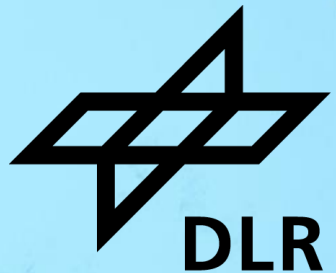
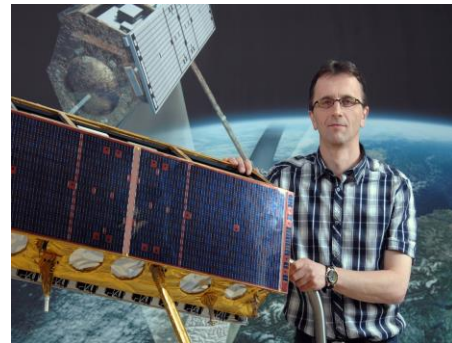


RESEARCH SOFTWARE ENGINEERING

Unabdingbar für kleine Analysenskripte ebenso wie für die Raketensteuerung



Softwareentwicklung in der Forschung



(Fast) jeder schreibt code

...aber die meisten Entwickelnden sind keine SoftwareentwicklerInnen!

Deutsches Zentrum für Luft- und Raumfahrt (DLR)

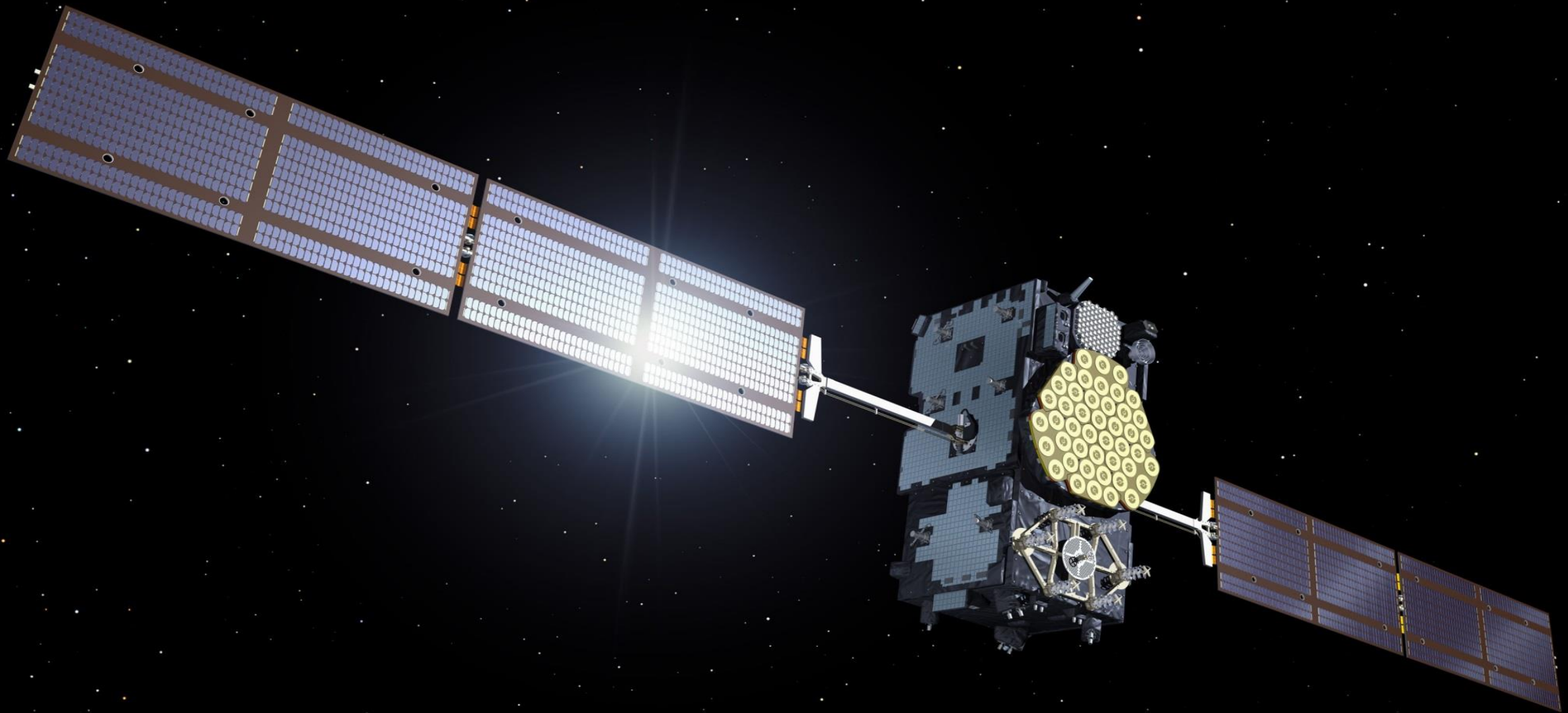


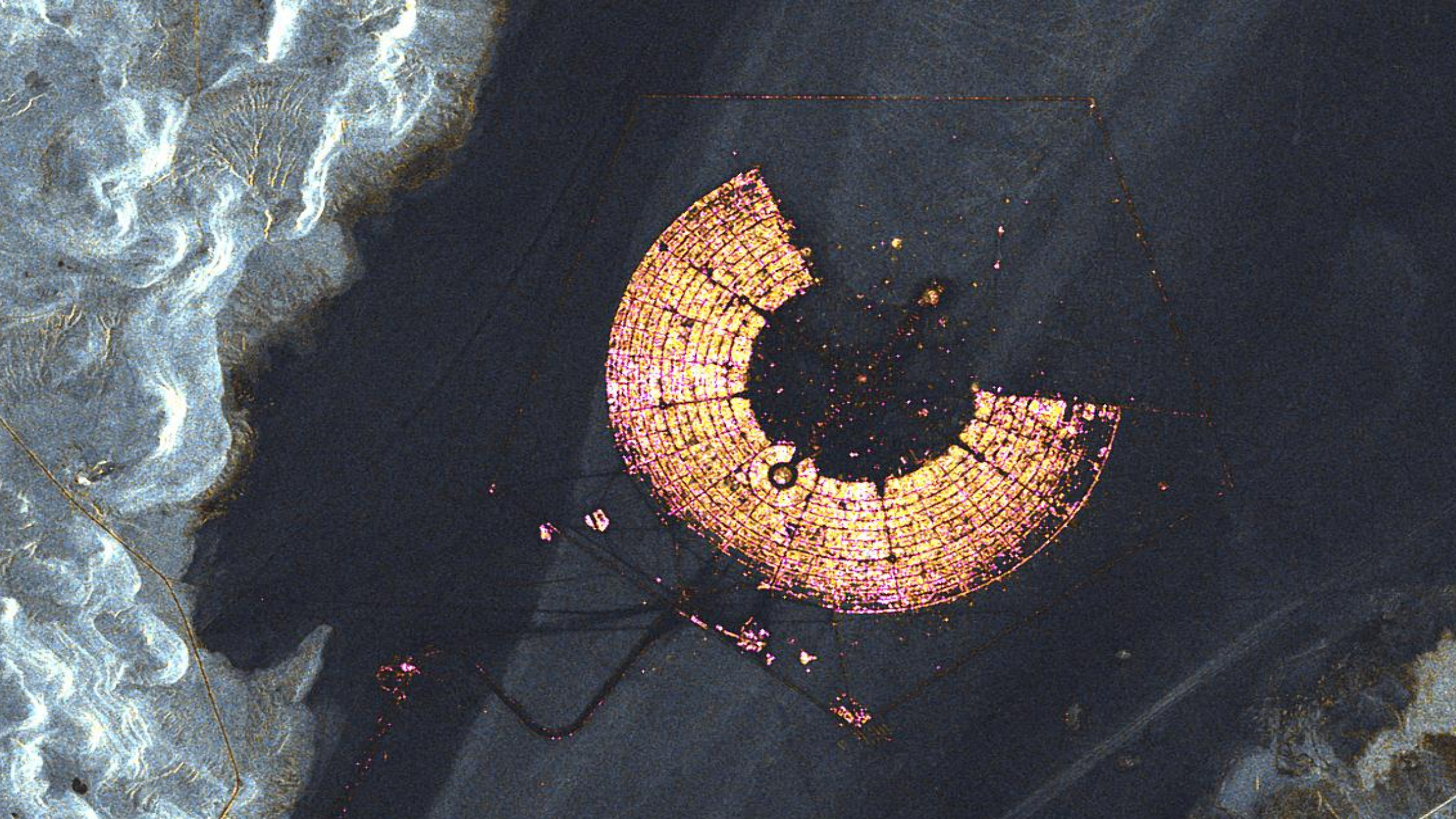
In Zahlen

- 10.000 Mitarbeitende
- 55 Institute und Einrichtungen
