# Analyzing Qualitative Data

Informatics 162W

February 4, 2013

## Bonus Point Opportunity

- Attend a talk by Maria Klawe
- Submit a short document (less than one page) relating her talk to our work in this class
  - “Gender and Technology” Bren Hall 6011 5 to 6PM Feb 7
  - “Blazing a Trail: What I learned from 25 years of being the first female in my job” Bren Hall 6011 2 to 3:30 PM Feb 7
- TurnItIn.com
  - Class ID: 6080138
  - Enrollment password: orgs123
- Due by Feb 10 (next Sunday)

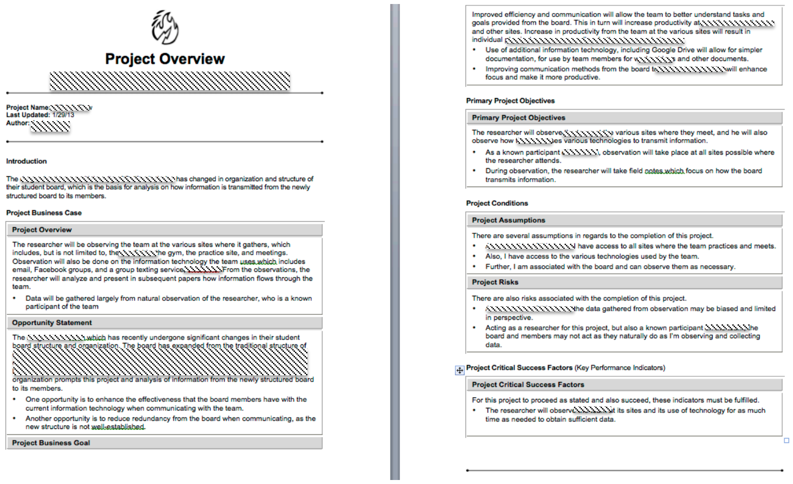
# Agenda

- Feedback on your writing
- Bonus Point Opportunity
- General information on qualitative analysis
- Thinking Topics (Lofland et al)
- Grounded Theory (Glaser and Strauss… and later Corbin)

# Writing

- Most of you did pretty well for your first attempt
- Some of you were disasters
  – Please show up to discussion and take notes on what is expected so that you can follow the format
  – Work out with your group when to do critiques and early drafts but DO THEM BEFORE THEY ARE DUE IN DISCUSSION
  – Your revisions will be graded at the end, not your early drafts, but still, try to do a decent job on early drafts ☺

# An example of a particularly good assignment



**Project Overview**

[redacted]

**Project Name:**[redacted]
**Last Updated:** 1/29/13
**Author:** [redacted]

**Introduction**

The [redacted] has changed in organization and structure of their student board, which is the basis for analysis on how information is transmitted from the newly structured board to its members.

**Project Business Case**

**Project Overview**

The researcher will be observing the team at the various sites where it gathers, which includes, but is not limited to, the[redacted]he gym, the practice site, and meetings. Observation will also be done on the information technology the team uses which includes email, Facebook groups, and a group texting service[redacted]From the observations, the researcher will analyze and present in subsequent papers how information flows through the team.

- Data will be gathered largely from natural observation of the researcher, who is a known participant of the team

**Opportunity Statement**

The [redacted] which has recently undergone significant changes in their student board structure and organization. The board has expanded from the traditional structure of [redacted] organization prompts this project and analysis of information from the newly structured board to its members.

- One opportunity is to enhance the effectiveness that the board members have with the current information technology when communicating with the team.
- Another opportunity is to reduce redundancy from the board when communicating, as the new structure is not well-established.

**Project Business Goal**

Improved efficiency and communication will allow the team to better understand tasks and goals provided from the board. This in turn will increase productivity at[redacted] and other sites. Increase in productivity from the team at the various sites will result in individual [redacted]

- Use of additional information technology, including Google Drive will allow for simpler documentation, for use by team members for [redacted]s and other documents.
- Improving communication methods from the board t[redacted]will enhance focus and make it more productive.

**Primary Project Objectives**

**Primary Project Objectives**

The researcher will observe[redacted] various sites where they meet, and he will also observe how [redacted]es various technologies to transmit information.

- As a known participant [redacted], observation will take place at all sites possible where the researcher attends.
- During observation, the researcher will take field notes which focus on how the board transmits information.

**Project Conditions**

**Project Assumptions**

There are several assumptions in regards to the completion of this project.

- [redacted] have access to all sites where the team practices and meets.
- Also, I have access to the various technologies used by the team.
- Further, I am associated with the board and can observe them as necessary.

**Project Risks**

There are also risks associated with the completion of this project.

- [redacted]the data gathered from observation may be biased and limited in perspective.
- Acting as a researcher for this project, but also a known participant [redacted]he board and members may not act as they naturally do as I'm observing and collecting data.

**Project Critical Success Factors** (Key Performance Indicators)

**Project Critical Success Factors**

For this project to proceed as stated and also succeed, these indicators must be fulfilled.

- The researcher will observe[redacted]t its sites and its use of technology for as much time as needed to obtain sufficient data.

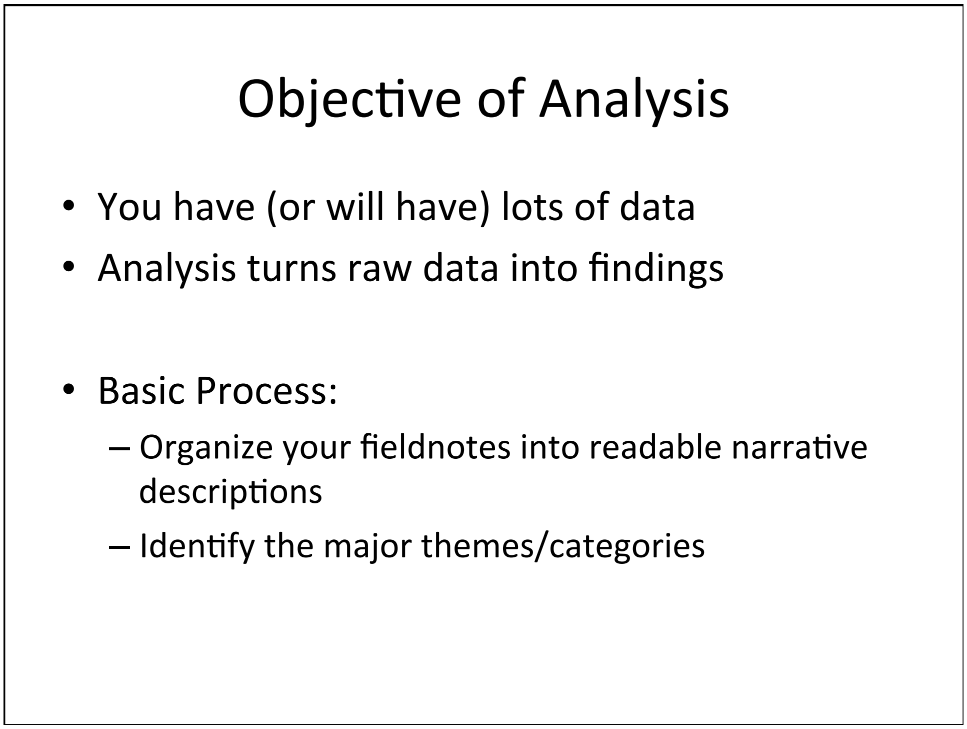
# Reminder of Writing Process

1) First draft delivered to group leader in time to be prepared for critique

2) Bring first draft and critiques to class

3) Group leader delivers all drafts to Steve at discussion, and each member saves critiques received to include in portfolio. This is a completion grade, but Steve and I will be looking to make sure people are basically understanding.

