



Human-Centered Software Engineering: DESIGNING FOR AND WITH HUMANS

Canadian Undergraduate Software Engineering Conference March 7-9, 2002, Montreal, Canada



Ahmed Seffah
Department of Computer Science
seffah@cs.concordia.ca
<http://hci.concordia.ca>



Outlines

- Designing for People and Usability
- Is Usability a Good Business for Software Engineers?
- How Usability can be Engineered?
- Concordia Research Chair on Human-Centered Software Engineering
- Conclusion

Designing for people!

- Is the software system easy to use (usability) and easy to learn (learnability)?
- Usability is the extent to which, a product can be used by a specified set of users to achieve specified goals with effectiveness, efficiency and satisfaction in a certain context of work [ISO 9241-11 standard]

Usability of what?

Computing Variety

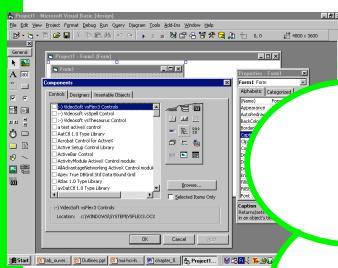


Wearable Computing

BlazerJet

CardPC, GPS, GSM 'phone, a novel Pinger receiver and both the audio interface and a Jornada 420 palmtop.

Multiple User Interface for Universal Accessibility



e.Book



Realthing UI

UI Style

Handheld Computing



Stationary Office Computers

Is usability a good business for software engineers?

■ User

- ▶ User satisfaction, productivity, performance, safety

■ Company

- ▶ Sales and competitiveness
- ▶ Development, maintenance and user support costs and time

Is usability a good business for software engineers?

American Express Customer Service

- Integrating a task-oriented wizard in an existing system
- Conclusions after usability tests
 - ▶ Training period – 12h > 2h
 - ▶ Productivity – 17 > 4 minutes/request
 - ▶ Errors rate – 20 > 2%

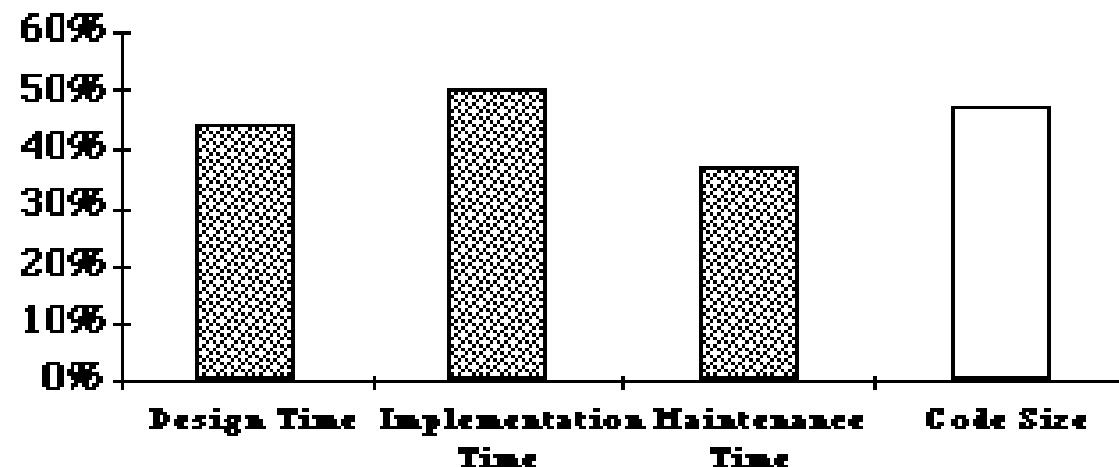
AT&T saved \$2,500,000 in training expenses as a result of usability improvement

The rule of thumb

- For each dollar a company invests in developing the usability of a product, the company receives \$10-\$100 in benefits (Karat, 1995)

Is usability a good business for software engineers?

- 47-60% of the code is devoted to the user interface
- 50% of the development time is spent on the user interface [MacIntyre et al, 1990; Mayer, 1995]



Is usability a good business for software engineers?

- Maintenance cost represents 80% of total software development costs
 - ▶ 80% related to problems of user with the system; not technical bugs (Boehm, 1991; Pressman, 1992; Martin, 1993)
 - 64% are related to usability problems (Laundauer, 1995)

What is wrong in software engineering?

