
Integrating HCI and Usability into Software Engineering:

The Imperative and the Resistance

Timothy C. Lethbridge, PhD, ISP, CSDP

Presentation to CapCHI

My Background

Professor of Software Engineering and Computer Science at the University of Ottawa

- Research interests: Software Engineering Education, Software Tools (with a focus on making them usable), Modeling.
- Teaching: Software Engineering, HCI/Usability, software modeling
- My PhD ('94) was on making knowledge management usable

Industry:

- I worked in the past for Nortel, and the Government of New Brunswick
- Past research with Mitel
- Ongoing research with IBM

My Background

My Book: Object-Oriented Software Engineering: Practical Software Development Using UML and Java

- Has a Chapter on UI design and usability

Key professional involvements:

- IEEE/ACM SE-2004 (curriculum development)
- IEEE-CS: CSDP (professional certification)
- Conference on Software Engineering Education and Training
- CIPS: Computer Science Accreditation Council

Agenda

1. The imperative

2. Usability and related qualities

3. The resistance

4. Some partial solutions

- Bringing the SE and HCI communities closer
- Improving education and individual certification
 - SWEBOK
 - SE-2004
 - CSDP
 - Accreditation
- Enabling corporations to systematically improve their capabilities

5. Conclusions

An Assertion

Lack of usability is the most critical problem facing software engineering

In other words, the biggest gains in

- software quality
- productivity
- cost-reduction
- user satisfaction
- profitability, etc.

... would come from focusing on usability and related issues.

Evidence for the assertion

Failures of projects due in significant part to usability

- E.g. FAA air traffic control systems

Observations of ‘great’ software vs. not-so-great

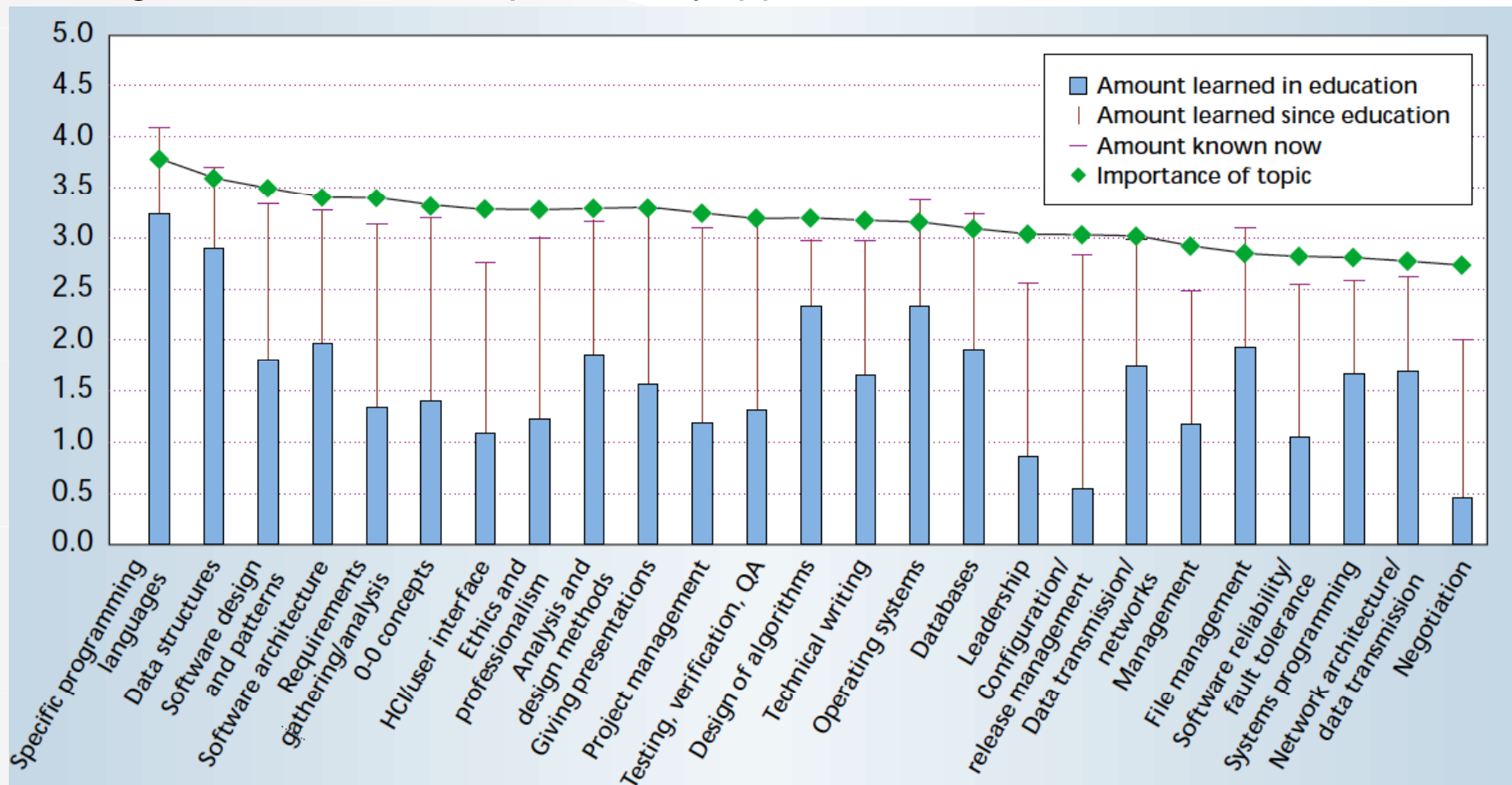
- Great:
 - Google (most of its products)
 - Excel (some of it)
 - Mac-OS (much of it)
- Not-so-great:
 - Windows (key parts of it)
 - Software development environments (many aspects)
 - Web sites (far too many of them)
- I have observed many users formally and informally