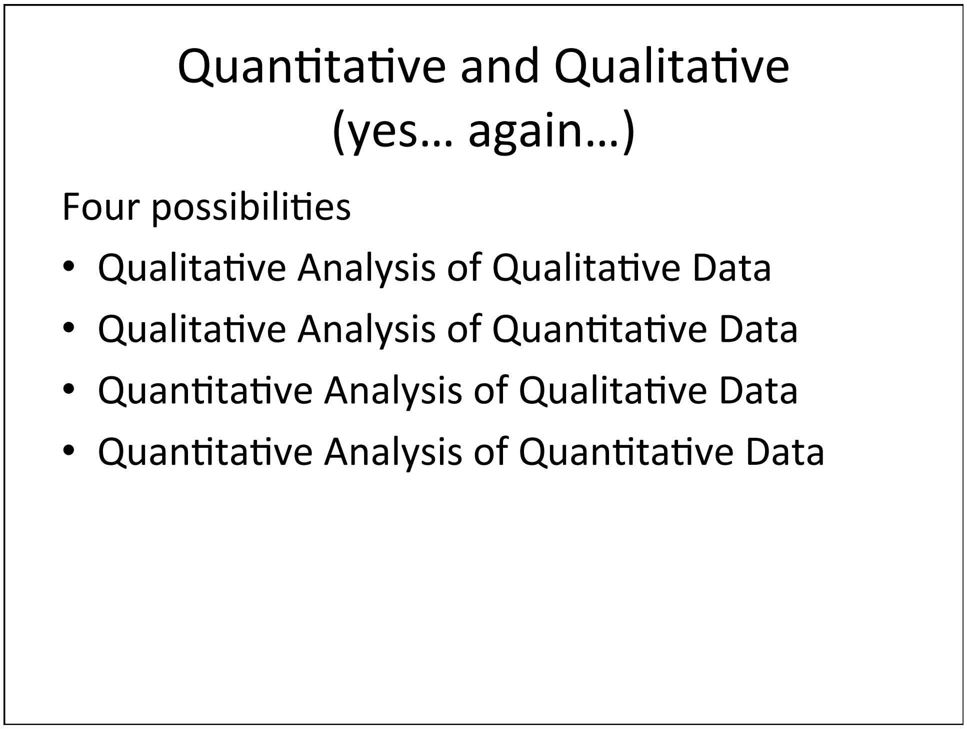
4) At the end, you turn in a portfolio made up of, for each project: Draft; critiques recieved; Final paper; reflection on critique/revision process (~250 words for this).



# Objective of Analysis

- You have (or will have) lots of data
- Analysis turns raw data into findings

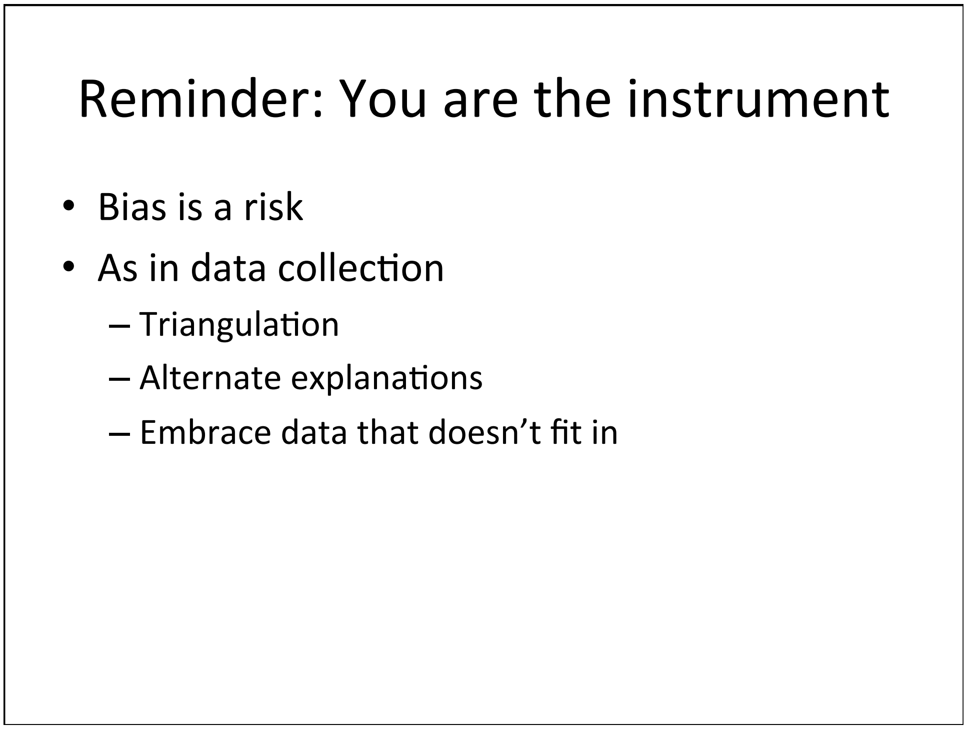
- Basic Process:
  - Organize your fieldnotes into readable narrative descriptions
  - Identify the major themes/categories



# Quantitative and Qualitative (yes… again…)

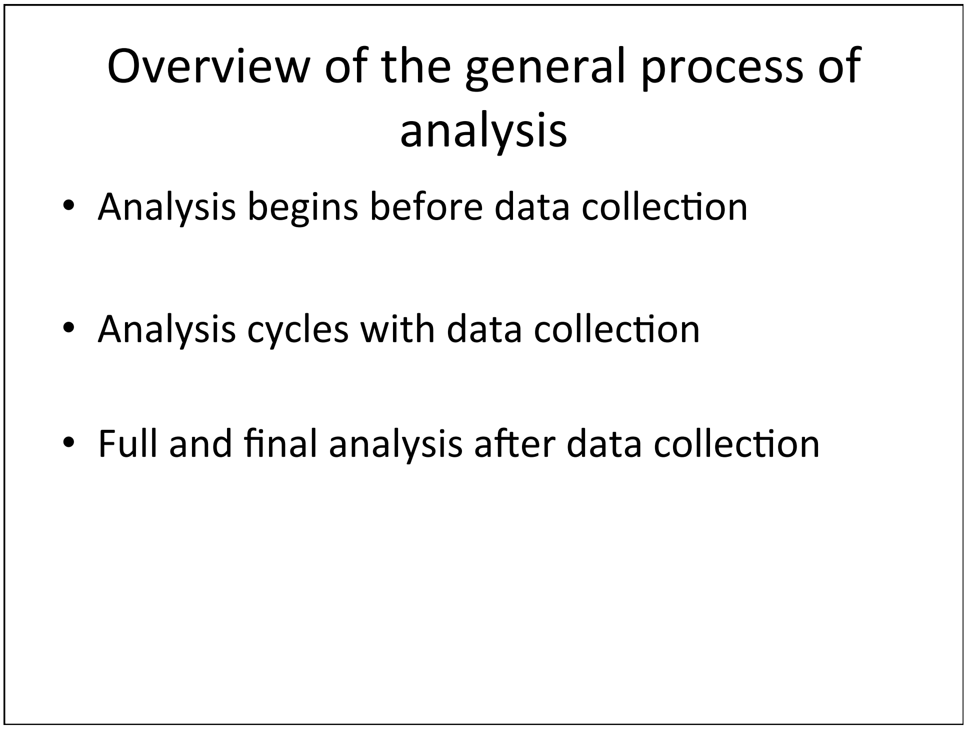
Four possibilities

- Qualitative Analysis of Qualitative Data
- Qualitative Analysis of Quantitative Data
- Quantitative Analysis of Qualitative Data
- Quantitative Analysis of Quantitative Data



## Reminder: You are the instrument

- Bias is a risk
- As in data collection
  - Triangulation
  - Alternate explanations
  - Embrace data that doesn't fit in



## Overview of the general process of analysis

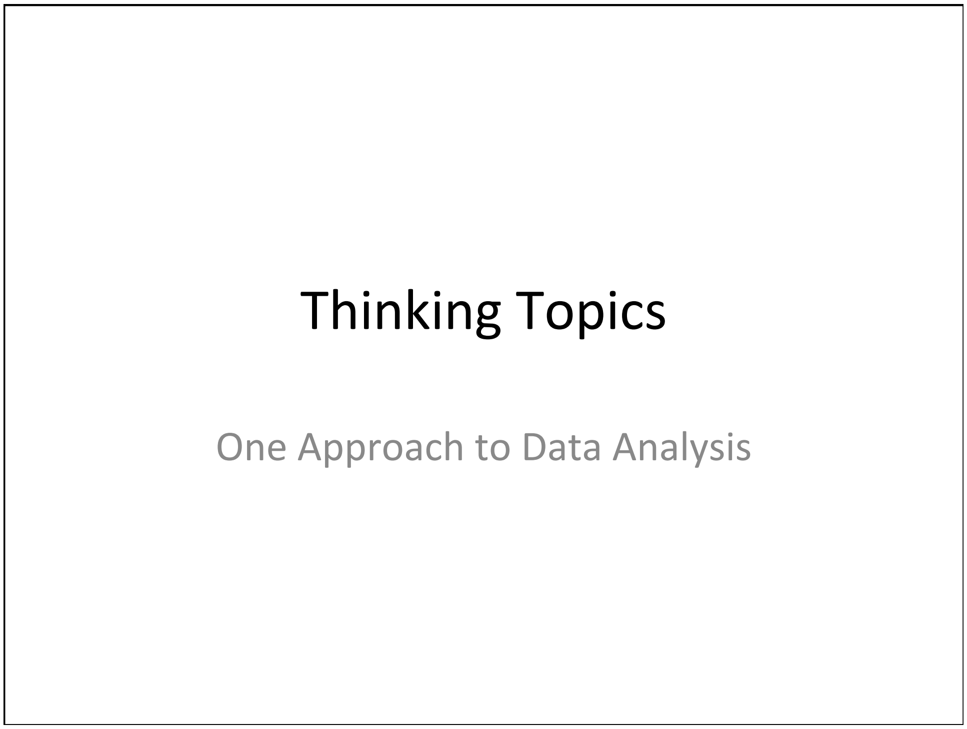
- Analysis begins before data collection
- Analysis cycles with data collection
- Full and final analysis after data collection

