Course "Empirical Evaluation in Informatics"

Case studies

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

- Example 1: Ramp-up of new members of a SW team
- Characteristics of case studies
  - unit of analysis
  - many sources of evidence (triangulation)
  - validity dimensions
- Example 2: A non-traditional approach to requirements inspections
"Empirische Bewertung in der Informatik"

Fallstudien

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

- Beispiel 1: Einarbeitung in ein Softwareteam
- Eigenarten von Fallstudien
  - Was ist der 'Fall'?
  - Nutzung vieler Datenarten, Triangulierung
  - Gültigkeitsdimensionen

- Beispiel 2: Ein unkonventioneller Ansatz für Anforderungs-Inspektionen
Example 1: Naturalization of SW immigrants

- Topic: What happens during the time when an experienced newcomer acclimates to a software project?
- Approach: exploratory multi-case case study
Goals and basic method

- Goals:
  - describe naturalization process
  - identify shortcomings and successes
  - characterize adaptation strategies used by immigrants

- Basic method: multiple interviews with four "immigrants"
  - 2 cases with 6 interviews spaced over first 4 months
  - 2 cases with 1 interview after 7 (or 8) months on the team
  - all interviews performed by the same investigator
Interview questions

- There are questions on background and on the naturalization process

- Examples:
  - What is your current assignment?
  - How did you gather information about the problem?
  - What resources did you use? (documentation, people)
  - What new things did you learn over the last week?
  - What new tools did you use over the last week?
  - What have you done to become more familiar with the software system?
  - Draw a diagram of your current understanding of the system

- Interviewees would also elaborate on their answers
  - How? Why? What else?
Analysis

• 17 variables of interest were determined from the material. Areas:
  • respondent characteristics,
  • orientation and training,
  • difficulties outside of learning about the system,
  • timing and type of tasks given, and
  • approaches used to understand the system

• The values were filled into a data matrix

• Pattern matching relates information from one or more cases to a theoretical proposition
  • Seven such propositions ("patterns") were found
Example answers

• "Most people operate under the assumption that
  • there are no documents, so you shouldn’t try asking for one."

• "I tried to [set up backups for my machine],
  • but I got stalled because I had to register my machine. So when
    that comes back, I’ll continue... ."

• "The system was humongous and I didn’t know what comes
  first or anything.
  • So the only way to do it is to dump everything [execution
    traces]. I didn’t do that from the beginning, but I found it really
    frustrating because I wouldn’t know what was actually being
    done."

• "I had to modify just four files at first.
  • It didn’t seem very challenging, but looking back, I appreciate
    the fact that they gave me something so isolated."
Patterns found

• Mentoring
  • Pattern 1: Mentoring is effective, though inefficient
  • Pattern 2: Lack of documentation forces immigrants to consult people

• Difficulties outside of the software system
  • Pattern 3: Administrative and environmental issues are a major source of frustration

• First assignments
  • Pattern 4: Initial tasks were simple or open-ended and began no earlier than after two weeks
  • Pattern 5: Mentors tend to pass on low-level information about the software system
Patterns found (2)

- Predictors of job fit
  - Pattern 6: Programmers who prefer to use bottom-up comprehension approaches have a smoother naturalization than those who don’t
  - Pattern 7: There needs to be a minimal interest match between immigrants and the software system.

- The study discusses specific evidence for and implications of each pattern
Conclusions drawn

• Immigrants could profit much from a high-level intro course about the system
  • focusing on architecture and design rationale
  • It cannot replace mentors, but would reduce their load
  • It would help in top-down understanding

• A recurring topic in the naturalization process is frustration
  • so avoiding frustration is a good improvement guideline

• Process improvements cannot be purely technical
  • they have to be organizational
Case studies: Main characteristics

- A case study is a prolonged observation of some phenomenon of interest in its natural setting

- Case studies are firmly bound to a certain **context**
  - The phenomenon of interest cannot be clearly separated from the context

- Case studies are **longitudinal**
  - They study a phenomenon over some time

- **Little control** is exerted
  - usually more control would be impossible

- The **observations are broad** and multi-faceted
  - both qualitative and quantitative
  - often additional observations are introduced during the study
Case study method

- Formulate research question
  - Types: **How? Why?**
- Find appropriate observation context
- Plan and implement data collection
  - and chose criteria for interpreting the data
- Collect data until satisfied
  - There may be no "natural" end of the observation period
- Analyze data
  - May be concurrent with data collection (to decide when to stop)
- Produce explanation (for why-questions)
  or description (for how-questions)
- Draw conclusions: Answer the question
Case study objectives

