

Constructing Requirements for Health Information Systems

Texas State University Health Information Management
Spring Conference
April 23, 2015

David Gibbs, MS, CPHIMS, CISSP
Account Chief Technologist
Hewlett Packard Enterprise Services

Agenda

- Problem
- Research Questions
- Perspective
- Challenges Identified
- Requirements as Knowledge
- Applying to HIM Lifecycles
- Recommendations



The Problem



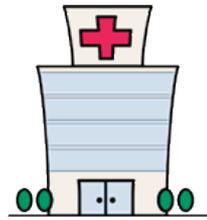
- *“The hardest single part of building a software system is deciding precisely what to build.”*
– Fred Brooks, 1987
- *“The difficulties encountered in accurately capturing system **requirements** have been suggested to be a **major factor in the failure** of 90% of large software projects.”*
– Davis, Fuller, Tremblay & Berndt, 2006
- *“It is clear that **requirements elicitation has not been done well** and that failure causes considerable problems.”*
– Davey & Cope, 2008

The Problem Continues...

- *“Studies show that 80 to 85 percent of project **failures are due to incorrect requirements.**”*
 - Don Reinertsen, 2011

- *“Research suggests that **requirements engineering is not well done in industry.**”*
 - Phillip Laplante, 2014

The Problem Applies to HIM



- *“A great opportunity for promoting HIT was lost because of **inadequate communications with stakeholders**. Even when project technologists have clinical background, they do not speak for everyone.”*

– Leviss, 2010
- *“Clinical information technology (IT) projects are highly complex **social endeavors** in unforgiving medical environments that happen to involve computers, not IT projects that happen to involve clinicians.”*

– Leviss, 2010

Applies to all stakeholders: consumers and providers of solutions

The Problem in Pictures



How the customer explained it



How the project leader understood it



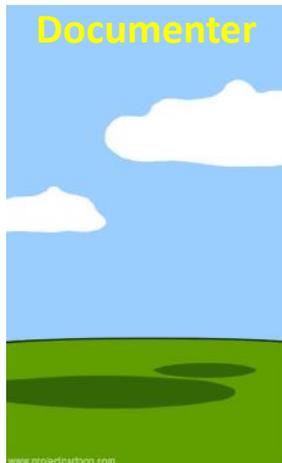
How the analyst designed it



How the programmer wrote it



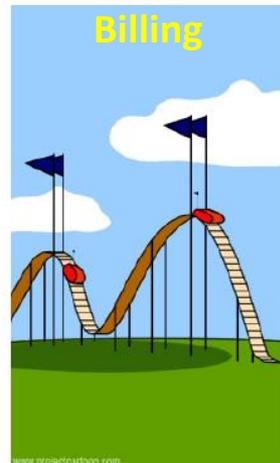
How the business consultant described it



