Entwicklungsprozesse mit der UML für eingebettete Systeme
10 Jahre Modellierung bei Continental, eine ehrliche Bilanz
## Continental Corporation
### Five Strong Divisions

<table>
<thead>
<tr>
<th>Chassis &amp; Safety</th>
<th>Powertrain</th>
<th>Interior</th>
<th>Tires</th>
<th>ContiTech</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle Dynamics</td>
<td>Engine Systems</td>
<td>Instrumentation &amp; Driver HMI</td>
<td>PLT, Original Equipment</td>
<td>Air Spring Systems</td>
</tr>
<tr>
<td>Advanced Driver</td>
<td>Fuel &amp; Exhaust Management</td>
<td>Commercial Vehicles &amp; Aftermarket</td>
<td>Commercial Vehicle Tires</td>
<td>Elastomer Coatings</td>
</tr>
<tr>
<td>Assistance Systems (ADAS)</td>
<td></td>
<td></td>
<td></td>
<td>Industrial Fluid Systems</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mobile Fluid Systems</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Power Transmission Group</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Vibration Control</td>
</tr>
</tbody>
</table>

PLT – Passenger and Light Truck Tires
## Business Unit Commercial Vehicles & Aftermarket

Segments and their Portfolio

### Tachographs, Telematics & Services

- Digital tachograph solutions
- Analogue tachograph solutions
- Accessories & services for workshops
- Fleet management
- Telematic products

### Vehicle Electronics*

- Instrument clusters OE & platform solutions
- Single gauges
- Body controller for trucks
- Driver working place
- Body builder module
- Off & On-Highway transmission ECU
- Chassis master control unit
- Light control module
- Platform multiplex solutions

*Selective products from product categories

### Independent Aftermarket

- ATE wear and tear parts for brakes
- VDO replacement parts (fuel systems, actuators for central locking systems, engine actuators, HVAC blower & fan systems, screen & headlight washer systems, sensors, tire pressure monitoring systems)
- Multi-brand diagnostics
- OE diagnostics & services
- Diesel repair service
- Brake service equipment & tools

### Original Equipment Services

- All original parts of Automotive Group
- Portfolio for service & replacement to OEMs (commercial & passenger vehicles)
<table>
<thead>
<tr>
<th></th>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Motivation</td>
</tr>
<tr>
<td>2</td>
<td>Introduction Tachograph</td>
</tr>
<tr>
<td>3</td>
<td>Project Structure Tachograph</td>
</tr>
<tr>
<td>4</td>
<td>Legacy Development</td>
</tr>
<tr>
<td>5</td>
<td>Future Development</td>
</tr>
<tr>
<td>6</td>
<td>Support and Barriers</td>
</tr>
</tbody>
</table>
Motivation
Target Achievement

› Abstract and understand complex Systems

› Formal Development Approach
  › How do we come to a solution (break down)
  › Why have we chosen this solution (rationale)
  › Where are requirements implemented (traceability)

› Effective and reliable Verification and Validation

› Consistency between Architecture, Documentation, Code, Test, …
Agenda

1. Motivation
2. Introduction Tachograph
3. Project Structure Tachograph
4. Legacy Development
5. Future Development
6. Support and Barriers
The Project

- Tachograph for European market
  - Common Criteria Security Evaluation (highest Attack level)

Project Domain

- Team: Multi-site / Multi-team
- Approach: Migration based on existing software, by isolating security related features into additional microcontroller (called „SecCon“)
- Restricted HW:
  - CPU: [redacted]
  - Code/Data: [redacted]
  - RAM: [redacted]
Architectural Principles

› Definition of a Static Structure
  › Central Entry Point (Architectural Views)
  › Clear Hierarchy for SW Modules (Layers)
  › Clear Structure within the Modules (Interface, Implementation)

