Agile approach to development and validation of AUTOSAR components

Dr. J. Albesa and Dr. E. Valencia
IDNEO Technologies

MATLAB EXPO 2019
Agenda

1. IDNEO introduction
2. Automotive software development process
3. Towards agile practices
4. Tools and methods
5. Projects and products
6. Conclusions
Engineering By IDNEO is to timely adapt with critical thinking, method and processes to an ever-changing environment while continuously delivering value to our customers at sustainable pace.
Automotive Software Development Process

A-SPICE Model

System Engineering Process Group (SYS)
- SYS.1 Requirements Specification
- SYS.2 System Requirements Analysis
- SYS.3 System Architectural Design
- SYS.4 System Integration and Test
- SYS.5 System Qualification Test

Software Engineering Process Group (SWE)
- SWE.1 Software Requirements Analysis
- SWE.2 Software Architectural Design
- SWE.3 Software Detailed Design and Unit Construction
- SWE.4 Software Unit Verification
- SWE.5 Software Integration and Test
- SWE.6 Software Qualification Test

Supporting Process Group (SUP)
- SUP.1 Quality Assurance
- SUP.2 Verification
- SUP.3 Configuration Management
- SUP.4 Joint Review
- SUP.5 Problem Resolution Management
- SUP.6 Change Request Management
- SUP.7 Documentation

Management Process Group (MAN)
- MAN.3 Project Management
- MAN.4 Risk Management
- MAN.5 Measurement

HIS Scope

Performance
Quality
Speed
Efficiency
Cost

Primary Life Cycle Processes
Organizational Life Cycle Processes
Supporting Life Cycle Processes

Acquisition Process Group (ACQ)
- ACQ.3 Contract Agreement
- ACQ.4 Supplier Monitoring
- ACQ.11 Technical Requirements
- ACQ.12 Legal and Administrative Requirements
- ACQ.13 Project Requirements
- ACQ.14 Request for Proposals
- ACQ.15 Supplier Qualification

Supply Process Group (SPL)
- SPL.1 Supplier Tendering
- SPL.2 Product Release

MATLAB EXPO 2019
Automotive shift towards Agile

IDNEO continuously inspects and adapts our products, processes and deliverables to Client Needs.

Industry trend to shorter development time and fast design iterations pave the road for agile practices adoption.
Agile in Automotive Software Development?

- Customer collaboration over contract negotiation
- Responding to change over following a plan
- Working software over documentation
- People over processes and tools

- Quality and maintainability require adhesion to established and audited processes.

Millions of units in the field to be maintained 10+ years!
Model-Based Design can help

Model Based Development is a established methodology which helps to agility...

- Abstraction from HW, early error detection, fast prototyping and design iterations, improved communication, ...

...but is not sufficient:

- Tools are required to ease process execution and produce process evidences.

- MathWorks products help from early design phases and design validation, up to coding and code verification (@IDNEO since 2015).
Design phase
From architecture to design

- SW modules defined in architecture using AUTOSAR authoring tool
- Module definition (ARXML artifact) imported into Simulink
- Model scaffolding automatically generated
- Design engineers fill in functional units with required implementation
Agile approach to development and validation of AUTOSAR components

Traceability requirements <-> design

Are we done yet?  What is this block meant for?

• Bidirectional traceability between design and requirements helps to answer those questions (+ needs to be demonstrated...).

• Direct linkage between Simulink design blocks and requirements management tool:
Design reports

• Design reports help at understanding and reviewing design.

• Full-blown HTML reports very useful for development and quality teams:
  • Browsable design, block configuration parameters, traceability design to code, ...

• Sometimes too much info for customers:
  • IDNEO custom reports using Report Generator API + DOM API to programmatically create PDF documents.
Design validation phase
From simulation to design validation

- With Model-Based Design, design and design validation become very coupled processes.

- Design engineer performs exploratory simulations right from the time of model inception.

- Need for testing increases with design maturity:
  - Test suite with controlled (repeatable and maintainable) test cases.

- Simulink Test used to create and execute model test suites.
  - @IDNEO, replaced internal tool that required high development maintenance effort.
Simulink Test

Test Manager to create test suites and define test cases

Test harnesses automatically created for the model under test

Controlled setting of inputs and parameters for each test case

Model output assessment based on different criteria

Test Case Templates

Simulation Test
- Input
- Output
- Assessment Criteria

Baseline Test
- Input
- Output
- Assessment Criteria

Equivalence Test
- Input
- Output
- Assessment Criteria
Test results and coverage reports

- Native test results report used to keep track of test status and outcomes.

- Cumulative test coverage computed and reported for the complete test suite.

- Coverage helps to assess test suite completeness, as well as to identify “dead” Simulink blocks.

- Justification feature useful for audit purposes.
Code generation and verification
Code generation

- Once design is ready, Embedded Coder and AUTOSAR support package are used for production code generation (C language).
- Interfaces and architectural dependencies automatically in-place.
- Straightforward integration into AUTOSAR platform, it compiles right away.
Code verification

**Simulink Test**
- Direct reuse of design test suite by running in SIL mode.
- Equivalence test between MIL and SIL.

**Polyspace**
- Built-in configuration for AGC:
  - Reduction of “noise” level in the analysis.
- ...and, of course, Polyspace auto-generated reports.
Products and projects
Model-Based Design with MathWorks products used in several projects for different product types...

- AdBlue dosing systems.
- Brake by wire.
- Fuel door lock.
- Engine sound control.
- Water injection for combustion engines.
- Door access module.
- In-vehicle camera monitoring system.
Agile approach to development and validation of AUTOSAR components

... and different software architecture layers

- High-level application logic.
- Software drivers close to HW.

Technical Article available in MathWorks site:

*Developing and Testing AUTOSAR Software Components and Complex Device Drivers with Model-Based Design*

By Enric Valencia, Ph.D., and Joan Albesa, Ph.D., IDNEO
Example 1: Wake-up logic for control unit

Main characteristics:

• “Simple” and small SW component, but in a full AUTOSAR architecture.

Key takeaways:

• Model template auto-generated from ARXML
• Auto-generation avoids manual boilerplate code.
Example 2: Driver for DC motor

Main characteristics:

- Control of HW actuator (DC motor) with diagnostics.

Key takeaways:

- Simulation of driver nominal behavior.
- Simulation of HW abnormal behavior (diagnostics), difficult to mimic in real system setup.
Example 3: Engine sound generation

Main characteristics:

- Signal processing algorithms: digital filtering, linear interpolation, ...

Key takeaways:

- Parameter override for each test case.
- Store test output as WAV for audio engineer.
- Baseline recording in Simulink Test.
Conclusions
Agile practices in automotive SW development

Model-Based Design contributes to agile by means of common language (model) and faster development time (simulation).

Automation capabilities of MathWorks tools enable agile in practice without compromising quality standards.
Quality enhancement

Early error detection from design phase.

Consistent model validation.

Improved code quality.
Maintainability

Better readability (model vs code).

Enhanced traceability.
Efficiency gain

A single tool-chain/team, many different application scenarios
WHAT’s your DREAM?