NordDesign 2018
August 14 – 17, 2018
Linköping, Sweden

Challenges in Agile Universal Design of ICT

Øyvind Nordeide Hjartnes[1], Miriam Eileen Nes Begnum[1]

[1]Department of Design, Faculty of Architecture and Design, Norwegian University of Science and Technology (NTNU), Teknologiveien 22, 2815 Gjøvik, Norway
oyvn@stud.ntnu.no
miriam.begnum@ntnu.no

Abstract

Universal design (UD) of ICT is about creating solutions that are usable and accessible for as many end-users as possible. Currently, agile development is a common approach in ICT-projects. This article investigates the challenges for ensuring UD in agile ICT-projects. We propose the term «agile universal design» (AUD) to denote UD in agile ICT-projects. Through a scoping review, we find that traditional user-centered and quality control activities may be seen as disruptive in the agile process. On the other hand, promotion of stakeholders and user involvement throughout the development process and in all phases fit well with UD approaches. Seven practical AUD challenges are identified and point to the fact that securing high quality usability aspects for users with a diverse set of needs require more than limited early attention followed by mere sporadic user focus. We find main AUD issues to be: a) capturing, communicating, keeping track of and quality assure requirements from stakeholders and users in the process towards developing a final solution, b) balance time spent on user-involved activities with development activities. The article discusses the challenges and the need for more research on AUD methodology.

Keywords: Agile, User-Centered, Universal Design, ICT-projects, Best-Practice

1 Introduction

The focus on universal design (UD) has increased steadily over the last decades. In Norway UD regulations enacted in July 2014 (BLD 2017; KMD, 2013), state all new ICT-solutions (including most apps) targeted to the public must adhere to a minimum accessibility level. As of 2021, all new ICT-solutions must also be universally designed. As a result, companies have been required to alter their practices, integrating UD in ICT design and development. There is ongoing industry and research focus on integrating user-centered design (UCD) approaches into agile development – suggesting user-centered agile (UCA) approaches (Miller, 2005; Sy, 2007; Beyer, 2010; Silva da Silva, Martin, Maurer & Silviera, 2011). However less focus is granted to extending the user-centered focus to UD within an agile framework – exploring AUD (agile UD). Agile and agile-like development approaches are currently common in ICT-

projects. This article focuses on identifying challenges and current practices for AUD, in order to pave the way for future research. Our overall research question is: *What are key challenges for ensuring UD in UCA projects?*

2 Background

The word «agile» is often applied to a development process which follows a certain set of practices, usually including face to face communication, iterative feedback loops and incremental delivery of software. Agile developments have a set of beliefs which underline such practices, focusing on achieving efficiency and reduced waste (Preece, Sharp & Rogers, 2015). Scrum and Extreme Programming (XP) are two of the most popular agile models, where software is delivered after 1-4 week long «Sprint» increments (Scrum Alliance, 2013).

Human-centered (also called user-centered) design is defined as anchored in user needs, with user focus in all phases of design and development (ISO, 2010). Begnum and Thorkildsen (2017) indicate methodological differences agile versus non-agile projects implementing UCD. UCA projects have less focus on methods directed at understanding needs and contexts of use, and more focus on interface design, while non-agile UCD projects value higher levels of user involvement earlier. UCA projects tend to prioritize implementing features over early user-involvement and understanding (Silva da Silva, Martin, Maurer & Silviera, 2011). This can result in processes which to a lesser degree consider user needs, and where direct user contact only occur in the evaluation phase. UCD teams often directly involve users and stakeholders – workshops are frequent in design and insight phases – and use a larger variety of methods (Begnum & Thorkildsen, 2017).

The 2005 Disability Act defines UD of ICT as the design of any services or systems created through an electronics-based process so that they may be used, accessed and understood to the greatest extent (NDA, 2017). The Norwegian Agency for Public Management and eGovernment (DIFI) defines the minimum criteria to be AA conformance of WCAG 2.0, with a few guideline exceptions (DIFI, 2017). Beyond adhering to regulations and requirements, the focus of UD of ICT is on achieving usable and accessible solutions. Harder & Begnum (2016) conducted an interview study with designers and developers on projects having recognized success with UD in ICT. Factors that promote and obstruct UD were identified, mainly related to anchoring an understanding of and culture for UD on organizational levels merging UX and UD work and having the time for these activities, early and iterative quality assurance (QA) and user testing, and team collaboration. Nine of the thirteen projects followed a fully agile development, while two implemented agile elements into existing processes.

