Software quality management in the automotive sector

Juraj Pančík,PhD., Aleš Vémola, PhD.,Robert Kledus. PhD., Marek Semela, PhD,Albert Bradáč, Ph.D.

{email: robert.kledus@usi.vutbr.cz, ales.vemola@usi.vutbr.cz, marek.semela@usi.vutbr.cz, albert.bradac@usi.vutbr.cz}

Institute of Forensic Engineers, Brno University of Technology, Purkyňova 464/118, 612 00 Brno, Czech Republic

Abstract. Software for automobiles is one from innovative factors in the automotive industry. Automobile is represented as an amount of embedded systems (embedded systems) and it is a very complex computing system. It is currently estimated that the average car has built-in software in the range of 100 million lines and in 2020 is already expected 300 million lines of code. The contribution is devoted to the management of safety and reliability of the software development for embedded systems designed for electromechanical (mechatronic) systems through quality assurance of embedded software. The contribution defines the term software quality assurance strategy, explains the role of standards such as ISO 26262 (Road vehicles - Functional safety), ISO 15504 (Automotive SPICE 3.0)

1 Introduction

The purpose of the contribution is to describe the cause of the auto recalls, to focus on software auto recalls associated with software errors, and to introduce some international standards to ensure automotive software quality.

Today's vehicles have evolved from a mechanical device into an integrated machine with embedded software powering grater performance in all major systems. These technological improvements are driving brand success stories, and consumer's experiences are shaped as much by the software and hardware. Now more than ever, software quality needs to be at the top of the list for major auto brands looking to preserve - and elevate - brand status. Controls and information systems require in a modern car 100 million lines of code that is more than a Boeing 787 (6.5 million lines). When a smartphone's software fails or a desktop computer's operating system crashes, it can cause a major inconvenience. But it pales in comparison to a vehicle software failure that could affect braking, acceleration or any number of functions while we're barreling down the highway at 100 km/h. That's why vulnerabilities in software-connected components – from internal malfunction to external hacking – have been the subject of increased attention by manufacturers, regulators and the public.

2 Auto recalls

An auto recall occurs when a manufacturer (or the National Highway Traffic Safety Administration (NHTSA) in the USA or another regulators) determines that a car model (or several models) has a safety-related defect or does not comply with a federal safety standard. When this happens, the automaker will alert owners to the problem and usually offer a free repair. Keep in mind that a recall doesn't mean that the entire vehicle will be replaced. NHTSA recorded 51 million vehicle recalls in 2015, slightly more than the 2014 total, which was adjusted downward from about 64 million to just under 51 million due to double-counting that occurred in 2014 related to the Takata recall (1). Through various announcements, the Takata recall has tripled in size over the past year. It is expected that the inflator recall will impact more than 42 million vehicles in the U.S., with the total number of airbags being between 65 and 70 million. Some key statistics about the 2015 recalls (2):

- There were almost 900 separate recall actions, nearly 100 more than the previous year.
- Takata air bag inflators were linked to approximately 42 percent of recalled vehicles in 2015 (more than 6.2 million) (1).
- The largest non-Takata recall of 2015 was issued by Toyota, related to a power window electrical switch that could short-circuit and potentially catch fire. It affected more than 1.8 million units.

The Figure 1 shows overall recall trends and unique campaigns and units affected by decade

Last year's most publicized story about an automobile defect was when the Environmental Protection Agency (EPA) cited Volkswagen for bypassing the emissions control system in almost 500,000 vehicles sold in the U.S. (and many more globally) (3). The EPA issued a notice of violation to the automaker, letting it know that the vehicles were discharging more pollutants than legally acceptable (4). Indications are that this could be the most expensive recall ever, topping out at $14.7 billion USD (5). The completion of this recall will likely be much more challenging than a typical safety recall. Once Volkswagen figures out how to fix the issue in accordance with EPA standards, it will then need to entice people to return to dealerships for the repair. However, owners

MMS CONFERENCE 2017, November 22-24, Starý Smokovec, Slovakia

DOI 10.4108/eai.22-11-2017.2274781

might resist the fix, particularly if it will have a negative effect on performance and gas mileage. As a result, VW will need strong incentives and proactive outreach to bring people into dealerships

3 Software recalls and technical service bulletins

There are few recorded examples of automotive software faults in last years:

