Chapter 12
Building Theories in Software Engineering

Dag I. K. Sjøberg, Tore Dybå, Bente C. D. Anda, Jo E. Hannay

Abstract In mature sciences, building theories is the principal method of acquiring and accumulating knowledge that may be used in a wide range of settings. In software engineering, there is relatively little focus on theories. In particular, there is little use and development of empirically-based theories. We propose, and illustrate with examples, an initial framework for describing software engineering theories, and give advice on how to start proposing, testing, modifying and using theories to support both research and practise in software engineering.

1. Introduction

> When should theorizing begin? "Theorizing should begin as soon as possible" What is the bulk of data necessary to begin theorizing? When is it neither too early nor too late to begin? Nobody can tell. It all depends on the novelty of the field and on the existence of theoretically-bent scientists prepared to take the risk of advancing theories that may not account for the data or that may succumb at the first onslaught from fresh information gathered in order to test the theories: this takes moral courage, particularly in an era dominated by the criterion of success, which is best secured by not attacking big problems. Two things, though, seem certain: namely, that premature theorizing is likely to be wrong – but not sterile – and that a long deferred beginning of theorizing is worse than any number of failures, because (1) it encourages the blind accumulation of information that may turn out to be mostly useless, and (2) a large bulk of information may render the beginning of theorizing next to impossible. (Bunge, 1967, p. 384).

In mature sciences, building theories is the way to gain and cumulate general knowledge. Some effort has been made to propose and test theories based on empirical evidence in software engineering (SE) (Hannay et al., 2007), but the use and building of empirically-based theories[1] in SE is still in its infancy.

---

[1] In this chapter, we focus on empirically-based theories; that is, theories that are built or modified on the basis of empirical research. Hence, in the reminder of this chapter, we use "theory" as short for "empirically-based theory" unless otherwise explicitly stated.

There are many arguments in favour of using theories. They offer common conceptual frameworks that allow the organization and structuring of facts and knowledge in a concise and precise manner, thus facilitating the communication of ideas and knowledge. Theory is the means through which one may generalize *analytically* (Shadish et al., 2002; Yin, 2003), thus enabling generalization from situations in which statistical generalization is not desirable or possible, such as from case studies (Yin, 2003), across populations (Lucas, 2003), and indeed, from experiments in the social and behavioural sciences (Shadish et al., 2002), with which experiments in empirical SE often share essential features.

Our position is that theories should be useful; we are not interested in theories purely as an academic exercise. As such, we adhere to the view of the philosophical school of pragmatism, "both specific beliefs and methods of inquiry in general should be judged primarily by their consequences, by their usefulness in achieving human goals" (Godfrey-Smith, 2001). Since SE is an applied discipline, SE theories should, at least ultimately, be useful to the software industry. Since each SE setting is unique, the theories would need local adaptations to be directly useful in concrete cases. Figure 1 illustrates that both research communities and industry may benefit from using SE theories.

Arguments in favour of theory have been voiced in the SE community by other researchers as well (Basili, 1996; Endres and Rombach, 2003; Herbsleb and Mockus, 2003; Kitchenham et al., 2002; Land et al., 2003; Sauer et al., 2000; Tichy, 1998; Jørgensen and Sjøberg, 2004). However, there has been little focus on what the nature of SE theories should be like, and how they should be described and built. Hence, in this chapter, we suggest that the description of a theory should be divided into four parts: the *constructs* (*what* are the basic elements), *propositions* (*how* do the constructs interact), *explanations* (*why* are the propositions as specified) and *scope* (what is the universe of discourse in which the theory is applicable). Moreover, we propose a diagrammatic notation for
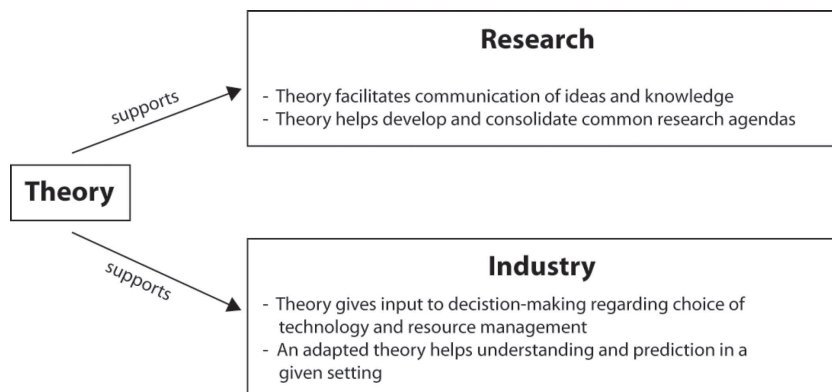

Fig. 1 Usefulness of theory for research and industry

describing the constructs, relationships and scope of a SE theory. In particular, each construct should belong to, or be derived from, one of the four archetype classes *Actor*, *Technology*, *Activity* and *Software System*. We believe that this structure for describing SE theories will support both researchers who propose theories and potential users of such theories.

The remainder of this chapter is organized as follows. Section 2 discusses categories of theories, elements of a theory and how theories may be formed and evaluated. Section 3 presents the framework for describing SE theories. Section 4 illustrates steps in theory building. Section 5 evaluates the example theory according to the criteria given in Sect. 2. Section 6 summarizes and describes topics for future work.

2. What Theories Are

The question of what constitutes a theory is a source of continuing discussion. Answers to this question depend on philosophical issues, practical issues, and not least, the field of study – indeed, the purpose of this chapter is to outline suggestions as to what theories for SE should be like.

There is no universally agreed upon definition of the concept of an empirically-based theory, nor is there any uniform terminology for describing theories. What is agreed is that it is difficult to provide necessary and sufficient conditions that delineate the concept of theory. Nevertheless, it is still possible to get a grasp on what a theory is. In sciences that are relevant to empirical SE, such as information systems, management, and social and behavioral sciences, discussions concerning theory tend to revolve around the following issues: (1) what a theory does, (2) what the elements of a theory are, (3) how theories are formed, and (4) how theories are evaluated. In the following, we summarize some of the answers to these questions.

2.1. What a Theory Does

The focus of this chapter is on theories that relate to observable phenomena, and that are built and modified based on empirical research. According to several accounts, this implies that a theory should offer explanations of why certain phenomena occur in the sense of predicting them. Moreover, the predictions should be testable, so as to render the theory refutable.

This familiar description of what a theory should do is hypothetico-deductive in nature, and would seem particularly suitable for empirical research. However, there are also other relevant modes of empirically-based theory. In the discipline of information systems, Gregor (2006) has classified theories into five types according to what they do.

I. *Analysis*. Theories of this type include descriptions and conceptualizations of "what is." Also included are taxonomies, classifications and ontologies in the sense of Gruber (1993). The lack of explicit explanation and prediction disqualifies this category as theory for many scholars (Bacharach, 1989; Sutton and Staw, 1995; Nagel, 1979).

II. *Explanation*. Theories of this type explicitly explain. What constitutes an explanation is a nontrivial issue. However, a common view is that an explanation answers to a question of *why* something is – or happens (rather than *what* happens) (Van Fraassen, 1980; Sandborg, 1998). Current views insist that explanations include notions of causality and asymmetry (if *A* explains *B*, then *B* should not also be a viable explanation of *A*) (Salmon, 1989).

