G Model
JSS-9255; No. of Pages 13
ARTICLE IN PRESS
The Journal of Systems and Software xxx (2013) xxx–xxx
Contents lists available at ScienceDirect
The Journal of Systems and Software
journal homepage: www.elsevier.com/locate/jss

Empirical research methods for technology validation: Scaling up to practice

Roel Wieringa*

Department of Electrical Engineering, Mathematics, and Computer Science, University of Twente, The Netherlands[1]

ARTICLE INFO

Article history:
Received 30 November 2012
Received in revised form
11 November 2013
Accepted 16 November 2013
Available online xxx

Keywords:
Empirical research methodology
Technology validation
Scaling up to practice

ABSTRACT

Before technology is transferred to the market, it must be validated empirically by simulating future practical use of the technology. Technology prototypes are first investigated in simplified contexts, and these simulations are scaled up to conditions of practice step by step as more becomes known about the technology. This paper discusses empirical research methods for scaling up new requirements engineering (RE) technology.

When scaling up to practice, researchers want to generalize from validation studies to future practice. An analysis of scaling up technology in drug research reveals two ways to generalize, namely inductive generalization using statistical inference from samples, and analogic generalization using similarity between cases. Both are supported by abductive inference using mechanistic explanations of phenomena observed in the simulations. Illustrations of these inferences both in drug research and empirical RE research are given. Next, four kinds of methods for empirical RE technology validation are given, namely expert opinion, single-case mechanism experiments, technical action research and statistical difference-making experiments. A series of examples from empirical RE will illustrate the use of these methods, and the role of inductive generalization, analogic generalization, and abductive inference in them. Finally, the four kinds of empirical validation methods are compared with lists of validation methods known from empirical software engineering. The lists are combined to give an overview of some of the methods, instruments and data analysis techniques that may be used in empirical RE.

1. Introduction

Empirical assessment of technology comes in two flavors, which in this paper will be called technology validation and technology evaluation, respectively. *Technology validation* is defined here as the assessment of a simulation of the technology in a simulation of its intended context of use, in order to predict what would happen if the technology were actually used by stakeholders in this intended context. We take the term "simulation" in a very wide sense as the representation of the functioning of one system or process by means of the functioning of another.[2] For example, a new requirements prioritization technique may be tested by experimenting with it in a classroom. This is a validation if the classroom experiment represents some aspects of what would happen if the technique was used in practice.

Validation always involves *scaling up to practice*, which means that successive tests take place under increasingly realistic conditions. For example, the inventor of a requirements prioritization technique may use this technique in a real-world project. This validation would resemble real-world use of the technique more than a classroom experiment, except that it is still the inventer herself who uses the technique.

A technology has been *transferred to practice* if it has been packaged, marketed, distributed, sold or otherwise made available to users, and is now being used independently from the context in which it was invented or tested. After transfer to practice other people than its inventors are using it, and they are using it to achieve their own goals, without help or other kind of intervention from its inventors, and after investment of their own time and/or money to learn to use the technology.

Technology validation is to be contrasted with *technology evaluation*, defined here as the empirical assessment of a technology as and when used in practice. For example, an RE researcher may study how a prioritization technique is used in real-world projects by means of observational case studies. Where a validation study aims to make predictions, based on simulations, about how a technology would perform if transferred to practice, an evaluation study assesses what has happened in the actual use of the technology after it has been transferred in practice. This follows terminology commonly used in the social sciences, where an evaluation study is an empirical assessment of some social intervention that

* Tel.: +31 53 489 4189.
E-mail address: r.j.wieringa@utwente.nl
[1] http://www.cs.utwente.nl/ roelw.
[2] http://www.merriam-webster.com.

0164-1212/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2013.11.1097

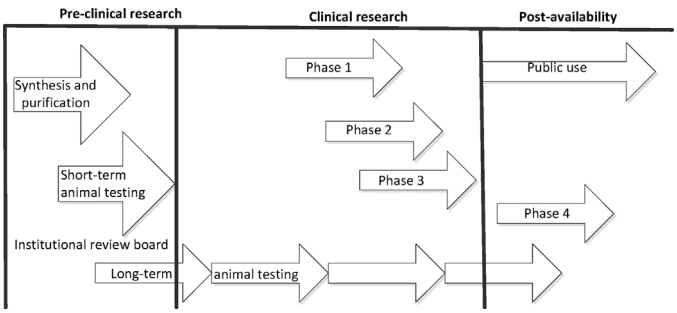

Fig. 1. The new drug development and review process of the U.S. Food and Drug Administration.

has been performed, such as a recently implemented teaching method in schools, to investigate its impact in practice (Babbie, 2007).

Technology validation is a process of scaling up to practice in all engineering sciences. For example, the inventors of the jet engine validated their designs by building increasingly realistic prototypes and testing them in increasingly realistic environments (Constant, 1980). In this paper I will summarize and analyze the ways in which we can scale requirements engineering (RE) technology up to practice.

The RE technology being validated could be techniques, methods, notations, algorithms, etc. used for various requirements engineering tasks such as requirements elicitation, goal analysis, requirements specification, requirements prioritization, traceability management, requirements maintenance, etc. *Requirements* in this paper are defined as desired properties of a system. *Goals*, by contrast, are states of the world desired by stakeholders, and for which the stakeholders have committed a budget (time or money) to achieve them. All requirements are goals because they are desired by stakeholders, and stakeholders have committed a budget to achieve them. But not all stakeholder goals are system requirements. Stakeholders have many goals not stated in terms of desired system properties at all.

Davis and Hickey (2004) proposed using the methodology of New Drug Development for scaling up RE technology. I will pursue this idea further in Section 2 and focus in particular on the inferences used in New Drug Development when generalizing from the object of validation research to instances of real-world use of the technology, and show that these inferences can be used in RE research too. In Section 3, I present four methods for empirical technology validation, and show how the generalization methods identified in Section 2 can be used in them. This is illustrated by a series of examples from empirical requirements engineering. Finally, in Section 4, I review the empirical software validation methods identified by Zelkowitz and Wallace (1997, 1998) and by Glass et al. (2001) and show how they fit into the framework presented in this paper, and add a list of examples of techniques for measurement and data analysis used in empirical software engineering. Section 5 ends the paper with a brief summary and outlook.

2. Scaling up

2.1. Scaling up in drug research

Davis and Hickey (2004) were the first to apply the New Drug Development and Review Process of the U.S. Food and Drug Administration to RE technology transfer. I summarize the process in Fig. 1. The following description is based on information provided by the FDA[3] and the explanations given by Davis and Hickey (2004), Cowan (2002) and Molzon and Pharm (2005). My analysis goes beyond that of Davis and Hickey by analyzing the three kinds of inference used in this process. I will indicate the analogy of each stage of the New Drug Development process with a stage in scaling up RE technology.

2.1.1. Pre-clinical research

Pre-clinical research is the exploration and validation of drugs before testing it on people. It consists of a synthesis and purification task, and of testing the drug on animals.

In *synthesis and purification*, a chemical is identified in the laboratory as a potentially effective drug, based on earlier experience reported in the literature, biochemical knowledge, and knowledge of the human body. A theory about why this could be an effective drug to improve a medical condition is postulated. This is the initial version of a design theory that will be tested and elaborated in the following stages of the New Drug Development process. It corresponds in RE research to the initial idea for a new RE technique and the initial justification that this idea might work to solve some RE problem. The rest of the process aims to validate and elaborate this design theory in increasingly realistic contexts.

Animal tests are done to show that the drug would be safe to use in people and to investigate in more detail the biochemical mechanisms that produce the drug's effects. If there are no negative effects in the investigated contexts (i.e. in animals), and if the mechanisms found in these contexts are expected to be similar to those in human bodies, then this is evidence that the drug is probably safe for humans, and a request is submitted to institutional review boards for permission to test the drug in humans. Usually two different animal species are taken, because a drug usually has different effects in different species (Cowan, 2002). Short term testing in animals can take up to three years but on the average takes 18 months.

The animals are used for testing as *natural models* of the intended real-world context of the drug, namely the human body. The analog in RE research would be testing a new RE technique on students in a laboratory, to study the effects of the technique and the mechanisms by which these effects are produced. Although the goal of this research would not be to establish evidence for safety of the technique, the purpose would still be to assess whether the benefit of using the technique in practice would outweigh the cost and risk of doing so.

Long-term animal research investigates the long-term effects of using a drug, and may continue into the post-availability stage. This is an example of validation research that continues after transfer of the new technology into practice. Long-term animal research has the logic of validation research, as it simulates the effects of a drug by using a model, and is used to predict what would happen to human bodies. This can be done in RE research too. For example, the effect of using the UML on the comprehension of programs can be investigated in the laboratory, using students as subjects, long after the UML had been transferred to practice. Insights from such a study could be used to predict the effect of UML on comprehension of programs in practice.

2.1.2. Clinical research

In clinical research, the drug is tested on people. It consists of three phases.

- In phase 1, random samples of 20–80 healthy subjects are used to test the drug. The goal is to investigate side effects and the

[3] http://www.fda.gov/Drugs/DevelopmentApprovalProcess/SmallBusinessAssistance/ucm053131.htm.

so-called *mechanism of action* of the drug, which is the biochemical interaction by a which a drug produces pharmacological effects. If possible, the effectiveness (positive effects) of the drug is investigated too. This phase may last several months and ends when researchers are sufficiently certain that the drug is safe to use in patients.