- 2014: Honda is recalling 175,356 gas-electric hybrid vehicles, including its popular Fit subcompact, over a software glitch in the engine control unit that puts the vehicle at risk of moving or speeding abruptly (6).
- 2014: Nissan told the National Highway Traffic Safety Administration that a software problem in the occupant classification system (OCS) of several models might cause airbags to not deploy in the event of a crash, prompting a 990,000 vehicle recall (7).
- 2015: Security experts identified a vulnerability that would allow a hacker to remotely control the entertainment system in a 2015 Jeep Cherokee, giving them access to various electronic control units in the vehicle. In response, FCA recalled 1.4 million vehicles equipped with 2013–2015 UConnect head unit systems (8).
- 2015: Jaguar Land Rover recalled approximately 65,000 Range Rover sport utility vehicles after discovering that a keyless entry software glitch caused some of the vehicles' doors to fly open unexpectedly, which could distract drivers or cause a crash (9).
- 2016: Volvo Car Group recalled 59,000 cars after some owners experienced their engines stopping and restarting while they were driving. Dealerships were asked to correct the software fault, which had not led to any accidents (10)
- 2016: The tragic news of Star Trek actor Anton Yelchin, crushed to death when his Jeep Grand Cherokee rolled backward down his driveway, has prompted Fiat Chrysler to speed up its recall plans to modify electronic gearshifts on more than 1.1 million Jeep and Dodge vehicles. The recall has been linked to hundreds of reported accidents and injuries (11).
- 2016: Nissan disabled the Nissan Connect app that allowed Leaf owners to control the vehicle's climate system, after a security expert identified a vulnerability that could allow hackers to access the Leaf's temperature control and download its driving log (12).
- 2017: Tesla recalls 53,000 cars over brake issue. Tesla has issued a voluntary global recall for some of its Model S and Model X cars to fix a problem with the electronic parking brake (EPB).The electric car maker said about 2% of the 53,000 vehicles built from February to October 2016 were affected, but all of those cars are being recalled. The company added it had no reports of accidents or injuries relating to the brake issue. In US trading, Tesla shares closed down 1% at $302.51 (13).

Recalls of software-related components and found that they have dramatically increased in the last few years. Since the end of 2012, there has been a marked increase in recall activity due to software issues. For the primary light vehicle makes and models we studied, 32 unique software-related recalls affected about 3.6 million vehicles from 2005 – 2012. However, in a much shorter time period from the end of 2012 to June 2015, there were 63 software-related recalls affecting 6.4 million more vehicles. From less than 5 percent of all recalls in 2011, software related recalls have risen to almost 15 percent in 2015. Overall, the amount of unique campaigns involving software has climbed dramatically, with nine times as many in 2015 than in 2011, as both Figure 32 and Figure 3 indicate. Over the years, more and more components rely on an automobile's internal computers instead of traditional analog systems. Such components include fuel mixture management, automatic braking, air bag sensors, and seats that detect the driver's weight and position. All have the potential to fail. Figure 4 shows trend to "subtilization" of source of software faults - the number of control units in the car is growing every year and this fact increases the probability of a software error of a specialized ECU. In 2011 only three software-related components were involved in recalls. In 2015, 20 automotive components were affected by software-related recalls. NHTSA has added several automotive component categories for Early Warning Statistics (EWR) reporting, including forward collision avoidance and automatic brake controls. According to the Insurance Institute for Highway Safety (IIHS), autonomous collision avoidance technology is being offered by as many as 22 OEMs as of January 2016. In 2015, three new software-related categories reported data for the first time (2):

- Automatic Braking, listed on 21 EWR reports, resulting in 26 injuries and 1 fatality
- Electronic Stability, listed on 6 EWR reports, resulting in 7 injuries and 1 fatality
- Forward Collision Avoidance, listed in 1 EWR report, resulting in 1 injury and no fatalities

In addition to software recalls, researchers discovered an increase in software related Technical Service Bulletins (TSB), which identify issues with specific components, yet stop short of a recall. TSBs are issued when

manufacturers provide recommended procedures to dealerships' service departments for fixing problematic components (120 unique TSBs in 2014 and 2015 years).

4 Some international standards to ensure of automotive software quality

4.1 Systems development life cycle (SDLC) and V - model

The systems development life cycle (SDLC), also referred to as the application development life-cycle, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system. The systems development lifecycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.

