

MASTER'S THESIS Using Already Existing Data to Answer Questions Asked During Software Change

STUDENTS Daniel Jigin & Oscar Gunnesson

SUPERVISORS Lars Bendix (LTH) & Christian Pendleton (PRAQMA)

EXAMINATOR Boris Magnusson (LTH)

Answering Developer Questions During Software Change

POPULAR SCIENCE SUMMARY **Daniel Jigin & Oscar Gunnesson**

Providing answers to questions developers have is important to avoid frustration and bad decisions. Today, questions about changes to the software are answered by gut feeling and by asking other people; we want to make these decisions data-based. We have investigated tools utilizing already existing information to know what questions we can answer today while not having to introduce new frameworks for project data structuring.

Studies of how developers spend their time have shown that they spend as much time searching for whom to contact in the organization to get answers to their questions, as they do getting the job done. But what is it that developers need and what do they feel that they are lacking and how can we help?

In traditional software development, configuration managers used to bring a status report about the software to the managers. We have suggested a more modern approach, fit for an agile methodology, where the status of the software is available for any worker at any time. When asking developers how they find answers they say that it is based on gut feeling coming from previous experiences. Adding data to the gut feeling would lead to more informed answers.

In our thesis, we have conducted interviews with developers that collected 57 questions. We have focused on questions concerning impact analysis and technical debt, although we found that there were questions concerning testing and implementation as well. Impact analysis is the process of identifying potential consequences of a change,

such as finding dependencies in the source code. Technical debt (TD) is a metaphor for describing the cost of taking short-cuts in development. Values on TD can be used to indicate the cost and improve the gut feeling of estimations.

We were introduced to a tool called CodeScene that collects existing data in software repositories and presents metrics like dependencies between configuration items and different code metrics. To know what data that is available today we have investigated and explored CodeScene. We have shown that tools can provide us with all data needed to perform an impact analysis, it is just not structured in a suitable way. We found that being able to select a file to get a set of all dependent files and include TD metrics for each file will help solidify the gut feeling developers have.

To validate the answers to the questions, a proof-of-concept tool was implemented with the purpose of showing that, if the data was presented in a suitable way, it could be used as an impact analysis tool. Furthermore, adding TD metrics of the affected files in the impact set will enable a data-driven discussion of effort estimation.