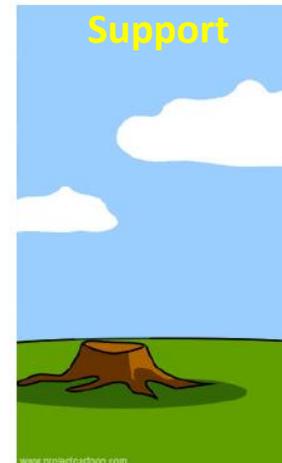
How the project was documented



What operations installed



How the customer was billed



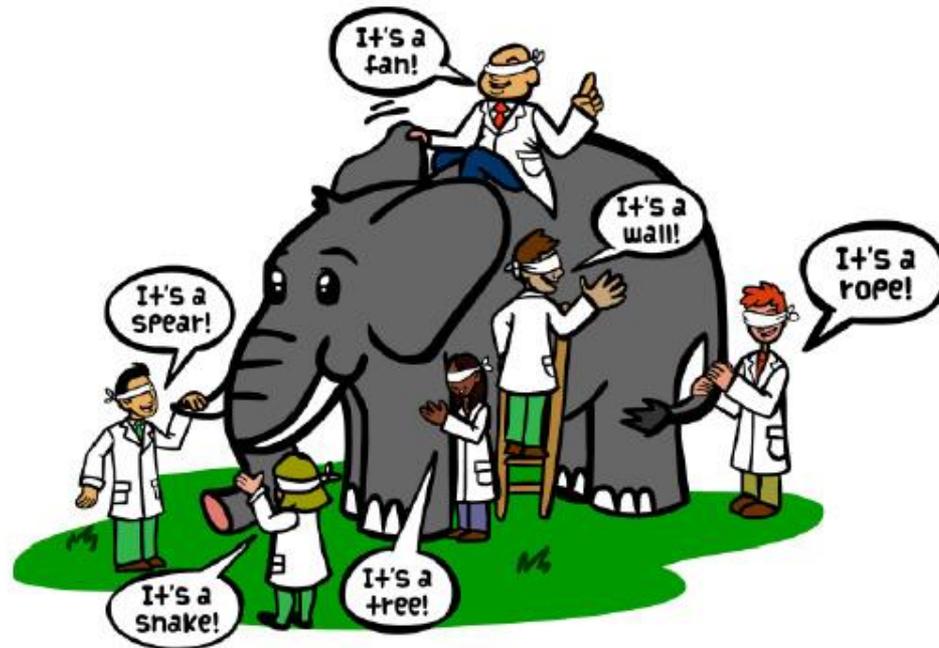
How it was supported



What the customer really needed

Summary of the Problem

Lack of common vision and understanding of the requirements



<http://theblindelephant.com>

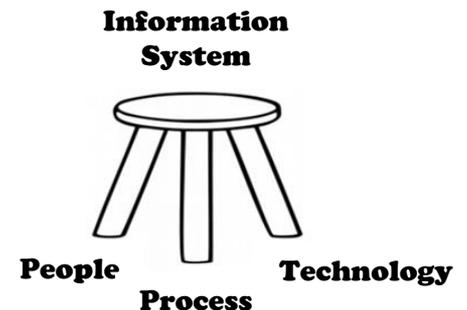
The Question



- Fundamental Question
 - Why is it so difficult to determine requirements?
- Refined into the Research Question
 - What challenges are encountered by stakeholders when identifying requirements for information systems?

Researcher's Perspective

- Experience
 - 30 years designing and delivering information systems
 - Direct observation across many projects
- Access
 - To experienced practitioners and their professional network
 - As information systems architect for a large technology company
 - From supporting complex projects for healthcare and government
- Interests
 - Adult, Professional & Community Education
 - People-side of information systems



Focus

- I am NOT
 - Replacing current tools
 - Introducing a formal method
- I am
 - Suggesting a **change of attitude**
 - Focused on getting to the **right** requirements



Sensitizing Concepts



- Requirements are subjective
 - Based on individual and consensus perspectives and priorities
 - Must be reconciled among stakeholders to resolve conflicts
 - Systems don't have requirements, people do

- Constructivism vs. Positivism

- Requirements must be constructed through collaboration
- *“The fundamental problem here is that folks believe that underlying every project there's some absolute, discoverable set of requirements.”*



– Thomas & Hunt, 2004

- Stakeholder Turnover

- Adding or removing a stakeholder may change requirements

Observations and “gut feelings”, but I didn't know *why* and *how*

Interview Prompts for Critical Incidents



Each participant:

1. Think of the most recent time when you were involved with an information system that was either purchased or developed and ***did not*** meet the requirements
2. ... incident with the most significant impact
3. ... generated a feeling of here we go again

Interview Questions



Questions:

1. Tell me about the project. (*Validation, warm-up*)
2. What factors led to requirements not being met?
3. What could have been done differently to improve the likelihood of success?
4. What was your role on the project? (*Clarifies perspective*)
5. How did the project turn out?

Example: Resolving Conflicting Requirements

- *“Instead of dealing with that issue then, fully and completely, to adequate resolution, we kind of tabled the issue. And in tabling the issue it didn’t go away and when it came time to deliver it was still there and hadn’t been resolved. I see that very consistently in the requirements processes that **people tend to push off things that are difficult** or not politically expedient and those things never go away and they ultimately **reveal themselves at a very inopportune time after a lot of time and a lot of money has been spent.**”*
 - Stakeholder on a project that was cancelled due to disappointing results

Identify show stoppers sooner rather than later

Example: Power



- *“These are not people you want to disappoint. They are more trusted than you are with the boss and the boss’ boss. And so **there is political cost to [speaking up].**”*
 - Stakeholder who knew failure was inevitable but kept quiet
- *“The staff in the surgical clinic **thought that they had no influence** in the system design and development.”*
 - Stakeholders who did not feel empowered to voice requirements
- *“They talked to my boss, who was an arrogant SOB, and **they did not talk to anybody else.** They just came to all these conclusions that he knew what he was talking about.”*
 - Stakeholder on a project that exceeded budget and disappointed users

Power issues may obstruct openness and suppress requirements

Example: Turnover



- *“You would have individuals that came in with different priorities or different visions of what they think, which was different than the last [leader].... Priorities would **change to meet the priority of the new individual.**”*
 - Stakeholder on a project with changing requirements
- *“It’s not like you could go and ask someone what did you mean when you wrote this? Those people were long gone by that time.”*
 - Stakeholder on a project cancelled after 10+ years of effort

Requirements are from people, not systems

Example: Assumptions

- *“They would deliver code down to us and I would look at it and just cry. It was so bad. It went on for months. It was horrible. It was just atrocious. They didn’t ask any questions. They just made all of these decisions. It was terrible.”*
 - Stakeholder on disappointing project
- *“The assumption that they understood each other when they really didn’t.”*
 - Stakeholder on a project that was cancelled after months of work

Validate assumptions ASAP as requirements or issues

Example: Diverse Perspectives

- *“It’s a cultural problem.”*
 - Offshore development for insurance system
- *“He had a real issue dealing with women.”*
 - Lack of respect inhibited collaboration