The V-model is a graphical representation of a systems development lifecycle. It is used to produce rigorous development lifecycle models and project management models. The V-model falls into three broad categories, the German Das V-Modell, a general testing model and the US government standard. The V-model summarizes the main steps to be taken in conjunction with the corresponding deliverables within computerized system validation framework, or project life cycle development. It describes the activities to be performed and the results that have to be produced during product development. The left side of the "V" represents the decomposition of requirements, and creation of system specifications. The right side of the V represents integration of parts and their validation .However, Requirements need to be validated first against the higher level requirements or user needs. Furthermore, there is also something as validation of system models (e.g. FEM). This can partially be done at the left side also. To claim that validation only occurs at the right side may not be correct. The easiest way is to say that verification is always against the requirements (technical terms) and validation always against the real world or the user needs.

4.2 Systems and software engineering: ISO/IEC 12207

The ISO/IEC 12207 Systems and software engineering – Software life cycle processes is an international standard for software lifecycle processes (14). It aims to be the standard that defines all the tasks required for developing and maintaining software. The ISO/IEC 12207 standard establishes a process of lifecycle for software, including processes and activities applied during the acquisition and configuration of the services of the system. Each Process has a set of outcomes associated with it. There are 23 Processes, 95 Activities, 325 Tasks and 224 Outcomes (the new "ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes" defines 43 system and software processes).The standard has the main objective of supplying a common structure so that the buyers, suppliers, developers, maintainers, operators, managers and technicians involved with the software development use a common language. This common language is established in the form of well-defined processes. The structure of the standard was intended to be conceived in a flexible, modular way so as to be adaptable to the necessities of whoever uses it. The standard is based on two basic principles: modularity and responsibility. Modularity means processes with minimum coupling and maximum cohesion. Responsibility means to establish a responsibility for each process, facilitating the application of the standard in projects where many people can be legally involved. The set of processes, activities and tasks can be adapted according to the software project. These processes are classified in three types: basic, for support and organizational. The support and organizational processes must exist independently of the organization and the project being executed. The basic processes are instantiated according to the situation.

4.3 Automotive SPICE: ISO/IEC 15504 and ISO/IEC 33001

ISO/IEC 15504 Information technology – Process assessment, also termed Software Process Improvement and Capability Determination (SPICE), is a set of technical standards documents for the computer software development process and related business management functions. It is one of the joint International Organization for Standardization (ISO) and International Electro technical Commission (IEC) standards, which was developed by the ISO and IEC joint subcommittee, ISO/IEC JTC 1/SC 7.ISO/IEC 15504 was initially derived from process lifecycle standard ISO/IEC 12207 and from maturity models like Bootstrap, Trillium and the Capability Maturity Model (CMM).ISO/IEC 15504 has been revised by: ISO/IEC 33001:2015 Information technology – Process assessment – Concepts and terminology as of March, 2015 and is no longer available at ISO.

Automotive SPICE (Software Process Improvement and Capability Determination) is a process maturity framework (model) to assess the capability and maturity of organizational processes to develop software resp. embedded systems in the automotive industry. It is a variant of ISO 15504 tailored to the needs of the automotive industry. The framework is used by automotive OEMs and suppliers to assess the capability and maturity of their development processes for software and embedded systems. The process reference part of the model defines the

central processes relevant to be inspected and to be performed in any software/embedded system development. The process assessment part of the model describes how to evaluate the capability of processes within the organization.

A maturity model is an organizational comparison tool designed to evaluate an organizations methods and processes against industry best practices; and based upon the results, provide a maturity rating (which is effectively a process and capability rating), enabling organizations to determine supplier suitability. Since 2005 when the Automotive SPICE model was published, many car manufactures have adopted ASPICE to evaluate both software and electronics suppliers. Key to the success of ASPICE is the scope of the model, accounting for domain specific models within an overall umbrella model. As we show later ASPICE maturity model is a requirement base for embedded software automotive suppliers according to the upcoming manufacturing quality management system in the automotive industry according to the IATF 16949.

