Course "Empirical Evaluation in Informatics"

**Surveys**

Prof. Dr. Lutz Prechelt
Freie Universität Berlin, Institut für Informatik
http://www.inf.fu-berlin.de/inst/ag-se/

- Example: SE education
- Method:
  - Set study goals
  - Select target population
  - Design questionnaire
  - Conduct survey
  - Evaluate results
"Empirische Bewertung in der Informatik"

**Umfragen**

Prof. Dr. Lutz Prechelt
Freie Universität Berlin, Institut für Informatik
http://www.inf.fu-berlin.de/inst/ag-se/

- Beispiel: Relevanz der Informatik-Ausbildung
- Methode:
  - Auswahl der Ziele
  - Auswahl der Zielgruppe
  - Fragebogenentwurf
  - Durchführung
  - Auswertung
Example:
Relevance of CS and SE education

- Source: T. Lethbridge: "What Knowledge Is Important to a Software Professional?", IEEE Computer, May 2000
  - see also http://www.site.uottawa.ca/~tcl/edrel/

- Research questions:
  Which parts of their education are considered how relevant by software engineering practitioners? Do they perceive their education as misaligned?

- Study format: Survey
Approach

- Uses a list of 75 topics from Computer Science (CS) and Software Engineering (SE) education
  - e.g. data structures, physics, project management, VLSI

- For each topic, asks 4 questions:
  - (1) how much was learned in education,
  - (2) how much was learned (or forgotten) since ,
  - (3) how useful the knowledge on the topic has been, and
  - (4) how influential on one's thinking the topic has been

- Determines the topics that are
  - deemed important but not taught widely ("knowledge gap")
  - deemed unimportant but taught widely
Survey population

- Web-based survey
  - employees of various companies (approached via mgmt)
  - postal mailing lists (e.g. university alumni)
  - email lists, Usenet newsgroups
- Over 200 participants
  - 186 participants were selected to form a balanced sample
- 54% from USA, 23% from Canada; 24 countries overall
  - 42% from software companies
- Education of participants:
  - 15% high school or college level (without degree);
    48% bachelor; 37% postgraduate
  - >60% CS, SE, or IS degrees; 50% other science or engineering;
    20% other disciplines
    - Many had more than one degree
Question 1

• How much did you learn about this in your formal education (e.g. University or College)?

  • 0=Learned nothing at all
  • 1=Became vaguely familiar
  • 2=Learned the basics
  • 3=Became functional (moderate working knowledge)
  • 4=Learned a lot
  • 5=Learned in depth; became expert (Learned almost everything)
Question 2

• What is your **current knowledge** about this, considering what you have learned on the job as well as forgotten?

  • 0=Know nothing
  • 1=Am vaguely familiar
  • 2=Know the basics
  • 3=Am functional (moderate working knowledge)
  • 4=Know a lot
  • 5=Know in depth / am expert (Know almost everything)
Question 3

How useful have the details of this specific material been to you in your career as a software developer or software manager?

Please leave blank if you know little about the material.

- 0=Completely Useless
- 1=Almost never useful
- 2=Occasionally useful
- 3=Moderately useful, but perhaps only in certain activities
- 4=Very useful
- 5=Essential
Question 4

• How much **influence** has learning the material had **on your thinking** (i.e. your approach to problems and your general intellectual maturity), **whether or not you have directly used the details of the material**?

Please consider influence on both your career and other aspects of your life.

Please leave blank if you know little about the material.

• 0=**No influence** at all
• 1=**Almost no influence**
• 2=**Occasional** influence
• 3=**Moderate** influence in some activities
• 4=**Significant** influence in many activities
• 5=**Profound** influence on almost everything I do
## Results: SE topics