Project Success Factors	% of Responses
1. User Involvement	15.9%
2. Executive Management Support	13.9%
3. Clear Statement of Requirements	13.0%
4. Proper Planning	9.6%
5. Realistic Expectations	8.2%
6. Smaller Project Milestones	7.7%
7. Competent Staff	7.2%
8. Ownership	5.3%
9. Clear Vision & Objectives	2.9%
10. Hard-Working, Focused Staff	2.4%
Other	13.9%

What is wrong in software engineering?

Project Challenged Factors	% of Responses
1. Lack of User Input	12.8%
2. Incomplete Requirements	12.3%
3. Changing Requirements & Specifications	11.8%
4. Lack of Executive Support	7.5%
5. Technology Incompetence	7.0%
6. Lack of Resources	6.4%
7. Unrealistic Expectations	5.9%
8. Unclear Objectives	5.3%
9. Unrealistic Time Frames	4.3%
10. New Technology	3.7%
Other	23.0%

What is wrong In software engineering?

Project Fail Factors

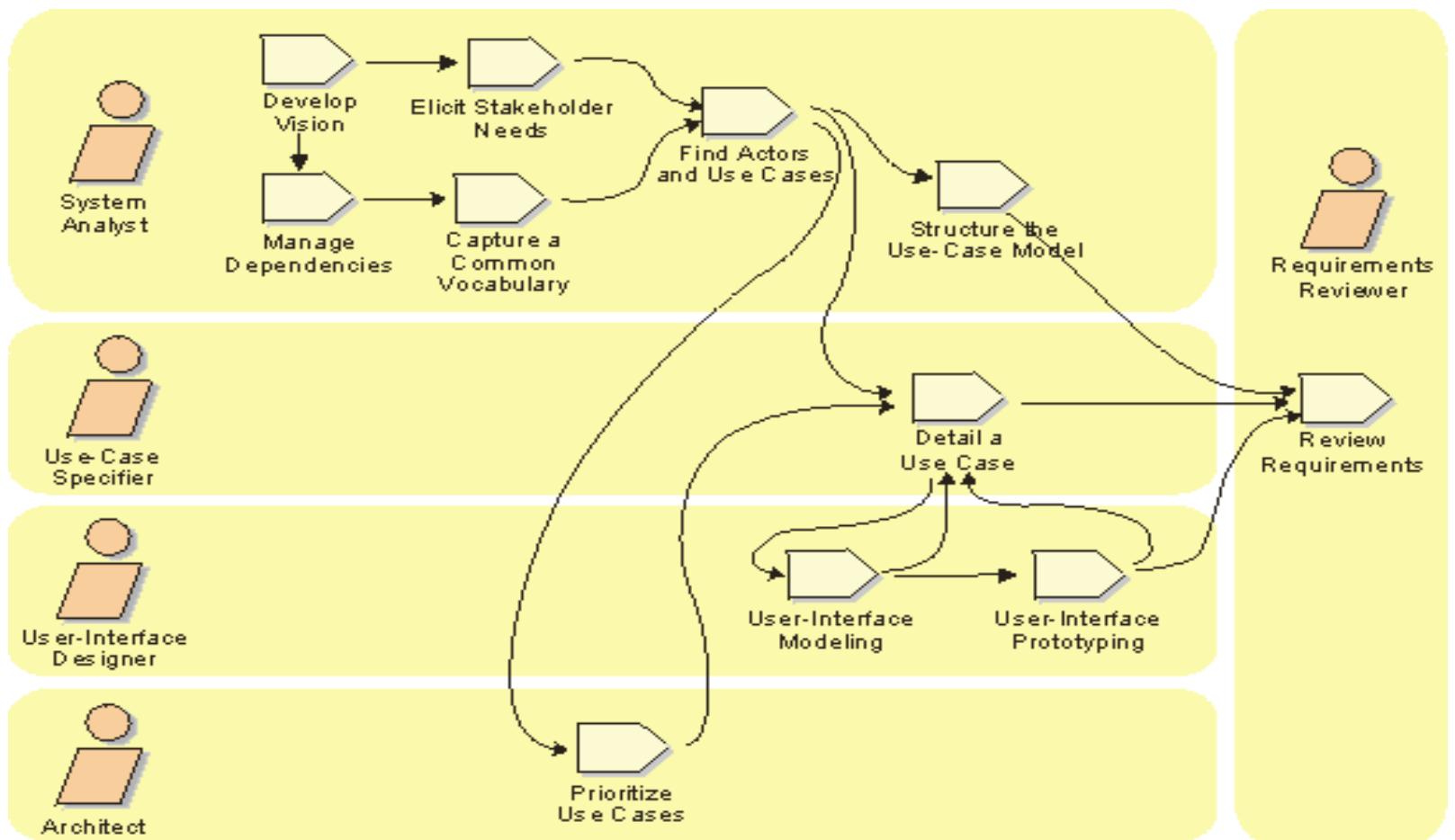
% of Responses

1. Incomplete Requirements	13.1%
2. Lack of User Involvement	12.4%
3. Lack of Resources	10.6%
4. Unrealistic Expectations	9.9%
5. Lack of Executive Support	9.3%
6. Changing Requirements	8.7%
7. Lack of Planning	8.1%
8. Didn't Need It Any Longer	7.5%
9. Lack of IT Management	6.2%
10. Technology Illiteracy	4.3%
Other	9.9%

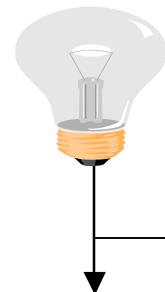
Involving user is key issue

- Ensures that the product is being designed so that users will be satisfied
 - ▶ User involvement is one of 12 best influences on software engineering [IEEE Software, January 2000]

User in software development lifecycle



How usability can be engineered?



