



## **KoKoVa: Kontinuierliche und Kompositionelle Validierung für Autonomes Fahren**

Simon Dierl, Malte Mauritz, Tim Tegeler, Dominic Wirkner, Falk Howar und Bernhard Steffen

## Automotive Software I: Software als Firmware

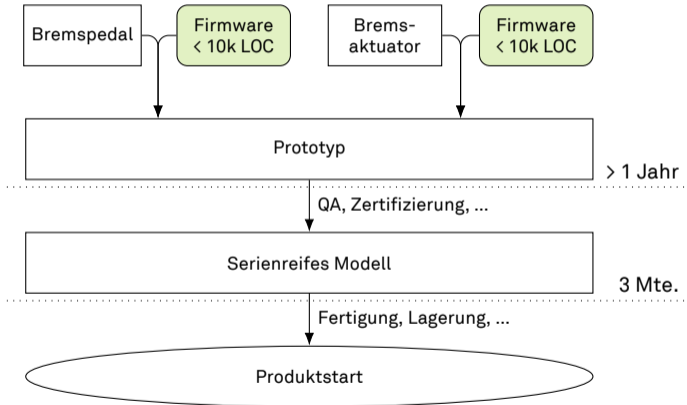
### Philosophie

- unveränderliche Firmware
- Gerätespezifisch
- Auslieferung mit Bauteil
- Keine Softwareupdates

## Automotive Software I: Software als Firmware

### Philosophie

- unveränderliche Firmware
- Gerätespezifisch
- Auslieferung mit Bauteil
- Keine Softwareupdates



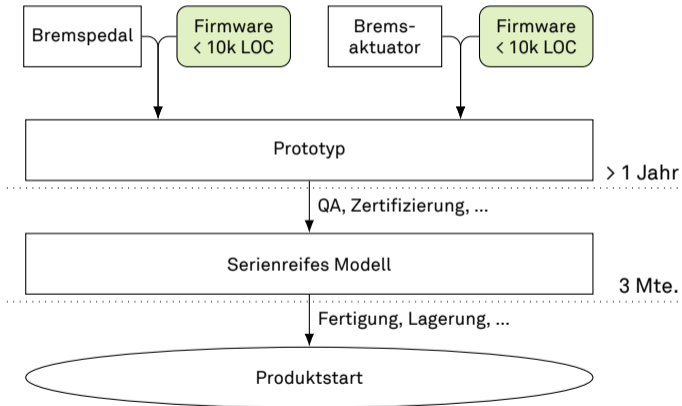
## Automotive Software I: Software als Firmware

### Philosophie

- unveränderliche Firmware
- Gerätespezifisch
- Auslieferung mit Bauteil
- Keine Softwareupdates

### Fazit

- Gut für simple Funktionen
- Zertifizierung
- Testprogramme



## Automotive Software II: Last-Minute-Entwicklung

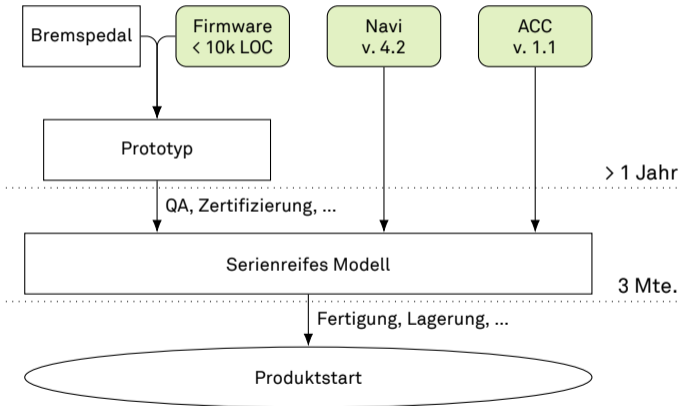
### Philosophie

- Zugänglich für Nutzer
- High-Level-Funktionen
- Auslieferung mit Auto
- Updates in Werkstatt

## Automotive Software II: Last-Minute-Entwicklung

### Philosophie

- Zugänglich für Nutzer
- High-Level-Funktionen
- Auslieferung mit Auto
- Updates in Werkstatt