III. *Prediction*. These theories are geared towards predicting what will happen, without explaining why. Examples are mathematical and probabilistic models of social and natural sciences.

IV. *Explanation and prediction*. Theories of this type combine the traits of II and III, and correspond to what many consider a "standard" conception of empirically-based theories.

V. *Design and action*. These theories describe "how to do" things, that is, they are prescriptive. Design science (Simon, 1996; Hevner et al., 2004; Hevner and March, 2003; March and Smith, 1995) is influential here. Although there is usually an implicit prediction that following the design principles will be beneficial, it is a matter of opinion as to whether this category describes theories (March and Smith, 1995).

These five types illustrate some of the diversity of what may be considered as theories. Our focus is very much on theories that explain phenomena. Thus, Types II and IV are those of primary interest. However, in practice, the explanatory *function* of a theory depends also on how the theory interacts with other theories and the current level of knowledge. For example, many view physical theories as belonging to Type III: Hawking states "that a physical theory is just a mathematical model and that it is meaningless to ask whether it corresponds to reality. All that one can ask is that its predictions should be in agreement with observation" (Hawking and Penrose, 1996, pp. 3–4), a sentiment also expressed by Feynman (1985). However, although they "merely" describe and predict what happens on the quantum level, these theories can thereby also be said to explain phenomena on the macro level (for example, why light refracts off oil films). Also, theories of Type I, that merely describe, may well provide explanations for other theories or phenomena. For example, the text comprehension model of Van Dijk and Kintsch (1983) describes how mental models of increasing complexity form during text comprehension. There are no explicit explanations or predictions, but in conjunction to program comprehension, the model provides an explanation as to why experts and novices follow different strategies when understanding code (Burkhardt et al., 2002). Generally, what constitutes an explanation is very much a pragmatic question.

2.2. What the Elements of Theory are

It seems to be broadly accepted that *constructs* and *relationships* between constructs constitute the basic building blocks of theories, and that it is important to delineate a theory's area of application by specifying *scope conditions*. Inspired by Dubin (1978), Whetten (1989) describes these elements as building blocks of theory in the following manner.

– *What* are the entities in terms of which a theory offers description, explanation, prediction or prescription? These are the constructs of a theory. Examples are "quarks" (quantum physics), "group process" (social science), "cognitive load" (cognitive psychology) and "programming skill" (SE). According to some epistemological positions (e.g., logical positivism), constructs must represent directly observable entities; while others (scientific realism) allow representations of hitherto unobserved entities ("gravity," "quarks," "feelings") that are postulated to exist; while still others (anti-realism, instrumentalism, pragmatism) see constructs only as useful instruments to provide descriptions, explanations, etc. In SE, the constructs would typically relate to people, organization, technology, activities and software system.
– *How* are the constructs related? Relationships between constructs make up a theory's propositions, and describe how the constructs interact. Constructs and their relationships are the basic constituents of all five types of theory above. Describing how things are related may give rise to predictions (Type III and Type IV theories).
– *Why* do the relationships hold? Answers to this question are what give the theory explanatory power (Type II and Type IV theories). Parts of this may already be provided in the propositions established above. Explanatory power may also arise from a theory's interaction in a research context.
– *Where, When,* and for *Whom* does the theory apply? *Scope conditions* are statements that define the circumstances in which the theory's propositions are supposed to be applicable (Cohen, 1989).

2.3. How Theories are Formed

The ways in which theories are built, and from what, say much about what theories are. Theories in SE may enter the stage in three ways to explain SE phenomena:

1. Theories from other disciplines may be used as they are.
2. Theories from other disciplines may be adapted to SE before use.
3. Theories may be generated from scratch in SE.

Modes (1) and (2) reflect that SE is a multidisciplinary discipline. Examples of the first mode are the use of theories from cognitive psychology to explain phenomena in program comprehension (Burkhardt et al., 2002; Abdel-Hamid et al., 1993;

Ramanujan et al., 2000), and theories from social and behavioural sciences to explain group interaction in requirements negotiation and inspection meetings (Land et al., 2003). Examples of the second mode can be found in (Sauer et al., 2000; Land et al., 2003; Herbsleb and Mockus, 2003), while the case described in Sects. 3–5 is an example of the third mode.

This chapter focuses on the concept of "SE theory," that is, theories with constructs and relationships defined from SE entities (Sect. 3). A SE theory thus arises through modes (2) and (3). The latter mode, generating theories from scratch, raises certain methodological issues as to how to build theories, and as a result, what theories are. In the following, we summarize some of these issues.

Referencing (Merton, 1968; Yin, 1984), Carroll and Swatman (2000) give three levels of sophistication or complexity of theories (for information systems):

Level 1. Minor working relationships that are concrete and based directly on observations
Level 2. Theories of the middle range that involve some abstraction but are still closely linked to observations
Level 3. All-embracing theories that seek to explain social behaviour. ("Social behavior" in (Carroll and Swatman, 2000) is here replaced with "SE.")

These levels set milestones in theory generation, but they may also represent full theories, depending on the rationale of the generation process one adheres to and the purpose of one's theory (Sect. 2.1). The development of SE theories from scratch (3) is in early stages, and immediate efforts will probably focus primarily on Levels 1 and 2. The case presented later produces a theory on Level 1.

The formation of theories is a process of continuous refinement and development involving inferences both from practise to theory as well as from theory to practise. Essential elements of this process are conceptual development, operationalization, confirmation or disconfirmation, and application, see Fig. 2.

Inductive methods sample singular observations in an enumerative fashion, in order to generate laws (covering laws) and empirical generalizations ("grounded theory" according to Glaser and Strauss (1967)). The inductive approach admits Levels 1 and 2 as *de facto* theories.

Other approaches view Levels 1 and 2 merely as intermediary steps towards, respectively, Levels 2 and 3. For example, the *abductive* approach to theory generation (Peirce, 1958; Haig, 2005) uses induction only as a first step to define phenomena (relatively stable, recurrent, general features) from observations, and then goes on to generate explanatory theories that explain these phenomena. Abductive inference (Peirce, 1958) introduces a creative aspect to theory generation, in that it transcends observation and is no longer strictly bound by facts (data). Instead, explanations rely on semantic models, i.e., simplified approximations of reality or useful conceptualizations (Franck, 2002; Rosenberg, 2001; Ruse, 1995). Examples are the ideal gas model and the rational choice model in economics that continue to be useful for educational purposes, even though empirical evidence disconfirms the literal interpretation of these models; and various models of the human brain as an information processing unit for explaining human cognition. This independence of
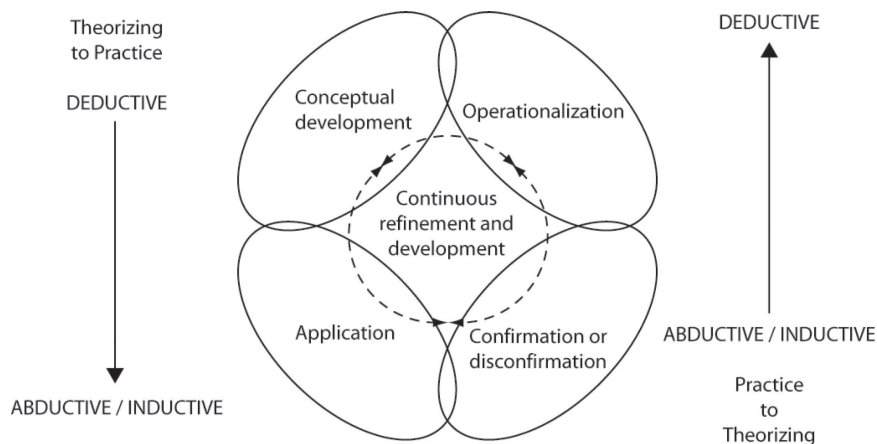
Fig. 2 Theory development consists of inductive and abductive aspects and deductive aspects, and may be initiated from both the practical or from the theoretical realm. Central to forming theory is conceptual development, that is, the conception of pertinent constructs and relationships through inductive and abductive processes. In order for the theory to be confirmed or disconfirmed in a deductive process, the conceptual elements must be operationalized into observable entities and measurable units on the one hand; and on the other hand, they must be applicable in real situations in practical disciplines. (The figure is adapted from (Lynham 2002)).

