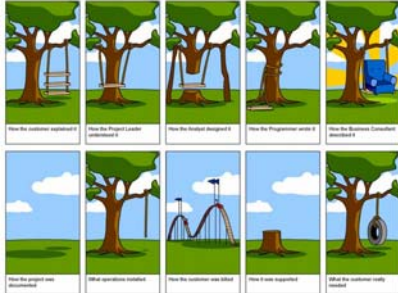


User-centered design

2009

Roman Bednarik



Organization of the course

- **175421 (5 ECTS)**
- **Lecturer – Roman Bednarik, bednarik@cs**
- **Lectures – Mo and Tu, 14-16, room 2d106, in English**
 - Introductory lectures
 - Student presentations
 - Invited speakers – Mirja Kälviäinen 20.1.2009, Carolina Islas Sedano 2.2.2009
 - Project consultations

Organization of the course

- **Demos**
 - Mo 12-14, Tu 12-14, room B180
 - Minnamari Naumanen, Tersia // Gowases
- **Course web page**
<http://cs.joensuu.fi/pages/bednarik/UCD2009/>
RSS channel

Organization of the course

- Grading

active participation

- Topic presentation 10%, compulsory
- Course project 36%
- A final exam 30%
- Demo tasks – X-tasks 24%, compulsory
- 100%
- Extra demo tasks - max 10%

- < 50% => fail, 50 – 59.9% => 1, 60 – 69.9% => 2, >90% => 5

Organization of the course

- Topic presentation 10%, compulsory
 - List of topics will be circulated
 - Schedule of presentations at the UCD2009 course web-page
 - Can be done alone or in pairs
 - Materials from Roman
 - Discussion in the class after presentation
- Course project
 - Distributed via www and in demos
 - Group work 2-3 persons

Organization of the course

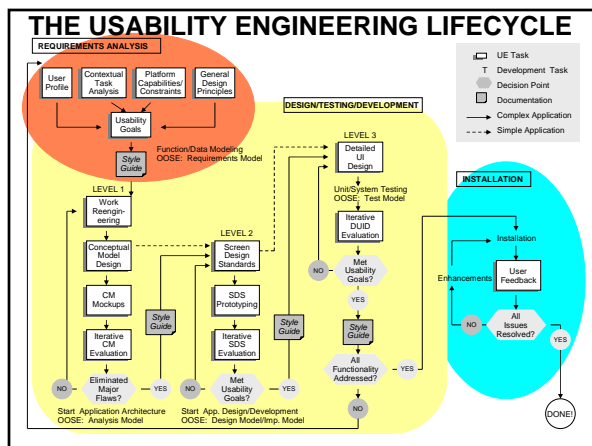
- X-tasks
 - Submit beforehand using email to the demo-teacher
 - Will be graded
 - Have ready in a paper version for demo session:
 - presentation and discussion is a part of X – task
 - Exceptions: if you cannot attend the X-demo agree with demo-teacher beforehand.

Assignment 1.

- Take a break
- Self-organize into 15 groups, now

Relation to other courses

- HCI / IVT (Ihminen ja vuorovaikutteinen teknologia)
 - Basic knowledge about principles of human-computer interaction
 - next time in spring 2009
- Designing graphical UI – implementation of UI from technological perspective
- UCD – User-Centered Design/ Käyttäjäkeskeinen suunnittelu
 - How to design interactive products with users
- UE – Usability engineering / Käytettävyystekniikat
 - How to evaluate usability and integrate its methods into the software development – next time in 2010



What is this course about?

- About how to design interactive products that consider
 - Who is going to be using them
 - How they are going to be used
 - Where they are going to be used

What is design

- A conversation between the designer and user
and between the desired outcomes and unwanted side effects
 - Beale, 2007

Designing software

- Traditional, engineering perspective
 - Software engineering
 - Programming, algorithms, data management, process management
 - Functionality
 - Software engineers similar to civil engineers
 - It is built technically in a proper way = it works, it's safe, it follows standards, no fatal errors, fulfils functional requirements
 - But where is human? Combination of both = UCD.

