End Users Who Meet Their Own Requirements

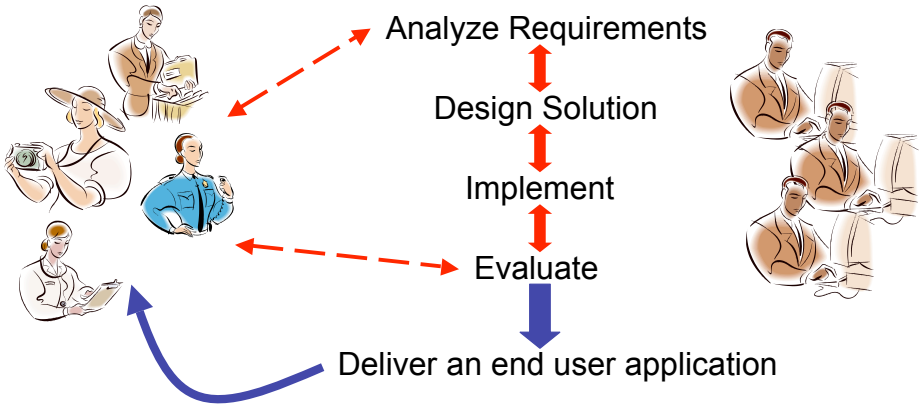
Mary Beth Rosson
Center for Human-Computer Interaction
College of Information Sciences & Technology
The Pennsylvania State University

13 September 2006 — Minneapolis, Minnesota

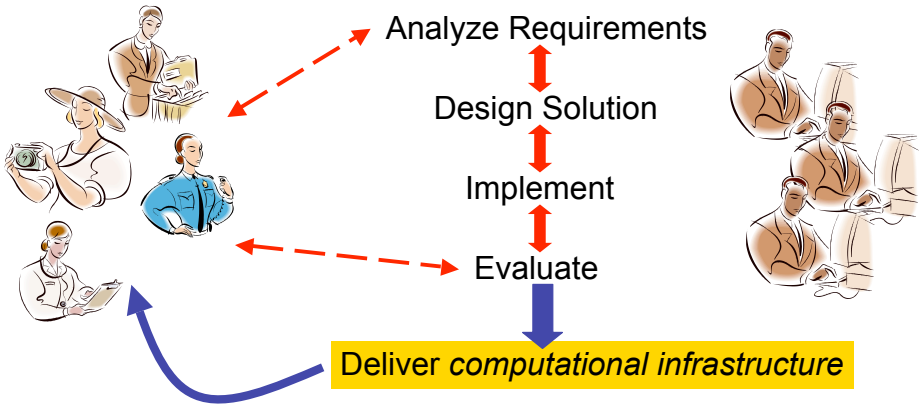
Overview

- Motivation
- Examples
- Challenges and opportunities
  - Usability
  - Universal access
  - Quality
- Final words

