ELEC 875
Design Recovery
and
Automated Evolution

Week 2 Class 1
Empirical Studies
Empirical Studies

- Studies Software Engineers
  ◊ what do they really do
  ◊ what do they really need

- Difficult
  ◊ multiple variables
  ◊ expensive
    - students/professional developers
    - real/artificial projects
  ◊ software engineers like other users are conditioned by their past

- Workshop on Empirical Studies of Software (WESS)
Lethbridge & Singer

• T.C. Lethbridge
  ◊ School of Information Technology and Engineering, University of Ottawa
• J. Singer
  ◊ National Research Council
• Study various companies in the Ottawa area
WESS ’97 Paper

- Understanding Software Maintenance Tools: Some Empirical Research
- Overview paper of Empirical Research
- What is a Tool
  ◊ Used by software engineer to perform a software engineering task
  ◊ hierarchical tools
- 5 Questions
  ◊ What tools and what tasks?
  ◊ What differences do tools make?
  ◊ Why use (or not use) a particular tool?
  ◊ What new tools or improvements to tools?
  ◊ How can tools be introduced to SEs?
WESS ’97 Paper

• Measures
  ◊ What tools are used
  ◊ Number of times each tool is used
  ◊ Elapsed time spend using a tool
  ◊ Goals and tasks for particular usage of a tool
  ◊ List of positive attributes
  ◊ List of negative attributes
  ◊ Time to perform a given task
WESS ’97 Paper

- Data Collection
  - Questionnaires (web based)
  - Interviews
    - General structured interviews
      60-90 minutes, 10 page protocol
    - Regular debriefings (every few weeks)
      30-60 minutes
  - Tool reviews
    30-60 minutes
WESS ’97 Paper

- Data Collection
  - Observation
    - real work (30 minute session)
    - use same tools and techniques?
    - artificial tasks
  - Automated logging of tool use
WESS ’97 Paper

- Data Interpretation
  ◊ Small group of engineers
    - statistical sample?
    - defined process
    - domain specific (complex real-time software)
  ◊ larger and more diverse groups?
WESS ’97 Paper

- Observations
  - Most used tools - editors
  - Second most used tools - searching tools
  - Explore software as much as edit software
  - Wish list - analysis tools
    - statistical sample?
WESS ’97 Paper

• Positive Features of Tools
  ◊ Ease of use
  ◊ Useful tools
  ◊ speed of tools

• Generic positive NF requirements
WESS ’97 Paper

• Negative Features of Tools
  ◊ lack of integration
    - don’t want to manually transfer data between tools
  ◊ wrong mix of features

• Difficulty introducing new tools
  ◊ resistance to new tools
  ◊ significant effort to learn new tool
    - will it be worth it?
  ◊ train a single individual to act as consultant within user group
WESS ’97 Paper

• Future Studies
  ◇ collaboration between researchers (spread effort and cost)
  ◇ questionnaires and logging tools in more companies (contacts)
  ◇ observe different engineers in different environments using same tools
  ◇ interviews with different groups of SEs

• Issues
  ◇ Same questionnaires and interview protocols
  ◇ similar methodologies - training/experience/presentation
• Same Research, more Depth
  ◊ identifies some problems with traditional ESP approaches
  ◊ understanding how programmers solve problems does not necessarily lead to better tools
• Usability vs Useful
  ◊ Usability - clarity of interface
  ◊ done in an artificial environment
    - isolated from other factors
    - user forced to use tool
  ◊ does not guarantee that the software is useable
    - would he use the software
CASCON ’97 Paper

- Telecommunications company (Mitel)
  
  ◊ several million lines of code
  ◊ well defined process

- Survey
  
  ◊ Reading Documentation tops the list
  ◊ look at source
  ◊ design near bottom of list
  ◊ 57% of time fixing bugs, 35% of time making enhancements
    - differs from published norms, survey effect or difference in business?
  ◊ Validity of surveys?
• Individual Study
  ◊ new employee (experienced)
  ◊ weekly meetings at start
  ◊ 3 weeks apart later
  ◊ mental model of system
  ◊ tasks, “new” information
  ◊ shadow user, record activities
    - observer effect?
  ◊ search is most frequent activity
CASCON ’97 Paper

- Group Study
  ◦ mental model of system
  ◦ interviews
  ◦ shadow user, record activities
  ◦ looking at source, searching is most frequent activities
  ◦ reading docs low on list (although high on survey)
Company Study
◊ company uses custom tools
◊ tool group collects statistics on tool usage (tools log their usage)
◊ compiles - 41% most often
  - nightly builds
  - testing groups
  - excluded
◊ search most frequent activity
◊ editors low - why?
CASCON ’97 Paper

• Results
  ◊ search seems to be where SEs spend most of their time
  ◊ improving search seems to present the greatest opportunity for support

• Just In Time Comprehension
  ◊ system too large to comprehend
    - general understanding
    - task determines what is comprehended
    - ignore rest of problem
CASCON ’97 Paper

- Tool Functional Requirements
  - search for semantic entities in source code
  - display results of search and relationships
  - searches are repeated (history)
- Non-functional requirements
  - system size
  - performance
  - more than one language
  - interoperability
  - independent interfaces (research)
  - support JIC
CASCON ’97 Paper

- Problems with Existing Tools
  ◊ grep
    - no syntax or semantics
    - does not understand relationships
    - time
  ◊ editor searches
    - no semantics
  ◊ IDEs
    - more semantics, limited languages
    - eclipse?
  ◊ analysis tools
    - no integration
CASCON ’97 Paper

- Problems with Existing Tools
  ◊ commercial browsing tools
    - no multiple languages (some improvement)
    - speed
    - limited integration
  ◊ academic
    - problems with integration, speed, automation