What is design

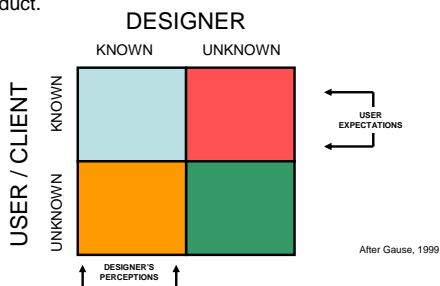
- A conversation between the designer and user
- Removing ambiguity
- Creating an opportunity
- Solving a problem – problem solving, often messy
- Searching an abstract state space for “good” solutions
- Defining the inside of a black box

Requirements

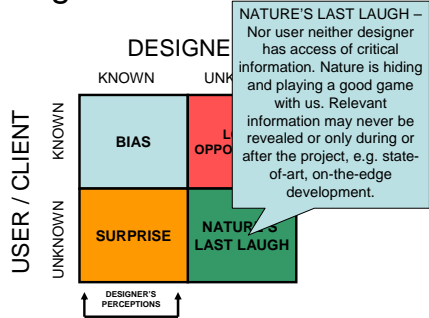
- Problem definition
- Assumptions
- Background and context capture
 - Social, physical, organizational, security, political, ...
- Users
- Risks
- PROBLEM => REQUIREMENTS => ????

Design window

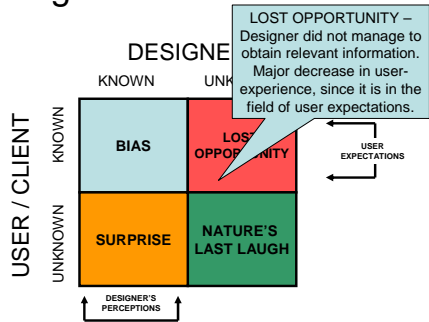
- Designer and User/Client
- **Designer** transforms the design requirements into the final product.



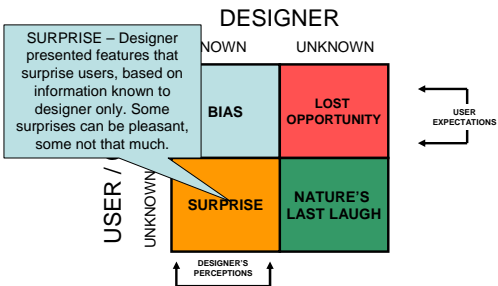
Design window - risks



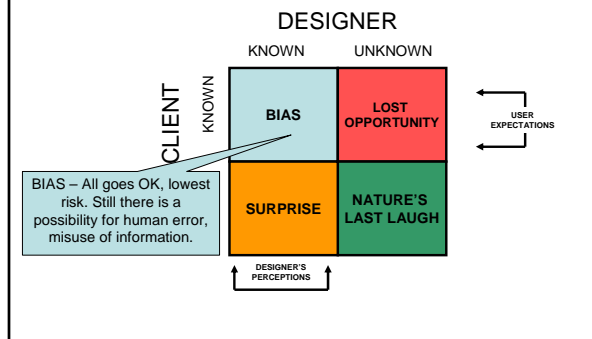
Design window - risks



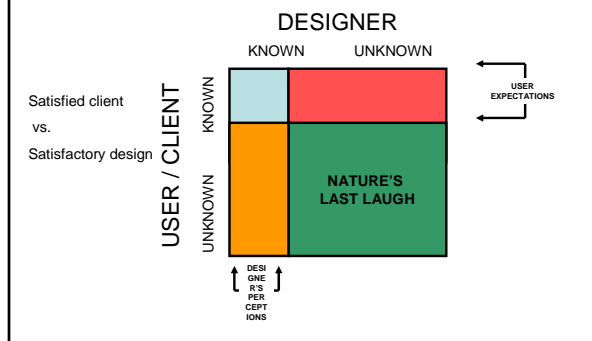
Design window - risks



Design window - risks



Design window – relative risks at the beginning of the project



More about design

- Design is a social activity – group-work, work with users and clients
- Design has consequences – on people's lives, designer has a responsibility
- Design is a creative process – envisioning solutions and surprises

Design errors = Usability – user experience problems

- Usability problems are seen as caused design errors.
- Errors can be corrected, if discovered, or can be avoided completely.

– Errors can be avoided, prevented - UCD
– Errors can be discovered - UE

Relative Cost to Fix an Error

DISCOVERY PHASE	COST RATIO
Requirements	1
Design	3 to 6
Coding	10
Development Testing	15 to 40
Acceptance Testing	30 to 70
Operation	40 to 1000

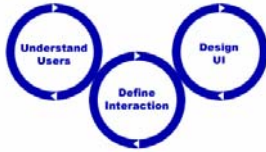
Based on 63 IBM, GTE, TRW projects. Barry W. Boehm, *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981

User-centered design

- UCD is a philosophy and a set of methods.
- UCD methods place the user in a critical/central position for both determining system requirements and ensuring they are achieved.

