1 Bio of presenter

Roel Wieringa (http://www.cs.utwente.nl/ roelw) is Chair of Information Systems and head of the Computer Science Department at the University of Twente, the Netherlands. His research interests include requirements engineering, risk assessment, and research methodology. He has written two books, on Requirements Engineering and on the Design of Reactive Systems. He serves on the board and program committees of various journals and conferences.

2 Title

Technical action research: designing and and reasoning about single case experiments

3 Abstract

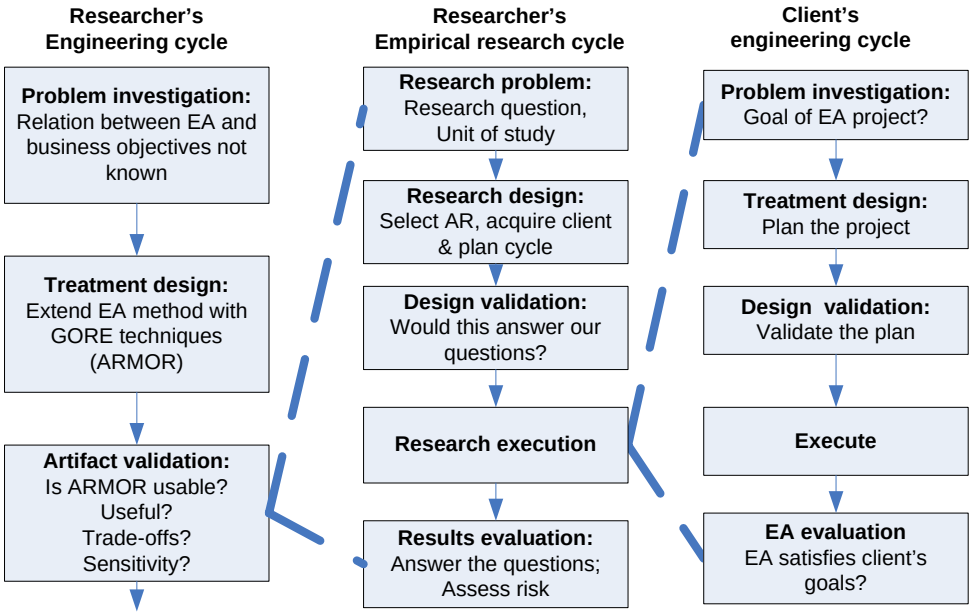
Information systems researchers often develop methods or techniques to be used in the development, maintenance and redevelopment of information systems. These techniques can be validated to some extent in the laboratory, using students as subjects, but to learn how the technique performs in a realistic situation the technique must be applied it in a practical context. This means that the researcher must use the technique herself in practice, or else teach practitioners the technique, and then observe how they use it in practice. We call this *technical action research* (TAR). In it, the researcher is playing three roles: (1) as a designer of the technique, (2) as a helper of a client, and (3) as a researcher searching for knowledge about the use of the technique in practice. The key to a methodologically sound TAR project is keeping these three roles separate.

TAR differs from consulting because consultants apply known techniques rather than experimental, unproven ones. They do not have the goal of testing a new technique but of using a proven technique to help the client. Knowledge about the technique acquired during a consultancy project is confidential [1].

TAR also differs from action research as performed in in the social sciences [5]. The aims of social science action research are that the subjects learn more about their own situation in order to prepare for emancipatory action. What this action will be is not known at the start of the project, and it will be determined by the subjects.

In TAR, the researcher develops a new technique, and helps the client with the technique, in order to learn something about the technique. The researcher is involved in two separate *engineering cycles,* where each engineering cycle consists of the following tasks:

- Problem investigation
- Treatment design
- Treatment validation



- Treatment implementation
- Implementation evaluation.

In the *researcher's engineering cycle,* the treatment is a new technique to be used in information systems development or maintenance. In the figure, the technique under development is called ARMOR.

In the *client's engineering cycle,* the treatment is some improvement action that the client wishes to perform to achieve their own goals. The researcher helps the client with this improvement action, and uses her own technique to do this. For the researcher, this is a validation of her technique in a practical context.

Between these two engineering cycles stands a *research cycle,* in which the researcher asks the relevant validation questions, for example whether the technique is usable or useful, and checks that these questions are answerable at least in the action research case. After completing the client cycle, the researcher answers the research questions in the research cycle.

The researcher must answer the research questions for the client's case, but then wants to draw lessons learned that go beyond this particular case. Statistical inference is not applicable here. Rather, case-based reasoning is applicable, in which the researcher can provide an argument that in future cases, similar this client's case, the answers to the research questions will be the same. In other words, the researcher may provide arguments that in similar cases, the techniques may be just as usable and useful as for this client.