Auto

Phil

- 
- 
- 
- 



<https://picota.de/en/98666276>

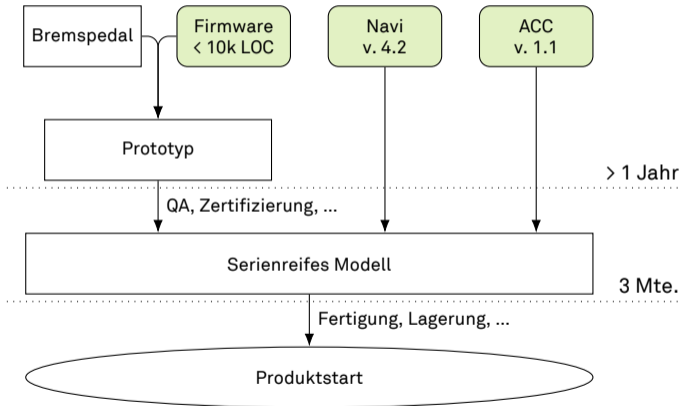
## Automotive Software II: Last-Minute-Entwicklung

### Philosophie

- Zugänglich für Nutzer
- High-Level-Funktionen
- Auslieferung mit Auto
- Updates in Werkstatt

### Fazit

- Software liegt auf kritischem Pfad
- Software-QA-Techniken





## Automotive Software III: Agile Entwicklung

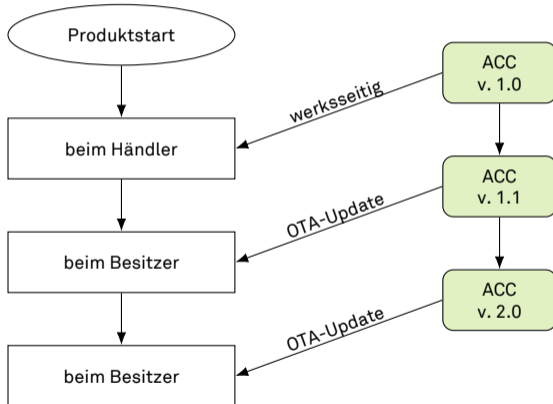
### Philosophie

- Auto als „PC“-Plattform
- Entkoppelte Entwicklung
- Produktlinien in SW
- OTA-Updates

## Automotive Software III: Agile Entwicklung

### Philosophie

- Auto als „PC“-Plattform
- Entkoppelte Entwicklung
- Produktlinien in SW
- OTA-Updates



Auto

Phil

- 
- 
- 
- 

**We actually hit it!**

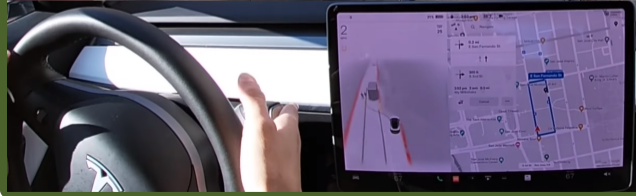


Source: AI Addict on YouTube

Auto

Phil

- 
- 
- 
- 



Source: AI Addict on YouTube

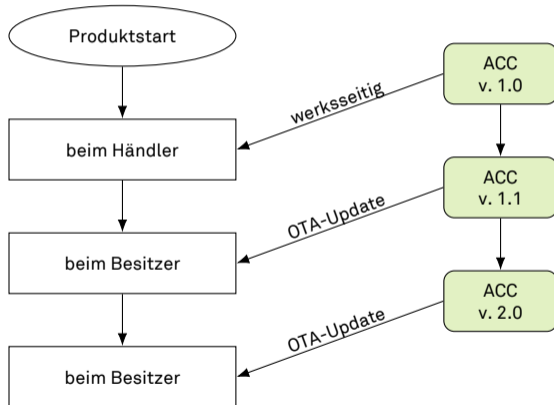
## Automotive Software III: Agile Entwicklung

### Philosophie

- Auto als „PC“-Plattform
- Entkoppelte Entwicklung
- Produktlinien in SW
- OTA-Updates

### Fazit

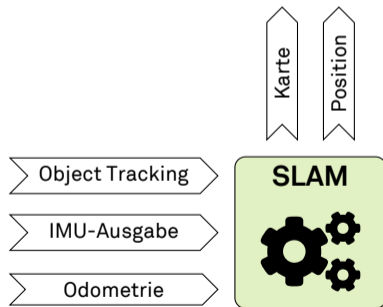
- Software und Hardware entkoppelt
- QA für Software unzureichend