- In phase 2, several hundred patients are used to investigate the effect of the drug in controlled studies by means of randomized controlled trials. The goal is to investigate the effectiveness in patients with a specific disease or condition, and to investigate short-term side-effects and identify possible risks. Phase 2 may last several months to two years and ends when knowledge about effectiveness, side-effects and risks is deemed sufficiently well established to do large-scale tests in phase 3.
- In phase 3, controlled and uncontrolled trials with several hundred to several thousand patients are done to gather additional evidence about effectiveness and safety. The goal is to find out if the effectiveness and safety claims can be generalized to the population of all possible patients. Phase 3 may last one to four years and ends when researchers think the claims about effectiveness and safety of the drug can be generalized to the population.

As we will see below, when validating RE technology, similar research goals exist, and they are pursued just as in drug research by experimental research in the lab and in the field.

2.1.3. Post-availability studies

After a drug has been approved and is made available to a market, assessment continues in so-called phase 4 studies, for example by surveys or observational case studies of patients using the drug. In our terminology, these are evaluation studies. Post-availability studies are done in RE research too, for example when researchers investigate the effect of using the UML on coding errors in real-world projects.

2.2. Design theories: artifact in context produces effects by mechanisms

I will now analyze the logic of drug validation in more detail, with a view to drawing conclusions about the logic of RE technology validation. In other words, I treat the drug development process as a *model* of the RE technology transfer process, that we can investigate to learn something about the RE technology validation process, just as we can use animals as models of people to learn something about how people respond to drugs. To abstract from whether we talk about drugs or RE technology, I will call the technology to be scaled up an *artifact*.

In what follows I present a number of observations of the process described in Section 2.1. The first observation is that the validation tasks in new drug development are divided into three stages: Conceptual validation, modeling, and field tests. In *conceptual validation* (corresponding to synthesis and purification), an artifact is tested by observing its behavior in a very artificial context such as a test tube. Most of the validation is done on paper and consists of computations, worked examples, mathematical proofs, informal arguments tested out with colleagues, etc. In the *modeling* stage (corresponding to animal testing and phase-1 clinical research), the artifact is tested out on a model. In drug development, these are animals first and healthy people next. There are important ethical constraints in both kinds of tests and the NDA process recognizes the need for an ethical review board at least when transitioning to tests with people. In the *field testing* stage (corresponding to phase-2 and phase-3 clinical research), real-world cases are used to test the artifact on. These real-world cases are treated as models of arbitrary population elements.

My second observation of drug validation is that what is validated is not the artifact but the artifact in a context, e.g. a drug in a body. Validation is the attempt to test the following prediction (Wieringa, 2009):

[Artifact × Context]will produce Effects.

Effects may change if the context changes, and so the artifact must be investigated in different contexts until it is clear in what range of contexts what range of effects is produced. For example, a new technique for eliciting requirements may be tested on its effects in small projects, large projects, embedded systems projects, information systems projects etc. and be found to be effective in some but not all of these contexts.

Third, when validating an artifact in context, researchers should not only aim at identifying the regular production of an effect in certain contexts, they should also aim to *explain* this effect in terms of underlying mechanisms. In drug research these are called the *mechanisms of action*. This term indicates the interactions by which a drug produces a pharmacological effect, including the binding of the drug to molecular targets, its effect on these targets, and the effect on biochemical pathways in the body. For example, caffeine has several mechanisms of action, two of which are that it antagonizes a biochemical compound (adenosine) that inhibits neurotransmitters, and that it increases the activity of neurotransmitters such as dopamine (Nehlig et al., 1992). These mechanisms explain why caffeine has a psychostimulant effect.

The concept of a mechanism of action is similar to that of a *principle of operation* used in engineering methodology (Vincenti, 1990), which is the top-level theory of the mechanism by which an artifact produces an effect in a context. For example, the principle of operation of an airplane is that by the shape of its wings, air above the wing flows faster relative to the wing than air below it, which according to Bernoulli's principle produces upward lift. But where the principle of operation is the highest-level view of how an artifact produces some effect in a context, a mechanism of action is the actual realization of this principle in the interactions between components of a low-level architectures of realized artifact in a real context. The principle of operation of an airplane explains why airplanes fly. The mechanism of action of a particular type of airplane consists of the detailed, low-level interactions among aircraft components and the surrounding air that actually produce the capability of this type of airplane to fly. The mechanisms of action exploit the Bernoulli principle, but do a lot more.

In RE too, mechanisms of action can be identified that explain observed effects. For example, Damian and Chisan (2006) describe the introduction of RE techniques in an organization and identify cross-disciplinary group meetings, and their interaction with other parts of the software engineering organization, as a mechanism that causes fewer defects, less rework, and improved effort estimates.

Validation research thus aims at making predictions of the form

[Artifact × Context] will produce Effects by Mechanisms.

We will call this a *design theory* (Wieringa et al., 2011).

When researchers design and validate an artifact, they start from an initial idea about the principle of operation of the artifact, which is a solution idea, but not yet an implemented and working solution. This states the top-level principle of operation. When scaling up an artifact to conditions of practice, this initial theory is tested and elaborated, until finally a street-tested architecture with mechanisms of action is delivered, that implements the top-level principle of operation.

The theory of an implemented artifact may be incomplete about the mechanisms that produce the effects, and in the extreme case be totally silent about them. For example, engineers may have found what the detailed structure and texture of a wing surface

- Design theory: [Artifact X Context] produces Effects by Mechanisms
- Value theory: [Effects X Stakeholders] produces Valuation.

Fig. 2. The structure of design theories and of value theories.

is that is most conducive to fuel efficiency, without understanding the precise mechanisms by which this happens. If mechanisms are not understood, a slightly different design or an existing design in a new, previously un-encountered context may fail for unknown reasons. For this reason, in the health sciences, evidence of regularity is not good enough to claim regular production of an effect: Knowledge of the underlying mechanisms is needed too (Russo and Williamson, 2007). In engineering, in the absence of knowledge of underlying mechanisms, safety risks are managed by testing design changes and sensitivity to context only in small steps (Petroski, 1994).

A fourth observation of the drug development process is that there is a second theory, that is stakeholder-related (Wieringa et al., 2011):

[Effects × Stakeholders] will produce Valuation.

I will call this a *value theory*. The theory states that various kinds of stakeholders who experience the effects will attach a positive, negative, or indifferent value to it.

The goal of drug research is not only to identify effects and mechanisms of an artifact in context, but also to identify the value of these effects for stakeholders. Stakeholders like some effects and dislike others. Effects that are liked are called "benefits" and effects that are disliked are often called "side-effects". Context properties that tend to produce effects that are disliked, are called "contra-indications" in drug research.

Finally, as all scientific theories, design theories as well as value theories are fallible. The researcher is not totally certain about them and must state the extent of his or her uncertainty. The uncertainty with which effects, benefits, costs and side-effects can be predicted, are called "risks".

All of these concepts are relevant for RE research too. For example, the use of mobile RE technology to elicit requirements has the benefit that user requirements may be more concrete, detailed and complete than is possible by other elicitation techniques. But that is not certain, and this is a risk (of a benefit not materializing). It may also have as side effect that the user may have the expectation that each and every need she enters, will be satisfied in the near future. This too is a risk (of a negative outcome). Also, mobile RE technology may result in huge amounts of textual and multimedia data that must be analyzed manually, which is a cost. In short, when validating RE technology, the RE researcher is not only interested in the effects of an artifact in context and the mechanisms that produce it, but also in the benefit, cost and risk of using this technology in some contexts (Fig. 2). Both kinds of theory are important, but in the rest of this paper, I will focus on the development and validation of design theories.

2.3. Inferences that support design theories

Looking once more at the drug development and review process in Fig. 1, we see that two kinds of inferences are used to generalize from experiments to the population of potential patients: Inductive generalization and analogic generalization. Inductive generalization is the statistical inference from a sample of test subjects to the population of subjects. Analogic generalization is the inference from models (such as animals, and healthy volunteers) to patients. This is represented in Fig. 3, where inductive generalization is the horizontal dimension and analogic generalization is the vertical dimension.

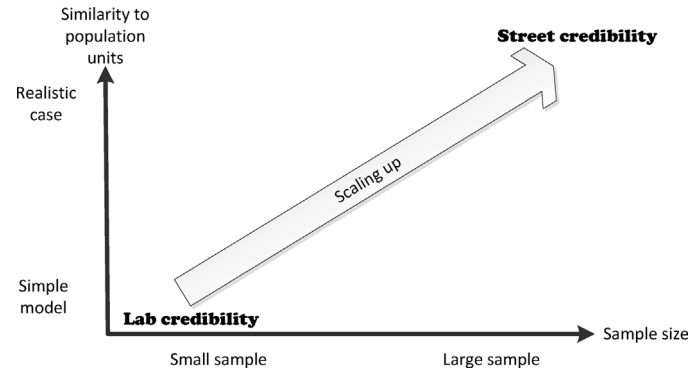

Fig. 3. Two kinds of inferences when scaling up to practice: inductive generalization from samples to population (horizontal) and generalization by analogy from experimental cases to real-world cases (vertical).