Peanut Butter Theory of Usability

- User Documentation
- Training Resources [Seffah-93]
- Help and Performance Support Systems [Seffah-90]

Product Orientation

- UI Metaphors
- Adaptable and adaptive UI [Seffah-97, 98]
- **Context-Sensitive User Interface**

Process Orientation

- Guidelines and Patterns [Seffah-01]
- **User-Centered Design** [Norman-86]

Task Model-Oriented Development [Forbrig, 01; Seffah, 02]

Moving to Human-Centered Software Development

Traditional software development

Versus

Human-centered development

Technology/developer-driven



User-driven

Component focus



Solution focus

Individual contribution



Multidisciplinary teamwork including users, customers, human factors experts

Focus on internal architecture



Focus on external attributes (look and fell, interaction)

Quality measured by product defects and performance (system quality)



Quality defined by user satisfaction and performance (quality in use)

Implementation prior to human validation



Implementation of user-validated solution only

Solutions are directed by functional requirements



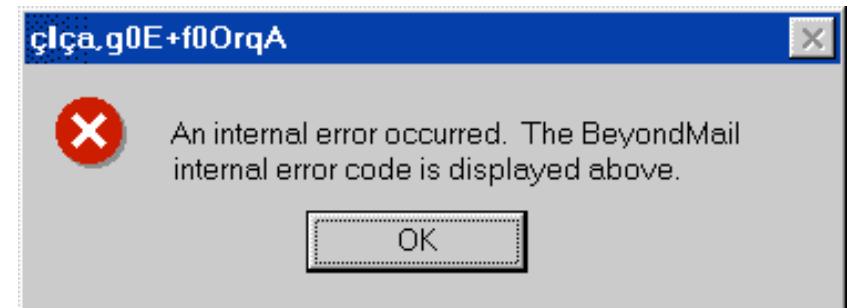
Understanding the context of use (user, task, work environment)

How usability can be engineered?

- All the aspects that affect software usability should evolve in parallel
 - ▷ **user interface**
 - ▷ help system
 - ▷ training plan and resources
 - ▷ user documentation
 - ▷ technical support
 - ▷ installation and configuration procedures
- They should be under one management process

