

Process oriented Software Quality Assurance – An experience report in Process Improvement – OEM perspective

Thomas Illgen, Stefan Ortmann
Carmeq GmbH, Berlin
thomas.illgen@carmeq.com

Introduction

In the early 19th Henry Ford has the vision of an economical build car priced for everyone. Henry Ford reaches his vision by a new organisation of well defined management and engineering processes. A historical citation is

„It means a lot to me to prove clearly that our ideas are all about accomplishable ... that they are not automotive specific but rather they are a part of a global code.

One day he explains that he in the future would build only one kind of a car and every car has the same chassis. He explained that each customer could paint his car as he want if it is black. His consequent way to his success by a very limited, standardised offer of one car was also his problem in the future when GM starts a model offensive with several configuration possibilities. What could we learn from Henry Fords vision and his implementation? He realised that individual construction steps and a lot of individual build parts is the main problem for reaching high quality and low costs. In modern cars we find the same problems today with complex software systems build by individual “artists”. A well defined software construction process and standards for key components seem to be a successful way for future software-engineering. One of the main topics of building complex and safety critical software systems is the establishing of constructing quality as key knowledge for automotive engineers.

Process oriented Software Quality Assurance

There are several definitions for the word process. Over all they define a process as a transformation of inputs in a set of well defined outputs. But the most important definition in the daily work is the process as a well defined

contract between engineers about how to construct an output.

In general we have to establish two different views on quality assurance. The “design” of quality which is the constructional view as a part of the project. The verification view is the destructional view of the classical quality assurance as a part of the customers view. The two roles with these two different tasks have to be organisational separated. The effort for this roles is different over the software lifecycle. The effort for the constructive quality assurance is very high in the beginning of the project and between two milestones. In the beginning the main task is to plan quality assurance tasks for the project based on the project plan. Between two milestones the main task is to establish the planed quality assurance actions. The effort for the verifying Quality assurance is in comparision very high around a milestone of the project plan. This verification could be seen as a hurdle in the software lifecycle which in the past often was a full stop in the engineering process. To avoid this full stop the constructive quality assurance could be seen as stairs to jump over this hurdle.

The task of the constructive quality assurance is to fix the quality targets of the project, to identify the work products, plan the quality assurance items for the work products, to measure the success and to establish successful methods. Further quality requirements from norms, safety requirements or other stakeholder outside the project have to be taken into consideration.

The quality assurance process over the project lifecycle could be descibed in the classical Plan-Do-Check-Act-cycle.

Process Improvement

Process capability maturity models like SPiCE are building a very good reference model for process improvement. The success factors for

	Construction (project)	Verification (global)
Plan	Quality targets, quality engineering methods, metrics, verification, project specific tailoring of existing quality processes	Global targets, acceptance process, audits, assessments, reporting, escalation, metrics
Do	Implement the plan, measure success and manage documentation	Verification of quality targets, acceptance test, internal and external assessments and audits, managing quality reporting
Check	Analyse project quality data, project quality reporting	Analyse global quality data, quality reporting
Act	Act by deviation of the planing and if risk get real which could be a problem for the planed quality	Act if quality milestones not reached, adoption of quality processes, establishing of successful quality processes and methods

process improvements are:

- Commitment of the management
- Commitment of the project
- “Friendly” user in the project implementation
- Verification of the successful employment of ressources
- Appropriate ressources in the improvement phase.

Therefore it is important to define quality assurance processes based on human skills. In the phase of improvement a personal cooperation and support of planed quality assurance methods is a must. Therefore the organisation of the improvement team could be subdivided in an operational team and a process and method support team. The work products of the support team have to be published to reach the hole organisation in a high efficient and marketing conform manner. It is a basic task to establish the project banner as a well known and respected brand. Process improvement is primary to convince people to do their work in another way, e. g. standardised and process-oriented.

Conclusion

The question left from the introduction is “Do we need a new Henry Ford of automotive software engineering”? Standardisation of software moduls and the separation of hardware and software lifecycles seem to be a basic step for higher quality and cost efficient production. Nevertheless the adoption of needed engineering processes to establish new technologies in an

acceptable quality is an often secondary handled problem. This is very often the killer of the introduction of a new technology because of the deficient acceptance in the organisation. Henry Ford showed us that a high technology product could only be cost effective produced by standardisation, effective engineering and management processes and under convince of the human factor. Software construction of complex, safty critical systems could not be longer a job for individual artists, it has to be the job of trained and motivated engineers, even in a classical industry such as automotive.