## Three common elements

- Data reduction (throughout)
  - The process of selection, focusing, simplifying, abstracting the raw data
- Data organization (grows as study occurs)
  - Organizing the reduced data in ways that allow you to begin to generate explanations
- Data explanation and verification (intense at end)
  - Drawing conclusions from the explanations
  - Testing the conclusions drawn: verifying their plausibility

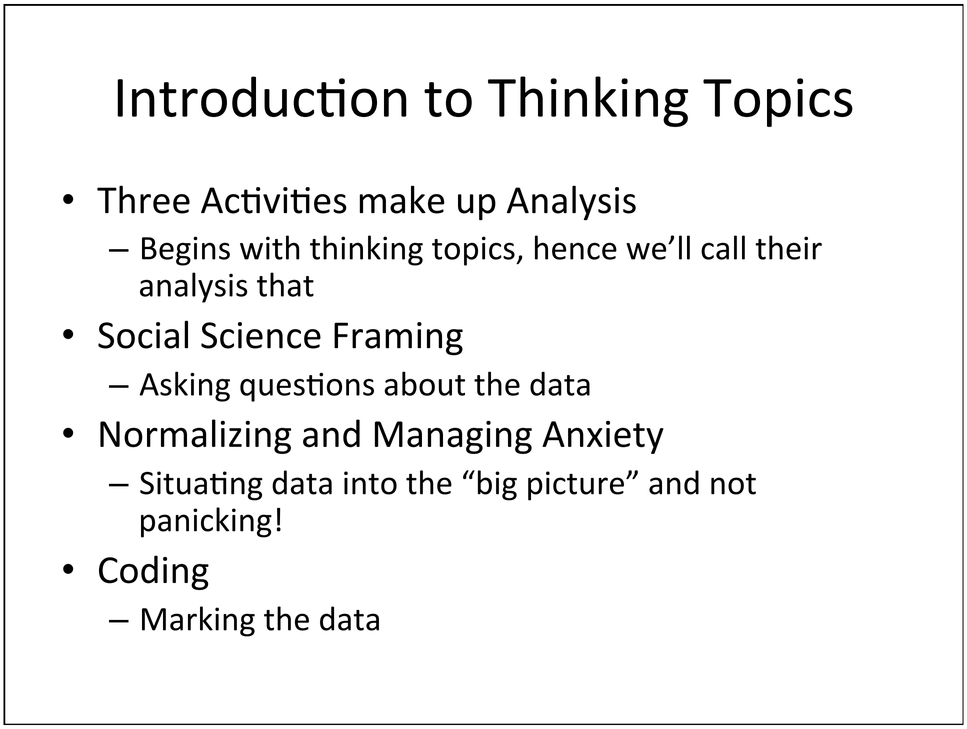
## Inductive/Deductive

- Inductive Analysis
  - Let the analytic themes emerge from the study of the data
- Deductive Analysis
  - Starting with a hypothesis for data analysis
- Might do both in one study... why?



# Thinking Topics

One Approach to Data Analysis



## Introduction to Thinking Topics

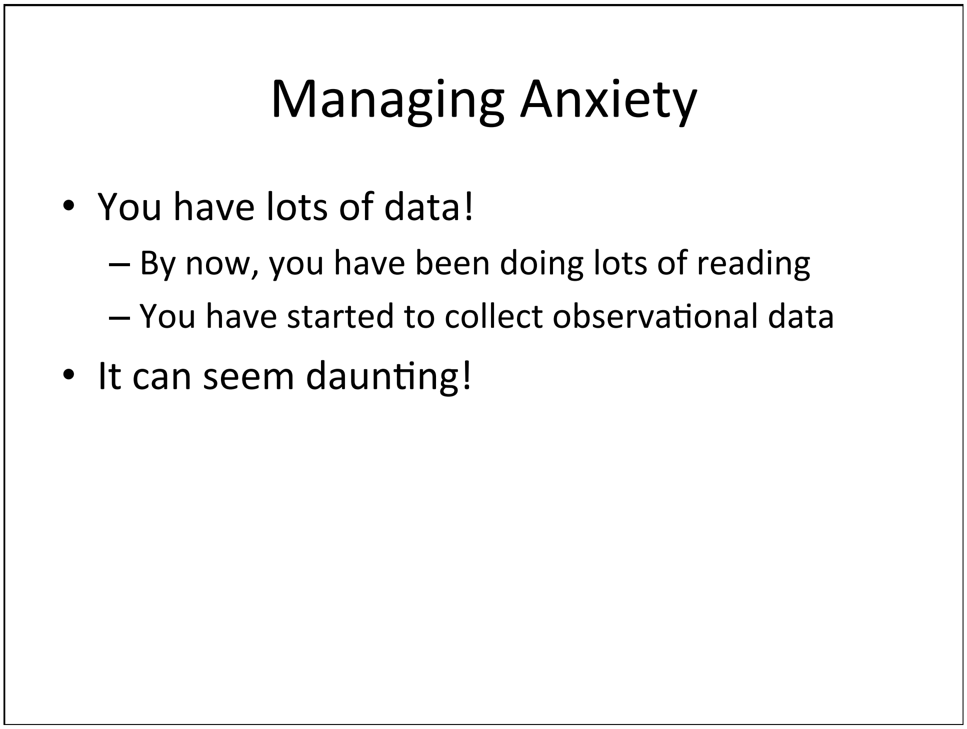
- Three Activities make up Analysis
  - Begins with thinking topics, hence we'll call their analysis that
- Social Science Framing
  - Asking questions about the data
- Normalizing and Managing Anxiety
  - Situating data into the "big picture" and not panicking!
- Coding
  - Marking the data

# Normalizing

- Simply put, it means to step back and ask why you've been studying this
- In other words, you're normalizing against the problem statement
- Why normalize?

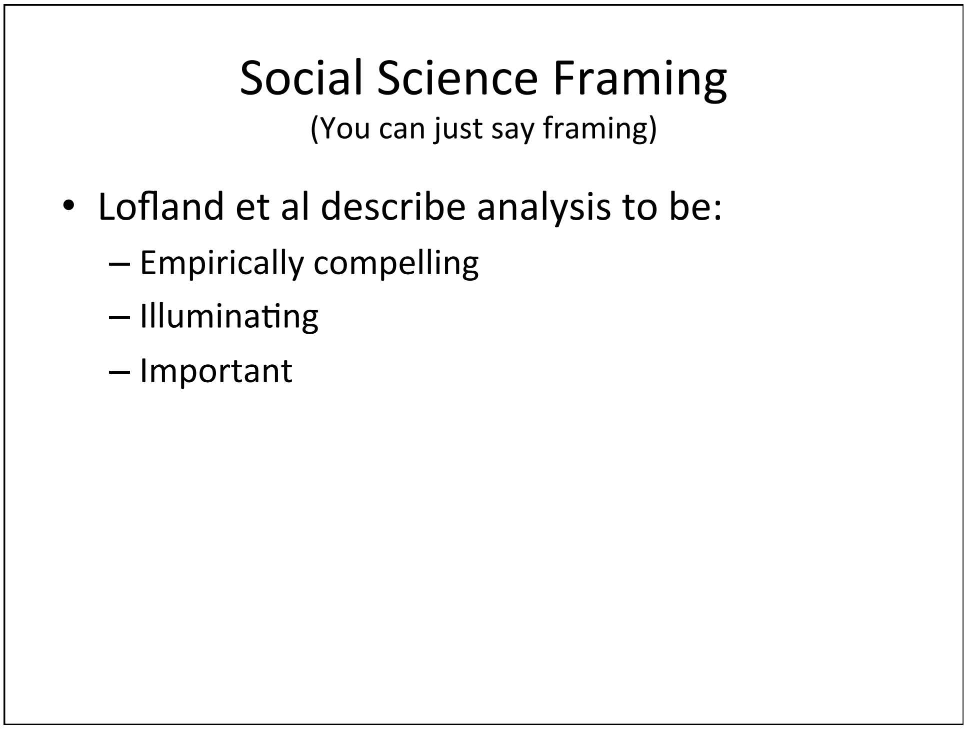
# Possible Sources for Normalizing

- It depends on the question and origins of your problem
  - What questions need to be answered, what type of answers constitute a result?
- Sources come from anything that reminds you why you're doing this and what type of outcome you want
- Possible Sources:
  - The notes you took when where you were first assigned the project
  - The notes you took about the literature you read



# Managing Anxiety

- You have lots of data!
  - By now, you have been doing lots of reading
  - You have started to collect observational data
- It can seem daunting!