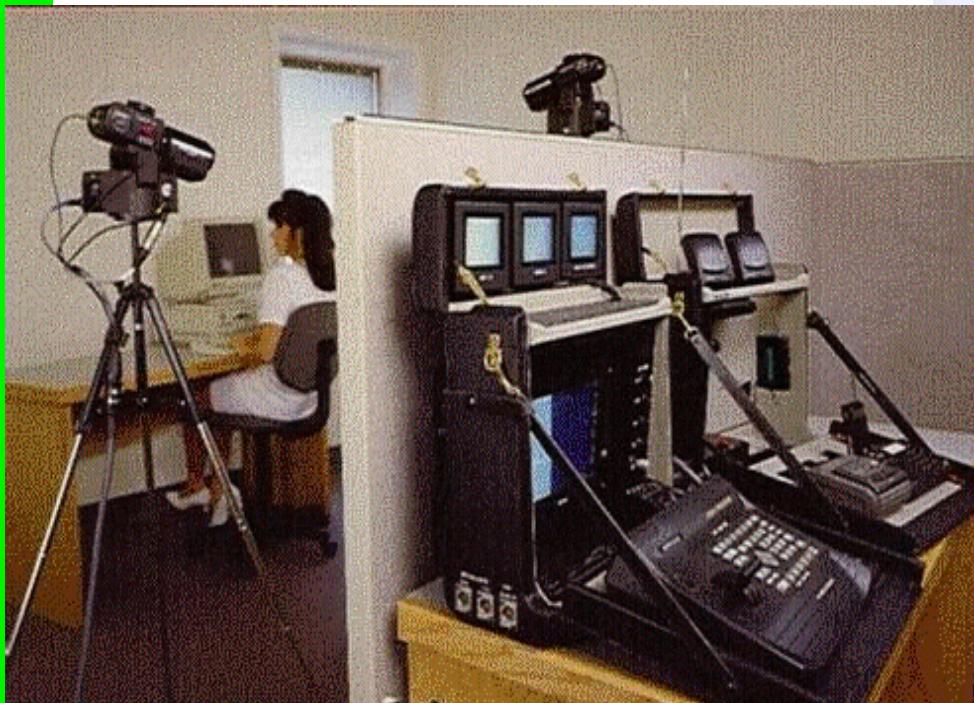
A fallacious Dichotomy

- Intimate cause/effect relationship between the internal software attributes and usability
- E.g. If a system must provide continuous feedback, the developer should consider this when designing time-consuming system functionalities