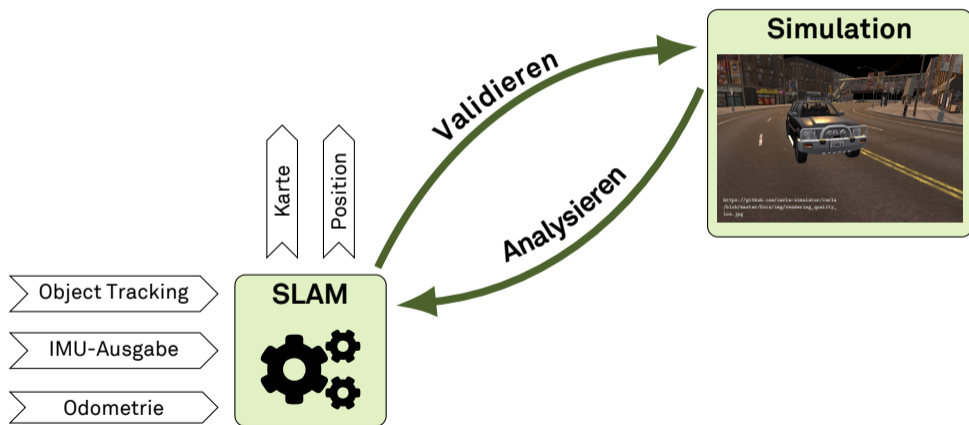
## Automotive Software: Zwischenfazit

**Klassische Techniken: skalieren nicht**  
**Agile Methoden: QA unzureichend**

## Unsere Ideen: Komponieren, Simulieren & Modellieren

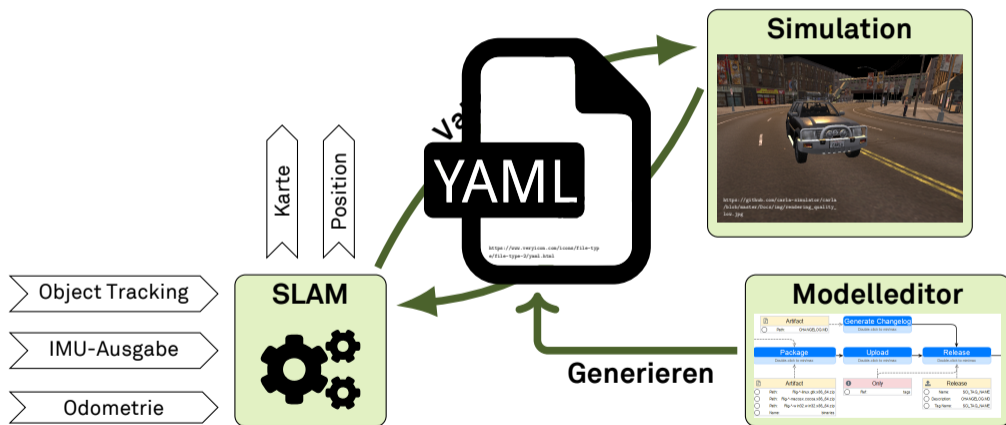


## Unsere Ideen: Komponieren, Simulieren & Modellieren





## Unsere Ideen: Komponieren, Simulieren & Modellieren



## Komponentenbasierte Entwicklung mit ROS

### ROS

- Robot Operating System
- Kein Betriebssystem!

## Komponentenbasierte Entwicklung mit ROS

### ROS

- Robot Operating System
- Kein Betriebssystem!
- Nachrichtenaustausch-Middleware
- Sprachunabhängig
- Eigenes Build- & Ökosystem

## Komponentenbasierte Entwicklung mit ROS

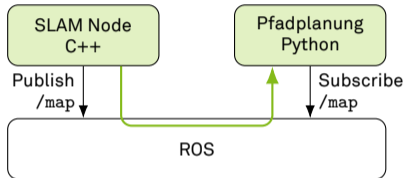
### ROS

- Robot Operating System
- Kein Betriebssystem!
- Nachrichtenaustausch-Middleware
- Sprachunabhängig
- Eigenes Build- & Ökosystem
- Neu: ROS 2

## Komponentenbasierte Entwicklung mit ROS

### ROS

- Robot Operating System
- Kein Betriebssystem!
- Nachrichtenaustausch-Middleware
- Sprachunabhängig
- Eigenes Build- & Ökosystem
- Neu: ROS 2