# Social Science Framing

(You can just say framing)

- Lofland et al describe analysis to be:
  - Empirically compelling
  - Illuminating
  - Important

# Framing involves asking 8 basic questions

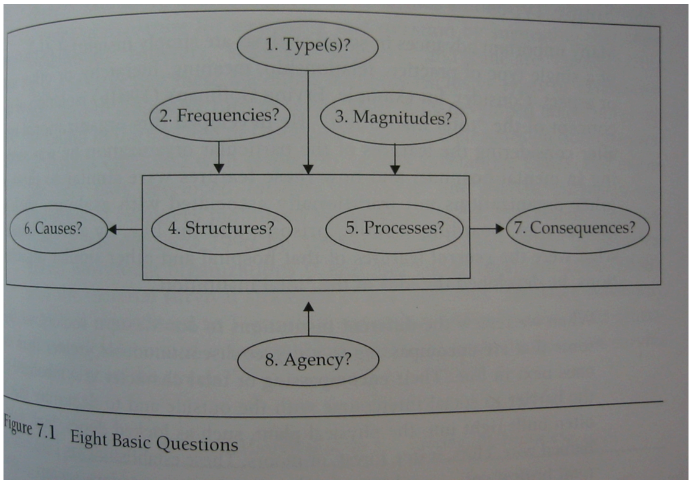


Figure 7.1 Eight Basic Questions

# 1: Type

- What is this set of things I see before me?
- What is this an instance of? What type of experience?

# 2: Frequency

- How often do we see something in the data set?
- Descriptive statistics!
- Useful for characterizing how many people experienced something

# 3: Magnitudes

- What was the size, intensity or strength of that?
- Did anyone tell you about something that might have endangered a patient's life?

# 4: Structure

- Arrangement of social groups
- Both formal and informal must be considered

# 5: Processes

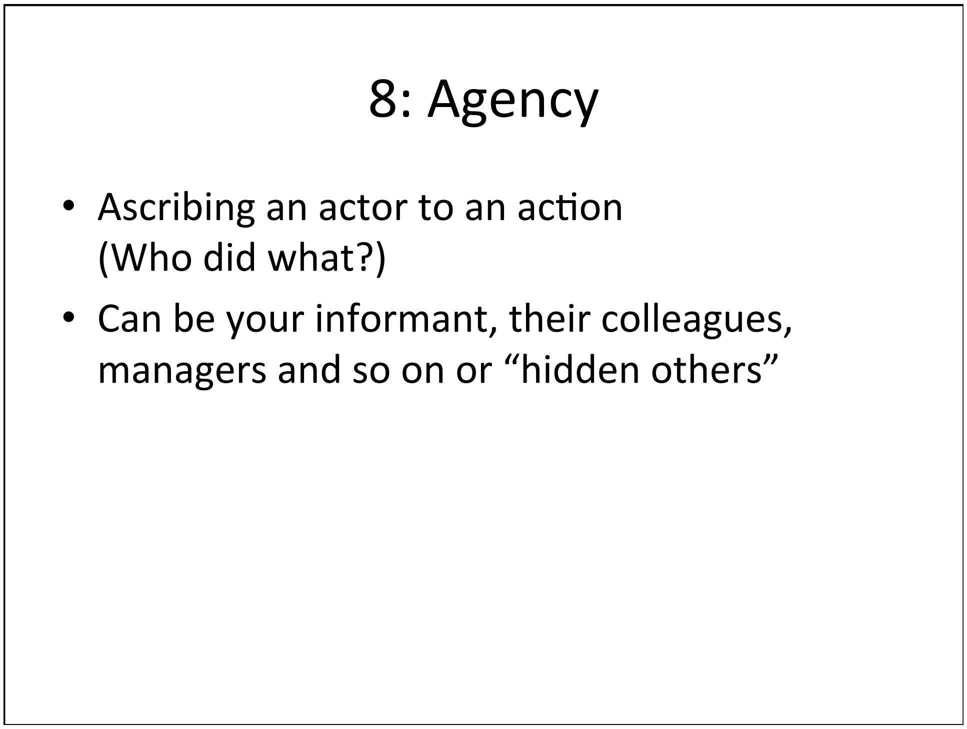
- How does something happen?
  (What are the steps, stages, that occur for an instance to occur)
- For example, how do people seek help?

## 6: Causes

- What circumstances must occur for event to happen?
  - Lofland et al say cause, but this NOT cause like cause and effect
  - More like… antecedents, context, or setting events

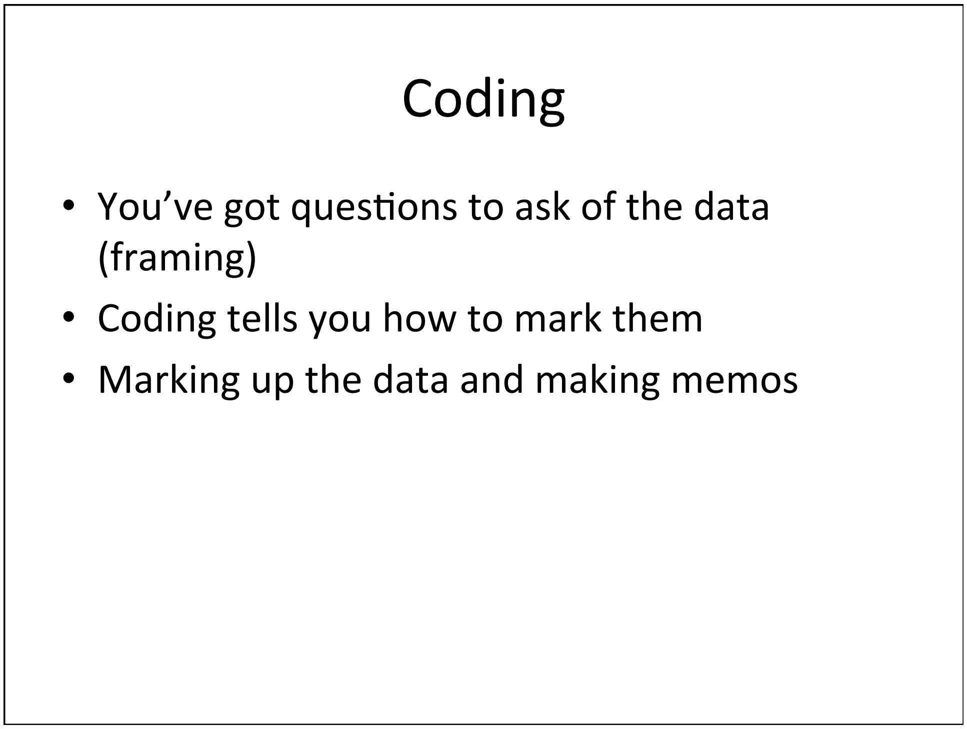
## 7: Consequences

- What happened next, as a result
- Looking for consequences gives your results a temporal feel



## 8: Agency

- Ascribing an actor to an action  
  (Who did what?)
- Can be your informant, their colleagues, managers and so on or “hidden others”



## Coding

- You’ve got questions to ask of the data (framing)
- Coding tells you how to mark them
- Marking up the data and making memos

## Marking the Data

- Literally involves going through all the data and asking questions
- When connections happen you go to the second part...

## Memoing

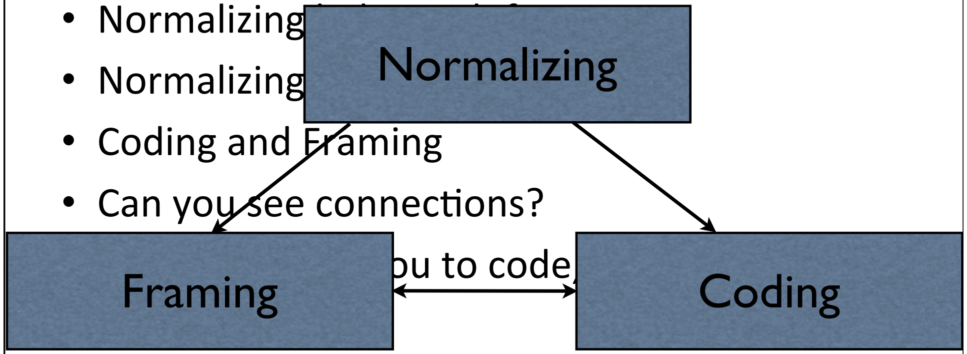
- When you find links among data write them down as a memo
- A memo is a series of notes that just explain what seems to be linked and why

