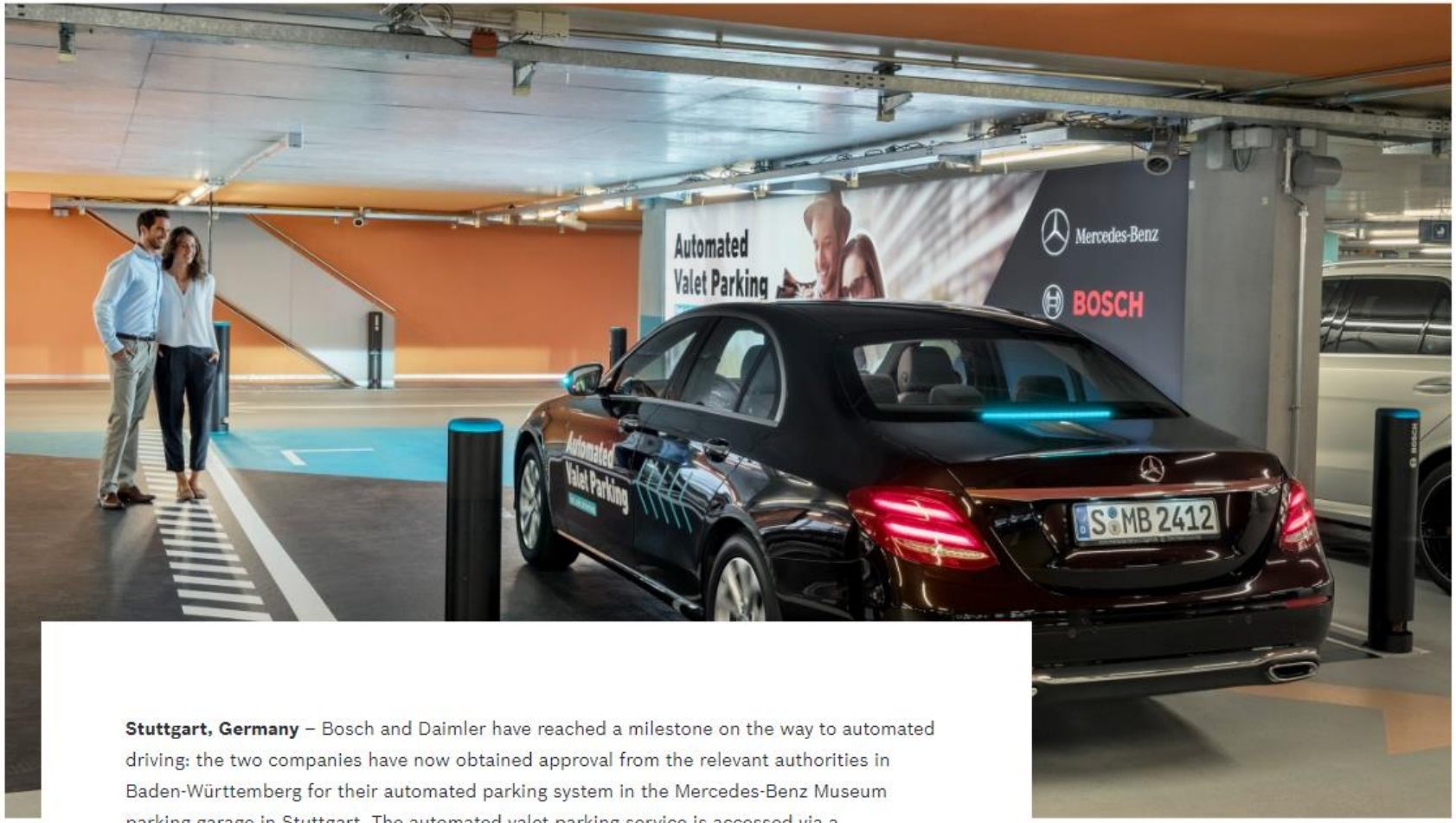
openGADES:

a modern and leightweight *DOCS AS CODE* approach ~~for Documentation~~ for Product Engineering