› Definition of a Workflow
  › Abstraction Methodology
  › Definition of Abstraction Level 0 – 3
    › Level 0: Cross System Level (Collaboration)
    › Level 1: System Level (Device)
    › Level 2: Sub-system Level (Controller – SW Architecture)
    › Level 3: Component Level (SW Module – SW Design)
System Overview (level 1)
Process – V-Model
Feature Driven Development (FDD)
## Agenda

1. **Motivation**
2. **Introduction Tachograph**
3. **Project Structure Tachograph**
4. **Legacy Development**
5. **Future Development**
6. **Support and Barriers**
Structure – Overview (level 1 & level 2)

- The Architecture consists of several Layers
- Additional Packages provides Architectural Rules
- Use Case Folders provide high level Abstraction
- Different Views on the Model provides a good Overview
Structure – Overview (level 1 & level 2)

- The Use Case folders are container for corresponding Feature Sets
- The Feature Sets are satisfying Requirements
- Sequence Diagrams are showing Scenarios of the Feature Sets on different levels and impact
- The Overview consists of all identified Feature Set represented as UML Use Cases
Structure – Architecture & Design

SW Architecture
(Level 2 Diagram)

1. Every minute
   enRequestService(eService)
   enRequestService(eService)
   enRequestService(eService)

2. The SecCon checks the plausibility of the data.
   getCurrentTime()
   checkPlausibility()

SW Design
(Level 3 Diagram)

1. Every minute
   enRequestService(eService)
   enRequestService(eService)
   enRequestService(eService)

2. The SecCon checks the plausibility of the data.
   time = getCurrentTime()
   checkPlausibility(odometerValue, time)
Structure – Views (level 2)

› The Service View shows an Overview of all existing Services

› It defines Types which represents the Message structure and the Service IDs

› The Module View shows all existing Modules and how they collaborate
Each Module is stored in its own CI

The Test Package is stored in a separate CI

Requirements are satisfied on Module level

Each Module can have more detailed Level 3 Sequence Diagrams
Structure – Implementation / Test (level 3)

› The Test Package contains Unit and Module Tests

› The Module Tests are again stored in a separate CI

› The Component consists of several Configurations

› The Tests are created according the defined Feature Sets
Agenda

1 Motivation
2 Introduction Tachograph
3 Project Structure Tachograph
4 Legacy Development
5 Future Development
6 Support and Barriers
The Project: New Module Development

- NCMDT is a new Module
  - Uses object oriented C
  - Provides Singleton Objects for accessing the Services
  - Provides callback Interfaces to be implemented by the clients
  - Documents its behavior with several SDs
  - Uses Code Generation for Statecharts

The Project DTCO3283R1

7-Oct-16

Alexander Schneider © Continental Corporation
The Release View provides information for several releases.

Within a Release is Information about all the changes:

- Each Feature has its own Package
- FeatureSet Modules show the related Module changes

They inherit from the real Modules and show the needed changes and responsibilities.

The Features have also SDs and Requirement traceability.
## Agenda

<table>
<thead>
<tr>
<th></th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Introduction Tachograph</td>
</tr>
<tr>
<td>3</td>
<td>Project Structure Tachograph</td>
</tr>
<tr>
<td>4</td>
<td>Legacy Development</td>
</tr>
<tr>
<td>5</td>
<td>Future Development</td>
</tr>
<tr>
<td>6</td>
<td>Support and Barriers</td>
</tr>
</tbody>
</table>
System: Cash Machine

Key Pad

Touch Panel

Card Reader
System: Cash Machine
Use Cases Analysis (level 0)
System: Cash Machine
Scenario: Use Case Execution I (level 0)

sd [Package] GetMoneyBBScenariosPkg [AuthenticationFailed]