## Timeline of Analysis

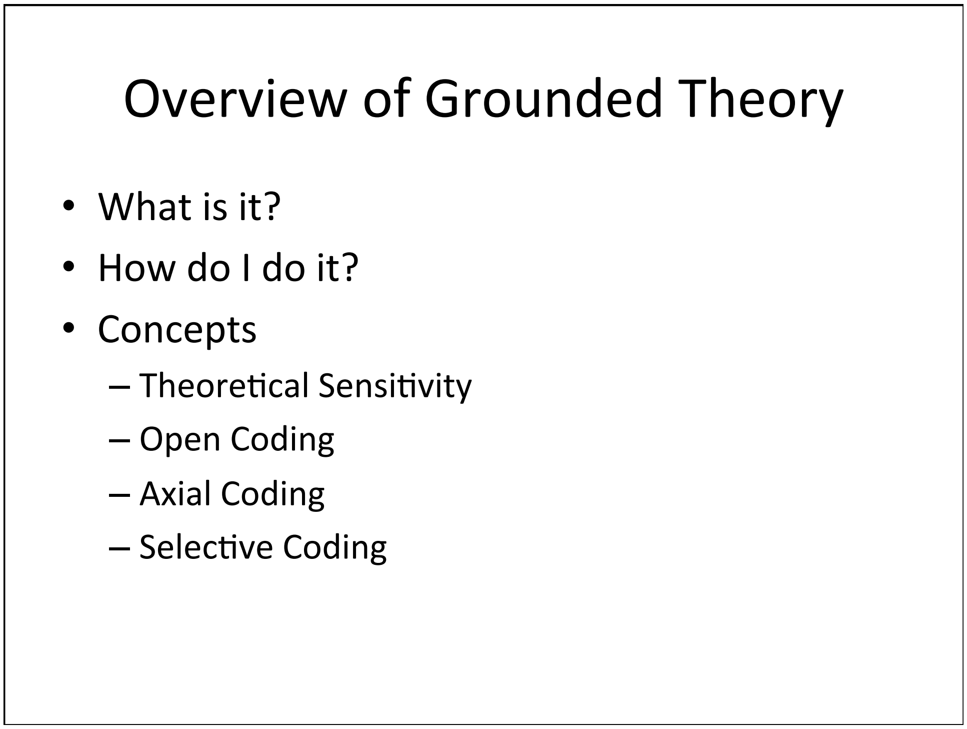
- Before analysis, data include individual notes
- Analysis takes this order and changes it to another one (topical)
- Post-analysis: the report
- Use the report, what needs to be answered and for whom as another mechanism to help you transform the interviews into an analyzed collection

## Moving Among Activities

- Normalizing
- Normalizing
- Coding and Framing
- Can you see connections?
- ou to code,

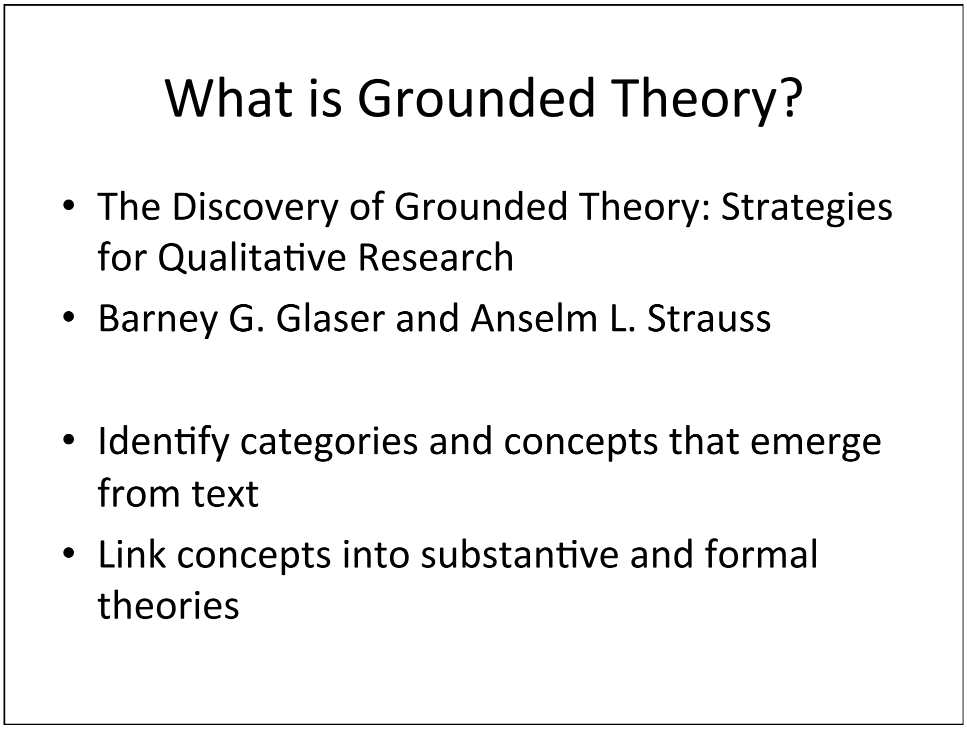


- At the same time, coding helps with framing



## Overview of Grounded Theory

- What is it?
- How do I do it?
- Concepts
  - – Theoretical Sensitivity
  - – Open Coding
  - – Axial Coding
  - – Selective Coding



## What is Grounded Theory?

- The Discovery of Grounded Theory: Strategies for Qualitative Research
- Barney G. Glaser and Anselm L. Strauss

- Identify categories and concepts that emerge from text
- Link concepts into substantive and formal theories

## Mechanics of GT

- Read through field notes to produce *analytic categories*
- As categories emerge pull data from categories together and compare them
- Think about how categories fit together into an explanation, or model
- Take models developed and check them against the data
- Present results using examples from the data

## Getting started with GT

- Theoretical sensitivity is about developing insight into data
- Use literature: readings on theory, research, and supporting evidence
- Use professional experience

## Example: Chronic Cancer Care

- Literature: medical AND human-computer interaction
- Professional experience: get others insights
- Cultural experience:  sat on cancer-related list-serves

## Step One: Open Coding

- Process of breaking down, examining, comparing, conceptualizing and categorizing data
- Labeling phenomena
- Discovering categories
- Developing categories: properties and dimensions

## Open Coding: Labeling Phenomena

- Break down raw full descriptive field notes
- By asking questions about notes
- For each phenomenon (incident, idea or event)
- Give each discrete phenomenon a name
- Compare it to others already discovered

## Open Coding: Discovering Categories

- Next, group concepts into *categories*
- Now you're asking questions about the concepts and the category
  - What are the phenomena in this category about?
  - What are they instances of?
  - Use answer to label category

## Open Coding: Properties and Dimensions

- Properties: characteristics or attributes of a category
- Dimensions:  locations of a property along a continuum

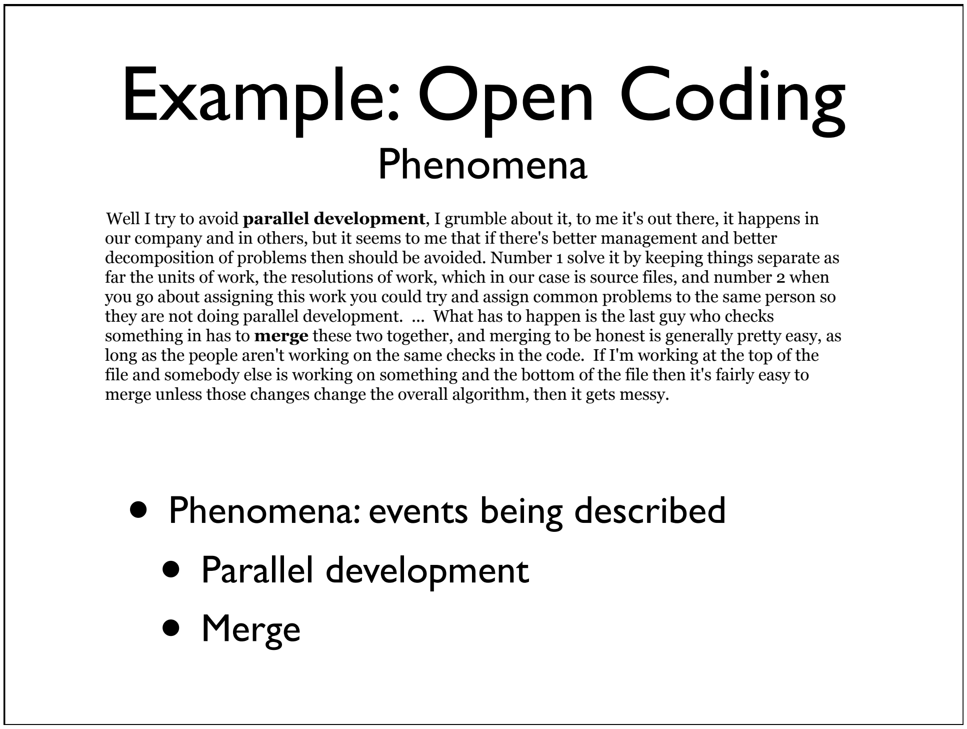
# Example

## Open Coding

- Context
  - Interview with a software developer about her work

Well I try to avoid parallel development, I grumble about it, to me it's out there, it happens in our company and in others, but it seems to me that if there's better management and better decomposition of problems then should be avoided. Number 1 solve it by keeping things separate as far the units of work, the resolutions of work, which in our case is source files, and number 2 when you go about assigning this work you could try and assign common problems to the same person so they are not doing parallel development.  ...  What has to happen is the last guy who checks something in has to merge these two together, and merging to be honest is generally pretty easy, as long as the people aren't working on the same checks in the code.  If I'm working at the top of the file and somebody else is working on something and the bottom of the file then it's fairly easy to merge unless those changes change the overall algorithm, then it gets messy.

