

Using Qualitative Methods in Software Engineering

Jeff Carver

Mississippi State University

Carolyn Seaman

University of Maryland Baltimore County
Fraunhofer Center, Maryland

Ross Jeffery

University of New South Wales
NICTA

*International Advanced School of Empirical Software Engineering (IASESE04)
August 18, 2004 - Los Angeles, CA*

Outline

1. Qualitative Methods Overview
2. Specific Qualitative Techniques
3. Qualitative Methods Examples

Part I - Overview

Definitions

- Qualitative data
 - Data in the form of text and pictures, not numbers
- Qualitative analysis
 - Analysis of qualitative data in order to discover trends, patterns, and generalizations
- Grounded theory
 - Theory formed bottom-up from the (usually qualitative) data
- Rich data
 - Includes a lot of explanatory and context information

Why Qualitative Methods?

- **Problem:** Difficult to answer complex SE questions with a purely quantitative approach because
 - Working with human subjects
 - Typically have small sample sizes
 - Experiments are expensive to run
 - Need some support for a hypothesis before investing effort in full experiment
 - Difficult to understand context variables
- **Solution:** Use a combination of quantitative and qualitative methods

Types of Results

A **Qualitative Study** will result in:

- Propositions tied to a trail of “evidence”
- Well-grounded hypotheses
- Complex findings that incorporate the messiness of the phenomenon under study
- Explanations
- Areas for future study

When to Use

- Studying human behavior
- Lack of concrete hypotheses
- Variables hard to define or quantify
- Little previous work
- Quantitative results hard to interpret

Advantages

- For Researchers
 - Richer results
 - Results more explanatory
 - Closer to sources of data
 - Avoid errors in interpretation
- For Practitioners
 - Richer, more relevant results
 - Terminology of results
 - More part of the research process
 - Opportunity to clarify and explain findings

Part II – Specific Techniques

Overview of Techniques

Data Collection

- Document Analysis
- Interviews
- Participant Observation
- Prior Ethnography
- Surveys

Data Analysis

- Auditing
- Coding
- Constant Comparison Method
- Cross-case analysis
- Member checking

Interviews: Description

Type of Technique:
Data Collection

Type of Data Collected:
Qualitative field notes recording
interviewee responses

Instrument Used:
Interview Guide

Method:

Data gathered interactively, usually
one on one, from the subjects

Benefits:

1. Level of structure can vary
2. Flexible and adaptable
3. Allows the researcher
more freedom than a
survey or questionnaire

Interviews: When to use

- Study focuses on the meaning of a phenomenon to the participants
- Individual accounts of the development or evolution of a process or practice are required
- Exploratory work is required prior to performing a quantitative study
- A quantitative study has been conducted, and the results need to be understood or verified

Interviews: Issues

- Level of structure
 - Unstructured, Semi-structured, Structured
- Degree of disclosure
- Purpose
- Interviewee Preparation
 - Pre-interview questionnaire, data, etc...
- Difficult interviewees
- Recording medium
 - Tape, paper, etc...
- Time consuming

Interviews: Advice

- Listen more than you speak
- Phrase questions in a straightforward, clear and non-threatening way
- Eliminate cues which may lead interviewee to respond in a particular way
- Do not appear bored
- Practice!!

Interviews: Example

- Semi-structured interviews meant to elicit information about use of a packaged OO framework.
 - Use an interview guide to ensure important topics are covered during interview
 - Two researchers
 - One to conduct the interview
 - One to take notes (optionally replaced by tape recorder)

Interviews: Sample Interview Guide

Interview Guide 2a: In-depth project interviews

Who: Developers on [Project1], [Project2], [Project3]

Subjects covered: general opinions of GSS processes and products

Duration: 60-90 minutes

What do you like about the current process using GSS?

What do you dislike about the current process using GSS?

Do you depend on any other groups, either for information or help with GSS, or for work to be done related to GSS?

What do you like about the applications resulting from using GSS?

