

Course introduction User-Centered Design

Prof. Cl. Müller-Birn

April 13, 2015

«UCD: User-Centered Software Development»

Outline

1. Examples from everyday life
2. Definitions of usability
3. From the importance of mental models
4. Class introduction (topics, organization, coordination, grading)
5. User-centered design process
6. Hands-on
7. Integrating user-centered design into your development practice

Course introduction **Motivation**







HOTELS FLIGHTS CARS PACKAGES

[Login](#)


Flights

[Round-trip](#) [One-way](#) [Multi-city](#)

| | | | |
|---------------|--------------|------------------|------------------------|
| Houston (IAH) | Denver (DEN) | 1 adult, Economy | Search |
|---------------|--------------|------------------|------------------------|

[Include nearby](#) [Show airport map](#) [Include nearby](#) [Show airport map](#) [Hide flexible dates ▲](#)

[Exact dates](#) [± 3 days](#) [Weekends](#)

Depart [Mon 2/16](#) [3 days before and after ▾](#)

Return [Sat 2/21](#)



Advice: **BUY** Confidence: 79%
 Prices may rise within 7 days ⓘ

Trip Length

Nights at destination
0 – 12



Stops

nonstop \$130
 1 stop \$253

Times

IAH ↔ DEN | [Feb 16 ± 3 days](#) → [Feb 21 ± 3 days](#) | Economy cabin | 1 traveler | [Change](#)

Fares shown are the best available matching your filter selections.

| Return ↗ | Wed 2/18 | Thu 2/19 | Fri 2/20 | Sat 2/21 | Sun 2/22 | Mon 2/23 | Tue 2/24 |
|----------|----------|----------|----------|----------|----------|----------|----------|
| Fri 2/13 | \$449 | \$509 | \$459 | \$539 | \$639 | \$599 | \$426 |
| Sat 2/14 | \$373 | \$433 | \$383 | \$463 | \$563 | \$523 | \$350 |
| Sun 2/15 | \$273 | \$333 | \$283 | \$363 | \$463 | \$423 | \$250 |
| Mon 2/16 | \$253 | \$313 | \$263 | \$342 | \$443 | \$349 | \$230 |
| Tue 2/17 | \$153 | \$213 | \$163 | \$225 | \$315 | \$303 | \$130 |
| Wed 2/18 | \$153 | \$155 | \$155 | \$155 | \$245 | \$245 | \$130 |
| Thu 2/19 | -- | \$195 | \$183 | \$195 | \$285 | \$285 | \$150 |

Defining usability (ISO 9241-11)

*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**.*

Effectiveness: can users complete tasks, achieve goals with the product, i.e. do what they want to do?

Efficiency: how much effort do users require to do this? (Often measured in time)

Satisfaction: what do users think about the products ease of use?

Defining usability (Nielsen, 2012)

Usability is a quality attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process.

- » Learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design?
- » Efficiency: Once users have learned the design, how quickly can they perform tasks?
- » Memorability: When users return to the design after a period of not using it, how easily can they reestablish proficiency?
- » Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- » Satisfaction: How pleasant is it to use the design?

(Nielsen, 2012)

When is a software useful?

Another important quality attribute is **utility**.

It refers to the design's functionality: **Does it do what users need?**

Utility = whether it provides the features you need.

Usability = how easy and pleasant these features are to use.

Useful = usability + utility.

Motivation Why do we care?



"The average user interface has some 40 flaws. Correcting the easiest 20 of these yields an average improvement in usability of 50%. The big win, however, occurs when usability is factored in from the beginning."

"Sun Microsystems has shown how spending about \$20,000 could yield a savings of \$152 million dollars. Each and every dollar invested could return \$7,500 in savings."

"The rule of thumb in many usability-aware organizations is that the cost-benefit ratio for usability is \$1:\$10-\$100. Once a system is in development, correcting a problem costs 10 times as much as fixing the same problem in design. If the system has been released, it costs 100 times as much relative to fixing in design."

Usability of a software is...

the extent to which it can be used by a particular user ...

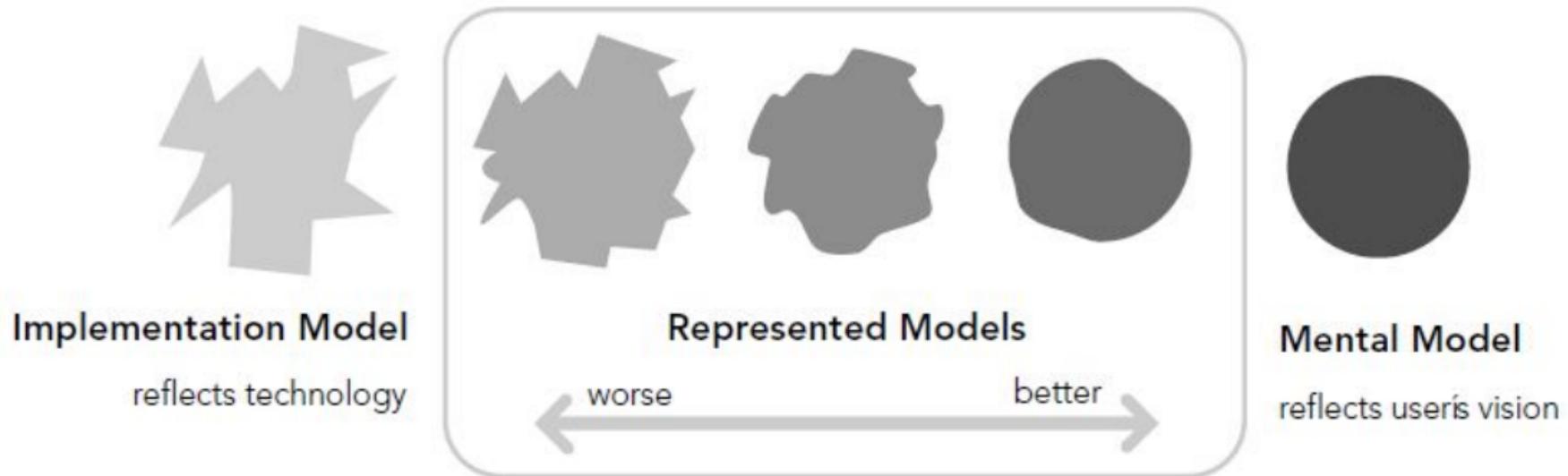
in order to archive specific goals in a certain context ...

effectively, efficiently and satisfactorily.



Developer watching videotape of usability test.

What is the problem?



Software interfaces should be designed to help users build productive mental models of a system (Preece, 1994).

What is the problem with mental models?

Unscientific: They are often based on guesswork and approximations.

Partial: They do not necessarily describe whole systems, just the aspects that are relevant to the persons who formulate them.