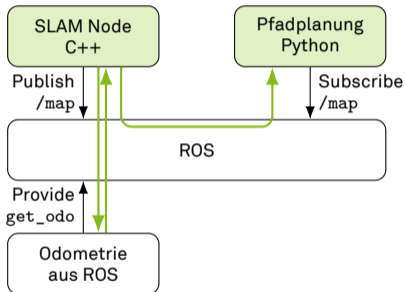


Morgan Quigley u. a. „ROS: an open-source robot operating system“. In: *ICRA workshop on open source software*. Bd. 3. 3.2. 2009, S. 5. URL: <http://www.robotics.stanford.edu/~ang/papers/icraoss09-ROS.pdf>

## Komponentenbasierte Entwicklung mit ROS

### ROS

- Robot Operating System
- Kein Betriebssystem!
- Nachrichtenaustausch-Middleware
- Sprachunabhängig
- Eigenes Build- & Ökosystem
- Neu: ROS 2

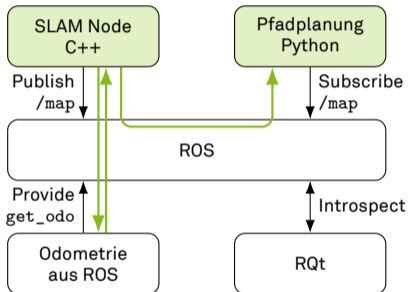


Morgan Quigley u. a. „ROS: an open-source robot operating system“. In: *ICRA workshop on open source software*. Bd. 3. 3.2. 2009, S. 5. URL: <http://www.robotics.stanford.edu/~ang/papers/icraoss09-ROS.pdf>

## Komponentenbasierte Entwicklung mit ROS

### ROS

- Robot Operating System
- Kein Betriebssystem!
- Nachrichtenaustausch-Middleware
- Sprachunabhängig
- Eigenes Build- & Ökosystem
- Neu: ROS 2



Morgan Quigley u. a. „ROS: an open-source robot operating system“. In: *ICRA workshop on open source software*. Bd. 3. 3.2. 2009, S. 5. URL: <http://www.robotics.stanford.edu/~ang/papers/icraoss09-ROS.pdf>

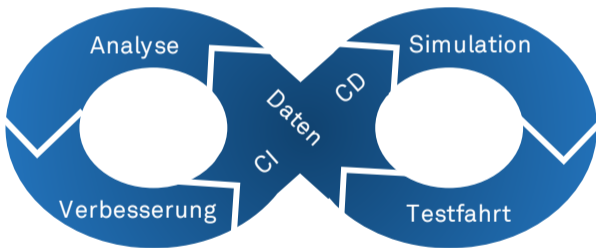
## Demo: turtlesim und rqt



## DevOps & YAML vs. Modellsprachen

### DevOps

- Entwicklung und Betrieb vereinen
- CI/CD ist Bindeglied



## DevOps & YAML vs. Modellsprachen

### DevOps

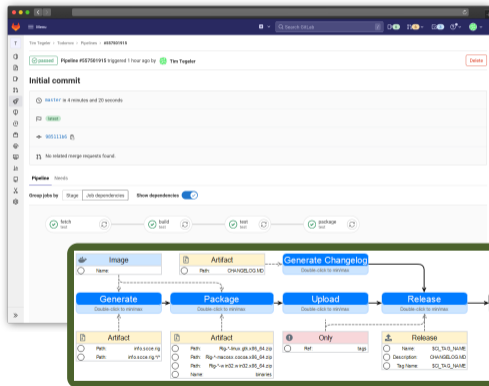
- Entwicklung und Betrieb vereinen
- CI/CD ist Bindeglied
- Übliches Format: YAML
- Schnell unübersichtlich!

```
1 image: node:16.7
2 fetch:
3   script:
4     - npm ci
5   artifacts:
6     paths:
7       - node_modules/
8       expire_in: 1 hour
9 build:
10  script:
11    - npm run build
12  artifacts:
13    paths:
14      - js
15  needs:
16    - fetch
17 test:
18  script:
19    - npm test
20  needs:
21    - fetch
22    - build
23 package:
24  script:
25    - npm run bundle
26  artifacts:
27    paths:
28      - out
29  needs:
30    - fetch
31    - build
32    - test
```