**Example courtesy of Beki Grinter at Georgia Tech**

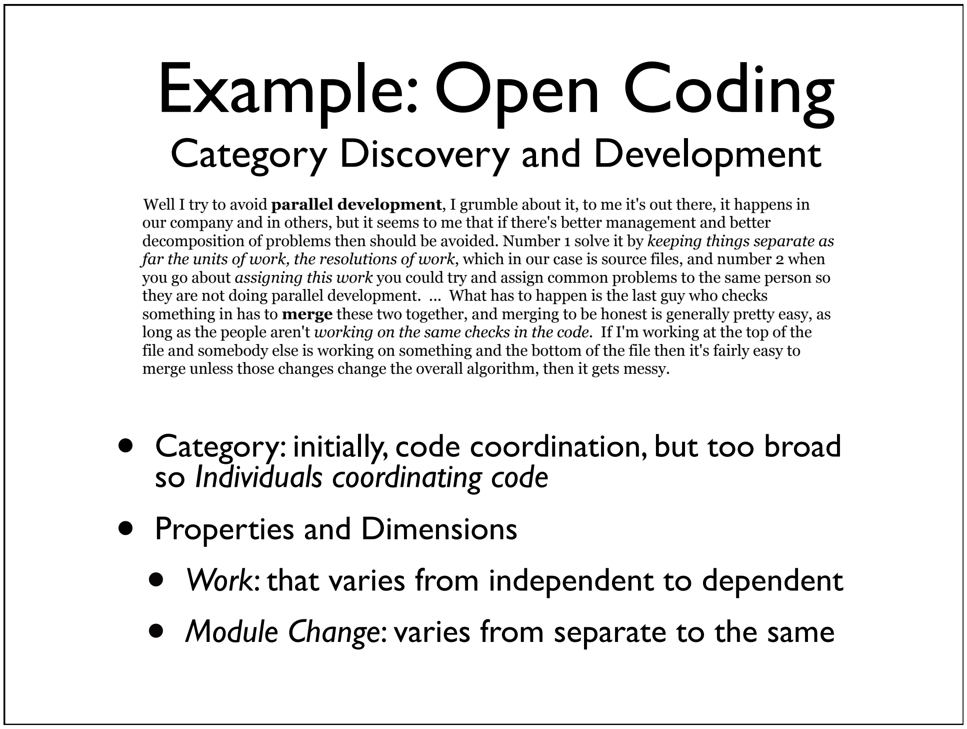


# Example: Open Coding

## Phenomena

Well I try to avoid **parallel development**, I grumble about it, to me it's out there, it happens in our company and in others, but it seems to me that if there's better management and better decomposition of problems then should be avoided. Number 1 solve it by keeping things separate as far the units of work, the resolutions of work, which in our case is source files, and number 2 when you go about assigning this work you could try and assign common problems to the same person so they are not doing parallel development. ... What has to happen is the last guy who checks something in has to **merge** these two together, and merging to be honest is generally pretty easy, as long as the people aren't working on the same checks in the code. If I'm working at the top of the file and somebody else is working on something and the bottom of the file then it's fairly easy to merge unless those changes change the overall algorithm, then it gets messy.

- Phenomena: events being described
  - Parallel development
  - Merge



# Example: Open Coding

## Category Discovery and Development

Well I try to avoid **parallel development**, I grumble about it, to me it's out there, it happens in our company and in others, but it seems to me that if there's better management and better decomposition of problems then should be avoided. Number 1 solve it by *keeping things separate as far the units of work, the resolutions of work*, which in our case is source files, and number 2 when you go about *assigning this work* you could try and assign common problems to the same person so they are not doing parallel development. ... What has to happen is the last guy who checks something in has to **merge** these two together, and merging to be honest is generally pretty easy, as long as the people aren't *working on the same checks in the code*. If I'm working at the top of the file and somebody else is working on something and the bottom of the file then it's fairly easy to merge unless those changes change the overall algorithm, then it gets messy.

- Category: initially, code coordination, but too broad so *Individuals coordinating code*
- Properties and Dimensions
  - *Work*: that varies from independent to dependent
  - *Module Change*: varies from separate to the same

# Step Two: Axial Coding

Taking categories and identifying

- The *conditions* that give rise to it
- *Context* into which it is embedded
- *Action/interaction strategies* in which it is handled, managed, carried out
- *Consequences* of those strategies

# Axial Coding

## Casual Conditions and Context

- Causal Conditions are events, incidents that lead to the occurrence of a category
- Context is the set of properties that pertain to a category

# Axial Coding

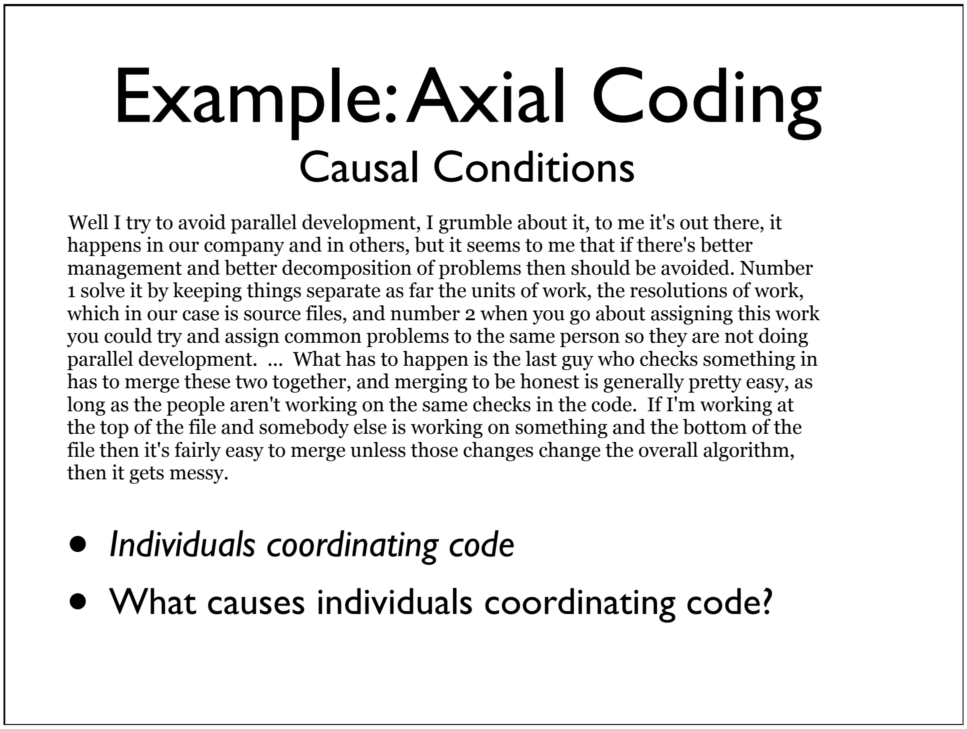
Intervening Conditions and (Inter)Actional Strategies

- Intervening Conditions are broader structural context pertaining to category
- Action/Interactional Strategies
  - What actions do individuals take with respect to the category?
  - How do groups or collectives interact and act with respect to the category?

# Axial Coding

Consequences

- Consequences
  - There are always consequences :-)
  - Here it means the outcomes of the (inter)actional strategies