4.4 Software engineering — Product quality: ISO/IEC 9126 and ISO/IEC 25010:2011

ISO/IEC 9126 Software engineering — Product quality was an international standard for the evaluation of software quality. It has been replaced by ISO/IEC 25010:2011. The fundamental objective of the ISO/IEC 9126 standard is to address some of the well-known human biases that can adversely affect the delivery and perception of a software development project. These biases include changing priorities after the start of a project or not having any clear definitions of "success". By clarifying, then agreeing on the project priorities and subsequently converting abstract priorities (compliance) to measurable values (output data can be validated against schema X with zero intervention), ISO/IEC 9126 tries to develop a common understanding of the project's objectives and goals. The standard is divided into four parts: quality model, external metrics, internal metrics and quality in use metrics. The quality model presented in the first part of the standard, ISO/IEC 9126-1 classifies software quality in a structured set of characteristics and sub-characteristics as follows:

- Functionality - "A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs." [Suitability, Accuracy, Interoperability, Security, Functionality compliance].
- Reliability - "A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time." [Maturity, Fault tolerance, Recoverability, Reliability compliance].
- Usability - "A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users."[Understandability, Learnability, Operability, Attractiveness, Usability compliance].
- Efficiency - "A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions."[Time behavior, Resource utilization, Efficiency compliance]
- Maintainability - "A set of attributes that bear on the effort needed to make specified modifications." [Analyzability, Changeability, Stability, Testability, Maintainability compliance].
- Portability - "A set of attributes that bear on the ability of software to be transferred from one environment to another." [Adaptability, Install ability, Co-existence, Replace ability, Portability compliance]

4.5 Functional safety - road vehicles: ISO 26262 and IEC 61508

Functional safety features form an integral part of each automotive product development phase, ranging from the specification, to design, implementation, integration, verification, validation, and production release. The standard ISO 26262 is an adaptation of the Functional Safety standard IEC 61508 for Automotive Electric/Electronic Systems. ISO 26262 defines functional safety for automotive equipment applicable throughout the lifecycle of all automotive electronic and electrical safety-related systems. The first edition, published on 11 November 2011, is intended to be applied to electrical and/or electronic systems installed in "series production passenger cars" with a maximum gross weight of 3500 kg. Draft of new edition of this standard was published at end 2016. It aims to address possible hazards caused by the malfunctioning behavior of electronic and electrical systems. Although entitled "Road vehicles – Functional safety" the standard relates to the functional safety of Electrical and Electronic systems, not to that of systems as a whole or of their mechanical subsystems. Like its parent standard, IEC 61508, ISO 26262 is a risk-based safety standard, where the risk of hazardous operational situations is qualitatively assessed and safety measures are defined to avoid or control systematic failures and to detect or control random hardware failures, or mitigate their effects. Goals of ISO 26262:

- Provides an automotive safety lifecycle (management, development, production, operation, service, decommissioning) and supports tailoring the necessary activities during these lifecycle phases.

- Covers functional safety aspects of the entire development process (including such activities as requirements specification, design, implementation, integration, verification, validation, and configuration).
- Provides an automotive-specific risk-based approach for determining risk classes (Automotive Safety Integrity Levels, ASILs).
- Uses ASILs for specifying the item's necessary safety requirements for achieving an acceptable residual risk.
- Provides requirements for validation and confirmation measures to ensure a sufficient and acceptable level of safety is being achieved.

4.6 Manufacturing quality management system in the automotive industry: ISO/TS 16949 and IATF 16949

The ISO/TS16949 is an ISO technical specification aimed at the development of a quality management system that provides for continual improvement, emphasizing defect prevention and the reduction of variation and waste in the automotive industry supply chain. It is based on the ISO 9001 standard and the first edition was published in June 1999 as ISO/TS 16949:1999. It was prepared by the International Automotive Task Force (IATF) and the "Technical Committee" of ISO. It harmonizes the country-specific regulations of quality Management systems. About 30 percent of the more than 100 existing automobile manufacturers affiliate the requirements of the norm but especially the large Asian manufacturers have differentiated, own requirements for the quality management systems of their corporate group and their suppliers. TS16949 applies to the design/development, production and, when relevant, installation and servicing of automotive related products. The requirements are intended to be applied throughout the supply chain. For the first time vehicle assembly plants will be encouraged to seek ISO/TS16949 certification.