3 Research Approach

A scoping review is undertaken to explore and refine the research question**.** A scoping review allows the researcher to form and synthesize current knowledge on a specific topic. The goal is to provide the opportunity to identifying gaps in current knowledge and pave the way for future research; using the scoping review as a starting point for a larger research effort (Jesson, Matheson & Lacey, 2011). Unlike traditional literature reviews, it doesn't necessary rely on the newest published research. Instead the search may be based on two or three key articles to provide a set of theories. This scoping review is based on Begnum & Thorkildsen

(2017) and Harder and Begnum (2016). Both articles indicate that anchoring a user focus in early in the project affect UD and UCD. Collaboration between designers and developers also seems key, as communication influences efficiency of user research work and strengthens a common focus. Based on Begnum and Harder (2016), the assumption is that successful UD requires high-contact user-centeredness, i.e. methods that involve users directly. As Begnum and Thorkildsen (2017) found indications that UCA processes may be «less» user-centered than non-agile user-centered processes, it is an interesting perspective to evaluate the «user-centeredness» of AUD. Based on the analysis of previous work, the following sub-questions guided the scoping review: Which practices emerge to ensure UD in agile ICT-projects? How does AUD practice compare to identified promoting factors for UD success in ICT-projects?

3.1 Searching, Screening and Analyzing Literature

Agile methodology search terms are derived from Begnum and Thorkildsen (2017) who suggest that “agile” covers “lean”, “scrum” and “extreme programming”. Further, “sprint” is considered a central part of any agile process and included as search term. “Universal Usability”, “Inclusive Design”, “Design for All”, “User-Sensitive Inclusive Design” and “Ability-Based Design” are all overlapping terms for UD (Harder & Begnum, 2016). We chose to focus on actively and broadly used terms, thus omitting “universal usability”, “user-sensitive inclusive design” and “ability-based design”. Thus, our initial search string was: (scrum OR "extreme programming" OR sprint OR agile OR lean) AND ("universal design"OR "inclusive design" OR "design-for-all").

Table 1. Final Search Results.

Database	Search String	Returned	Included
IEEE	("agile development" OR "agile methodology" OR "agile process" OR scrum OR "extreme programming" OR sprint) AND ("universal design" OR "inclusive design" OR "design for all")	21	3
Springer-Link	"universal design" OR "inclusive design" OR design+for+all OR e-inclusion OR disability OR impairment OR accessibility AND "agile development" OR "agile methodology" OR "agile process" OR scrum OR "extreme programming" OR sprint NOT medicine OR obesity OR "body composition" OR geriatric OR cardiology OR "lean mass" AND “computer science”	101	8
ACM	"universal design" OR "inclusive design" OR design+for+all OR e-inclusion OR disability OR impairment OR accessibility AND "agile development" OR "agile methodology" OR "agile process" OR scrum OR "extreme programming" OR sprint OR lean NOT medicine OR obesity OR "body composition" OR geriatric OR cardiology OR "lean mass"	42	3
Total		**191**	**14**

Oria is a cross-database search which was our starting point to identify which databases should be included, consequently identifying ACM, IEEE and Springer-Link as relevant.

Individual search returned 1 result from ACM, 165 from Springer-Link and 124 from IEEE. However, iteratively adapting the search to the three different databases yielded more precise results, reviewing keywords used in relevant articles returned. For all three databases, the revision (agile OR lean) to ("agile development" OR "agile methodology" OR "agile process") yielded better results. In Springer-Link, the search term “computer science” was added to narrow its broad range of topic, whereas ACM digital library tends to yield very specific results. Both needed a broader set of terms to cover UD; “e-inclusion”, “disability”, “impairment” and” accessibility”. Further, Springer-Link and ACM limit the number of irrelevant medical results through exclusion terms “Obesity”, “body composition”, “geriatric”, “cardiology” and “lean mass”. For IEEE these search terms were distracting Final searches returned 191 results, see Table 1.