What do you dislike about the applications resulting from using GSS?

Have there been any problems with the interface between GSS and other COTS products?

What do you see as the top risks associated with the use of GSS? How would you mitigate these risks?

Participant Observation: Description

Type of Technique:
Data Collection

Type of Data Collected:
Qualitative field notes and
Quantitative data

Instrument Used:
Pre-defined forms

Method:

Researchers becomes part
of the observed group in
order to observe the activities
of the subjects first-hand

Benefits:

Data potentially more accurate
than post-hoc self reporting

Participant Observation: When to use

- With small groups
- On processes that are relatively short
- When events occur frequently
- When prime motivation is to find out what is going on (exploratory) rather than to confirm hypotheses
- When you have time

Participant Observation: Issues

- Important to have a **system** to guide information capture (data form)
- Getting all of the information may take as long as the original observation
- Subject **discomfort**
- Fly-on-the-wall vs. active participant
- Getting **too friendly** (going “native”)

Participant Observation: Advice

- Record observations on the spot
- Read through observations soon after event to clarify and fill in details
 - Prepare good detailed notes within 24 hours of observation session
 - Do not proceed to a second observation without completing the first one

Participant Observation:

Example

- Observation of code inspection meetings
 - Researcher does not participate in the inspection
 - Used data forms as well as field notes

Participant Observation: Sample Data Form

Inspection Data Form

Class(es) inspected

Inspection date:

Time:

Author:

Moderator:

Reviewers:

Name

Responsibility

Preparation time

Present

Amount of code inspected:

Complexity of classes:

Discussion codes:

D = Defects

Q = Questions

C = Classgen defect

U = Unresolved issues

G/D = Global defects

G/Q = Global questions

P = Process issues

A = Administrative issues

M = Miscellaneous discussion

Time logged (in minutes):

D_____ Q_____ C_____ U_____ G/D_____ G/Q_____ P_____ A_____

M_____

Participant Observation:

Sample Field Notes

The "step" function is a very important but complicated function. [Reviewer1] did not have time to review it in detail, but [Author] said he really wanted someone to go over it carefully, so [Reviewer1] said she would later.

There was a 4-minute discussion of testing for proper default values. This is a problem because often the code is such that there is no way to tell what a particular variable was initialized to. [Reviewer2] said "I have no way to see initial value". This was a global discussion, relevant to many classes, including [Reviewer2]'s evidently.

Surveys: Description

Type of Technique:
Data Collection

Type of Data Collected:
Qualitative and Quantitative
Objective and Subjective

Instrument Used:
Written form to be completed
by the subjects

Method:

Subjects complete the survey
form without the intervention
of the researcher

Benefits:

Can include both closed and
open ended questions to collect
different types of data

Surveys: When to use

- Majority opinion is important
- Subject population is large and distributed geographically
- Subjects are motivated to respond
- Many or most of the questions can be phrased as closed questions
- There is time to allow an adequate response interval

Surveys : Issues

- Wording and terminology
- Length
- Dissemination strategy
 - General availability on the Web
 - Email to selected population
 - Personal handout
 - Follow-up
 - Mandate?
- Structure (open vs. closed questions, etc.)

From Robson, 2002, p. 323

Surveys: Advice

- Pilot, pilot, pilot!!!
- Minimize number of completely open questions
- Minimize length
- Design data collection spreadsheet or database before finalizing survey design
- Review instructions carefully
- Take care of the logistics

Surveys: Example

- A survey of software maintainers regarding their information gathering practices
 - Maintainer background – experience, etc.
 - Characteristics of maintenance project
 - Ratings of a predefined list of information sources on usefulness, availability, etc.
 - Questions on most and least frequently used information sources
 - “Wish list”
- Full survey at <http://research.umbc.edu/~cseaman/maintenance.htm>

Surveys: Example

Background question:

1. How many years of experience do you have in any aspect of software development or maintenance (e.g. programming, testing, management, design, etc.)?