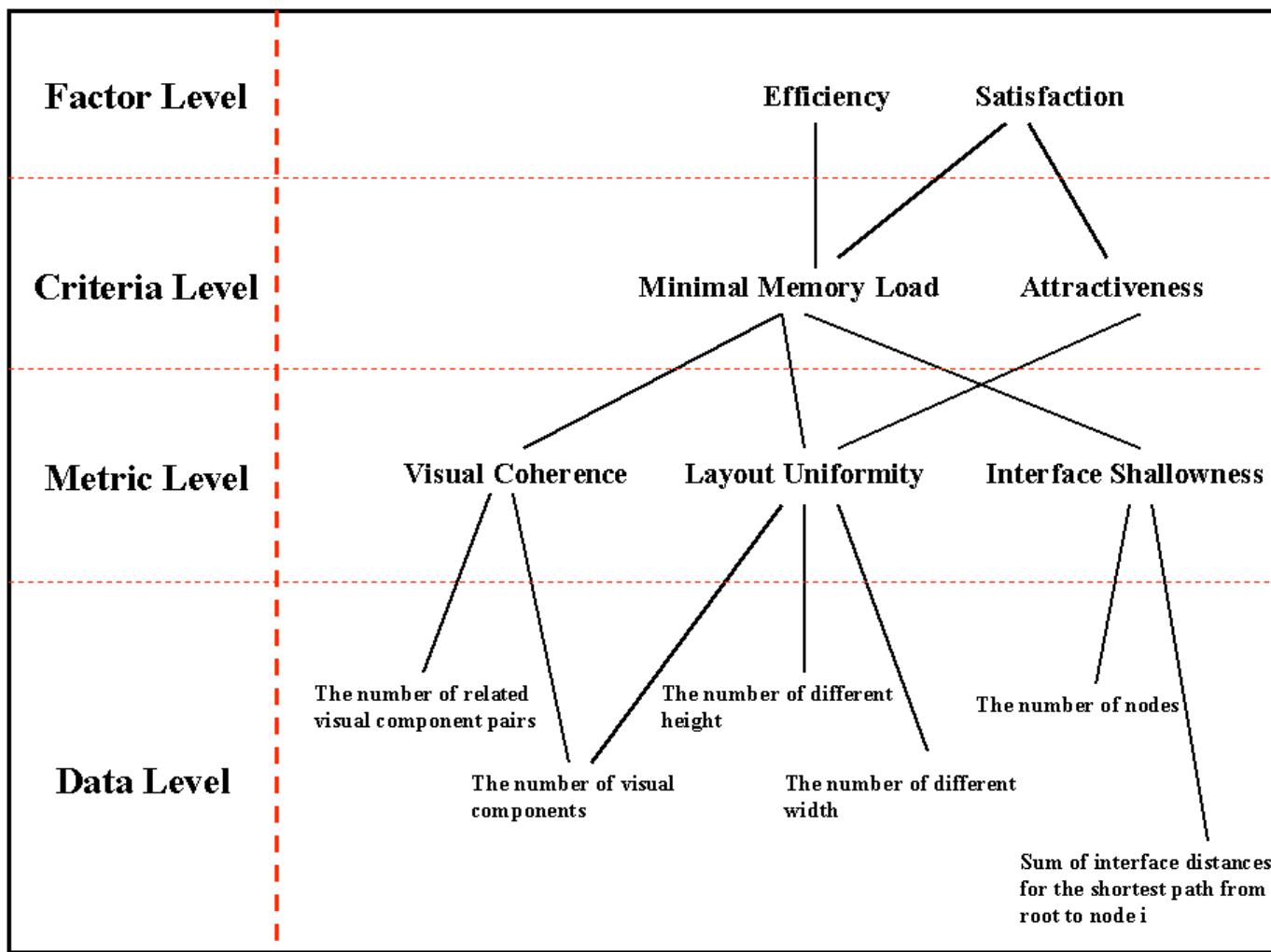


How Usability can be Engineered?

- Observing and quantifying the user behavior and feedback

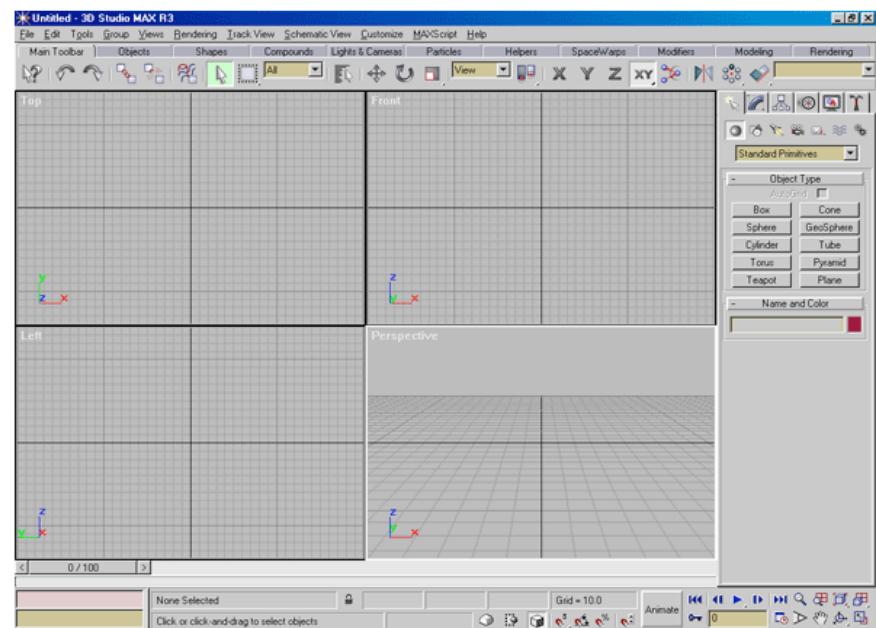
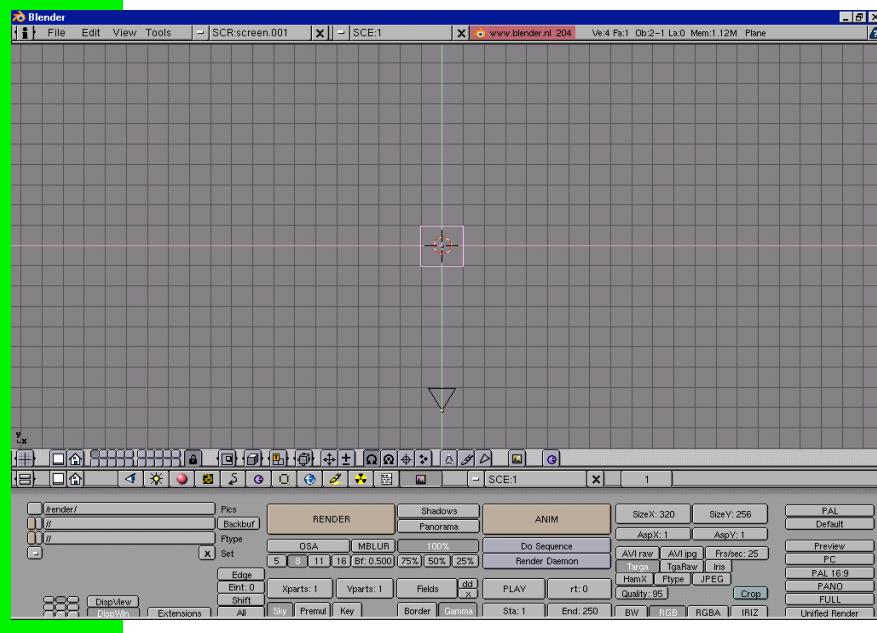


