MODEL-BASED SOFTWARE DEVELOPMENT: AN OEM’S PERSPECTIVE.
MODEL-BASED SOFTWARE DEVELOPMENT AT BMW.

- Software development with MATLAB & Simulink is used in wide range for the vehicle software developed at BMW.
- BACE (BMW AutoCoding Environment) is BMW’s central configuration of the MATLAB & Simulink tool chain.
  - It is currently in use by about 400 users.

<table>
<thead>
<tr>
<th>Door Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy Management, DAS functions, Seat-Belt-Reminder</td>
</tr>
<tr>
<td>Chassis Control functions</td>
</tr>
<tr>
<td>SW Components in Combustion and Electrical Engine Control, Battery Management</td>
</tr>
<tr>
<td>… and further SW in BMW motorcycles</td>
</tr>
</tbody>
</table>
BACE FEATURES.
CHALLENGES.
TIME-TO-MARKET DECREASES. EXAMPLES: E-MOBILITY AND CONNECTED MOBILITY SERVICES.
DIGITALIZATION - COMPLEXITY OF FUNCTIONS INCREASES.

EXAMPLE: HIGHLY AUTOMATED DRIVING.
**THESE CHALLENGES FOSTER USING AGILE DEVELOPMENT PROCESSES.**

An analogy to motorsports

<table>
<thead>
<tr>
<th>In motorsports</th>
<th>In an agile software project</th>
</tr>
</thead>
<tbody>
<tr>
<td>It is crucial to</td>
<td>short release and feedback cycles</td>
</tr>
<tr>
<td>break in time</td>
<td>allow us to detect problems early</td>
</tr>
<tr>
<td>and to steer</td>
<td>and to solve them</td>
</tr>
<tr>
<td>in order to</td>
<td>in order to</td>
</tr>
<tr>
<td>reach the finish line faster.</td>
<td>develop the right function in the required quality and budget faster.</td>
</tr>
</tbody>
</table>
AGILE CONSISTS OF P/M/T AS WELL AS OF VALUES, PRINCIPLES AND CULTURE.

Agile Software Development

Processes, Methods, Tools:

- Scrum (an agile process)
- Continuous Integration
- Build & Test Automation
- ...

Values, Principles, Culture:

- Agile Manifesto (4 core values)
- short release and feedback cycles
- Culture
  - Collaboration
  - Team responsibility
  - Self-organization
  - ...
CONTINUOUS INTEGRATION INSTEAD OF BIG-BANG INTEGRATION.

- Plan and **integrate** continuously.
- Feature **fast feedback**.
- Check-in, compile and **test frequently**.
- Invest in **build- and test automation**.

Key features:
Adaptable, transparent.
Continuous integration enables to innovate faster and deals better with complexity.
EXAMPLE ON (IMPORTANCE OF) CONTINUOUS INTEGRATION.
EXAMPLE FOR CI USAGE: DEVELOPMENT OF BACE FOR AUTOSAR CODE GENERATION, WITH MATHWORKS.

Starting point:
- BMW specific Toolbox BACE (BMW AutoCoding Environment), based on MATLAB, Simulink and Embedded Coder
- Used for non-AUTOSAR SW-development (modeling, code-generation, etc.)
- High degree of customization
- Successful use in a variety of series projects

Intermediate step:
- Support for AUTOSAR only

Task 1:
- Extension of BACE for AUTOSAR

Task 2:
- Merge of the toolboxes for AUTOSAR- and non-AUTOSAR-based development

[Seider, Validas AG]
BACE WITH AUTOSAR.
SOFTWARE DEVELOPMENT WORKFLOW.

Specification of architecture and SW component interfaces

Modelling

Code Generation

Data Declaration

Conversion

Import

Modeling

Static checks

Code gen.

Source code

Implemented, supported or adapted in BACE

Workflows for non-AUTOSAR-development

AUTOSAR Import with Update (Delta-Import)

[Seider, Validas AG]
PILOT PROJECTS PROVIDED FAST FEEDBACK ON RELEASES.

3 pilot projects
- From different BMW departments
- Milestone successfully completed
- 24 SW-Cs
- ~120,000 LOC generated from SW-C-models

Important:
- Representative selection of pilot projects (AUTOSAR has many features)

[Seider, Validas AG]
SUCCESS FACTORS IN THE DEVELOPMENT OF BACE FOR AUTOSAR AND RELATION TO AGILE PRINCIPLES.

| User Support                        | (Consulting, local support, documentation) |
| Pilot Projects                      | (Proof of usability and adequacy)          |
| **Collaboration**                    | (BACE users, BACE developers, tool vendor, management) |
| **Customization**                   | (Configuration, Add-Ons, „Glue Tools“)     |
| **Synchronization**                 | (Continuously; synchronization of requirements and release cycles) |
| **Requirements Analysis**           | (Initial; used SW development process; prioritization) |
| **Infrastructure**                  | (CMS, Issue-Tracking, Continuous Integration, automated tests, etc.) |

related to agile principles  related to Continuous Integration and agile
CONCLUSION AND SUMMARY.
AGILE AND MODEL-BASED DEVELOPMENT.

– Model-based development supports working on „the right level of abstraction“.

– It can
  – facilitate communication between domain experts and software specialists and
  – increase speed of development (generate instead of code).

– Both factors support agile development.
  – Communication and short development cycles are essential in agile development.

– Wish-list:
  A GERRIT based review process for Simulink models – as adoption of established practices from open-source development projects.
SUMMARY.

- A large part of the vehicle software at BMW is generated with embedded Coder from MATLAB & Simulink models.

- Support for Agile Development and Continuous Integration is central for the future evolution of this tool chain.
THANK YOU FOR YOUR ATTENTION.