Andreas Karsten (XC-AD/EYM2) Thomas Woyke (AE- SW Campus)

openGADES
It‘s roots



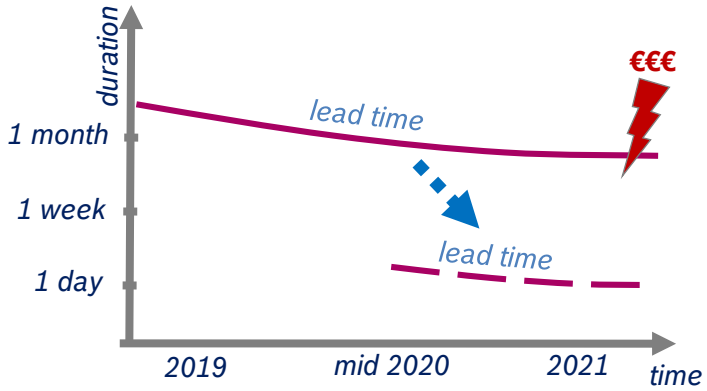
Stuttgart, Germany – Bosch and Daimler have reached a milestone on the way to automated driving: the two companies have now obtained approval from the relevant authorities in Baden-Württemberg for their automated parking system in the Mercedes-Benz Museum parking garage in Stuttgart. The automated valet parking service is accessed via a smartphone app and requires no safety driver. This makes it the world's first fully automated driverless SAE Level 4[1] parking function to be officially approved for everyday use.

https://www.youtube.com/watch?v=8IRA6FPoaJU

openGADES
Vision: Everything as Code

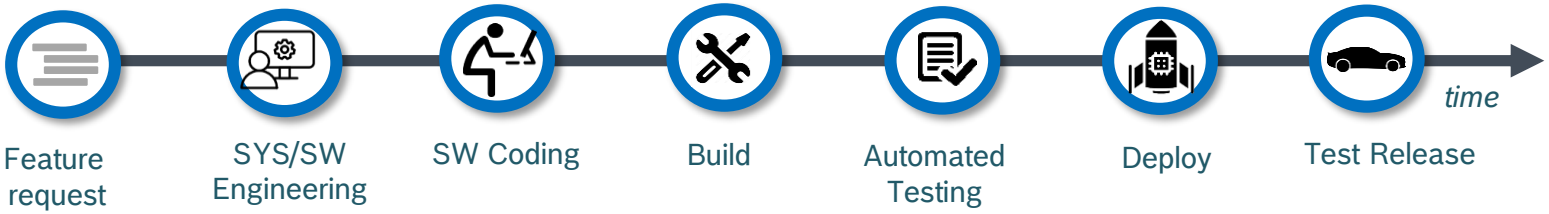
Current situation @automotive

- Low feature development velocity
- High scalability cost to cover high number variants