My survey of practitioners (next slide)

Analysis of benefits and cost savings

Practitioner survey about 75 topics taught to computer science students

Lethbridge, T.C. (2000), "What Knowledge is Important to a Software Engineer?", *IEEE Computer*, May, pp. 44-50.



More on the survey

HCI was second in terms of “knowledge gap”

- Where importance most exceeds current knowledge

The top 5 (out of 75)

- Negotiation
- HCI/user interfaces
- Leadership
- Real-time system design
- Management

I believe not much has changed since 1998

Cost savings from focusing on HCI and usability

Reduced training and support costs

Reduced time to learn the system

Greater efficiency of use

Reduced costs by only developing features that are needed

Reduced costs associated with changing the system later

Better prioritizing of work for iterative development

Greater attractiveness of the system, so users will be more willing to buy and use it

Hmm.

But isn't X the biggest problem?

Where:

X=Lack of Security

- A key problem, but it is widely recognized now that you have to deal with it
- Often caused by poor usability; solutions contribute to poor usability

X=Poor Project Management

- Huge problem for development managers; but usability affects us all in a more pervasive way

X=Low Reliability

- Also a key problem, but more 'manifest' than lack of usability
- Well accepted as a problem that must be tackled

Lack of usability is the bigger problem because it is not recognized

But we know how to do it right!

We can make brilliantly usable products!

- Exemplars abound

We have methods!

- “User Centered Design”
- Or now ... “Outside-In Development”

And material to learn from

- E.g. Useit.com <http://www.useit.com/>

Agenda

1. The imperative

2. Defining terms: Usability and related qualities

3. The resistance

4. Some partial solutions

- Bringing the SE and HCI communities closer
- Improving education and individual certification
 - SWEBOK
 - SE-2004
 - CSDP
 - Accreditation
- Enabling corporations to systematically improve their capabilities

