

# Introducing Usability Engineering and Universal Design in the Computer Science Curriculum

## Authors:

Kirsten Ribu, Oslo University College, Postboks 4, St. Olavs plass, 0130 Oslo,  
kirsten.ribu@iu.hio.no Miriam Nes, Avenir, Miniam.nes@avenir.com

**Abstract**— *Software and information systems are usually developed for a stereotypical user: Young, white, male and non-disabled. In order to understand and include a variety of user groups, universal design and usability engineering must become part of the software engineering curriculum. Learning to develop systems for diverse user groups means that students must acquire knowledge of the relations between humans and technological solutions, as well as the requirements of disabled users. In order to communicate the importance and meaning of accessibility and usability, classroom exercises like using a computer with one hand, surfing the Internet wearing sunglasses or glasses smeared with oil or grease, and distorted computer screens can illustrate the challenges of designing software solutions for all users. We describe how students can be made aware of the social importance of software engineering, and conscious of professional and democratic responsibilities. The paper outlines suggestive curriculum, defining the major disabled user groups and their accessibility and usability requirements, and describes how to approach teaching universal design and usability. At Oslo University College, a new bachelor program called 'Applied Computer Technology' was initiated in 2005. The main focus of this program is the practical use of information systems in specific contexts, and the principles and ideas outlined in this paper will be incorporated into the third year course 'Modern Methods for Systems Development'. The students will work on real projects, developing solutions for real users. The work will involve user participation, iterative development, and the HCI principles learnability, robustness and flexibility. Usability testing is an important part of the development process. In co-operation with organizations for the disabled, project groups consisting of 4-5 students will develop solutions for future users like people with reduced motorical skills, Parkinson's disease, as well as dyslectic, blind and visually impaired. The course starts in August 2007. Results will be reported in a follow-up paper in next year's engineering conference.*

**Index Terms**— *accessibility, disabled users, diversity, universal design, usability.*

## INTRODUCTION

Of all the millions of Internet sites only a few web pages are designed to be universally accessible. Any piece of software, however complex or simple, demands instruction in how to use it. However, such instruction is seldom available, and should ideally be superfluous. The frustrations of the average computer users are great, and the problems for disabled users like the blind, dyslectics and the motorically disabled are enormous. Modern technology and computer software is developed with specific users and contexts in mind, excluding others. Programmers and computer engineers dominate the field, and the training and attitude of these cultures result in hard-to-use technology, creating a digital apartheid, or digital divide, in the population [1].

Software systems are not constructed in order to suit users with special needs. Interaction design has been considered a programming task, although programmers do not have the training necessary to solve the problems posed by advanced interface and interaction design. In Norway, there is a government policy that all official web-sites containing information for the public must be accessible for all. Constructing information barriers is considered unethical, and inclusion of the disabled is now being integrated into the law as part of the anti-discrimination laws, defining discrimination of the disabled illegal in the same way as discriminating against people on the grounds of religion, ethnicity, gender or age. The Directorate for Health and Social Affairs [2] has a National Resource Centre for Participation and Accessibility for persons with disabilities in Norway, the Delta Centre. 'The work of The Delta Centre concerns how the use of assistive technology and adaptation of the environment together yields a better standard of living for persons with disabilities. Our vision is participation and

accessibility for all. The objective is that disabled people have the same opportunities as others to participate in an active way in society.' [3].

The principles of usability engineering and universal design must therefore become part of the curriculum. At Oslo University College, there is a new bachelor program called 'Applied Computer Technology' which is not a traditional engineering education program, but focuses on the user perspective of computer systems. The program teaches the basics of information technology like programming, systems development, databases and networking paired with social sciences, project management and human computer interaction. Interaction design and universal design are not primarily technical challenges, but an approach to understanding system behavior and information, and how this should be presented to the user. Human-computer interaction design, user experience, user satisfaction and cognitive abilities are important ingredients in the courses taught on this program. Usability testing will be established as a principle in all systems development and programming courses at the Faculty of Engineering in the next few years.