The goal was to select 10 to 15 peer-reviewed articles. A combined focus on UD and agile is required for inclusion. Some form of discussion of both topics together, directly or indirectly, was considered fitting to ensure relevance. In addition, UCD was an inclusion criterion, with involvement of users as per the ISO–standard (ISO, 2010). The articles had to focus on how to include marginalized users or ensure UD. They may target a single group, i.e. people with hearing impairments, as long as findings can be generalized to other groups. 14 articles were included. They are read using the SQ3R approach; a survey, question-based and focused re-reading approach (Jesson, Matheson & Lacey, 2011). We use an open and interpretative analysis approach to iteratively summarize and form emergent theories on topics within AUD.

4 Findings

Seven issues emerge as important challenges to solve in order to ensure UD in agile projects.

4.1 Requirements are hard to elicit

Involving users with severe disabilities introduces added challenges with regards to needs elicitation and collaborative communication. Guerrero-García et.al. (2017) suggest artifacts and metaphors can be helpful to elicit needs, such as capturing project vision and persona on worksheets to display so that they are constantly visible. This helps the team focus design on the variety of different users with different needs, providing functionality specific to users of different abilities. Together with scenarios they help communicate needs of users that cannot be present in a cycle or phase (Gkatzidou, Pearson, Green & Perrin, 2011). Among stakeholders scenarios and personas may be particularly useful to elicit requirements and user needs, as these can help contextualize the problems. User and task-focused representations are preferred to traditional software developments such as use cases (Prior et. al. 2013). In some cases, a user advocate can improve communication between users and agile team (Gkatzidou, Pearson, Green & Perrin, 2011; Prior, Waller, Black & Kroll, 2013). Needs can be elicited from experts if users are unavailable.. When eliciting needs with experts it is important to validate these needs later with target users (Røssvoll & Fuglerud, 2013).

4.2 Insights are hard to keep track of

As requirements emerge they must hold a manageable form, and a challenge is keeping track of insights iteratively gathered from user-centered activities during the agile process; especially insights related to needs and context of use (Guerrero-García, González-Calleros &

González, 2017). Agile prefers "working software" to "comprehensive documentation" (Agilemanifesto.org 2001), and advocates documentation should be kept to a minimum. Modifying the agile rules too much can create production blocks. But as good communication is found to be a prerequisite for successful UD, especially in communication between developers and designers, there is a need for certain documentation to be present. A common language among users, stakeholders and team members is presented as a prerequisite for eliciting needs and co-design in agile development (Raike et. al., 2008; Memmel, Reiterer & Holzinger, 2007). Use of UCD techniques combined with efforts of information and documentation sharing is promoted. Several propose to include experts and other stakeholders in user-centered work to assure data collected from end-users are not missed (Guerrero-García, González-Calleros & González, 2017; Røssvoll & Fuglerud, 2013; Gonzalez et. al., 2013). Røssvoll & Fuglerud suggest gathering requirements in one document make them easier to manage (2013). Some researchers suggest additional sprints or time dedicated to work on documentation (Williams et. al., 2015; Guerrero-García, González-Calleros & González, 2017). In addition to being discussion pieces, hi-fi prototypes can thus save valuable resources in the team (Memmel, Reiterer & Holzinger, 2007).

4.3 Limited User Requirement Oversight

The cycles and iterations of an agile process compensate for limited early insights as opposed to more traditional waterfall-like development models (Kaneyama, Goto & Nishino, 2015). As agile processes are adaptive to changing requirements, they are viewed as well suited to UD and collaboration with users (Williams et. al., 2015; Raike et. al., 2008). Nonetheless, one of the main problems addressed is how changing requirements affect the development process of inclusive systems. It is widely recognized that initial sprints should include methods to learn about users and contexts of use (Prior et. al., 2013, Kaneyama, Goto & Nishino, 2015; Scandurra, Holgersson, Lind & Myreteg, 2013; Guerrero-García, González-Calleros & González, 2017). Techniques such as observation of users' daily activities, document analysis and interviews with users and stakeholders are among those recommended. Further, as full up-front user requirement oversight is not likely, continuous user involvement and emergent requirement discovery should extend a shorter up-front requirements elicitation phase (Raike et. al., 2008). Reaching a common and correct understanding of needs is necessary to achieve accessibility and usability (Gonzalez et. al., 2013; Scandurra, Holgersson, Lind & Myreteg, 2013; Memmel, Reiterer & Holzinger, 2007). The need for continuous close collaboration with stakeholders, experts and (disabled) users seem to increase in AUD compared to UCA.