- 35 Standorte und Büros











D-CMET

DLR









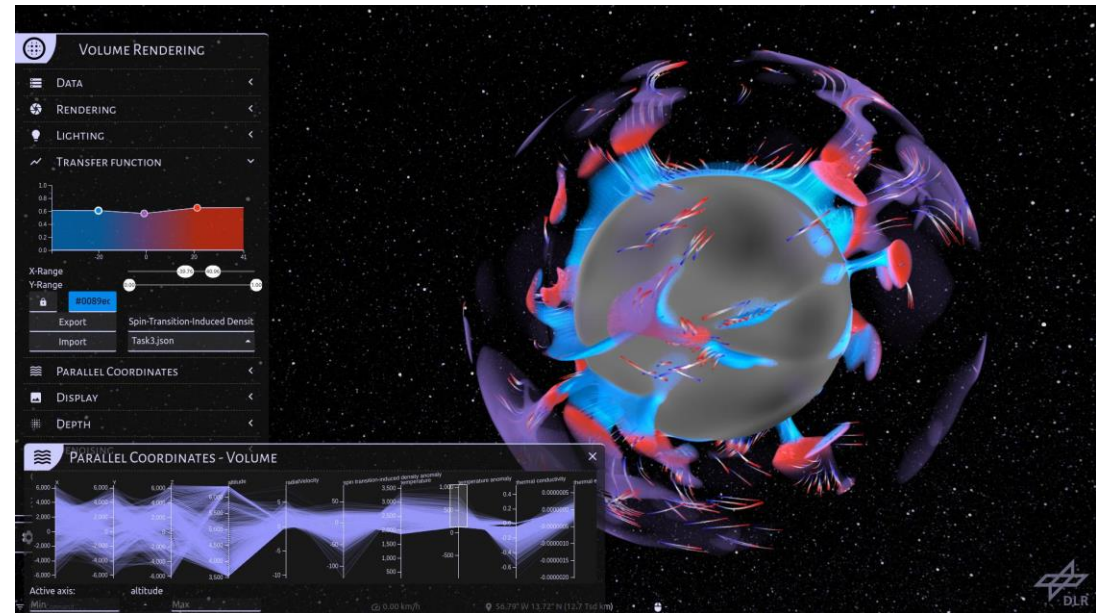
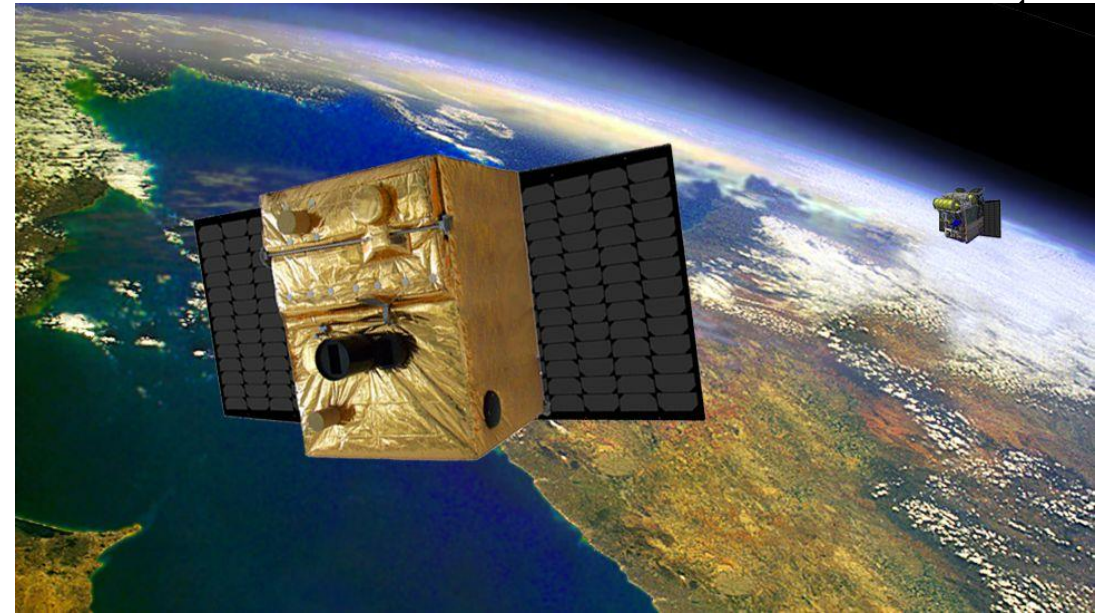
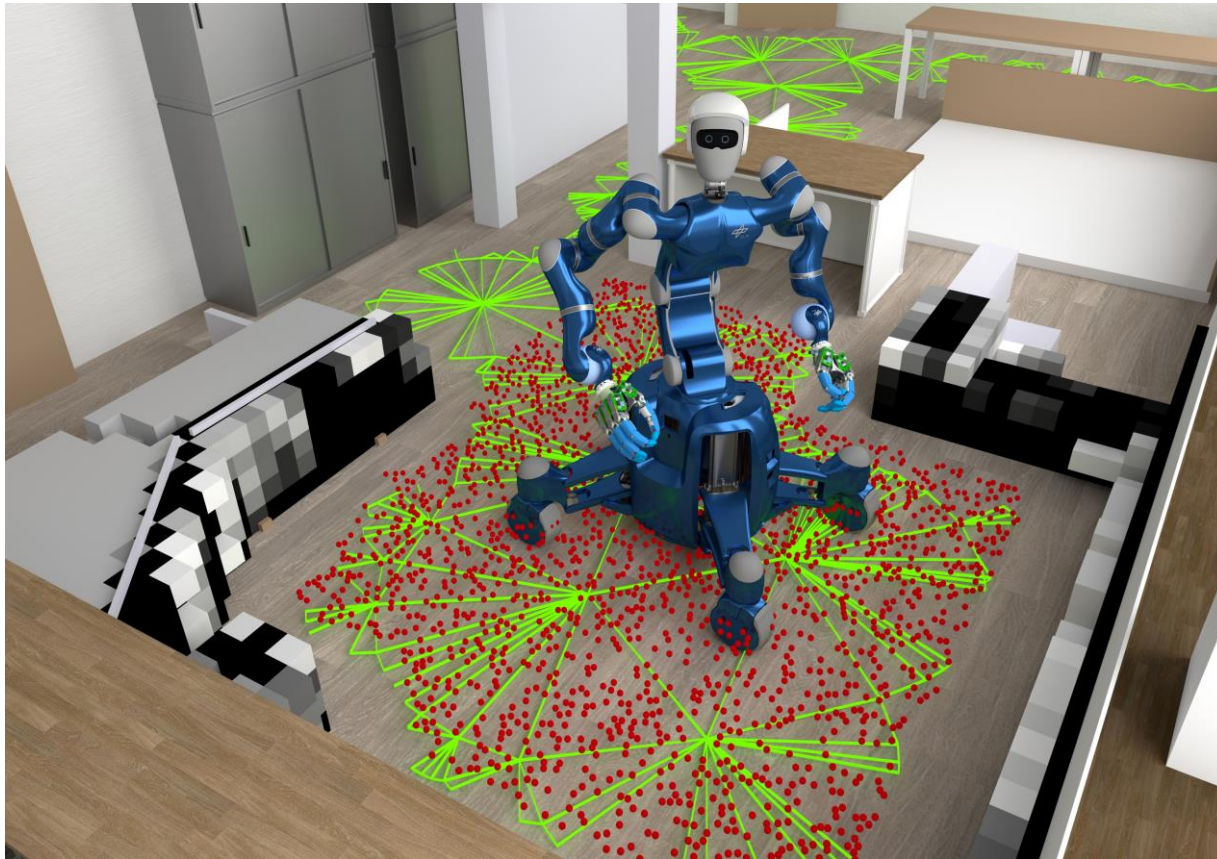