☐ < 6mo

☐ 6mo-2 yrs

☐ 2 yrs – 5
yrs

☐ 5 yrs – 10 yrs

☐ > 10 yrs

Surveys: Example

Rating information sources (actual list is longer)...

4. Below is a list of information sources that some software maintainers use to gain information about the systems that they are trying to maintain. For each information source, please indicate how *useful* you consider that source. If a source is not generally available to you, put an **X** in the first column, **N/A** (for “not available”). Otherwise, mark a column corresponding to a rating scale with **1** = Not at all useful, **2** = A little useful, **3** = Somewhat useful, **4** = Quite useful, and **5** = Very useful.

Information sources	N/ A	1	2	3	4	5
A. Original developers of the system						
B. Writers of the original system requirements						
C. Current users of the system						
D. Current system operators (e.g. sys admin)						
E. Customers making a change request						
F. Other maintainers						

Surveys: Example

More info on most frequently used sources...

7. Please list the **letters of the top three most frequently used** information sources (from the list in Question 6) that you consult in your work, and describe briefly why you use that source frequently (e.g. It's the handiest, or it's the only source for a particular type of information, or it's always accurate, etc.).

Source 1 _____

Why: _____

Source 2 _____

Why: _____

Source 3 _____

Why: _____

Surveys: Example

Ending, open-ended question...

What, in general, makes an information source useful to you in your maintenance work?

Coding: Description

Type of Technique:
Data Analysis

Type of Data Operated On:
Qualitative

Source of Data:
Field notes
Questionnaire responses
...

Method:
Codes are tags for assigning meaning to pieces of textual data collected during a study

- Benefits:
1. Facilitates identification of trends, patterns, generalities, etc..
 2. Can also be used to extract quantitative data from qualitative data

Coding: When to use

- Continuously throughout the process of data collection
- Always code one set of field notes before heading to the next observation site

Coding: Issues

- Coding Scheme
 - Existing or new
- Category types
 - Predefined vs. Emerging
- Time consuming, tiring process
- Helpful to have multiple coders to compare results

Coding: Advice

- Create an initial list of categories and evolve
 - If an existing coding scheme can be adapted, consider using it
- Develop categories that are:
 - Focused
 - Objective
 - Context independent
 - Explicitly defined
 - Exhaustive
 - Mutually Exclusive
 - Easy to record

From Robson, 2002, p. 332;
Miles, 1994, p. 65

Coding: Sample Field Notes Coding

Discussion of
specific function

Amount of time
spent discussing
“default values”

The "step" function is a very important but complicated function. [Reviewer1] did not have time to review it in detail, but [Author] said he really wanted someone to go over it carefully, so [Reviewer1] said she would later.

There was a 4-minute discussion of testing for proper default values. This is a problem because often the code is such that there is no way to tell what a particular variable was initialized to. [Reviewer2] said "I have no way to see initial value". This was a global discussion, relevant to many classes, including [Reviewer2]'s evidently.

Scope of
discussion

Constant Comparison:

Description

Type of Technique:
Data Analysis

Type of Data Operated On:
Qualitative

Source of Data:
Field notes

Method:

Coding → Grouping →
Writing field memo →
Forming hypotheses

Benefits:

1. Used to generate Grounded Theory
2. Repeated in parallel with data collection
3. NOT a process of formalizing hunches

Judging Validity of Results

- Validity of methods
 - Triangulation
 - Documentation
 - Contradictory evidence
- Weight of evidence
 - How much is enough?
 - Variety as well as quantity of evidence

Qualitative and Quantitative Methods Together

- Qualitative and quantitative methods best used in combination
- Can simply be used in parallel to address the same research questions
- There are other strategies to better exploit the strengths and weaknesses of the methods

Summary

- Empirical software engineering researchers are addressing more and more complex research questions that have increasingly human elements
- Qualitative methods, usually in combination with quantitative methods, can be helpful in handling this complexity
- Qualitative methods are flexible and rigorous
- Qualitative analysis provides richer, more relevant, and more explanatory results
- The most effective research designs combine qualitative and quantitative methods

