Theories in software engineering

EMSE-UVic Fall 2020
Margaret-Anne Storey

[Building Theories in Software Engineering](#) by Dag I. K. Sjøberg, Tore Dybå, Bente C. D. Anda, Jo E. Hannay, Guide to Advanced Empirical Software Engineering pp 312-336, Spring 2008

When we think about theories, what does "usefulness" or "utility" mean?

What does a theory do?

Research
- Theory facilitates communication of ideas and knowledge
- Theory helps develop and consolidate common research agendas

Theory supports

supports

Industry
- Theory gives input to decision-making regarding choice of technology and resource management
- An adapted theory helps understanding and prediction in a given setting

Fig. 1 Usefulness of theory for research and industry

https://link.springer.com/chapter/10.1007/978-1-84800-044-5_12

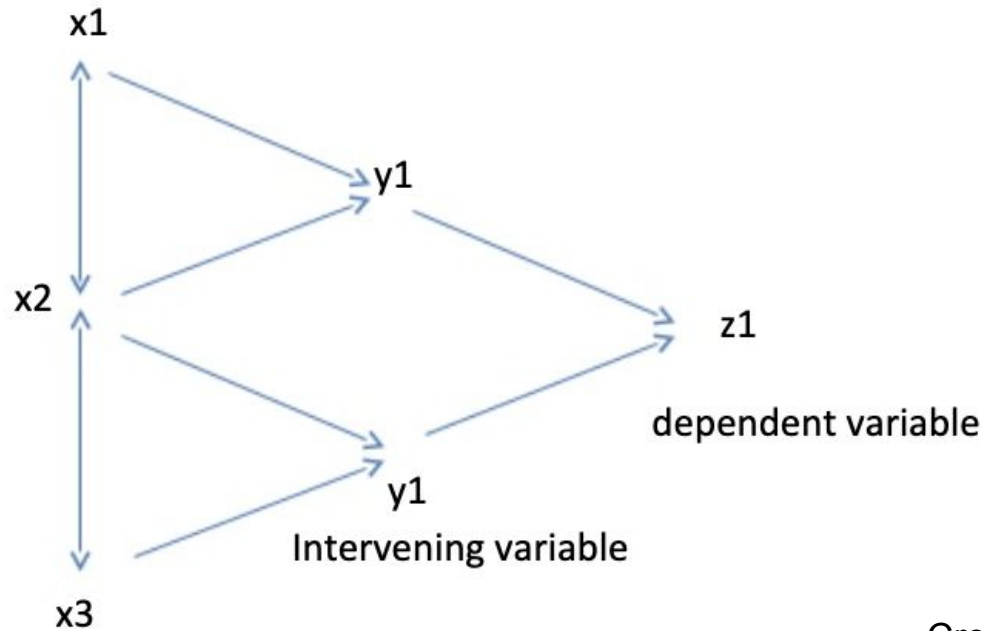# How to generate theories?

Inductive methods (e.g., grounded theory)

Deductive methods (derive testable hypotheses from a theory and check them for empirical support

