

# Agile Usability Engineering

by Thomas [Mommel](#)

Agile Usability Engineering is a concept to describe a combination of methods and practices of agile development and usability engineering. Therefore, this entry commences with a brief note on agile methods.

In recent years, agile methods for software and web engineering have reached widespread acceptance in the community. In contrary to classic, heavy-weight software engineering processes like the V-model, agile methods (Ambler 2002) begin coding very early while having a shorter requirements engineering up-front as well as less documentation. Following the paradigm of Extreme Programming (Beck 1999), implementation of code takes place in small increments and iterations, and small releases are delivered to the customer after each development cycle. During a short claims analysis, called the exploration phase, the development team writes user stories trying to describe user needs and roles; the interviewed people need not necessarily be the real users of the later software product. Seen from a human-computer engineering perspective, Extreme Programming (XP) thus often fails to collect real user data and starts coding with just assumptions about user needs. The development in small increments may work properly as long as the software has no focus on the user interface (UI). Changes to software architecture most often have no impact on what the user sees and interacts with.

With the UI, it's a different story. When designing UIs like websites, continuous changes of the user interface due to fast iterative design may conflict with user expectations and learnability, provoke inconsistency and possibly lead to user dissatisfaction. Evaluation of small releases with stakeholder participation does not ensure that the whole system provides a consistent conceptual, navigational or content model.

Nevertheless, the numerous discussions about agile approaches to user interface design (UID) have lead to a movement in the human-computer interaction community, which has begun to reconsider its user-centered heavy-weight usability lifecycles (see table 1, compare Mayhew 1999).

## Heavy-Weight Processes

Detailed, up-to-date documentations and models

## Light-Weight Processes

Cards and hand-drawn abstract models. Travel light. Communicate

g

rather than document.

High-fidelity prototypes	Abstract prototypes, use simplest tools
Develop and prove concepts with user feedback. Iterate.	Courage. Design for needs (user's tasks) rather than user expectations. Retrieve design from models rather than continuous user feedback.
Time-consuming usability evaluations, workshops with intense stakeholder integration	Fast usability inspections. No need to evaluate if models are right.

*Table 1 : Comparison of heavy- and light-weight processes exemplified by a few chosen aspects of principles and methods.*

Increasingly, software engineering (SE) and UID have to cope with a shorter time-to-market, whereas the quality of the delivered software must not suffer. This continuous shortening of development lifecycles is a great challenge to both project management and the applied methods and tools. Therefore, many usability engineering (UE), UID and agile method experts have developed light-weight approaches or so-called **agile usability engineering** (AUE).

### **An example of an agile approach to user interface design**

Constantine and Lockwood (1999) were one of the first and most important practitioners to come up with an alternative, light-weight approach to UID (see table 1). They believe that relying too heavily on user needs and feedback may lead to an overhasty narrowing of the design space. Designers run the risk of designing what the users *wish* as opposed to what they really need. In their opinion, iterative rapid prototyping is like trial-and-error design and may never lead to an optimal solution if too many stakeholders are involved and too many opinions influence the design process. Instead, their usage-centered design approach is based upon abstract models to describe users, tasks and content. Users are still involved as resources of information and validation, but the design is a straightforward transformation of models into design. Usage-centered design may therefore be described as a design by engineering approach. Due to the usage of easy-to-understand and easy-to-manage artifacts like hand-drawings or simple cards for documentation and

hand drawings or simple cards for documentation and abstract low-fidelity prototypes, usage-centered design can be rated as one of the first agile approaches to UID.

<b>User-Centered Design</b>	<b>Usage-Centered Design</b>
Focus is on users: user experience and user satisfaction	Focus is on usage: improved tools supporting task accomplishment
Driven by user input	Driven by models and modeling
Substantial user involvement	Selective user involvement
<ul style="list-style-type: none"> <li>• User studies</li> <li>• Participatory design</li> <li>• User feedback</li> <li>• User testing</li> </ul>	<ul style="list-style-type: none"> <li>• Explorative modeling</li> <li>• Model variation</li> <li>• Usability inspections</li> </ul>
Design by iterative prototyping	Design by modeling
Highly varied, informal, or unspecified processes	Systematic, fully specified process
Design by trial-and-error, evolution	Design by engineering

*Table 2 : Comparison of user- and usage-centered design.  
Source: Constantine and Lockwood (2002)*