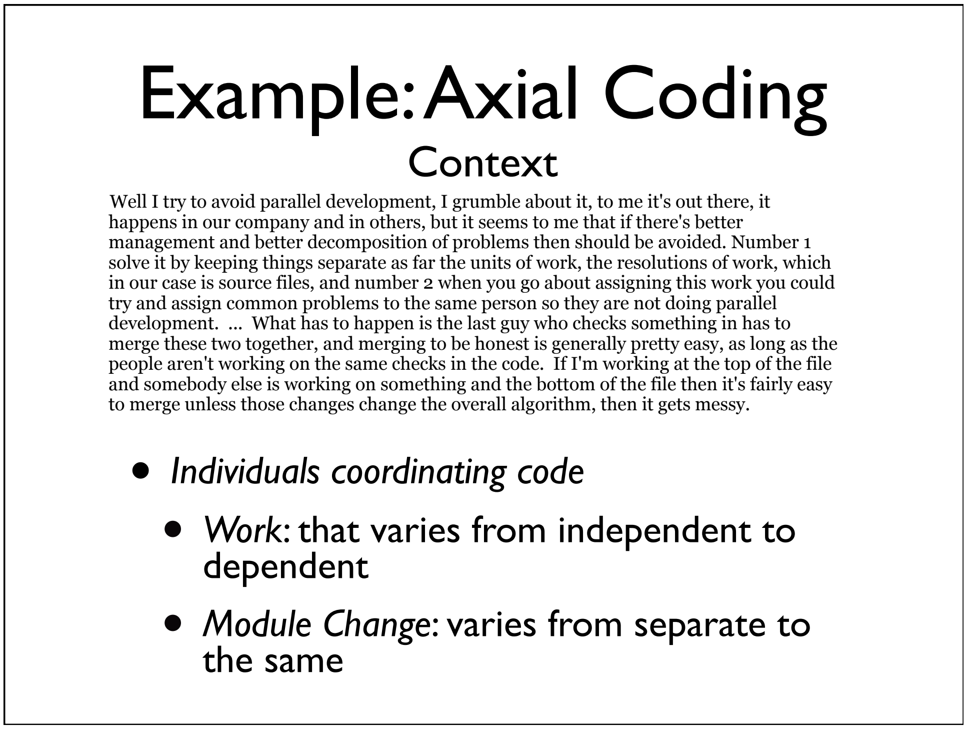


# Example: Axial Coding

## Causal Conditions

Well I try to avoid parallel development, I grumble about it, to me it's out there, it happens in our company and in others, but it seems to me that if there's better management and better decomposition of problems then should be avoided. Number 1 solve it by keeping things separate as far the units of work, the resolutions of work, which in our case is source files, and number 2 when you go about assigning this work you could try and assign common problems to the same person so they are not doing parallel development. ... What has to happen is the last guy who checks something in has to merge these two together, and merging to be honest is generally pretty easy, as long as the people aren't working on the same checks in the code. If I'm working at the top of the file and somebody else is working on something and the bottom of the file then it's fairly easy to merge unless those changes change the overall algorithm, then it gets messy.

- *Individuals coordinating code*
- What causes individuals coordinating code?

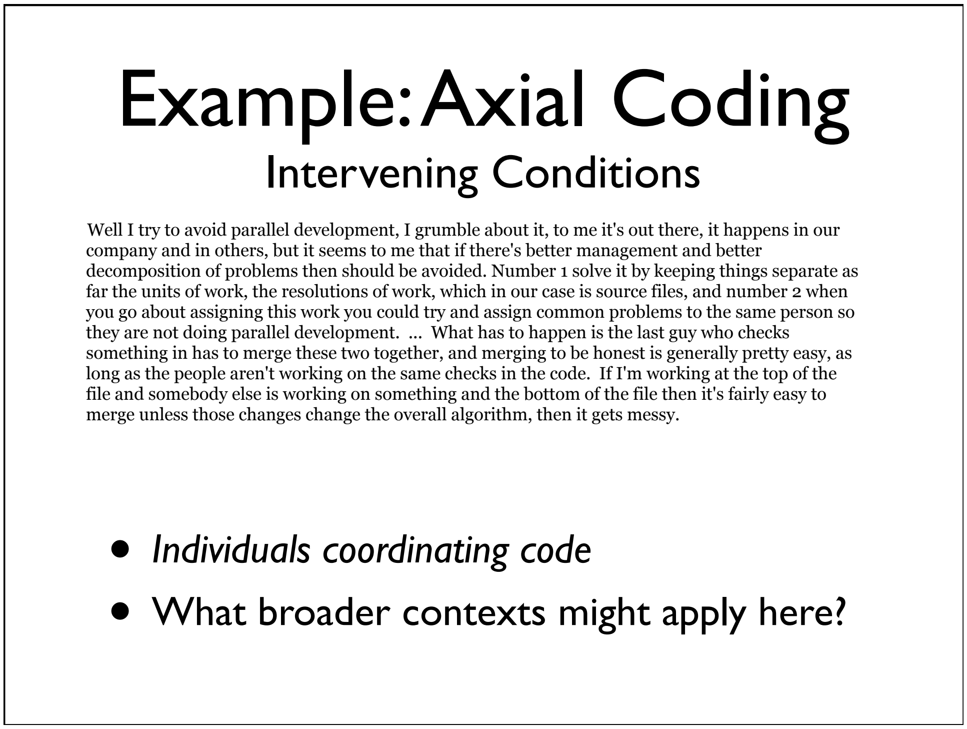


# Example: Axial Coding

## Context

Well I try to avoid parallel development, I grumble about it, to me it's out there, it happens in our company and in others, but it seems to me that if there's better management and better decomposition of problems then should be avoided. Number 1 solve it by keeping things separate as far the units of work, the resolutions of work, which in our case is source files, and number 2 when you go about assigning this work you could try and assign common problems to the same person so they are not doing parallel development. ... What has to happen is the last guy who checks something in has to merge these two together, and merging to be honest is generally pretty easy, as long as the people aren't working on the same checks in the code. If I'm working at the top of the file and somebody else is working on something and the bottom of the file then it's fairly easy to merge unless those changes change the overall algorithm, then it gets messy.

- *Individuals coordinating code*
  - *Work*: that varies from independent to dependent
  - *Module Change*: varies from separate to the same

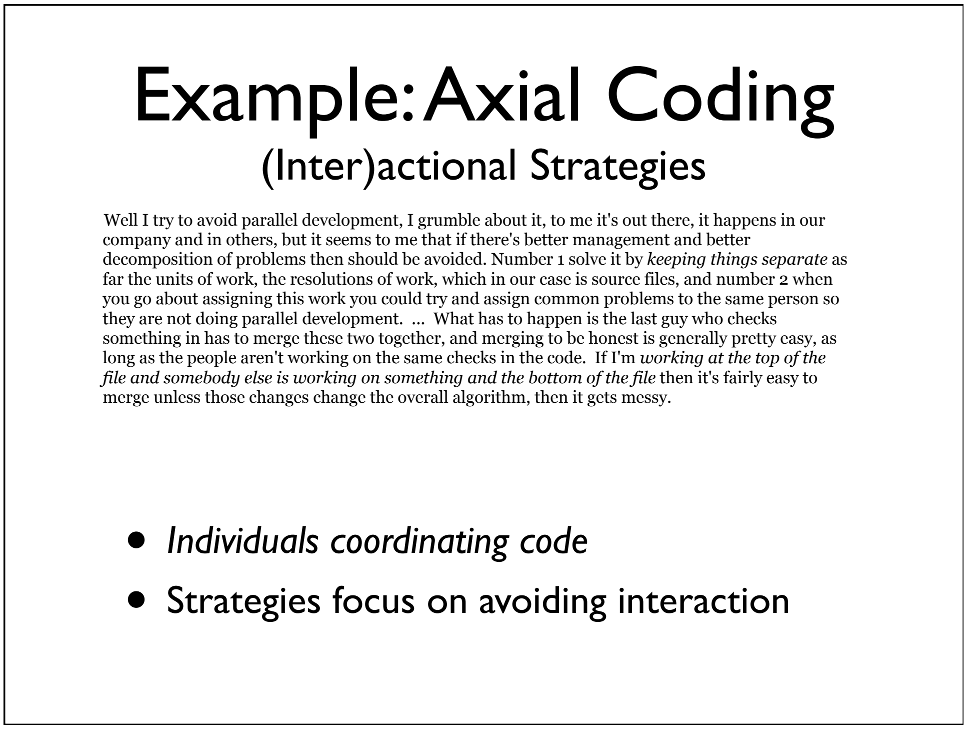


# Example: Axial Coding

## Intervening Conditions

Well I try to avoid parallel development, I grumble about it, to me it's out there, it happens in our company and in others, but it seems to me that if there's better management and better decomposition of problems then should be avoided. Number 1 solve it by keeping things separate as far the units of work, the resolutions of work, which in our case is source files, and number 2 when you go about assigning this work you could try and assign common problems to the same person so they are not doing parallel development. ... What has to happen is the last guy who checks something in has to merge these two together, and merging to be honest is generally pretty easy, as long as the people aren't working on the same checks in the code. If I'm working at the top of the file and somebody else is working on something and the bottom of the file then it's fairly easy to merge unless those changes change the overall algorithm, then it gets messy.

- *Individuals coordinating code*
- What broader contexts might apply here?



# Example: Axial Coding

## (Inter)actional Strategies

Well I try to avoid parallel development, I grumble about it, to me it's out there, it happens in our company and in others, but it seems to me that if there's better management and better decomposition of problems then should be avoided. Number 1 solve it by *keeping things separate* as far the units of work, the resolutions of work, which in our case is source files, and number 2 when you go about assigning this work you could try and assign common problems to the same person so they are not doing parallel development. ... What has to happen is the last guy who checks something in has to merge these two together, and merging to be honest is generally pretty easy, as long as the people aren't working on the same checks in the code. If I'm *working at the top of the file and somebody else is working on something and the bottom of the file* then it's fairly easy to merge unless those changes change the overall algorithm, then it gets messy.