We discuss these generalizations in the next two paragraphs. Next, we discuss a third kind of inference, called abductive inference, that can be used to support analogic as well as inductive inference. Finally, we combine these inferences in two kinds of reasoning:

- In case-based reasoning, analogic generalization about cases is supported by abductive inference (vertical dimension of Fig. 3);
- In sample-based reasoning, inductive generalization about samples is supported by abductive inference (horizontal dimension of Fig. 3).

2.3.1. Inductive generalization

Inductive generalization is the generalization from samples to populations using statistical inference, such as statistical hypothesis testing or statistical parameter estimation (Hacking, 2001). Sample sizes in drug research start at about 30 elements, and increase to hundreds or even thousands of elements. The larger the sample, the larger the power of the experiment to discern small effects.

I call this kind of inference *inductive*. The term "induction" is given different meaning by different people, but here I follow Douven (2011) in calling an inference inductive if it is based purely on statistical data. In the context of this paper, this means that inductive inference is statistical inference, in which sample data are used to estimate a statistical population parameter or to test a statistical hypothesis about a population parameter. Inductive inference is the horizontal dimension in Fig. 3.

2.3.2. Generalization by analogy

The vertical dimension of Fig. 3 is *generalization by analogy* of the object of study (OoS) to the real-world population units to which the researcher wishes to generalize. The *object of study* has the structure

(model of the artifact) × (model of the context),

and is the entity studied by the researcher. See also Fig. 4. The model of the artifact is often an artifact prototype, and the model of the context can be a simulated context in the laboratory. In field research, the model of the context is a real-world context that stands as model for other real-world contexts. The treatment and measurement elements of Fig. 4 will be discussed later.

Generalization by analogy reasons about cases. For example, if in one agile development project performed for a small company, we have observed that the company lacked the resources to put a customer on-site, we may infer that in similar cases, a similar thing may happen. Each generalization by analogy reasons from one or more similar source cases to one or more similar target cases.

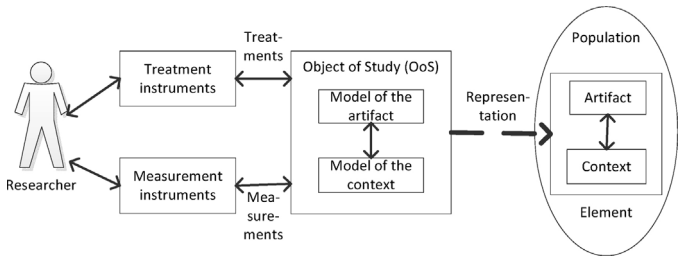

Fig. 4. The structure of validation research.

This contrasts with inductive generalization, which reasons from a sample of cases to the population of all cases. For example, if we have observed that in a random sample of 100 agile projects performed for small companies, 90% of the companies do not put a customer on-site, we may estimate from this a confidence interval for the proportion of the population in which no customer is put on-site.

To add more illustrations we discuss the role of analogic generalization in the New Drug Development process. In synthesis and purification, researchers build a prototype of the drug, interacting with some biochemical processes, that has sufficient similarity with processes in the human body to be able to draw some preliminary conclusion about the effect of the drug in the human body. In RE research, this would be similar to hand-testing a new technique, or formally proving some properties to show what the technique can do in an idealized context.

Next, drug researchers experiment with the drug in animals, used as natural models of the human body. As pointed out earlier, this would be analogous to the use of new RE techniques in student projects, which are then used as natural models of real-world projects with practitioners. There is detailed research in some fields of drug research to assess which animal species are valid models of humans with respect to which research questions, and for which research questions they are not (Willner, 1991). We find the same kind of "similarity research" in engineering research too. For example, to be able to reason from observations of a model in a wind tunnel to the behavior of airfoils in real flight, there must be a *theory of similarity* between wind-tunnel models and real-world flight (Vincenti, 1990). To my knowledge there is little research in this area in RE, but there has been some similarity research in software engineering that studies with respect to which research questions student behavior in student projects is similar or dissimilar to the behavior of professional software engineers in software projects (Höst et al., 2000; Runeson, 2003; Sjoberg et al., 2003; Svahnberg et al., 2008).

In the clinical phase, drug researchers start with healthy people, and then continue with ever larger samples of patients. In RE research this would correspond to using new technology first in pilot projects in companies with a mature RE process, continuing with ever larger samples of pilot projects in companies with low levels of RE maturity.

Generalization by analogy also includes reasoning by extreme cases, in which one case is known to be similar to other cases in some relevant aspects, but extremely different in another aspect. For example, from the observations that an RE technique is easy to understand and use by Master's students in software engineering, one might conclude that it will also be easy to understand and use by experienced software engineers. Master's students and software professionals are similar in some respects, but they are dissimilar in the extent of experience in software engineering that they have. Students are an extreme case w.r.t. extent of experience. The theory of similarity used to support this analogic inference is that increase in experience of otherwise similar subjects, preserves understandability and usability of a technique.

In general, generalization by analogy must be supported by some theory of similarity between the OoS and all population elements, that explains why a phenomenon observed in a model could lead to conclusions about population units. What theory is needed, depends on the question asked. Students may be good models of practitioners when validating effort estimation techniques but not when validating multi-stakeholder prioritization techniques (cf. the experiment by Höst et al., 2000).

2.3.3. Abductive inference

There is a third kind of inference used in drug validation research, called abductive, and that can be used to support both inductive and analogic generalization. *Abductive inference* is reasoning from observed phenomena to what is considered the best explanation of the phenomena (Douven, 2011). There are many kinds of abduction, and here I define one kind, that I call *mechanistic abduction*, in which observed phenomena are explained in terms of component-based mechanisms that produced them. I define a *component-based mechanism*, in turn, as a repeatable process in which system components interact to respond to a stimulus. This concept of mechanism is known in object-oriented software engineering, where a UML collaboration diagram represents a mechanism consisting of software objects that pass each other messages when responding to a stimulus (Cook and Daniels, 1994). But component-based mechanisms can occur in any kind of system, as we saw when we discussed the concept of mechanism of action of a drug. Component-based mechanisms are used to explain biological phenomena in terms of the interactions between cells and chemical substances, or the interactions between the organs of an organism (Bechtel and Richardson, 2010; Bechtel and Abrahamsen, 2005). In the social sciences, component-based mechanisms are used to explain social phenomena in terms of interactions between people, organizations, institutions and other social systems and their components (Bunge, 2004; Hedström and Ylikoski, 2010).

In RE too, component-based mechanisms can explain the effects of an RE technology in terms of interactions between components of a social, technical, physical, and digital systems. I already mentioned the mechanism identified by Damian and Chisan (2006), by which cross-disciplinary group meetings, and their interaction with other parts of the software engineering organization, resulted in fewer defects, less rework, and improved effort estimates. As another example, Seyff et al. (2010) identified two mechanisms that reduce the use of audio recording in mobile RE: Participants felt uncomfortable if they voice recorded their needs in a public place, and public places often contained too much background noise to do the recording.

To sum up, abductive inference is the identification of component-based mechanisms that explain effects. They complete the prediction

[Artifact $\times$ Context] will produce Effects.

with an explanation of the mechanisms by which the effects are produced. As indicated earlier, researchers will not always be able to explain all mechanisms of interaction between an artifact and a concrete, practical context. To the extent that less mechanisms are known, there is less confidence that statistical regularities in behavior are stable under changes in context.

2.3.4. Case-based and sample-based reasoning

Adding abductive inferences to analogic and inductive generalization, respectively, we get case-based reasoning (CBR) and sample-based reasoning (SBR) (Fig. 5).

Case-based reasoning is the vertical dimension of our diagram of scaling up (Fig. 3). It consists of two steps, namely abduction and generalization by analogy. In the first step, a single case is

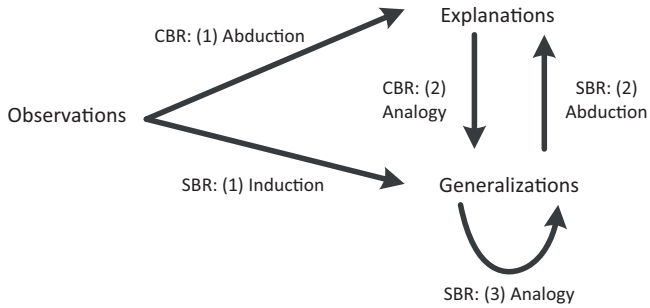

Fig. 5. Reasoning from observations to explanations and generalizations in case-based reasoning (CBR) and in sample-based reasoning (SBR).

analyzed to identify an architecture of the case in terms of its components and their interactions, so that this architecture may provide an explanation of observed effects in terms of component-based mechanisms. For example, if in a case of agile development performed by an independent developer for a small company, no client representative is on-site, then this can be explained by the limited resources of the small company. This is an abductive inference.

Next, in CBR we can generalize by analogy by postulating that in cases with the same or a similar architecture (independent developer doing agile development for a small company), the same effects will be produced by the same or similar mechanisms. The theory of similarity that supports the generalization here is that small companies have limited resources and will prefer to trust the developer rather than spend their scarce resources on agile requirements engineering. In CBR, the theory of similarity that supports the analogic generalization is stated in terms of a component-based mechanism.

