**Formulating research questions for development projects**

*Martin Stacey*

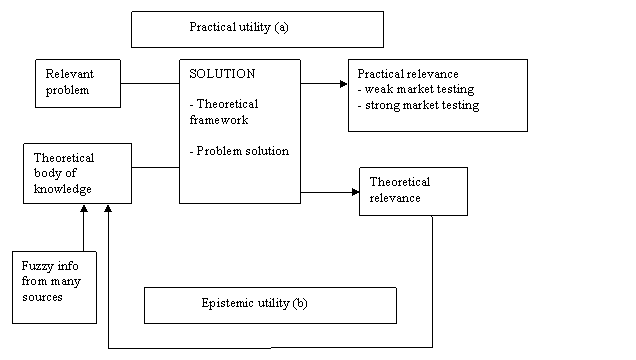
**The problem: Making a contribution to knowledge by building a system**

A masters project needs to be rigorous scholarly work that makes a contribution to the world’s knowledge. It needs to tell us something. For a masters degree, this doesn’t need to be especially new or original, but it does need to include a serious and scholarly attempt to investigate one or more questions. If you don’t know how your project on your chosen topic is going to make a contribution to knowledge, it’s not a suitable topic for a masters project.

But lots of MSc projects in computer science and related disciplines are primarily about building running software systems. You want to build a system that does something interesting, but it may not be obvious to you how it’s going to make an academic contribution. The trick is formulating your objectives for what you want to build as research. Doing this well will help you satisfy the requirements for a masters degree while developing your technical skills and building something real, help you formulate what you want to do in rigorous and general terms, which will help you do it, and help you actually *make* a contribution to knowledge.

This document is about how to do this.

**The theory: Constructive research and design science**

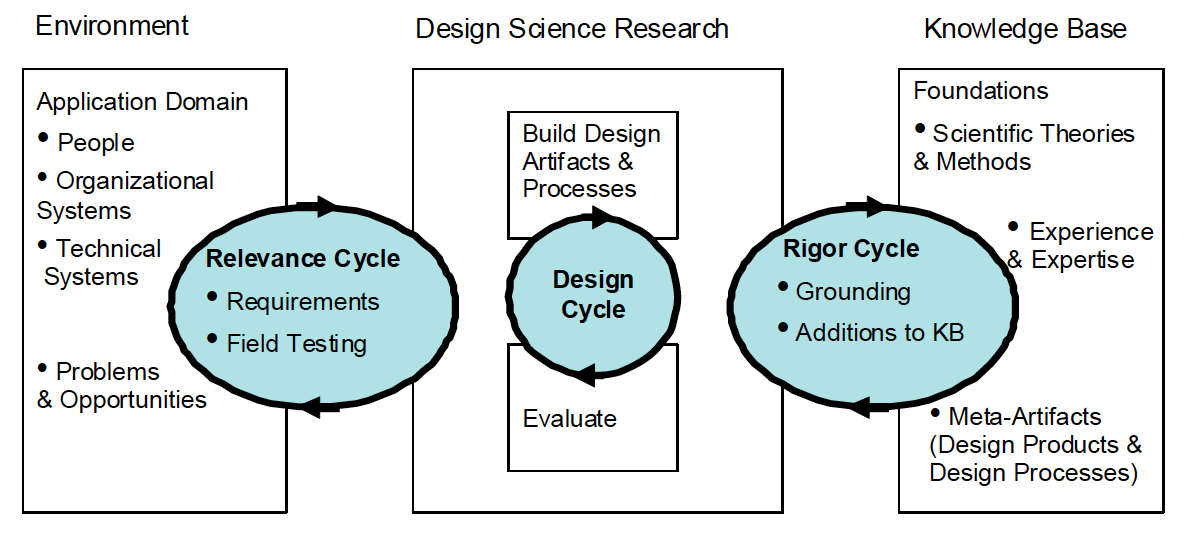


*Figure 1: Constructive research, from Wikipedia, created by Rasvan Constantinescu, 2005.*

Academics like research grounded in theory and drawing on previous work in the area. You’re not the first to worry about how to make building stuff look like real research, and actually *be* real research. The key buzzwords are *constructive research* and *design science*. Both of these turn development into research with the key idea of building systems to answer questions – you state the hypothesis that the system will solve this problem (or these techniques or these development methods solve this class of problems), and then test the hypothesis by building the system and evaluating how well it solves the problem. You thus create knowledge about the problem itself, and what happens when you try to solve it a certain way, about the power and utility and limitations of the techniques you use, and about the effectiveness of the development methods you use.

*Constructive research* is research by constructing stuff, and looking at relevance to both practical needs and theoretical knowledge (see Figure 1). Lehtiranta et al (2015) summarize as follows: “The research process involves the following: (1) selecting a practically relevant problem; (2) obtaining a comprehensive understanding of the study area; (3) designing one or more applicable solutions to the problem; (4) demonstrating the solution’s feasibility; (5) linking the results back to the theory and demonstrating their practical contribution; and (6) examining the generalisability of the results.”