<table>
<thead>
<tr>
<th>Category</th>
<th>Topic</th>
<th>Overall importance (Q3 + Q4)</th>
<th>Learned in education (Q1)</th>
<th>Learned on the job (or forgotten since education) (Q2 – Q1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>General software design</td>
<td>Data structures</td>
<td>◻️</td>
<td>◻️</td>
<td>◻️</td>
</tr>
<tr>
<td></td>
<td>Algorithm design</td>
<td>◻️</td>
<td>◻️</td>
<td>◻️</td>
</tr>
<tr>
<td></td>
<td>Software design and patterns</td>
<td>◻️</td>
<td>◻️</td>
<td>◻️</td>
</tr>
<tr>
<td></td>
<td>Software architecture</td>
<td>◻️</td>
<td>◻️</td>
<td>◻️</td>
</tr>
<tr>
<td></td>
<td>Object-oriented concepts and technology</td>
<td>◻️</td>
<td>◻️</td>
<td>◻️</td>
</tr>
<tr>
<td></td>
<td>Specific programming languages</td>
<td>◻️</td>
<td>◻️</td>
<td>◻️</td>
</tr>
<tr>
<td>Software engineering methods</td>
<td>Requirements gathering and analysis</td>
<td>◻️</td>
<td>◻️</td>
<td>◻️</td>
</tr>
<tr>
<td></td>
<td>Formal specification methods</td>
<td>◻️</td>
<td>◻️</td>
<td>◻️</td>
</tr>
<tr>
<td></td>
<td>Analysis and design methods</td>
<td>◻️</td>
<td>◻️</td>
<td>◻️</td>
</tr>
<tr>
<td></td>
<td>Performance measurement and analysis</td>
<td>◻️</td>
<td>◻️</td>
<td>◻️</td>
</tr>
<tr>
<td></td>
<td>Testing, verification, and quality assurance</td>
<td>◻️</td>
<td>◻️</td>
<td>◻️</td>
</tr>
<tr>
<td></td>
<td>Software reliability and fault tolerance</td>
<td>◻️</td>
<td>◻️</td>
<td>◻️</td>
</tr>
<tr>
<td></td>
<td>Maintenance, reengineering, and reverse engineering</td>
<td>◻️</td>
<td>◻️</td>
<td>◻️</td>
</tr>
</tbody>
</table>
### Results: SE topics (2)

<table>
<thead>
<tr>
<th>Software management</th>
<th>Project management</th>
<th>Software metrics</th>
<th>Software cost estimation</th>
<th>Configuration and release management</th>
<th>Process standards such as CMM, ISO9000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Essential subsystem design</td>
<td>Human-computer interaction/user interfaces</td>
<td>Databases</td>
<td>File management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specialized application techniques</td>
<td>Computational methods for numerical problems</td>
<td>Simulation</td>
<td>Artificial intelligence</td>
<td>Pattern recognition and image processing</td>
<td>Computer graphics</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Lutz Prechelt, prechelt@inf.fu-berlin.de
## Results: CS and science topics

<table>
<thead>
<tr>
<th>Category</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-time and systems programming</td>
<td>Operating systems</td>
</tr>
<tr>
<td></td>
<td>Systems programming</td>
</tr>
<tr>
<td></td>
<td>Data transmission and networks</td>
</tr>
<tr>
<td></td>
<td>Parallel and distributed processing</td>
</tr>
<tr>
<td></td>
<td>Real-time system design</td>
</tr>
<tr>
<td>Computer hardware</td>
<td>Digital electronics and digital logic</td>
</tr>
<tr>
<td></td>
<td>Microprocessor architecture</td>
</tr>
<tr>
<td></td>
<td>Computer system architecture</td>
</tr>
<tr>
<td></td>
<td>Network architecture and data transmission</td>
</tr>
<tr>
<td></td>
<td>Telephony and telecommunications</td>
</tr>
<tr>
<td>Other electrical and computer engineering</td>
<td>Analog electronics</td>
</tr>
<tr>
<td></td>
<td>Digital signal processing</td>
</tr>
<tr>
<td></td>
<td>Data acquisition</td>
</tr>
<tr>
<td></td>
<td>Robotics</td>
</tr>
<tr>
<td></td>
<td>VLSI</td>
</tr>
<tr>
<td>Computer science theory</td>
<td>Programming language theory</td>
</tr>
<tr>
<td></td>
<td>Formal languages</td>
</tr>
<tr>
<td></td>
<td>Computational complexity and algorithm analysis</td>
</tr>
<tr>
<td></td>
<td>Information theory</td>
</tr>
</tbody>
</table>
### Results: CS and science topics (2)

