Loughborough University

Loughborough University
Institutional Repository

Using human factors standards to support user experience and agile design

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Citation: MAGUIRE, M., 2013. Using human factors standards to support user experience and agile design. In: Stephanidis, C. and Antona, M. (eds.). Proceedings of 2013 7th International UAHCI Conference, HCI International, Las Vegas, USA, 21-26 July, pp. 185-194.

Additional Information:

- The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-642-39188-0_20.

Metadata Record: https://dspace.lboro.ac.uk/2134/15600

Version: Accepted for publication

Publisher: © Springer

Please cite the published version.

Loughborough University

This item was submitted to Loughborough's Institutional Repository (https://dspace.lboro.ac.uk/) by the author and is made available under the following Creative Commons Licence conditions.

creative commons

COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full license).

Disclaimer

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/

Using human factors standards to support user experience and agile design

Martin Maguire
Loughborough Design School, Loughborough University
LE11 3TU, UK
m.c.maguire@lboro.ac.uk

Abstract: The ISO 9241-210 standard provides a framework for human-centred design (HCD) activities comprising the four stages: context of use, specification of user and organisational requirements, design solutions, and evaluation against requirements. Other parts of the 9241 standard cover user interface design and usability. This paper uses the HCD framework to emphasise user experience (UX) design and methods used to help create good user experiences. It also relates the framework to an agile software development environment. It is concluded that the flexible and iterative nature of ISO 9241-210 makes it a good basis for both user experience design and an agile development process.

Keywords: Human-centred design, User experience (UX), Usability, Agile, Standards

1 Introduction

Despite the development of electronic products that people become attached to and the prevalence of user friendly graphical interfaces, many of them are hard to use. This leads to a poor user experience causing people to abandon them or fail to use them effectively. The same applies to work systems where poorly designed user interfaces can cause frustration for those who have to work with them on a daily basis leading to inefficient working and costly technical support. Redeveloping a product or system to improve both the usability and user experience is always more expensive once it has been developed. An agile approach allows a system under development to be adapted more easily to user feedback and changing user needs.

Three important concepts for producing more acceptable systems for users are *user experience (UX)*, *usability* and *agile design*. Although there are many definitions of each of these concepts, some useful and concise ones are as follows:

User experience: 'A person's perceptions and responses that result from the use or anticipated use of a product, system or service' [1] and 'All-aspects of the end-user's interaction with the company, its services, and its products' [2]. In general UX is con-

cerned with how people feel about a system. Guo [3] defines four elements of user experience: value (is it useful?), usability ('is it easy to use?'), adoptability (is it easy to start using?'), and desirability (is it fun and engaging?'). The last of these seems more relevant to consumer products than traditional systems although the 'gamification' of work systems is now a hot topic for research. Rubinoff [4] considers the UX of websites and expresses the main components as: branding, usability, functionality and content. He also provides opinion rating scales for evaluating each component

Usability: A common definition is 'the ease of use and learnability of a human-made object [5]. The ISO 9241-11 definition is 'The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use' [6]. In this way usability is often seen as a quality objective forming part of the user experience, and focusing on the user's ability to achieve objectives or tasks. This definition lends itself to the objective measurement of task achievement, while satisfaction in completing the tasks requires subjective assessment. However usability is also defined as a set of attributes which must be designed into a software product to make it easy to use [7], [8]. This allows the use of design principles or heuristics to evaluate the usability of a system.

Agile: 'A group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change.' [9]. Thus it is seen as a more flexible approach to system development that breaks the design into smaller chunks with frequent reviews and adaption of user requirements if required. These ideas have existed for some time [10] but the term was introduced in 2001 with the 'agile manifesto' [11].

The human factors community has done much work to develop a human-centred design (HCD) approach to system design and dialogue design principles and guidelines for specific user interface styles (menus, form-filling, GUIs, etc.) This paper looks at how human factors standards can be used as a framework for supporting usability and UX design and agile software development.

2 The HCD process and user experience design

According to the ISO 9241 human-centred design standard [1] there are four essential activities which should be undertaken in order to incorporate user needs into the software development process.