*Design science* (a term used in the field of information systems) has about as much to do with design and science as Jerusalem artichokes and the Holy Roman Empire. [The academic study of how designing is or should be done is *design studies* – whether this is or should be science is an issue that is debated but the term design science is sometimes used for research on design that is meant to be scientific; engineers often use the term design science for the scientific knowledge used in designing.] The design science idea is that research happens in and through designing and the research contributes to the designing. The key to this is rigorous evaluation of what the system does and how well it solves the problem and what can be learned from it.

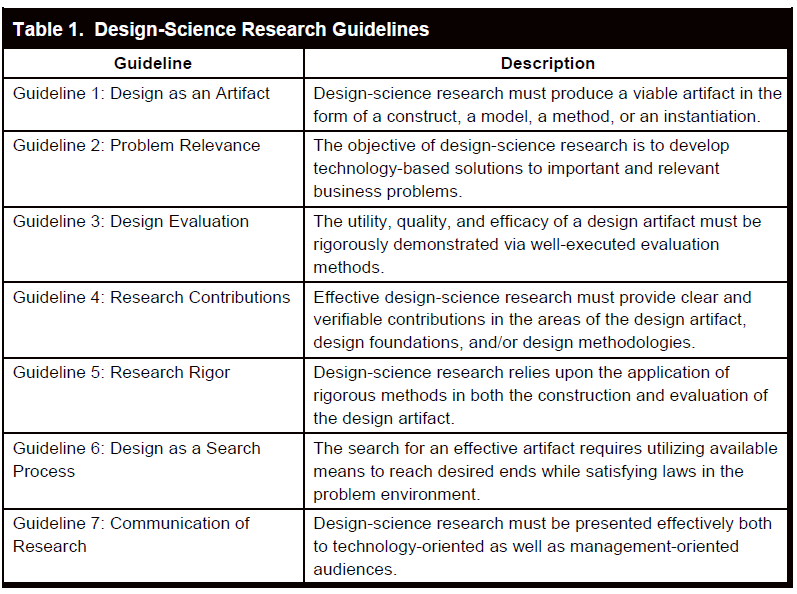


*Figure 2: Design Science Research Cycles, from Hevner (2007).*

Alan Hevner, the chief guru of design science, talks about three cycles of research (see Figure 2): the Design Cycle, in which the artefact is built, evaluated and modified; the Relevance Cycle, in which the artefact is compared to the real world, to see how it meets the requirements and what it can tell you about what the requirements are and the nature of the problem; and the Rigour Cycle, in which the development of the artefact draws on theoretical knowledge and knowledge about techniques and methods, and contributes to that knowledge (Hevner, 2007). They are cycles because they are repeated and the transfer of information goes both ways.

Hevner et al (2004) characterize ‘the design-science paradigm’ as seeking ‘to extend the boundaries of human and organizational capabilities by creating new and innovative artefacts’. Note that the artefact isn’t necessarily running software – it might be a method for solving a class of problems or some other usable thing.

Hevner et al (2004) argue that doing all the phases of the research rigorously is essential for producing worthwhile results. They put forward a set of guidelines for how to do good research by building systems (in Table 1), and discuss in some detail what following them involves. Although this paper is quite long, it’s worth reading for a sophisticated analysis of how building artefacts for practical purposes can be research, and thus is a good source of ideas for how to formulate your project.



*Table 1: Design-Science Research Guidelines, from Hevner et al (2004).*

**The starting point: Understanding your problem**

Okay, so what do you want to do, and how do you want to do it?

Development projects start with a problem to solve or a need to meet. Having a novel problem is a great way for a project to make a scholarly contribution, but it isn’t the only way, or a recipe for a straightforward feasible project. Another good way is applying novel methods to a familiar problem.

You need to start with as subtle and sophisticated understanding as you can get of the problem itself. For an interactive system designed for real or hypothetical users, this means understanding what the users would really want to do with it, and how they would want to interact with it. A thorough and well-documented requirements analysis adds a lot to a project.

Problem Analysis

Problem Formulation

System Development

Artefact

Evaluation

Results of Evaluation

Development Methods

Techniques

Evaluation Methods

Need

Analysis Methods

Domain Knowledge

*Figure 3: System Development Process*

It’s vital to remember that few development projects start with a clear, exact and correct statement of the problem and the requirements for the system, in academia or the real world. You need to think hard at the beginning, but your understanding of your problem will and should develop through the course of the project. In particular, you’ll learn new things about the user requirements for an interactive system as soon as you show prototypes to users.

Your evaluation of your system might not just be about whether the system met its requirements, but whether it was trying to solve the right problem. A lot of artificial intelligence research has progressed by using the failures of AI systems to learn more about what solving the problem really involves.

**The contribution: What are you doing that’s new?**

To make a contribution, you need to do something that’s new, so you can find new knowledge. There are different sources of newness, and different places where you can make a contribution to knowledge (as shown in Figure 3).

*What is* ***new****?*

*What is* ***modified*** *from previous work?*

*What is* ***extended*** *from previous work?*

*What is* ***repeating*** *previous work?*

(Repeating something someone else has done, to see if you get the same results they did, is good but shouldn’t normally be the only thing you do. Testing whether techniques or development methods work in different situations is good.)