One of

- **Exploration**
  - Gain an overview of a hardly understood phenomenon
- **Characterization**
  - Describe in detail how something works
- **Validation**
  - Check whether a pre-formulated assumption is true
  - Typically these are existence proofs

- Case studies aim at deep understanding
- The target phenomenon is
  - an existing situation (such as a project, team, system)
  - or an intervention (such as a process, method, tool)
Examples of case study research questions

**Why questions:**

- Why does this organization follow this process model?
- Why do developers prefer this notation?
- Why do programmers fail to document their code?
- Why have formal methods not been adapted more widely for safety-critical systems?

**How questions:**

- How are inspections carried out in practice?
- How does agile planning work in practice?
- How does software evolve over time?
- How does a company identify which project to start?

How questions tend to be wider than why questions.
Main case study problem

• In a single-case study, there is but a single object of interest
  • The "case"
  • We can take repeated measurements of that same case over time, but each of them may be unreliable
  • We can measure many different aspects of the case
  • Note: There are multiple-case case studies as well
    • But the number of cases will rarely be more than a dozen

• We are often interested in multiple variables

• How can we make sure our conclusions are reliable?

• Solution approach
  • Rely on multiple sources of evidence
  • Bring them together to "triangulate" your variables
Case study design

- Like for an experiment, the measurements to be made during a case study should ideally be designed in advance
  - so that the data can (presumably) answer the question
  - Limited knowledge may make this designing hard
  - Additional data is often found during the study

- The design is often influenced by prior knowledge (assumptions, called **propositions**)
  - Propositions indicate where to look for evidence

- The central technical design decision concerns the **unit of analysis**:
  - What exactly is the 'case' of the case study?
  - Sometimes we consider units and subunits
Case study design: elements

1. Research question(s)

2. Propositions (may be missing)

3. Unit(s) of analysis

4. Method of analysis
Unit of analysis

- Not always obvious
- Must be chosen to fit the research question

Examples:
- For a study of how software immigrants naturalize, it can be
  - individual immigrant; development team; organization
- For a study of pair programming, it can be
  - programming session; pair of programmers; development team; organization
- For a study of software evolution, it can be
  - modification request; file; system; system release etc.
Method of analysis

This consists of two parts

1. A mechanism or logic for how to link the observations to the propositions (if any)
2. Criteria for interpreting the observations in terms of the research question

- Both of these aspects are not very well understood
  - There is little theory for how to do this in general
  - We need to find plausible ways for each study separately
Generalization from case studies

- In a well-designed survey or controlled experiment, we generalize quantitatively from a random(!) sample to a whole population
  - *Statistical generalization* (level-1 inference)
  - There are well-defined procedures for this, using notions such as significance, confidence interval, effect size, etc.

- Note: In practice, true random samples from a well-defined population are quite rare

- In a case study, statistical generalization is impossible
  - Even in multiple-case studies, as the cases cannot claim to form a random sample
Generalization from case studies (2)

- In case studies, we have to use *analytical generalization* instead:
  - Compare your results to previously existing theory
  - Replication: 2 or more cases all support the same theory
  - Best if multiple cases support one theory but do not support another (rival) theory
    - The purpose of a case study is untangling multiple competing explanations of the same phenomenon ("theory triangulation")

- Analytical generalization is level-2 inference
  - Can also be used for surveys, experiments etc. after statistical inference
  - Can be quantitative as well as qualitative

- Case study design goal:
  Make successful analytical generalization likely
How many cases do we need?

Case study types:

• Types 1 and 2 (single-case):
  • Type 1 (holistic): 1 context, 1 unit of analysis
  • Type 2 (embedded): 1 context, n units of analysis

• Types 3 and 4 (multiple-case):
  • Type 3 (holistic): k contexts, 1 unit of analysis in each
  • Type 4 (embedded): k contexts, n_i units of analysis each

• When are single-case studies sufficient?
  • it is a critical case (for testing some theory)
  • it is an extreme or unique case
  • it is the only case available at all
  • it is arguably a representative or typical case

• In most situations multiple-case studies are preferable
Multiple-case studies and replication

After investigating case 1, for case 2 we may expect

- either similar results
  - then it is like replicating an experiment
- or different results (because of differences in context)
  - then it is like doing a related experiment.