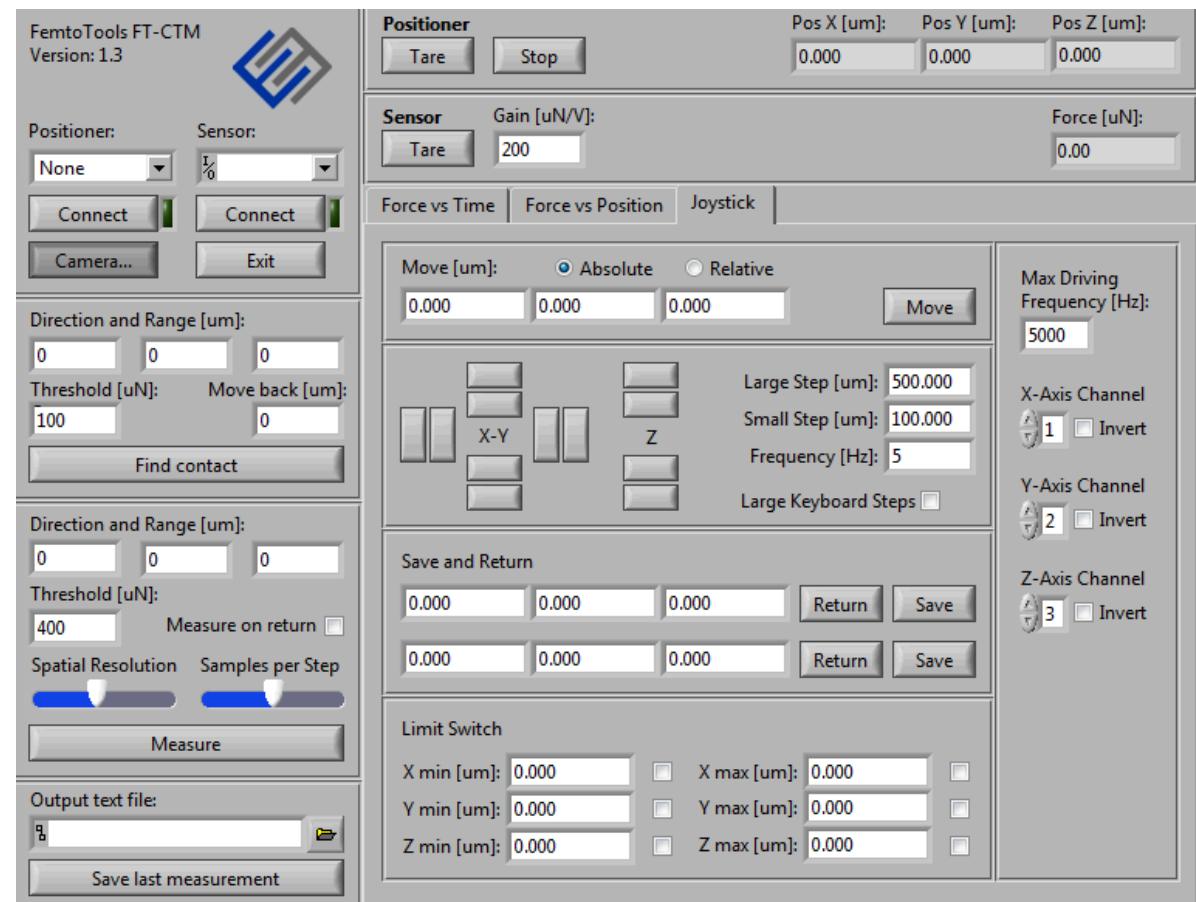
Unstable: They are not concrete formulations, but evolve and adapt to the context.

Inconsistent: They do not necessarily form a cohesive whole; some parts may be incompatible with other parts of the same model.

Personal: They are specific to each individual and are not universal concepts that can be applied generically.

Common design methods employed to support and influence users' mental models

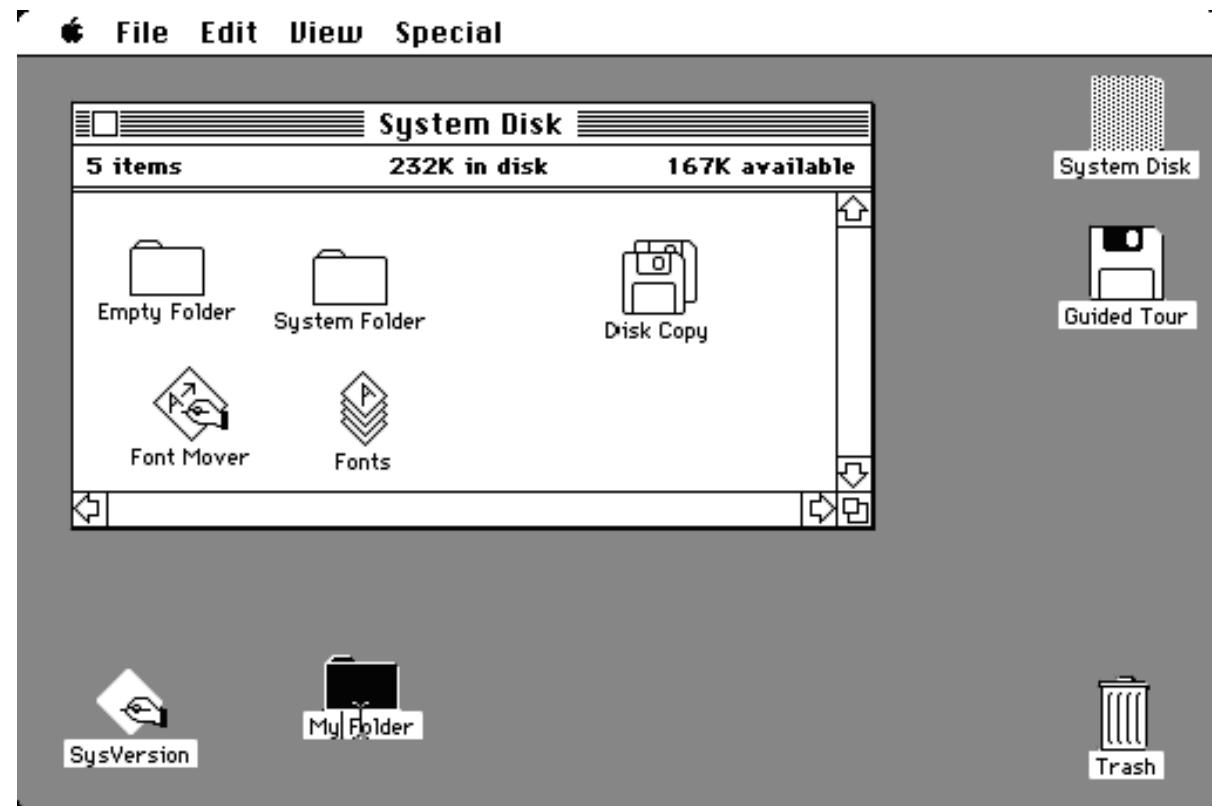
» Simplicity



(Davidson et al, 1999)

Common design methods employed to support and influence users' mental models

- » Simplicity
- » Familiarity



(Davidson et al, 1999)

Common design methods employed to support and influence users' mental models

- » Simplicity
- » Familiarity
- » Availability

The screenshot shows a section of an e-commerce website for books. At the top, a header reads "More Items to Consider". Below it, two sections are displayed: "You looked at" and "You might also consider". Each section contains a thumbnail image of a book cover, the book's title, its author, its price, and a "LOOK INSIDE!" button. Red arrows point from the "You looked at" and "You might also consider" headers to their respective "LOOK INSIDE!" buttons. Below these sections, another header reads "Related to Items You've Viewed". It contains similar "You looked at" and "You might also consider" sections with book recommendations. The books shown include "JavaScript: The Good Parts" by Douglas Crockford, "JavaScript: The Definitive Guide" by David Flanagan, "CSS: The Missing Manual" by David McFarland, "Learning jQuery 1.3" by Chaffer, Kar, "Forms that Work: Designing Web Forms for Usability" by Cynthia Johnson, "Don't Make Me Think: A Common Sense Approach to Web Usability" by Steve Krug, "Letting Go of the Words: Writing Web Content That Works" by James (Guru) Reiss, and "Designing Web Interfaces" by Jeff Johnson.

| Section | Book Title | Author | Price |
|------------------------------------|--|--------------------------------|------------------------|
| You looked at | JavaScript: The Good Parts | Paperback by Douglas Crockford | \$29.99 \$19.79 |
| | JavaScript: The Definitive Guide | Paperback by David Flanagan | \$49.99 \$31.49 |
| You might also consider | CSS: The Missing Manual | Paperback by David McFarland | \$34.99 \$23.09 |
| | Learning jQuery 1.3 | Paperback by Chaffer, Kar | \$39.99 \$35.9 |
| Find similar items | | | |

(Davidson et al., 1999)

Common design methods employed to support and influence users' mental models

- » Simplicity
- » Familiarity
- » Availability
- » Flexibility



(Davidson et al, 1999)

Common design methods employed to support and influence users' mental models

- » Simplicity
- » Familiarity
- » Availability
- » Flexibility
- » Feedback

The screenshot shows a web form titled "Update Your Passcode" from the Bank of America Online Banking website. The form includes fields for creating a new passcode and re-entering it, along with a "Quick Help" section detailing passcode format rules. A callout box highlights specific validation rules for the passcode field.