## **TECHNOLOGY DEVELOPMENT IN THE INFORMATION AGE**

The engineering approach to software development is rooted in the industrial age, based on a mechanical world view, which regards the world as a machine [4]. Instead of considering the realities of the information age where the central aspects are not tangible products, but information and communication, software is viewed as a traditional product, created in the same way as industrial products. However, when developing software, the product is invisible until it is completed. Therefore, finished software is often unsuitable or difficult to use because it is the result of the work of programmers, which includes defining user requirements, database logic, system behavior, and making technological decisions. If usability testing is conducted at all, it usually happens when the software is completed. Programmers use themselves as references, and the software they construct is often made for other software engineers: male, young and physically fit computer experts. However, the information age places other demands on the construction of software than the traditional line of production.

The mechanical world view has resulted in a *hard system* development approach. The traditional waterfall development model belongs to this view and is still very much alive, although modern approaches to systems development like agile methods ensure better reliability, usefulness and correctness of the systems built. The mechanical world view is still unquestioned in both industry and education. In this perspective, including universal design and human-computer interaction principles is perceived as undesirable because it adds costs and slows down the development process. Delivering the product on time, however badly designed, is considered to be a criterion of success.

The solution to this unhappy state of affairs is to attack the question of universal design already in the planning stage [1]. User participation, involving professional interaction designers in the planning phase, designing before programming, and conducting usability testing at all stages of development are sound practices that should be integrated into the production process.

## **UNIVERSAL DESIGN, ACCESSIBILITY AND USABILITY**

The goal of the universal design strategy is to create enabling contexts for all users. Universal design means designing products and environments for all to the greatest extent possible, without the need for adaptation or specialized design [5]. It is an approach to developing and designing technological products from a distinct perspective, namely, one of respecting and valuing the diversity in human capabilities, technological environments and contexts of use. Products, environments and information should be created in order to be useful to as many people as possible. A product that is universally designed is well designed because attendance is given to the needs of diverse users throughout the entire design and construction process. Planning the user interaction from the start with the needs of disabled users in mind will place focus on software as a designed whole, useful and user friendly. In the outcome, all users will benefit from this approach.

Modern technical solutions always involve software. Software development is complex, involving a lot of work, time and money, and though the solutions may actually be accessible to all, they are often not useful. This important distinction is often overlooked. The term universal design places emphasis on both universal accessibility and universal usability. Universally designed therefore also means well designed, ie. useful for the users in question. Universal design in relation to software could be defined as referring to the *core functionality* of a product, and not necessarily to all extensions. This does not mean that total accessibility and usability adjustments will be included, but that the goal is to design the main functionality of a product for the broadest possible range of users [6]. Adaptations may complement the design. Determining whether a product is universally designed means determining if it is useful for the user group it is designed for.

## **DESIGNING THE COURSE 'MODERN METHODS IN SYSTEMS DEVELOPMENT'**

### **Course Philosophy**

Changing the design and programming culture in industry also means changing the approach to teaching software development in universities and colleges. Designing a new course curriculum means studying and understanding the needs for that course, and planning with specified goals and outcomes in mind. The upcoming anti-discrimination law in Norway against the disabled is the underlying motivation for designing this course, as well as the demographic changes in society, with an increasing elderly population. Industry needs employees, and many disabled wish to work instead of becoming welfare clients. Improved technology and better software solutions for all can make this possible without great cost or effort if inclusion of diverse user needs is planned from the beginning of the development process. Software developers must be prepared to design all computer systems according to universal design principles, and students should therefore learn to design universally accessible and usable products and computer systems early in their studies. This will establish universal design as an overall principle and standard in education and industry. This is the main motivating factor behind a new course in the curriculum, called 'Modern Methods for Systems Development'.

Many problems regarding user interface design are caused by the ambiguous word 'design'. From a software developer's perspective, design means 'program design', ie. code, and maybe UML class diagrams. Graphical user interface design is regarded as a 'finish' or 'layer' on top of the program, often conducted by graphic designers and visual artists. However, *interaction design* is not so much the graphic design of an interface, as the understanding and implementation of the workings of the communication between user and software, or technological solution involving software. In human-computer interaction, communication may be visual, audio or tactile. In order to understand the challenges of creating solutions for diverse user groups, software developing teams must study working environments and work routines *in situ*, using an anthropological approach. This means interviewing and observing people in their environments to try to understand how they think and behave, then identify their needs and create the processes to design appropriate solutions [1]. Interaction designers must be included already in the planning stage, and not be given the task of slapping a visually pleasing but user unfriendly interface on top of a finished software solution.