direct correspondences with reality is favored by aspects in the epistemological directions of anti-realism, instrumentalism and pragmatism. Such models typically constitute Type II and Type IV theories on Level 3. Methods such as induction and abduction are essentials in the conceptual development of theories built from scratch, see Fig. 2.

Deductive methods derive testable hypotheses from a theory and check these for empirical support.

2.4. How Theories are Evaluated

The evaluation of theories involves both logical and empirical standards (Cohen, 1989). However, in order to be able to evaluate the goodness of a theory, we must first establish the criteria by which it is to be evaluated. Several such criteria are described in the literature (Bunge, 1967; Cohen, 1989; Dubin, 1978). Which criteria one adheres to depends on the type of theory one is attempting to generate, as well as on the framework of generation one is adhering to. For the purpose of evaluating empirically-based theories in SE, we believe that the criteria shown in Table 1 are most relevant.

The *hypothetico-deductive* framework sees the criterion of *falsifiability* (Popper, 1959), as the demarcation criterion between science and non-science. It assumes

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

the presence of a falsifiable theory, which gives rise to hypotheses that are tested by observation. Although this framework as such has been overtaken by other frameworks (Ruse, 1995), the principle of *testability* remains fundamental for empirically-based theories. There are no commonly agreed set of criteria for evaluating testability, but we will emphasize the criteria as follows: (1) The constructs and propositions of a theory should be clear and precise such that they are understandable, internally consistent and free from ambiguities. (2) It must be possible to deduce hypotheses from the theory's propositions, so that the theory may be confirmed or disconfirmed. (3) The theory's scope conditions must be explicitly and clearly specified, so that the domain or situations in which the theory should be (dis-)confirmed and applied is clear.

Note that in social and behavioral sciences, with which empirical SE shares many methodological issues, deeming a theory as false based on its predictions, is rarely feasible (Lindblom, 1987; Weick, 1989). If a prediction is not supported by empirical evidence, alternative theories or refinements of existing theories are sought, rather than theory rejection; or a new phenomenon is defined, which in turn starts the theory generation process for *that* phenomenon. Moreover, several theories may provide descriptions, explanations, etc. for a given phenomenon; all of which may be empirically adequate in the sense of not having been disconfirmed (Rosenberg, 2001; Haig, 2005). One must therefore have criteria that give inferences to *best* descriptions, explanations, predictions, etc. Therefore, in addition to testability, other theory appraisal criteria are equally important.

Related to testability is the degree to which a theory is supported by empirical evidence. Such evidence is also important in choosing among alternative descriptions, explanations, predictions, etc. *Empirical support* requires that the theory is tested in empirical research. Pursuing empirical evidence has the added advantage of treating both confirming and disconfirming evidence as informative. Furthermore, pursuing such evidence clearly points in the direction of designing a series of studies that complement one another (Basili et al., 1999).

Explanatory power can be viewed as a theory's ability to provide explanations of why something happens. Two criteria are (Thagard, 1992): (1) *Analogy*, that is, the degree to which a theory is supported by analogy to well-established theories. Explanatory power is seen as increased if a theory's constructs and relationships are formulated in terms of what is familiar and understood. (2) *Explanatory breadth*, that is, the degree to which a theory accounts for and predicts all known observations within its scope. Some explanations apply to particular events, while others apply to general phenomena or regularities. Nevertheless, if theory *B* can be deduced from theory *A*, then theory *A* has more explanatory breadth than theory *B* (Cohen, 1989). A theory of high explanatory breadth would include all relevant constructs and relationships, and account for all known data in the field to which it applies. Thus, the broader the scope of a theory (i.e., the range of phenomena encompassed by the theory), the greater the explanatory breadth of its propositions.

Parsimony is the extent to which unnecessary constructs and propositions are excluded. It is defined in (Bacharach, 1989) as the ratio of propositions to testable hypotheses; the more hypotheses a proposition accounts for, the better. Thus parsimony interacts with explanatory (and predictive) power. There is a delicate balance with explanatory breadth, i.e., should some factors be deleted because they add little additional value to our understanding? Or as Whetten (1989, p. 490) formulated it: "Sensitivity to the competing virtues of parsimony and comprehensiveness is the hallmark of a good theorist."

Generality pertains to the extent to which a theory has a wide scope and how setting-independent the theory is. A major purpose of generalizing is to increase the explanatory breadth of a theory (Cohen, 1989). However, there is a trade-off here: Higher generality means broader applicability, but may demand more effort in operationalizing constructs and relationships to a given situation; while lesser generality might make a theory immediately applicable, but may compromise its explanatory power by abandoning explanation in terms of basic underlying mechanisms. Nevertheless, sensitivity to context is especially important for empirically-based theories: "Observations are embedded and must be understood within a context. Therefore, authors of inductively generated theories have a particular responsibility for discussing limits of generalizability" (Whetten, 1989, p. 492).

Finally, and of particular importance in an applied field, such as SE, is the *utility* of a theory, which refers to the degree to which the propositions of the theory can be used as input to decision-making, understanding and prediction in a given industrial setting (cf. Fig. 1). A good theory would thus be able to reduce the complexity of the empirical world, or in the words of Kurt Lewin (1945): "There is nothing so practical as a good theory." The utility aspect is far from new; about a century ago, this was also the focus of the pragmatists John Dewey (1899–1924) and William James (1907): "An idea agrees with reality, and is therefore true, if and only if it is successfully employed in human action in pursuit of human goals and interests, that is, if it leads to the resolution of a problematic situation in Dewey's terms."[2]

---

[2] The Internet Encyclopedia of Philosophy, http://www.iep.utm.edu/d/dewey.htm

3. Framework for Describing SE Theories

An SE theory is supposed to explain or predict phenomena occurring in SE. The typical SE situation is that an *actor* applies *technologies* to perform certain activities on an (existing or planned) *software system*. These high-level concepts or "archetype classes" with examples of sub-concepts or subclasses are listed in Table 2. One may also envisage collections of (component) classes for each of the (sub) classes. For example, component classes of a software system may be requirement specifications, design models, source and executable code, test documents, various kinds of documentation, etc.

In addition, appropriate characteristics of the classes, and their relative effect, should also be identified and measured. For example, the usefulness of a technology for a given activity may depend on characteristics of the software engineers, such as their experience, education, mental ability, personality, motivation, and knowledge of a software system, including its application domain and technological environment. Note that contexts or environments are supposed to be part of the descriptions of the respective archetype classes.