5. Conclusions

Types of quality

Manifest: Obvious from an initial look at software

- Feature richness
- Graphical appeal

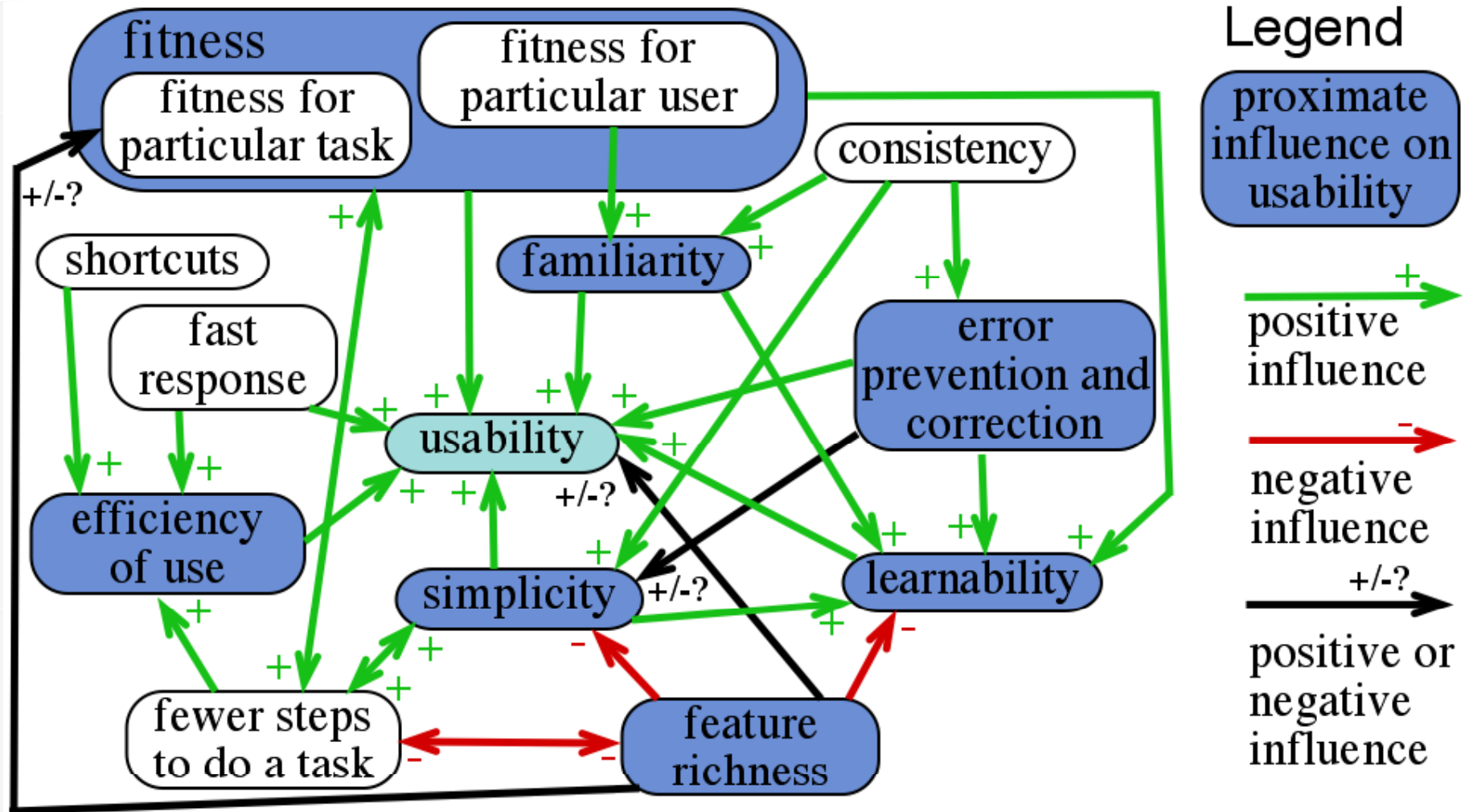
Semi-latent: Only become obvious with deployment and use

- Usability
- Consumability
- Availability
- Robustness
- Security

Latent: Primarily affect long-term product health

- Maintainability
- Scalability

Usability: Ability to get a task done easily



Consumability - Closely allied to usability

Ability for customers to get rapid return on investment.

Factors include

- all aspects of usability
- installability / uninstallability
- lack of inter-product dependency/interaction
- low footprint: lack of run-time resource consumption
- backward-compatibility
- integratability
- appropriate prioritization of iterations and features to yield business value
- low purchase cost

Agenda

1. The imperative
2. Usability and related qualities
- 3. The resistance**
4. Some partial solutions
 - Bringing the SE and HCI communities closer
 - Improving education and individual certification
 - SWEBOK
 - SE-2004
 - CSDP
 - Accreditation
 - Enabling corporations to systematically improve their capabilities
5. Conclusions

Sources of resistance to adopting usability practices

Primarily: Inadequate education of students, practitioners and managers

Which leads to:

- Lack of introspection ability of software developers
 - Unable to think like users
- Lack of integration of usability into core development processes
- Persistent beliefs
 - Most software is usable enough
 - Usability can be fixed at the “end”
 - Usability should be left to the HCI experts

