Simulation as Tool

Continuous Software Testing with Simulation while Development Phase

Matthias Burg, Continental Teves AG & Co. oHG
Outline

1. Why software testing is important
2. Simulation strategy: Continuous Testing
3. Outcome and prerequisites
The role of software within development

- Mechatronic systems realize functions as e.g. braking, driver assistance systems or automated (cooperative) driving

Mechatronic System = Hardware + Electronics + Information Technology
Architecture: Divide and Conquer

Vehicle

Automated driving system

Camera

Electronic computing unit

Software

Electronic computing unit hardware

Software

…

Lens

Chassis

Electronics

Software
How components are typically developed

- Each software is placed on a hardware component: Without hardware no execution of software is possible. This is called context of software.
- For this reason, while developing software, it is meaningful to avoid changes of hardware which is in the context of the software. Best practice is to freeze hardware development while software development is ongoing.

Component development lifecycle

Software

Hardware
Simulation – What does this mean?

Definition of simulation:

Simulation is the implementation of a model or models in a specific environment that allows the model's execution or use over time.


Many reasons and characteristics to conduct simulation:

- Grades of model accuracy
- Architecture hierarchy level
- Purpose e.g. verification, validation or proof of design concept
Generic simulation approach

- The simulation environment consists of models that generate input or receive output of the System under Test (SuT e.g. a software component).

- Open loop vs. closed loop:
  - Closed loop considers feedback of the SuT

- Virtualization of the outer environment is utilized to test the SuT
Purpose driven tailoring of simulation approaches

- **Goals:**
  - Enable *agile development* by a continuous testing environment
  - Show maturity of the vehicle system
  - Give *early feedback* to developers and stakeholders
  - Enable failure localization

- **Election of simulation approaches according to added value effort**

- **Adapt simulation strategy to project boundary conditions: available**
  - Manpower
  - Resources
  - Working process
Simulation strategy

- Simulation is utilized at different architecture hierarchy levels: After each successful test, the next higher level may be reached.

1. After software build, simulation is used to create stimuli for unit tests.
2. At system component level simulation is used for an integration test.
3. At system level a closed loop vehicle simulation is used to verify functionality.
Continuous testing

- Continuous testing describes a method which aims to give early feedback about software development from source code level to product level.

- Automated execution of
  - Software build
  - Tests
  - Analysis
  - Reporting to stakeholders
Continuous testing workflow

GitHub

Developer

Jenkins

Gate 1: Static and Unit tests

Gate 2: Integration ("smoke") test

Gate 3: System test

Regression test

New Code?

Compile

Yes

Yes

Yes

Yes

Yes

Yes

No

No

No

No

Gate 1 successful?

Gate 2 successful?

Gate 3 successful?

Observes for changes

Effort

Simulation Report

September 19th & 20th, 2018

Ko-HAF – Simulation as Tool

11
Outcome: How simulation can help

- By this approach, early feedback could be given to developers.
- Failures could be localized easier.
- Real vehicle testing is done only with high mature software.

Outcomes:

- Check for functional failures at vehicle level.
- Check for interface failures after integration of all software modules.
- Check for failures at software module level.
- Perform a complex regression test only with release candidates.
Requirements to Continuous testing

- **Enabler:**
  - Requirements of System of Interest (SoI) available
  - Architecture of system available
  - Interfaces of System under Test (SuT) defined

- Simulation environments for all gate checks must be available
- Configuration management in order to manage code
- Automated evaluation and analysis of simulation results
- The process has to be lived by all project members
Thank you for your attention!

The contents of this presentation (including but not limited to texts, images, photos, logos, etc.) and the presentation itself are protected by intellectual property rights. They were created by the project consortium Ko-HAF and/or licenced by the project consortium. Any disclosure, modification, publication, translation, multiplication of the presentation and/or its contents is only permitted with a prior written authorisation by the consortium.

© Copyright Project Ko-HAF, 2018, Contact: projektbuero@ko-haf.de