Sample-based reasoning, the horizontal dimension in our diagram of scaling up (Fig. 3), is more complex. It consists of three steps (Fig. 5). From observations of a statistically meaningful sample of the population (e.g. a random sample of at least 30 elements), the researcher infers statistically that the population has some characteristics. For example, from an experiment with a sample of student projects, the researcher may be able to infer statistically that there is a correlation between the use of some requirements notation and the quality of requirements specifications in the population of student projects. This inference is fallible, because the sample may coincidentally show a pattern that is absent from almost all other samples from this population.

Second, the researcher may then list all possible causes of such a correlation, and be able to argue that the best explanation is that that the notation *caused* the difference in quality. This is an inference to the best explanation, i.e. an abduction. If the researcher can explain the postulated impact of notation on quality by some intermediate cognitive mechanism, that is postulated by a previously established theory, then this is a second, mechanistic abduction, that increases the support for the first one.

Third, the researcher may want to generalize the claim about the population further to similar populations, by analogy. For example, from a statistical generalization about student projects, the researcher may want to generalize further to the population of all real-world projects with junior software engineers, and justify this generalization by the similarity of the architecture and mechanisms of the student projects to the architecture and mechanisms of real-world projects with junior software engineers. This analogic generalization too may be supported by a mechanistic explanation, if the mechanism that explains the phenomenon in student projects, can also explain that phenomenon in professional software engineering projects.

Double support for causal claims, in statistical evidence provided by statistical difference-making experiments, and in independently verified mechanisms that can explain the causal relationships inferred from the statistical experiments, seems to be common practice in the health sciences (Russo and Williamson, 2007). Thus, sample-based reasoning and case-based reasoning have a useful supplementary relationship. After providing support for an inductive generalization about the effect of an artifact, the researcher may do some case studies, or some single-case mechanism experiments as described later, in an attempt to find and understand the mechanisms that produces this effect. Or, the other way around, after finding that a mechanism has produced an effect in a few cases, the researcher may do a statistical difference-making experiment to support the claim that this effect can be generalized statistically to the population. Thus, the two generalization dimensions in the diagram of scaling up (Fig. 3) must be traveled together.

2.4. Validity of inferences

All three kinds of inferences discussed are fallible, meaning that their conclusions could be false even if their premises are true. The researcher must therefore spell out the reasons that support the conclusion, and also summarize the reasons why the conclusions could be false after all. This is called a discussion of validity of the conclusions. Since “validity” suggests “justifiable” or even “truth”, this term is misleading. A less misleading term would have been “plausibility” or “support”. However, I will stick to the accepted terminology.

In Table 1 we can see that the three kinds of inferences correspond to three well-known kinds of validity. *Conclusion validity* is the support for the conclusion of a statistical inference. Threats to conclusion validity include low power, small sample, non-random sample, non-random allocation, violation of assumptions of statistical algorithms, etc. Note that even if conclusion validity would be sufficiently well argued, it still possible that the experiment is one of the 5% experiments that shows a statistically significant difference by chance, i.e. without there being a mechanism that produces the difference.

Internal validity is the support for an explanation of a phenomenon by causal mechanisms that produced the phenomenon. A major threat to internal validity is that outcomes of an experiment may not only be explained by a mechanism that leads from treatment to outcome, but by other mechanisms too. For example, if the OoS contains people, then history, maturation, and attrition may influence the outcome, in addition to the influence of instruments, tests, the experimenter, semantic ambiguities, etc. in the experiment (Shadish et al., 2002, page 54 ff.). For the reader of a research report to assess the support for the abductive inference that the observed outcome is produced by some mechanisms, alternative explanations must be listed explicitly.

External validity appears in two flavors in Fig. 5: in case-based reasoning and in sample-based reasoning. In *case-based reasoning*, external validity is the validity of the analogic inference from a single-case explanation to all similar cases. For example, a mechanism observed in [(artifact prototype) × (simulated context)] in the laboratory is generalized to all [artifact × context] cases in the real world. In *sample-based reasoning*, external validity is the validity of the analogy of one population to another population. For example, a conclusion about the population of student projects is generalized to a conclusion about the population of real-world projects. In both flavors, external validity is the validity of the inference from the studied OoS to all similar cases in the real world. As observed by Gigerenzer (Gigerenzer, 1984), determining external validity is an empirical question. If conclusions from an experiment in context A are generalized, fallibly, to context B, then one can test this

Table 1
The three kinds of inference and some validity considerations.

Inductive inference	Estimation of a population parameter, or decision about a statistical hypothesis about the population, based on observations of a sample.	Conclusion validity	Are the assumptions of the statistical algorithms satisfied? Random sample? Homogeneous sample? Random allocation? Statistical power and effect size? Reliable measures? Logical errors in reasoning from sample statistics to population hypotheses? Etc.
Abductive inference	Explaining a phenomenon by identifying the causal mechanisms that produced it.	Internal validity	Are there alternative explanations? Is there a common cause that could explain the phenomena? Can the context of the experiment, the behavior of the experimenter, or phenomena in the sample of subject explain the outcome of the experiment? Etc.
Analogic inference	Concluding that a target will have the same properties as a source (the experiment) because of some similarity between them.	External validity	Is there a theory of similarity? Does the theory of similarity justify the conclusions? Are the mechanisms in the target the same as those in the source? Are there other mechanisms that could interfere with the mechanism of interest? Is the effect context-sensitive? Etc.

generalization by repeating the experiment in context B. This is in fact what is done when scaling up from the lab to the real world.

Threats to external validity are sensitivity of the effects of an artifact to the context in which it is used, dissimilarity of the treatment used in the lab to treatments used in practice, interference of other mechanisms with the mechanism of interest, absence of a theory of similarity that could justify the generalization, etc. Shadish et al. (2002, pages 86 ff.) and Wohlin et al. (2012, page 110) provide detailed discussions.

3. Methods for validation research

We will discuss the empirical validation methods using the structure of Fig. 4. We have used this structure earlier to make a checklist for empirical research reports (Wieringa et al., 2012). The researcher uses an object of study (OoS) to represent elements of the population, where in our case the population elements have the structure [artifact × context]. Therefore, the OoS has this structure too, consisting of a model of the artifact and a model of the context. The OoS is a *model* of an arbitrary population element in the sense that it is similar to population elements, and can be studied by the researcher to learn something about population elements (Apostel, 1961). An example would be an OoS that consists of a prototype of a software product, interacting with a simulation of a problem context; or an RE technique (the artifact) interacting with a student project (the context).

In statistical research, the researcher studies a sample of OoS's of statistically meaningful size. In case research, the researcher studies a small sample or even a single OoS.

In experimental research, the researcher applies a treatment to an OoS and then measures what happens. In observational research, the researcher measure what happens, but does not apply a treatment. Measurement as well as treatment usually require instruments.

In the diagram of Fig. 4, all interactions are bidirectional: One cannot treat an OoS without the OoS exerting some influence on the treatment instrument, and one cannot measure an OoS without exerting some influence on the OoS.

The concept of treatment needs some explanation. So far we have taken a *component-based view* of the world, in which the world is modeled as a hierarchy of systems, that contain subsystems, that contain sub-subsystems, etc. Thus, the population consists of artifacts interacting with a context, and research has a structure of components as shown in Fig. 4. In this view, a treatment is the insertion of a component in a context. For example, a doctor treats a patient (the context) by given them a drug (the artifact), and a consultant helps a software engineering organization (the context) by inserting an improved RE technique (the artifact).

Note that the artifact not only consists of a product, for example a drug or an RE tool, but also of a process, for example the protocol for taking the drug or the procedure by which to use the tool. The experimental treatment then consists of making this product available and giving an instruction in the process.

We can describe the same experimental situation also in the more traditional view, in which a treatment is setting the level of an independent variable. This is a more abstract, variable-based view of experiments, that will be convenient to use in Section 3.4 on statistical experiments. Until then, it is more illuminating if we use the component-based view of Fig. 4. Table 2 lists four groups of validation research methods that we will discuss in the following sections.

3.1. Expert opinion

In the conceptual stage of validation, before the artifact is tested on models or in the field, the researcher can elicit the opinion of experts about the possible usability and usefulness of the artifact. This is observational empirical research, because the researcher does not intervene in an object of study. The researcher elicits opinions. It is also not a statistical survey with the aim to estimate the distribution of opinions in the entire population of experts. Rather, it is an attempt to get early information about expected usability

Table 2
Validation research methods.

Methods	Examples
Researching expert opinion	• Eliciting expert opinion using interviews, • Questionnaires, or • Focus groups
Single-case mechanism experiments	• Testing an artifact prototype on a simple example in the lab • Testing an artifact prototype on a realistic example in the lab • Testing an artifact prototype on a realistic example in the field
Technical action research	• Using an artifact prototype to help a client • Teaching the use of an artifact prototype to a client by which they can solve some of their problems
Statistical difference-making experiments	• Comparing the effect of prototypes of two or more artifacts on a sample of simulated contexts in the lab • Comparing the effect of prototypes of two or more artifacts on a sample of contexts in the field

Please cite this article in press as: Wieringa, R., Empirical research methods for technology validation: Scaling up to practice. J. Syst. Software (2013), http://dx.doi.org/10.1016/j.jss.2013.11.1097

and usefulness of the artifact in real-world contexts. Thus, statistically meaningful sample sizes are not needed; useful opinions are needed.