robomobil@dlr.de

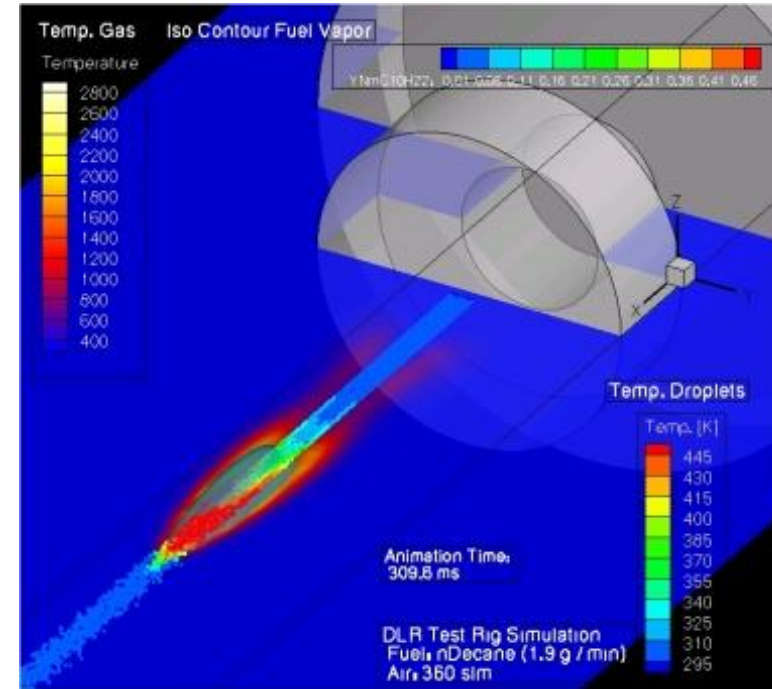
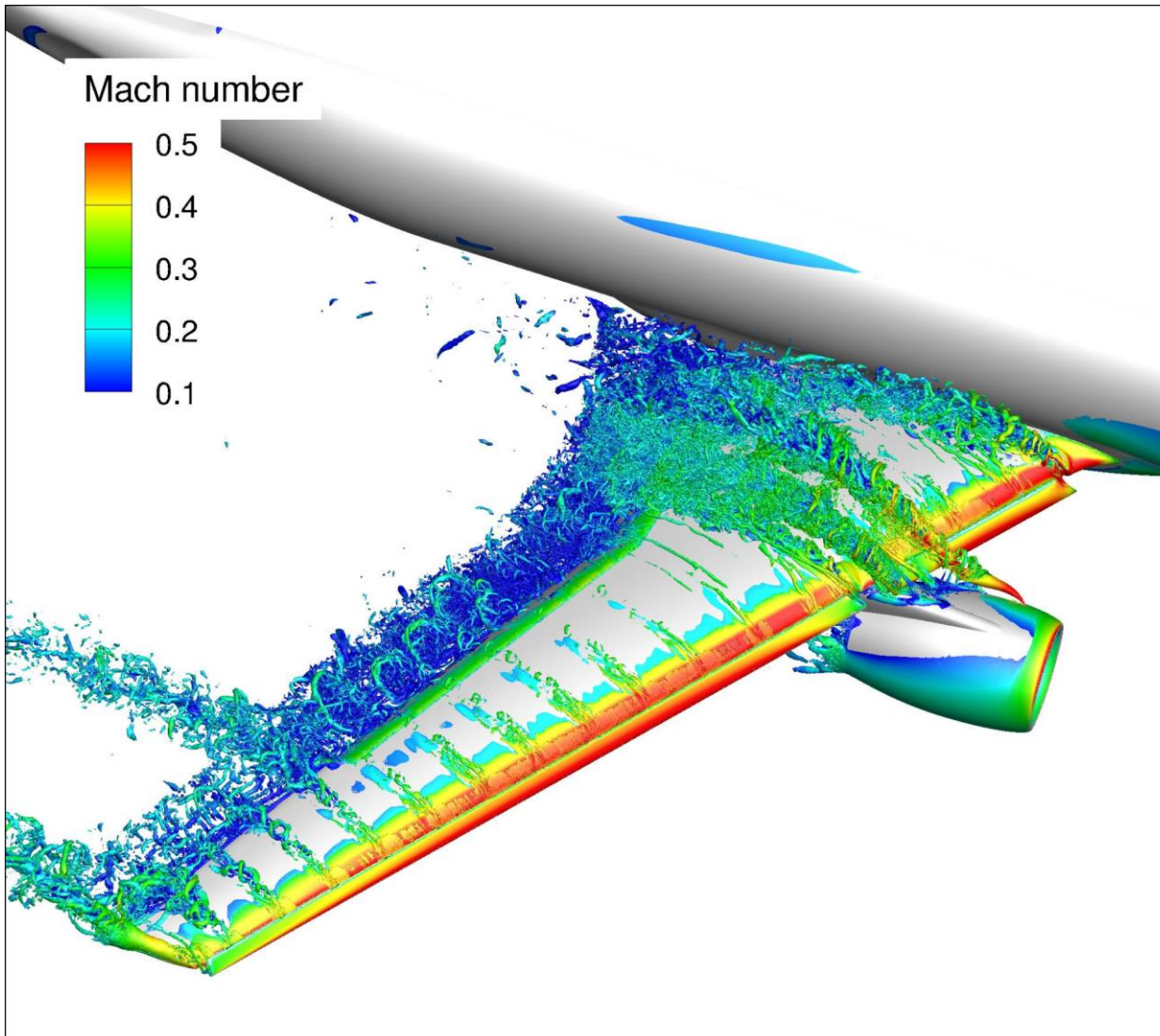
DLR

DLR SOFTWARE

Beispiele für DLR Software



Beispiele für DLR Software



Beispiele für DLR Software



The screenshot displays the RCE (RCE Client 1) software interface, which is used for managing complex engineering workflows. The main window shows a workflow graph with components like Aerodynamic, Optimizer, CoG cons, Mass model, Water mass, ThermoRAD, ThermoISO, Mesh, and Trajectory. A 3D model of a satellite component is visible in the TIGLViewer window.