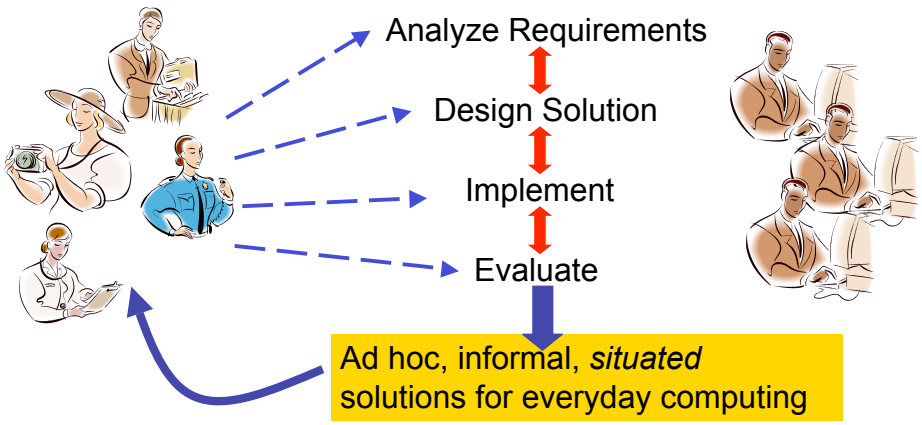
Tried and true

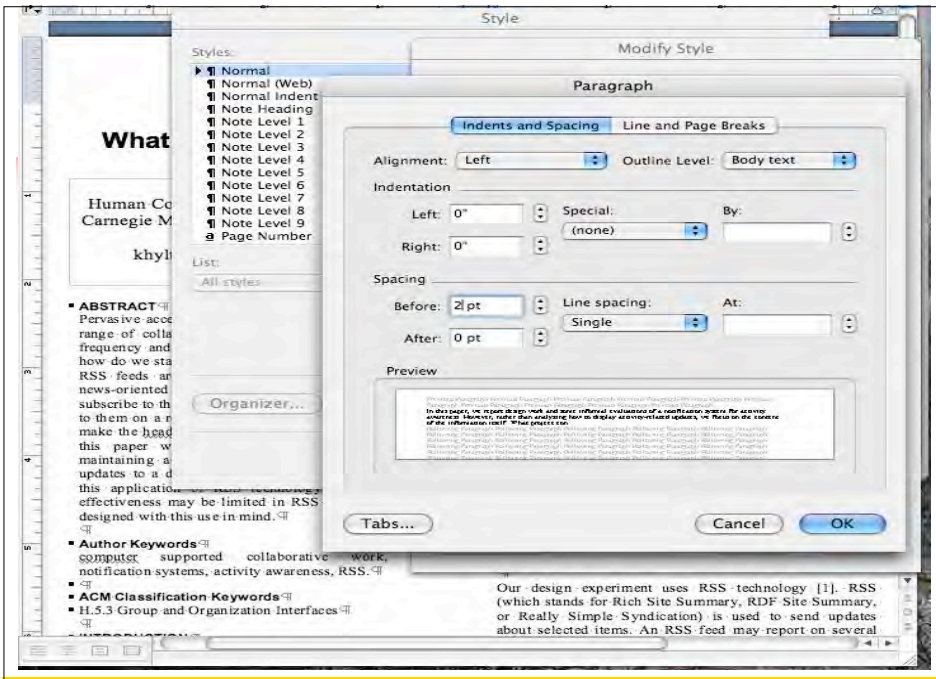


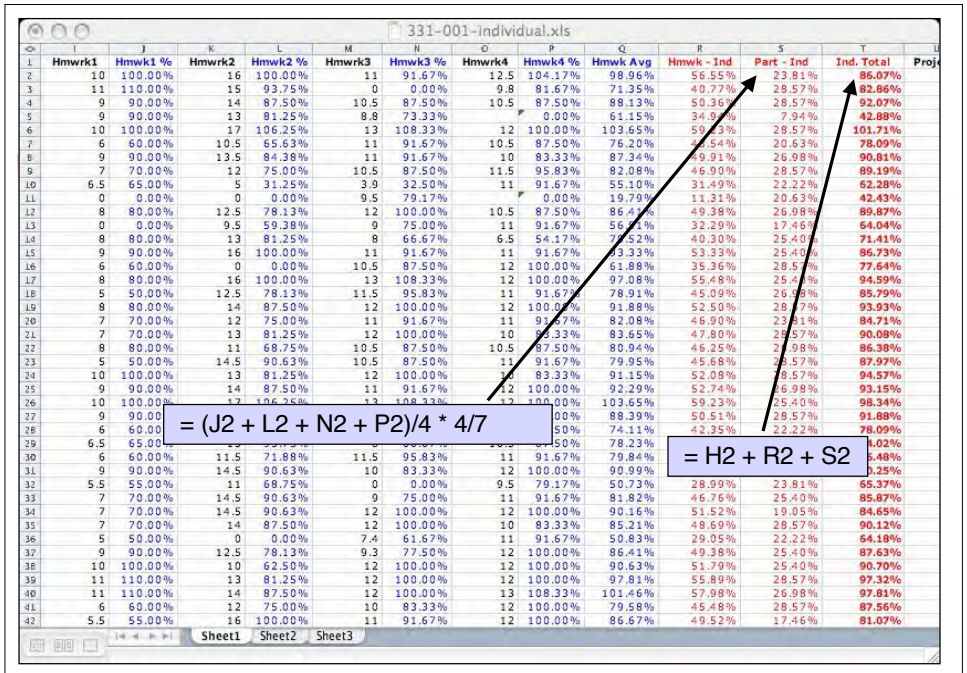
A more modern view



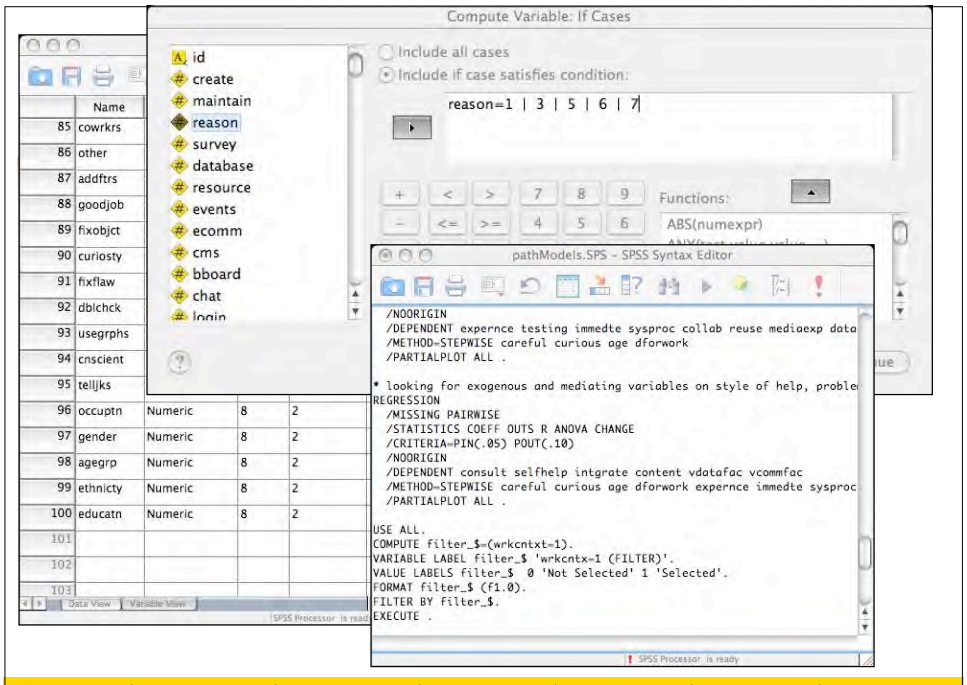
A more modern view



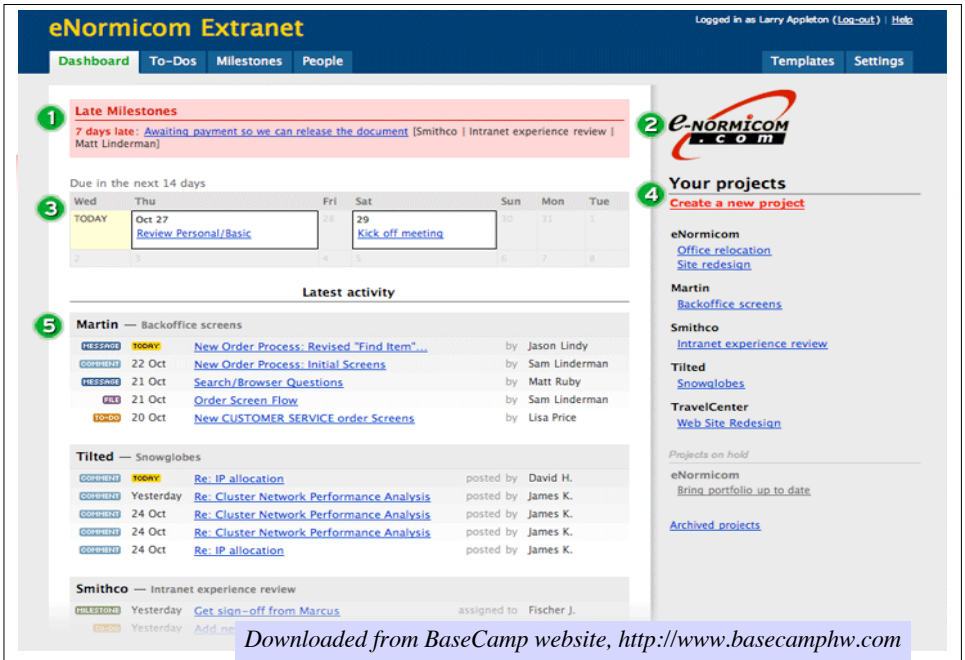




Motivation | Examples | Challenges | Usability | Access | Quality | Final words



Motivation | Examples | Challenges | Usability | Access | Quality | Final words



Downloaded from BaseCamp website, http://www.basecamphw.com

How do we develop the necessary *socio*-technical infrastructure to help end users meet their own requirements — effectively and safely?

- collaborative, ad hoc problems and task goals;
- expectation that computers are *tools* that should support his/her activities.

The opportunities are great

- Personal development
  - Raising the level of computing literacy
- Feelings of control and self-efficacy
  - Computers as flexible tools
  - Ability to *appropriate* IT to actual needs
- Economic benefits
  - Short-circuit software lifecycle
  - Timely production of informal solutions

But so are the challenges!