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete mathematics</td>
<td>Predicate logic</td>
</tr>
<tr>
<td></td>
<td>Set theory</td>
</tr>
<tr>
<td></td>
<td>Graph theory</td>
</tr>
<tr>
<td></td>
<td>Automata theory</td>
</tr>
<tr>
<td></td>
<td>Queuing theory</td>
</tr>
<tr>
<td></td>
<td>Combinatorics</td>
</tr>
<tr>
<td>Probability and statistics</td>
<td></td>
</tr>
<tr>
<td>Linear algebra and matrices</td>
<td></td>
</tr>
<tr>
<td>Continuous mathematics</td>
<td>Differential and integral calculus</td>
</tr>
<tr>
<td></td>
<td>Differential equations</td>
</tr>
<tr>
<td></td>
<td>Control theory</td>
</tr>
<tr>
<td></td>
<td>Laplace and Fourier transforms</td>
</tr>
<tr>
<td>Natural science</td>
<td>Physics</td>
</tr>
<tr>
<td></td>
<td>Chemistry</td>
</tr>
</tbody>
</table>
## Results: Other topics

<table>
<thead>
<tr>
<th>Category</th>
<th>Topic</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business</td>
<td>Economics</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Accounting</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Marketing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Management</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Entrepreneurship</td>
<td></td>
</tr>
<tr>
<td>Psychology and philosophy</td>
<td>Psychology</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Philosophy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ethics and professionalism</td>
<td></td>
</tr>
<tr>
<td>Technical writing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>People skills</td>
<td>Giving presentations to an audience</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Leadership</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Negotiation</td>
<td></td>
</tr>
<tr>
<td>Second language other than English</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results summary

• Some topics appear to be much over-emphasized in the formal education compared to perceived later usefulness
  • e.g. continuous mathematics

• Others appear much more important in practice than the education reflects, in particular
  • software management
  • people skills
  • requirements gathering
  • quality assurance
Internal validity problems

- Self-selection bias
  - Maybe many participants just wanted to lament about their education?
- Subjective answers
  - Real amount of knowledge or usefulness is unknown
- 15% answers from respondents without formal degree
- Scale violations:
  - Taking the difference $Q2 - Q1$ (knowledge now minus originally learned) requires equal-sized difference scales
  - Forming the sum $Q3 + Q4$ (usefulness plus influence) requires equal-sized ratio scales
    - What does the sum mean anyway?

How good is the credibility overall?
External validity problems

- Some of the answers strongly reflect north-american curricula

- Importance of topics is continually shifting over time!
  - some parts of the snapshot may already be obsolete

- What mix of software industry branches is represented in the sample?
Generic survey method

1. Decide on objectives
2. Select a target population
3. Design the survey instrument (questionnaire)
4. Administer the survey
   • supervised, unsupervised, or semi-supervised
5. Collect, validate and analyze the data
6. Answer the research questions
1. Decide on objectives

- A good understanding of the survey goals is required to select a compact set of questions
  - Too-long questionnaire will reduce number of respondents and may reduce the quality of the answers
- Surveys are suitable for measuring attitudes, much less suitable for determining factual situations

Basic types of objectives:
- Cross-sectional: snapshot
  - What is the status now?
- Longitudinal: cohort observation
  - How does the status change over time?
    - Requires multiple rounds of surveying with the same participants
- Retrospective: explanation
  - What are the reasons for the status?
2. Select a target population

Check questions:

- What kind of respondents do I need to reach my goals?
  - Need to use language/terminology appropriate for them
- How many such respondents do I need?
  - Is that realistic with acceptable effort?
- How can I reach these people?
- How can I motivate them to participate?
  - What response rate should I expect?
- How many irrelevant or distortive participants should I expect?
  - Can I recognize these from their answers and sort them out?
- So where and how should I advertise my study?
3. Design the survey instrument (questionnaire)