### **Course Goals**

Students should acquire a professional attitude towards developing systems for all users. The demographic situation in the western world is changing, and the population is growing older. Many people who today are confident technology users may soon find themselves disabled in various every-day situations where technology is involved. In this course, the students shall become conscious of the needs of diverse users like the print-disabled; meaning blind, visually impaired, elderly and dyslectic, as well as the motorically challenged, like people with Parkinson's disease or traffic injured. The philosophy is that when developing for diverse user groups with specific needs, focus will shift from the developer's fixation on code and technical solutions to the design of useful systems and interfaces, solutions that are suited to all users. Focus is on accessible technology in a democratic society, also on the role and responsibilities of the expert, and ethical and legal aspects of developing software systems. The goal is to enhance students' independent thought, social awareness, empathy and understanding of the needs of diverse users.

The students on the program 'Applied Computer Technology' are not primarily programmers, and their interests and abilities lie in the direction of management and human-computer interaction design. 'Modern Methods for Systems Development' is a third year course, and the students have already had basic courses in systems development and databases, and are familiar with UML modeling and various process models for software development. They have completed courses in HCI, organization theory, social aspects of technology and systems, project management and general programming. There is no need for yet another course in software development and design, presenting the philosophy and workings of RUP and other industry software development processes, with practical exercises in constructing working systems with databases in a traditional manner. Instead, the course will seek a new approach to teaching systems development, by introducing the principles of universal design from the first day, and emphasising the importance of early planning, including experts in human-computer design in the process. Every bit of work the students conduct on this course must be done with the philosophy of accessibility and usability in mind, in designing everything from the simplest web-page to the most complex software solution.

### **Learning Methods**

Normally in the education system, the scientific methods and approaches to understanding the part played by technology in society are seldom discussed. A mechanical world view and hard systems thinking have dominated. A different approach, *soft* systems development, focuses on co-operation between stakeholders and various user groups and believes in mutual learning and understanding between groups of people.

The first challenge is to convey to young, healthy and technologically talented students what it means to be disabled in a society where technology invades every aspect of every-day life. Many young people and especially students of technology are so familiar with technological devices that they often fail to recognize the problems other people are faced with. In order to acquire an understanding of what it means to be disabled, for instance dyslectic or visually impaired, demonstrations are

necessary. Letting the students surf the Internet wearing dark glasses, or without the use of a mouse can illustrate what using a computer is like for the visually impaired. By physically experiencing the difficulties even minor disabilities may lead to, young people achieve an idea of the challenges they are faced with when creating technical solutions for the disabled.

Next, the students should learn to identify and describe the needs of specific users. The project groups will conduct *in situ* studies of work performed at workplaces, and interviews with the people who perform the work tasks using more or less suitable technology. The actual student work will be project-based. The students will work in groups of 4-5. They will from the beginning come into contact with users through collaborating partners in industry and organisations for the disabled, who welcome this opportunity to work with students on a professional basis. Users will be involved in the process from the start. User participation in the development process is an important factor in order for marginalized users to become socially and digitally included, and is an established technique for uncovering different user perspectives. UNESCO suggests mandatory participation of disabled users in political and industrial plans. When designing for disabled users, the students will conduct interviews with and observations of people with disabilities, and acquire a deeper understanding of specific user needs through planning, designing and prototyping solutions in constant collaboration with the users.

### **Course Topics**

The course will include lectures on general topics and principles, supplemented with guest lectures by experts. The students will not code working systems in a traditional manner, but focus on creating prototypes. Lectures on the following topics will be accompanied by practical exercises:

- Project work
- Prototyping techniques
- Interview techniques, in situ studies and observation
- Planning and modeling techniques
- Practical project work with supervision and continuous feed-back from teachers, users and employers
- Presentations technique
- Usability testing with real users
- Cost estimation
- System evaluation using feature analysis
- Principles of universal design

### **Pedagogical Considerations and Techniques**

A course of this kind places large demands on the instructor. The work load and responsibility can be eased by involving the students actively in the learning process. Research shows that students benefit from collaborating with and receiving instruction from their peers [11]. When students teach each other, they have a high sense of responsibility. Since the bachelor program 'Applied Computer Technology' also aims at educating future project managers and leaders as well as aiming at teaching good communication with users, practicing presentation and communication techniques is an important part of the program.