## DevOps & YAML vs. Modellsprachen

### DevOps

- Entwicklung und Betrieb vereinen
- CI/CD ist Bindeglied
- Übliches Format: YAML
- Schnell unübersichtlich!
- Besser: „Hochsprache“ für CI/CD
- Visualisierungen statt Text



## Modellgetriebene Entwicklung von CI/CD-Workflows mit Rig

### Rig

- IDE für CI/CD
- Grafische Modell-DSL
- Ausgabeformat: `.gitlab-ci.yml`

## Modellgetriebene Entwicklung von CI/CD-Workflows mit Rig

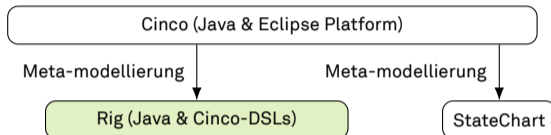
### Rig

- IDE für CI/CD
- Grafische Modell-DSL
- Ausgabeformat: `.gitlab-ci.yml`
- Meta-modelliert in Cinco
- Aktuell: Low-Level-Primitiven
- Kein Support für Simulationen

## Modellgetriebene Entwicklung von CI/CD-Workflows mit Rig

### Rig

- IDE für CI/CD
- Grafische Modell-DSL
- Ausgabeformat: `.gitlab-ci.yml`
- Meta-modelliert in Cinco
- Aktuell: Low-Level-Primitiven
- Kein Support für Simulationen

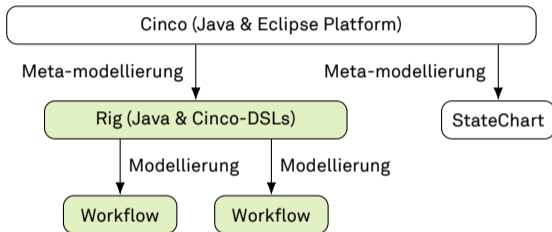


Tim Tegeler, Frederik Gossen und Bernhard Steffen. „A model-driven approach to continuous practices for modern cloud-based web applications“. In: *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. Jan. 2019, S. 1–6. DOI: [10.1109/CONFLUENCE.2019.8776962](https://doi.org/10.1109/CONFLUENCE.2019.8776962)

## Modellgetriebene Entwicklung von CI/CD-Workflows mit Rig

### Rig

- IDE für CI/CD
- Grafische Modell-DSL
- Ausgabeformat: `.gitlab-ci.yml`
- Meta-modelliert in Cinco
- Aktuell: Low-Level-Primitiven
- Kein Support für Simulationen

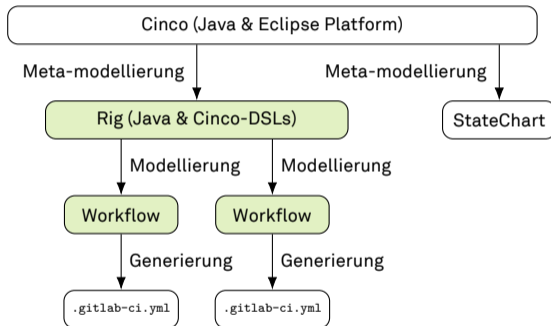


Tim Tegeler, Frederik Gossen und Bernhard Steffen. „A model-driven approach to continuous practices for modern cloud-based web applications“. In: *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. Jan. 2019, S. 1–6. DOI: [10.1109/CONFLUENCE.2019.8776962](https://doi.org/10.1109/CONFLUENCE.2019.8776962)

## Modellgetriebene Entwicklung von CI/CD-Workflows mit Rig

### Rig

- IDE für CI/CD
- Grafische Modell-DSL
- Ausgabeformat: `.gitlab-ci.yml`
- Meta-modelliert in Cinco
- Aktuell: Low-Level-Primitiven
- Kein Support für Simulationen



Tim Tegeler, Frederik Gossen und Bernhard Steffen. „A model-driven approach to continuous practices for modern cloud-based web applications“. In: *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. Jan. 2019, S. 1–6. DOI: 10.1109/CONFLUENCE.2019.8776962



# Demo: Rig

## Use Case: Formula Student & Jarvic

### Formula Student

- Studentischer Konstruktions- und Fahrwettbewerb für Rennautos
- Seit 2010: auch elektisch
- Seit 2017: auch **autonom**