Bank of America

Online Banking En Español

Update Your Passcode

Quick Help

* = required information

Passcode Format

- Must be 8 and 20 characters
- Must include at least 1 letter and 1 number
- Can include uppercase and/or lowercase letters (Passcode is case-sensitive)
- Cannot contain the following special characters: \$ < > & ^ ! []
- Cannot contain spaces
- Cannot be the same as your Online ID

Create your new Passcode

...
Note: Your Passcode must cor
Online ID.

Your Passcode:

- Must be 8 to 20 characters
- Must include at least 1 letter and 1 number
- Cannot contain spaces or invalid symbols
- Cannot be the same as your Online ID

Re-enter your new Passcode

Continue

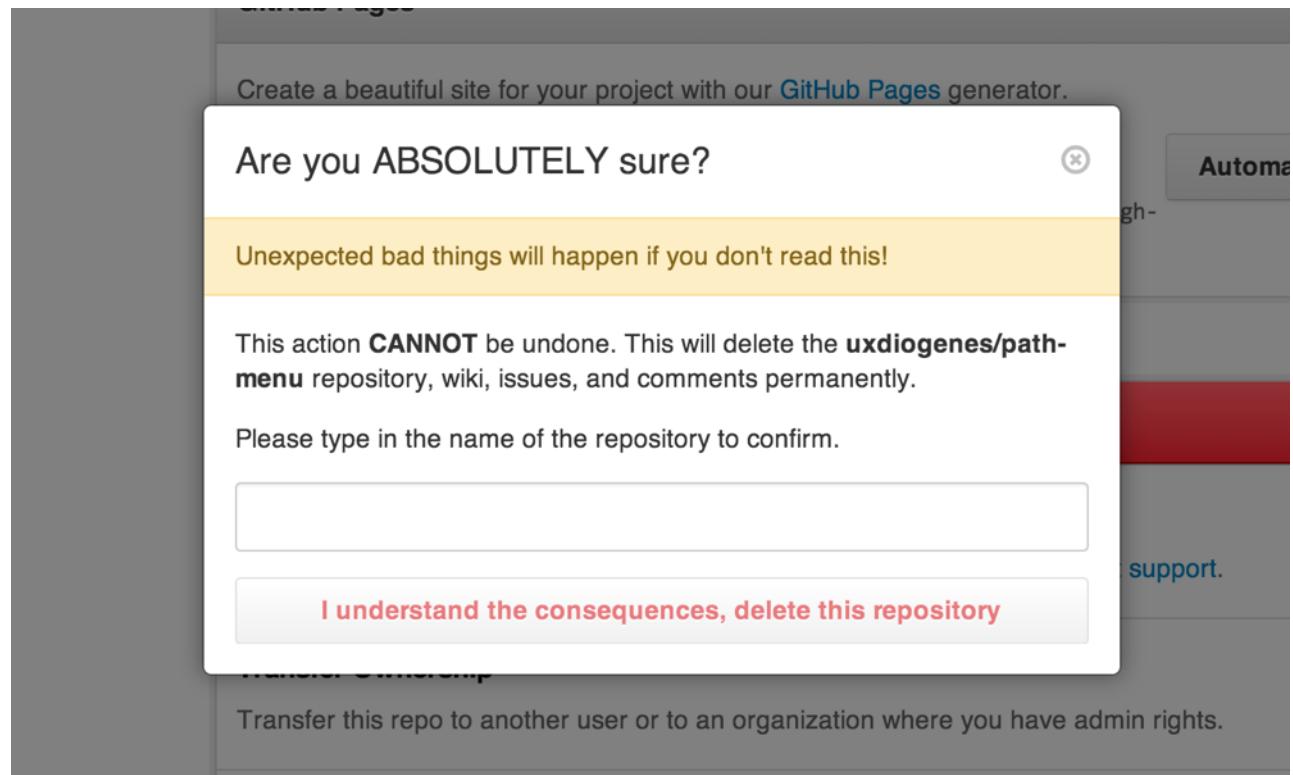
Secure Area

Bank of America, N.A. Member FDIC. Equal Housing Lender ©
©2010 Bank of America Corporation. All rights reserved.

(Davidson et al, 1999)

Common design methods employed to support and influence users' mental models

- » Simplicity
- » Familiarity
- » Availability
- » Flexibility
- » Feedback
- » Safety



(Davidson et al, 1999)

Common design methods employed to support and influence users' mental models

- » Simplicity
- » Familiarity
- » Availability
- » Flexibility
- » Feedback
- » Safety
- » Affordances



(Davidson et al, 1999)



How do we capture and validate users' mental models?

Activity (paper + pen + 5 minutes)

Question:

What do you think, how can YOU capture a users mental model?



Focus Groups

Surveys

Interviews

Contextual Inquiry Questionnaires

Usability Testing

Observations

Usability of a software/product is...

the extent to which it can be used by a particular user ...

in order to archive specific goals in a certain context ...

effectively, efficiently and satisfactory.

What do I do with this information?

How do I determine whether I have done all right?

How can I get this information?

I need to know who are my users.

I need to know the objectives of my users and the context of their work.

I need to know prior knowledge, the mental model, and the cognitive abilities of my users.

Topics of this course

User-Centered Design – *Understanding the user*

- Determining target groups, objectives, tasks, mental models, working context, prior knowledge
- Methods of data collection and –analysis

Web Site and Application Design – *Specifying and developing the design*

- Interaction design and information architecture
- Information design
- Interface and navigation design

Usability Testing – *Evaluating and improving the design*

- Types of usability tests and expert evaluation
- Preparing and executing tests
- Analyzing and evaluating tests

Preliminary lecture schedule

| | |
|----------|---|
| 13.04.15 | Course introduction and user-centered design process |
| 20.04.15 | User research: interview, participant observation |
| 27.04.15 | User modeling: user groups, scenarios, task analysis |
| 04.05.15 | Human factors relevant to the visual design of software |
| 11.05.15 | Interaction design and information architecture prototyping (low and high fidelity) |
| 18.05.15 | Usability assessment: formative vs. summative evaluation |
| 25.05.13 | <<Pfingsten>> |

Preliminary lecture schedule (cont.)

| | |
|----------|--|
| 01.06.15 | Interface, navigation and interaction design |
| 08.06.15 | Usability inspection: action analysis, heuristic evaluation, cognitive walkthrough |
| 15.06.15 | Accessibility and internationalization |
| 22.06.15 | Continuous usability testing via remote usability testing |
| 29.06.15 | Project presentation |
| 06.07.15 | Designing social web applications and visualizations |
| 13.07.15 | Exam |

Course in a nutshell

You will learn how to integrate methods of user-centered design into agile development projects.

You will learn how to study users and collect data.

You will learn how to summarize data in models to identify user needs and to derive users' requirements.

You will learn how to design and prototype your user interfaces.

You will learn how to evaluate your designs.

Course Organization

Context of this course

Course is part of module in the bachelor program

Normally in the 5th semester of your study

Most of the class material will be in English

We are adapting this course continuously to your needs, so your feedback is appreciated!

You can practically apply what you have learnt in a software project on UCD.

General course organization

The class takes place every Mon, 14 - 18 PM, in room SR 006 (Takustraße 9).

