Current and future research methods

MIA MÅNSSON
Department of Computer Science
June 21, 2016

Current research methods

I haven't started my PhD studies yet, so I can not really tell what my current research methods are. My research area would reside within the Internet of Things category, and specifically software updates. As for my master thesis work, which my PhD studies most likely will be some form of continuation of, I used a method of building a software artifact and evaluating it. It was evaluated in terms of how it solved the problem statement and I also made some repeatable, but very non-rigorous experimental measurements of performance, obviously, as I hadn't read the "Statistically rigorous java performance evaluation." [1] at the time...

As we discussed in the meeting last week, it is fairly common that researchers prefer to look for problems that can be solved by their favourite research method. Hence the importance of reflecting if there actually could be a better method to solve a particular problem.

The articles in my field that I've read so far seem to be rather engineering-minded where the goal is to improve or build something useful, typically some sort of software artifact as a proof of concept. An other method that seem to be common in my field is literature study. Literature studies seem to be useful when there is not yet a complete solution to a problem, but instead they do an analysis of what other researchers have done, and utilize their ideas in order to explore what is needed and what is still lacking in order to solve their own specific problem.

There are most likely lots of methods not so commonly used that could improve the overall result of the research, for instance when evaluating the result of the research.

Future research methods

I definitely think there is potential of incorporating several methods in my future research. For example one could do a:

Field study This may be useful during the "make observations" - step in the scientific method. A field study is likely a good way to find problems that actually need to be solved and to understand them better. For my research topic, it may be useful to do a field study (and a literature study) in the beginning of my research in order to see how the industry currently handles the problem, and what solutions that are already present.

Statistical evaluation is, as I already mentioned above, probably useful to add when evaluating a result our outcome in one form or the other, particularly when software performance is of significance.

Case study can ofcourse be difficult to isolate, but may be helpful in many cases to evaluate how the object of study behaves in its context in real life.

Methodological triangulation i.e.combining both qualitative and quantitative data, may help to make the validity of the result stronger [2]. So for instance combining the results of a case study with experimental results might be one way to evaluate a result and look at it from different angles.

When the research contribution is a software artifact, as it was in my master thesis, it was fairly straightforward to do experiments on the artifact, i.e. generate quantitative data. But I think it sometimes may run the risk of being less general and not so useful for the rest of the world.

Methodological triangulation could perhaps help to make my research results more generally valid. For instance; in addition to experiments it may also be helpful to do a thorough analysis of how the problem can be solved more generally. This would probably help others to implement a similar solution in their software systems. I think this may be a good idea partly because the industry and other researchers may be forced to use different frameworks than the ones used when building the research artifact, but also because such artifacts may not always be a good fit for production software, for instance since development costs usually have to be much lower for research artifacts.[3]

References

1. Andy Georges, Dries Buytaert, and Lieven Eeckhout. Statistically rigorous java performance evaluation. *ACM SIGPLAN Notices*, 42(10):57–76,

2007.

2. Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14(2):131–164, 2009.

3. Shriram Krishnamurthi and Jan Vitek. The real software crisis: Repeatability as a core value. *Communications of the ACM*, 58(3):34–36, 2015.