- *“I perceive that you think you are better than me.”*

- Group cultures: North / South



- Distinct roles
 - Military / Civilian / Contractor

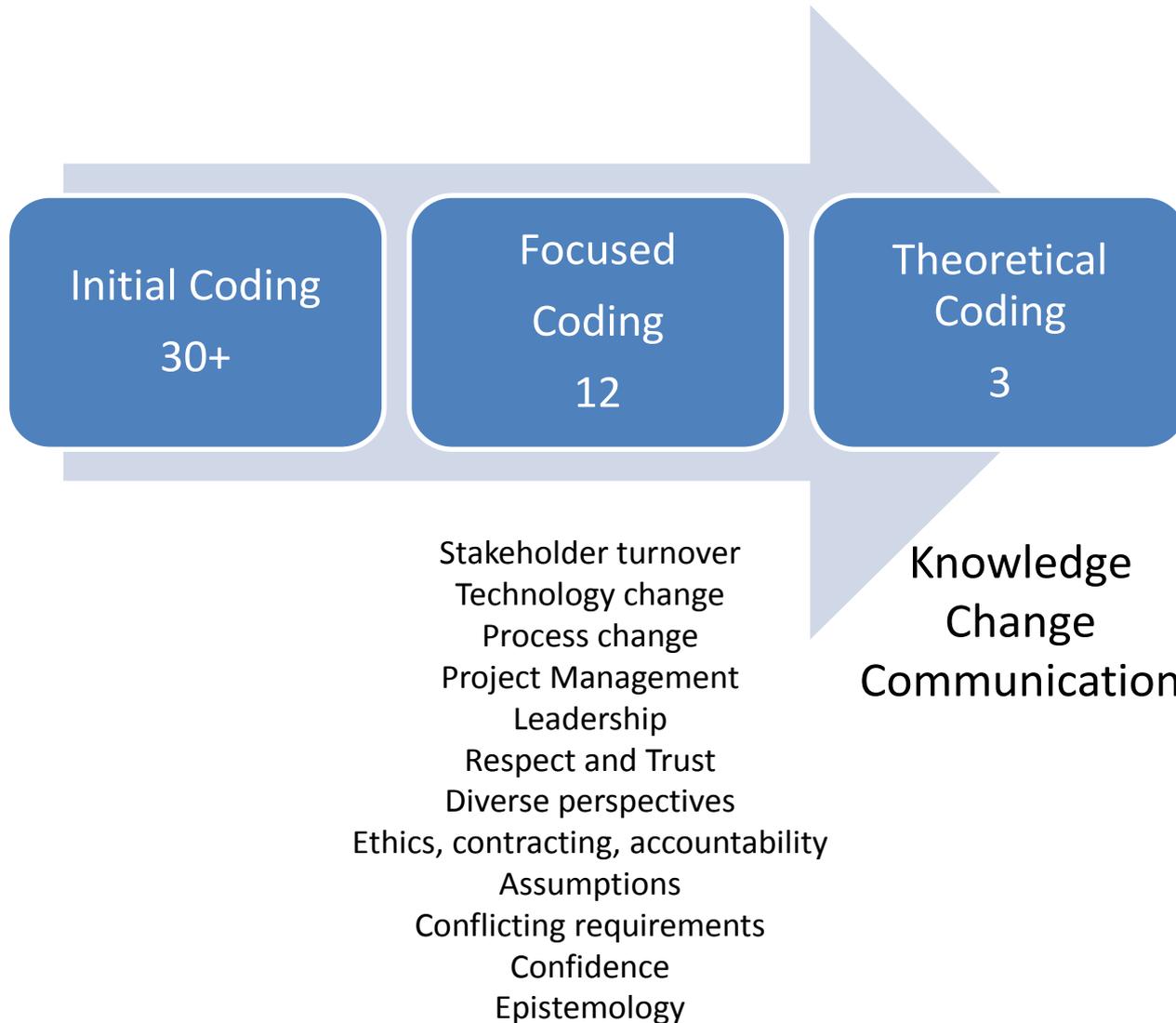
Different perspectives may hinder communication and understanding

Challenges Identified

- Turnover of personnel
- Time pressures
- Power issues
- Trust issues
- Diverse perspectives
- Perpetuating assumptions
- Resolving conflicting requirements
- Accountability issues
- Commitment issues
- Confidence issues
- Positivist approach

These challenges hinder common vision and understanding

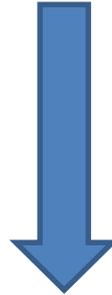
Grounded Theory Coding



Evolution of Society

- Lenski's Typology of Society

- Hunter/Gatherer
- Horticultural
- Agricultural
- Industrial



Reducing Risk

- Today

- Information Age
- Why are we still *gathering* requirements?



Constructing requirements means approaching requirements as knowledge to be proactively learned

Theories of Knowledge

Western Philosophy

- Ancient
 - 700 B.C – 400 A.D.
 - Plato, Aristotle, Socrates
- Medieval
 - 500 – 1700
- Modern
 - 1700 – Present

Positivism

- August Comte, 1830's-40's
- Stages of quest for truth
 - Theological (divine influence)
 - Metaphysical (reason, nature)
 - Positive (logic, scientific)
- Positivism
 - “The truth is out there”
 - Can be tested and proven

Gathering requirements is a positivist approach

Theories of Knowledge

Constructivism

- Piaget, 1960's
- Physical world exists separately from our understanding of it
- Knowledge is ***constructed*** by each individual based on previous experience

Social Constructionism

- Berger and Luckmann, 1966
- Knowledge is ***constructed*** by social interaction