Instructors of this course is Prof. Dr. Cl. Müller-Birn

Rather than separating the course content into a lecture and an exercise, we offer an integrated class

What does this mean?

Each class contains well-coordinated collective and individual learning phases.

Why is this useful?

We want to increase the efficacy of acquiring knowledge during classes.

Course coordination

Register!

Our main platform for communication and coordination is Blackboard: <https://lms.fu-berlin.de/>

We provide a UCD forum in Blackboard!

Please do not contact us in a one-one-manner, use always one-to-many for general, course- or assignment-related questions!

For general information check our course website:

<https://www.mi.fu-berlin.de/inf/groups/hcc/teaching/Sommersemester-2015/UCD-Nutzerzentrierte-Softwareentwicklung.html>

Grading

Your final grade is only based on the result of your written exam.

But

In order to actively participate in this course, you need to fulfill ALL of the following requirements

- you have to submit (n-2) of all assignments,
- you need to get at least 50 % of all points in each submitted assignment,
- the mean (= average) of all your assignments need to be above 60 %

You can acquire maximal 2 points per task in an assignment.

↑ 2 points

→ 1 point

↓ no points

All your homework assignments will be part of a final report that is part of the exam.



Questions?

Short break: 15”

The User-Centered Design Process: Integrating Agile Development and UCD

Prof. C. Müller-Birn

April 13, 2015

Course «User-Centered Design»

Learning Goals

1. You can describe characteristics of user-centered design.
2. You can explain the process of user-centered design and the “missing part”.
3. You can describe the main characteristics of agile development (esp. Scrum).
4. You can explain how Scrum and user-centered design can be integrated and which advantages exist for doing so.

What the user really needed



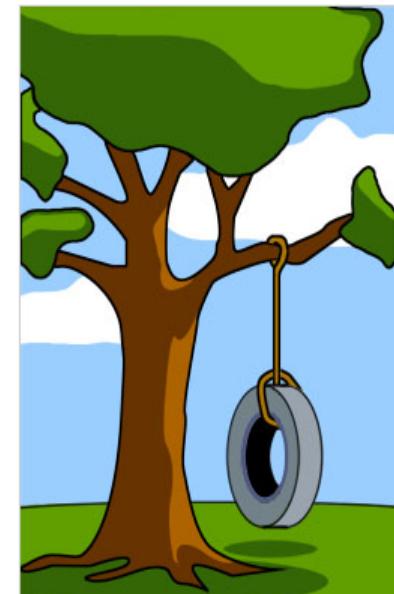
www.projectcartoon.com
How the customer explained it



www.projectcartoon.com
How the project leader understood it



www.projectcartoon.com
How the analyst designed it



www.projectcartoon.com
What the customer really needed



User-Centered Design (UCD)

What is user-centered design?

User-centered design is a **process** to find out

- who are your users,
- how do they work and think and
- what are the goals of the stakeholders involved in the project.

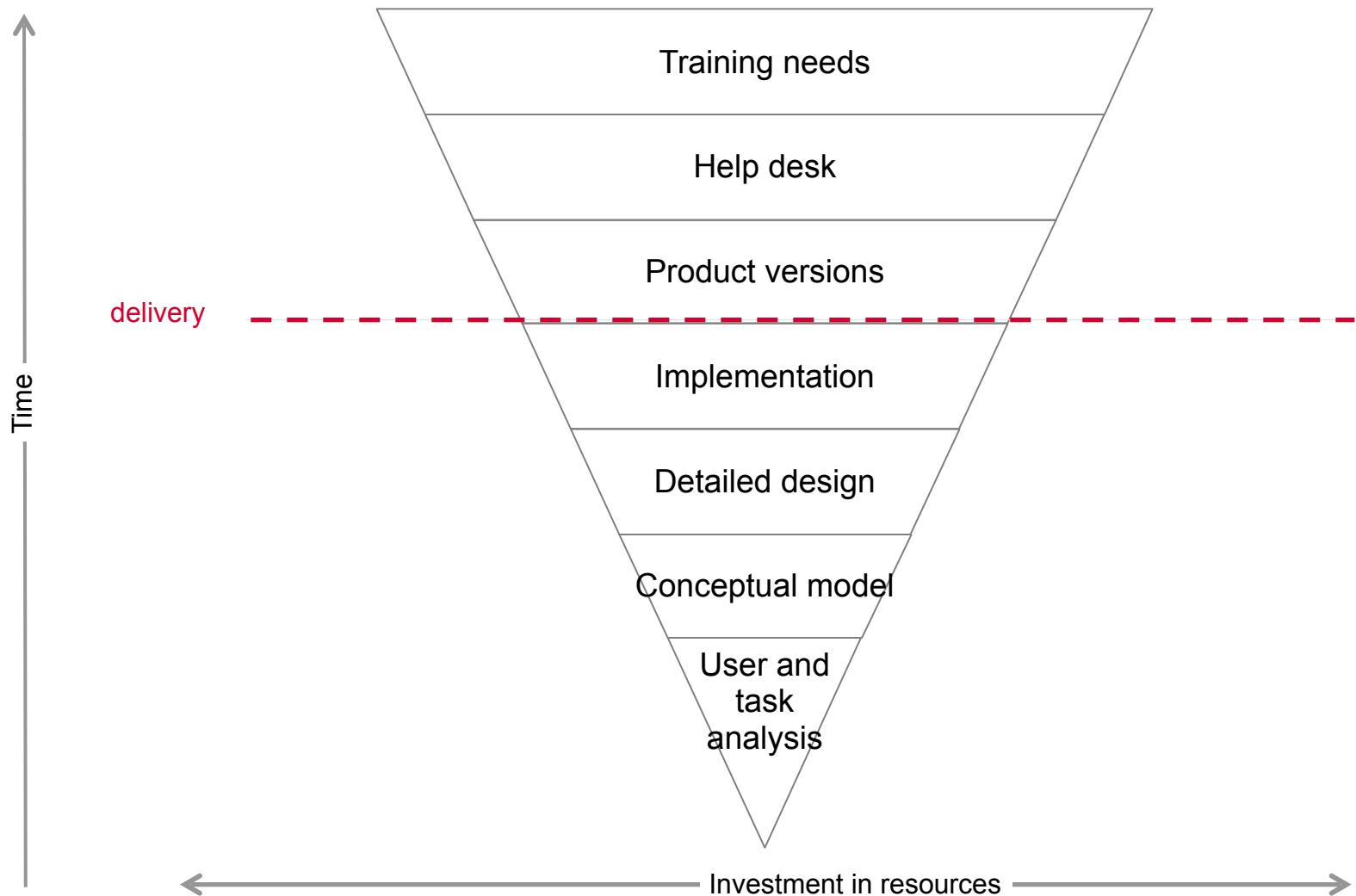
We collect **data** about

- user profiles,
- the work-environment of the users,
- scenarios, describing how users will use the application, and
- tasks the user will tackle with this application.