:Customer

:Uc_GetMoney

evInsertCard()

  cardInserted()

  cardId=readCardId()

  askForPin()

  pinEntered()

  cardPin=readPinNumber()

  verifyPin(Pin:cardPin)

  authenticatePin(Pin:cardPin)

  notifyCustomer()

evEnterPin()

  pinEntered()

  cardPin=readPinNumber()

  verifyPin(Pin:cardPin)

  authenticatePin(Pin:cardPin)

  notifyCustomer()
System: Cash Machine
Use Cases Partitioning (level 0)
System: Cash Machine
Scenario: Use Case Execution II (Happy Day – level 0)

› The Lead Architect defines the Specification

› All parties are involved

› Result of an Iterative Analysis and Design Process
System: Cash Machine
Static software architecture (level 2)
System: Cash Machine (Not shown in talk)
Software construction (level 3)

AuthenticatorOverview

Authenticator
authenticatePin("
comparePins(st...

HashCalculator
1
PinRequester
1

Requirements::id11
«Requirement»
ID = 11
The Authentication must
not take longer than 30
ms.
«satisfy»

System: Cash
Machine
(Not shown in talk)
Software construction (level 3)

Detailed Design

ENG.5.BP6

Software Integration Test
Software Arch. & Design
Software Detailed Design
Software Module Test
Implementation / Unit Test

Software construction (level 3)

7-Oct-16
Alexander Schneider © Continental Corporation

Continental
System: Cash Machine
Dynamic software architecture (level 2)

1. Insert the card
   - insertCard()
   - cardInserted()
   - readCardId()
   - cardId()
   - askForPin()

2. Enter the pin
   - enterPin()
   - readPinNumber()
   - pinNumber()
   - verifyPin()
   - authenticatePin(cardId, pinNumber)
   - authenticated()

3. Request the money
   - askForAmount()
   - (4 digit)

If a card is inserted, it is read to identify the customer.
To authenticate the customer, a pin is requested.
If a pin is entered, it is read to authenticate the customer.
The stored and hashed pin on the card is authenticated against the entered pin.
If the customer is authenticated, it is asked for the amount of money.

1. Insert the card
1. Insert the card
3. Request the money
   --> to be continued ...
System: Cash Machine (Not shown in talk)
Software construction (level 3)
The stored and hashed pin on the card is authenticated against the entered pin.

```plaintext
authenticatePin(cardId, pinNumber)
g gotten(requestPin(cardId))
storedPinHash()
calculateHash(pinNumber)
enteredPinHash()
comparePins(storedPinHash, enteredPinHash)
authenticated() Requirements::id7
The stored and hashed pin on the card is authenticated against the entered pin.
```
## Agenda

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Motivation</td>
</tr>
<tr>
<td>2</td>
<td>Introduction Tachograph</td>
</tr>
<tr>
<td>3</td>
<td>Project Structure Tachograph</td>
</tr>
<tr>
<td>4</td>
<td>Legacy Development</td>
</tr>
<tr>
<td>5</td>
<td>Future Development</td>
</tr>
<tr>
<td>6</td>
<td>Support and Barriers</td>
</tr>
</tbody>
</table>
Summary

› Full Bilateral Traceability
› Full Consistency
› Improves Communication
› Validity of Documentation
› Reuse of Documentation

› Lots of Powerful Diagrams
› Performed in real Projects with Assessments
Summary

› Guided and Formal development process
  › Reducing Errors
  › Simpler Tracing
  › Growing Design

› Single Source
  › Common data storage
  › Less redundancies
  › Reliable, Consistent and Traceable
  › More re-use

› Automation
  › Executable Models
Support and Barriers

› Support for lots of Standards
  › Process Models (CMMI, (Automotive)SPICE, …)
  › Quality Characteristics (ISO9126, ISO25010, …)
  › Safety Standards (ISO26262, ISO61508, …)

› Barriers
  › Management needs to understand the need of this technology
  › Management needs to support it (Time, Budget, Commitment)
  › Employees need to be qualified
  › Employees need to get the chance to collect experience
  › Projects need to have support from Experts (Coaching)
Thank you for your attention!