- Search for similar, previously used questionnaires
  - Sociologists call them "instrument": development is difficult
  - Analyze them (and the experience made) and adapt them
    - Piece your questionnaire together from multiple sources

If you need to design your own:
- Minimize the number of questions
  - Standardize the response format where possible
    - e.g. strongly agree, agree, disagree, strongly disagree
- Design each question carefully (see next slide)
- Put the demographic questions at the end
  - So people already know what information they have provided
  - So they are less likely to drop out near the end
- Ask for global comments on both the topic of the survey and the survey itself
Design the instrument: questions

For each question, make sure you have:

- **Clear purpose**
  - Respondent must understand the role of the question in the context of the survey and for the goals of the survey

- **Single purpose**
  - A question must not mix two issues

- **Complete, precise, unambiguous formulation**
  - Use simple and complete sentences
    - What is simple depends on the population
    - Avoid jargon and specialized terminology
      - e.g.: "How would you rate your training/education experiences regarding co-occurring disorder clients to date?"
    - Avoid negations
  - What exactly does the question refer to?
    - time, context, entity/attribute
  - What not?
Design the instrument: question types

- **Open questions:**
  - Respondents formulate their own answer
  - Advantages:
    - wider spectrum of possible insights
    - less dependent on prior knowledge of questionnaire designers

- **Closed questions:**
  - Respondents choose among fixed answer categories
    - e.g. single choice, multiple choice, numeric, date
    - Never forget the category
      - "don't know"/'none of these"/"does not apply"
  - Advantages:
    - easy quantitative evaluation
    - reduced ambiguity, less danger of irrelevant answers
Design the instrument: validation

Any questionnaire must be pilot-tested (e.g. by exit interviews):
- (in particular for unsupervised surveys)
- Find out whether overall purpose is clear
  - and why the participant should be motivated
- Find out whether purpose and formulation of all questions are clear
  - Detect ambiguities
  - Detect obscure terminology etc.
- Find out if time to complete is acceptable
- Find out whether layout and user interface are acceptable

Never forget pilot-testing after the very last 'correction'
Design
the instrument: validity, reliability

• Validity
  • The degree to which the instrument really measures what it was designed to measure
  • Assessing validity is methodically quite difficult and is beyond the scope of this course
    • See a textbook on social science research methods

• Reliability
  • The degree to which the instrument will give the same results when used in the same circumstances
    • if reliability is low, validity will be limited
  • Assessing it requires a number of respondents answering the questionnaire again after some time (e.g. a few weeks)
    • questions that are difficult to decide for the respondents typically lead to low reliability
4. Administer the survey

Basic types of administration:

- **Supervised**
  - An interviewer asks questions, answers clarification questions, and records answers (one-on-one, e.g. telephone)

- **Unsupervised**
  - The participant is completely on his/her own with the questionnaire (e.g. web-based)
  - Issues: multiple participation, question misunderstandings, joke answers, random answers

- **Semi-supervised**
  - An interviewer gives some introduction to a group of participants and answers questions, but the filling-in is unsupervised
5. Collect, validate and analyze the data

Tasks:

- Collect data into machine-readable form
  - Avoid/detect mistakes when typing in paper questionnaires
- Validate data:
  - Detect and remove duplicates (e.g. sent by email or http)
  - Detect invalid responses (needs consistency check questions)
  - Detect ambiguous questions (by incoherent answer structure)
- Perhaps balance the respondent set:
  - If some subgroups are over-represented,
    - either sample a subset from each of these (if you have enough data)
    - or weight subsets differently during analysis
- Analyze
6. Answer the research questions

When drawing conclusions from a survey

• keep in mind the limitations of your sample
  • in particular non-representativeness

• keep in mind possible validity problems such as
  • bias in the questions,
  • bias in the answers,
  • ambiguities and misunderstandings
Summing up: Surveys

- Surveys can be a low-cost means of collecting interesting information
  - They can be cross-sectional, longitudinal, or retrospective

- Try to reuse or adapt existing questionnaires where possible

- Carefully design and validate in any case

- Watch out for sampling bias!
  - You will almost always have some
  - but if you do not understand what it is, your data becomes dubious
Thank you!