- Usability of the tools and techniques
  - Design and programming are hard!
- Universal accessibility
  - Familiar set of IT diversity concerns
  - Plus, an aging population of end users
- Quality of the end-user-created solutions
  - Even when created to meet informal needs, “bad” software is still bad

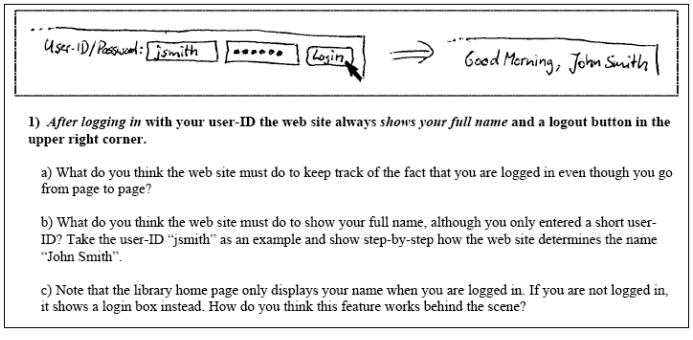
Building usable tools

- Majority of usability practice focused on traditional application design
- Psychology of programming, end user programming (EUP) are open topics
  - Studying design, problem solving is hard!
  - Inherent tradeoff: ease of use *vs.* power
- Example: Natural programming tools

CLICK:

Click, a Lightweight Internet Construction Kit

- Base tools on nonprogrammers' natural mental models when they think about computation



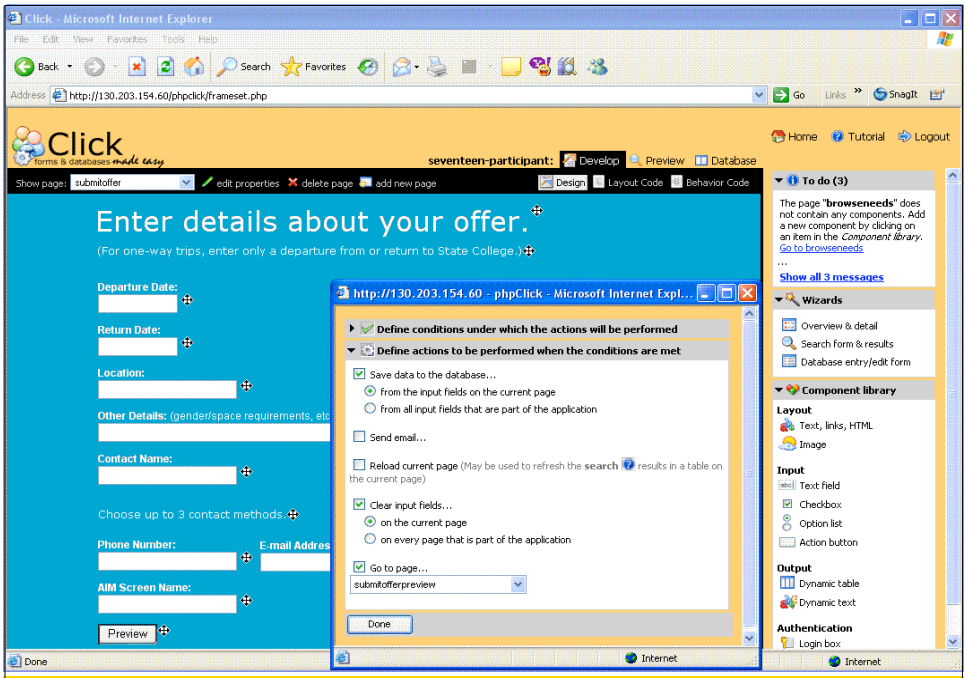
1) *After logging in* with your user-ID the web site always *shows your full name* and a logout button in the upper right corner.

a) What do you think the web site must do to keep track of the fact that you are logged in even though you go from page to page?