In terms of Fig. 4, the population is *not* the set of all possible experts but the set of all possible [artifact × context] elements. So the object of study is not the expert either. Rather, the expert is an instrument to measure an imaginary OoS, namely a mental image that the expert has formed of real-world [artifact × context] elements. This is an unreliable instrument, but one that nevertheless can give useful information.

Positive or uncritical opinions of experts are not very useful, because experts may be motivated by the desire to finish the interview quickly, or to be nice to the researcher. Negative or critical opinions on the other hand are very useful, especially if the expert can indicate which element of the artifact design would not be usable or useful in which context, and why.

Example 1. Al-Emran et al. (2010) present an optimization method for product release planning. Input to the optimization method is a set of product release plans, consisting of a sequence of features to be implemented in subsequent releases of a product. The optimization method then finds the plan that is most robust, in terms of time-to-market, resource assignment, and task schedule, with respect to differences in task workload and developer productivity. That is, it selects the release strategy that is least influenced by differences in workload and productivity.

The researchers tested the optimization method among others by submitting it to experts, asking their opinion about it. The researchers sent a questionnaire about the method to 25 product development experts and received 13 responses. Many responses were uninformative in that the respondents thought the method was usable and useful. Some respondents, though, complained that using this optimization decreased their understanding of the release planning process, and others complained that they required more justification of the result before they would adopt the recommendation. These remarks point at potential improvement needs of the method.

Collecting expert opinion combines the two dimensions of scaling up. Experts imagine a sample of cases (informal sample-based reasoning) and imagine what mechanisms would occur in each of those cases (informal case-based reasoning). Because of the informality of their reasoning, their opinions must be treated with caution, but nevertheless they must be treated seriously.

3.2. Single-case mechanism experiments

I use the term *single-case mechanism experiment* to indicate experiments in which the researcher investigates one OoS in order to test the effect of some mechanism that the researcher believes to be present in the OoS. Software engineers do this when they test a software prototype by feeding it input scenarios that represent possible scenarios in the intended context of use. Aeronautical engineers do it when they test an airfoil in a wind tunnel.

I give some examples from RE research before I discuss the logic of single-case mechanism experiments.

It is not my purpose here to judge the quality of the analogic generalizations or of explanations given by authors, but merely to illustrate what the role of analogy and of mechanistic explanations in validation experiments is.

Example 2. Gacitua et al. (2011) propose a new algorithm for the identification of single- and multi-word abstractions in requirements documents, and describe an experiment in which they compare the performance of this algorithm with human judgment. The algorithm compares the frequency of a term in a document with its frequency in a reference document of the language used in the requirements document, such as a corpus of standard English. Terms that are rare in the reference document, but occur frequently in the requirements document are likely to indicate important concepts in the domain.

To test this algorithm, the authors selected a book on a technical domain, and used its body as if it were a requirements document to analyze. The index of the book was used as a reference list of domain concepts. Thus, the artifact to be tested is the algorithm, and the book and its index is the context; both make up the OoS. The treatment in this experiment is the request to identify abstractions in a book. The measurement is the measurement of recall and precision with respect to the index terms.

Concerning external validity, the authors argue that the book's domain is similar to the domain of RE documents, that the size of the document is similar to documents in RE projects, and that the concept abstraction scenario similar to that of a requirements engineer who has to familiarize herself with a new domain. Also, they argue that the hierarchical structure of the index is representative of the structure of multi-word terms in the intended population.

Internal validity is the question whether the mechanisms built into the artifact explain the observed effects. In this experiment, the frequency-based mechanism yielded low recall and precision. The authors' explanation is that the identification of abstractions by people does not take place by a frequency-based method, and that frequency in general is not a sufficiently powerful mechanism to identify abstractions.

Example 3. Seyff et al. (2010) tested a tool for mobile requirements engineering in the field. They gave mobile phones running the tool to nine subjects, who used it for a few days to gather requirements for a system that supports daily commuting, and requirements for a system that supports shopping activities. The requirements were stated in text or audio. After the experiment, subjects were debriefed, and researchers transcribed recorded needs into system requirements.

In this example, the OoS consists of an artifact prototype, interacting with a realistic context. The context consists of the mobile phone on which the prototype runs, the users using the prototype, and the environment in which the users move. The treatment consists of the instructions to the users to use the tool for two RE purposes. The treatment instrument is the instruction session in which the users were instructed. The measurements consist of the data (text or audio) entered by the user as well as the answers of the users to researchers' questions in the debriefing session.

The similarity between these OoS's and the envisaged population of future real-world mobile RE processes is threatened by potential differences between future tools and the one used in this experiment, and possibly also by differences in elicitation methods. Remember that the artifact in this example consists of a mobile RE tool plus the process for using it.

The researchers made a mechanism (a mobile RE tool) available to users to test if it produced the expected effects (recorded contextual end-user needs). The mechanism had the expected effect in all nine investigated cases. A threat to the validity of this observation is that the subjects may have wanted to be nice to the researchers, which would be a factor co-producing the expected effect. Other users, without a friendly disposition to the researchers, may have failed to produce the effects when interacting with the tool.

These experiments allow analogic inferences from the OoS to the population, and can be placed along the vertical dimension of our diagram of scaling up (Fig. 3). As we saw, analogic generalizations must be supported by a theory of similarity, that explains why an observation on the source of the analogy can lead to a conclusion about the target of the analogy. In the component-based view of the world that we take, the similarity between source and target of an analogy must be architectural, and the theory of

similarity must indicate some component-based mechanism that produced a response in the experiment, and can produce a similar response in the real-world cases of the population. Hence the name "mechanism-based experiments". These experiments do not use statistical inference to support claims about the population, but they use a theory about mechanisms to support claims about the population.

We distinguish mechanisms in the artifact from mechanisms in the context.

- The artifact is by design a collection of component-based mechanisms that responds to input. Part of software testing consist of validation whether these mechanisms, if implemented correctly, indeed have the desired effects. This may lead to surprises, in the sense that unexpected phenomena may turn up (e.g. bugs) that are the results of unexpected mechanisms in an implementation. In general, in algorithm validation, there may be unexpected mechanisms in the program because our ability to program may exceed our ability to understand what we programmed.

  This is less likely to happen when testing a method. Step-by-step methods such as the Rational Unified Process build up their results in a simple manner, by instructions of the form "bring about result *X*", which, if performed correctly, leads to the creation of result *X*. If these methods are performed by capable software engineers in an ideal context, we usually do not run against unexpected mechanisms in the method itself, because methods are relatively simple step-by-step procedures.
- Once a method has been shown to be usable by the researcher and his or her students, the important research question is whether it still works under conditions of practice, i.e. in the real world. In real-world contexts, there may be components or mechanisms that impact the production of the desired result of a method step in unexpected ways. An example of an unexpected mechanism in mobile RE is the tendency of users to be very brief in the textual specification of their needs when they were in a physically confined space, and enter explanations by audio later. This may make needs analysis more time consuming, which in turn may reduce the timeliness and cost-effectiveness of the requirements specification. The researchers may use this information to change the method.

3.3. Technical action research

Technical action research (TAR) is a case-based mechanism experiment too, but I list it separately because it is also something else: It is a real-world consultancy project (Wieringa and Morali, 2012). In a TAR project the researcher uses an artifact in a real-world project to help a client, or gives the artifact to others to use them in a real-world project (Engelsman and Wieringa, 2012), and uses this experience to learn about the robustness of the intended effects and the mechanisms that bring them about, in uncontrolled conditions of practice.

Example 4. Morali and Wieringa (2010) describe a method to assess confidentiality risks when outsourcing the management of IT systems. They then describe how Morali used this method to actually assess the confidentiality risks in the outsourcing relationship between a large manufacturing company and a large outsourcing service provider.

In this example, the artifact is a new risk assessment method, and the context consists of Morali applying this method to a risk assessment problem in a large company. Morali played a dual role as researcher giving an instruction how to use an artifact to an OoS in which she herself was the user of the artifact. The measurements taken consisted of all intermediate working documents of the project, plus the diary of Morali in her role as user of the method.

The similarity of this TAR project to the population of all such risk assessment projects is that a confidentiality risk in an IT management outsourcing situation is assessed. There is also a dissimilarity, which is that in most projects in this population, Morali will not be the one doing the risk assessment. This is a threat to external validity that must be dealt with by repeating TAR projects like this with other researchers.

Internal validity is the question whether the method indeed delivered its expected results, and whether any mechanisms in the context influenced this. The method did deliver its expected results, but only repeated TAR projects can show whether or not this is the due to the method only or also to the user of the method (Morali), the quality of the documentation available in the company, etc.

TAR is a special kind of mechanism-based experiment, and in the process of scaling up they take the researcher closer to the real world in the vertical dimension of the process of scaling up (Fig. 3). The inferences in TAR are of the same kind as those in other case-based mechanism experiments. Events in the case are explained in terms of mechanisms, and any generalization to the population is supported by a theory that says that these mechanisms can occur in population elements too. However, generalizations from TAR projects have an additional threat to validity, because the researcher may have contributed positively to the observed events in a way that cannot be replicated.