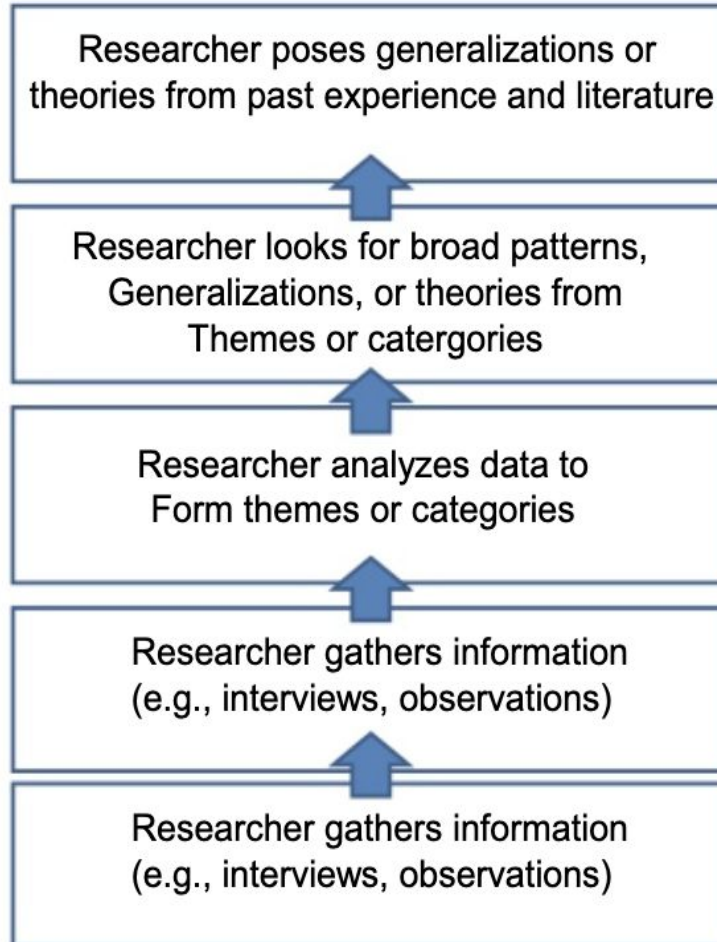Abductive methods (a leap from initial observations to an explanatory theory)

Deductive approach

Researcher tests or verifies a theory

Researcher tests hypothesis
Or research questions form the theory

Researcher defines and operationalizes
variables derived from the theory

Researcher measures or observes variables
using an instrument to obtain scores

11
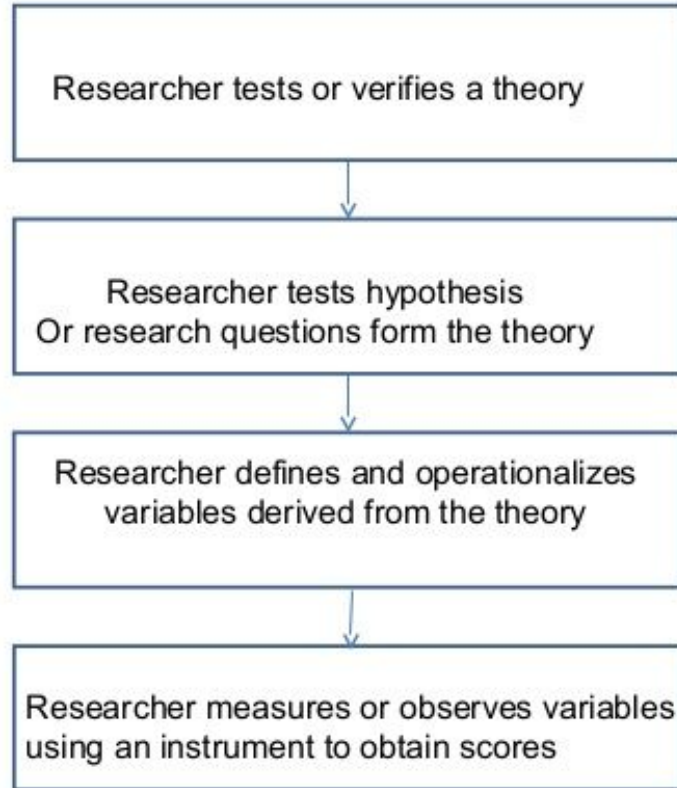
Creswell & Creswell

Three independent variables influence a single dependent variable mediated by two intervening variables.

x1

x2

x3
Independent variable

y1

y1
Intervening variable

z1

dependent variable

Creswell & Creswell

Inductive approach

Researcher poses generalizations or theories from past experience and literature

⬆

Researcher looks for broad patterns, Generalizations, or theories from Themes or catergories

⬆

Researcher analyzes data to Form themes or categories

⬆

Researcher gathers information (e.g., interviews, observations)

⬆

Researcher gathers information (e.g., interviews, observations)

Creswell & Creswell

The role of theory in your studies….