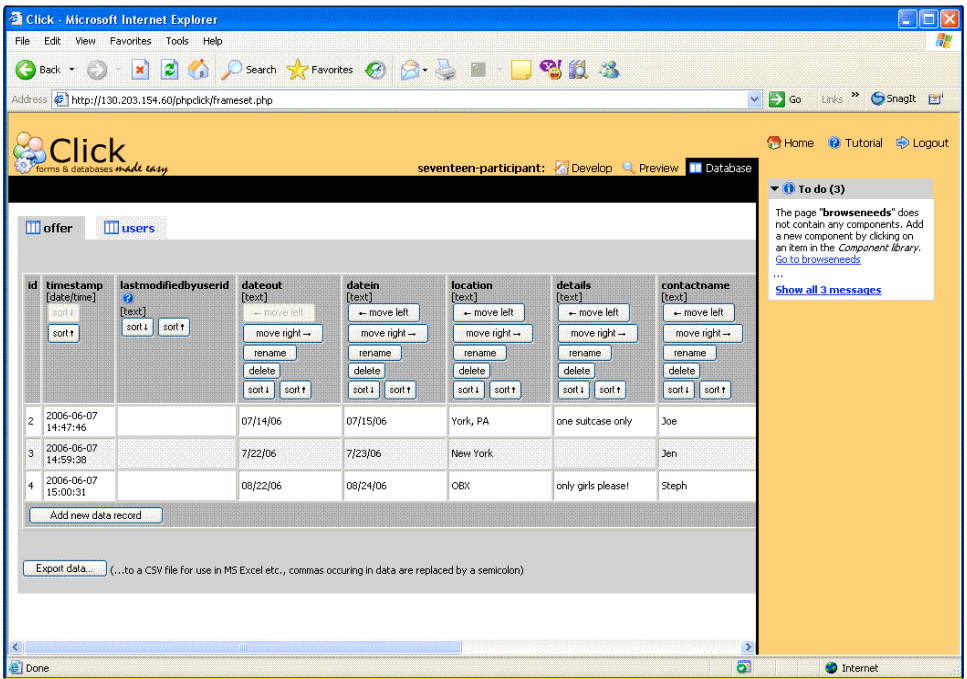
b) What do you think the web site must do to show your full name, although you only entered a short user-ID? Take the user-ID "jsmith" as an example and show step-by-step how the web site determines the name "John Smith".

c) Note that the library home page only displays your name when you are logged in. If you are not logged in, it shows a login box instead. How do you think this feature works behind the scene?

PhD work of Jochen Rode (VT, 2005)

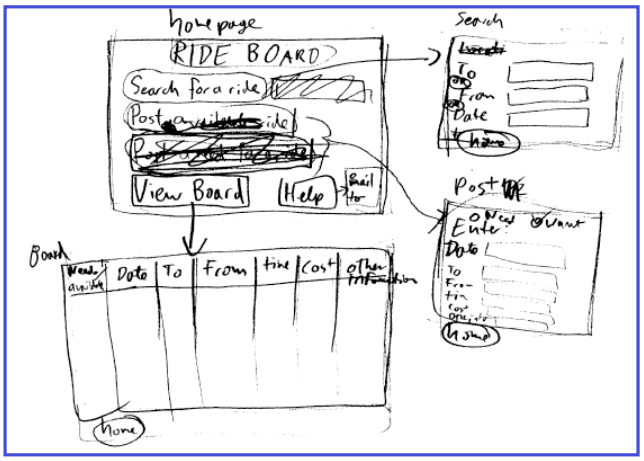


Motivation | Examples | Challenges | Usability | Access | Quality | Final words



Motivation | Examples | Challenges | Usability | Access | Quality | Final words

Up-front design in CLICK?



Ensuring accessibility

- Computing pervades all aspects of the world, not just the "technical" users
  - Participation from full range of user population essential for diverse needs
- CS training, expertise increasingly rare in some significant segments
  - Women, minorities, the elderly
  - Same concerns apply to EUP

Storykit Alice 3D:
Reaching out to girls (Kelleher, 2006)

- Girls “turned off” by math & science starting in middle school
  - What can we do to mitigate?
- Start with what girls like to do - what are their age- & gender-specific “requirements”?
  - Girls this age like to tell stories, especially ones with romance or other social relations
  - So: give them cool tools that let them “program” their own stories

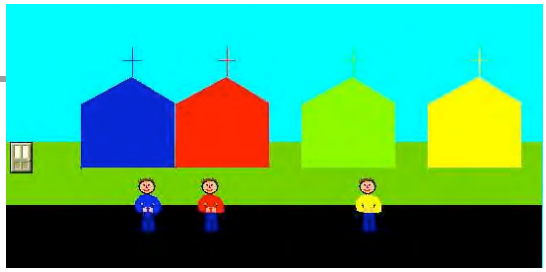
Katelin Kelleher’s Ph.D. dissertation (CMU)

Motivation | Examples | Challenges | Usability | Access | Quality | Final words

Community Sims:
Exploring cross-generation collaboration

- Different segments of the population have distinct interests, skills
  - E.g., kids are eager, experimental while the elderly tend to be wise, cautious
- Can an EUP project serve as a focus for collaborative learning?
  - Kids learn to think about problems, the elderly learn to use a novel tool

Programming in pairs *across generations*



Predefined roles created a smooth design and implementation collaboration

Elders were the simulation **designers**, kids did the **building**

Ensuring quality outcomes

- End users are active, focused on goals
  - Production paradox: Users want to produce, not stop to *learn* something
- They are not software engineers!
  - No overarching process
  - No analysis, design, testing skills
- How much of a problem is this?