Both are especially applicable to information systems requirements which are highly subjective

Emerging Theory

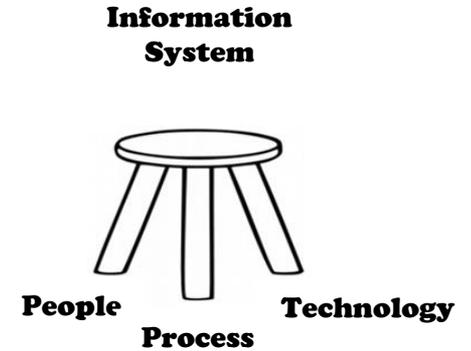
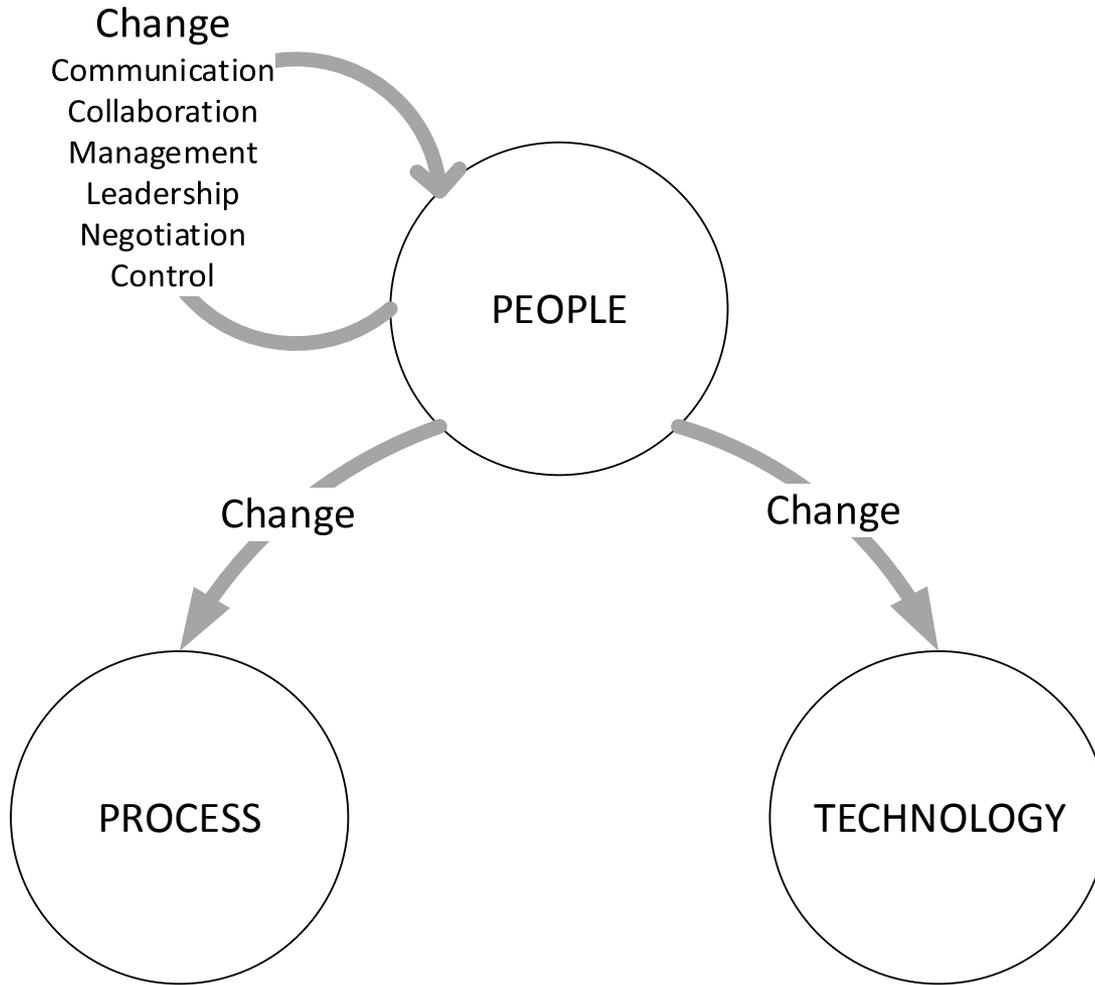


Information systems requirements elicitation commonly involves challenges of *knowledge*, *change* and *communication* which may be mitigated by approaches of social constructionism and constructivism