Hence, we propose that the constructs of an SE theory should typically be associated with these archetype classes themselves, any subclass specialised from them, possibly successively, or any class that is a component of the archetype classes or subclasses. The constructs could also be any of the attributes of those classes. An SE theory may be defined as a theory that includes at least one construct that is SE specific. For example, if the theory only relates to Actor, then the actor must be a software engineer or an SE team, SE project, etc.

The challenge of selecting or defining appropriate subclasses or component classes that represent constructs of a theory illustrates the need for commonly accepted taxonomies in SE. If the constructs of SE theories do not follow from well-defined and well-understood categories of phenomena, then new theories will frequently require new constructs, and as a consequence theories become difficult to understand and to relate to each other. Hence, development of taxonomies is needed to support theory building.

In the social and behavioural sciences, several scholars argue that theories should be general in the sense of being independent of time and place (Markovsky, 1994; Wagner, 1994; Cohen, 1989). SE theories, being more applied, and at the

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.

current stage of development, would seem to be somewhat dependent of both time and place. The fact that reality changes also in the SE world means that the validity or usefulness of an SE theory may be temporary. This, in turn, might indicate that *time* should be a factor of an SE theory, for example, change in education, and thus skill, of software engineers may change the validity of a theory. However, we would recommend not including time as part of the theory, but rather attempt to identify the underlying factors that may change over time. In the example of skill above, one should indicate in either the propositions or scope that the theory applies for a certain skill level.

Similarly, *place* is not interesting in SE per se. Place may be a placeholder for cultural, organisational and technological context factors that may affect a theory. However, we would also in this case urge scholars to be explicit on the underlying factors that, we believe, would be associated with one of the four archetype classes.

The constructs, propositions and their explanations, and the scope of a SE theory should be explicitly and clearly presented. We will illustrate how these four parts may be used in a simple example theory. This example is meant to illustrate the main initiating steps of building an SE theory from scratch (Mode (3) at Level 1, Sect. 2.3). Table 3 shows the constructs, the propositions, two examples of explanations, and the scope of an initial theory of the effect of using a development method based on UML (Booch et al., 1999) (in contrast to not using a thorough and systematic method covering all the phases from requirements analysis to testing). The background and steps in the development of the theory will be described in Sect. 4. For space considerations, only explanations E4 and E5, corresponding to, respectively, propositions P4 and P5 are shown in Table 3. The archetype classes associated with the respective constructs are shown in Fig. 3.

We also propose a notation (partly based on UML) to illustrate theories graphically. Figure 3 shows the relationships among the constructs of the UML-based development theory, including what affects what, using this notation. The notation has the following informal semantics:

A construct is represented as a class or an attribute of a class. A class is drawn as a box, and its name is written in the top of the box, e.g., "Distributed project" in Fig. 3). A class may be a subclass (using the UML generalization arrow) or a component class (drawn as a box within another box, e.g., "Team" is a component of "Distributed project"). Typically, if the construct is a particular value of a variable, then the construct is modelled as a subclass or component-class, e.g., the value "Distributed project" of the variable "Actor." On the other hand, if focus is on the variation of values, then the construct is a variable that is modeled as an attribute, e.g., "Costs." An attribute is written as a text in the lower part of a class box (below a horizontal bar).

A relationship is modelled as an arrow; an arrow from *A* to *B* means that *A* affects *B*, where *A* is a class or an attribute, and *B* is an attribute. In a relationship, *B* may also be a relationship itself, represented by an arrow. *A* is then called a moderator, e.g., "Training" in Fig. 3. This means that *A* affects the direction and/or strength of the effect of the relationship *B* (Baron and Kenny, 1986). The relationships

Table 3 Constructs, propositions, example explanations and scope of the theory of UML-based development

Constructs	
C1	*UML-based development method*
C2	*Costs* (total number of person hours in the project)
C3	*Communication* (ease of discussing solutions within development teams and in reviews)
C4	*Design* (perceived structural properties of the code)
C5	*Documentation* (the documentation of the system for the purpose of passing reviews as well as for expected future maintainability)
C6	*Testability* (more efficient development of test cases and better quality, i.e., better coverage)
C7	*Training* (training in the UML-based method before the start of the project)
C8	*Coordination* (of requirements and teams)
C9	*Legacy code* (code that has not been reverse engineered to UML-models)

Propositions	
P1	The use of a UML-based development method increases costs
P2	The use of a UML-based development method positively affects communication
P3	The use of a UML-based development method positively affects design
P4	The use of a UML-based development method positively affects documentation
P5	The use of a UML-based development method positively affects testability
P6	The positive effects of UML-based development are reduced if training is not sufficient and adapted
P7	The positive effects of UML-based development are reduced if there is insufficient coordination of modelling activities among distributed teams working on the same project
P8	The positive effects of UML-based development are reduced if the activity includes modification of legacy code

Explanations	
E4	The documentation is
	– More complete
	– More consistent due to traceability among models and between models and code
	– More readable, and makes it easier to find specific information, due to a common format
	– More understandable for non-technical people
	– May be viewed from different perspectives due to different types of diagram
E5	Test cases based on UML models
	– Are easier to develop
	– Can be developed earlier
	– Are more complete
	– Have a more a unified format
	Moreover, traceability from requirements to code and test cases makes it is easier to identify which test cases must be run after an update

Scope	
	The theory is supposed to be applicable for distributed projects creating and modifying large, embedded, safety-critical subsystems, based on legacy code or new code

are specified further into propositions of the theory, as indicated in Fig. 3; the propositions P6–P8 are examples of moderators.

The scope of the theory is also illustrated in the diagram. Scope conditions are typically modelled as subclasses or component classes. Figure 3 shows that our
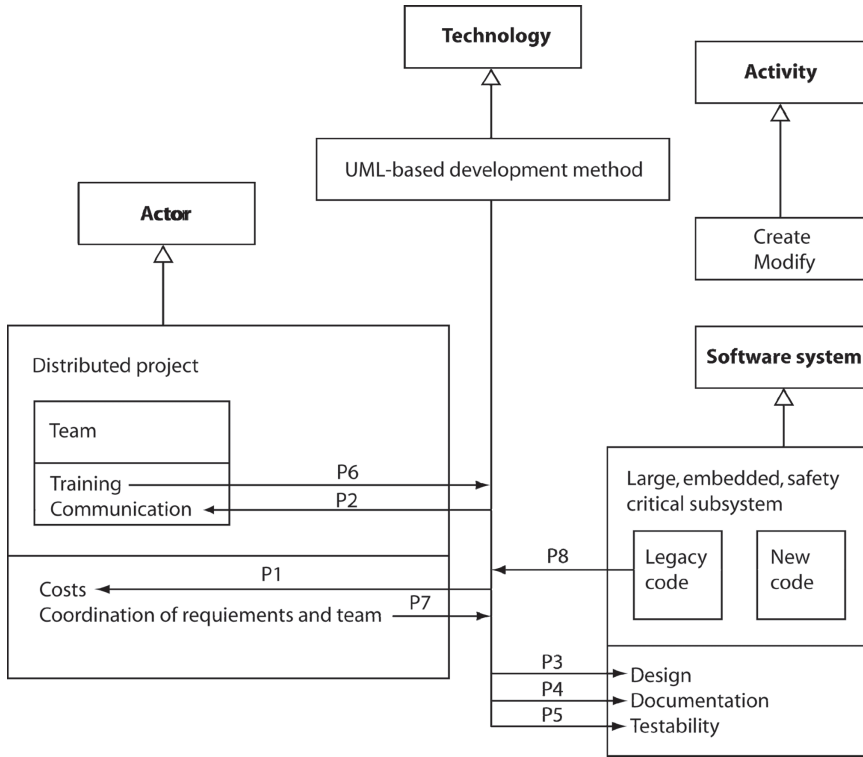
Fig. 3 A theory for the effect of UML-based development