- Understand and specify the context of use;
- Specify the user and organisational requirements;
- Produce design solutions;
- Evaluate designs against requirements.

The activities are carried out in an iterative fashion with the cycle being repeated until the design objectives have been attained. These activities are described in more detail and how they relate to user experience design. It is also shown how they can be ap-

plied to agile software development and how they can support this process, as illustrated in Figure 1 below.

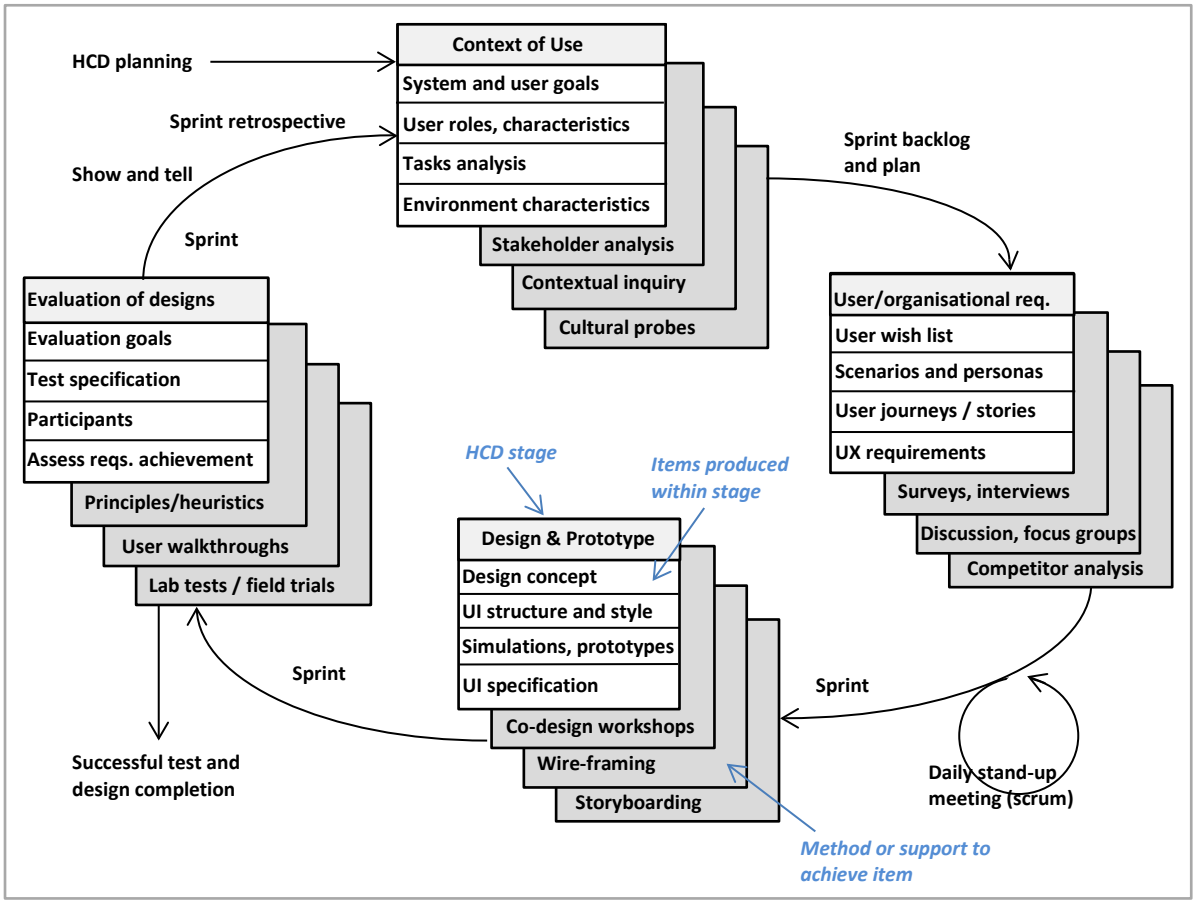

Fig. 1. ISO human-centred design process annotated for UX design and agile development

The HCD framework is extended as follows:

- Each of the main four activities (context of use, user and organisational requirements, etc.) is shown as part of a cycle. For each activity there is a set of outputs or items produced within that stage (e.g. system and user goals, user roles and characteristics, etc.).
- Methods that might be used to generate those outputs are shown as boxes overlaid by each activity (e.g. stakeholder analysis, contextual inquiry, etc.). These methods are applicable to either or both multi-user system and consumer product development.
- The iterative cycle is annotated with labels indicating possible stages related to an agile development process.