This is valid only if our theory provides arguments

- when to expect similar results and
- when to expect different results

- If we have such expectations (derived from a theory), then
  - meeting these expectations lends high credibility to the case study
  - seeing them fail requires revising some propositions
    - (but we do not necessarily know how)
Use multiple sources of evidence

- The small number of cases must be compensated by the breadth of the observations

- We try to use all six possible sources of evidence (2 actively, 4 passively):
  1. Interviews
     - open-ended, focused, or formal survey
  2. Participant-observation
     - observer participates in setting (intense, but danger of bias)
  3. Direct observation
     - via presence-at-site or specialized automated measurement
  4. Documentation (unstructured, semi-structured)
     - email, agendas, minutes, reports, previous studies, etc.
  5. Archival records ((semi-)structured, quantitative)
     - service records, logs, budgets, survey data, etc.
  6. Physical artifacts
     - e.g. hand-drawn multi-person design sketches
Triangulation

• For maximum breadth of observation we try to observe each single thing in more than one way

• This is called *triangulation* (approach target from different directions)

• Kinds:
  1. **data triangulation**: different data sources
  2. **investigator triangulation**: different observers or evaluators
  3. **theory triangulation**: interpret observations from point of view of multiple competing theories
  4. **methodological triangulation**: complement case study by surveys, experiments etc.
Case study database

• The large variety of data makes it hard to maintain proper overview

• Thus one should keep a formal case study database:
  • list all relevant materials
  • describe their structure
  • include all their content (or pointers)

• A well-formed database may be useful for later studies as well
  • to retrieve information that was not part of the results

• One should maintain an explicit chain of evidence
  • explicitly linking questions asked to data collected to conclusions drawn
  • Has much higher level of detail than result report
Data analysis in case studies

• The breadth of data makes it hard to combine it all.
  • There are few standard methods
  • Ad-hoc procedures need to be invented

• Goals for the procedures:
  • Present and consider all the evidence
  • Include prior knowledge or expert knowledge
  • Clearly separate evidence from interpretation
    • As in journalism: news versus commentary
  • Consider multiple hypotheses and explanations

• General strategies:
  • Rely on theoretical propositions (and focus accordingly)
  • Think about rival explanations and focus on differences
  • Develop a case description otherwise
Analytic techniques

- Pattern matching
  - Compare observations to predictions

- Explanation building
  - Incrementally account for more and more observations

- Time series analysis
  - Trace quantitative data over time; statistical analysis

- Cross-case synthesis
  - In multiple-case studies: Concentrate on evidence that is compatible and consistent across cases

- Details are beyond our scope
The validity universe

(Mostly not specific to case studies)

- **Construct validity**
  - Is our study design adequate for what we want to find out?
  - intentional v.; representational v.; observation v. (predictive v.; criterion v.; concurrent v.; convergent v.; discriminant v.)

- **Internal validity**
  - (For explanatory or causal studies:) Have confounding variables (and hence rival hypotheses) been eliminated?
  - Reliability: Would repeating the study on the same cases come to the same findings?

- **External validity**
  - Generalizability of findings to other situations
    - typically much stronger in multiple-case studies
The case study report

- Presenting a case study is particularly difficult

- Typical approaches:
  - Top-down case description, bottom-up analysis description
  - Multiple-case studies: One chapter per case or per case tuple comparison
  - Chronological
  - Theory-building: Each section adds one piece to a theoretical argument
  - Suspense: Reveal results first, then explain them step-by-step in an interesting way
  - Question and answer format

- One should decide on the format during study design!
  - Advice: Start writing early
"Case study": Notion and term

- In Informatics, case studies as defined here are sometimes called "field studies" instead
  - (and often not done properly; both is recently getting better)

- In Informatics, the term "case study" is also sometimes used
  - for a trial of a technique in a non-realistic setting
    - even just an informal illustration of its use;
  - for what should be a controlled experiment, except it has n=1
  - for a controlled experiment where no findings are statistically significant

- "Case study" as defined here is a term from social science methodology
  - it describes a middle ground between quantitative and qualitative research
Literature

- Robert K. Yin:
  "Case Study Research: Design and Methods",
Sage Publications, 2002
Example 2: A specific form of inspections


- Characterizes the specific approach to inspections as chosen due to the particular conditions in one organization

- Study type: Case study
Inspections

- A number of reviewers analyze a document (requirements, design, code, test plan, etc.) to identify defects
- The defects are collected and validated, then repaired

Advantages of inspections:
- Defects are found earlier (reducing rework cost)
- More defects may be found (improving final quality)
- Defects may be found with less effort
- Reviewers learn information from the document
- Reviewers learn about style and techniques

Disadvantages of inspections:
- Inspections consume resources and produce waiting time
- If badly done, inspections can reduce motivation
Inspection parameters

Where inspections can vary:

- **Sizing parameters**
  - Number of reviewers; preparation time; meeting time; re-reviews; etc.