TAR is useful as a final validation stage before transferring a technology to practice, because it is closer to real-world practice than other case-based mechanism experiments. A single TAR project is not enough to justify the claim that an artifact is applicable in the entire target population of possible projects. But it does justify the claim that the artifact is usable and useful in some real-world projects, and it can provide useful information to the researcher for further improving the artifact.

3.4. Statistical difference-making experiments

A *statistical experiment* is an experiment with a sample of OoS's to infer a statistical property of the population. For example, it may estimate the population mean of a variable, with a confidence interval, from observations of the sample mean. Or it may test a statistical hypothesis about the population mean by observations from a sample.

In contrast to case-based mechanism experiments, the sample size is relevant, because as indicated in Fig. 5, inference is sample-based, not case-based. Statistical experiments support inductive inferences about samples, which is the horizontal dimension of scaling up (Fig. 3). They do not require a theory of mechanisms to generalize inductively to a population, but as we have seen in Section 2.3, providing such a theory does give additional support to an inductive generalization, because it would decrease the likelihood that the inductive generalization is based on a coincidental pattern in the data.

To describe statistical experiments we need to switch from the *component-based view* that we have taken up till now, in which the world consists of components and interactions, to a *variable-based view* of the world, in which the world consists of variables and relationships. Any description of the world in terms of components and interactions can be replaced by a more abstract description in terms of variables and relationships. In this variable-based view, a treatment consists of setting the value of an independent variable, and effects are measured by measuring the values of dependent variables. If there are two treatments, often one is called the "treatment" and the other the "control", dividing the sample into a treatment group and a control group.

Inductive inference from sample to population can take place in a variety of ways, depending on how OoS's were selected (sampling) and how treatments were allocated to OoS's. In a *randomized controlled trial* (RCT), the sample is random and the allocation of treatments to sample elements is random too (Shadish et al., 2002; Sedgwick, 2011). This makes it possible to use the central limit theorem to support the inductive inference from sample to population. There are two ways to do this, by hypothesis testing and by confidence intervals (Hacking, 2001; Wonnacott and Wonnacott, 1990). In *statistical hypothesis testing* the experimenter may observe a difference in the sample, that would be very unlikely (probability less than 5%) to occur if a difference did not exist in the population. The researcher will then infer that, plausibly, the difference exists in the population. In the estimation of *confidence intervals*, the experimenter may estimate a population difference by the sample difference using a 95% confidence interval around the sample mean. This estimation may be right or wrong, but if she follows this estimation rule always, she will be wrong in the long run in only about 5% of the inferences (Hacking, 2001).

Having inductively (and fallibly) inferred that there is a statistical correlation between independent and dependent variable in the population, the experimenter tries to abduce a causal explanation of this difference, and does this by trying to exclude any other possible cause other than the difference between treatments. Random sampling and random allocation can only introduce chance fluctuations that disappear on the average in the long run. But after allocation, treatments must be applied, and outcomes measured, and so the experimenter must also check whether application or measurement, or any other event during the experiment, could have contributed to the measured difference. This is all part of the discussion of internal validity summarized in Table 2.

If all these alternative causes are excluded, the difference between treatments is the only remaining possible cause of the observed difference in dependent variables. This is an abductive inference. Note that random sampling and allocation is used both in the inductive inference step, where it facilitates application of the central limit theorem, and in the abductive inference step, where it facilitates the exclusion of other causes than the treatment.

Random sampling is difficult to achieve in practice, so we find many *quasi-experiments* in software engineering and elsewhere (Kampenes et al., 2009; Shadish et al., 2002; Sjoberg et al., 2005), in which sampling is not random or allocation of treatments to elements is not random. For example, subjects may self-select into treatment or control groups, or the researcher may allocate treatments to elements according to a property of the elements. Quasi-experiments cannot use the mathematical techniques based on the central limit theorem for their statistical inference, but there are other reasoning techniques that can be used for statistical inference in quasi-experiments (Shadish et al., 2002).

RCTs and quasi-experiments both take a so-called difference-making view on causality, which is why I call them here *difference-making experiments*. In this view, variable X has a causal influence on variable Y if X makes a difference to Y. That is, if X had a different value, with all other things being equal, then the value of Y would be different as well (Holland, 1986; Woodward, 2003).

For example, suppose in an RCT, a sample of projects using programming method A performed better on the average than a sample of projects solving the same problems using method B, and suppose that this difference is statistically significant, i.e. it is unlikely to be observed in a sample if it would not exist in the population. So it is unlikely that the difference is the result of chance alone. So the researcher is justified to look for a cause. There may be many causes for the difference, including the available resources to the projects, the competence of project personnel, and the difference between methods A and B. If the researcher can rule out all causes other than the difference between A and B, then the statistical difference supports the claim that the difference between methods A and B is the cause of the difference in project performance.

Example 5. Prechelt et al. (2002) describe an experiment to compare the difference between maintenance tasks done on programs where design patterns were described in comment lines, and maintenance tasks done on programs where design patterns were not commented. The programs were identical except for the presence of so-called Pattern Comment Lines (PCLs).

The artifact is here the presence of PCLs and the context is the program, maintenance task, and maintainer. The subjects self-selected into the samples, which makes it hard to know which hypothetical population they are a random sample of, but we will assume that it consists of computer science students performing maintenance tasks on programs of similar size and complexity as those used in the experiment. The treatment is the instruction to perform maintenance tasks. Treatments were allocated randomly to subjects. The measured variables were task completion time and correctness of result.

The researchers found a slight improvement of task time and result correctness when PCLs were present, that was statistically significant. This means that there is a low probability that this observation would be made in the sample, if this improvement would be absent from the population. This supports the inductive generalization that an improvement exists in the population of all programs of the same size and complexity being maintained by students. The authors discuss possible causes for this improvement other than the presence of PCLs, and conclude that there is no evidence that there are other causes (Prechelt et al., 2002, page 599, Threats to internal validity).

They additionally identify a cognitive mechanism that could be responsible for this causal relationship (abductive inference, Fig. 5). This mechanism is postulated by a theory, formulated by several researchers earlier, that program comprehension works by the formation and validation of hypotheses, of which the efficiency is greatly enhanced by beacons, which are hints about familiar kinds of structures (Prechelt et al., 2002, page 596). PCLs are such beacons. This could explain the causal influence of PCLs maintainability. It increases the support for the claim that the observed improvement is systematic rather than a coincidental event.

The authors are reluctant to generalize, by analogy, from the population of experimental maintenance situations similar to this experiment, to the population of real maintenance tasks (Prechelt et al., 2002, page 599). However, they do reason that, if PCLs had an improvement effect for relatively small well-commented programs, they might have an even better effect on large ill-commented programs (Prechelt et al., 2002, page 604). This is reasoning by analogy.

In an interesting aside, the authors observe that experiments comparing different syntactic forms to express the same meaning all have the methodological problem that the two forms rarely have the exact same meaning. The authors give general advice about a methodologically sound setup of such experiments. This is a case-based reasoning by analogy (Fig. 5), in which their experiment is an example for other, similar experiments.

Statistical difference-making experiments support reasoning along the horizontal dimension of our diagram of scaling up (Fig. 3). We see in this example first an inductive inference and then two abductions. First, the statistical correlation between two variables is inductively inferred to exist in a population, based on observations in the sample. This is inductive inference. Next, it is argued that this statistical correlation between independent and dependent variable is a causal relationship from independent to dependent variable, by ruling out all other possible causes than the difference in treatments (abduction 1). Third, this causal relationship was explained by a cognitive mechanism postulated by

a previously established theory (abduction 2). This increases confidence in the causal conclusions that the authors drew from the experiment.

4. Related work

The reasoning schema "[Artifact × Context] → Effect by Mechanisms" has been proposed in slightly different forms in social science (Pawson and Tilley, 1997) and in management science (Van Aken, 2004). It has some similarity with the satisfaction argument as proposed by Jackson in software engineering (Jackson, 2000). Wieringa (2003) calls it the systems engineering argument, because it shows how a component must interact with other components to produce desired behavior of a composite system. It is simpler than the structure for design theories proposed by Gregor and Jones (2007). More discussion is provided elsewhere (Wieringa et al., 2011).

Douven (2011) gives a convenient introduction to abductive reasoning, also called "reasoning to the best explanation". Mechanistic abduction is similar to theoretical model abduction as discussed by Schurz (2008).

The concept of mechanism has been proposed by philosophers who analyzed the structure of explanation in the physical, biological, and social sciences (Glennan, 1996; Machamer et al., 2000). It has been adopted as an explanatory construct in biology (Bechtel and Richardson, 2010; Bechtel and Abrahamsen, 2005) and in the social sciences (Bunge, 2004; Hedström and Ylikoski, 2010; Elster, 1989). All of these authors have slightly differing concepts of mechanism. Illari and Williamson present a survey and unification (McKay Illari and Williamson, 2012), which is very similar to the concept that I have used here.

There is a huge literature on causality and I cannot even begin to cite the relevant literature here. There are two views, one that causality is difference-making, the other that a causal relationship is a mechanism, and within each view there are several points of view. For example, the Bayesian theory of Pearl is an example of a difference-making view, described in a book (Pearl, 2009) and summarized in a paper (Pearl, 2009b). Holland (1986) is an older, exceptionally clear exposition of the difference-making view, staying within the framework of frequency-based statistics. Williamson (2011) surveys some mechanistic theories, and compares them with difference-making views.