An agile process starts with a wish list of user requirements, and a plan for a set period of time (e.g. two weeks) to implement a subset of those functions. This could form one iteration around the HCD activity cycle, called a 'sprint'. The activity involves a multi-disciplinary team of people in different roles: the product owner, business analyst, team leader (or 'scrum master'), UX team, front and back end developers, and QA testers. At the end of each sprint, there is 'show and tell' session to review what functions have been implemented, and amend the 'sprint backlog' to show what functions are left to be developed. The aim at the end of each sprint is to produce a potentially shippable product increment (although this would not be appropriate for all types of system). Another feature which contrasts with traditional systems design processes is that agile is open to user requirements changing and the sprint backlog being amended ready for a new plan for the next iteration. To help monitor progress there is a daily stand up at the beginning of each day where the whole team highlight any problems or ask questions before work continues. This meeting is called a 'scrum' and is led by the scrum master who acts as a buffer between the team and any distracting influences. See [12] and [13] for full details.

The HCD activities described by the ISO human-centred design standard can be seen to fit into an agile process, with each iteration conducted over a relatively short period of time. Thus the human factors activities may need to be carried out in a less formal way than traditionally while still maintaining validity and high standards.

The following sections describe the four main HCD activities and how they might be adapted to emphasise user experience and agile design. Before these activities commence, there should be a planning phase to organise how they fit into the overall system development process. This will determine the formation of the human factors or UX group and how they will integrate with the development team to make effective inputs and evaluate prototype designs.

2.1 Understand and specify the context of use for the system

The quality of use of a system, including usability and health and safety aspects, depends very much upon the context in which the system will be used. For example in an office or workplace there are many features of the environment as well as characteristics of the users and tasks which can impinge on the usability of a new software product. Capturing this information is not only important for informing design decisions but also to provide a sound basis for later evaluation activities. In some cases contextual information may already be known, particularly where a system has been used for some time and only upgrades or enhancements are being made. If a new product or system is to be introduced, then the relevant context information must be collected and understood. The characteristics of the users may include features such as the users' age range, gender, task and IT skills, motivation and knowledge. It will be useful to record what training or skills the users already have in using certain types of software e.g. a particular operating system or application. This may be based on professional or work experience but equally may be acquired through the use of personal technology e.g. a home computer, tablet or mobile phone. Task descriptions may include the procedural steps identified through task analysis, frequency of per-

formance, level of complexity and accuracy required. To assist with this process, the ISO/IEC DIS standard 25063 specifies how context of use descriptions should be compiled for existing or new systems [14]. The socio-technical system in which the system is placed may also influence usage such interruptions by other workers, requiring that the user be able to save work uncompleted or to perform tasks concurrently. See [15] for a discussion of how user interface design may affect or be affected by the socio-technical system in which it is placed. Effective user representation covering the different stakeholders is necessary for all this information to be made explicit.

Supporting user experience design. A range of methods exist to explore the context of use. In a multi-user system, stakeholder analysis needs to be done in order to ensure that all the different user roles and types are covered. Interviews and observations may be conducted as part of a *contextual inquiry*, studying the user's current activities in their own natural environment – at the user's workplace, at home, on the move, etc. The researcher watches users perform their own work tasks and discusses with them any artefacts they generate or use. In addition, the researcher gathers detailed re-tellings of specific past events when they are relevant to the project focus. The aim is to find out what motives them, what frustrates them and what would really help. This is the starting point for creating a good user experience [16].