In a **quantitative study**, the **theory** is used as **a lens to guide which variables** should be measured or isolated

But in a **qualitative study**, the **theory** is used to help **label and categorize** (code) the data

But a theory may not be available at the outset, may be an **emerging theory**

Theories also play a role in connecting research to the relevant **literature**

How are theories formed?

May **borrow** theories **as-is** from other disciplines

May **adapt** theories to SE from other disciplines

May be **generated** from scratch (or evolve) in SE

Building Theories in Software Engineering by Dag I. K. Sjøberg, Tore Dybå, Bente C. D. Anda, Jo E. Hannay, Guide to Advanced Empirical Software Engineering pp 312-336, Spring 2008

Types of theories

I.   **Analysis:** descriptions/conceptualizations of "what is" (taxonomy, classification, lack of explanation/prediction)
II. **Explanation:** Answers to a question of why something is or happens (notions of causality/asymmetry)
III. **Prediction:** Predicting what will happen (without necessarily explaining why)
IV. **Explanation and Prediction:** combine traits of II and III
V.   **Design and Action:** "how to do things", prescriptive (think Design Science)

Building Theories in Software Engineering by Dag I. K. Sjøberg, Tore Dybå, Bente C. D. Anda, Jo E. Hannay, Guide to Advanced Empirical Software Engineering pp 312-336, Spring 2008

Elements of a theory

What: constructs of the theory which are the entities in terms of which a theory offers description, explanation, prediction or prescription

How: relationships between constructs, make up a theory's propositions, how constructs interact

Why do the relationships hold (Type II and IV theories)

Where, when and for whom: scope conditions

Complexity of theories….

Level 1: minor working relationships that are concrete and based directly on observations

Level 2: involve some abstraction but are still closely linked to observations

Level 3: Overarching theories that seek to explain software engineering behaviour

https://link.springer.com/chapter/10.1007/978-1-84800-044-5_12

How to evaluate theories?

Table 1 Criteria for evaluating theories

Testability	The degree to which a theory is constructed such that empirical refutation is possible
Empirical support	The degree to which a theory is supported by empirical studies that confirm its validity
Explanatory power	The degree to which a theory accounts for and predicts all known observations within its scope, is simple in that it has few ad hoc assumption, and relates to that which is already well understood
Parsimony	The degree to which a theory is economically constructed with a minimum of concepts and propositions
Generality	The breadth of the scope of a theory and the degree to which the theory is independent of specific settings
Utility	The degree to which a theory supports the relevant areas of the software industry

https://link.springer.com/chapter/10.1007/978-1-84800-044-5_12

Framework for describing SE theories

Table 2 Framework for SE theories

Archetype class	Subclasses
Actor	Individual, team, project, organisation or industry
Technology	Process model, method, technique, tool or language
Activity	Plan, create, modify or analyze (a software system); see Sjøberg et al. (2005)
Software system	Software systems may be classified along many dimensions, such as size, complexity, application domain, business/scientific/student project or administrative/embedded/real time, etc.