## Use Case: Formula Student & Jarvic

### Formula Student

- Studentischer Konstruktions- und Fahrwettbewerb für Rennautos
- Seit 2010: auch elektisch
- Seit 2017: auch **autonom**
- Einzelrennen auf unbekanntem Kurs
- Hütchen als Bahnbegrenzung
- **Metrik:** Rundenzeit, Abzüge für Kollisionen

## Use Case: Formula Student & Jarvic

### Formula Student

- Studentischer Konstruktions- und Fahrwettbewerb für Rennautos
- Seit 2010: auch elektisch
- Seit 2017: auch **autonom**
- Einzelrennen auf unbekanntem Kurs
- Hütchen als Bahnbegrenzung
- **Metrik:** Rundenzeit, Abzüge für Kollisionen

### Jarvic

- In Dortmund: GET racing Dortmund e. V.
- Fahrfunktion: **Jarvic**
- ROS 1-basiert

Bereits Unterstützung für:

- Simulationen
- Modellautos

Use

Form

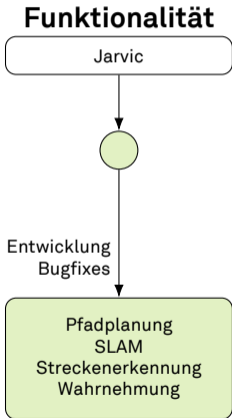
- 
- 
- 
- 
- 
- 



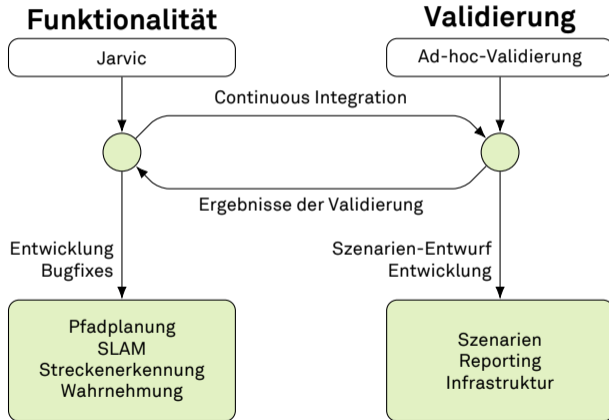
[https://commons.wikimedia.org/wiki/File:2018\\_Alexander\\_Lee\\_P10308.jpg](https://commons.wikimedia.org/wiki/File:2018_Alexander_Lee_P10308.jpg)  
2.0

## Demo: AADC-Auto

## Mögliches Vorgehen

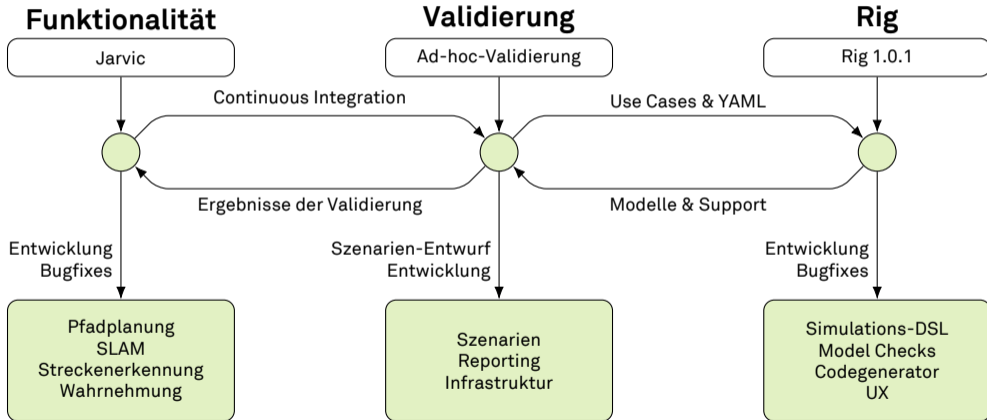


## Mögliches Vorgehen





## Mögliches Vorgehen



## Wieso PG KoKoVa?

### Was ihr mitnehmt:

- Hands-on-Projekt mit Relevanz
- Erfahrung mit Automotive und Robotik
- Erfahrung mit DSLs und Modellierung

## Wieso PG KoKoVa?

### Was ihr mitnehmt:

- Hands-on-Projekt mit Relevanz
- Erfahrung mit Automotive und Robotik
- Erfahrung mit DSLs und Modellierung