UCD at SAP

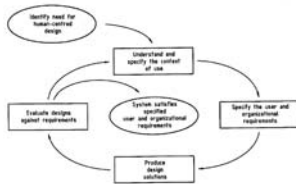
1. Understand Users
2. Define Interactions
3. Design UI



www.sapdesignguild.org

UCD in ISO 13407

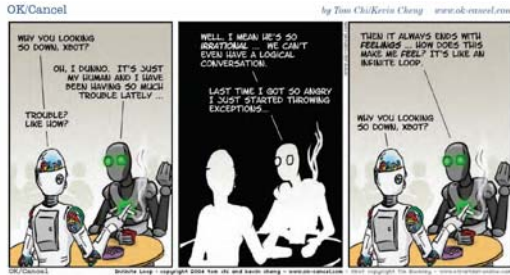
- Human-centred design is an approach to interactive system development that focuses specifically on making systems usable. It is a multi-disciplinary activity. (ISO 13407)



User-centered design

- A wide range of methods practitioners use.
- Goal of design is to avoid usability and user experience problems
- Matured enough to have standards – ISO 13407

Human vs. Machine



Human centered development vs. Machine-centered development (Norman)

People	Machines
Creative	Unoriginal
Compliant	Rigid
Attentive to change	Insensitive to change
Resourceful	Powerful, fast
Emotional	Unimaginative
Prone to err	Error-free
...

- Still too many products are made in a machine-centered, technology way => unusable, too many functions, no value to users.

Machines for people, or people for machines?

Design goals - User experience vs Usability

- Usability goals
- User experience goals

Usability

- "Usability: an 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."

ISO 9241

Usability goals

- Easy to learn
- Efficient to use
- Easy to remember
- Few errors
- Safe to use
- (Subjectively) pleasing - vague

User experience

- Satisfying
- Enjoyable
- Doing the right job
- Fun
- Helpful
- Apprehensive
- Motivating
- Rewarding
- Supporting creativity
- ... and many more

Design goals - User experience vs Usability

- Usability goals –
 - meeting specific criteria
- User experience goals –
 - quality of user experience in every day life applications

Challenges of UCD

- Access to relevant information and users
- Applicability of methods
- Assumptions
- Attitudes
- Conflicting needs of different groups
- Conflicting needs of inclusiveness and equity with efficiency and quality => trade-offs
- Cooperative user – motivation
- Organizational flexibility
- ...

- Participation of people in the **design** of artifacts, services, institutions, technology, media and social movements is key to creating futures that are effective and equitable. (Schuler, 2004).

Gould and Lewis (1985)!

- We recommend that **typical users** (e.g., bank tellers) be used, as opposed to a “group of a variety of experts” (e.g., supervisors, industrial engineers, and programmers). We recommend that these potential users become part of the design team from the very outset when their perspectives can have the most influence, rather than using them post hoc as part of an “analysis team (of) end user representatives.”

Origins of PD

- Suom: osallistava suunnittelu (osallistaa, tehdä osalliseksi = to (make to) participate)
- Began in a political context: Scandinavian workplace democracy movement (1960's – 1970's)
 - University researchers performed experiments in alliance with organized labor

Origins of PD

- Draws upon many fields
 - Graphic design, software engineering, architecture, public policy, psychology, anthropology, sociology, labor studies, political science

What is participation?

- A social process of
 - involving users in the design and development of products – urban areas, software, furniture, shoes, ...
- Encouraging ownership of the community

What is participation in software development?

- A social process, too!
- E.g. by using prototypes with real users to drive the development of UI

4 Principles of PD

- Computerization is not to automate human skills but to give the workers better tools for doing their job
 - Exception might be expert systems or systems in which we want automation

4 Principles of PD

- The users knows best how their work and work environment should be improved
 - Designer is a consultant of how to achieve this
 - Users have opportunity to have a say, and they are the experts

4 Principles of PD

- The user's views and attitudes towards technology are as important to the success as what they can do with the technology
 - Effective system can be useless

4 Principles of PD

- Technology should be considered as processes in the context of workplace
 - Not as an individual product
 - A network of different technologies, organizations, practices, and people, affecting each other

Challenges of PD

- Participation in design can succeed only if it
 - Makes a difference for the users
 - Implementation of the results is likely
 - Is fun
- How to encourage participation?