On October 3rd, 2016 IATF 16949:2016 was published by the IATF and supersedes and replaces the current ISO/TS 16949, defining the requirements of a quality management system for organizations in the automotive industry. Deadline for transition from ISO/TS 16949 becomes IATF 16949 is 14 September 2018. In addition to ISO 9001:2015, besides another requirements the new requirements is expected for products with embedded software. This new clause adds requirements for organization-responsible embedded software development and software development capability self-assessments. Organizations must use a process for quality assurance of products with internally developed embedded software, and have an appropriate assessment methodology to assess their software development process. The software development process must also be included within the scope of the internal audit program; the internal auditor should be able to understand and assess the effectiveness of the software development assessment methodology chosen by the organization like in previous text mentioned Automotive SPICE.

5 Conclusions

As resume of this contribution we should emphasis next ideas:

- Today’s automotive software is very complex and huge (100 million lines of code). The human’s safety and life depends at the quality of the automotive software. Each automotive software should be conform to functional safety standard for road vehicles ISO 26262. Internal and external assessors are performing estimation component’s functional safety for each automotive components with embedded software.
- By automotive software faults generated car’s recalls growth every year (total number of recall units and unique campaigns). Their relative ratio at this moment is about 15% and its trend is to be higher. Automotive software errors and subsequent car’s recalls cause serious financial and moral losses.
- The developing of automotive components including automotive software is performed in regulated environments. One side of this regulation environment is represented by state regulator (National Highway Traffic Safety Administration (NHTSA) in the USA), next side is represented by automotive OEM (Original Manufacturing Component) component‘s suppliers (contractors) together with theirs clients – the automakers companies. Each supplier of automotive software should performed its software developing processes with conformance with automaker required software developing process capability level (according to the Automotive SPICE, standard ISO/IEC 33001). Internal and external assessors are performing estimation process capability level for each automotive components with embedded software.
- The development of automotive software is performed under control and supervision many industrials standards. The automotive embedded software is also a part of latest automotive manufacturing and supply chain quality standard IATF 16949. Automakers and their OEM supplier’s deadline for transition to standard IATF 16949 is September 14. 2018.

6 References

[1] Takata Airbag Recall - Everything You Need to Know. CR Consumer Reports. [Online] 6 4, 2017. [Cited: 6 27, 2017.] http://www.consumerreports.org/cro/news/2016/05/everything-you-need-to-know-about-the-takata-air-bag-recall/index.htm.
[2] STEINKAMP, N.: SRR 2016 Automotive Warranty & Recall Report. [Online] 6 27, 2017. http://www.prnewswire.com/news-releases/srr-2016-automotive-warranty--recall-report-reveals-spike-in-software-related-recalls-explores-emerging-risks-such-as-hacking-data-breaches-300256836.html.
[3] Volkswagen Light Duty Diesel Vehicle Violations for Model Years 2009-2016. [Online] EPA, 2017. [Cited: 6 28, 2017.] https://www.epa.gov/vw.
[4] Volkswagen Clean Air Act Civil Settlement. [Online] EPA, 2016. [Cited: 06 28, 2017.] https://www.epa.gov/enforcement/volkswagen-clean-air-act-civil-settlement.
[5] Volkswagen to Spend Up to $14.7 Billion to Settle Allegations of Cheating Emissions Tests and Deceiving Customers on 2.0 Liter Diesel Vehicles. [Online] 6 28, 2016. [Cited: 6 28, 2017.] https://www.epa.gov/newsreleases/volkswagen-spend-147-billion-settle-allegations-cheating-emissions-tests-and-deceiving.
[6] Software Glitch in Electronic Controls Units Springs Honda Recall. [Online] VECTOR, 7 14, 2014. [Cited: 6 28, 2017.] https://www.vectorcast.com/news/embedded-software-testing-news/automotive-software-news/software-glitch-electronic-controls.
[7] NISSAN: [Online] 4 11, 2014. [Cited: 6 28, 2017.] https://www-odi.nhtsa.dot.gov/acms/cs/jaxrs/download/doc/UCM452755/RCDNN-14V138-1942P.pdf.
[8] GREENBERG, A.: HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT. [Online] 7 21, 2015. [Cited: 6 28, 2017.] https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.
[9] Jaguar Recalls 65K Range Rovers Over Door Latch Issue. [Online] LexisNexis Company, 7 8, 2015. [Cited: 6 29, 2017.] https://www.law360.com/articles/676832/jaguar-recalls-65k-range-rovers-over-door-latch-issue.
[10] Volvo Cars recalls 59,000 cars over software fault. [Online] REUTERS, 2 21, 2016. [Cited: 6 29, 2017.] http://uk.reuters.com/article/uk-volvocars-recall-idUKKCN0VT0SY?type=companyNews.
[11] MCCAFFEREY, B.: Functional Safety Challenges for Automotive Systems. [Online] VECTOR, 6 28, 2016. [Cited: 6 28, 2017.] https://www.vectorcast.com/blog/2016/06/functional-safety-challenges-automotive-systems.
[12] ENGADGET: Nissan disables its Leaf remote control app (update). [Online] Oath Inc, 2 24, 2016. [Cited: 6 29, 2017.] https://www.engadget.com/2016/02/24/nissan-leafs-connected-climate-control-has-a-security-flaw/.
[13] BBC: Tesla recalls 53,000 cars over brake issue. [Online] BBC , 4 21, 2017. [Cited: 6 29, 2017.] http://www.bbc.com/news/business-39663382?ocid=socialflow_twitter.
[14] ISO/IEC 12207, Wikipedia [Online] 2017. [Cited: 11 6, 2017.] https://en.wikipedia.org/wiki/ISO/IEC_12207.