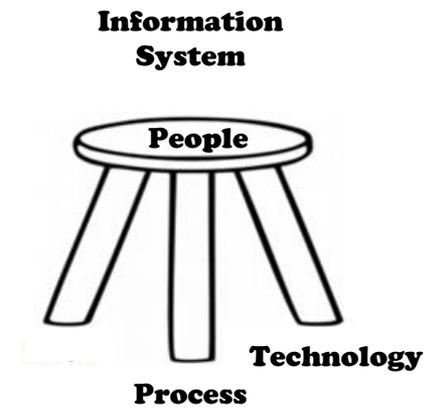
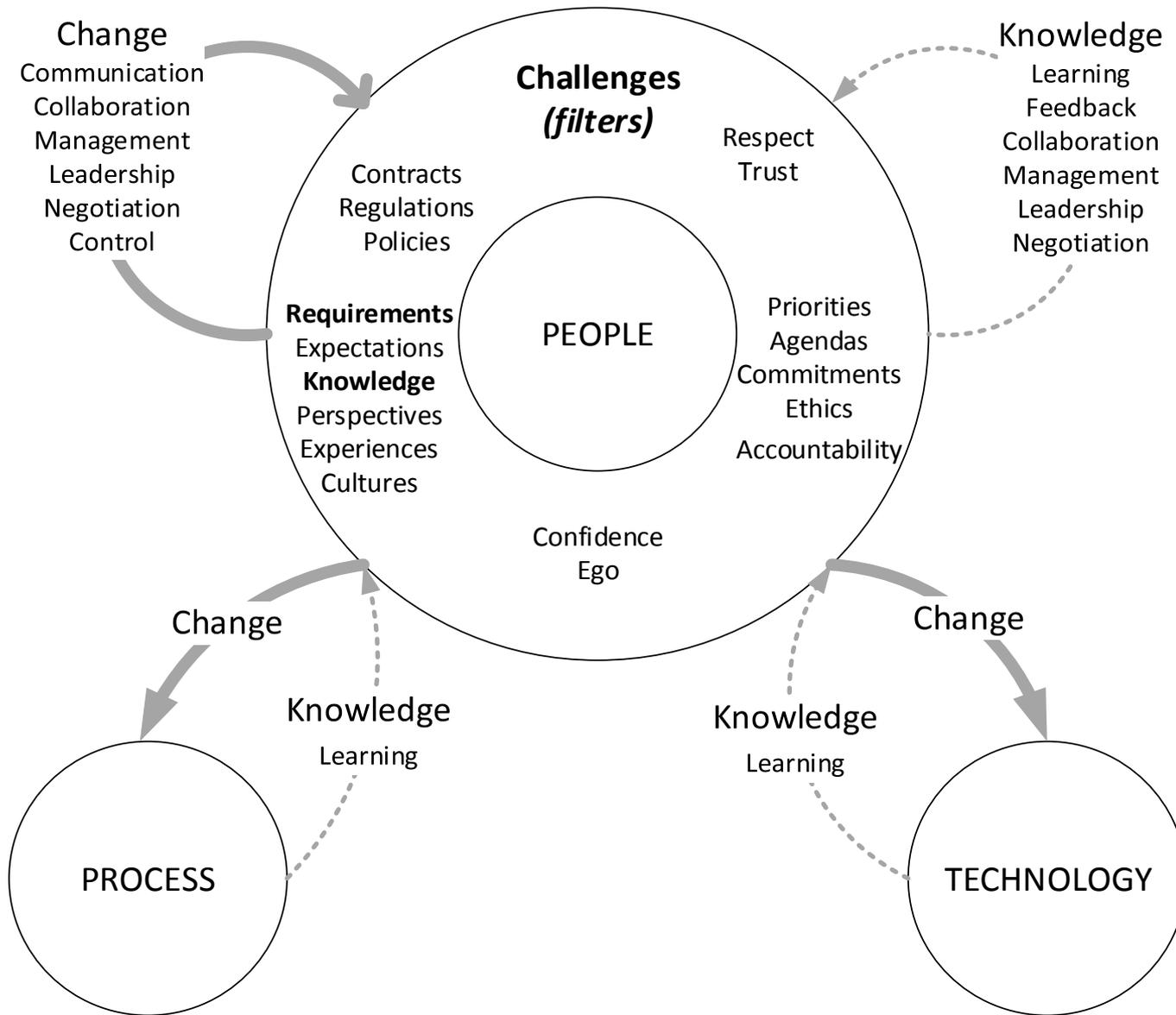
Supporting concepts

- Requirements must be *learned* by stakeholders and from stakeholders, not gathered
- An information system is successful only if it is *perceived* to be successful by the stakeholders
- Application of Knowles' principles of Adult Learning may help improve elicitation of information systems requirements

Traditional Model



Constructivist Model



Review



- Research question
 - What challenges are encountered by stakeholders when identifying requirements for information systems?
- Answer
 - Knowledge
 - Change
 - Communication

Challenges are more social and less technical

Requirements are Knowledge

- Requirements are a form of **knowledge**
- Knowledge must be **learned**, not gathered
- Knowles' adult learning principles apply:
 - Adults need a **reason** to learn
 - **Experience** is the foundation for new knowledge
 - Adults need to be **involved** and responsible
 - Immediate **relevance** encourages learning
 - Adults are **problem-centered** versus content-centered
 - **Internal motivation** is more effective than external



Approaching requirements as knowledge to be constructed
may help improve project success rates