Measurable attributes



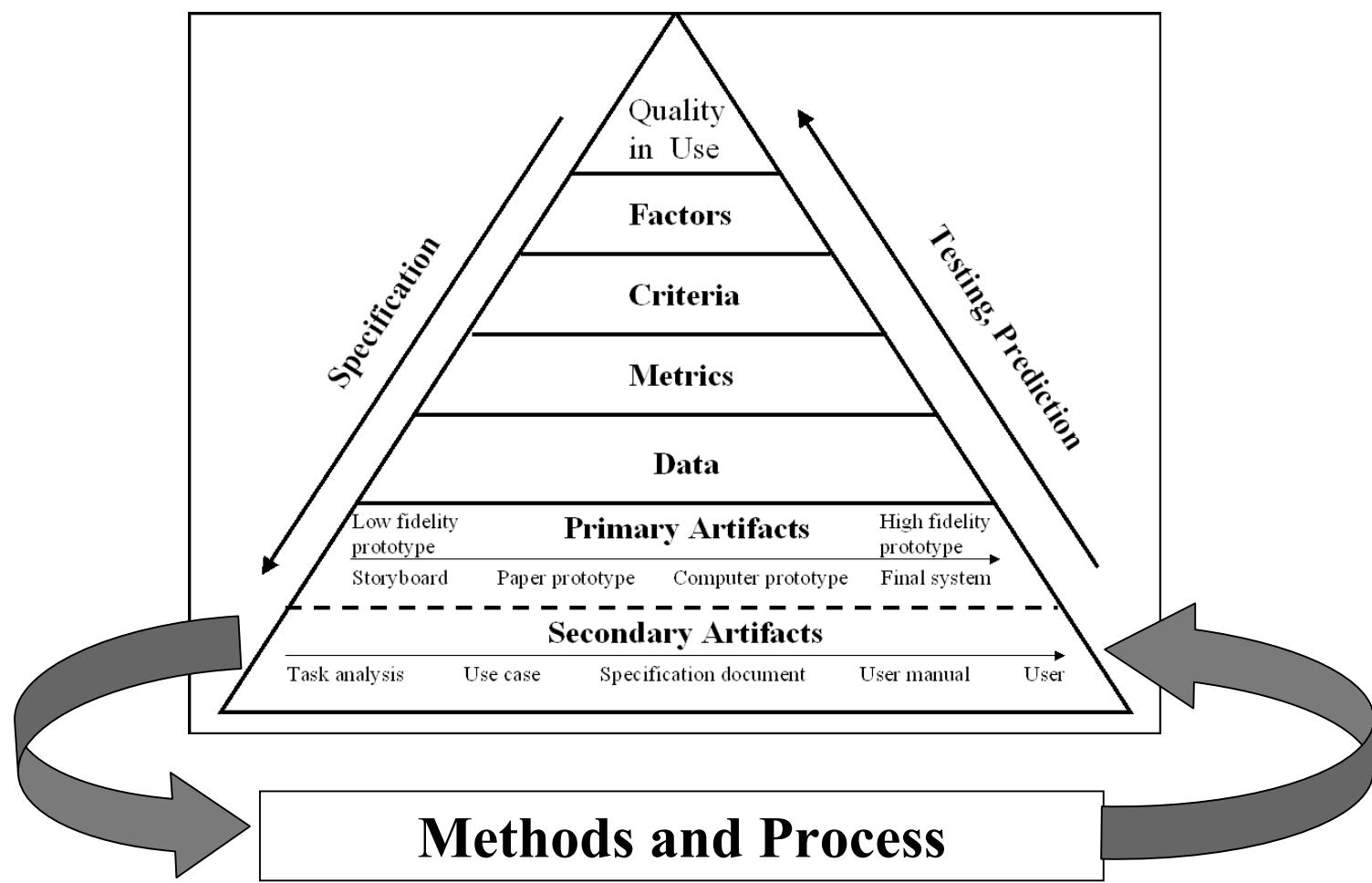
Layout uniformity or how well visual components are arranged