Skeptical SW professionals

"… we now have systems on the Web that dilettantes built in their spare time while holding down a job in marketing, accounting, hardware repair, or even medicine. They've given little if any thought to systematic testing, maintainability, design, and yes, security. These systems are available to the entire Internet community—geography and international borders no longer buffer our data from programming mistakes."

—Warren Harrison, Editor of *IEEE Software*, July/August 2004

But also real world costs

"In adopting a new and complex accounting standard in a short period of time, Fannie Mae had to put in place a system and process to capture all open commitments and mark them to market …To implement this standard, Fannie Mae utilized information from its internal automated systems in conjunction with spreadsheets that made additional calculations necessary under the new rule."

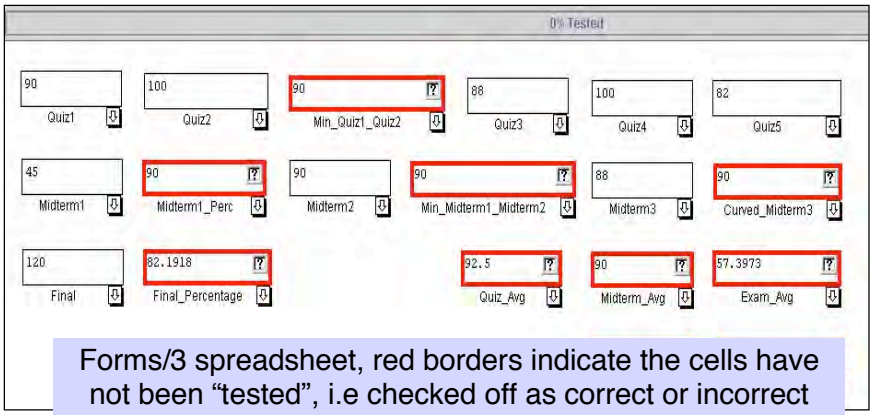
One of the spreadsheets contained a $1.2 Billion error.

—*New York Times*, October 30th, 2003

Who is responsible?

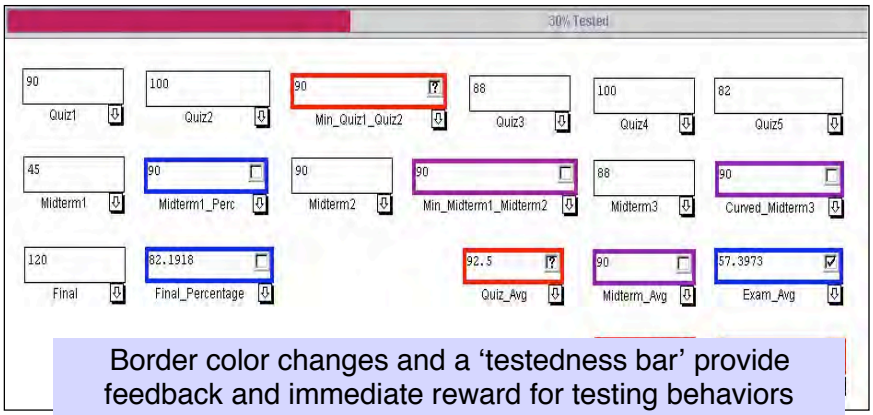
- Software professionals: Smarter tools
  - Need to provide just the right support at just the right time
- End users: Learn new skills, values
  - Embed quality values in learning process
  - Rely on community-based processes to reinforce and develop quality culture

WYSIWYT:
A smarter tool (Burnett et al., OSU)



Forms/3 spreadsheet, red borders indicate the cells have not been “tested”, i.e checked off as correct or incorrect