4.4 User Involvement Takes Time

User-involved approaches are widespread in the scoped literature, with stakeholders and users appearing as frequent collaborators for requirements elicitation and design. To integrating UD with agile development, user needs are identified prior to and during development, ensuring usable software is being developed (Memmel, Reiterer & Holzinger, 2007). User-involvement in certain activities is presented as imperative to success, such as evaluation (Scandurra, Holgerssob, Lind & Myreteg, 2013). However, it may be time-consuming and costly to do a user-centered project, and even more so focused on UD and involvement of marginalized user groups. Researchers imply agile processes don't inherently support UCD work, and that user involvement may delay or alter the agile process – necessitating an integrated approach (Gkatzidou et.al., 2011). Gkatzidou et.al. (2011) and Williams et.al. (2015) find that methods such as workshops result in a lot of design alternatives and design feedback, and working with

this data can slow down development. Bonacin, Baranauskas and Rodrigues (2009) highlight that adapting user-centered techniques to an agile schedule is challenging as developers struggle to balance tasks when also required to participate in non-coding tasks.

4.5 Quality Assurance Takes Time

Based on the elusive nature of requirements related to eliciting, keep track of and communicating, QA is also challenged. User needs must be evaluated continuously in inclusive design processes (Lucke & Castro, 2016). Failure is typically recognized later in the process when users evaluate the solution, resulting in added cost (Gkatzidou, Pearson, Green & Perrin, 2011). However, assessing accessibility early can reduce cost, which usually accumulates with late assessments (Reichling & Cherfi, 2013; Scandurra, Holgersson, Lind & Myreteg, 2013). It is recommended that end users test prototypes early and throughout the process (Røssvoll & Fuglerud, 2013). Williams et. al. (2015) suggest testing with at least hearing-impaired, visually impaired and cognitively impaired users. Traditional user tests take time to prepare and carry out. Testing in each cycle can add time delays (Røssvoll & Fuglerud, 2013). As user-involved QA takes time and effort, and frequent user-evaluations slows down the process, getting the time and money to adequate ensure UD seems a major AUD challenge. Efforts are thus made to adapt or develop new evaluation techniques tailored to agile processes (Memmel, Reiterer and Holzinger, 2007). Bonacin, Baranauskas and Rodrigues (2009) model evaluation workshops at the end of each development cycle. Williams et.al. (2015) run user-trials between sprints instead of traditional post-sprint meetings. Other researchers prefer informal expert assessments as means of evaluation reducing the need for user trials and detect major usability issues prior to testing with end-users (Kaneyama, Goto & Nishino, 2015; Røssvoll & Fuglerud, 2013; Gonzalez et. al., 2013).. However, expert evaluations must also be validated with user tests (Røssvoll & Fuglerud, 2013).

4.6 No AUD Process Model to Guide

Custom process models can be developed by teams with expert knowledge of development methodology, and adapted to specific design situations (Bonacin, Baranauskas & Rodrigues, 2009). However, as of today there is no general AUD process model available, and projects must design AUD development processes on their own. In Bonacin, Baranauskas and Rodrigues (2009) model, user involved design is in focus. Users can contribute with experiences and ideas for conceptualization and design, and later evaluate. Gkatzidou et.al. (2011) in their UIDM model (users, innovators, developers and modelers) ensure stakeholders are included in every step of planning, implementation and evaluation.

4.7 Lacking Team Effort Undermines Efforts

In order to achieve UD, a solution must have “usable accessibility” as well as “technical accessibility” (Reichling & Cherfi, 2013; Røssvoll & Fuglerud, 2013). To ensure inclusiveness and usability for all, it is important to shift from focus on implementing a quantity of features, to value UD and UX feature qualities. Having the team knowledge to achieve “technical accessibility” is further essential. It can be difficult for inexperienced developers to interpret guidelines on their own and translate them into action (Law & McKay, 2007). Further, negative attitudes among team members can undermine efforts to ensure UD. The team should ideally also have enough knowledge to educate customers on best practice.