https://link.springer.com/chapter/10.1007/978-1-84800-044-5_12

When we refer to software systems (see Table 2), what do the systems include/exclude?
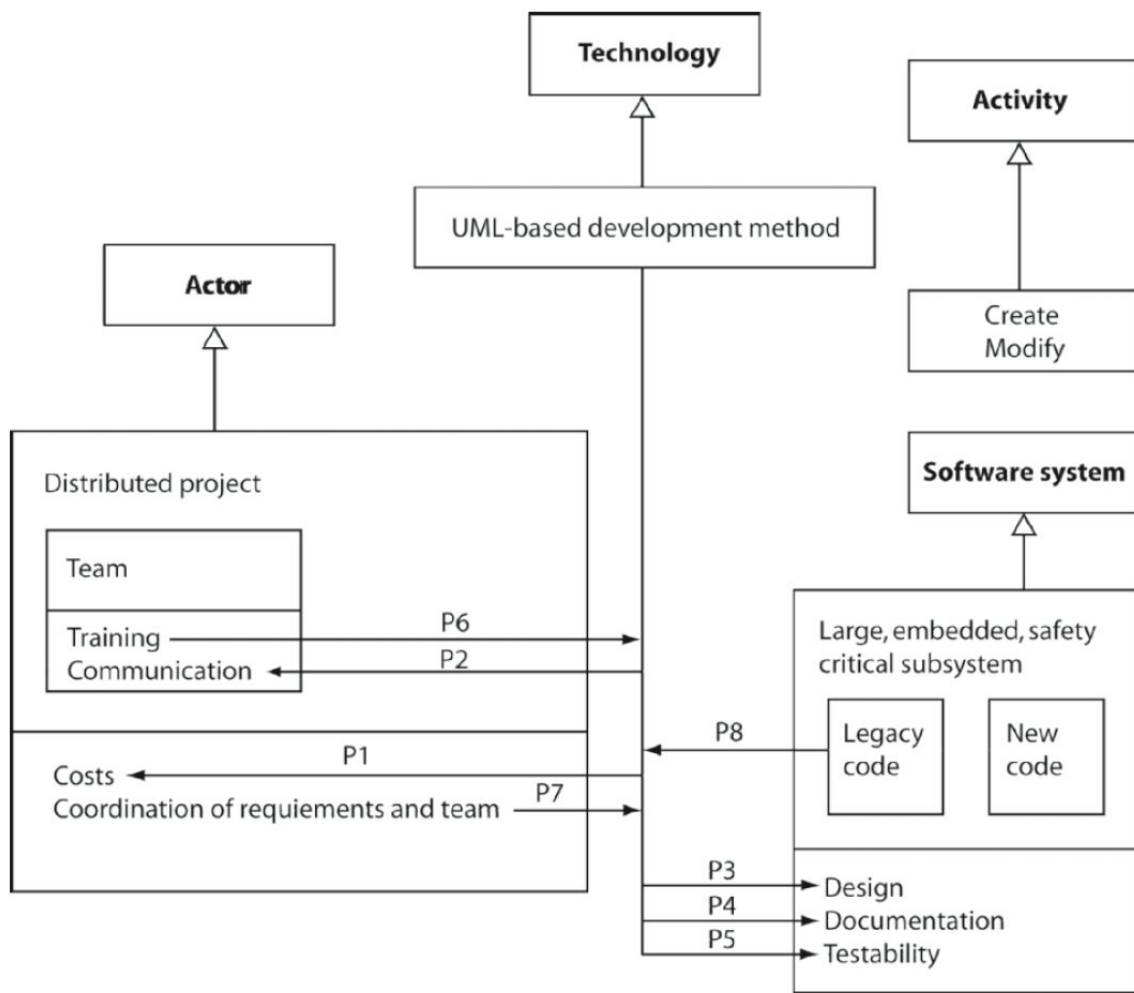
Fig. 3 A theory for the effect of UML-based development

Example of a theory...

What is your take on Figure 3?
What does it tell us?
How readable is it?
Can we make predictions using it?
What is missing?
How might we put it to the test?

More examples, see this paper!

https://alannaholeson.com/papers/EUDbook2-2ndEd-theory+EUSE-distrib.pdf

Steps in building SE theories

1. **Define the constructs** of the theory
2. **Define the propositions** of the theory
3. **Provide explanations** to justify the theory
4. **Determine the scope** of the theory
5. **Test the theory** through empirical research (choose an appropriate research setting/sample, operationalize constructs into empirical variables, operationalize theoretical propositions into empirically testable hypotheses

https://link.springer.com/chapter/10.1007/978-1-84800-044-5_12

Do you agree that "place" is not interesting per se in SE (see p322)?