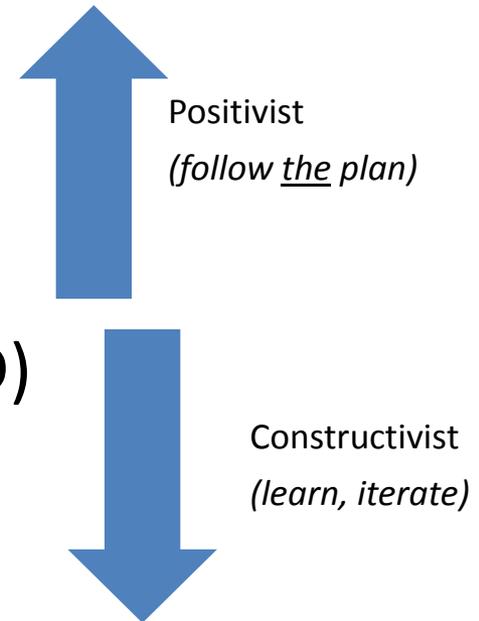
Implications



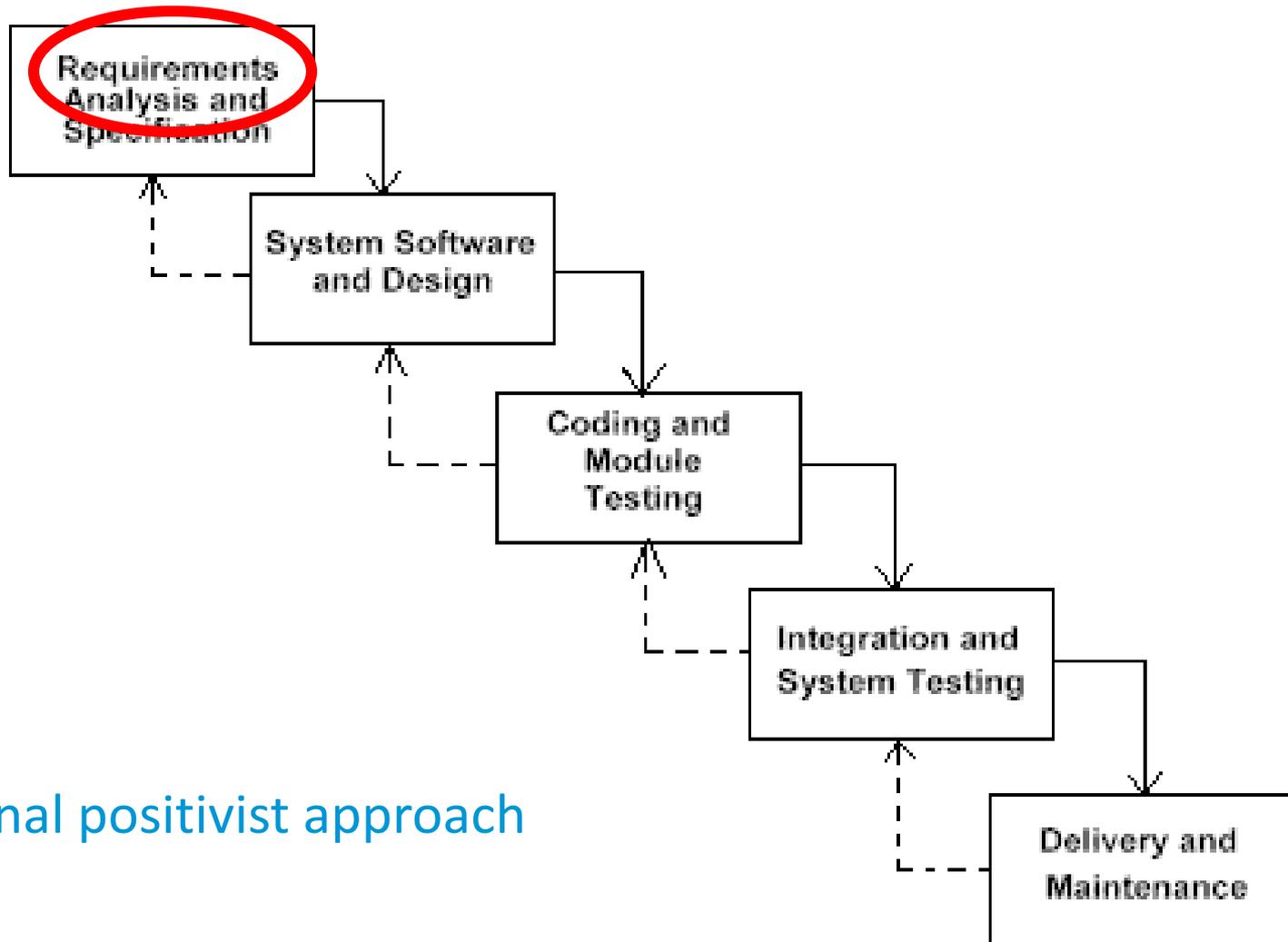
- Future research
 - Leverage challenges identified for quantitative study
 - Compare results to different populations
 - small business, academic, outside the U.S.
 - Explore organic vs. mechanistic factors
 - Explore impacts of personality types and temperaments
- Practice
 - *Computer* people need to be *people* people
 - Suggests why traditional elicitation methods fail
 - Suggests why emerging methods are perceived as more effective

System Development Lifecycle

- Models over time, with increasing iteration
 - Waterfall
 - Modified Waterfall
 - Spiral
 - Rapid Applications Development (RAD)
 - Rational Unified Process (RUP)
 - Agile

