AUTOMATED SYSTEM TESTING TOWARDS CONTINUOUS INTEGRATION IN EMBEDDED ENVIRONMENTS (AUTOMOTIVE INDUSTRY)

Presented by Sebastian Dengler and Patrick Wunner
Agenda

Motivation
- Continuous Integration in a nutshell
- Development Life Cycle

Integration Concept
- Concept
- Implementation

Experience
- Issues we faced
- In progress
- Success
MOTIVATION

Continuous Integration
Development Life Cycle
Continuous Integration

In a nutshell:
- automated build process (e.g. Jenkins)
- automated testing (on several stages)
- automated fast feedback about quality

Advantages:
- Reduce risks
- Assure quality
- Increase confidence
Development Life Cycle

- Automated Build
- Integration Control Board
- Bug-Fixes and Implementation of new Features
- Automated Test Start
- Fast Feedback
- Detailed Analysis
INTEGRATION CONCEPT

How to integrate Hardware-in-the-Loop testing into a Continuous Integration environment
Integration Concept

- Control
  - Automated Build (Jenkins)
  - Jenkins
  - scheduled

- Preparation
  - Automated Preparation Test Environment

- HIL

- Feedback
  - Results
    - HTML Report
    - Formatted for Jenkins Plug-In
  - Presentation Results (Jenkins)

- Flash Test
- Initialisation
- Smoke Test
- Complete Test Set

© All rights reserved
Control

Test Start by Jenkins
- After a successful software build process
- Smoke tests with immediate feedback
- Complete (regression) test set when smoke tests are passed

Test Start by Test Team
- Every software at every time
- Select single tests due to test reasons
- Export results to Jenkins or only use them locally
Preparation HiL Test Systems

Setup Environment
- Copy software to local folder on test system
- Select correct software branch according to software version
- Synchronize test scripts and environment with repository

Preparation
- Start correct environment simulation model (dSpace, Canoe, ...)
- Prepare local files and third party software (A2L, Canape, ...)
- Flash software to ECU
Hardware-in-the-Loop

**Basic Idea:**
- Fast feedback for development team
- Less effort for test team to provide results
- More time for test development
- Save resources for effective testing

**Flowchart:**
1. Automated Build
2. Automated HiL Preparation
3. Smoke Test
4. Failed: Send email to build manager, ...
5. Abort all other tests
Hardware-in-the-Loop

- Automated Build
- Export test results to Jenkins
- Smoke Tests
- Run complete (regression) test set
- Automated HiL Preparation
- Passed
  Smoke Tests: Send email to build manager and team
Simultaneous Test Starts on several Test Systems

Features
- Centralised control for simultaneous test starts
- Parallel test execution on several test systems
- Centralised feedback
Feedback

- Fast feedback for management
  - Management Summary (database)
  - Visualise current status and trends

- Fast feedback for development team
  - automated summary emails
  - different levels of detail
  - different recipients (sw managers, competence leads, ...)

- Visualisation Results
  - Evaluation
  - Jenkins (Allure)
  - Statistics
EXPERIENCE

Issues we faced, progress we made and goals we achieved
Issues we faced

Test System
- Several development branches
- Initialise real time system (reload model)
- Configure third party systems and software (e.g. dSpace Control Desk NG)

Environment (IT/Administration)
- Approval for sending automated emails (test system / shared user account)
- Access confirmation for HiL control computer to connect to software folders
- Access confirmation for HiL control computer to connect to repository

Development Process
- Automated build process
- Test implementation on all stages available in time
- Software designed for testability
In Progress

- Statistical Overview
  - Smoke Test
  - Complete (regression) test set
  - Test results across several test systems (benefits: regression and trend analysis, dependencies, fault propagation analysis)
  - Test results across several software versions (benefits: regression and trend analysis, dependencies, fault propagation analysis)

- Results Database (final version)
- Automatically reload simulation model according to software branch
- Selection of single test cases and test sequences by Jenkins interface
Achievements

- Complex automotive project in maintenance phase can be tested by ½ (software) test manager
- Stable test results in nightly test runs
- Nightly tests possible without late and early shift
- Reduced test start time down to 20%
- Test engineers can concentrate on test design and development
- Software engineers get direct feedback from HiL test system
- Staged software and system test strategies possible
- Reduced errors due to complex manual test starts
- iTestStudio (iSyst) completely integrated into CI process
Thank you for your attention.