example theory is constrained to "Distributed projects," "Create, modify" activities, and "large, embedded, safety critical subsystems" of a software system. This means, for example, that "Plan" and "Analyse" (two other subclasses of the archetype class "Activity") are outside the scope of this theory. In this example, all the archetype classes are included, but, generally, if any of the archetype classes are not included, then it is assumed that the theory is so general that it is independent of those classes. Note that one purpose of defining the four archetype classes is that we claim that any scholar who propose a SE theory should at least consider whether all of them should be included and specifed. For example, a theory of group performance in software development technical review (Sauer et al., 2000) was perceived by Land et al. (2003) to be too general for a SE context, and was thus specialised to also include, for example, dependencies to various components of a software system, such as requirements documents, designs, codes, test cases/plans and user manuals.

4. Steps in Building SE Theories

The theory-building process in an applied discipline such as SE is a continuous and iterative process of proposing, testing, and modifying theories. We do not always have to start from scratch when proposing a new theory; we can often start the process by adapting and modifying existing theories either from within SE or from related disciplines. However, in many cases, there are no established theories, neither in SE nor in the related disciplines, that are relevant for answering important SE research questions. In these cases, we may attempt to build theories by conducting, for example, case studies and experiments. We may also establish theories by reviewing and synthesizing related research in SE or by reviewing and synthesizing relevant research in related disciplines. Section 4.1 describes five steps in the building of theories. Section 4.2 illustrates each step by an example from an exploratory case study of UML-based development. Note that in practice these steps will often be carried out iteratively and partly in parallel.

4.1. Five Steps in Theory Building

4.1.1. Step 1: Defining the Constructs of the Theory

The first step of the theory-building process involves identifying and defining the constructs of the theory. In the context of this first step, there are five ways in which we might seek to make a theoretical contribution (Weber, 2003):

- Defining new constructs as the basis for building a new theory about some phenomena. These constructs might encompass phenomena that have not been the focus of prior theories. Alternatively, they might conceive phenomena that have been the focus of prior theories, but in a different way. As a result, we need to build a new theory of the phenomena that reflects this conception.
- Introducing new constructs into an existing theory to better account for the phenomena that are the focus of the theory.
- Deleting constructs from an existing theory to provide a more parsimonious account of the phenomena that are the focus of the theory.
- Adding and deleting constructs from an existing theory to provide a different, and hopefully better, account of the phenomena that are the focus of the theory.
- Defining the constructs of an existing theory more precisely or conceptualizing them in somewhat different ways.

4.1.2. Step 2: Defining the Propositions of the Theory

The second step of the theory-building process consists of specifying the propositions of the theory. In the context of this second step, there are four ways in which we might seek to make a theoretical contribution (Weber, 2003):

- Defining new propositions among existing or new constructs in a theory to better account for the phenomena that are the focus of the theory.
- Deleting propositions among the constructs of an existing theory to provide a more parsimonious account of the phenomena that are the focus of the theory.
- Adding and deleting propositions among the constructs of an existing theory to provide a different, and hopefully better, account of the phenomena that are the focus of the theory.
- Define the propositions in an existing theory more precisely or conceptualize them in somewhat different ways, for example, by specifying the functional form of a proposition previously conceived as a simple association between two constructs.

4.1.3. Step 3: Providing Explanations to Justify the Theory

The third step of the theory-building process, providing explanations – the "why" – of the theory, is probably the most challenging. The core issue of this step is to provide explicit assumptions and logical justifications for the constructs and propositions of the theory. In the context of this third step, there are five ways in which we might seek to make a theoretical contribution:

- Explicitly stating the assumptions of the conceptual underpinnings of the constructs and propositions of the theory.
- Challenging or extending existing knowledge of the constructs and propositions of the theory.
- Borrowing perspectives from other disciplines to explain the constructs and propositions of the theory.
- Providing logical justifications based on interpretations of an empirical study.
- Providing logical justifications based on interpretations of a synthesis of all prior empirical evidence within the scope of the theory. Such synthesis, which possibly includes replicated studies, might also expand the scope of a theory:

> Methodological authorities generally regard replication, or what is also referred to as "repeating a study," to be a crucial aspect of the scientific method. … Heavily differentiated replication leads to extensions of the scope of the result and hence its subsequent practical applicability, that is, to other firms, other industries, different types of executives, other years, or whatever. … Varying the conditions between different replications not only extends the scope of the generalization and determines its limits, but also tells us about some of the factors that do, or do not, affect the result *causally*.
>
> (Lindsay and Ehrenberg, 1993)

4.1.4. Step 4: Determining the Scope of the Theory

The fourth step of the theory-building process is concerned with determining the scope of the theory, which is especially important for empirically-based SE theories.

In the context of this fourth step, there are two ways in which we might seek to make a theoretical contribution (Weber, 2003):

- Specifying more precisely the values of a construct for which the theory will hold, or conversely, specifying more precisely the values of a construct for which the theory will not hold.
- Specifying more precisely the combinations of values of the constructs for which the theory will hold, or conversely, specifying more precisely the combinations of values of the constructs for which the theory will not hold.

4.1.5. Step 5: Testing the Theory Through Empirical Research

The last step of the theory-building process involves examination of the validity of the theory's predictions through empirical studies. In the context of this last step, different types of empirical studies might be applied, which entails different method-specific sub-steps as well as method-specific strengths and limitations in the theory-building process. For example, the following separates case studies from experiments with respect to theory building:

- In case studies, new insights typically evolve based on the data, while in experiments, previous knowledge must often be applied to explain results.
- In case studies, hypotheses are examined for each case study unit, while in experiments they are examined for an aggregate of the units using statistical hypothesis building/testing.
- Theories derived from case studies tend to become less general than those derived from experiments.
- Theories derived from case studies typically have more focus on explanations than those derived from experiments.

In testing a theory, the following general steps must, nevertheless, be considered:

- Choosing an appropriate research setting and sample. The sample does generally not only include the actors, but also the sample of technologies, activities (tasks) and systems.
- Operationalizing theoretical constructs into empirical variables.
- Operationalizing theoretical propositions into empirically testable hypotheses.