Causes of the problems in education

Faculty don't have the background either

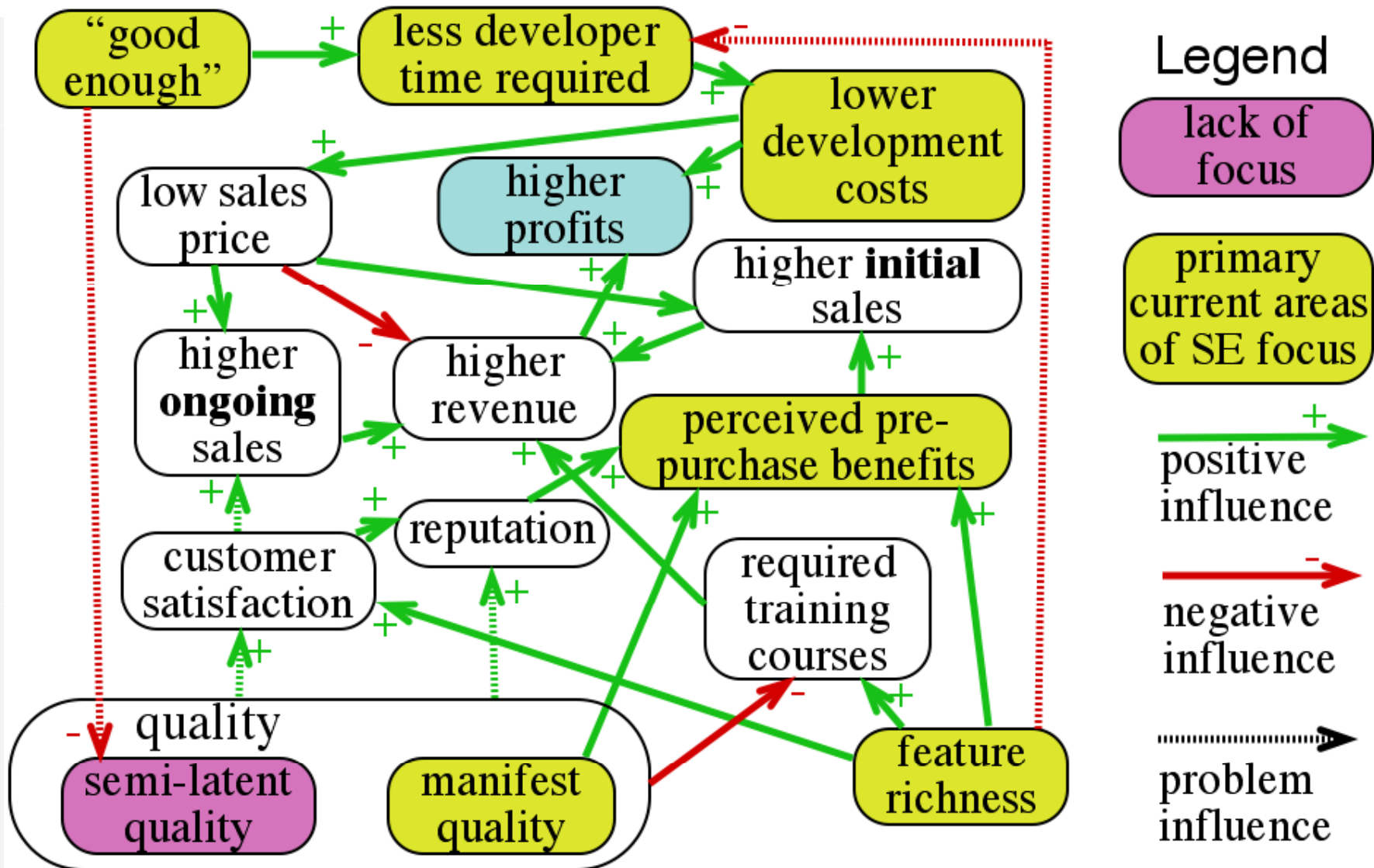
The field of HCI is seen as distinct and separate

The field is perceived as too soft: There is a tendency to focus teaching and research on hard deterministic areas

There is little industry push for research in this area

There is little employer pull for greater education

Software engineers tend to place focus in other areas



Some partial solutions

The SE and HCI communities need to work more closely together

- Core UI development and evaluation topics should be considered jointly part of the two fields

Improve education by ensuring HCI permeates curriculum models, certifications, accreditation, etc.

Enable corporations to be certified for their usability capabilities

- Would help “pull” education up by creating a need for professionals
- Clients might come to learn that better software comes from such companies

Should aspects of HCI be considered also integral to Software Engineering?