Bibliography

- Babbie, E. Survey Research Methods. Belmont, CA: Wadsworth. 1990.
- Day, I. Qualitative data analysis: A user-friendly guide for social scientists. New York: Routledge. 1993.
- Eisenhardt, Kathleen M. "Building theories from case study research." Academy of Management Review, 14:4, 1989.
- Gilgun, J.F., "Definitions, Methodologies, and Methods in Qualitative Family Research." Qualitative Methods in Family Research. Sage, 1992. pp. 22-29
- Glaser, Barney G. and Anselm L. Strauss. The Discovery of Grounded Theory: Strategies for Qualitative Research. de Gruyter, 1967.
- Jick, Todd D. "Mixing qualitative and quantitative methods: Triangulation in action." Administrative Science Quarterly, 24:4, 1979.
- Judd, Charles M., Eliot R. Smith, and Louise H. Kidder. Research Methods in Social Relations. Harcourt, Brace, Jovanovich, 1991.
- Miles, Matthew B. and A. Michael Huberman. Qualitative Data Analysis: An Expanded Sourcebook. Sage Publications, 1994.
- Robson, C. *Real World Research*. Blackwell Publishers. 2002.

Part III – Research Design Examples

Example 1:

A Study of Communication and
Organization in Inspection
Meetings

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

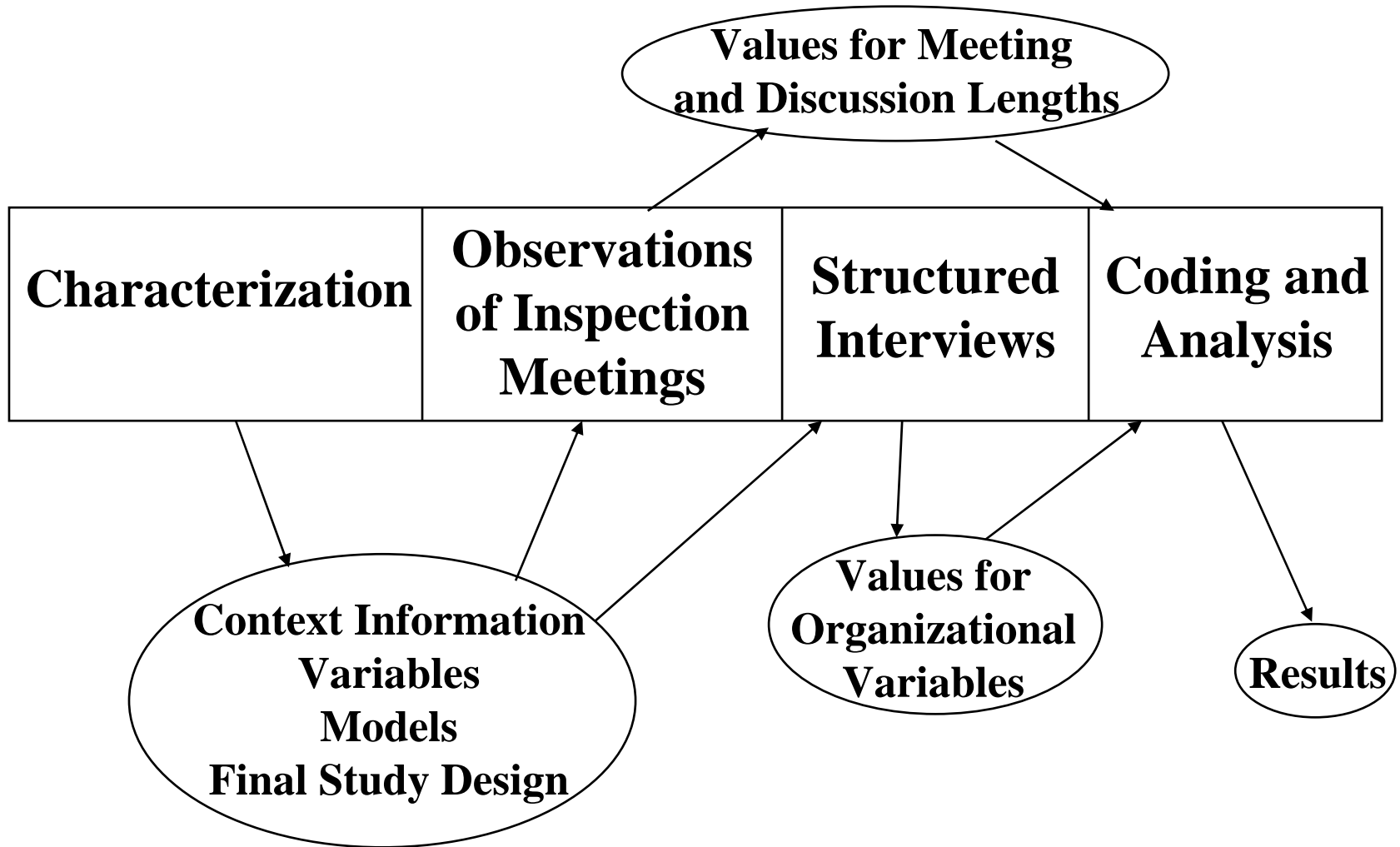
Opportunities

- 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



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

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

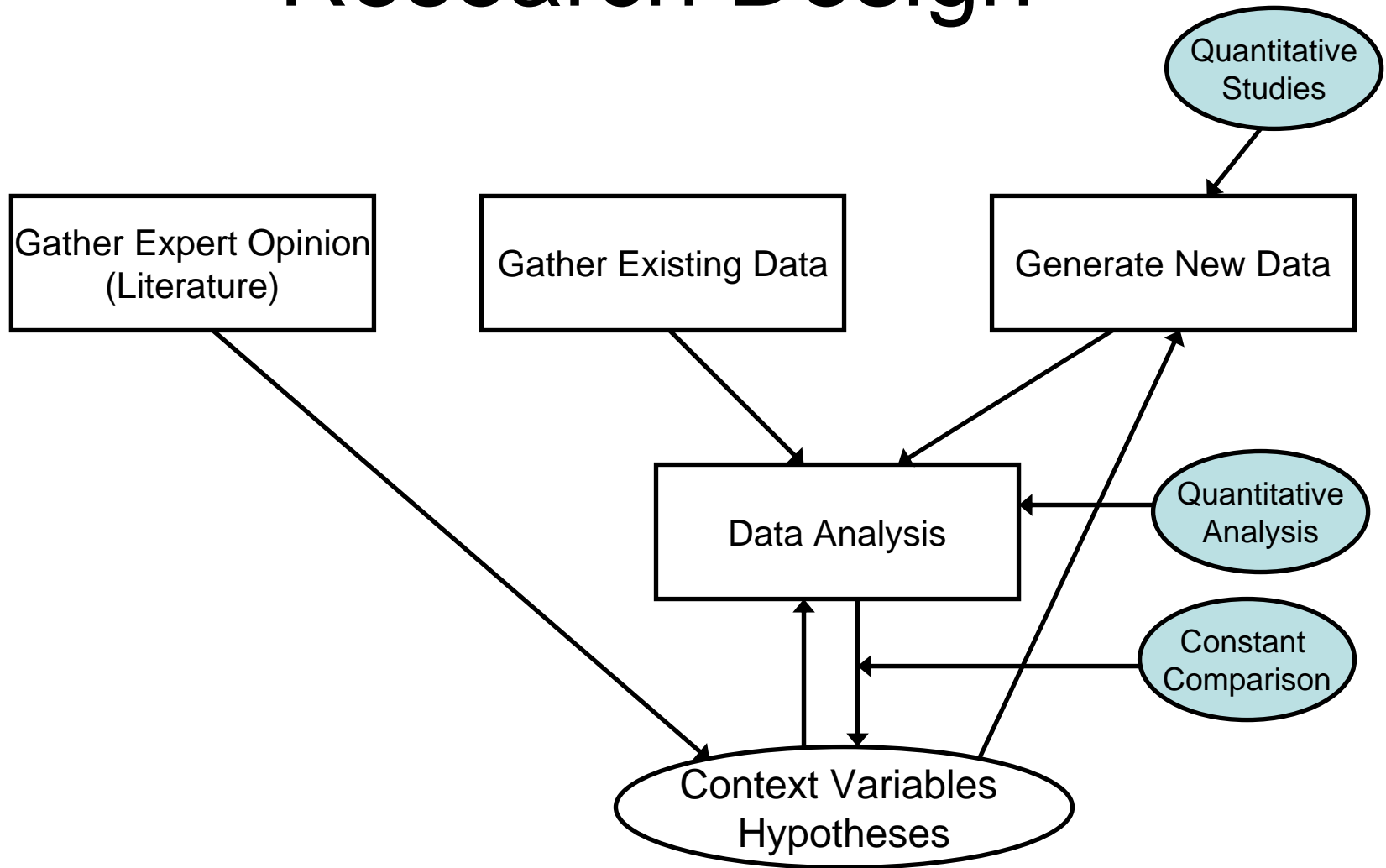
Opportunities

- 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



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

Example 3:

Deriving Process Models using Grounded Theory Methods

Ross Jeffery

An exploratory study

Descriptive Process Modeling Domain

- Small software development organization
- Method of constant comparison vs. expert process modeler

