SUMMARY

The purpose of this graduation thesis is to derive a software testing improvement methodology applicable to mainly but not only electric drives. To achieve this goal, analysis of existing software testing and process improvement methods were analyzed and used as a base for desired methodology. To verify the effectiveness of derived methodology, a case study was performed at ABB Estonia.

This thesis is separated into three chapters. First chapter focuses on the analysis of software testing methods, including embedded and drives' software. Potential problems related to testing process are analyzed to provide a better representation of why a need for improvement might occur. Second chapter focuses on the analysis of well-known existing process improvement methodologies, that could be used as a base. The aim of the chapter is to combine those methodologies in a way that could be used to improve software testing processes and provide guidance for people or teams responsible for them. Final chapter describes the case study conducted by the author. As a part of software testing team, he has used derived approach to improve a group of inefficient tests executed by his team. In this chapter existing testing methods are analyzed, as well as the improvement steps and outcomes. Some parts of the solution are described as well, however more attention is drawn to analysis is process improvement parts.

The results show that the methodology derived from general process improvement tools and methodologies can be used for improving software testing. It can be used in a team or by a single contributor as a roadmap for the improvement. It is based on Cause and Effect Analysis, PDCA cycle and 5S, if applicable. Following the methodology author was able to map required steps to actions, which were driving the improvement in the case study. Throughout the improvement process, it was much simpler to plan and analyze throughout the process due to well-defined structure.

Considering outcome of the case study, the time a responsible person must spend on testing was decreased from 2 days to 1 hour. That will help the team to do more work during each release cycle, be more productive and, of course, save costs for the company. Automation of software loading, after some extra improvement could decrease setup required person working time from one day to less than one hour as well, which would further increase the efficiency of the process.