I argue yes:

- Usability should have no special status as compared to reliability, efficiency, maintainability, etc.
- Design involving users and their needs must *drive* software development

Should aspects of HCI be considered also integral to Software Engineering?

Yes, but this does not preclude HCI specialists

– Analogies:

- A software engineer must be capable of designing architectural elements for storing data and information
 - But there will always need to be database specialists
- Similar argument can be made for
 - Security experts, performance experts, requirements experts, etc.

Core UCD techniques can be part of the SE field *and* part of a broader HCI field

Agenda

1. The imperative
2. Usability and related qualities
3. The resistance
4. Some partial solutions
 - Bringing the SE and HCI communities closer
 - **Improving education and individual certification**
 - **SWEBOK**
 - **SE-2004**
 - **CSDP**
 - **Accreditation**
 - Enabling corporations to systematically improve their capabilities
5. Conclusions

“Software Ergonomics” in SWEBOK

SWEBOK = Software Engineering Body of Knowledge

- An IEEE-CS product published in 2004
- Development: 1998-2004
- I was involved merely as a reviewer, but have worked closely with the editors on a number of projects
- www.swebok.org

Software Ergonomics was relegated as a Related Discipline

- Along with Computer Engineering, Computer Science, Project Management, Management, Mathematics, Quality Management and Systems Engineering

“Software Ergonomics” in SWEBOK

Summary of topics:

- Cognition
- Formal methods in cognitive science
- Cognitive AI: Learning and Reasoning
- Information extraction from speech and text
- The nature of HCI
- The use and context of computers
- Human-machine fit and adaptation
- Human characteristics
- Computer system and interface architecture
- Dialog architecture
- Development techniques: Design, implementation, evaluation

“Software Ergonomics” in SWEBOK

How was the 2004 list chosen?

- Larger list compiled from several sources
- Refined by 3 people:
 - They were asked to indicate which of the topics should be known by a software engineer
 - Topics only included if 2 or 3 agreed

I argued for integration of some HCI and usability concepts in the SWEBOK core

- I was a lone voice in the wilderness

SWEBOK: The Next Generation

For the next iteration, the editors are open to incorporating some human factors in the core

- The more voices encouraging this, the better

SE2004: Background

Joint project of IEEE-CS and ACM

- sites.computer.org/ccse/

Overall project has created curriculum recommendations in

- Computer Science
- Software Engineering
 - Formerly “CCSE”
 - Computing Curriculum - Software Engineering
 - <http://sites.computer.org/ccse>
 - Involved hundreds of international reviewers
- Computer Engineering
- Information Systems
- Information Technology

Main components of SE2004

Ch. 3 contains ‘student outcomes’, key to producing effective curricula

Ch. 4: The knowledge that should be taught

- “Software Engineering Education Knowledge” (SEEK)

Ch. 5: General guidelines for SE Education

Ch. 6: Curricula: Sequences of courses and course outlines

SEEK (SE-2004 Chapter 4) Knowledge Areas

Computing Essentials (172h / 35%)

Mathematical and Engineering Fundamentals (89h / 18%)

Professional Practice (35h / 7%)

Software Modeling and Analysis (53h / 11%)

Software Design (45h / 9%)

Software Verification & Validation (42h / 9%)

Software Evolution (10h / 2%)

Software Process (13h / 3%)

Software Quality (16h / 3%)

Software Management (19h / 4%)

Human factors in SE2004

Coverage is spread through the Knowledge Areas

- Computing Foundations
 - Basic principles (e.g. providing error messages); basic UI construction tools
- Professional Practice
 - Basic psychological principles; communication; social issues
- Modeling and Analysis
 - Analysing quality requirements, including usability
 - User centred design
 - Requirements elicitation techniques focused on users

Human factors in SE2004

Coverage continued

- Software Design
 - Design for usability
 - Interface design (whole unit)
 - HCI principles
 - Modes, navigation
 - Visual design, coding
 - Response time, feedback
 - Localization and internationalization
 - HCI design methods
 - *Multi-media*
 - *Metaphors and conceptual models*
 - *Psychological issues*

Human factors in SE2004

Coverage continued

- Verification and validation
 - Testing for usability
 - User interface evaluation techniques
 - Heuristic evaluation, cognitive walkthroughs, observation
 - Web usability
 - *Formal HCI experiments*
- Quality
 - Usability
- Process, Evolution and Management areas:
 - No specific coverage

Human factors in SE2004

Result of all the above:

- A course in Usability / HCI is mandatory for all SE students following the curriculum

It was a battle to get HCI included to the above extent!