Where it is necessary to gather information from users with minimal influence on their actions, or when the process or event being explored takes place intermittently or over a long period, *cultural probes* are useful [17]. For example, they could be used, to explore errors in process control caused by the equipment used in order to redesign it, or situations where people have found the consumption of video at home particularly enjoyable, with the aim of designing AV equipment to promote them. Selected participants are briefed, given a kit of paper, pens, diary, cameras, etc. and briefed what to record i.e. specific events, feelings or interactions over a specified time period. Typically, a follow-up interview is conducted after the briefing session to ensure that participants are actively engaged, and are collecting the required information. At the end of the specified period, the materials are collected and analysed with a de-briefing session. Hearing about people's lives through their own documentary records helps to understand what is important to them – key information for UX design.

2.2 Specify the user and organisational requirements for the system

Requirements elicitation and analysis is widely accepted to be the most crucial part of software development. Indeed, the success of a software development programme can largely depend on how well this activity is carried out. Building on the context of use descriptions obtained from the previous activity, an explicit statement of the user requirements for the new software should be formulated. These may include the functional and non-functional requirements (including usability and UX requirements). The functional requirements for a cooking application might include the range of recipes stored, audio instructions for cooking them, and the resulting energy cost calculations. The non-functional requirements of an application may be for it to usable

by at least 90% of a sample of users without instructions, for users to be able to find a recipe in no more than 3 steps, and to receive user satisfaction ratings of at least 4 or 5 on a five point scale from 80% of the user sample. Potential metrics are provided in [6]. The requirements and metrics should relate to the user or task characteristics and the organisational or physical context in which it is located (although these may not necessarily be replicated in the test). It is sometimes hard to specify usability requirements in a testable form as users often find it difficult to specify what metrics and criteria values should be applied and there is the danger that designing to pass the tests may be to the detriment of the whole design Also requirements are likely to change during system design as users see the early designs and envisaging new usage possibilities.

Supporting user experience design. User requirements are normally captured in a user requirements document and then once a contract has been signed are fixed. However with agile development the design can remain flexible and functions may be dropped and new functions proposed as the design develops. The initial list is specified as a 'wish list' and the customer, design team and user representatives will agree on which subset of functions to develop first and then make further selections in subsequent interactions. To assist with the process, *scenarios and personas* [18] may be developed, recording typical usage situations and illustrating different user types based on the context of use activity. They are intended to act as stimulus for the design team of different user characteristics and situations that should be designed for.

User journeys are another useful technique [19]. These are the step-by-step journeys that a user takes to reach their goal with the current system. A journey may, for example, consist of a number of website pages and decision points that carry the user from one step to another to reach their goal. This journey is then redesigned to form an 'ideal' user journey free from frustration. The output is traditionally a flow diagram demonstrating each page and decision point throughout the entire process.

Another technique in agile development is the creation of *user stories.* Each story is one or more sentences in the everyday or business language of the system user or stakeholder that captures what a user does or needs to do as part of his or her job function [20]. User stories are used within agile software development as the basis for defining the functions a business system must provide and to facilitate requirements management. It captures the 'who', 'what' and 'why' of a requirement in a simple, concise way, often limited in detail, by what can be hand-written on a small paper notecard. User stories are written by or for the product owner as that user's primary way to influence the functionality of the system being developed. They may also be written by developers to express non-functional requirements (security, reliability, performance, quality, etc.).

User experience requirements may overlap with the user wish list and include both functional and non-functional requirements. It may be helpful to categorise them appropriately and use them to create statements for testing their achievement during the evaluation of the design.

For example from [4] in relation to websites:

- Branding: *"The site provides visitors with an engaging and memorable experience";*
- Functionality: *"Users receive timely responses to their queries or submissions";*
- Usability: *"The site prevents errors and helps the user recover from them";*
- Content: *"Content is structured in a way that facilitates the achievement of user goals".*

UX can then be assessed by asking users to what extent they feel that the proposed design meets each of these statements.

2.3 Produce designs and prototypes of the system

Having identified the relevant contextual information and usability requirements for the system, the next stage is to explore ideas for design solutions. This will then lead to the creation of a high level UI flow chart to show the main structural elements of the design. The general style of the design should also be considered including high level screen layouts or wireframes. Designs are brought to life by creating visualisations, simple mock-ups or interactive prototypes representing the proposed design and presenting them to representative samples of users to perform tasks and provide feedback (within the assessment activity 2.4).