Figure 1 Overall recall trends. Unique units affected by decade

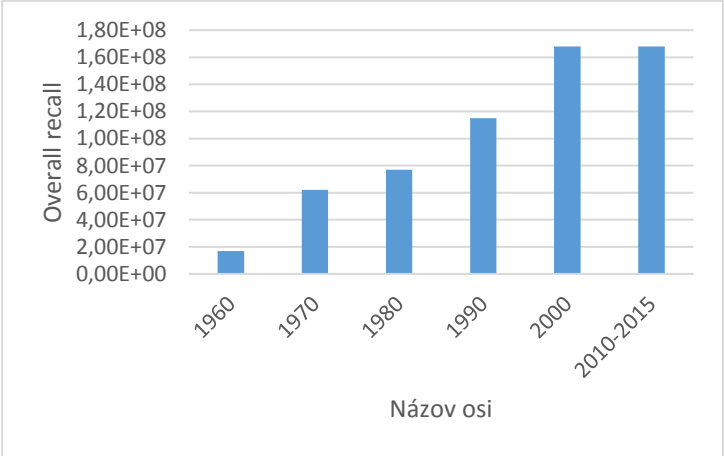

Figure 2 Overall recall trends. Unique campaigns affected by decade

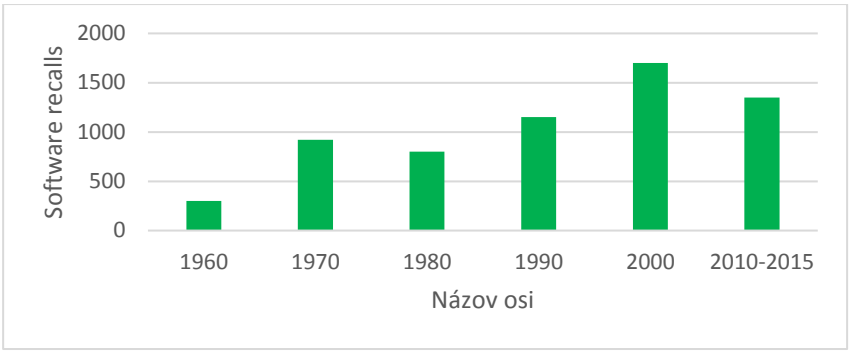

Figure 3 Summary of software recalls by year (2006-2015) – unique campaigns

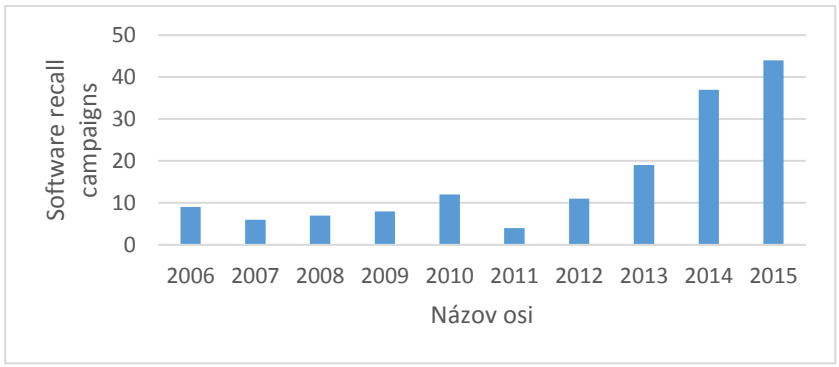

Figure 4 Percentage of software recall campaigns according ECU software (years 2006-2015)