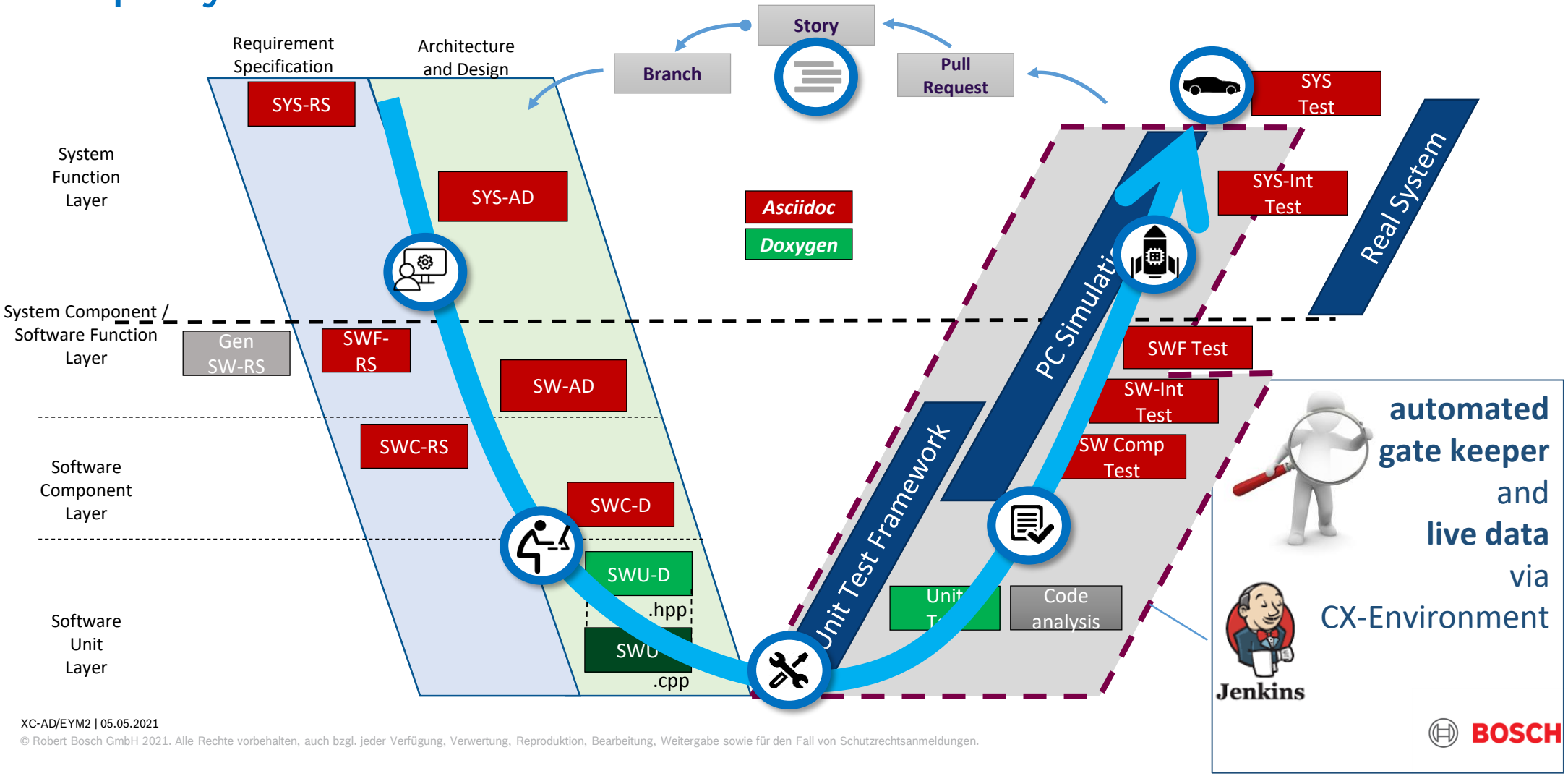
What do we need to change?

- Increase the rate of automation
- Enable a seamless transition from „spike“ to „stabilize“ maturity level
- Get faster by orders of magnitude (**establish IT-Productivity**)
  - Local optimizations will not lead to required improvements!