Generalization by analogy as discussed here is one form of *analytic induction*, propagated by Yin as the way to generalize from cases (Yin, 2003), but actually originating from the sociologist Znaniecki (1968). The clearest and most accessible description of analytic induction is given by Robinson (1951): The researcher (1) roughly defines a class of phenomena and (2) formulates a hypothesis about a mechanism that is postulated to occur in these phenomena. This is our theory of similarity, and we have only considered the case where the theory describes a structure of interacting components that implement a mechanism. Next, a single case that satisfies the definition is investigated. If observations falsify the hypothesis, then either the definition is refined to exclude the case at hand, or the hypothesis is reformulated to match the observations. After investigating a number of cases, the definition and hypothesis may reach a stable state. The researcher then generalizes by claiming that all similar cases contain similar mechanisms, which will produce similar effects. This kind of case-based reasoning moves us upward in the diagram of generalization (Fig. 3). Znaniecki lists a few historical examples from biology, physics and sociology where this kind of reasoning was followed (Znaniecki, 1968, pages 236–237).

Zelkowitz and Wallace (1998) presented a survey of empirical validation methods in software engineering that I compare in Table 3 with the list in Table 2. In the terminology of this paper, their list contains validation methods but some other kinds of methods too.

Table 3
Validation methods identified by Zelkowitz and Wallace (1998) and by Glass et al. (2001).

This paper	Zelkowitz and Wallace (1998)	Glass et al. (2001)
Validation research methods		
• Expert opinion		
• Single-case mechanism experiment	• Simulation	• Field experiment
	• Dynamic analysis	• Laboratory experiment – Software • Simulation
• Technical action research	• Case study	• Action research
• Statistical difference-making experiment	• Replicated experiment	• Field experiment
	• Synthetic environment experiment	• Laboratory experiment – human subjects
Other research methods		
• Observational case study	• Case study	• Case study
	• Field study	• Field study
• Meta-research method	• Literature search	• Literature review/analysis
Measurement methods		
• Methods to collect data	• Project monitoring • Legacy data • Lessons learned	• Ethnography
Inference techniques		
• Techniques to infer information from data	• Static analysis	• Data analysis
		• Grounded theory • Hermeneutics • Protocol analysis

Their *assertion* method has been omitted because, as they also point out, it is not a research method. It is an experimental use of a new technology by the developer in the laboratory. *Simulation* is executing a product in a simulated environment. This is a single-case mechanism experiment because the product is an implemented mechanism to be tested. *Dynamic analysis* is the execution of a product under controlled conditions, similar to simulation but not aimed at simulating real-world environments. It is a single-case mechanism experiment too.

A *case study* could be the use of a new technology in an industrial project (Zelkowitz and Wallace, 1998, page 26), in which case we classify it as a technical action research project, or an observational study of a project (Zelkowitz and Wallace, 1998, page 25), in which case we classify it as an observational case study. Case studies therefore appear twice in Table 2.

Replicated experiments and *synthetic environment experiments* are statistical comparison of groups of projects, where in different groups, a task is performed differently. These are statistical difference-making experiments, performed in the field or in the lab.

The difference between observational *case studies* and *field studies* defined by Zelkowitz and Wallace (1998, page 26) is that a case study is intrusive where a field study is not. They are both classified as observational case studies in the terminology of this paper, because in both, the research method is observational and influence of the researcher on the object of study is to be minimized. Observational studies are, in the terminology of this

paper, suitable as methods for evaluation studies of implemented technology, but not as research methods for validating new technology not yet transferred to the market.

Literature search is part of any research but may be expanded into a full-blown research method, also called a systematic literature review (Kitchenham, 2004).

Project monitoring is the collection, by the researcher, of data produced during a project and *legacy data* is the collection of documents such as source code, specifications, and test plans after the project is finished. For a full-blown research method, we need a design of the way the researcher will interact with the object of study, including measurement methods and any experimental intervention, and inference design. In the terminology of this paper, project monitoring and legacy data as described by Zelkowitz and Wallace, are measurement methods.

Lessons learned is the collection and analysis of lessons learned documents from projects. In Table 2 this is classified as a measurement method too but because it also contains analysis, we could also classify it as a form of observational field study.

In *static analysis* the completed product is investigated, for example to analyze its complexity. It is similar to the study of legacy data but it is here classified as an inference technique because it refers to a collection of analysis methods.

Glass et al. (2001) list the empirical research methods shown in the third column of Table 3. Non-empirical methods such as *conceptual analysis* and *mathematical proof* have been omitted, and design activities, viz. *concept implementation* and *instrument development* have been omitted too. Since it is not clear from the description by Glass et al. whether experiments are of the single-case mechanism kind or of the statistical difference-making kind, they are classified as both. I discuss the new entries in this column.

Ethnography is the detailed collection and description of daily events in a social group, without analysis, which is classified here as a measurement method. *Grounded theory* is the analysis of textual data produced by people, to extract the theories held by these people (Strauss and Corbin, 1998). I consider this to be a descriptive analysis method. *Hermeneutics* is the phenomenon that to interpret human behavior, you have to understand their cultural and conceptual framework, but the only way to understand their cultural and conceptual framework is to interpret their behavior. This leads to an inference strategy in which the researcher iterates over updating his or her conceptual framework and interpreting human behavior in that framework. *Protocol analysis* is the analysis of thinking-aloud protocols, useful for cognitive psychology. It is a data analysis method.

Table 3 shows that these lists of software engineering research methods are mutually consistent and can be integrated in my framework for validation research methods, and extend it with other methods. The overview is not complete, happily, as new methods and instruments for research keep being developed.

5. Summary and conclusion

Empirical validation of technology before it is transferred to practice requires investigating the effects of the interaction of the artifact with its context, and explaining these effects by means of the underlying mechanisms that produces these effects. Scaling up to practice thus produces a design theory of the form "[Artifact × Context] produces Effects by Mechanisms".

Producing support for such a theory involves two kinds of inferences, along the vertical and horizontal dimensions of the process of scaling up. *Analogic inferences* in the vertical dimension reason from case-based mechanism experiments to real-world instances of [Artifact × Context], and *statistical inferences* along the horizontal dimension reason from observed sample behavior to the population of all possible instances of [Artifact × Context]. Both inferences are supported by *abductive inferences*, that postulate mechanism-based explanations of cause-effects influences. Mechanism-based explanations refer to the components of [Artifact × Context] and their interactions.

We discussed the following research methods to validate artifacts:

- Expert opinion, in which experts reason informally about samples (horizontally) and mechanisms (vertically), which provides an initial sanity check of an artifact design;
- Single-case mechanism experiments, in which the researcher reasons vertically about mechanisms and their effects in increasingly realistic artifacts in increasingly realistic contexts;
- Technical action research, in which the researcher reasons vertically about mechanisms and their effects when an artifact is applied in a real-world project to help a client;
- Statistical difference-making experiments, in which the researcher reasons horizontally from effects observed in samples to effects inferred in populations.

These methods can be used with measurement instruments and data analysis methods known from software engineering and elsewhere.

This paper has given some examples of use of these research methods, but this is just one step on the way to scaling up these methods to empirical RE research. Increasing use of these methods will teach us more about the usability and usefulness of these research methods in empirical validation of RE technology.

Acknowledgements

This paper benefitted from comments by Vincenzo Gervasi and Walter Tichy. I would like to thank the anonymous reviewers of this paper for their constructive critique.

References

Al-Emran, A., Pfahl, D., Ruhe, G., 2010. Decision support for product release planning based on robustness analysis. In: Proceedings of the 18th IEEE International Requirements Engineering Conference (RE 2010), IEEE Computer Society, Sydney, Australia, pp. 157–166.
Apostel, L., 1961. Towards a formal study of models in the non-formal sciences. In: Freudenthal, H. (Ed.), The Concept and Role of the Model in the Mathematical and the Natural and Social Sciences. Reidel, Dordrecht, The Netherlands, pp. 1–37.
Babbie, E., 2007. The Practice of Social Research, 11th ed. Thomson Wadsworth, Belmont, USA.
Bechtel, W., Abrahamsen, A., 2005. Explanation: a mechanistic alternative. Studies in the History and Philosophy of Biological and Biomedical Sciences 36, 421–441.
Bechtel, W., Richardson, R., 2010. Discovering Complexity: Decomposition and Localization as Strategies in Scientific Research. MIT Press, Cambridge, Massachusetts (Reissue of the 1993 edition with a new introduction).
Bunge, M., 2004. How does it work? The search for explanatory mechanisms. Philosophy of the Social Sciences 34, 182–210.
Constant, E., 1980. The Origins of the Turbojet Revolution. Johns Hopkins, Baltimore.
Cook, S., Daniels, J., 1994. Designing Object Systems: Object-Oriented Modelling with Syntropy. Prentice-Hall, Upper Saddle River, New Jersey.
Cowan, C., 2002. The process of evaluating and regulating a new drug: phases of a drug study. AANA Journal 70, 385–390.
Damian, D., Chisan, J., 2006. An empirical study of the complex relationships between requirements engineering processes and other processes that lead to payoffs in productivity, quality and risk management. IEEE Transactions on Software Engineering 32, 433–453.
Davis, A., Hickey, A., 2004. A new paradigm for planning and evaluating requirements engineering research. In: 2nd International Workshop on Comparative Evaluation in Requirements Engineering, pp. 7–16.
Douven, I., 2011. In: Zalta, A. (Ed.), The Stanford Encyclopedia of Philosophy (Spring 2011 Edition). http://plato.stanford.edu/archives/spr2011/entries/abduction/
Elster, J., 1989. Nuts and Bolts for the Social Sciences. Cambridge University Press, Cambridge, UK.
Engelsman, W., Wieringa, R.J., 2012. Goal-oriented requirements engineering and enterprise architecture: two case studies and some lessons learned. In: Requirements Engineering: Foundation for Software Quality (REFSQ 2012), Essen, Germany, pp. 306–320 (volume 7195 of Lecture notes in computer science, Springer).

