Applying MathWorks Tools to Automotive Embedded Software Development

Neil Robson
Changan UK R&D Centre Ltd
2016-10-05
CONTENTS

1. Changan Introduction
2. Capability Building
3. Lead Project
4. Results
5. Next Steps
6. Conclusion
1. Changan Introduction

Top Four China Automotive Industry Groups

Vehicle sales volumes

- Million vehicles
- Year

Changan Powertrain Brand

Changan Ford

Changan Suzuki

Changan Mazda

Changan PSA
1. Changan Introduction

- Birmingham, UK: Power Train
- Chongqing: Study on comprehensive technology of automobile
- Haerbin: Vehicle development
- Yokohama, Japan: Interior and exterior trimming, automobile engineering
- Detroit, USA: Chassis technology and Autonomous vehicles
- Turin, Italy: Styling, bodywork
- Beijing: New energy automotive technology
- Shanghai: Vehicle integration
- Jiangxi: Power technology
**1. Changan Introduction**

**Changan UK R&D Centre Limited (CAUK)**

**Our mission:**
- To design and develop high quality, high performance powertrains to support Changan’s global requirements
- To develop a technology road map and expertise for Changan Powertrain
- To create an environment and opportunities for staff development

**Our vision:**
- To become a World Class leading Powertrain R&D centre, a promoter and practitioner of green powertrain technologies
2. Capability Building

- Why do projects with high software content repeatedly fail?

2. Capability Building

- Development of complete “Systems” capability for “Software Intensive High Integrity Systems”

CAUK key System and Software Engineering Implementation Strategy:

- Gain sponsorship from Changan senior management
- Set up a well-funded capability team focusing on System and Software Engineering improvement
- Phased introduction approach with carefully selected external support
- Document Process, methods and tools in a SEMP

Changan UK strategy focuses on People First then Infrastructure, Process and finally Tools
2. Capability Building

Focus of this presentation

- Process development
- Tool selection and integration
2. Capability Building

• Software Product Lines: developing a “software factory”*

• Planned and strategic reuse of software assets

• Management of variability to achieve different behaviours from a common software base

* Software Engineering Institute at Carnegie Mellon is a good resource http://www.sei.cmu.edu/productlines/
3. Lead Project

• Advanced Hybrid Powertrain for Plug-in Hybrid Electric Passenger Vehicle (PHEV)

Key Features:
• Downsized 3 cylinder turbocharged gasoline direct injection engine
• Dual Clutch 7 speed transmission
• P2 hybrid module
  • High torque output electric machine
  • Drive in Electric Vehicle (EV) mode with K0 clutch open
• Large traction battery allows in excess of 50 km EV range
• Regenerative Braking system to harvest the kinetic energy under braking demand
3. Lead Project

• Complex control system architecture

Initial CAUK software development focus
4. Results

• Enabling the “software factory”

• Flow system requirements efficiently into the software development process
• Design and establish the software architecture
• Enable configuration and variation management
• Minimise documentation overheads
• Develop an efficient code generation process
4. Results

- Flow system requirements efficiently into the software development process

Model Driven Approach

- Simulink & Stateflow
- Embedded Coder
- Simulink Projects
- Report Generator

Use of Mathworks Consultancy Services to leverage tool advantage

Software Architecture

Function Library

System Design Requirements

Technical Design Specifications

SharePoint

MATLAB

MATLAB Projects

Perforce

Variant control

F1
F2
F3
F3' F3''

F1
F2

.A2L, S19

Input

Output
4. Results

- Design and establish the software architecture

“Controls” deals with the physics of the specific actuators to achieve key states (No actuator knows about any other!)

“Demands” drives an individual resource through its key states (No resource knows about any other!)

“Shift Sequencers” contain the sequences for individual shift types

“Modes” control When and What type of shift is allowed

“Co-ordinators” choreograph the available resources
4. Results

- Enable configuration and variation management
- Version control is not enough!
- Process for multiple concurrent developers
- Model comparison and merge is your friend – use the tools!

Use of advanced branching

Model merge

www.perforce.com
4. Results

- Minimise documentation overheads
- Creation of Design documentation utilising Simulink Report Generator
  - Leverage Mathworks expertise
  - The model is the documentation

Model Documentation Automatic Generation
4. Results

- Develop an efficient code generation process – Simulink Embedded Coder
  - Configuration of data dictionary
4. Results

• Develop an efficient code generation process – Simulink Embedded Coder
• Target code generation optimisation^ 
• Automate software build – generation of binary files and calibration tool data files*

^ Software target optimisation > 15% improvement in execution speed 
* Software build environment optimisation and automation has resulted in > 10 times improvement in build times
The MBSE approach deployed on the project has already realised significant benefits and reduced technical risks.

- **Model Based Approach - Flow from System to Software:**
  - **Improved Collaboration**
    Substantial less translation work due to more diagrams
  - **Structured and clearer Requirements**
    Documentation at Vehicle and Powertrain level typically 50% larger in content
  - **Issues found Earlier**
    100s more technical issues found for the same phase of development
  - **Features and Re-use considered from the Outset**

“\textit{The MBSE approach deployed on the project has already realised significant benefits and reduced technical risks}”
5. Next Steps

- Optimisation of processes and leverage tool capability for V&V (Further development of Model Advisor checks, Simulink Design Verifier, Simulink Test)
  - Efficient ISO26262 compliance (Simulink V&V, certification kit)

- Improve traceability and integration between Systems and Software processes and tools

- Expansion of process/team/tools for global software collaboration (US, Europe, China)

- Cyber-security and autonomous vehicle applications

Other organisations experience may or may not be relevant to your situation.

Picking what works for you and discarding what doesn’t is a pragmatic approach.

Key learning points from our experience are:
- Consider capability pyramid as a balanced set of requirements
- Initial focus must be on basic processes and team formation
- Tool evaluation is time-consuming but will pay back multiple times
- Develop your requirements for process and tools
- Leverage tool supplier expertise – don’t sink the ship for a penny-worth of tar
- Road testing process and tools in a real-world application is the best way to fine tune

The End