Below the main workflow, there are several data tables and charts:

- Margin Table:**

sgl	Margin [%]	Mass with margin [kg]
		1982.40
	20.00	396.48
		2378.88
		1330.00
		300.00
		4008.88
		4150.00
		141.12
- Mode Overview Table: Satellite**

SEI	Parameter	Unit	default	SafeMode	Science
▲ Satellite	powerAvgTotal	W	249.320	101.405	183.154
▲ Power	powerAvgTotal	W	28.000	20.000	28.000
▶ SolarWings	powerAvgTotal	W	17.000	9.000	17.000
▲ Communication	powerAvgTotal	W	11.000	11.000	11.000
▶ Communication	powerAvg	W	23.320	21.004	23.320
▶ Antenna	powerAvgTotal	W	21.000	21.000	21.000
▶ Antenna	powerAvgTotal	W	2.320	0.004	2.320
▶ AOCs	powerAvgTotal	W	88.000	0.085	0.150
▶ Propulsion	powerAvgTotal	W	27.000	5.000	27.000
▶ Propulsion	powerAvg	W	27.000	5.000	27.000
▲ DataHandlingSystem	powerAvgTotal	W	53.000	77.000	53.000
▶ DataHandlingSystem	powerAvg	W	0.000	0.000	0.000
▶ CentralBoardComputer	powerAvgTotal	W	53.000	77.000	53.000
▶ CentralBoardComputer	powerAvg	W	53.000	77.000	53.000
▲ Structure	powerAvgTotal	W	30.000	0.000	30.000
- Parameter Mode Values:** A bar chart showing power values for Default, SafeMode, and Science modes across different subsystems.
- Subsystem contribution: Default Mode:** A pie chart showing the power contribution of various subsystems in the default mode.

Softwareentwicklung im DLR

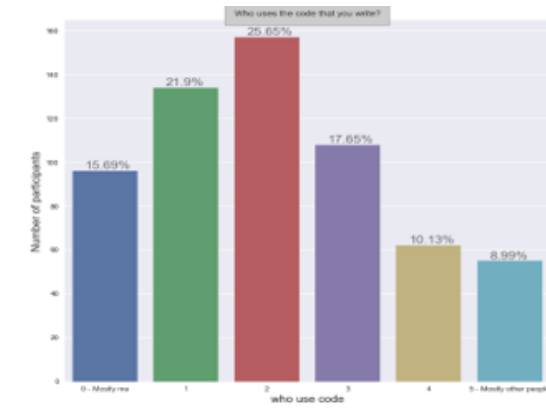
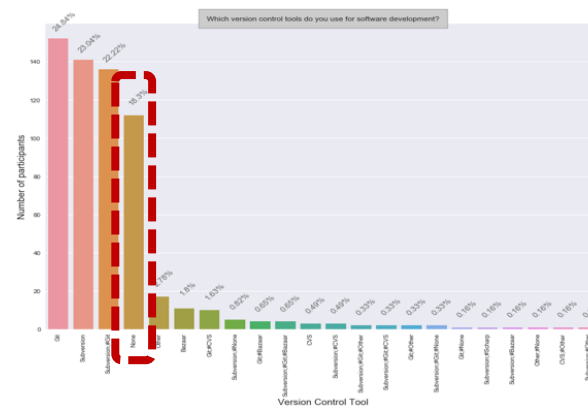
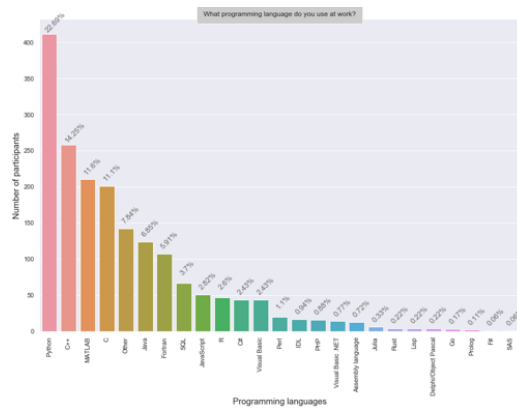


Softwareprojekte

Programmiersprache

Technologien (VCS)