For the purpose of describing the extent to which a theory has been validated, we introduce the two terms *scope of interest* and *scope of validity* of a theory (Fig. 4). "Scope of interest of a theory" is what we have simply denoted "scope of theory" above. In contrast, a theory's scope of validity refers to that part of the scope of interest in which the theory has actually been validated. The scope of validity of a theory is the accumulated scopes of validity of the results of the studies that have tested the theory, or the studies from which the theory has been generated. Figure 4 shows that three studies have been conducted, and the area made up by the three scopes of validity of the three studies corresponds to the scope of validity of the theory (so far). The ultimate goal is that the scope of validity becomes equal
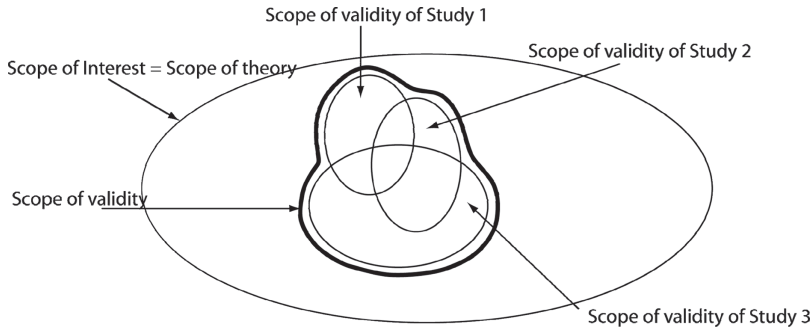
Fig. 4 Scope of interest versus scope of validity

to the scope of interest. The first consideration to make in testing a theory is to make sure that the study fits the theory's scope of interest. Otherwise, the results would be irrelevant to that theory. Moreover, in a given study, typically only a part of the scope of interest can be tested. If that part has not been tested before, and is supported by the study, then the current scope of validity has been extended. However, note that empirical support or inconsistencies between theoretical propositions and empirical observations do not necessarily imply that the theory is validated or disconfirmed, respectively. Judgements regarding the validity of the theory require that the study is well conducted, and not encumbered with, for example,

- Invalid operationalization of theoretical constructs and propositions
- Inappropriate research design
- Inaccuracy in data collection and data analysis
- Misinterpretation of empirical findings

4.2. Example of Generating Theory from an Exploratory Case Study: An Initial Theory for UML-Based Development in Large Projects

The example theory presented in Sect. 3 was derived from an exploratory case study that was conducted in the global company ABB (Anda et al., 2006; Anda and Hansen 2006). The purpose of the case study was to investigate the use of a UML-based method, and in particular to identify benefits and challenges, as well as their causes, of applying such a development method in a large, distributed development project. The goal of the project was to develop a new safety-critical process-control system based on several existing systems. The development took place at four sites in three countries. The total workforce comprised approximately 230 people, and approximately 100 of them were involved in using the UML-based method. This was the first project in ABB with large-scale use of UML. The company consequently

wanted to find out whether the UML-based development method improved the quality of the development process and the resulting software product compared with earlier projects that had not used UML.

Data was collected through individual interviews, questionnaires and project documents.

4.2.1. Step 1: Defining the Constructs

In this case study, as is frequently the situation in case studies, much of the data collected was in the form of texts, for example, transcripts of interviews and project documents. These texts were subject to qualitative analysis based on the principles of "grounded theory" (Strauss and Corbin, 1998), which is an established technique for distilling concepts from textual data. Central concepts are candidate constructs for a theory. Hence, the constructs of a theory derived from one or more case studies in this way are well grounded in the data of the case(s).

The interviews of the case study were analyzed using the grounded-theory principles of open, axial and selective coding. In open coding, categories of phenomena are identified; in axial coding, categories are related to each other; and in selective coding, the central categories that are candidates for constructs are identified. The following characteristics of the actors (project, teams and individuals), activities and software system, with corresponding definitions for use in this context, were identified and evolved into the constructs given in Table 3.

4.2.2. Step 2: Defining the Propositions

After identifying the constructs, the next step in text analysis, according to "grounded theory," is to analyze emerging relationships between the constructs. In the ABB case study, relationships were identified from the interviews, for example, relationships were identified between the use of the UML-based development method and several positive aspects of the project documentation such as more documentation, better structured documentation. The identified relationships were checked against each case, that is, against each interview. Relationships that had clear support from the data were candidates for being included in the propositions of the theory. Furthermore, the relationships were validated using questionnaires (although not all relationships could be validated in this way) and compared with literature on UML-based development. Finally, the relationships that were supported by all the data, and that included the candidate constructs identified in Step 1, were aggregated in to the propositions described in Table 3.

Ideally, we would have liked the relationships expressed in the propositions to be more quantitative, in accordance with the view of Dubin (1978, p. 170): "the proposition predicts the specific values that one unit will have in relation to the values of another." Hence, the propositions listed in Table 3 may be regarded as initial propositions. Follow-up studies may help quantify the propositions to some

extent, but it seems unrealistic in the near future to provide quantitative propositions in SE. At least, another of magnitude of more empirical studies would then be needed (Sjøberg et al., 2007).

4.2.3. Step 3: Providing Explanations

Explanations for each proposition were identified in the same way as were the propositions. The difference between a proposition and an explanation is that the former is a relationship among constructs, and the latter is a relationship among constructs and other categories, which are not central enough to become constructs (see explanation of "grounded theory"-terminology given under step (1)). This step is typically more elaborate in theories derived from case studies than in theories derived from experiments, because qualitative data, which typically are better at explaining phenomena, are more frequently collected. For two of the propositions, the corresponding explanations were shown in Table 3.

4.2.4. Step 4: Determining the Scope

Since this theory is derived from "grounded theory," the scope of validity of the study would form the starting point for the scope of the theory, which would generally be too narrow to be interesting for a theory. Nevertheless, defining the initial scope is not trivial; the number of potential scope conditions of a case study is large, and there is little guidance in the SE literature regarding how the scope of a case study should be documented, Kitchenham et al. (2002) state: "Be sure to specify as much of the industrial context as possible. In particular, clearly define the entities, attributes and measures that are capturing the contextual information."

In practice, judgment must be exercised in the description of scope conditions and the level of detail of their description. Below we will describe what we consider to be the relevant conditions for the scope of validity of the theory (which is the same as the scope of this case study since the theory is only based on one study so far, see Fig. 3). We will then describe what we think should be the scope of the theory. The scope of validity is too narrow as a scope of a theory, because it would make the theory applicable to very few software projects. This theory is at Level 1 (Sect. 2.3), which indicates a scope of interest relatively similar to the scope of validity of the study, but based on the study and on other work on UML-based development, we propose a wider scope of the theory.

Technology

– *Scope of validity:* In the UML-based development method applied in the study, use case diagrams, sequence diagrams and class diagrams were compulsory, while the use of other UML diagrams was at the discretion of the individual teams.
– *Scope of interest:* UML-based development methods

Actor

– *Scope of validity:* The project was distributed with development at four sites in three countries. Some of the teams were also distributed with team members working at different sites. The teams were medium-sized (typically 8–10 people in each team), the team members mostly had good knowledge of the application domain, their educational background was typically at the level of an MSc, and most were newcomers to the use of UML at the start of the project, but became quite proficient in UML during the project due to it's size.
– *Scope of interest:* Projects with distributed teams

Software system

– *Scope of validity:* The system to be developed was large (approximately 1,000 requirements and 3–4 mill. lines of code), which was divided into approximately ten large subsystems. The software was embedded, C and C++ were used as programming languages, the system was safety-critical and the development followed the requirements of the safety standard IEC61508. Some parts of the system were developed from scratch while others were based on legacy code of existing systems.
– *Scope of interest:* Large, embedded, safety-critical system, possibly based on legacy code.

Activity

Both scope of validity and scope of interest are "create" and "modify."

4.2.5. Step 5: Testing the Theory

This example theory has not yet been tested.

5. Evaluating the Example Theory

This section evaluates the initial theory for UML-based development in large projects described in Sect. 3 according to the criteria presented in Sect. 2.