*Where* in the development process are you *doing* something new? *Where* in the development process are you able to *find out* something new?

This can be in the *problem*, in the *techniques* your system uses, the *development methods* you employ to build your system, the *tools* you are using to build your system, and in the *evaluation methods* you use to see how well your system meets the need and what you can learn from it.

What is the relationship between what you’re doing that’s new (or new-ish) and what other people have done before? If you’re tackling a new problem, how is it related to other problems that other people have tacked? Can you identify a class of related problems that could be solved the same way?

What can you learn about what you’ve done that’s new from your system?

**The primary questions: How successful are the techniques and methods?**

What can building your system *tell you*? Don’t just have one question. Posing a range of questions including concrete and specific questions and general and theoretical questions will help you frame your project and make it easier to generate some results at the end. Which of these are relevant to your problem? How can you adapt and rephrase these and make them more specific to your problem?

* **Does the artefact solve the problem?** (Phrased as a hypothesis: The research investigates the hypothesis that the artefact solves the problem. Does the evidence support or disconfirm the hypothesis?) *How well does the artefact solve the problem? How much of the problem does the artefact solve, and in which situations? With what weaknesses and limitations? What are the sources of the weaknesses and limitations?*
* **Do/can the techniques used in the artefact solve the problem?** *What is needed to achieve a solution to the problem? Can the techniques be used to solve a particular class of problems? With what limitations? Can the techniques be extended to handle a wider range of problems? What classes of problems are you in a position to make claims about?* (Remember, there is nothing wrong with admitting that you chose the wrong approach to solving the problem, if you can explain why.)
* **Was the application of the development methods in the process of developing the artefact successful?** *How did the development method contribute to the success or failure of the project? How successfully was it adapted and applied? What are the weaknesses and limitations of the method? How can these limitations be overcome?*
* **Was the application of the evaluation methods in the process of evaluating the success of the artefact successful?** *How did the evaluation method contribute, or not, to figuring out what worked and what didn’t, and what lessons can be learned? What are the weaknesses and limitations of the method? How can these limitations be overcome?*
* **Is the problem the artefact was designed to solve actually the real problem?** *What can we learn from the research about the nature of the problem itself? Does the artefact or the methods it employed offer a partial solution to the real problem or is the approach the wrong one? What can we learn about the application domain?*

**The secondary questions: How strong are the conclusions?**

Just having questions and answers isn’t enough. You also need to evaluate how strong your conclusions are, and how strong the evidence is for them, and what limitations and reservations they have. Intelligent thought about what you can conclude or can’t is far more important than having strong conclusions.

* **How convincing are the results?** *How trustworthy are the assessments or observations or experimental results? To what extent have the research methods produced reliable data? How much confidence can we have that the data is interpreted correctly?*
* **How strong is the conclusion?** *How forceful a statement does the evidence point to? How exact are the numerical values or other statements of information?*
* **How strong is the evidence for the conclusion?** *Is the evidence weak or limited? Is the evidence contradictory? Why?*
* **What is the scope of the conclusion?** *What range of systems or situations does the conclusion apply to? Does the evidence offer weaker support for more general conclusions?*
* **How does the conclusion fit or contradict previous research findings or theory?** *Can you make predictions from theoretical knowledge about what ought to be the case? Are these supported? Do the conclusions fit or conflict with earlier, more general findings? Why? Do the conclusions fit or conflict with earlier work on different but similar systems or situations?*
* **How can the conclusions be tested in further work?** *How could further development or evaluation of the system provide better answers to the questions? What further work could make the conclusions more robust or disconfirm them? What further work could make the conclusions more general? What of this would be worthwhile and cost-effective to do?*

**References and Further Reading**

*A practical introduction to constructive research:*

**Liisa Lehtiranta, Juha-Matti Junnonen, Sami Kärnä and Laura Pekuri** (2015) The Constructive Research Approach: Problem Solving for Complex Projects, in Beverly Pasian (Ed.) [*Designs, Methods and Practices for Research of Project Management*](http://www.gpmfirst.com/books/designs-methods-and-practices-research-project-management), Gower, 978-1-4094-4880-8. <http://www.gpmfirst.com/books/designs-methods-and-practices-research-project-management/constructive-research-approach>

*A short paper introducing the 3 cycle view of Design Science:*

**Alan R. Hevner** (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, 19, issue 2 article 4. <http://aisel.aisnet.org/sjis/vol19/iss2/4/>

*The key paper on Design Science, with over 7000 citations in Google Scholar. Rather longer, but the place to go for a sophisticated discussion of the relationship of system development and evaluation to research, thus a source of ideas about formulating your project:*

**Alan R. Hevner, Salvatore T. March, Jinsoo Park and Sudha Ram** (2004). Design Science in Information Systems Research, *MIS Quarterly*, 28, 75-105. <http://www.csis.ul.ie/staff/brianFit/phd-seminar-series/Hevner-et-al-2004-misq--des-sci.pdf>