Whereas Constantine and Lockwood defined their design approach based on modeling, others, like Gundelsweiler et al. (2004), have developed approaches based on similarities between XP and agile methods on one hand, and user experience (UE) on the other. Gundelsweiler et al. (2004) presents an agile user-centered approach to software engineering and proves that the incremental and iterative procedures of XP, and other related processes of agile methods, can be brought together with current UE practices. Gundelsweiler et al. (2004) integrates both methods of user-centered design (see e.g. Norman & Draper 1986) and usage-centered design (e.g. Constantine & Lockwood 1999). Other examples of combining XP and UE are Obendorf et al. (2006) and Holzinger & Slany (2006). They term their approaches XPnUE and eXtreme Usability (XU).

### How to learn more

If you want to learn more about Agile Usability Engineering, a good place to start would be to read Ambler (2002) and Beck (1999). While Ambler gives you an impression on agile methods, Beck introduces you to the principles and practice of XP. You might also visit on

of the following websites.

The Agile Manifesto, <http://agilemanifesto.org/>

The Agile Alliance, <http://www.agilealliance.com/>

Next, you should get an impression on classic, heavy-weight usability engineering approaches such as (Mayhew 1999) or (Preece et al. 2002). General information about usability can also be found at the Usability Professionals' Association (UPA), <http://www.upassoc.org/>

Constantine and Lockwood (1999) should be your next bed-time reading resource about first approaches to light-weight software and usability engineering. After you worked through this literature, you are to start with research papers about combinations of UE and AM (AUE).

A very interesting platform for sharing thoughts and experience is the Yahoo! Group on agile usability: Agile-Usability Group, <http://groups.yahoo.com/group/agile-usability/>

### **User Contributed Notes**

Do you have any comments/additions that you would like visitors to this page to see? Please submit your comments for the benefit of other visitors!

## References (bibliography)

[what's this?](#)

[Ambler](#), Scott W. (2002): *Agile Modeling*. New York, [John Wiley and Sons](#)

[Beck](#), Kent (1999): *Extreme Programming Explained: Embrace Change*. Reading, MA, [Addison-Wesley Publishing](#)

↑ View publication info on [Amazon.com](#) or [.co.uk](#) 

[Constantine](#), Larry L. and [Lockwood](#), Lucy A. D. (2002): *Usage-Centered Engineering for Web Applications*. In [IEEE Software Magazine](#), 19 (1) pp. 42-50

[Constantine](#), Larry L. and [Lockwood](#), Lucy A. D. (1999): *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*. [ACM Press](#)

↑ View publication info on [Amazon.com](#) or [.co.uk](#) 

[Gundelsweiler](#), Fredrik, [Mommel](#), Thomas and [Reiterer](#), Harald (2004): Agile Usability Engineering. In: [Keil-Slawik](#), Reinhard, [Selke](#), Harald and [Szwilius](#), Gerd (eds.) [Mensch & Computer 2004: Allgegenwärtige Interaktion](#) September 5-8, 2004, Paderborn, Germany. pp. 33-42.

[Holzinger](#), Andreas and [Slany](#), Wolfgang (2000): *XP + UE -> XU: Praktische Erfahrungen mit eXtreme Usability*. In [Informatik Spektrum](#), 29 (1) pp. 91-97

[Mayhew](#), Deborah J. (1999): *The Usability Engineering Lifecycle: A Practitioner's Guide to User Interface Design*. [Morgan Kaufmann Publishers](#)

↑ View publication info on [Amazon.com](#) or [.co.uk](#) 

[Norman](#), Donald A. and [Draper](#), Stephen W. (eds.) (1986): *User Centered System Design: New Perspectives on Human-Computer Interaction*. Hillsdale, NJ, [Lawrence Erlbaum Associates](#)

↑ View publication info on [Amazon.com](#) or [.co.uk](#) 

[Obendorf](#), Hartmut, [Schmolitzky](#), Axel and [Finck](#), Matthias (2006): XPnUE – defining and teaching a fusion of eXtreme programming and usability engineering. In: [HCI Educators Workshop 2006 - HCIED2006](#) March 23–24, 2006, Limerick, Irland.

[Preece](#), Jenny, [Rogers](#), Yvonne and [Sharp](#), Helen (2002): *Interaction Design: Beyond Human-Computer Interaction*. [John Wiley and Sons](#)

↑ View publication info on [Amazon.com](#) or [.co.uk](#) 

## Changes to this page