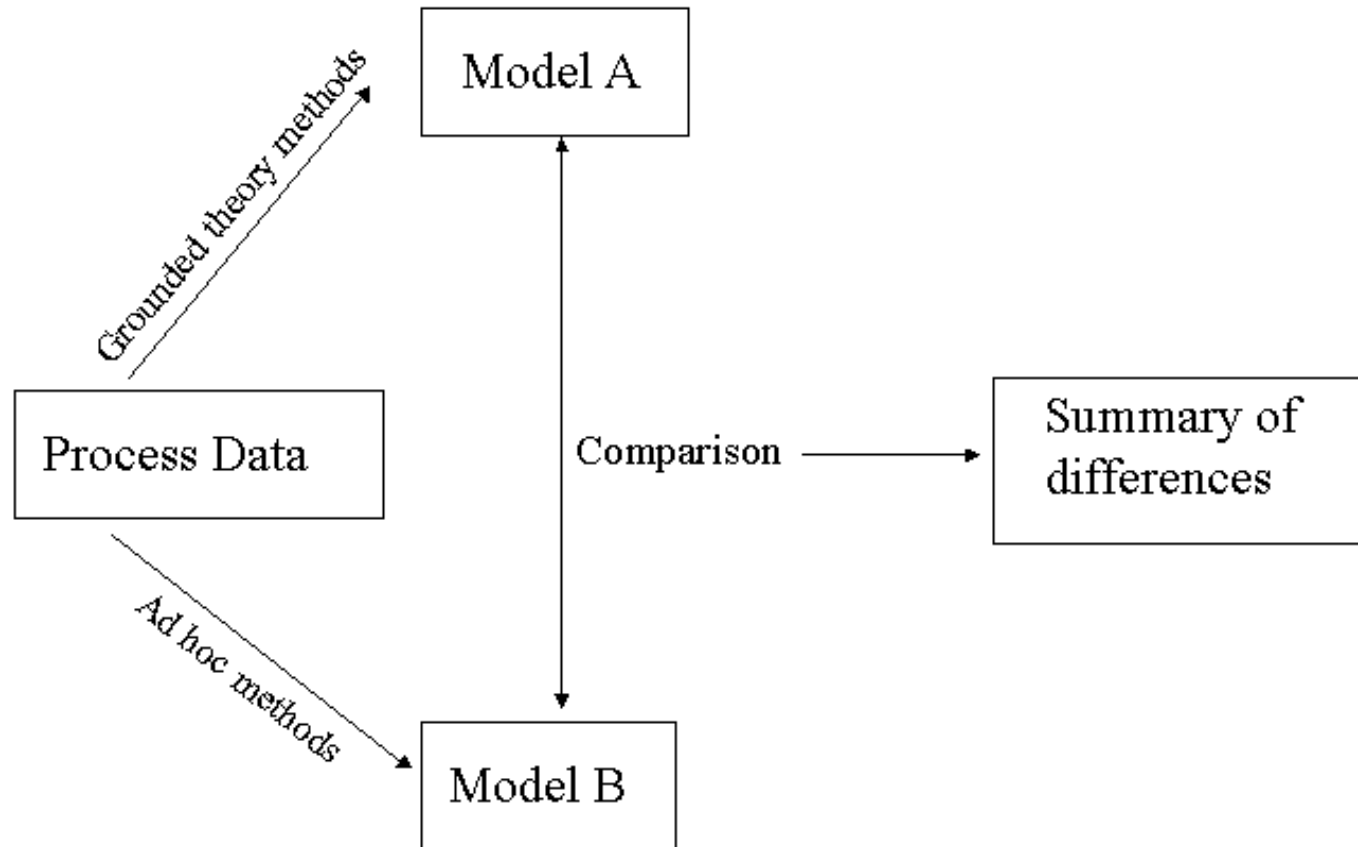
Research Questions and Variables

- Could constant comparison be used to elicit a process model from text data?
- Would a model produced using constant comparison be equivalent to a model produced by an expert process modeler using as hoc technique?
- How can constant comparison contribute to process modeling?

Research Methods

- A qualitative analysis of emails, agendas & meeting notes, reports.
- Two research participants:
 - (a) a psychologist experienced in method but with limited software engineering experience.
 - (b) a software engineer with three years experience in process modeling

Study Overview



Study Design

	<i>Approach</i>	<i>Data</i>	<i>Tools</i>
<i>Researcher A (psychologist)</i>	<i>Qualitative Methods + Nvivo</i>	<i>e-mails + minutes</i>	<i>Spearmint</i>
<i>Researcher B (software engineer)</i>	<i>Ad Hoc/ Experiential</i>	<i>e-mails + minutes</i>	<i>Spearmint</i>

* No exchange of information between the researchers during modeling.
Time sheets kept of tasks,

Analysis

- Equivalence was determined using:
 1. Semantic content
 2. Structure
 3. Decomposition hierarchies.

By mapping process concepts across the models. Process concepts identified by joint analysis of the models searching for equivalence.

Lessons Learned

- Models differed considerably in detail.
- Difficult to identify equivalence.
- The issue of objective determination of software artifact quality was existent.
- More agreement on “entities” than on “relationships”. A “process concept” was used to define a collection of entities.
- Agreement on 64 process concepts. Model A had a further 43 and model B had a further 72. (33% agreement)

Lessons Learned

- Model A had more layers of decomposition
- Model B was more detailed.
- Model A contained 20 entities represented as instances rather than abstractions
- Model A included some behavioral modeling
- Model A took longer to produce

Example Results

Approach	Positives	Negatives
Constant comparison	<ul style="list-style-type: none">- explicitly encourages abstraction to a refinement hierarchy- ensures coverage of all the data	<ul style="list-style-type: none">- misses isolated but important concepts- still relies on experience and skill to interpret the data
Ad hoc relying on experiences	<ul style="list-style-type: none">- more ability to judge the importance of concepts, even if they only appear in the data once (but not methodically)- ability to understand technical discussions in the data	<ul style="list-style-type: none">- coverage is not ensured- decomposition is not encouraged- relies solely on experience and skill of process engineer

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.