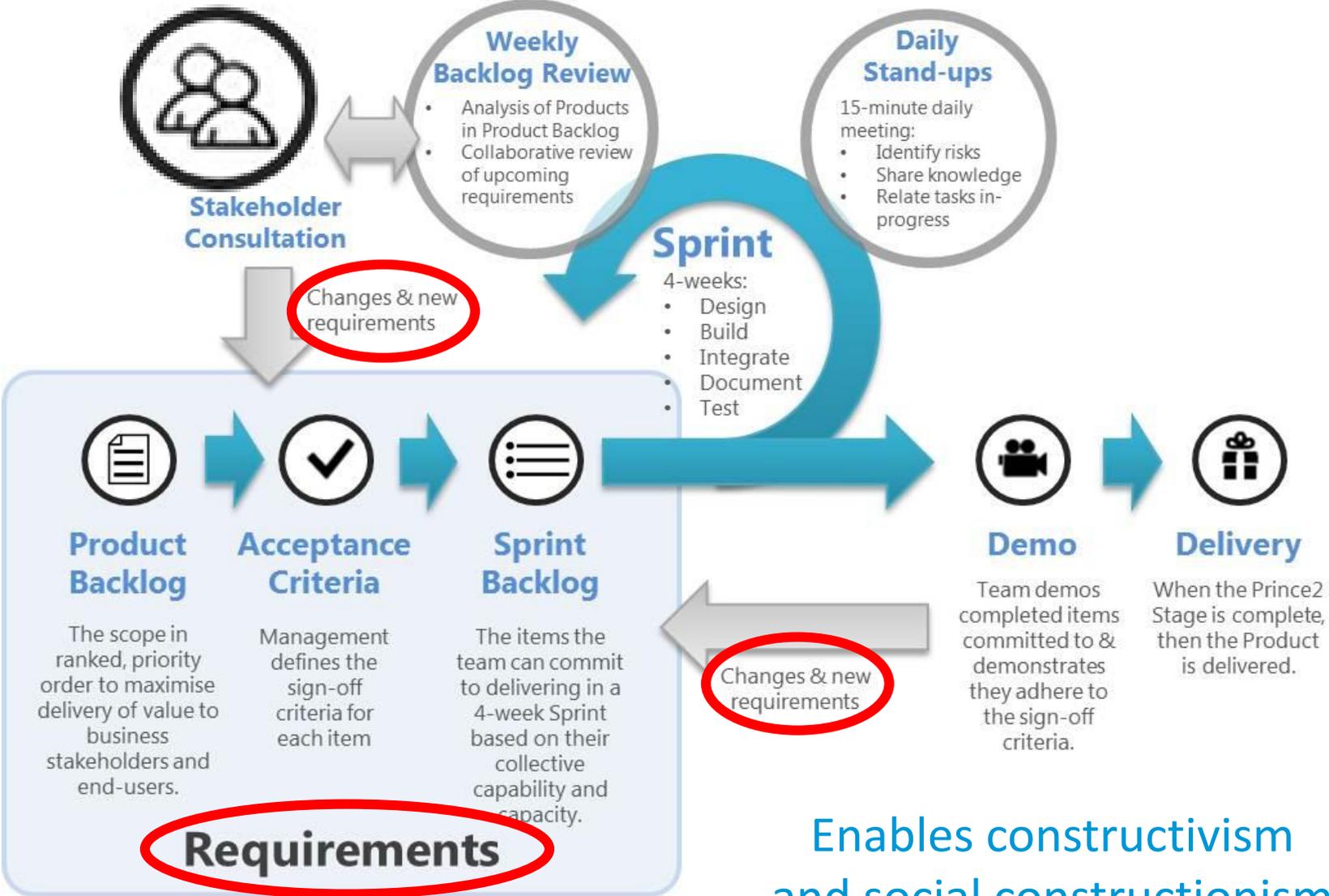


Waterfall Lifecycle



Traditional positivist approach

Agile Lifecycle



Recommendations

- Involve all stakeholder groups in collaboration
- Limit turnover, re-validate requirements
- Resolve conflicting requirements ASAP
- Resolve assumptions ASAP
- Empower all stakeholders to fully engage
- Recognize and address diverse perspectives
- Balance egos and personal agendas with perceived success of the project

Recommendations

- Approach requirements as a proactive, creative, learning activity, not a passive gathering activity
- Watch for and address obstacles to common vision and understanding
- Be proactive!
- Don't accept challenges as "normal"
 - "Normal" too often leads to failure
- Stop *gathering* requirements
- Advocate *constructing* requirements