Gacitua, R., Sawyer, P., Gervasi, V., 2011. Relevance-based abstraction identification: technique and evaluation. Requirements Engineering 16, 251–265.
Gigerenzer, G., 1984. External validity of laboratory experiments: the frequency–validity relationship. American Journal of Psychology 97, 185–195.
Glass, R., Vessey, I., Ramesh, V., 2001. Research in software engineering: an empirical study, Technical Report TR105-1. Information Systems Department, Indiana University.
Glennan, S., 1996. Mechanisms and the nature of causation. Erkenntnis 44, 49–71.
Gregor, S., Jones, D., 2007. The anatomy of a design theory. Journal of the AIS 8, 312–335.
Höst, M., Regnell, B., Wohlin, C., 2000. Using students as subjects – a comparative study of students and professionals in lead-time impact assessment. Empirical Software Engineering 5, 201–214.
Hacking, I., 2001. An Introduction to Probability and Inductive Logic. Cambridge University Press, Cambridge, UK.
Hedström, P., Ylikoski, P., 2010. Causal mechanisms in the social sciences. Annual Review of Sociology 36, 49–67.
Holland, P., 1986. Statistics and causal inference. Journal of the American Statistical Association 81, 945–960.
Jackson, M., 2000. Problem Frames: Analysing and Structuring Software Development Problems. Addison-Wesley, Reading, UK.
Kampenes, V., Dybå, T., Hannay, J., Sjøberg, D., 2009. A systematic review of quasi-experiments in software engineering. Information and Software Technology 51, 71–82.
Kitchenham, B., 2004. Procedures for Performing Systematic Reviews, Technical Report TR/SE-0401/0400011T.1. Keele University/National ICT Australia.
Machamer, P., Darden, L., Craver, C., 2000. Thinking about mechanisms. Philosophy of Science 67, 1–25.
McKay Illari, P., Williamson, J., 2012. What is a mechanism? Thinking about mechanisms across the sciences. European Journal of the Philosophy of Science 2, 119–135.
Molzon, J., Pharm, M., 2005. The FDA and the drug-approval process. In: O'Donnell, J. (Ed.), Drug Injury: Liability, Analysis and Prevention. , 2nd ed. Lawyers & Judges Publishing Company, Tucson, Arizona, pp. 3–15.
Morali, A., Wieringa, R.J., 2010. Risk-based confidentiality requirements specification for outsourced it systems. In: Proceedings of the 18th IEEE International Requirements Engineering Conference (RE 2010), Sydney, Australia, IEEE Computer Society, Los Alamitos, California, pp. 199–208.
Nehlig, A., Daval, J.-L., Debry, G., 1992. Caffeine and the central nervous system: mechanisms of action, biochemical, metabolic and psychostimulant effects. Brain Research Reviews 17, 139–170.
Pawson, R., Tilley, N., 1997. Realistic Evaluation. Sage Publications, Los Angeles, USA.
Pearl, J., 2009. Causality. Models, Reasoning and Inference, 2nd ed. Cambridge University Press, Cambridge, UK.
Pearl, J., 2009b. Causal inference in statistics: an overview. Statistical Surveys 3, 96–146.
Petroski, H., 1994. Design Paradigms Case Histories of Error and Judgment in Engineering. Cambridge University Press, Cambridge, UK.
Prechelt, L., Unger-Lamprecht, B., Philippsen, M., Tichy, W., 2002. Two controlled experiments assessing the usefulness of design pattern documentation in program maintenance. IEEE Transactions on Software Engineering 28, 595–606.
Robinson, W., 1951. The logical structure of analytic induction. American Sociological Review 16, 812–818.
Runeson, P., 2003. Using students as experiment subjects-an analysis on graduate and freshmen student data. In: Proceedings of the Seventh International Conference Empirical Assessment and Evaluation in Software Engineering (EASE '03), pp. 95–102.
Russo, F., Williamson, J., 2007. Interpreting causality in the health sciences. International Studies in the Philosophy of Science 21, 157–170.
Schurz, G., 2008. Patterns of abduction. Synthese 164, 201–234.
Sedgwick, P., 2011. Random sampling versus random allocation. British Medical Journal 343, d7453, http://dx.doi.org/10.1136/bmj.d7453.
Seyff, N., Graf, F., Maiden, N., 2010. Using mobile RE tools to give end-users their own voice. In: Proceedings of the 18th IEEE International Requirements Engineering Conference (RE 2010), IEEE Computer Society, Sydney, Australia, pp. 37–46.
Shadish, W., Cook, T., Campbell, D., 2002. Experimental and Quasi-Experimental Designs for Generalized Causal Inference. Houghton Mifflin Company, Boston, USA.
Sjøberg, D., Anda1, B., Arisholm1, E., Dybå, T., Jørgensen1, M., Karahasanovicacutel, A., Vokáccaron, M., 2003. Challenges and recommendations when increasing the realism of controlled software engineering experiments. In: Conradi, R., Wang, A. (Eds.), Empirical Methods and Studies in Software Engineering. Springer, pp. 24–38 (LNCS 2765).
Sjøberg, D., Hannay, J., Hansen, O., Kampenes, V., Karahasanović, A., Liborg, N.-K., Rekdal, A., 2005. A survey of controlled experiments in software engineering. IEEE Transactions on Software Engineering 31, 733–753.
Strauss, A., Corbin, J., 1998. Basics of Qualitative Research: Grounded Theory, Procedures and Techniques. Sage Publications, Los Angeles, USA.
Svahnberg, M., Aurum, A., Wohlin, C., 2008. Using students as subjects – an empirical evaluation. In: Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '08), ACM, pp. 288–290.
Van Aken, J., 2004. Management research based on the paradigm of the design science: the quest for field-tested and grounded technological rules. Journal of Management Studies 41, 219–246.
Vincenti, W., 1990. What Engineers Know and How They Know It. Analytical Studies from Aeronautical History. Johns Hopkins, Baltimore, Maryland.
Wieringa, R., Morali, A., 2012. Technical action research as a validation method in information systems design science. In: Peffers, K., Rothenberger, M., Kuechler, B. (Eds.), Seventh International Conference on Design Science Research in Information Systems and Technology (DESRIST). Springer, pp. 220–238 (LNCS 7286).
Wieringa, R., Daneva, M., Condori-Fernandez, N., 2011. The structure of design theories, and an analysis of their use in software engineering experiments. In: International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE Computer Society, pp. 295–304.
Wieringa, R., Condori-Fernandez, N., Daneva, M., Mutschler, B., Pastor, O., 2012. Lessons learned from evaluating a checklist for reporting experimental and observational research. In: International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE Computer Society, pp. 157–160.
Wieringa, R., 2003. Design Methods for Reactive Systems: Yourdon, Statemate and the UML. Morgan Kaufmann, San Fransisco, USA.
Wieringa, R.J., 2009. Design science as nested problem solving. In: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, ACM, Philadelphia, New York, pp. 1–12.
Williamson, J., 2011. Mechanistic theories of causality, Part I and Part II. Philosophical Compass 6, 421–444.
Willner, P., 1991. Methods for assessing the validity of animal models of human psychopathology. In: Boultin, A., baker, G., Martin-Iverson, M. (Eds.), In: Neuromethods, Vol. 18: Animal Models in Psychiatry I. The Humana Press, pp. 1–23.
Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Weslén, A., 2012. Experimentation in Software Engineering, 2nd ed. Springer.
Wonnacott, T., Wonnacott, R., 1990. Introductory Statistics for Business and Economics, 4th ed. Wiley, Hoboken, New Jersey.
Woodward, J., 2003. Making Things Happen, A Theory of Causal Explanation. Oxford University Press, Oxford, UK.
Yin, R., 2003. Case Study Research: Design and Methods, 3rd ed. Sage Publications, Los Angeles, USA.
Zelkowitz, M., Wallace, D., 1997. Experimental validation in software engineering. Information and Software Technology 39, 735–743.
Zelkowitz, M., Wallace, D., 1998. Experimental models for validating technology. Computer 31, 23–31.
Znaniecki, F., 1968. The Method of Sociology. Octagon Books, New York (First printing 1934).

Roel Wieringa is Chair of Information Systems at the University of Twente, The Netherlands. His research interests include requirements engineering, risk assessment, and design research methodology. He has written two books, on Requirements Engineering and on the Design of Reactive Systems. His next book, Design Science Methodology for Information Systems and Software Engineering will appear in 2014 with Springer.