Acceleration through Smart Combination of Methodologies and People
What are the expectations of next generations?
Customer in the center

Vehicle will be built around Functions

Source: ©WJP

• Vehicle functionality defined by software
• Decoupling between hardware and software
• Driving software as business model over vehicle lifetime
• Vehicle is becoming a software product – continuous next
Key challenges for Software Defined Vehicle - FUTURE

- > 200-300 million lines of code expected
- Level 5 AD driving will take up to 1 billion lines of code

Source: a slide from a 2020 presentation by Herbert Diess highlights the VW software ambition
Key challenges for Software Defined Vehicle Dev. Process

- Software Delivery Speed defined by Development Process
- Continuous Update and Deployment over Lifetime
- Cybersecurity, Risk and Variants Management until EOL
What can we learn from SW development?
Requires modern software development strategies
Continuous DevOps

DevOps is composed by two main elements

- Continuous Integration AND Continuous Deployment

DevOps is first of all a mentality shift

Speed | Rapid Delivery | Scalability
Staging for Software

Stage I
Software Testing

Stage II
Functional Testing & Calibration

Stage III
Shadow-Mode

Stage IV
OTA Ring Release’s

Software in the Loop

Testing Labs

In-Field Testing

Fleet Rollout

Labs In-Field Testing
What is the new narrative?

Testing software and HW development updates in sync with the development cycles for all vehicle variants

Reduce physical testing effort, increase flexibility and reduce costs

Conduct the right test in the correct test environment
Consistent, integrated Toolchain

Continuous Design, Verification and Validation
Application of the concept to tangible value creation

<table>
<thead>
<tr>
<th>Development Tasks</th>
<th>Simulation</th>
<th>Virtual Test Bed</th>
<th>Component Test Bed</th>
<th>Engine Test Bed</th>
<th>Powertrain Test Bed</th>
<th>Chassis Dyno</th>
<th>Road Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Today</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Future</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Virtual part of the prototype**
- **Physical part of the prototype**

**REDUCE**
- **Time**
  - By making sound decisions earlier
- **Costs**
  - By shifting tasks in cheaper environments
- **Quality**
  - By increasing system knowledge earlier

**INCREASE**
- Making early, fast and sound decisions
- Bridge the best from virtual & real world
- Reuse & Ease of use
- Bringing together versatility and flexibility
- Collaboration and knowledge sharing
The **AVL Value Creation Approach** to maximize the **Output** and enable **Left-Shift**

<table>
<thead>
<tr>
<th>Simulation</th>
<th>Virtual testing</th>
<th>Component testing</th>
<th>Engine testing</th>
<th>Powertrain testing</th>
<th>Chassis dyno testing</th>
<th>Road testing</th>
</tr>
</thead>
</table>

**1. Minimising the effort and maximising the output**

- Higher efficiency and productivity within the same development environment
- Increase of automation level
- Decrease set-up time

*Minor impact on development process*

**2. Shifting the workload**

- Shift of development tasks in testing environments with higher degree of virtualisation
- Way to move testing and validation tasks to an earlier point in time in the PDP

*Large impact on development process*

![Diagram showing workload with conservative approach and workload with increased productivity and advanced methods](image)

- Workload with conservative approach
- Workload with increased productivity and advanced methods / frontloading
## HV-Integration

**Productivity gains in a nutshell**

<table>
<thead>
<tr>
<th>Prototype vehicle only</th>
<th>With Integration and Powertrain Testbed usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Approximately 28 IN-VEHICLE TEST WEEKS (8/5)</em></td>
<td><em>Approximately 4 IN-VEHICLE TEST WEEKS (8/5)</em></td>
</tr>
<tr>
<td><em>Approximately 500 WEEKS REQUIRED</em></td>
<td><em>Approximately 12 INTEGRATION TB TEST WEEKS (16/5)</em></td>
</tr>
</tbody>
</table>

- **85% LESS PROTOTYPE VEHICLE USAGE**
- **40% LESS ENGINEERING EFFORT**
- **300 WEEKS REQUIRED**

### Possible Efficiency gain:
Emulation of missing components

### 50% Cost reduction
Reproducible & up to 24/7 automated Testing
Less Prototypes & Function driven development & Testing
Higher Test Coverage
Manpower shift and optimization → Frontloading

*Data from AVL PTE in comparison with road testing, for one prototype phase (LEAD Variant)*
Value Creation program

Identify ways to improve process and methodology, start pilot projects and knowledge transfer

**Value Creation work streams**
- Expert workshops
- Platform for expert discussions

**Based on existing infrastructure & processes**
- Start with analysis of existing workflows/tasks
- Estimate existing effort distribution over environments
- Prioritization of tasks with highest efficiency impact

**Improvement measures in specific roadmaps**
- Process and organizational improvements
- Mature and advanced methodology
- Advanced tools to simplify and facilitate testing and development

**Implementation and pilot projects**
- Hands-on training with engineers and staff
- Training concepts
- Knowledge transfer
With AVL’s staged software delivery process risk can be managed proactively.

Engineering know how to support our customer within new software centric development.

Combining automotive know-how with software development is key for speed.

Consistent and seamless toolchains for continuous verification & validation.
We owe it to the planet