5 Discussion

Research efforts that seek to achieve inclusive or UD of ICT solutions consider user-centered methods useful (Gonzalez, et.al., 2013; Bonacin, Baranauskas & Rodrigues, 2009; Røssvoll & Fuglerud, 2013). Begnum and Thorkildsen (2017) indicate that agile UCD processes are less user-centered than non-agile UCD processes, but this does not hold true for the literature-based AUD practices. Instead, the AUD processes largely favor collaborative and user-involved design methodologies. A lot of focus is given to user-involved methodology, including stakeholders, experts and end-users. The literature suggest a high degree of user-centeredness in agile processes ensure UD. These activities may be time-consuming but are believed to save time and cost later on. Research however implies that there are difficulties adapting these methods to the agile process. Issues arise when UCD work is required to be done in a timely fashion (Bonacin, Baranauskas & Rodrigues, 2009). This is particularly true for elicitation and design phases, where a proper understanding of user needs anchors the process. A general impression is that experts and stakeholders are perhaps involved as "stand-ins" to a much larger extent than what is needed in comparison to direct end-user focus.

There are also issues related to communication and documentation in all parts of the process. A poor user needs understanding can be a product of a lack of communication between team members, lack of triangulation of research methods or inclusion of experts or stakeholders. It is as such challenging to capture, communicate and quality assure requirements, ideas and insights from stakeholders and users with diverse abilities and disabilities. Comparing the discussion in scoped literature to Harder and Begnum (2016) highlights the need for knowledge among team members of agile processes and UCD techniques. Members need to understand UD values and the proper usability engineering methods to adapt these to development processes that fit the specific context and users.

Begnum and Thorkildsen (2017) confirm prototypes and scenarios are used to facilitate design discussions. A number of articles deal with how to avoid added cost of doing UD, usability and user-centred work in agile projects. However, there appears to be few attempts to adapt design methods to be more agile. There is more focus on adapting the agile processes to make room for the UCD methods and techniques within the cycles of development, and some attention is given to making user evaluation more efficient.

User testing every cycle is experienced as cumbersome and costly, while too much expert review without user feedback risks less usable and accessible solutions. Expert evaluation is a common method used in development cycles, and is considered efficient and timesaving. A general practice seems to be that experts and stakeholders test the solution before it reaches end-users, allowing the team to fix obvious usability and accessibility issues and as such save resources. It's recommended that evaluation work start as early as possible to avoid increasing cost. A common recommendation is having evaluation work run one sprint ahead of development work. There seems to be a need for more research on making quality assessment methods more fitting for AUD, for example increased re-usability of design artifacts.

When comparing best practice recommendations to Harder and Begnum (2016), we see similarities in the key factors promoting successful UD. Most notably is the emphasis on accessibility and UD from the very start and throughout the process, with the inclusion of external and internal experts. Also mutually recognized is the importance QA and

interdisciplinary cooperation based on a common understanding of UD. Including developers as part of user-testing first hand with disabled users is explicitly mentioned at least once, while most focused on including the entire team. While Harder and Begnum (2016) find that UD should be included in all phases, the AUD literature is less explicit about this and is focused on including user-centered and participatory methodology in all phases. What was not emerging from the AUD literature was the importance of an UD culture within organizations. This could be due to researchers working independently on developing solutions and not being part of an organization. However, the AUD literature is concerned with minimizing costs between UD and usability work. As such, the importance of ensuring adequate resources are allocated to UD efforts within a real-life project is recognized. This, as well as the need to elicitate real user needs from early phases, seems to be the driving force behind the importance of UD anchored on organizational levels.

Comparing Harder and Begnum (2016) with the AUD literature highlights AUD workflow issues. Some researchers note the agile process is not fully compatible with UCD methodology, calling for more knowledge on how to more efficiently employ UCD methods in agile processes in order to reduce cost (Røssvoll & Fuglerud, 2013). Harder and Begnum (2016) identify tentative promoting and obstructing factors critical to the success of UD. These findings required more work with regards to generalizability, and this also appears to be a common issue within the reviewed literature. Literature on the topic largely reveals prevalence of context-dependent frameworks and models with principles, guidelines and methods that seek to ensure UD of ICT solutions. Røssvoll & Fuglerud (2013) find best-practice recommendations should be adapted to individual projects with care and consideration to the specific situation. This is understandable, as proposed frameworks and models reviewed are adapted to specific problems for a certain user group, as such there is little generalizability.

