Using Qualitative Methods in Software Engineering

Extracted from a tutorial by

Jeff Carver, Mississippi State University

Carolyn Seaman, University of Maryland Baltimore County & Fraunhofer Center, Maryland

Ross Jeffery, University of New South Wales & NICTA

At the International Advanced School of Empirical Software Engineering (IASESE04)
August 18, 2004 - Los Angeles, CA
Research Design Examples
Example 1:

A Study of Communication and Organization in Inspection Meetings

Carolyn Seaman
Objective

• Characterize communication patterns and organizational relationships and the interaction between the two
• Research questions:
  – How much time do developers spend in various types of technical communication with each other?
  – What are the different types of organizational relationships that exist between developers?
  – How do the organizational relationships that exist among a group of developers affect how effectively and efficiently they communicate on technical matters?
Problems

• Important organizational relationships are not documented well
• People have a hard time describing how they communicate
• Little previous work on characterization of either communication patterns or organizational relationships
• Needed a setting where organizational relationships and technical communication might interact
Opportunities

• A large project developing a mission planning tool for NASA, in the beginning of the coding phase
• A large number of code inspections planned, involving a variety of configurations of people from different organizational contexts
• Government agency and contractor – processes and organizational structure well defined and documented
Solutions

• Use participant observation to document actual communication patterns (length of discussions, types of discussions, topics)
• Use structured interviews and documents to identify important relationships
• Length of study will allow for refinement of categories, for characterizing both communication and organization
• Extraction of quantitative values will allow some statistical analysis
Research Design

Values for Meeting and Discussion Lengths

Characterization | Observations of Inspection Meetings | Structured Interviews | Coding and Analysis

Context Information
Variables
Models
Final Study Design

Values for Organizational Variables

Results
Lessons Learned

• Ensuring accuracy of data
  – Multiple coders
  – Timely interviews
  – Triangulation
  – Audiotaping of interviews

• Interpretation of data
  – Too much quantitative data without enough context
  – Qualitative data was needed for interpretation
  – Extensive field notes
  – Well organized field notes

• Too little data
  – Fewer variables to allow partitioning
  – Qualitative data helps fill the gaps
Example Results

• Hypotheses
  – All organizational variables affect some form of communication effort
    • e.g. higher familiarity ==> lower global discussion time
  – Often conditioned on values of size and complexity

• Qualitative observations and interpretation
  – e.g. many topics are discussed outside of the meeting if the participants are close
Example 2:

Context Variables and the Variation in Inspector Performance

Jeff Carver
Objective

• Generate well-grounded hypotheses about the effects of context variables on software inspection

• Research questions:
  – Which variations in individual inspectors will impact defect detection effectiveness
  – How do the variables interact with each other
Problems

• Software inspections are effective for defect detection but the results are not consistent from one inspector to the next
• Little previous work on the impact of variations in the human inspector
• Needed some existing data as well as the opportunity to conduct new studies
Opportunities

• Availability of a large body of data from previous software inspection studies
  – Unlimited access to data
  – Collected the right background information

• Software Engineering courses being taught at the University of Maryland
  – Opportunity to conduct two new studies
Solutions

• Use the Constant comparison technique from grounded theory
• Perform literature search to get initial hypotheses
• Analyze existing empirical data to refine hypotheses
• Conduct new studies, based on above findings, to continue refinement
Research Design

- Gather Expert Opinion (Literature)
- Gather Existing Data
- Generate New Data

Data Analysis

Context Variables
Hypotheses

Quantitative Studies
Quantitative Analysis
Constant Comparison
Lessons Learned

• Usefulness of Grounded Theory
  – Constant comparison useful in SE research
  – Can work on both qualitative and quantitative data

• Need a mix of classroom and industrial data
  – Allows results to generalize
  – Experience/expertise of industrial subjects will likely differ from that of students

• Hard to decouple the effects of variables
  – Many results could be explained by multiple variables
  – More studies need to be conducted to isolate effects
Example Results

• Hypotheses Generated
  – Application Domain Knowledge is helpful in a requirements inspection but not in a design inspection
  – Observing a well-done inspection is helpful for training a novice in the inspection process

• Qualitative Methods
  – Can be used on quantitative data to generate grounded hypotheses
Conclusions

• It seems unlikely that the use of qualitative methods alone can compensate for experience in process modelling and software engineering.

• An experienced process engineer should utilise qualitative methods to help ensure coverage of the data and to encourage decomposition of the process model.

• The tradeoffs between cost, repeatability and coverage will need to be explored.