# Assignment 2, WASP Software Engineering Course

ali nouri

November 2022

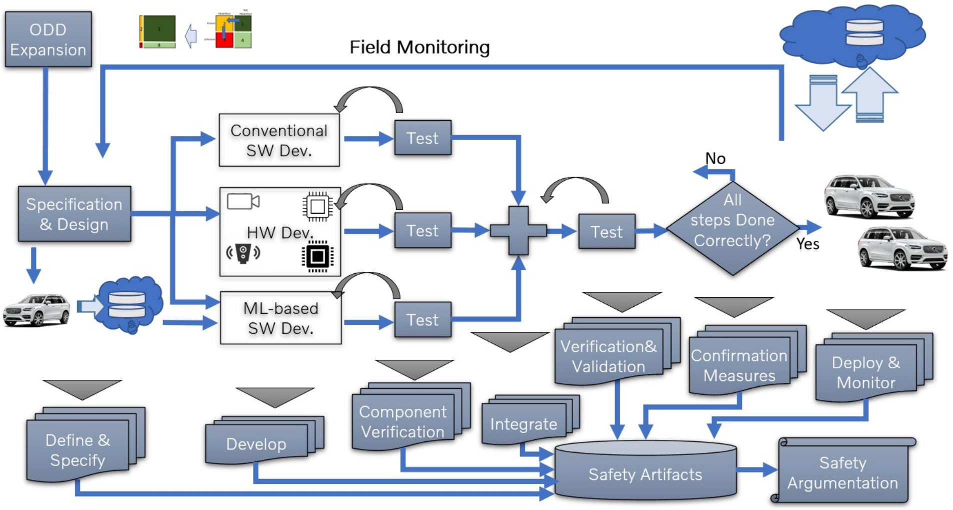

Figure 1: Iterative development process for autonomous driving and its contribution to safety argumentation.

## 0.1 My Research

Autonomously driven (AD) function is a promising technology toward safer roads, by preventing accidents caused because of human errors. However, due to immature technology, and complexity of the system, they can be the cause of hazardous behavior. So not only there is a need for extra measures and safety concepts to have a safe AD, but also to show and argue for safety of the system to gain the trust of the stakeholders. Fig. 1 presents the abstract of AD development process.

Arguing for safety of complex systems, which changed rapidly and continuously (so called DevOps), is not possible by using the traditional methods. So, we gathered all design and argument related challenges of having a safe AD in DevOps Ecosystem as following [1]:

**CH1:** Impact analysis and tailoring before each iteration

**CH2:** Requirement updates (FUSA, SOTIF, or Cyber-Security)

**CH3:** Change in one discipline affects others

**CH4:** Hardware limitations

**CH5:** Safety analysis methods

**CH6:** Software architecture

**CH7:** Verification

**CH8:** Validation

**CH9:** Safety argumentation

The challenges will be studied in this PhD and possible solutions would be proposed. Volvo Cars and Zenseact together with Chalmers University of Technology are partners in this PhD project.

## 0.2 Regulations and Compliance

Depend on the context and application of each specific software there would be regulations in place. It could be as generic as General Data Protection Regulation (GDPR) [2] or more specific regarding safety or cyber-security aspects of the application. In my research there are multiple regulations to be considered from more generic to more specific ones. UN Regulation No. 157 or Automated Lane Keeping Systems [3] is one of the more specific regulations which uniquely defined for safety of AD. Any automotive OEM shall consider ALKS and ask certification parties to assess and audit their product and process in order to be able to sell AD functionality in Europe. ALKS is referring to other regulations and standards which are taking care of different aspects of AD like safety, cyber-security, and privacy. There are other regional regulations as well which are more granular than ALKS and impose more restrictions on the process or on the function. For example in some European countries the feature expansion can't be done after the certification (except minor bug fixings), and if needed for each vehicle there is a need for re-certification which make it impossible to do continuous improvement.

## 0.3 Requirements Engineering

As it is shown in Fig. 1 the development starts with requirements specification, which consists of several abstraction levels [1]. Due to complexity of these systems, there is a need for distributed development, considering intellectual property of the stakeholders as well, which leads to several abstraction levels of requirements. The requirements not only specify what is expected from the suppliers or development teams, but also it would be an input for verification and validation of the implemented components or subsystems. In our research the focus is on safety aspects in the contexts of ISO 26262 (functional safety) [4] or ISO 21448 (safety of intended functionality or SOTIF) [5]. Functional safety requirements take care of hazardous events caused by systematic software failures, and SOTIF is taking care of technology limitations or inefficiencies of specifications.

As it is also shown in Fig. 1, the conventional software development has a separate path than the ML based software. In ML based software development the requirement specification can't go further to the black box of ML component to define the more granular behaviour of the software and data plays the role of requirements from there onward.

## 0.4 Continuous Evolution, Maintenance and Deployment

Fig. 1 shows the iteration loops in the development of AD which is not only to fix the bugs but also to improve the function. At start the RD decide on the ambition which I name it most lovable product, and then due to technology limitations or limited development effort, reduce it to minimum viable product. After some bug fixing and performance improvement iterations the feature improvement starts and then the minimum lovable product will be designed and implemented. In our context it is a highway pilot which can be activated in limited road segments and limited periods during the day. This process would be repeated continuously to reach the maximum lovable product, which in this context is an autonomous driving function which can be activated everywhere.

Continuous deployment does not stop since due to reasons like distributional shift in the operational situation, there would be a need for continuous data collection and improvement of ML based software.

## 0.5 future trends

Due to unbounded complexity of the operational situation of AD, ML based software development is one of the solutions to be used in the system to handle this complex task. But due to stochastic behavior and uncertainties in the ML based software, using them in safety critical applications would be challenging.

As it is mentions unlike conventional software, which the requirement specification will be done till the last unit level with defining the exact behavior, there is no possibility to specify more granular behavior of ML based software, and then argue for completeness of verification. This is why the data is very important in development of these software components. Although due to technology limitation the usage of ML based software in safety critical systems is limited at this time, by improvement in the technology (both hardware and software) they will be used more even in safety critical function in near future.

One solution is to combine ML based software by deterministic software to monitor, or safe guard them. In this way the ML based component takes care of non safety related aspects, and the conventional software inherit the safety requirements and integrity.

All in all at least in near future the software architecture consists of both ML and non-ML based based software, since each have its own strength.

# Bibliography

[1] A. Nouri, C. Berger, and F. Törner, “An Industrial Experience Report about Challenges from Continuous Monitoring, Improvement, and Deployment for Autonomous Driving Features,” in *Proceedings of the 47th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*, (Maspalomas, Gran Canaria, Spain), Sep. 2022.

[2] Dr. Cristian Klein, “Regulatory compliance: Bridging the gap between legalease and software engineering,” 2022.

[3] UNECE, “UN Regulation No. 157 - Automated Lane Keeping Systems (ALKS).” https://unece.org/transport/documents/2021/03/standards/un-regulation-no-157-automated-lane-keeping-systems-alks, 2021.

[4] “ISO 26262:2018 (all parts), Road vehicles — Functional safety,” Standard, International Organization for Standardization, 2018.

[5] “ISO/FDIS 21448, Road vehicles — Safety of the intended functionality,” Standard, International Organization for Standardization, 2022.