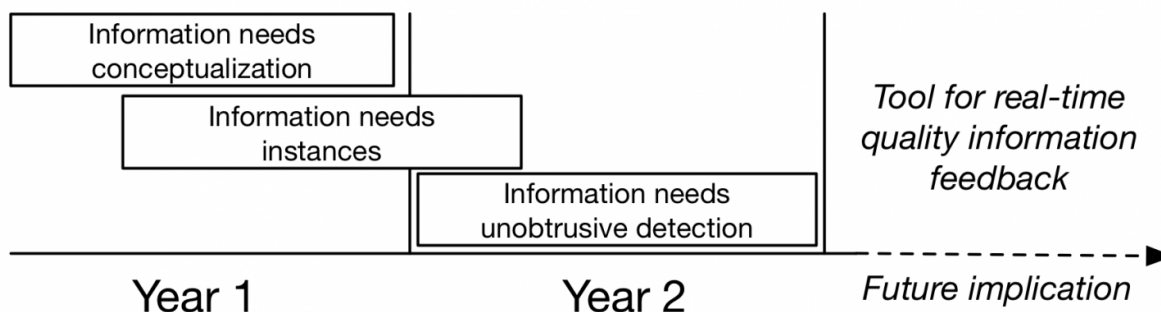# New development tool based on 'software quality information needs' and three case studies

April 21 2016

Timeline of the research activities and future implication. Credit: Dr. Daniel Graziotin

While constantly developing, software takes over more and more aspects of our life at both individual and community level. Thus, software failures and security are easily becoming major concerns which need to be addressed on the spur of the moment.

A way to do so, according to computer scientist Dr Daniel Graziotin, University of Stuttgart, is adopting a new concept, which he terms 'Software quality information needs', along with multi-angled extensive empirical evidence to produce a development tool to improve software quality. His Grant Proposal is published in the open access journal

Research Ideas and Outcomes (*RIO*).

The proposed research is in the context of [an earlier DFG Grant Proposal](#), authored by Prof. Dr. Stefan Wagner, University of Stuttgart, and published in the same journal. The earlier project idea suggests novel tools analysing software changes before and during their implementation. Similarly to Dr Daniel Graziotin's idea, it is based on fast and focused feedback loop.

The presently proposed research, planned to take 24 months, is set to start with establishing the 'software quality information needs' construct. The author coins this theory in order to conceptualise and provide deeper understanding of the information essential for a developer when performing code changes or designing new parts of a system.

The project is to then go on to produce metrics to detect and satisfy a developer's needs. As a result, optimally unobtrusive measurement techniques are to be developed and evaluated in three empirical studies.

About 120 software engineering students from the University of Stuttgart are to be recruited to provide empirical evidence. While they are working on either real-world or university software projects, they are to be observed, regularly interviewed and asked to think aloud. Their insights will be further enriched through a post-task interview. The findings are to answer the question "How can we conceptualise information needs when dealing with software quality?".

"What information is needed when dealing with software quality?", is to be covered by the second empirical study, which plans to involve the software engineers of Daimler, Porsche, and Bosch, since the automotive industry is particularly concerned with software quality issues. The engineers are to fill in mostly open-ended surveys and thus, provide a broad view of software quality information needs and their priority from

a practitioner's perspective.

Building on the above case studies, the German multinational engineering and electronics company Robert Bosch GmbH is to be approached for validated questionnaires and behavior patterns tests, such as keystroke frequency and typos detection. Ultimately, the findings are to answer how software quality information needs to be detected unobtrusively through behavioral patterns.

To exemplify the tool, based on the described research and its expected findings, the scientist uses fictional software developer, called Anne. While working on a system routine to be applied in a banking application, she is notified by the integrated development environment (IDE), that there might be some quality issues.

It turns out that she is employing a design pattern that is not frequently employed in similar cases, so the IDE suggests that she browses some StackOverflow.com related questions and answers regarding the design pattern. Because she has also created part of the procedure by cloning code from another part of the project, the tool offers her to help to refactor the cloned code.

"Providing a software developer with the right kind of information about the current state of and the effect of changes on software quality can prevent catastrophic software failures and avoid opening up security holes," Dr Daniel Graziotin argues.