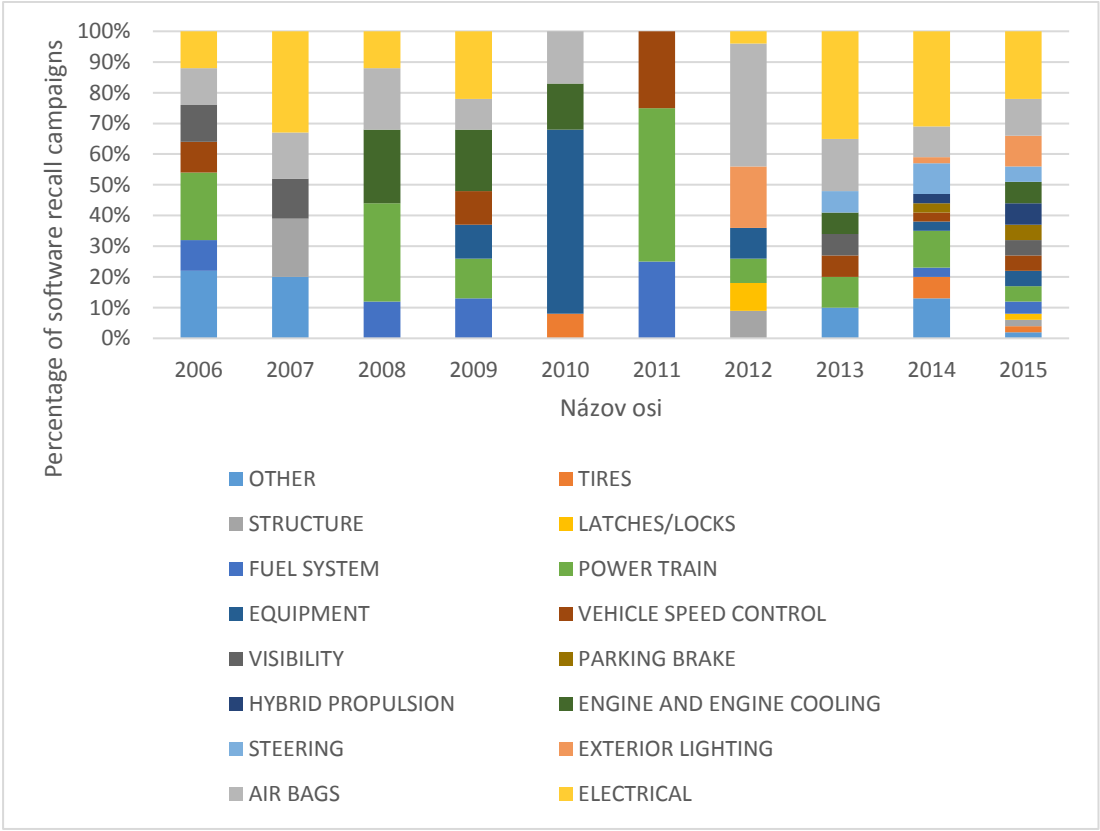

Figure 5 Systems Development Life Cycle (SDLC). Source: Adopted from Wikipedia

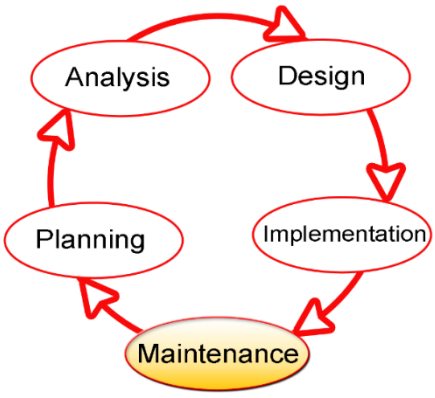

Figure 6 Software development V-model Source: Adopted from wikipedia