



Thesis title: *Quality Assurance under Extreme Time Pressure - A Hackathon Case Study*

Thesis description:

Software engineering refers to development of software according to a systematic and organized approach, aiming to effectively produce high-quality software with reduced uncertainty. It doesn't have to follow strict processes though, the prevailing trend in industry the last decade has been agile and lean development - focusing on tight customer feedback and working software, and relentlessly eliminating anything that isn't adding value [1]. Agile and lean development are popular approaches to reduce time-to-market, and to adapt to ever-changing end-user needs and expectations [2]. Still, quality is fundamental to make customers happy, as reflected by the agile manifesto stating "continuous attention to technical excellence" and the second lean principle being "build quality in".

An extreme case of lean development is found at hackathons, a programmers' meetup during which small teams do rapid prototyping of concepts from ideation to demo. Hackathons typically last a day or two, thus creating extreme time pressure – absolutely no time should be wasted on work that does not provide value to the demo! However, while certain quality aspects such as maintainability is not in focus, other qualities surely are, e.g., usability, visual appearance, and functional correctness. How do hackathon teams address quality under extreme time pressure?



Obviously, there is an approach to quality assurance (QA) also at hackathons. But where do teams focus their energy? Which software engineering practices remain under this extreme form of lean development? How is the end-user perspective managed (i.e., requirements engineering)? How is the source code managed? Is code review a common practice? What about continuous integration? How often do teams gather in quick meetings to align their views? How is software testing done? Is there any automated testing? Is all testing exploratory? There are numerous questions, but very little research is available on hackathons – albeit the findings could also help us understand time pressure in industry [3].

The objective of the thesis project is to conduct a case study to scientifically study QA at a hackathon. Essential steps in the project is to select a hackathon and to read up on the domain it targets (internet of things, e-health, game development, humanitarian software etc.). Get approved by the hackathon organizers to do a study. Then preparations are needed to prepare the data collection (interviews, surveys, source code analysis etc.). Finally, attend the hackathon to collect your data, and bring it back to the lab for analysis using a combination of qualitative and quantitative methods. Could you find any QA differences between successful and not-so-successful hackathon teams? Compare your findings to the body of research on software development under time pressure, and develop guidelines for QA at Hackathons.

- [1] Huo et al., Software Quality and Agile Methods, In Proc. of the Annual International Computer Software and Applications Conference, pp. 520-525, 2004.
- [2] Mkandla and Dwolatzkyg, Defining Agile Software Quality Assurance, In Proc. of the Int'l. Conf. on Software Engineering Advances, 2006.
- [3] Prechelt et al., Quality Experience: A Grounded Theory of Successful Agile Projects Without Dedicated Testers, In Proc. of the 38th Int'l. Conf. on Software Engineering, pp. 1017-1027, 2016.

Tasks:

- Study existing research literature on software development under time pressure.
- Design a study to explore which software engineering QA best practices remain at super-lean Hackathons
- Attend a Hackathon, collect qualitative and quantitative data.
- Analyze the data and compare to related research on time pressure and software quality.
- Develop guidelines for QA at Hackathons, and finally validate your recommendations with Hackathon participants.

Key skills:

- Essential: software engineering, development processes, software quality
- Preferred: requirements engineering, software testing, interview studies

Applications should include a brief cover letter, CV, and recent grades. In your application, please provide examples of previous programming experiences or other projects that you consider relevant for the position.



Expected start time: January-March 2017

Location: SICS Swedish ICT AB, Lund

Contact person/s: Dr. Markus Borg (markus.borg@sics.se)



About SICS:

SICS Swedish ICT is a leading research institute for applied information and communication technology in Sweden. SICS is a part of RISE, Research Institutes of Sweden, a non-profit research organization owned by the Swedish government and industry. SICS' mission is to contribute to the competitive strength of Swedish industry by conducting advanced and focused research in strategic areas of computer science, and actively promote the uptake of new research ideas and results in industry and society at large. SICS is an active participant in collaborative national, European, and other international R&D programs.

The Software and Systems Engineering Laboratory (SSE) has as its research theme "successful development of software-based systems for the digital society". The focus is on providing relevant solutions for software and systems development in a society that becomes increasingly connected and automated, and frequently updated through continuous deployment. Special attention is given to large-scale cyber-physical systems, where traditional embedded systems are evolving into systems-of-systems through connectivity, using technologies from the Internet of Things. Currently, the research activities in the SSE lab focus on four areas: process evolution, system architecture, software ecosystems, and system qualities.

The RISE institutes SP, Swedish ICT and Innventia are merging in order to create a unified institute sector and become a stronger innovation partner for businesses and society. At the end of the 2016 we will change our name to RISE. Read more at www.ri.se/en/about-rise

SICS Swedish ICT AB,
Software and Systems Engineering Lab
Ideon Science Park
Building Beta 2 3v
Scheelevägen 17
SE-223 70 Lund