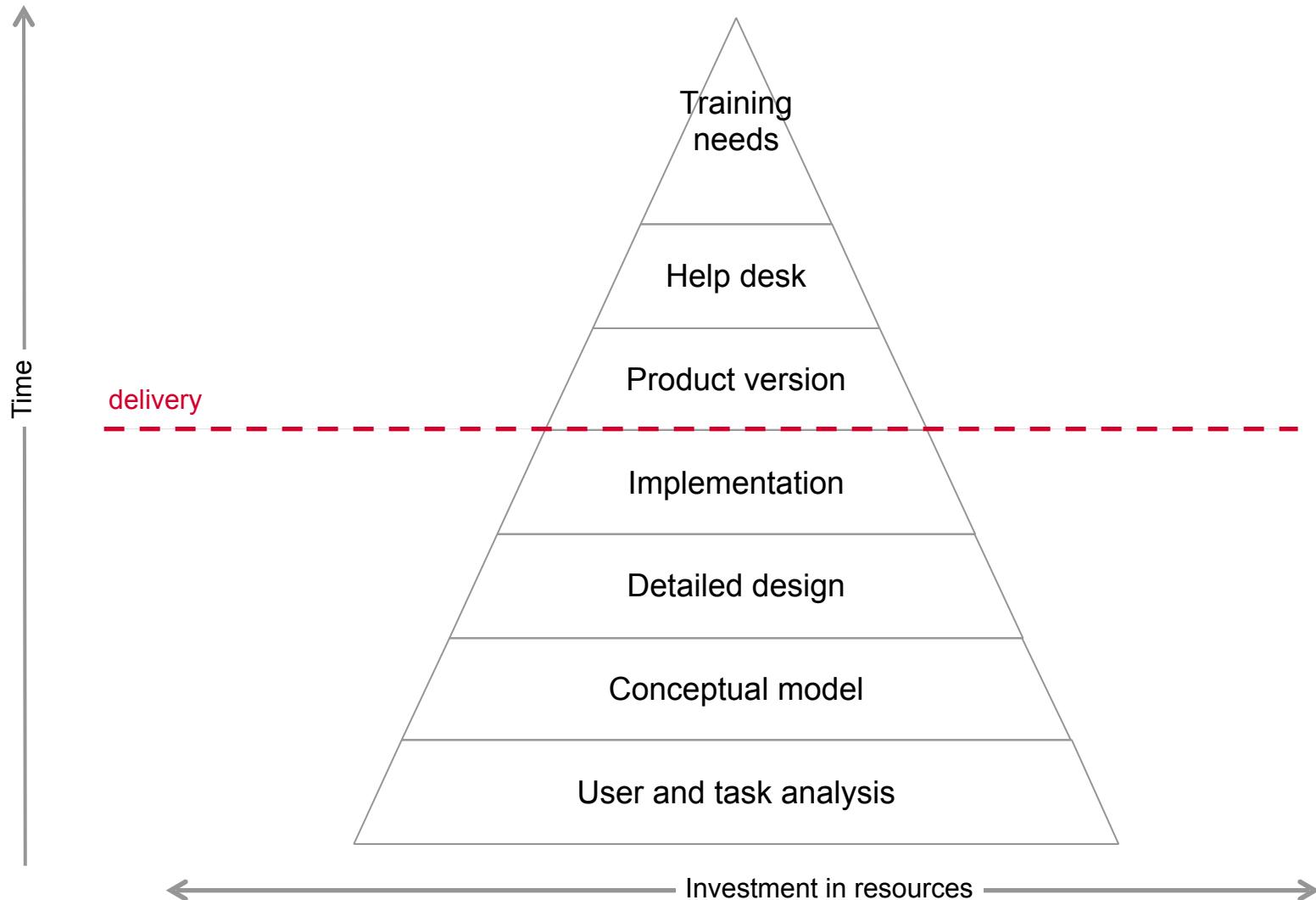
Aspects of user-centered design

- An active involvement of users in the design process
- An excellent understanding of the users' needs
- Assignment of functions according to the user and their terminology
- An iterative progress concerning the software design
 - Requirements – design – evaluation
 - Contradicts the idea „getting it right the first time“
- A multi-disciplinary design process
 - Developers, (web-)designers, managers & project leaders, marketing staff, etc. should be part of the team
 - Usually they all tend to use a different terminology