Early prototypes may be made using basic materials such as paper, card, post-it notes, and stickers, to create representations of screens, menus and windows that can be manipulated by hand to support user testing. Alternatively the prototype may be software based and interactive, giving a closer representation of the final product. However while realism can have its advantages, it is better to avoid putting too much effort into a particular version until the design concept has been worked out and validated. Programs such as Visio and PowerPoint are often sufficient to create simple prototypes, although specialist prototyping and wire-framing tools exist such as Axure RP and Balsamiq which can provide more advanced interaction capabilities.

As the design develops, a user interface (UI) specification should be produced to ensure that the design covers all aspects of the interaction required to support the user's tasks [21]. The specification should include a visual overview of each screen and include display rules, specific messages in specific conditions, and links to other screens. This helps the design team in thinking through the UI design and how users will actually experience content within it.

The ISO 9241 standard includes many parts on software user interface design. These include guiding principles (parts 110 onwards); user support (121 onwards); dialogue techniques such as menus, commands, direct manipulation (130 onwards); technology dependent interface components and technologies (141 onwards); application-area specific (151 onwards) and tactile and haptic interactions (900 onwards). Guidance on software accessibility is provided in part 171 and [22].These documents can be used to assist the creation of design solutions and can provide the basis for creating checklists to evaluate prototypes.

Supporting user experience design. The first step in the process of developing a design is usually to conduct a conceptual design meeting to generate ideas for the first user prototype. This should be done using the list of UX and usability requirements generated previously. In accordance with the brainstorming method, all ideas should be recorded no matter how radical or improbable they may seem. They are all then considered and a selection of them made as a basis for one or more potential designs. Before mapping out design solutions, it is useful to stay at a high level and think about the overall experience that the design should offer the user. This may be thought about in terms of the general impression to be conveyed, major features, likely starting points for different tasks and possibly visual style and colour schemes. It may be beneficial to try out various styles, stimulated by the scenarios and personas or to make up a *mood board* – a collage of images, text, and objects to illustrate visually the direction or style of design to be pursued.

User-computer interface or dialogue principles are also a helpful reference to ensure that the design will offer a good user experience. The ISO 9241-110 standard [23] describes seven user-computer interface or dialogue principles which, as shown in Table 1, give some coverage of Guo's four elements of user experience [3]:

Table 1. ISO 9241 Dialogue principle categorised by UX elements

Value: • Suitability for the task	**Usability:** • Self-descriptiveness • Controllability • Error tolerance
Adoptability: • Conformity with user expectations • Suitability for learning	**Desirability:** • Suitability for individualisation

Within [1], each principle is accompanied by recommendations to illustrate it together with examples from specific contexts of use. These are expressed in general terms and not specific technologies. Within the other parts of the ISO standard, user interface technique descriptions should be kept updated with new styles of interaction that promote a good UX e.g. touch screen gestures to manipulate content (swiping, enlarging and reducing) and remote gestures for device control. These styles of interaction may be of benefit to all sectors and not just consumer and mobile products.

2.4 Carry out user-based assessment of the system or prototype

Once a prototype or an operational version of the software is available, a user-based assessment of the product can be carried out. This activity will confirm the extent to which usability and UX objectives have been met as well as providing further information for refining the design. It is advisable to carry out evaluations at the earliest opportunity, before making changes becomes too expensive. There are a number of aspects to this activity including: choice of assessment method, planning the evaluation, recruiting user participants, collecting and analysing the results, and reporting

recommendations for change. The activity is repeated for each design iteration to evaluate whether the user and organisational objectives have been met.

Supporting user experience design. A number of user scenarios, based on the personas, should be developed relating to the different stakeholders and their tasks. During testing, each scenario should be carried out by each user with the latest design prototype. The test session should record task effectiveness (completion and accuracy), task efficiency and user satisfaction (see [6]). The user should also be given the chance to comment freely on the design once they have experienced it. The results are assembled and metrics calculated. A decision should then be made to decide whether the design is acceptable or needs to be refined with a modified prototype. A heuristic evaluation may also be carried out based on dialogue principles [23] or heuristics, for example [24] and [25], to help show where the usability or UX problems lie. When conducting user assessments, being aware of user attitudes is also important. Users may complete the tasks satisfactorily but a lack of enthusiasm may indicate that it is unappealing in some way.