When comparing our findings to Begnum and Thorkildsen (2017), literature points to a including a wider set of user needs and early, to a larger degree involve end-users in order and iteratively quality ensure UD in agile ICT development. Promotion of stakeholders and user involvement throughout the design and development process is deemed important to ensure UD quality, however how to integrate the recommended activities into the an agile process without delaying or adding to much extra cost is still not clear. Findings indicate the agile processes are suitable to emerging requirements and iterative design efforts, but that high levels of user-involvement and frequent quality control evaluations may be perceived as disruptive to the agile development process. In order to balance the agile rapid speed of development with the necessary degree of user-centered anchoring, trade-offs must be made. Real-life AUD process issues are relevant for future research efforts. There seems to be a large focus on how iterations and sprints can be altered to make room for the UCD work, especially in early and late phases of projects. Future AUD models could focus on a set of defined principles important to ensuring the goals, accessibility and usability, as well as agile principles. Such modeling may help ensure a common understanding of accessibility throughout the team as well as guide the process activities and workflow.

6 Conclusion

This article uses the term AUD to denote UD in agile ICT-projects. A scoping review was undertaken to provide an overview of current AUD challenges, to pave the way for future research. Seven AUD challenges are identified: 1) Requirements are hard to elicit, 2) User-

centered insights are hard to keep track of, 3) User requirement oversight is limited, 4) User-centered activities takes time, 5) Quality assurance takes time, 6) AUD process model to guide development is lacking, and 7) Lacking team collaboration undermines UD efforts. Synthesizing the findings, we find key AUD challenges related to a) capturing, communicating, keeping track of and quality assure requirements from stakeholders and users as part of the agile development process, and b) balance time spent on user-involved activities with development activities.. Further AUD research should focus on strategies for continuous needs elicitation and QA, creating a general AUD process model and provide guidance on how and when to merge user-involvement into agile development with minimal team disruption.

References

Begnum, M. E. N. & Thorkildsen, T. (2015) *Comparing User-Centered Practices in Agile Versus Non-Agile Development.* NOKOBIT Norsk konferanse for organisasjoners bruk av informasjonsteknologi. *23*(1).

Beyer, H. (2010). User-Centered Agile Methods In: Carrol, J.M. (ed.) *Synthesis Lectures on Human-Centered Informatics. 3*(1), 1-71. Morgan & Claypool Publishers. doi: 10.2200/S00286ED1V01Y201002HCI010

BLD (Ministry of Children, Equality and Social Inclusion). (2017). *LOV-2017-12-19-115 Universell utforming i diskriminerings- og tilgjengelighetsloven*. Regjeringen.no

Bonacin R., Baranauskas M.C.C. & Rodrigues M.A. (2009). *An Agile Process Model for Inclusive Software Development.* In: Filipe J., Cordeiro J. (eds) Enterprise Information Systems. ICEIS 2009. Lecture Notes in Business Information Processing, *24*.

DIFI (2017) *Kva seier forskrifta?*. Retrieved from: https://uu.difi.no/krav-og-regelverk/kva-seier-forskrifta (Accessed 26. November 2017)

KMD (Ministry of Local Government and Modernisation). (2017) *FOR-2017-09-13-1417 Forskrift om universell utforming av informasjons- og kommunikasjonsteknologiske (IKR)-løsninger.* Lovdata.

Gkatzidou, V., Pearson, E., Green, S. And Perrin, F. (2011). *Widgets to support disabled learners: a challenge to participatory inclusive design.* OzCHI '11 Proceedings of the 23rd Australian Computer-Human Interaction Conference. 130-139.

Gonzalez, C., Toledo, P., Muñoz, V., Noda, M., Bruno, A., and Moreno, L. (2013). *Inclusive educational software design with agile approach.* ACM International Conference Proceeding Series. 149–155. doi: 10.1145/2536536.2536559

Guerrero-García J., González-Calleros J.M., González C. (2017) FlowagileXML: An HCI-Agile Methodology to Develop Interactive Systems for Children with Disabilities. In: Guerrero-Garcia J., González-Calleros J., Muñoz-Arteaga J., Collazos C. (eds) *HCI for Children with Disabilities.* Human–Computer Interaction Series. Springer, Cham

Harder & Begnum (2016). *Promoting and obstructing factors for successful universal design of ICT.* NOKOBIT Norsk konferanse for organisasjoners bruk av informasjonsteknologi. *24*(1).