Testability

The constructs and propositions of the theory are understandable, internally consistent and free from ambiguities, at least from the point of view of developers and practitioners familiar with the topic of the theory. Hypotheses can be derived from the propositions, the scope conditions are clearly defined, although some of the constructs, such as "large" and "distributed," assume the existence of taxonomies of software systems in order to be precisely defined. The theory can be empirically tested in case studies or surveys of development projects that fall within the scope of the theory. Most material for such testing, in the form of inter-

view guides and analysis procedures are available for use, see (Anda et al., 2006). Such empirical testing would consist in testing whether the propositions of the theory are supported in other projects. The scope condition indicating "large subsystems" means that it is difficult, that is, would be very costly, to test this theory in experiments. We consider the testability of this theory as moderate.

Empirical support

There are few other empirical studies on benefits and challenges of UML-based development. Three empirical studies on UML-based development have a similar or wider scope than the scope of our theory (Baker et al., 2005; Petit, 2004; Dobing and Parsons, 2006). These studies all have a slightly different focus than the study on which our theory is based, but they support different propositions of our theory; (Petit, 2004) supports P2 on communications, (Dobing and Parsons, 2006) supports P4 on documentation, and (Baker et al., 2005) supports P5 on testing. Furthermore, two studies on UML-based development have different scope conditions; Arisholm et al. (2006) report a controlled experiment with students performing maintenance activities. The results support P3 on design. MacDonald et al. (2005) report a student project that supports P2 on communication and P8 on legacy development. If more empirical studies are conducted on UML-based development, it may be possible to extend the scope of our theory and in that case those two studies may also be included as part of the empirical support for the theory. Since the example theory is supported or partly supported by all comparable empirical studies on UML-based development, we consider the empirical support for this theory to be moderate.

Explanatory power

Many factors influence the results of software creation and modification activities. Hence, we expect that SE theories will seldom have high explanatory power. This theory is at Level 1 (see Sect. 2) and accounts for some, but far from all aspects of software creation and modification with the use of UML-based development. We consider the explanatory power of the theory as low.

Parsimony

A theory derived from one case and with the use of "grounded theory" will typically be quite complex, with many constructs and propositions, but we have attempted to use a minimum of constructs and propositions in this theory. We consider the parsimony of the theory as moderate.

Generality

The scope of this theory is narrow, something which is typical for theories at Level 1 theories. We consider the generality of the theory as low.

Utility

This theory can be used in the decision making in projects for which it is relevant with little adaptation. We consider the utility of the theory as high.

6. Summary and Future Work

The motivation for the work reported in this chapter is that without a stronger focus on theory building in the empirical SE community, we will probably continue to produce many isolated, exploratory studies, which will limit our ability to aggregate knowledge. Even a weak theory may frequently be better than no theory.

We have described a framework that we believe will benefit the process of proposing, testing and modifying and describing SE theories. We illustrated the framework with an example of how to build theories systematically from an exploratory case study using the technique of "grounded theory." Future work will include describing how to build theories from experiments and from systematic reviews of the SE literature.

The framework suggested above is not intended as "silver bullets" to build and document theories; theory development requires significant reflection and skill regarding study design and argumentation. Hence, there is a need for more systematic teaching of research methods and theory building as part of SE education.

During our work with a survey to identify and describe theories used in SE experiments (Hannay et al., 2007), we experienced that there is no simple way of identifying empirically-based theories that are used or built in SE. There are web sites for collecting and documenting theories in psychology[3] and information systems[4]. In the same manner, Simula Research Laboratory has begun building a site for empirically-based SE theories, see se-theory.simula.no. We believe that this will make it easier for scholars to find relevant theories for their research and that this will stimulate the community to collaborate on building new theories and on improving existing theories.

References

Abdel-Hamid, T.K., Sengupta, K. and Ronan, D., Software project control: an experimental investigation of judgement with fallible information, *IEEE Transactions on Software Engineering*, 19(6): 603–612, 1993.

Anda, B.C.D. and Hansen, K., A case study on the application of UML in legacy development. In *ISESE'2006 (Fifth ACM-IEEE International Symposium on Empirical Software Engineering)*, J. Maldonado and C. Wohlin (eds.), Rio de Janeiro, Brasil, ACM Press, September 21–22, pp. 124–133, 2006.

Anda, B.C.D., Hansen, K., Gullesen, I. and Thorsen, H.K., Experiences from using a UML-based development method in a large safety-critical project, *Empirical Software Engineering*, 11(4): 555–581, 2006.

---

[3] http://changingminds.org/explanations/theories/theories.htm

[4] http://www.istheory.yorku.ca/

Arisholm, E., Briand, L.C., Hove, S.E. and Labiche, Y., The impact of UML documentation on software maintenance: an experimental evaluation, *IEEE Transactions on Software Engineering*, 32(6): 365–381, 2006.

Bacharach, S.B., Organizational theories: some criteria for evaluation, *Academy of Management Review*, 14(4): 496–515, 1989.

Baker, P., Loh, S. and Weil, F., Model-driven engineering in a large industrial context – motorola case study. In *MoDELS 2005, LNCS 3713*, L. Briand and C. Williams (eds.), New York, Springer-Verlag, pp. 476–491, 2005.

Baron, R.M. and Kenny, D.A., The moderator-mediator variable distinction in social psychological research: conceptual, strategic, and statistical considerations, *Personality and Social Psychology*, 51(6): 1173–1182, 1986.

Basili, V.R., Editorial, *Empirical Software Engineering*, 1(2), 1996.

Basili, V.R., Shull, F. and Lanubile, F., Building knowledge through families of experiments, *IEEE Transaction on Software Engineering*, 24(4): 456–473, 1999.

Booch, G., Rumbaugh, J. and Jacobson, I. *The Unified Modeling Language User Guide*, Boston, MA, Addison-Wesley, 1999.

Bunge, M., *Scientific Research: The Search for a System*, New York, Springer-Verlag, 1967.

Burkhardt, J.M., Detienne, F. and Wiedenbeck, S., Object-oriented program comprehension: effect of expertise, task and phase, *Empirical Software Engineering*, 7(2): 115–156, 2002.

Carroll, J. and Swatman, P.A., Structured-case: a methodological framework for building theory in information systems research, *European Journal of Information Systems*, 9: 235–242, 2000.

Cohen, B., *Developing Sociological Knowledge: Theory and Method*, 2nd edn, Belmont, CA, Wadsworth Publishing, 1989.

Dobing, B. and Parsons, J., How UML is used, *Communications of the ACM*, 49(5): 109–113, 2006.

Dewey, J., *The Middle Works, 1899–1924*, Vol. 15, J.A. Boydston, (ed.), Carbondale, Southern Illinois University Press, 1976–1983.

Dubin, R., *Theory Building*, Free Press, New York, 1978.

Endres, A. and Rombach, D., *A Handbook of Software and Systems Engineering. Empirical Observations, Laws and Theories. Fraunhofer IESE Series on Software Engineering*, Pearson Education Limited, 2003.

Feynman, R.P., *QED – The Strange Theory of Light and Matter*, Penguin Science, Harmondsworth, 1985.

Franck, R., *The Explanatory Power of Models*, Dordrecht, Kluwer Academic Publishers, 2002.

Glaser H.G. and Strauss A.L., *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Hawthorne, NY, Aldine Publishing Company, 1967.

Godfrey-Smith, P., Pragmatism: philosophical aspects, *International Encyclopedia of the Social & Behavioral Sciences*, 17: 11954–11958, 2001.