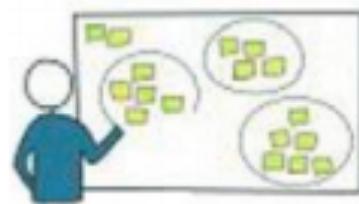
Less UCD = more “hidden” costs



More UCD – less “hidden” costs



User-centered design activities



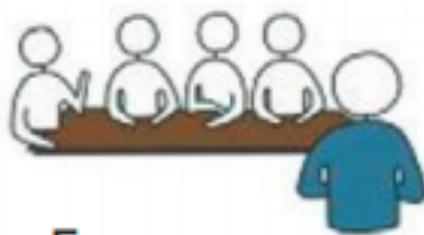
Affinity diagramming



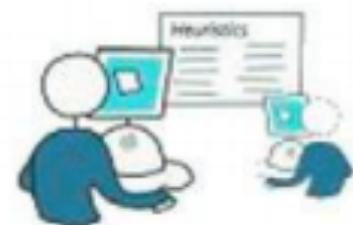
Card sort



Contextual inquiry



Focus group



Heuristic evaluation



Interview



Participant



Researcher



Researcher / Participant

User-centered design activities



Remote testing



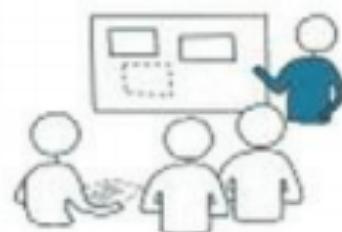
Task analysis



Usability test



Data analysis



Design critique



Diary / photo study



Participant

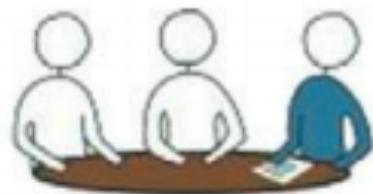


Researcher



Researcher / Participant

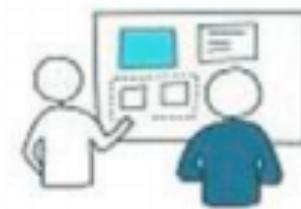
User-centered design activities



Meeting



Paper prototyping



Participatory design



Participant

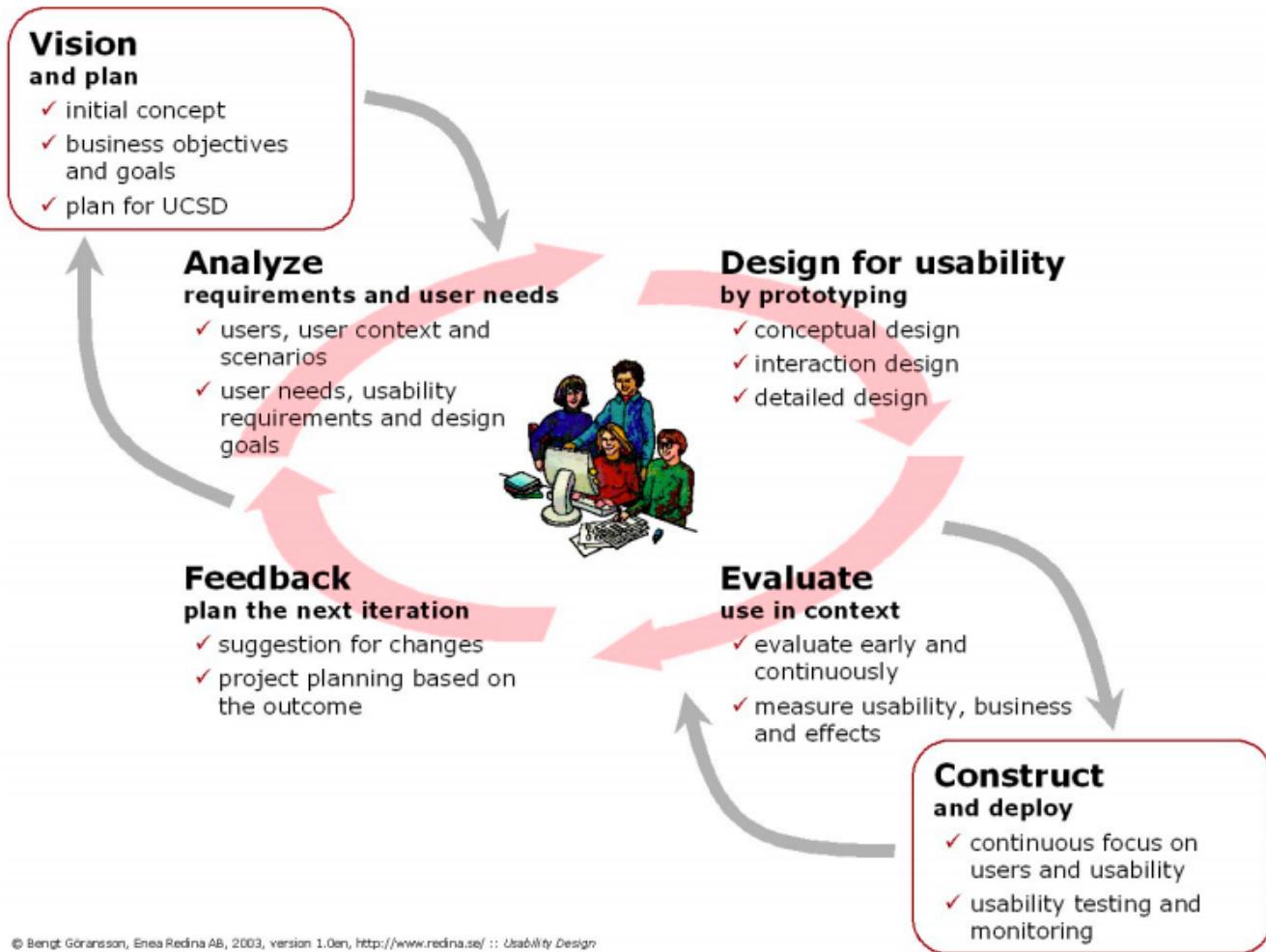


Researcher



Researcher / Participant

Process of user-centered design



© Bengt Göransson, Enea Redina AB, 2003, version 1.0en, <http://www.redina.se/> :: Usability Design

Hands-on:

Expert review (groups of 4-5 students, 30'')

Don't look at the chosen software (yet)!

Write a (short) story.

Try to use it (following the story).

Now look at it (now that you've had a chance to use it).

Present results (3" each group)



How can you use UCD in your own development practice?

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

- Individuals and interactions** over processes and tools
- Working software** over comprehensive documentation
- Customer collaboration** over contract negotiation
- Responding to change** over following a plan

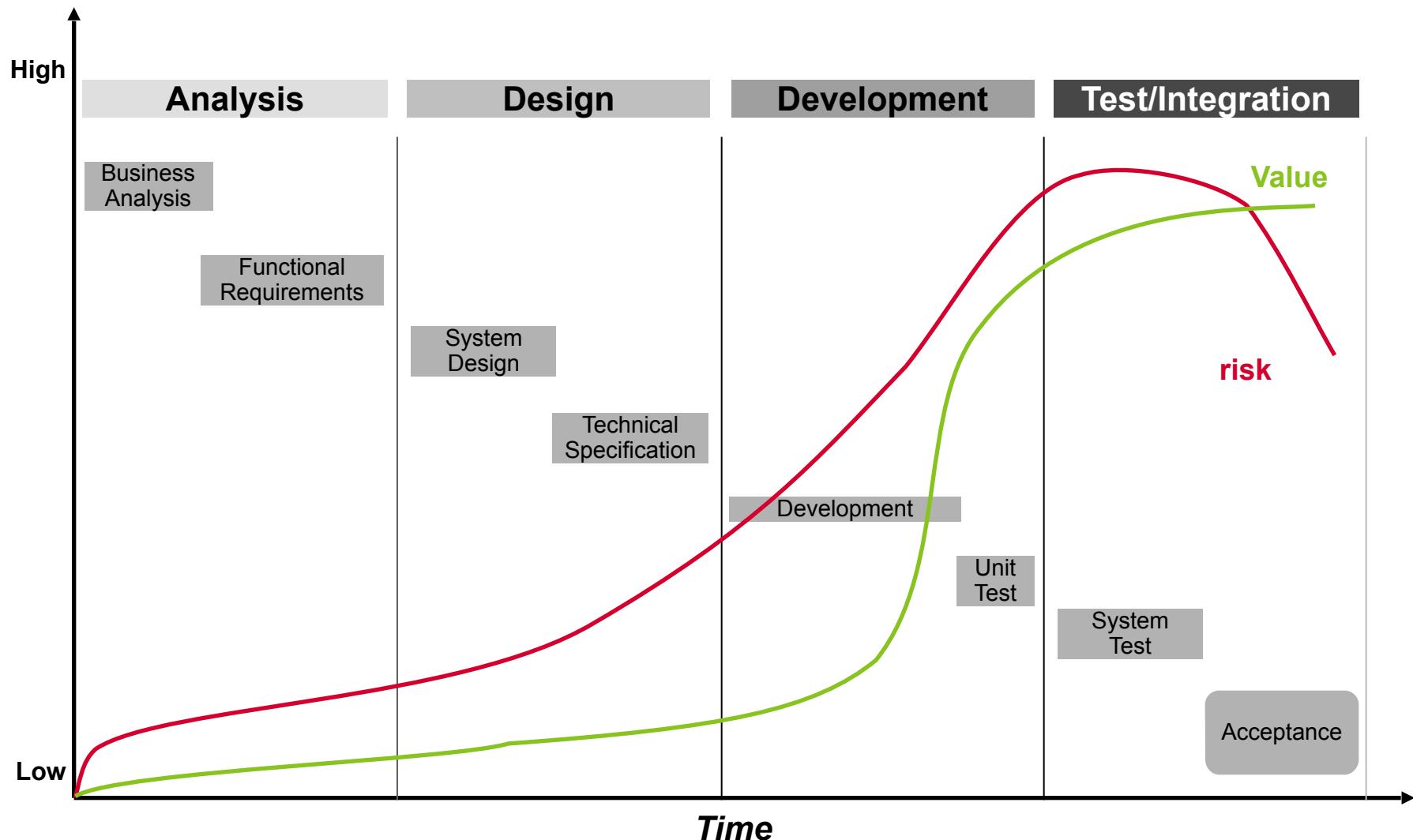
That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