XC-AD/EYM2 | 05.05.2021

openGADES
Exemplary Docs-as-Code Ideal

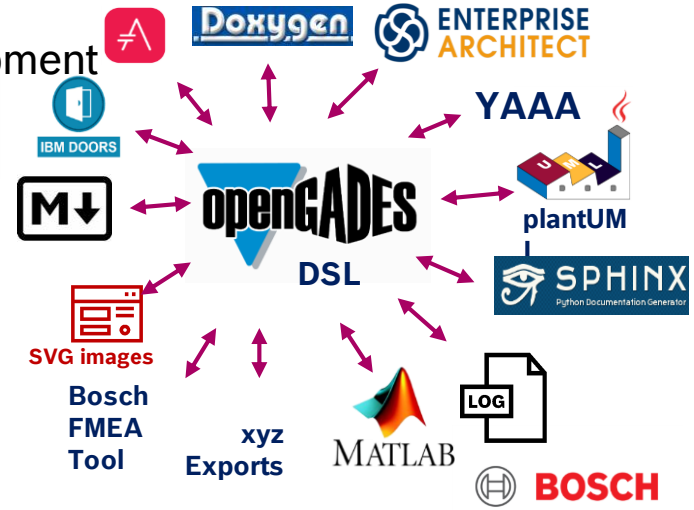


openGADES
What does Open GADES stand for?

open:	Built on common, open source components Driven by the “open source community” mindset
Generic:	Cross-platform; not specialized to a single, specific use case (e.g. requirements or architecture)
Ascii-based:	make use of the nice sw-development tools: editors, versioning, diff/merge, reviews
Documentation:	primary goal: creation and maintenance of technical documentation in a simple and lean manner
and	Traceability and a formal model come „on top“
Engineering:	Usage from the very start of the product life cycle, along the complete „V-model“ (comp. ASPICE) from concepts and design decisions to the release
System:	Modular framework of use-case specific, state-of-the-art modules (not a monolithic tool!)

openGADES
What is OpenGADES

- openGADES (**G**eneric **A**scii–based **D**ocumentation and **E**ngineering **S**ystem)
- openGADES is a customizable, lightweight toolchain for software development, supporting process requirements e.g. from ASPICE and ISO26262.
  - Add features required in the normative context, mainly formal attribution, traceability and analysis features
- Guiding Principles
  - Use standard Tools known from the modern IT development
  - Docs as code approach: same tools, same workflow as for software development
- KISS: Keep it simple and smart
  - Reuse instead of re-invent
  - Ecosystem: loosely coupled modules instead of a monolith



openGADES
Annotation Language

Item Definition

Unique Names

```
--
@ID{SWR_MyRequirement, asil=B,
someAttr=someValue}

The system shall …
--
```


XC-AD/EYM2 | 05.05.2021

openGADES
Annotation Language

Item Definition

Artifact Type Definition

```
--
@ID{SWR_MyRequirement, asil=B,
someAttr=someValue}

The system shall …
--
```

openGADES
Annotation Language

Item Definition

Key Value Attributes

```
--
@ID{SWR_MyRequirement, asil=B,
someAttr=someValue}

The system shall …
--
```


XC-AD/EYM2 | 05.05.2021

openGADES
Annotation Language

Item Definition

```
--
@ID{SWR_MyRequirement, asil=B,
someAttr=someValue}

The system shall …
--
```

Item Reference

```
--
@ID{SWD_MyDesignItem, asil=B,
from=SWR_MyRequirement}

My Design Item is responsible …
--
```

Upstream Reference

openGADES
Annotation Language

Item Definition

```
--
@ID{SWR_MyRequirement, asil=B,
someAttr=someValue}

The system shall …
--
```

