CS 5714 (also ISE 5714)  
Usability Engineering

Dr. Mary Beth Rosson  
Department of Computer Science  
rosson@vt.edu, x6470  
office hours: T/H 9:30-11, McBryde 501

GTA: Vlad Rynkov, vlad@csgrad.cs.vt.edu  
Web: http://ei.cs.vt.edu/~cs5714
what is it we are trying to “engineer”?
usability “in the large”

- the system must do something useful
  - simplify or enhance an existing set of tasks
  - or leverage new technology to invent new tasks, maybe even new goals
- and it must do it in a usable way
  - measured on several different dimensions (Shneiderman):
    - ✔ ease of learning
    - ✔ good retention
    - ✔ speed of performance
    - ✔ few errors
    - ✔ high satisfaction
design versus programming

◆ user *behavior* can be analyzed, designed, tested, without any programming
  – many options for describing, modeling, building low-tech prototypes
  – even functional prototypes may be poorly constructed from the perspective of software quality

◆ ultimately software must implement the interaction
  – anywhere from 50% to almost 100% will be UI code
  – usability objectives become *non-functional requirements*
  – how to specify/document what HCI work has done?
  – research issue: how to better integrate HCI design with software construction and its iteration
barriers to usability engineering?
the engineering process

needs and constraints of existing situation

an open-ended process of knowledge discovery and iterative refinement

useful and effective system
H&H: it’s all about evaluation!

- Deployment & Maintenance
- Task analysis / Functional analysis
- Software Development
- Requirements specification
- Conceptual design / Formal design
- Prototyping
- Evaluation
the many faces of evaluation...
moving through a “design space”
—asking questions, i.e. *evaluating*, to guide your path—

◆ observing and interpreting the current situation
  – interviews, videos/notes, artifact analysis, usage logs, diaries
◆ modeling users’ (or organizations’) existing behavior
  – task analysis, simulations, regression models
◆ inventing new situations (even if just a small delta)
  – “what if” analyses, tradeoffs, cost-benefit or feasibility
◆ implementing new situations (prototype or for real)
  – pragmatic constraints, completeness, validation
◆ refining, fielding, maintaining the new situations
  – formative and summative user testing, then back to the beginning
### Scriven’s (1967) Classic Matrix:

—evaluation goals versus evaluation methods—

<table>
<thead>
<tr>
<th></th>
<th>Formative</th>
<th>Summative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intrinsic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pay-off</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
“If you want to evaluate a tool, say an axe, you might study the design of the bit, the weight distribution, the steel alloy used, the grade of hickory in the handle, etc., or you might just study the kind and speed of the cut it makes in the hands of a good axeman.”
Studies of Work:
user characteristics, tasks, artifacts, organization goals

Business Analysis:
stakeholders, costs, schedule, business objectives

Science & Technology:
prior art, new technology, science base, guidelines

needs

vision

DESIGN:
functionality, conceptual model, user interaction

Formative intrinsic:
user and task models, tradeoffs, rationale. UI heuristics

Prototype:
vertical or horizontal, or a system version

Deployment:
marketing, user feedback, profit

Formative pay-off

Summative pay-off

Abstraction & Cumulation

Copyright © 1999 Mary Beth Rosson

Usability Engineering, Jan 19: 12