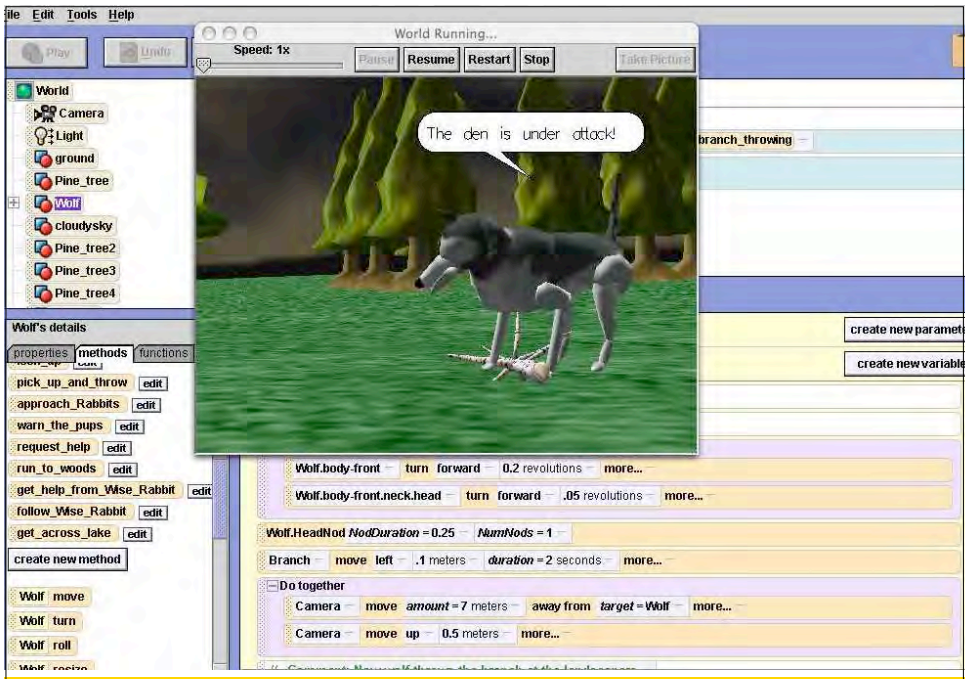
WYSIWYT, cont’d



Border color changes and a ‘testedness bar’ provide feedback and immediate reward for testing behaviors

Learning by debugging:
Teaching kids to test and fix their code

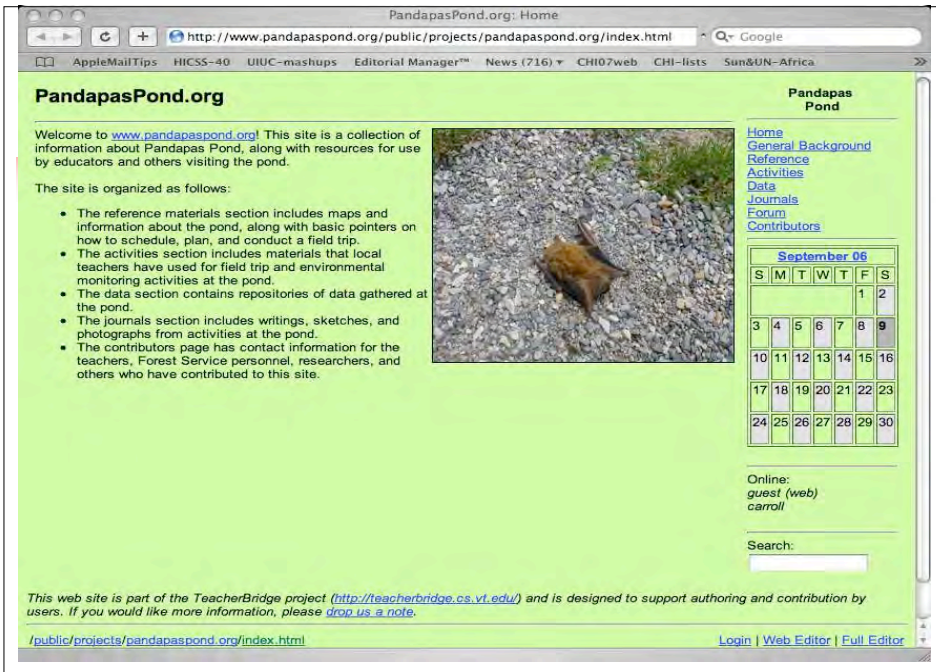
- Attract attention, curiosity by presenting a “broken” example
  - E.g., a story in Alice 3D where characters fall over, fail to launch a ball, etc.
- Invite learners to find and fix errors
  - Commentary that guides, explains
  - Learn from example, but also the skills and confidence for testing and debugging