Blender
LU = 81%



3DStudioMax
LU = 63%

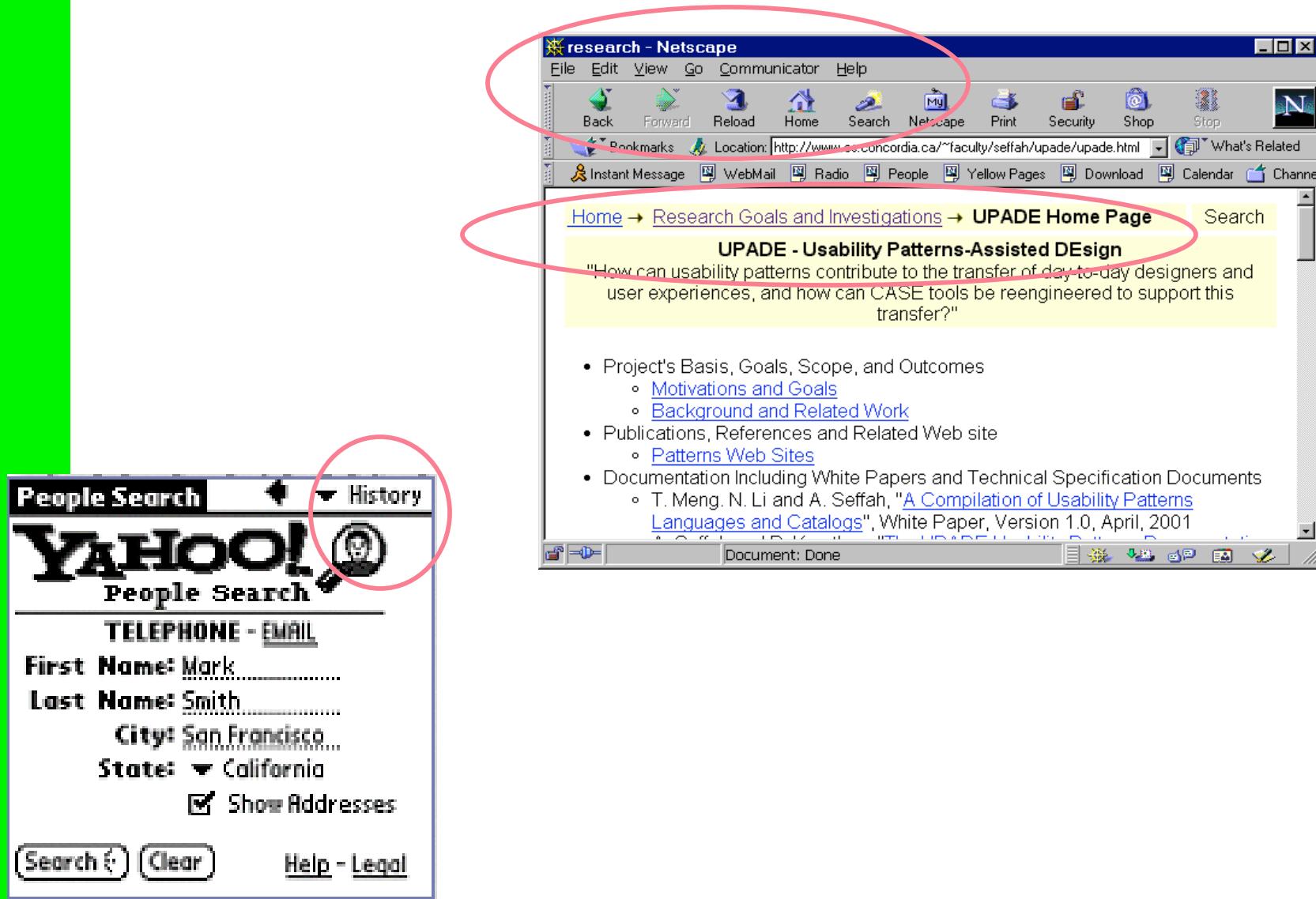
QUIM project: integrated map for usability measurement