Closing / Q&A



- Contact Information
 - David@DavidGibbs.com



<http://www.robkendall.co.uk/982/>

Backup Slides

If needed for Q&A

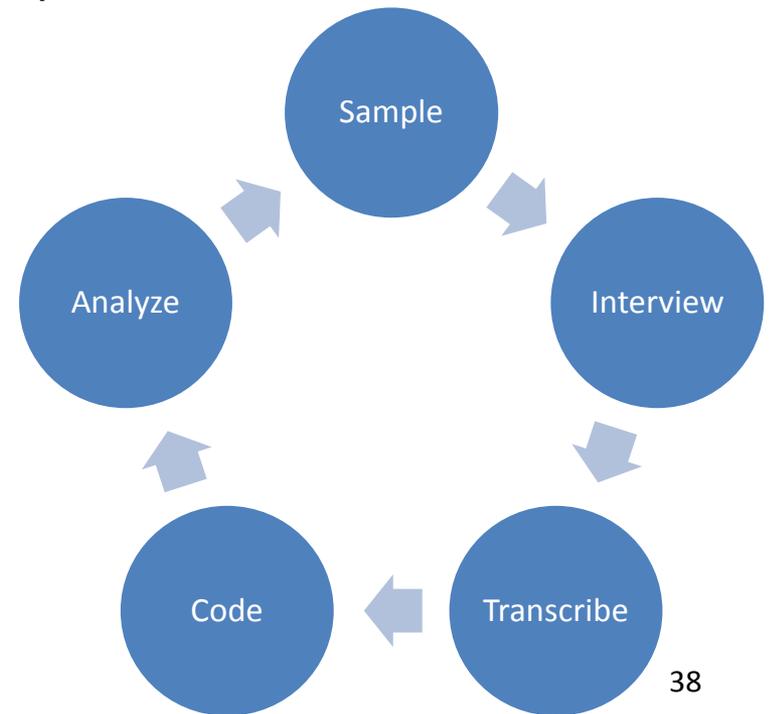
Methodology



- Qualitative study
 - Grounded Theory according to Charmaz (2006, 2014)
 - Semi-structured interviews with Critical Incident Technique
- Participants
 - Theoretical sampling
 - Practitioners known to me or referred as having relevant experience
 - 8 participants
 - Sampling stopped when sufficient data for emerging theory, saturation of concepts

Methodology Continued

- Data
 - 60-90 minute private interviews
 - Recorded and transcribed after each interview
 - 23 critical incidents collected and analyzed
 - Reinforced by literature sources and personal observations
- Analysis
 - Constant comparative method
 - Initial coding, line-by-line, *in vivo*
 - Focused coding, section-by-section
 - Theoretical coding
 - Memo writing using spreadsheets, abstracts, and presentations



Grounded Theory Steps

1. Identify problem and questions
2. Identify sensitizing concepts
3. Perform data collection and initial coding
4. Analyze codes forming tentative categories
5. Continue data collection with focused coding
6. Analyze codes and categories forming tentative concepts
7. Theoretical sampling to collect more data
8. Adopt categories as theoretical concepts
9. Reexamine earlier data, loop back to step 3
10. Create models of concepts
11. Write first drafts and loop back to step 5

Trustworthiness

- Acknowledge bias
- Participant cross-checking
- Theoretical saturation
- Theoretical validity
- Participation rate
- Audio recording, transcribed promptly by me
- Independent extraction



– Butterfield et al., 2005

Limitations

- Participants limited to information systems experience focused on
 - Healthcare
 - Insurance
 - Government
- Participants from the geographic regions of
 - Austin, TX
 - San Antonio, TX
 - Washington, DC
- Broad definitions for requirements and stakeholders
 - Future research could focus on specific types of each
- Interpretations are limited by my own perspective, experience, and bias
- Time, deadlines



“A man’s got to know his limitations”
– Dirty Harry

The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

References

- Berger, P. & Luckmann, T. (1967). *The social construction of reality: A treatise in the sociology of knowledge*. New York: Anchor.
- Brooks, F. P. (1987). No silver bullet: Essence and accidents of software engineering. *Computer*, 20(4), 10-19.
- Butterfield, L. D., Borgen, W. A., Amundson, N. E., & Maglio, A. T. (2005). Fifty years of the critical incident technique: 1954-2004 and beyond. *Qualitative Research*, 5(4), 475-497.
- Davey, B., & Cope, C. (2008). *Requirements Elicitation - What's Missing? Issues in Informing Science and Information Technology*, 5, 543-551. Retrieved from <http://proceedings.informingscience.org/InSITE2008/IISITv5p543-551Davey466.pdf>.
- Davis, C. J., Fuller, R. M., Tremblay, M. C., & Berndt, D. J. (2006). Communication challenges in requirements elicitation and the use of the repertory grid technique. *Journal of Computer Information Systems*, 47, 78-86.

References continued

- Knowles, M. (1980). *The modern practice of adult education: From pedagogy to andragogy*. Wilton, CT: Association Press.
- Laplante, P. A. (2014). *Requirements engineering for software and systems*. CRC Press.
- Leffingwell, D. (2011). *Agile software requirements: Lean requirements practices for teams, programs, and the enterprise*. Boston: Pearson.
- Lenski, G., Nolan, P., & Lenski, J. (1995). *Human societies: An introduction to Macrosociology*, (7th ed.). New York: McGraw-Hill.
- Leviss, J. (2010). *H.I.T. or miss: Lessons learned from health information technology implementations*. Chicago: American Health Information Management Association.
- Piaget, J. (2001). *The psychology of intelligence*. New York: Routledge.

References continued

- Reinertsen, D. (2011). Forward. Leffingwell, D. (2011). *Agile software requirements: Lean requirements practices for teams, programs, and the enterprise*. Boston: Pearson.
- Thomas, D., & Hunt, A. (2004). Nurturing requirements. *IEEE Software*, 21(2), 13-15.