- **Types or roles of reviewers**

- **Defect detection procedures**
  - e.g. ad-hoc, checklists, perspectives, scenarios, question-answering, walkthrough in meeting, etc.

- **Defect collection procedures**
  - e.g. meeting (different kinds); electronic meeting; asynchronous electronic meeting; one-to-one meetings; no meetings

- **Defect repair and re-review procedures**

- ...and more
The context: DaimlerChrysler

- Introduced inspections during the 1990s
  - good track record
  - have established process descriptions, tutorials, internal coaching/consulting, inspection experience base
  - constant improvement of the inspection process

- Our case: A set of embedded systems responsible for driver and passenger comfort
  - 50 requirements documents
  - each was typically 20-50 pages and
  - typically contained about 10-16 functional requirements
  - 70% of requirements are considered fairly stable
  - Goals of inspection:
    - improve quality of requirement specifications;
    - enhance common understanding;
    - eliminate open points, mistakes, and ambiguities.
Inspection design

- 2000 pages of requirements: A parsimonious inspection process is required

- 19 inspections (for the 50 documents)
  - focus on quality attributes: correctness, consistency, testability, maintainability

- 2 inspectors each (one also acting as moderator)

- Detection: Active involvement of inspector required
  - has to build a model (UML or SDL) of the artifact

- Collection: Present models in meeting,
  - focussing on requirements defects found
Propositions:
Claimed advantages

- Ensures each inspector is well prepared for meeting
  - half-hearted preparation is less likely

- Technical justification if available for every defect proposed
  - as it is explained in the context of the model

- Discussion between inspector and author is based on technical content
  - personal conflicts are avoided

- Presentations make meetings more interesting
Analysis approach

• The study analyzed this inspection method as follows:

• How does this method differ from a traditional method with respect to
  • effort (for preparation, for meeting)
  • number of defects found (as accepted in meeting)
  • size of documents?

• Data collected for each inspection:
  • document size (in pages and other metrics)
  • preparation effort (in person minutes)
  • meeting effort (in person minutes)
  • number of non-trivial defects accepted in meeting
Hypotheses

The analysis proceeds by checking the following hypotheses (about which something is known for conventional inspections):

• H1: The larger the inspection effort, the more defects are found
• H2: The larger the document size, the more defects are found
• H3: The larger the document size, the more effort is spent
• H4: Different inspectors will find similar numbers of defects
• H5: The meeting results outperform each individual inspector
Results:
size, preparation time, meeting time

- Size has one outlier; preparation time dominates effort
- Number of defects: about one per two pages
Results: Effort and defects

- Preparation time correlates strongly (0.7) with defects found, while meeting time and document pages do not.
Results: Size and defects

- Number of scenarios and number of requirements correlate strongly (0.69, 0.74) with defects found, while number of document pages does not.
Results:
Size, effort, and defects found

• ...and so on
• leading (somehow) to the following path diagram for explaining number of defects found:
Results: Relationships

- Helpful?
Note on the case-studyness

• This is a borderline case study:
  • It is hardly longitudinal
  • The analysis is rather quantitative
  • There is little focus on the procedural HOWs or WHYs
    • In particular, the effect from the model-building is not analyzed!

• On the other hand
  • context is important
  • no control is exerted (retrospective study)

• Note that the unit of analysis is the whole set of inspections

• Another note:
  • The article is fairly precise when talking technically about statistics, but sometimes sloppy when talking about causality (which is sometimes implied where it is in fact unknown)
Summary

• A case study investigates a small number of cases in depth
  • describes and takes into account the context
  • uses a broad spectrum of observations (many sources of evidence)
  • uses observations over time (longitudinal study)

• It involves little or no control

• It unifies qualitative and quantitative observations
  • Both analysis and conclusions tend to be argumentative rather than numerical

• The goal is an understanding that is specific, but deep
Thank you!