The authors of this paper have extensive previous experience with student project work, and have acknowledged the importance of preparing the students for problems that will arise. The students must sign a group contract before work begins, and are encouraged to handle conflicts within the group in early stages of the conflict. By agreeing beforehand on how to react to for instance to negligence of work, failing to attend group meetings or complete deliverances, there is a better chance of handling problems when they emerge and prevent project delay or collapse. Previous experience in other courses has shown that student treat each other fairly, and when well prepared, act according to the contracts they have agreed on. In difficult cases they consult the instructor, but they are encouraged to handle the situations themselves. The instructor will take action only in extreme cases, for instance when having to exclude a student from a group or from the course.

### **Practical Project Work**

The students are familiar with the basic principles of universal design, and these skills must now be put to practice under expert guidance by an interaction designer. Project work will be conducted in collaboration with real users with various disabilities, mainly users with motorical and visual impairment. The work will involve user participation, incremental development and the HCI principles learnability, robustness and flexibility. The course has about 60 students. Several project groups may work on the same problem, creating different solutions, so there is a need for approximately 8-10 project ideas.

The users will present an actual problem they need to be solved, and act as task employers. The students will interview and observe users and make descriptions of future systems using modeling techniques and prototypes. Often, disabled users have ideas about software solutions they think may be useful, and such ideas could turn into sensible design. For instance, one may observe a person heavily disabled by Parkinson's disease using a computer and a certain piece of software. Parkinson's disease leads to symptoms similar to those commonly experienced by elderly people: tremor, rigidity, slowness in movements and reaction, visual disturbances, dementia, pains, weakened voice, lack of energy and difficulties with memory and balance. By selecting the users with the most critical requirements, the principles of 'reversed universal design' are applied. When solving the most difficult design tasks first, one ensures that the solution is useful for the most heavily disabled. The solution is then further developed to suit all users. Traditional approach has been the other way round: To adapt finished solutions to the needs of various groups of disabled users. However, the principles of universal design state that one should not adapt, but design appropriate solutions for all. Using the principle of reversed universal design, developers uncover the problems together with the user, design prototypes with user participation and suggest technical devices that will ensure better usability of the solution for Parkinson patients in general. The same is done for other user groups with other disabilities. The challenge is to create usable solutions for all by analyzing the needs of diverse users. The goal is to achieve better design for all users. For instance, the largest group of users of talking books today are not blind, but ordinary people who prefer to listen to books rather than reading themselves.

### **Course Literature**

This is a completely new course on soft systems development and universal design, and to our knowledge, no suitable text books exist. There are vast numbers of books on systems development, systems design and management, agile development, HCI, interaction design etc., but no one book that is suitable for this course. So, in accordance with the principle of designing a tool to fit the needs of specific users, learning material is gathered and created specifically for the course. The material may in time become a text book. So far, it consists of a compendium containing a book chapter on 'Social, Legal and Ethical Aspects of Computer Systems' [12], guidelines for creating universally designed web pages supplied by the Delta Centre [3], links to the Data Inspectorate web pages [13], in addition to suitable extracts of text books on usability engineering, universal design, and software engineering. The principles of 'fair use' for education purposes secure copyright considerations.

### **Evaluation**

Evaluating project-based work poses several questions. There is no final written exam, the project reports with results and presentations and demonstrations of prototypes will be used in the evaluation process. The course instructor conducts talks with each group on a regular basis, where students report on their work, and receive guidance. The main object is to participate in a group and to learn by doing, and learning is better when feed-back is given regularly.

The project work is given grades from A to E. It is not possible to fail in this kind of course, unless no work is done, and when receiving guidance and course feed-back it is difficult to achieve a grade lower than C. It is fairly easy for an experienced instructor to detect if certain students are not doing the required work. Also, fellow students will complain, because they will be burdened with additional work. The group is given the same grade, except in cases where it is obvious that a certain student has not participated in the work. Project groups or individual students may - if considered necessary- be examined orally in order to establish the final grades.