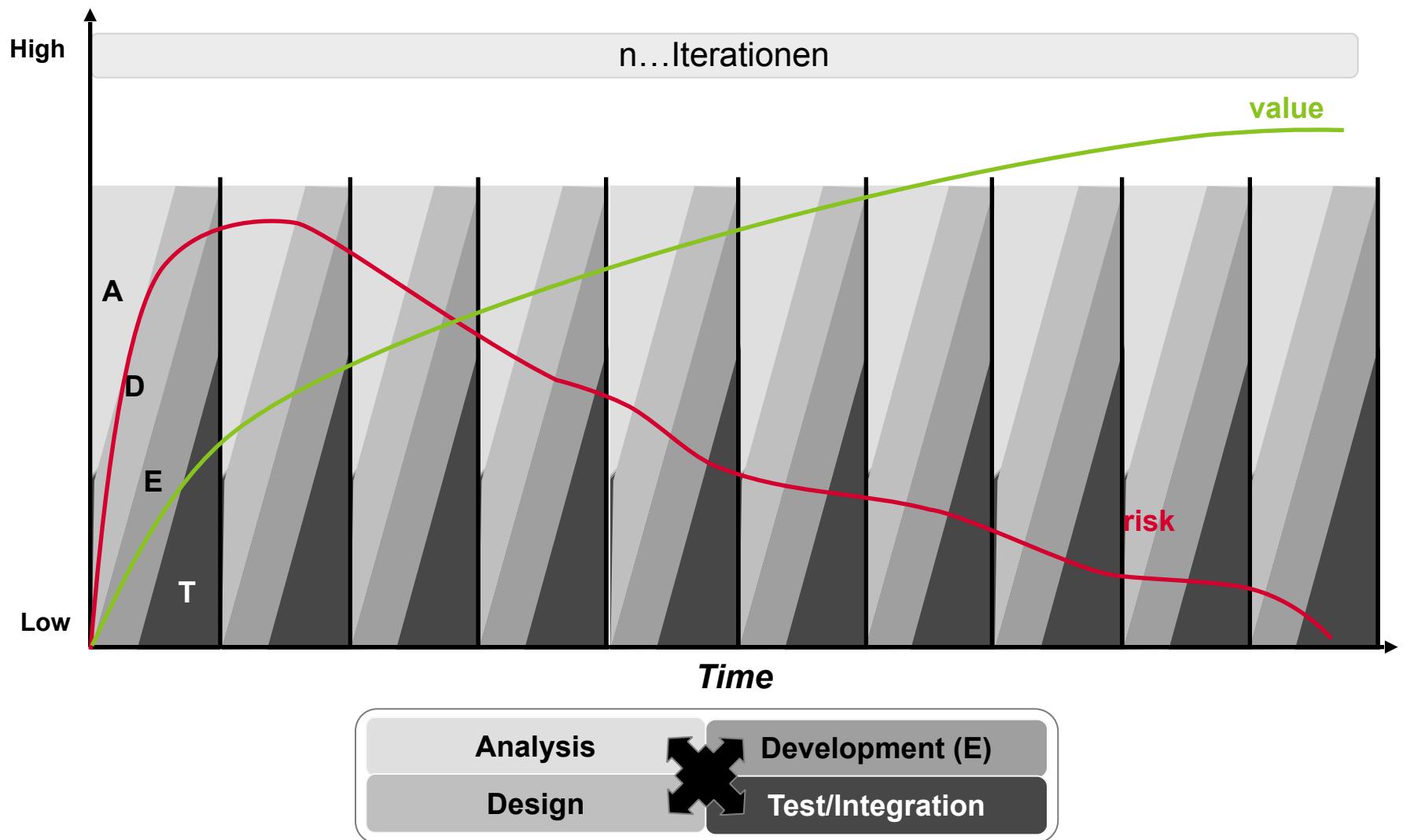
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

Traditional added value



Agile added value



Traditional vs. agile development

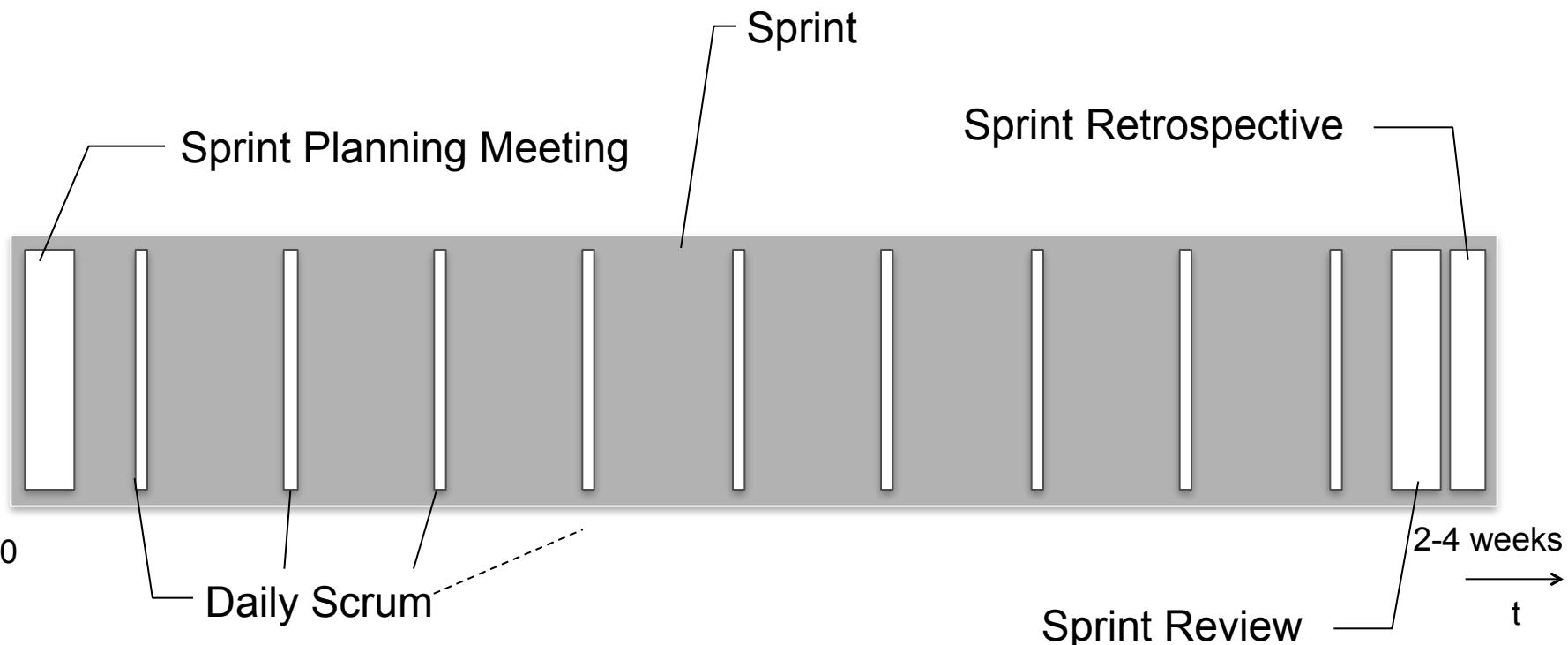
Traditional development

- Waterfall (sequential process)
- Success = Compliance with predictive plans
- Progress = Diverse artifacts
- Resistance to change

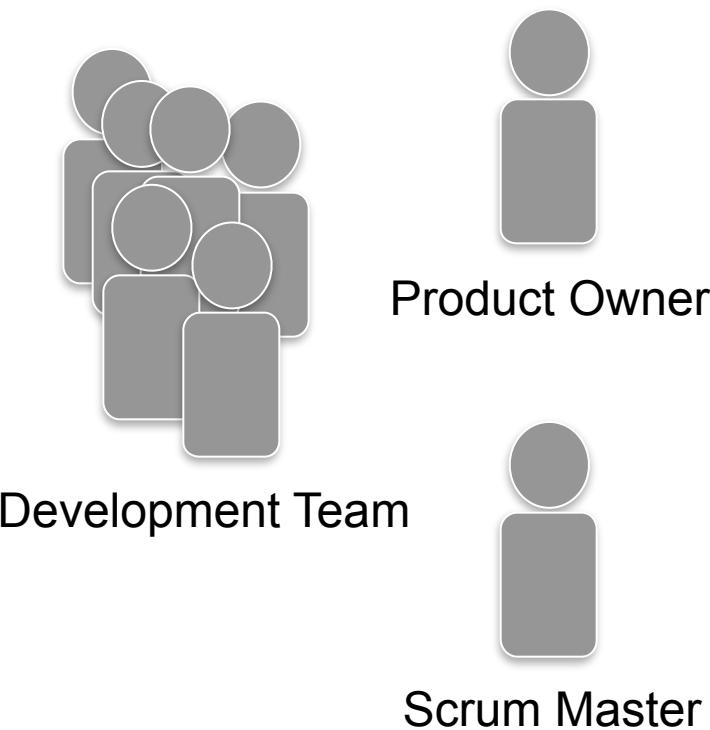
Agile development

- Iterative and incremental process
- Success = delivered business value
- Progress = Working software
- Changes are desired

