csse/TECHRPTS/2005/usccse2005-505/usccse2005-505.pdf (2005)
9. Ferre, X. & Bevan, N. Providing guidance to software developers on selecting usability methods through Usability Planner. In: Forbrig, P., et al (eds) *5th workshop on software and usability engineering cross-pollination: Patterns, Usability and User Experience (PUX 2011).*
10. Ferre, X., Juristo, N., Moreno, A.M.: Improving Software Engineering Practice with HCI Aspects. *Software Engineering Research and Applications - Lecture Notes in Computer Science*. Vol 3026, Springer (2004) 349-363
11. Galer, M. & Russell, A. The presentation of human factors to designers of IT products. In *Human-Computer Interaction: INTERACT '87*, eds. H.J. Bullinger and B.Shackel, Amsterdam, North Holland (1987)
12. Glass, R.L. *Facts and Fallacies of Software Engineering*. Addison-Wesley, Boston (MA), USA (2003)
13. GNU Project. The GNU General Public License v3.0. http://www.gnu.org/copyleft/gpl.html
14. IEC 62508. Guidance on human aspects of dependability (2010)
15. ISO 9241-210. Human-centred design for interactive systems (2010)
16. ISO WD 9241-230. Human-centred design and evaluation methods (2011)
17. ISO TR 16982. Usability methods supporting human-centred design (2002)
18. ISO TS 18152. A specification for the process assessment of human-system issues (2010).
19. Jokela, T. Certification of the User-Centredness of Development Organisation – A Way for Ensuring User Acceptance even before the Development of Software? *Proceedings of UITQ 2005 Workshop* (2005)
20. Maguire, M. Methods to support human-centred design. *Int. J. Human-Computer Studies*, 55 (2001)
21. Monahan, K., Lahteenmaki, M., McDonald, S., & Cockton, G.: An Investigation into the Use of Field Methods in the Design and Evaluation of Interactive Systems. In *Proc. HCI 2007* (2007).
22. Nielsen, J. *Usability engineering*. Academic Press (1993)
23. Pew, R. W., Mavor, A. S. (eds): *Human-System Integration in the System Development Process: A New Look.* National Academies Press. www.nap.edu/catalog/11893.html (2007)
24. Seffah, A. and Metzker, E. The obstacles and myths of usability and software engineering. *Communications of the ACM 47*, 12 (Dec. 2004), 71-76. (2004)
25. UPA: Usability Body of Knowledge. www.usabilitybok.org
26. UsabilityPlanner. www.usabilityplanner.org
27. Vredenburg, K., Mao, J.Y., Smith, P.W., Carey, T. A *Survey of User-Centered Design Practice*. Proc. of CHI-2002. (2002)
28. Wixon, D. Evaluating usability methods: why the current literature fails the practitioner. *Interactions*, 10 (4) (2003)