ISO, International Organization for Standardization. (2010). *ISO 9241-210:2010 Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems*. Retrieved from: www.iso.org

Jesson, J., Matheson, L., and Lacey, F. (2011). Doing Your Literature Review: Traditional and Systematic Techniques. SAGE Publications Ltd. ISBN: 9781848601543

Kaneyama, T., Goto, T., Nishino, T. (2015). *Methodology for developing ICT based course material for children with a developmental disability based on EPISODE*. IEEE 13th

International Conference on Industrial Informatics (INDIN). doi: 10.1109/INDIN.2015.7281982

Law C. M., McKay E. (2007) Taking Account of the Needs of Software Developers/Programmers in Universal Access Evaluations. In: Stephanidis C. (eds) *Universal Access in Human Computer Interaction*. Coping with Diversity. UAHCI 2007. LNCS, *4554*, 420-427. doi: 10.1007/978-3-540-73279-2_47

Lucke, U. and Castro, T. (2016) *The Process of Inclusive Design*. 2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT), 446-447.

Memmel T., Reiterer H., Holzinger A. (2007) Agile Methods and Visual Specification in Software Development: A Chance to Ensure Universal Access. In: Stephanidis C. (eds) *Universal Access in Human Computer Interaction.* Coping with Diversity. UAHCI 2007. LNCS, *4554*, 453-462. doi: 10.1007/978-3-540-73279-2_51

Miller, L. (2005). *Case study of costumer input for a successful product*. Proceedings of Agile Conference (AGILE), 225-234. Denver, Colorado, USA: IEEE Computer Society.

NDA (2017) *What is Universal Design*. Retrieved from: http://universaldesign.ie/What-is-Universal-Design/Definition-and-Overview/

Preece, J., Sharp, H. and Rogers, Y. (2015). Interaction design. 4th ed. Chichester: Wiley.

Prior, S., Waller, A., Black, R. And Kroll, T. (2013). *Use of an agile bridge in the development of assistive technology.* CHI '13 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 1579-1588.

Raike A., Saad-Sulonen J., Scheible J., Suzi R., Toikkanen T. (2008) *Visual Tools for Accessible Computer Supported Collaboration*. In: Miesenberger K., Klaus J., Zagler W., Karshmer A. (eds) Computers Helping People with Special Needs. ICCHP 2008. LNCS, *5105*. Springer, Berlin, Heidelberg.

Reichling, M. & Cherfi, S. S. (2013) *Integrating Accessibility as a quality property in web developments*. IEEE 7th International Conference on Research Challenges in Information Science (RCIS). doi: 10.1109/RCIS.2013.6577698

Røssvoll, T. H. & Fuglerud K. S. (2013) *Best Practice for Efficient Development of Inclusive ICT.* In: Universal Access in Human-Computer Interaction. Design Methods, Tools, and Interaction Techniques for eInclusion: 7th International Conference, UAHCI 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, Part I., 97-106.

Scandurra, I., Holgersson, J., Lind, T. & Myreteg, G. (2013). *Development of Novel eHealth Services for Citizen Use – Current System Engineering vs. Best Practice in HCI*. In: Kotzé P., Marsden G., Lindgaard G., Wesson J., Winckler M. (eds) Human-Computer Interaction – INTERACT 2013. LNCS *8118*, Part II, 372–379. doi: 10.1007/978-3-642-40480-1_24

Silva da Silva, T., Martin, A., Maurer, F., Silviera, M. (2011). *User-Centered Design and Agile Methods: A Systematic Review*. Proceedings of Agile Conference (AGILE), 77-86. Salt Lake City, UT, US: IEEE Computer Society.

Scrum Alliance. (2013) *We iterate... So Are We Agile?* Retrieved from: http://www.scrumalliance.org/community/articles/2013/march/we-iterate-so-we-are-agile

Sy, D. (2007). Adapting usability investigations for agile user-centered design. *J. Usability Studies, 2*(3), 112-132.

Williams D., Wang MT., Chang CH., Ahamed S.I., Chu W. (2015) *Agile Development for the Creation of Proper Human-Computer Interfaces for the Elderly.* In: Bodine C., Helal S., Gu T., Mokhtari M. (eds) Smart Homes and Health Telematics. ICOST 2014. LNCS, *8456,* 29-38. doi: 10.1007/978-3-319-14424-5_4