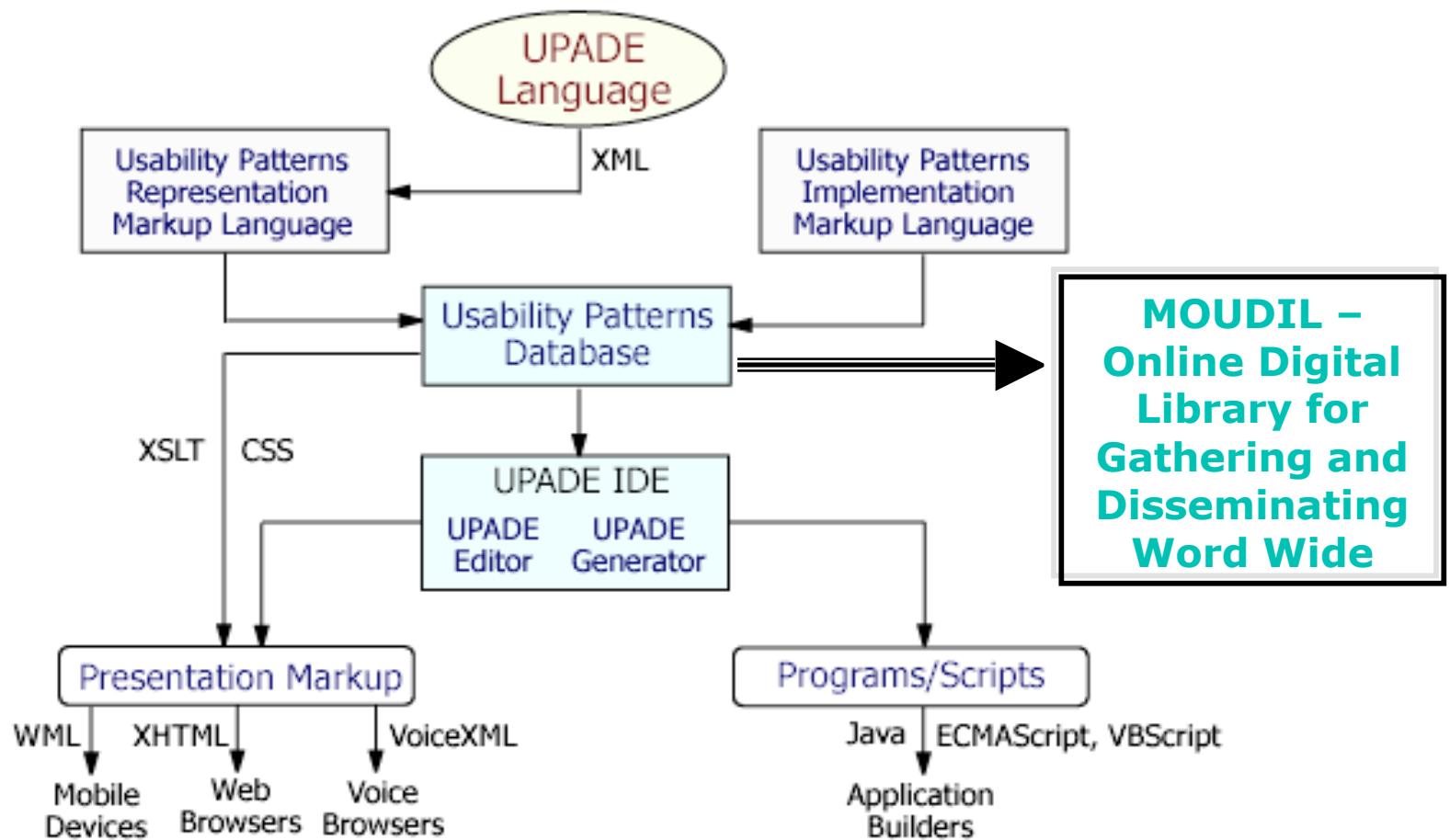
How usability can be engineered?

- Gathering and disseminating user experiences and best design practices
 - ▶ Design guidelines and heuristics
 - ▶ Usability patterns
 - UPADE Project
 - MOUDIL

Convenient Toolbar Pattern



UPADE project: usability patterns-oriented designs



How usability can be engineered?

- **The most fundamental principle:** Early involvement of users and a strong commitment to usability and user requirements
 - ▶ Who are the future users?
 - ▶ What kind of tasks they need to accomplish?
 - ▶ In which environment the software will be used?

More Investigations

- User experience (Usability, User, Usage) is just one point!!!
- CASE tools are hard to master and learn
 - ▶ Initial training (Seffah, 99), transition to new features and versions (Seffah, 98),
- Programmer experience (Programmer, Implementation, Learnability)
 - ▶ Human-centric CASE tools (Rilling and Seffah, 02)
- Maintainer experience (Programmer, Maintenance, Understandability)

Merci ! Thank you!

Ahmed Seffah

- Yes, usability (sorry, quality in use, human experiences) is an excellent business for software engineers
- Yes, human experiences can be engineered

- How, it takes a multidisciplinary team for observing human, analyzing their experiences and based on these experiences we - software engineers – can build more human-centered software technology

Visit our Website to know more
hci.cs.concordia.ca