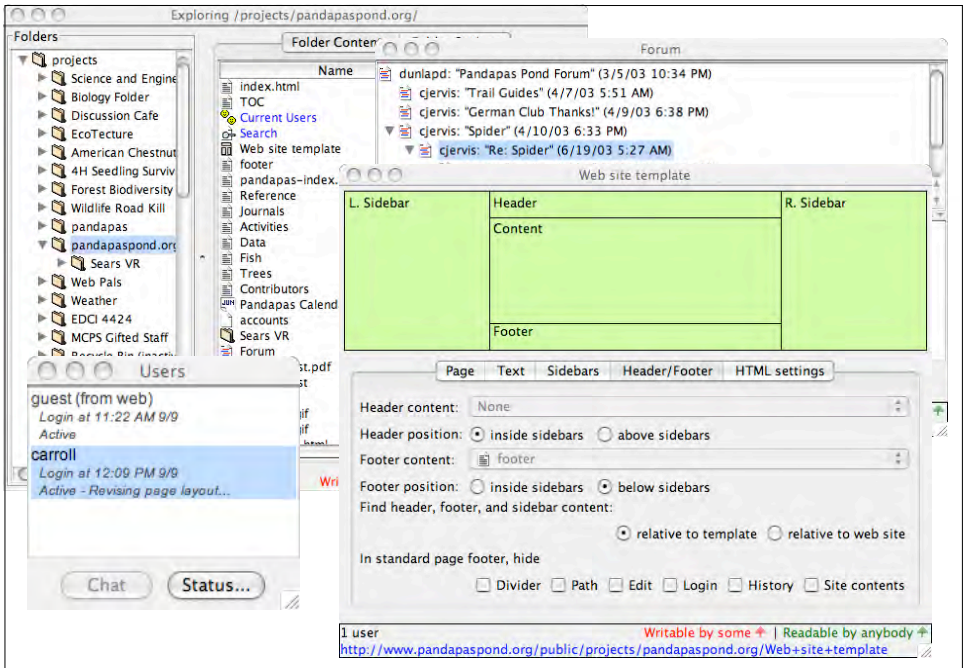
Motivation | Examples | Challenges | Usability | Access | Quality | Final words

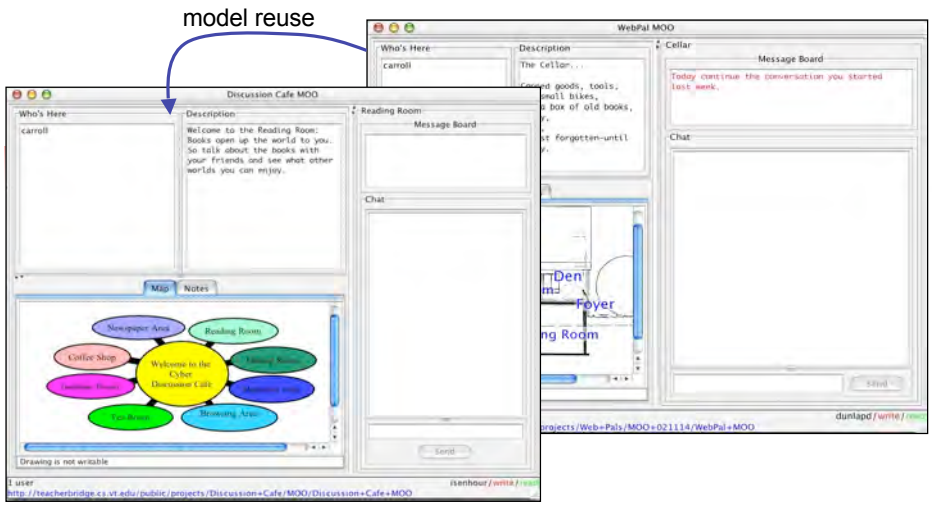
Teacher Bridge:
Building and sharing in a community

- Teachers (and other communities) contain individual with diverse profiles
  - Expertise, motivation, needs, etc.
  - Infrastructure needs multiple entry points
- Online tools enable learning community
  - Advanced, motivated users lead the way
  - Reputation and reuse promote culture of mentoring and collaboration



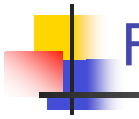
Motivation | Examples | Challenges | Usability | Access | Quality | Final words





A bit like open source with **already-known community**—established practices for mentoring, modeling, reputation-building, and sharing

Motivation | Examples | Challenges | Usability | Access | Quality | Final words



Final words

- End users are already meeting their own software requirements
  - With whatever resources they can find
- We must maximize the upsides of this trend, while minimizing downsides
  - Usability of end user development tools
  - Universal accessibility the infrastructure
  - Tools, training, culture to promote quality

Thank you!

Many collaborators, including John Carroll, Jochen Rode, Philip Isenhour, Jayne Litzinger, Julie Ballin, Cheryl Seals, Tracy Lewis, Craig Ganoe, Dan Dunlap

This work was funded in part by The National Science Foundation (CNS-0353309, CCF-0405612, IIS-0342547); several example projects are part of the EUSES Consortium.