- *Individuals coordinating code*
- Strategies focus on avoiding interaction

# Example: Axial Coding

## Consequences

Well I try to avoid parallel development, I grumble about it, to me it's out there, it happens in our company and in others, but it seems to me that if there's better management and better decomposition of problems then should be avoided. Number 1 solve it by keeping things separate as far the units of work, the resolutions of work, which in our case is source files, and number 2 when you go about assigning this work you could try and assign common problems to the same person so they are not doing parallel development. ... What has to happen is the last guy who checks something in has to merge these two together, and *merging to be honest is generally pretty easy, as long as the people aren't working on the same checks in the code.* If I'm working at the top of the file and somebody else is working on something and the bottom of the file then it's fairly easy to merge unless those changes change the overall algorithm, then it gets *messy*.

- *Individuals coordinating code*
- Not keeping things separate -- leads to what consequence?

# Step Three: Selective Coding

- The process of selecting the core category
- Then relating it to other categories
- Three processes
  - Explicating the story line
  - Relating other categories to the core
  - Validating the story line

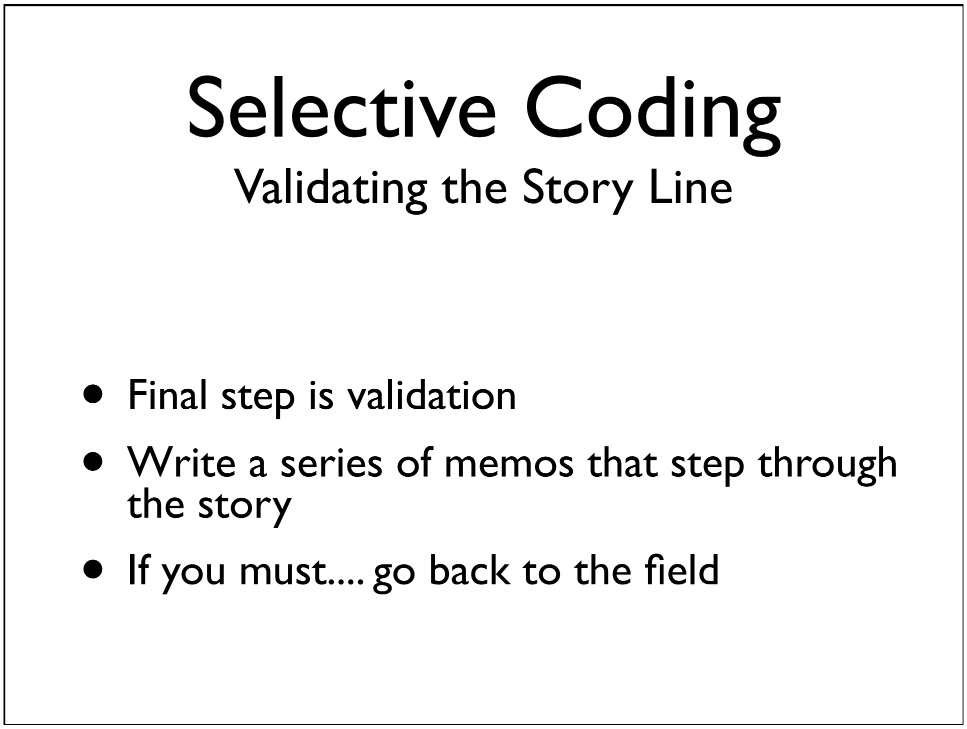
## What's this story line thing all about?

- Commit, commit commit!
- Finding the story
  - Ask yourself what seems the most striking/interesting
  - Does one category seem more central?

# Selective Coding

## Relating Categories to the Story Line

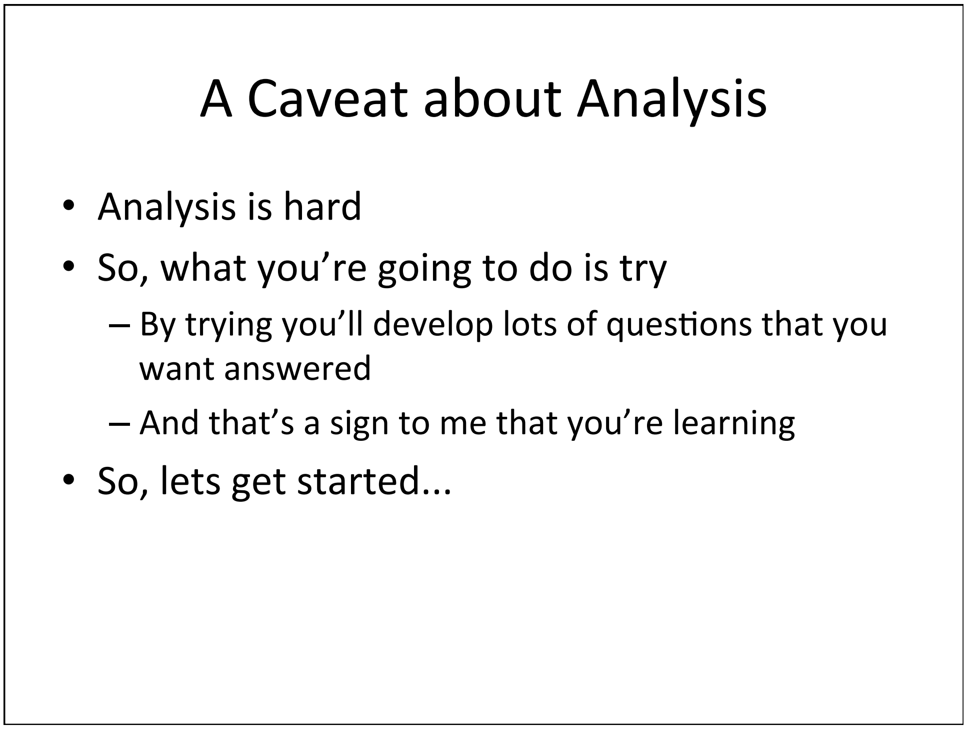
- First, outline core's properties and dimensions
- Relate other categories to the core



# Selective Coding

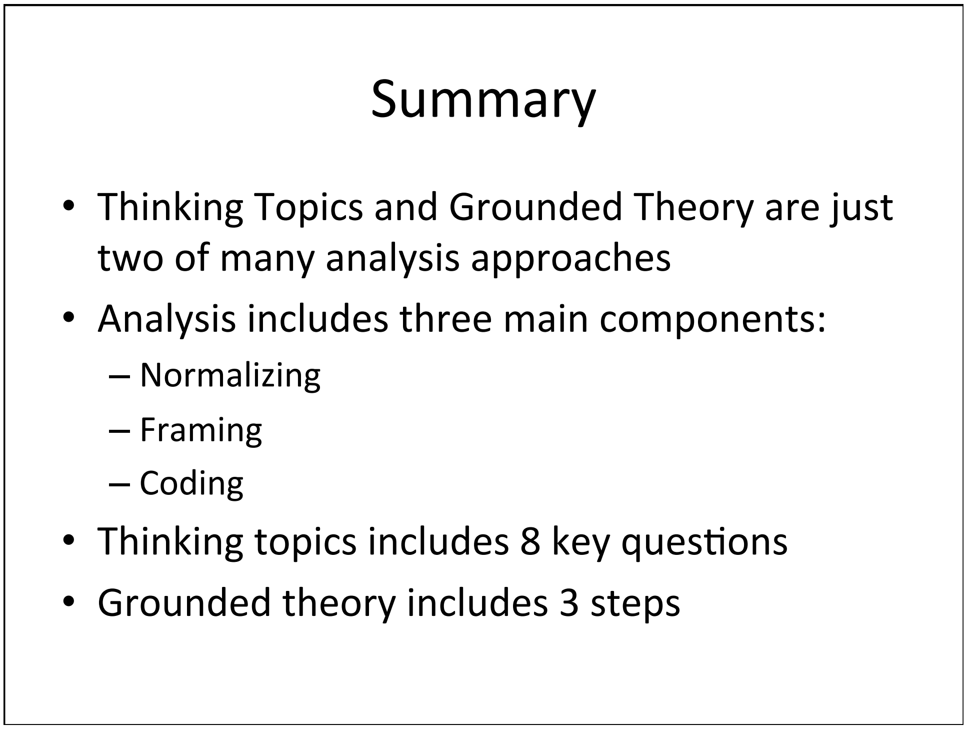
## Validating the Story Line

- Final step is validation
- Write a series of memos that step through the story
- If you must.... go back to the field



# A Caveat about Analysis

- Analysis is hard
- So, what you're going to do is try
  - By trying you'll develop lots of questions that you want answered
  - And that's a sign to me that you're learning
- So, lets get started...



## Summary

- Thinking Topics and Grounded Theory are just two of many analysis approaches
- Analysis includes three main components:
  - Normalizing
  - Framing
  - Coding
- Thinking topics includes 8 key questions
- Grounded theory includes 3 steps