Gregor, S., The nature of theory in information systems, *MIS Quarterly*, 30(3): 611–642, 2006.

Gruber, T.R., A translation approach to portable ontology specifications, *Knowledge Acquisition*, 5(2): 199–220, 1993.

Haig, B.D., An abductive theory of scientific method, *Psychological Methods*, 10(4): 371–388, 2005.

Hannay, J.E., Sjøberg, D.I.K. and Dybå, T., A systematic review of theory use in software engineering experiments, *IEEE Transactions on Software Engineering*, 33(2): 87–107, 2007.

Hawking, S. and Penrose R., *The Nature of Space and Time*, Princeton University Press, Princeton, NJ, 1996.

Herbsleb, D.J. and Mockus, A., Formulation and preliminary test of an empirical theory of coordination in software engineering, *ACM SIGSOFT Software Engineering Notes*, 28(5): 138–147, 2003.

Hevner, A.R. and March, S.T., The information systems research cycle, *IEEE Computer Society*, 36(119): 111–113, 2003.

Hevner, A., March, S.T., Park, J., and Ram, S, Design science research in information systems, *MIS Quarterly,* 28(1): 75–105, 2004.

James, W., *Pragmatism: A New Name for Some Old Ways of Thinkin*g, New York: Longman Green and Co, 1907.

Jørgensen, M. and Sjøberg, D.I.K., Generalization and theory-building in software engineering research. In *Empirical Assessment in Software Engineering (EASE2004)*, *IEE Proceedings*, pp. 29–36, 2004.

Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C., El Emam, K. and Rosenberg, J., Preliminary guidelines for empirical research in software engineering, *IEEE Transaction on Software Engineering*, 28(8): 721–734, 2002.

Land, L.P.W., Wong, B. and Jeffery, R., An extension of the behavioral theory of group performance in software development technical reviews, *Proceedings of the Tenth Asia-Pacific Software Engineering Conference Software Engineering Conference*, pp. 520–530, 2003.

Lewin, K., The research center for group dynamics at Massachusetts Institute of Technology, *Sociometry*, 8: 126–135, 1945.

Lindblom, C.E., Alternatives to validity. Some thoughts suggested by Campbell's guidelines, *Knowledge: Creation, Diffusion, Utilization*, 8: 509–520, 1987.

Lindsay, R.M. and Ehrenberg, A.S.C., The design of replicated studies, *The American Statistician*, 47: 217–228, 1993.

Lucas, J.W., Theory-testing, generalization, and the problem of external validity, *Sociological Theory*, 21: 236–253, 2003.

Lynham, S.A., The general method of theory-building research in applied disciplines, *Advances in Developing Human Resources*, 4(3): 221–241, 2002.

MacDonald, A., Russel, D. and Atchison, B. Model-driven development within a legacy system: an industry experience report, *Proceedings of the 2005 Australian Software Engineering Conference (ASWEC'2005). IEEE Computer Society*, pp. 14–22, April 2005.

March, S.T. and Smith, G.F., Design and natural science research on information technology, *Decision Support Systems*, 15(4): 251–266, 1995.

Markovsky, B., The structure of theories. In *Group Processes*, M. Foschi and E.J. Lawler, (eds.), Nelson-Hall Publishers, Chicago, pp. 3–24, 1994.

Merton, R.K., *Social Theory and Social Structure*, 3rd ed, The Free Press, New York, 1968.

Nagel, E., *The Structure of Science*, Hackett, Indianapolis, 1979.

Peirce, C.S., *Collected Papers*, Harvard University Press, Cambridge, MA, 1958.

Petit, R.G., Lessons learned applying UML in embedded software systems designs, *Proceedings of the Second IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (WSTFEUS'04)*, Vienna, Austria, May 11–12, pp., 75–79, 2004.

Popper, K., *The Logic of Scientific Discovery*, Hutchison, London, 1959.

Ramanujan, S., Scamell, R.W. and Shah, J.R., An experimental investigation of the impact of individual, program, and organizational characteristics on software maintenance effort, *Journal of Systems and Software*, 54(2): 137–157, 2000.

Rosenberg, A., *Philosophy of Science: A Contemporary Introduction*, Routledge, London, 2001.

Ruse, M. (1995*). Theory. The Oxford Companion to Philosophy. T. Honderich*, Oxford University Press, New York, 870–871.

Salmon, W.C., Four decades of scientific explanation. In *Scientific Explanation*, P. Kitcher and W.C. Salmon, (eds.), *Minnesota Studies in the Philosophy of Science*, Vol. 13, Minnesota Press, Series, pp. 3–219, 1989.

Sandborg, D., Mathematical explanation and the theory of why-questions, *The British Journal for the Philosophy of Science*, 49(4): 603–624, 1998.

Sauer, C, Jeffery, D.R., Land, L. and Yetton, P., The effectiveness of software development technical reviews: a behaviorally Motivated program of research, *IEEE Transactions on Software Engineering*, 26(1): 1–14, 2000.

Shadish, W.R., Cook, T.D. and Campbell, D.T., *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*, Houghton Mifflin, Boston, MA, 2002.

Simon, H.A., *The Sciences of the Artificial*, MIT Press, Cambridge, MA, 1996.

Sjøberg, D.I.K., Dybå, T. and Jørgensen, M., The future of empirical methods in software engineering research. In *Future of Software Engineering (FOSE '07)*, L. Briand and A. Wolf, (eds.), IEEE-CS Press, Minneapolis, US, pp. 358–378, 2007.

Sjøberg, D.I.K., Hannay, J.E., Hansen, O., Kampenes, V.B., Karahasanović, A., Liborg, N.-K. and Rekdal, A.C., A survey of controlled experiments in software engineering, *IEEE Transactions on Software Engineering*, 31(9): 733–753, 2005.

Strauss, A. and Corbin, J., *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 2nd ed, Sage, Thousand Oaks, CA, 1998.

Sutton, R.I. and Staw, B.M., What theory is not, *Administrative Science Quarterly*, 40: 371–384, 1995.

Thagard, P., *Conceptual Revolutions*, Princeton University Press, Princeton, NJ, 1992.

Tichy, W.F., Should computer scientist experiment more? 16 excuses to avoid experimentation, *IEEE Computer*, 31(5): 32–40, 1998.

Van Dijk, T.A. and Kintsch, W., *Strategies of Discourse Comprehension*, Academic Press, New York, 1983.

Van Fraassen, B., *The Scientific Image*, Oxford University Press, New York, 1980.

Wagner, D.G., The growth of theories. In *Group Processes*, M. Foschi and E. J. Lawler, (eds.), Nelson-Hall Publishers, Chicago, pp. 25–42, 1994.

Weber, R., Editor's comments, *MIS Quarterly*, 27(3): 3–12, 2003.

Weick, K.E., Theory construction as disciplined imagination, *Academy of Management Review*, 14(4): 516–531, 1989.

Whetten, D.A., What constitutes a theoretical contribution, *Academy of Management Review*, 14(4): 490–495, 1989.

Yin, R.K., *Case Study Research: Design and Methods*, Sage Publications, Thousand Oaks, CA, 1984.

Yin, R.K., *Case Study Research: Design and Methods*, *Applied Social Research Methods Series* 5, 3rd ed, Sage Publications, Thousand Oaks, CA, 2003.