### Was wir bieten:

- Laborraum OH12 / 2.007
- 2 Modellautos
- Bei Bedarf Serverressourcen
- **Kaffeevollautomat**

## Wieso PG KoKoVa?

### Was ihr mitnehmt:

- Hands-on-Projekt mit Relevanz
- Erfahrung mit Automotive und Robotik
- Erfahrung mit DSLs und Modellierung

### Was wir bieten:

- Laborraum OH12 / 2.007
- 2 Modellautos
- Bei Bedarf Serverressourcen
- **Kaffeevollautomat**

### Was wir erwarten:

#### Muss-Kriterien:

- Grundkenntnisse Linux
- Programmieren

#### Außerdem:

- Hohe Lernbereitschaft

## Wieso PG KoKoVa?

### Was ihr mitnehmt:

- Hands-on-Projekt mit Relevanz
- Erfahrung mit Automotive und Robotik
- Erfahrung mit DSLs und Modellierung

### Was wir bieten:

- Laborraum OH12 / 2.007
- 2 Modellautos
- Bei Bedarf Serverressourcen
- **Kaffeevollautomat**

### Was wir erwarten:

#### Muss-Kriterien:

- Grundkenntnisse Linux
- Programmieren

#### Außerdem:

- Hohe Lernbereitschaft

#### Gerne zusätzlich:

- Kenntnisse zu den Technologien
- Erfahrung mit C++, Python, Java
- Vorkenntnisse zu CPS

## Ziele der Projektgruppe

### Minimalziel

- 1 Implementierung von wiederverwendbaren Komponenten:
  - Wahrnehmung
  - SLAM
  - Racetrack-Steckenerkennung
  - optimierte Pfadplanung
- 2 Lauffähig in Simulation und auf echter Hardware
- 3 Validierung durch geeignete Simulationen via CI/CD
- 4 Rig-Unterstützung für die validierenden Workflows
- 5 Seminar, Zwischen- und Endbericht

## Ziele der Projektgruppe

### Minimalziel

- 1 Implementierung von wiederverwendbaren Komponenten:
  - Wahrnehmung
  - SLAM
  - Racetrack-Steckenerkennung
  - optimierte Pfadplanung
- 2 Lauffähig in Simulation und auf echter Hardware
- 3 Validierung durch geeignete Simulationen via CI/CD
- 4 Rig-Unterstützung für die validierenden Workflows
- 5 Seminar, Zwischen- und Endbericht

### Erweiterungsideen

- Vollständig lauffähige Fahrfunktion
- Szenarienbibliothek vergrößern
- Kooperation mit z. B. RosBag Bazaar-Konsortium
- Weitere Aufgabenstellungen
- Teilnahme am P<sup>5</sup>-Preis

## Lizenzierung

### Jarvic

- Internes Projekt des GET racing Dortmund e. V.
- **Nicht** Open Source
- Wettbewerbsvorteil bei der Formula Student



## Lizenzierung

### Jarvic

- Internes Projekt des GET racing Dortmund e. V.
- **Nicht** Open Source
- Wettbewerbsvorteil bei der Formula Student

### Rig

- Open-Source-Projekt des Lehrstuhls 5
- Eclipse Public License 2.0
- Öffentliche Codebasis
- Merge Request möglich

## Lizenzierung

### Jarvic

- Internes Projekt des GET racing Dortmund e. V.
- **Nicht** Open Source
- Wettbewerbsvorteil bei der Formula Student

### Rig

- Open-Source-Projekt des Lehrstuhls 5
- Eclipse Public License 2.0
- Öffentliche Codebasis
- Merge Request möglich

### Eigenentwicklungen

- Kein Veröffentlichungszwang
- Freie Lizenzwahl

## Lizenzierung

### Jarvic

- Internes Projekt des GET racing Dortmund e. V.
- **Nicht** Open Source
- Wettbewerbsvorteil bei der Formula Student

### Rig

- Open-Source-Projekt des Lehrstuhls 5
- Eclipse Public License 2.0
- Öffentliche Codebasis
- Merge Request möglich

### Eigenentwicklungen

- Kein Veröffentlichungszwang
- Freie Lizenzwahl

### Immer

- Nutzungsrecht durch GET racing Dortmund e. V. für autonomes Fahren
- Nutzungsrecht für Fakultät in Lehre und Forschung

**Bei Fragen: fragen!**