Still not included enough in CS education

CSDP

Certified Software Development Professional

- Requires a degree (or equivalent), 4 years experience and exam

Certification exam offered by IEEE-CS

- I am a CSDP holder and am on the development committee

Domains:

1. Process management - 20%
2. Initiating, conceptualizing, planning - 21%
3. Doing, developing - 25%
4. Checking - 19%
5. Operating, maintaining, retiring - 13%
6. General (e.g. math and computing) - 2%

Human factors in CSDP

Mention in two knowledge areas out of 51

- Executable UI models [$<1\%$]
- Human computer interaction (e.g., user interface standards, ergonomics, task analysis) [3%]

No mention in list of tasks

Human factors in CSDP

Next Steps:

- An entry-level CSDP is being developed
 - I am on the committee and am working to incorporate a requirement for HCI/Usability
- Core CSDP will be due for revision soon

Accreditation

Engineering accreditation by the Canadian Engineering Accreditation Board pays no heed to human factors

- They even rejected our inclusion of Cognitive Psychology as a basic science for software engineers

CIPS accreditation of CS and SE programs does not require the topic either

- This is something I plan to work on as a board member

Agenda

1. The imperative
2. Usability and related qualities
3. The resistance
4. Some partial solutions
 - Bringing the SE and HCI communities closer
 - Improving education and individual certification
 - SWEBOK
 - SE-2004
 - CSDP
 - Accreditation
 - **Enabling corporations to systematically improve their capabilities**
5. Conclusions

Proposal: The Model for Usability Maturity (MUM)

A 5-level CMM-like categorization of organizations

- Based on their interaction with human end-users in software development
- (Each level beyond 1 builds on the previous levels)
- Level 1: Haphazard
- Level 2: Defined input from users / usability awareness
- Level 3: Iterative interaction with users / design for usability
- Level 4: Controlled and measured involvement of users
- Level 5: Continually improving usability

Level 1: Haphazard

If you can get users to use the system, then it is considered good enough!

Level 2: Defined input from users / usability awareness

Users involved in requirements reviews

- **Feedback from users** at reviews is incorporated into the next stage

Design team members have basic training in usability

Design team adheres to usability standards for look and feel

Reuse of well-understood controls, styles etc.

Level 3: Iterative interaction with users / design for usability

Users actively involved in decision making

Use case / task analysis

Competitive analysis

Design with careful attention to usability guidelines

Usability design decisions are carefully analyzed and the decisions are recorded

- E.g. options analysis, tables of pros and cons

Level 3: Continued

User input and feedback with repeated prototypes

- From paper prototypes to functional prototypes

Informal qualitative observations of users or heuristic evaluations

Discount usability engineering

Level 4: Controlled and measured involvement of users

User input and feedback at all stages

Design team has trained usability experts or cognitive scientists

Careful cost-benefit optimization including usability

Setting of quantitative usability objectives

- For all parts of the system that will be subject to regular or critical-situation use
- For learnability, efficiency of use, etc.

Level 4: Continued

User-centered design **with formal usability studies**

Measurement of usability so as to determine progress towards goals

Level 5: Continually improving usability

Active development of new UI understanding and innovations

Formal experiments **to validate new UI modalities**

- metaphors, controls, widgets, styles etc.

Anthropological studies **of human tasks**

- to enable optimized UI design

Scientific study **of**

- Users and their work practices
- Usability and the software engineering process

Potential benefits of MUM

It provides a visible framework companies can use to incrementally improve

If even a few companies adopted it and were certified, they might have a competitive advantage in some situations

Note: This is only a rough proposal

- It would need lots of work to become a reality**

Conclusions

Usability and HCI are critical factors for the overall improvement for software

- I say: “The most critical”

There is a lack of education in the field, and the resulting lack of knowledge/experience in industry

There is progress:

- SE2004, Next iteration of SWEBOK

Potential for the future

- CSDP, Accreditation awareness, corporate certification

What can you, as HCI professionals, do?

1. Raise your awareness of the link between SE and HCI

- A pure traditional HCI course needs to be supplemented by a focus on integrating with SE

2. Speak out in favour of improvements to

- educational standards
- HCI courses
- Accreditation
- Etc.

Thank you!

Questions?