But not any similarity is relevant for such generalization. To separate relevant from irrelevant similarities, the researcher should identify and explain the *mechanisms* that produced the effects of the technique in the client's case. These mechanisms, in turn, depend on the *architecture* of the case, where architecture here is the set of entities that make up the case, their capabilities, and their influence relations. For example,the client organization consists of people, roles they play, organization units they work for, software, physical places, business processes, etc. If the researcher can show that the effects of the technique in the client's case were produced by particular mechanisms, and that these mechanisms were the result of the architecture components, capabilities and influences of the case, then the researcher can claim that in cases with a similar architecture, the technique will probably have similar effects. This is illustrated by means of a number of published examples of TAR as well as examples from practice. Practitioners habitually draw architecture diagrams of new ways of organizing the information systems development and maintenance process, and of the role new techniques play in this architecture. These diagrams can be used to explain the effect of a technique in this architecture, as well as to assess whether the desired effects of the technique will occur in the case at hand or not.

Architecture-based reasoning from the current case to possible future cases is fallible, and there are some obvious threats to validity: The future users of the technique may not be as competent as the researcher in the use of the technique; the researcher may interpret the effectiveness of the technique too optimistically, or may misunderstand the mechanisms that produced the technique's effects; future cases may not be relevantly similar to the current case; or they may be relevantly similar, but may contain additional mechanisms that will disturb or destroy the effects of the artifact. The researcher must acknowledge and assess these threats to external validity of the TAR project.

4 Scope

Information systems research is about the design and use of information systems in real-world contexts. To validate new techniques for information systems, the researcher eventually has to try out his or her technique in practice and draw lessons learned from this about the suitability of the technique for practical situations. Technical action research (TAR) is a research method where the researcher uses his or her newly designed method to help a client to achieve their goals better. The aim of this tutorial is to teach how to set up and reason about a TAZR project in a methodologically sound way.

Examples will be taken from published research in information systems and software engineering by the presenter [2, 3, 4, 6]. I will also engage the audience in a discussion about possible TAR projects that members of the audience would like to do.

5 Background of the attendees

The intended audience consists of researchers and engineers who want to validate new technology empirically. This includes PhD students and industrial researchers who have designed a new technique, method, software architecture, algorithm or other kind of artifact, and want to show how this new artifact would perform in practice. The audience is assumed to have at least a bachelor-level background in information systems, software engineering or computer science.

6 Material

The attendees will receive a hardcopy of the slides.

7 Timetable (90 minutes)

The companion tutorial proposal "Empirical validation research methods" can be followed independently of this one. They can be combined into a 3-hour tutorial starting with the validation research tutorial and continuing with this one.

- What is technical action research? (15 minutes)
  - Example
  - Characteristics
  - Contrast with consulting
  - Contrast with classical action research
  - Why technical action research (for the researcher)
  - Why technical action research (for the client)
- The logical structure of technical action research (40 minutes)
  - The engineering cycle
  - The researcher's cycle and the client's cycle
  - The empirical research cycle
  - The nested structure of technical action research
  - Examples
- Generalizing from technical action research (30 minutes)
  - Case-based reasoning versus statistical inference
  - Reasoning by analogy
  - Architectural inference
  - Threats to external validity
  - Examples
- Summary and take-home message (5 minutes)

References

[1] R.L. Baskerville. Distinguishing action research from participative case studies. *Journal of Systems and Information Technology*, 1(1):25–45, March 1997.

[2] W. Engelsman and R.J. Wieringa. Goal-oriented requirements engineering and enterprise architecture: Two case studies and some lessons learned. In *18th International Working Conference on Requirements Engineering: Foundations for Software Quality (REFSQ)*. Springer, 2012.

[3] W. T. B. Hordijk and R. J. Wieringa. Rationality of cross-system data duplication: A case study. In *Advanced Information Systems Engineering (CAiSE), 22nd International Conference, Hammamet*, volume 6051 of *Lecture Notes in Computer Science*, pages 68–82, London, 2010. Springer Verlag.

[4] A. Morali and R. J. Wieringa. Risk-based confidentiality requirements specification for outsourced it systems. In *Proceedings of the 18th IEEE International Requirements Engineering Conference (RE 2010), Sydney, Australia*, pages 199–208, Los Alamitos, California, September 2010. IEEE Computer Society.

[5] G.I. Susman and R. Evered. An assessment of the scientific merits of action research. *Administrative Science Quarterly*, 23(4):582–603, December 1978.

[6] E. Zambon, S. Etalle, R. J. Wieringa, and P. H. Hartel. Model-based qualitative risk assessment for availability of IT infrastructures. *Software and Systems Modeling*, 10(4):553–580, June 2011.