3 Conclusions

This paper shows how the ISO 9241-210 Human-Centred Design standard [1] represents a useful framework for user experience design in an agile software environment, while other parts of the standard such as ISO 9241-110 Dialogue principles [23] and ISO 9241-11 Guidance on usability [6], can support it. The agile concept was also developed in order to create software efficiently and flexibly which in itself can promote iterative human-centred design. It may be argued that adopting the ISO framework as part of an agile process helps to ensure that the human factors activities are given a clear structure to help ensure that they are carried out effectively to create software that both meets user and organisational needs and gives a great user experience.

4 References

1. ISO 9241-210: Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems. Geneva: International Organisation for Standardisation (2010)
2. Nielsen-Norman Group: www.nngroup.com/about-user-experience-definition (2013)
3. Guo, F.: More than usability: the four elements of user experience, part 1. UXmatters. www.uxmatters.com/mt/archives/2012/04/more-than-usability-the-four-elements-of-user-experience-part-i.php 24 April (2012)
4. Rubinoff, R.: How to quantify the user experience, Sitepoint. http://www.sitepoint.com/quantify-user-experience/ 21 April (2004)
5. Wikipedia: Usability. http://en.wikipedia.org/wiki/Usability (2013)

6. ISO 9241-11: Ergonomics of human-system interaction – Part 11: Guidance on usability. Geneva: International Organisation for Standardisation (1998)
7. ISO/IEC 9126: Software engineering – product quality. Geneva: International Organisation for Standardisation (2001)
8. Bevan, N.: Human-computer interaction standards. Proceedings of the 6th International Conference on Human Computer Interaction, Yokohama, July 1995. Anzai and Ogawa (eds.), Elsevier (1995)
9. Wikipedia: Agile software development. http://en.wikipedia.org/wiki/Agile_software_development (2103)
10. Edmonds, E. A.: A process for the development of software for non-technical users as an adaptive system. General Systems vol. 19, pp. 215–18 (1974)
11. Highsmith, J.: Manifesto for agile software development: Agile Alliance. http://agilemanifesto.org/ (2001)
12. Crispin, L., Gregory, J.: Agile testing: a practical guide for testers and agile teams. Addison-Welsey Signature (2009)
13. Jongerius, P.: Get agile: scrum for UX, design and development. BIS Publishers BV (2013)
14. ISO/IEC DIS 25063: Systems and software engineering – systems and software product quality requirements and evaluation (SQuaRE) – common industry format (CIF) for usability: context of use description (2011)
15. Maguire, M.: Sociotechnical systems and user interface design – 21st century relevance, Applied Ergonomics: Special issue on Advances in Socio-Technical Systems Understanding and Design (In press) (2013)
16. Chinnapattan, P.: A guide to conducting contextual inquiry user research. http://www.webcredible.co.uk/user-friendly-resources/web-usability/contextual-inquiry.shtml (2013)
17. Stalker-Firth, R.: Inside your users' minds: the cultural probe. A List Apart. http://alistapart.com/article/culturalprobe 27 March (2013)
18. Nielsen, L.: Personas – User focused design. Springer (2013)
19. Experience solutions: What are user journeys? www.experiencesolutions.co.uk/questions/what_are_user_journeys.php (2013)
20. Wikipedia: User story. http://en.wikipedia.org/wiki/User_story (2013)
21. Brandenburg, L.: How to create a user interface specification. www.bridging-the-gap.com/how-to-create-a-user-interface-specification/ (2013)
22. ISO/IEC 24786: Information technology – user interfaces – accessible user interface for accessibility settings. Geneva: International Organisation for Standardisation (2009)
23. ISO 9241-110: Ergonomics of human-system interaction – Part 110: Dialogue principles. Geneva: International Organisation for Standardisation (2006)
24. Nielsen, J.: 10 Usability heuristics. Nielsen Norman Group www.nngroup.com/articles/ten-usability-heuristics/ (2013)
25. Shneiderman, B.: Eight golden rules of interface design. http://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html (2013)