Theory building…

Joe's theory – describes how UML diagrams are a stylized form of external memory used in a collaborative group

- His theory says what they are used for (meetings, shared understanding..)
- His theory must define meaning of the terms such as "diagram", "discussion"
- Should explain why the diagrams are used in some settings and not in others
- Why some things are included in the diagrams, and other things are not
- His theory should be predictive of how a team may use UML based on certain factors
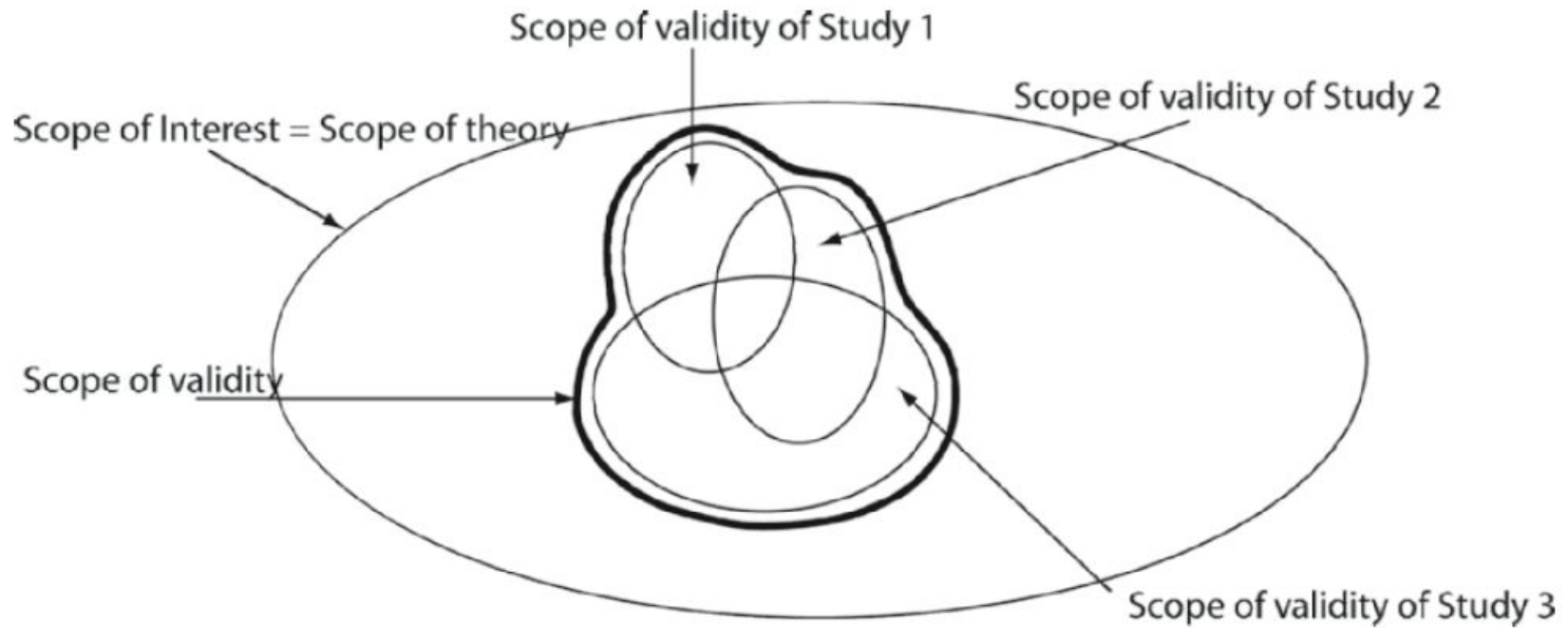
Fig. 4 Scope of interest versus scope of validity

Figure 4: What if the scope of validity across studies was only the intersection of the Venn diagram, rather than the union?

Activity:
Research Questions -> Methods -> Theories

1. Why do engineers ignore security warnings in their code?
2. Does test driven development improve code quality?
3. Which code review tool design is easier to use for new team members?
4. Do the nature of discussions in online technical forums deter the involvement of women?

Research question and whether it is clear	Method(s): why use them and their limitations	Theory type (descriptive, predictive, etc.)