Comparações Constantes

Baseado em

Using Qualitative Methods in Empirical Studies of Software Engineering. ***Carolyn Seaman***. University of Maryland Baltimore County. ESELAW'2009, São Carlos, Brazil.

Constant Comparison Method

- Qualitative analysis method
- Meant to generate grounded theory
- Operates on a set of field notes
- Basic process:
  - coding
  - grouping
  - writing field memo
  - forming hypotheses
- Repeated periodically in parallel with data collection

What's a Code?

- A label
- A concept
- A topic
- A category
- A relationship
- A theme

What's Coding?

- Open coding - assigning codes to pieces of textual data
  - Coded "chunks" can overlap
  - One chunk can have several codes
- Axial coding - grouping, categorizing, combining coded chunks
- Selective coding - making sense of it

Open Coding

What's here? What are the pieces?

- Identification/discovery of concepts
- Classification (labeling of phenomena)
- Abstraction (this is part of that)
- Comparative analysis (this is different from that)
- Categorization (organization, grouping)
- Value-neutral, at least initially
  - "complexity" not "high complexity" or "low complexity"

Open Coding Process

- Preparing for coding
  - Read the data
  - Read background material and research design
  - Create pre-formed codes, if applicable
- Coding by hand
  - Document markup (colored pens, etc.)
  - Photocopy, scissors, and envelopes
  - MS Word comments
  - Excel
- Coding tools – NVivo, Atlas TI
- Coding scheme
  - *Pre formed* or *post formed* codes
  - Constant iteration
  - Structure develops over time

Open Coding and Quantification

- One form of coding
- Objective is to derive quantitative data from qualitative data for future statistical analysis
- Usually involves counting
  - How many subjects said…?
  - How many times did subjects use the term …?
  - How many times did …?
- Or timing
  - How long did subjects spend doing…?
  - How long did it take to …?
- Inevitably loses richness
- Often seems a little like missing the point
  - What's the point of collecting rich data when you're just going to condense it down to numbers?
- But often is an effective and necessary way to reduce the size of the data

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**
Reviewer raises a question or concern and it is determined that it is a defect which the author must fix; time recorded may include discussion of the solution

Q **Questions**
Reviewer asks a question, but it is not determined to be a defect.

C **Classgen defect**
Reviewer raises a defect caused by classgen; author must fix it, but it is recognized as a problem to eventually be solved by classgen

U **Unresolved issues**
Discussion of an issue which cannot be resolved; someone else not at the meeting must be consulted (put name of person to be consulted in () beside the code); this includes unresolved classgen issues. It also includes issues which the author has to investigate more before resolving.

G/D **Global defects**
Discussion of global issues, e.g. standard practices, checking for null pointers, which results in a defect being logged (does not include classgen defects)

G/Q **Global questions**
Same as above, but no defect is logged

P **Process issues**
General discussion and questions about the inspection process itself, including how to fill out forms, the order to consider material in, etc., but not the actual execution of these tasks.

A **Administrative issues**
Includes recording prep time, arranging rework, announcing which products are being inspected, silence while people look through their printouts, filling out forms.

M **Miscellaneous discussion**

Time logged (in minutes):

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

freshly painted room – smells + is hot
just had a task meeting – 39 classes needed in 6 weeks
SM: "This is a nightmare, and it's going to get worse."
– started 30 minutes late because of meeting

Class(es) inspected: ANI.3, EV3, EV3.1 **Date:** 3/15/96 **Time:** 2:00 **Page** 1 **of** 2

Time	Participants	Code	Notes
30	SM	A	get started; SM having problems finding right files
31	AP → RK	G/D	○ change to null – actually several different small defects
33	AP, SM, MI	Q	don't change now, wait for TB3
34	AP	D	couts
35	MI	G/D	"good thorough test plan" – some FYIs – not standard format – do for next TB – other style – don't take time now
36	MI	Q	MI went through everything she did – no defects – showed RK+SM something on paper – don't change for now
38	MI → RK, SM	Q	re DB filename
40	RK → M	Q	nothing on category
41	SM	A	○ null instead of 0 – had trouble finding it
42	SM → RK	G/D	ParameterError exception – trying to figure out where it's thrown
44	SM → RK, MI	D	similar to above "This leads me to my BIG QUESTION" – SM
46	SM → RK	U	RK catching error that will never happen MI: you're making it a lot more complex than you need to – too much error checking discussion of meanings of various parameters – MI: "action item for the 2 of you to battle out"
47	SM → RK, MI	U (RK+SM)	why is certain error generated by classgen?
53	RK → SM	Q	clarification
55	RK → MI, SM	Q	ParameterError – handle differently from the way classgen does it
56	SM → RK, MI	D	

Lots of time for everyone trying to find right place in printout – small print is a factor

"Ken already gave you his stuff, correct?" – SM to RK
yes

I'm having a hard time concentrating

CA standing up by door

MI gave RK marked up copy of test plan

EV3

EV3.1

43 CA leaves

printout very small – hard to read

– SM + AP talking off their gbs

47 CA comes back

Axial Coding

How are things related?

- Initial process of reassembling
- Relationships among categories and codes
- Structure (why?)
- Process (how?)
- Explanations not causal prediction

Selective Coding

How does it all fit together?

- Also called **sense making**
- Relationships among relationships
- Theory construction
- The central category
- Storyline memos
- Role of literature
- Write, write, write!!!
- **Field Memos**