Cross Reference

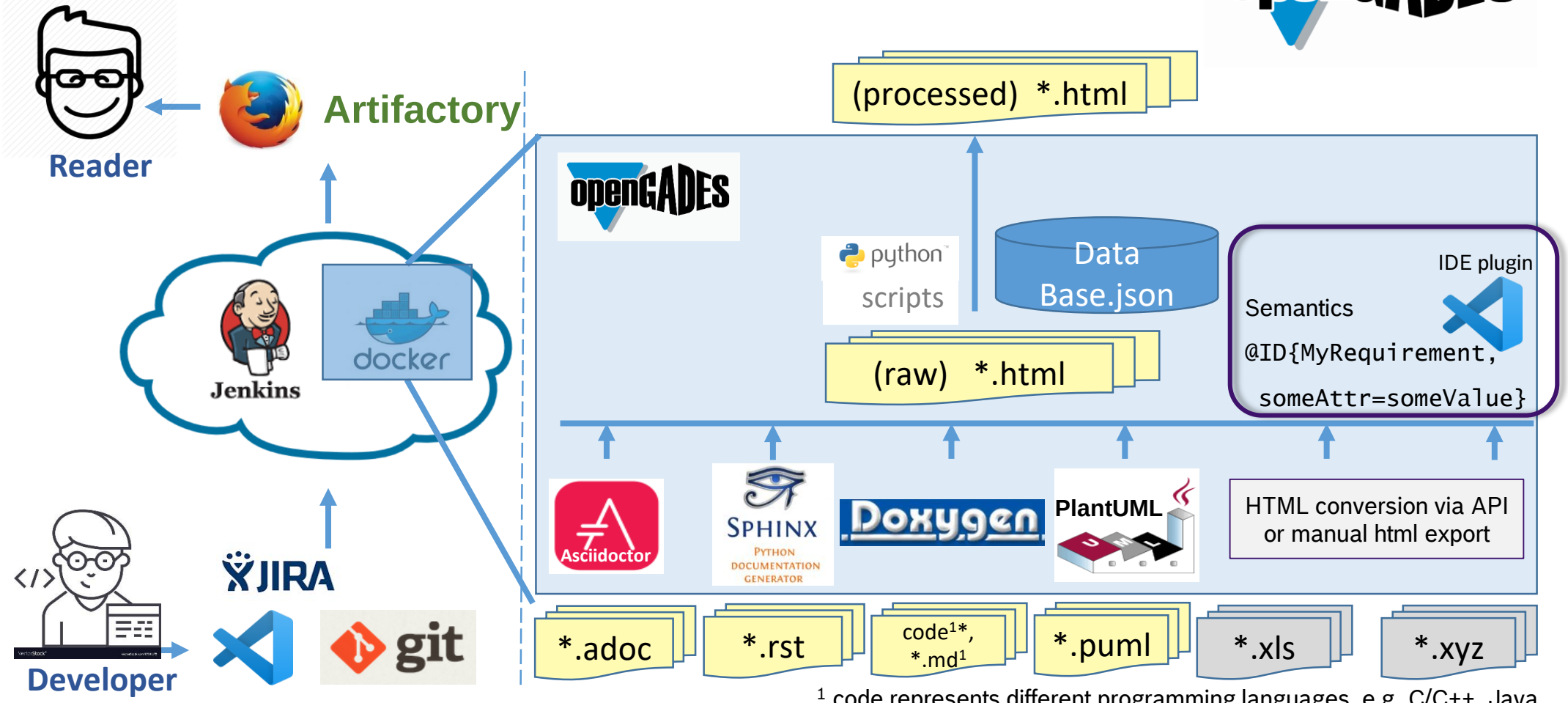
```
Due to the requirement
@IDREF{SWR_MyRequirement} we take
the following approach
```

Cross Reference

BOSCH

Docs-as-Code with openGADES

OpenGADES: How does it work - Internal structure



[1] code represents different programming languages, e.g. C/C++, Java, including markdown files

openGADES
Internal structure: building blocks

3rd Layer: Visualization (Use-Case specific views, diagrams, reports)
Based on the formal model data base:
e.g. Requirements Flow Diagram, Component Structure, Traceability Graph, Signal Flow, Deployment Diagramm, ...