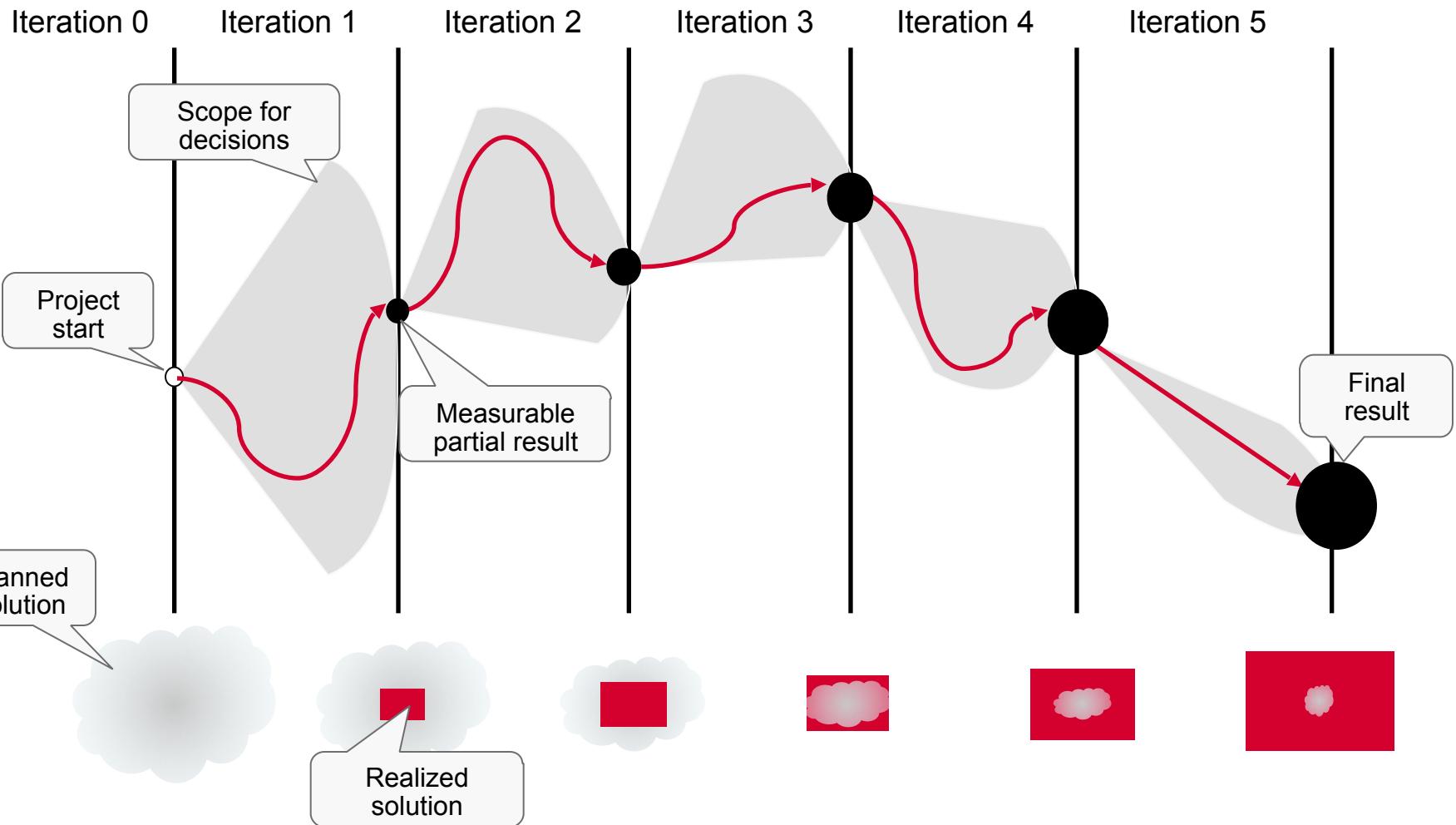
Scrum activities



Scrum team



Iterative development of a software



Do you want to know more?

The Scrum Guide

The Definitive Guide to Scrum:
The Rules of the Game



October 2011

Developed and sustained by Ken Schwaber and Jeff Sutherland

Scrum.org®

Improving The Profession Of Software Development



Integrating Agile Software Development and User-Centered Design

Benefits of integrating agile development and UCD

Development effort is not wasted on programming things that eventually would turn out to be wrong as requirements and design concepts are elaborated up-front (as opposed to agile).

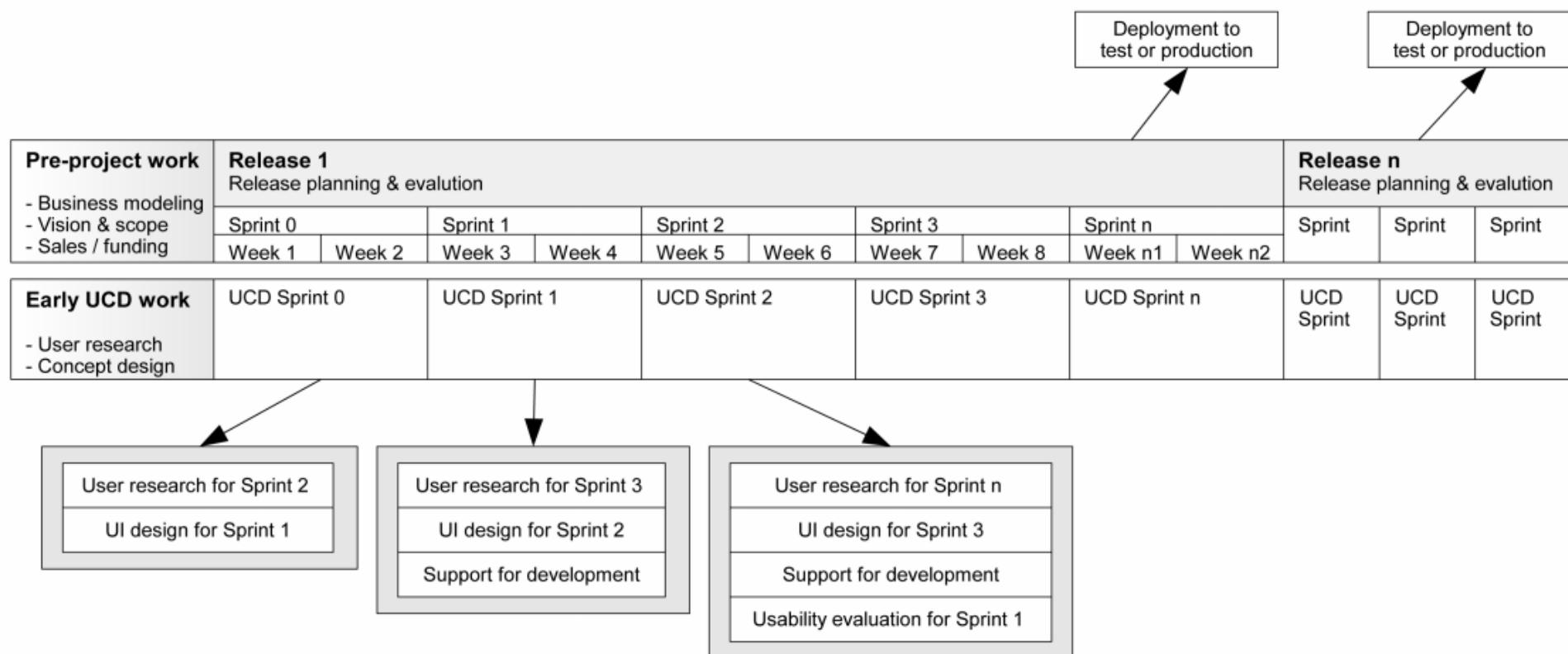
Usability investigations are conducted throughout the entire project.

The concept of time-boxed sprints helps developers to focus features that are important to users and to test with the related specific set of user tasks.

Timely feedback is always available.

(Rannikko, 2011)

Integrated user-centered development process



(Rannikko, 2011)

Wrapping up

You can define usability.

You can discuss characteristics and different methods to improve it.

You know what problems exist with mental models.

You are able to define user-centered design.

You can explain how user-centered design and software development can be integrated.