### **COLLABORATING PARTNERS**

In order to conduct a course of this kind, it is important to have partners who can give guest lectures, provide information and support, and come up with ideas and specific tasks for student projects on a voluntary basis. There is a small budget for guest lecturers, but most of the partners contribute free of charge. The Delta Centre is an inspiring partner [3]. The Centre collaborates with several colleges in Norway, and is striving to inspire universities and colleges to include universal design in the engineering curriculum and establish these principles as a formal college policy. A representative of the Centre gives introductory guest lectures on topics like universal design and principles for constructing accessible web pages.

From the spring of 2007, the Research Council of Norway supplies funding for a new type of project in Norway, namely support for establishing resource networks. The company MediaLT [14] has created a network within information technology and universal design, and was one of the first companies in Norway to receive support for administrating a resource network. Several of the members work with disabled users, and will be happy to furnish the students with project tasks. Other partners are organisations for the disabled, for instance the Parkinson society and the Society of the Blind. The contact and collaboration with Parkinson patients is well established through the ongoing research project mentioned in the beginning of this paper. Student projects are welcome. The master thesis of Miriam Nes presents a study of a learning aid system for reading disabled; dyslectics and visually impaired, and uncovers the need for better play-back software [10]. Here

lie several potential student projects. Contact is also established with the Society for the Blind and MediaLT, who are ready to furnish projects.

## SUMMARY

Software is made by programmers who have little or no training in designing systems for diverse users. Software engineers construct solutions for themselves and other software engineers, and although they regard their work as design, it is in fact only program design. The results are badly designed systems, unsuitable for many users. Elderly and disabled users become digital outcasts when technology erects barriers. This is a serious democratic problem in modern society. Demands for universally designed solutions often meet with resistance in the software development community, and the blame is laid on the additional costs. However, including universal design principles into the planning stage, using the principles of human computer interaction, and involving interaction designers from the start will not only reduce costs, but greatly improve the products for all users, because universally designed products are well designed. In order to change the culture in industry, these ideas and principles must be taught in universities and colleges, and students must be included in discussions on the impact the choice of methodology has on both software development processes and the results for users. A new course, 'Modern methods for Systems Development' at Oslo University College, seeks to educate the students in soft systems thinking and principles of universal design, by providing the students with real tasks to solve. The work will be conducted in project groups designing solutions for diverse users with various and complex needs, in collaboration with real users. This third years course starts up in August 2007 with the aid of partners in industry, research and organisations for the disabled. The results will be presented in a follow-up paper at next year's icee conference.

## REFERENCES

- [1] Cooper, J. "The Inmates re Running the asylum", Google books.
- [2] URL <http://www.shdir.no/english>. Last accessed: 20.08.2007
- [3] URL <http://www.shdir.no/deltasenteret/english/> 27.04.2007
- [4] Hendrix, E. Permanent injustice: Rawl's theory of justice and the digital divide. *Educational Technology Society*, 8 (1), 63-68, 2005.
- [5] The Center for Universal Design, North Carolina State University, USA, URL <http://www.design.ncsu.edu:8120/cud/index.htm> Last accessed: 01.05.2007
- [6] Stephanidis, C, Akoumianakis, D, "Universal design: Towards universal access in the information society", *CHI 2001 Workshops*, 31 March – 5 April 2001
- [7] URL: <http://www.ieee.org>
- [8] Sommerville, I. *Software Engineering*. Addison Wesley. 7th edition.
- [9] Barbara Ann Kitchenham. Evaluating software engineering methods and tool, part 3: Selecting an appropriate evaluation method - ractical issues. *Software Engineering Notes*, vol 21, no 4, July 1996, p:9-12, 1996.
- [10] Nes, M. Appraising and Evaluating the use of Daisy. For Print/Disabled Students in Norwegian Primary and Secondasy Education. MSc Thesis, 2007, The University of Oslo.
- [11] J. Biggs, "Teaching for quality learning", SRHE & Open University Press, ISBN 0-335-20171-7, 1999
- [12] Fagernes, S., Ribu, K. Social, Legal and Ethical Aspects of Computer Systems. *Handbook of System Administration*. Elsevier. To be published 2007.
- [13] URL: <http://www.datatilsynet.no> english. Last accessed 20.08.2007
- [14] URL <http://www.mediaLT.no>. Last accessed 20.08.2007