JavaScript HTML Data-Driven Documents

2nd Layer: Traceability and the formal model as database
Construction of the formal model as database (json, html, csv), including plausibility checks (e.g. validity of links)
Linking between artifacts for traceability

python

1st Layer: Documentation
Raw documentation in ascii files, export to HTML (e.g. Asciidoc + Doxygen)
Every browser suitable for preview
Alternatives: all other source formats which can be exported to HTML

Asciidoctor SPHINX Python Documentation Generator Doxygen PlantUML

BOSCH

openGADES
Experience summary: openGADES@AVP

Nimmo Matthew (CS/ENG-AVP1) hat 06.06.2019 kommentiert

In the project AVP (Automated Valet Parking) we developed a safety-critical, distributed, system of systems from scratch. Practically all our documentation, i.e. requirements engineering, architecture, test specifications was done next to the code. In fact we see the documentation as so integral to the system that we like to say we "code in English".
We sought a text-based documentation system (in order to make use of all the nice code handling tools available), opted for asciidoctor (because as a LML the source is very readable) and integrated it into our continuous toolchain. It renders very nicely in html where traceability is realised with hyperlinks. Whether you change software or natural language documents, workflow is quite similar.
Used properly, it satisfies the requirements of functional safety, as attested by TÜV who is assessing our system.
It has spawned the BIOS project openGADES where it is being further developed and can be appropriated by other projects.
For projects that deal with a high standard for documentation, I recommend openGADES.

https://connect.bosch.com/blogs/f32e7a82-e978-4f57-a030-20098aef2180/entry/How_to_use_Social_Coding_for_Documentation?lang=de_de#threadid=7ab7edbb-dc4f-47be-9da5-bcc5b9c6376e

openGADES
Community Approach

I generally like it but I need this feature a lot faster

It‘s cool but I miss a feature, that is not on the backlog

Well it would be great, if we could continue using our test tool, but that is currently not supported

openGADES
Community Approach

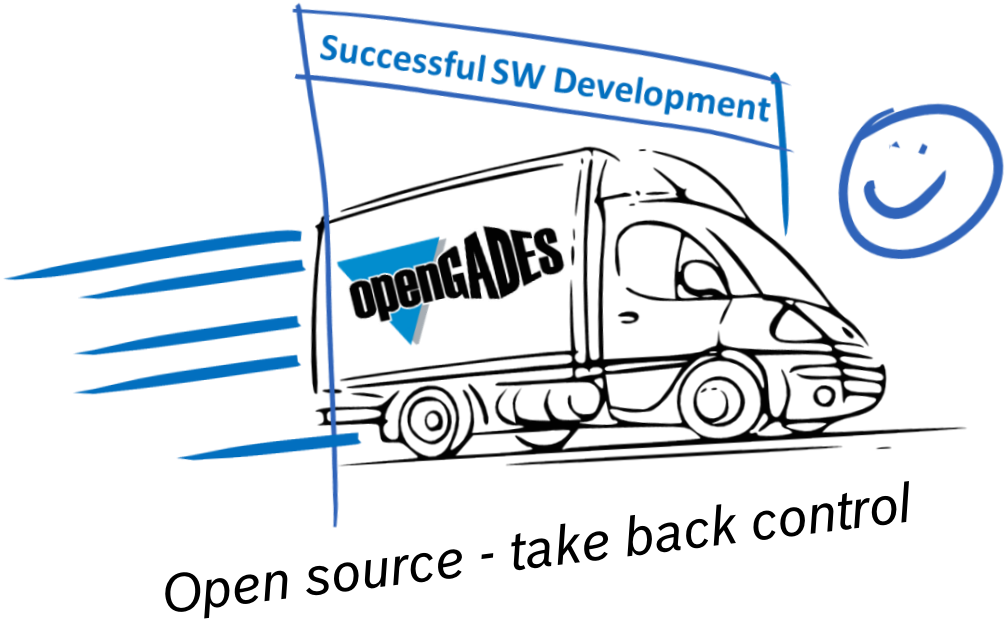
Contributions are welcome ...

XC-AD/EYM2 | 05.05.2021

openGADES
Summary

- openGADES is a proven in use toolchain alternative for development of (safety critical) software

- community approach to enable a demand driven feature growth



XC-AD/EYM2 | 05.05.2021

openGADES