Zielgruppe



Die Softwareprojekte des DLR sind in vielerlei Hinsicht vielfältig

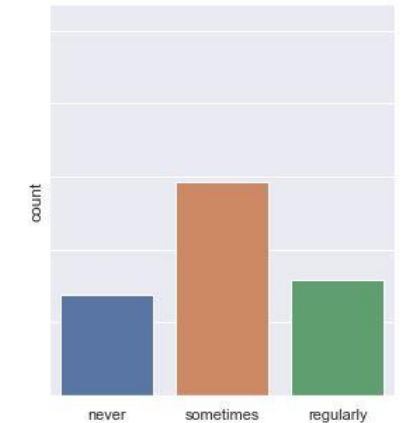
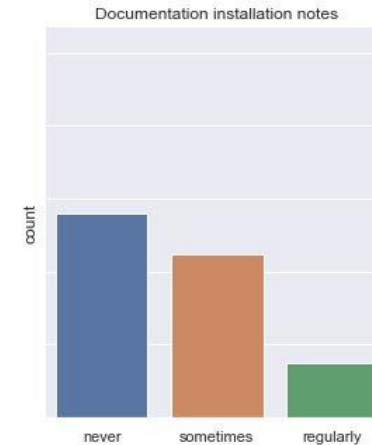
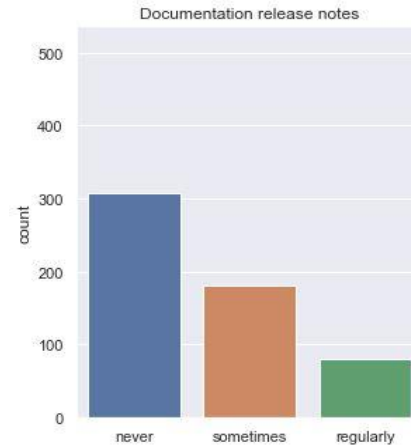
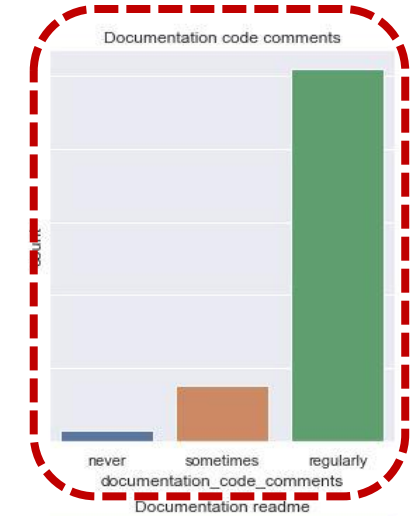
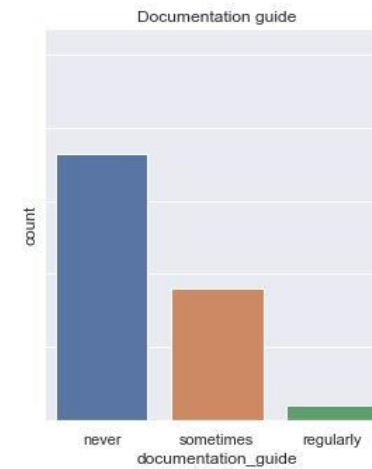
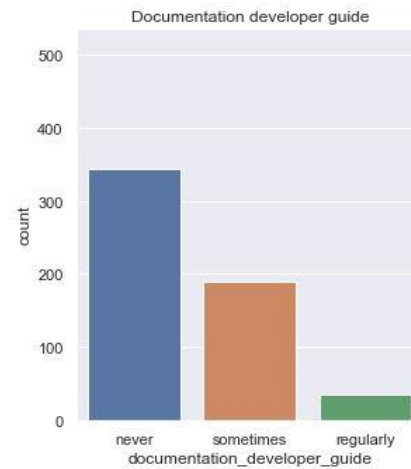
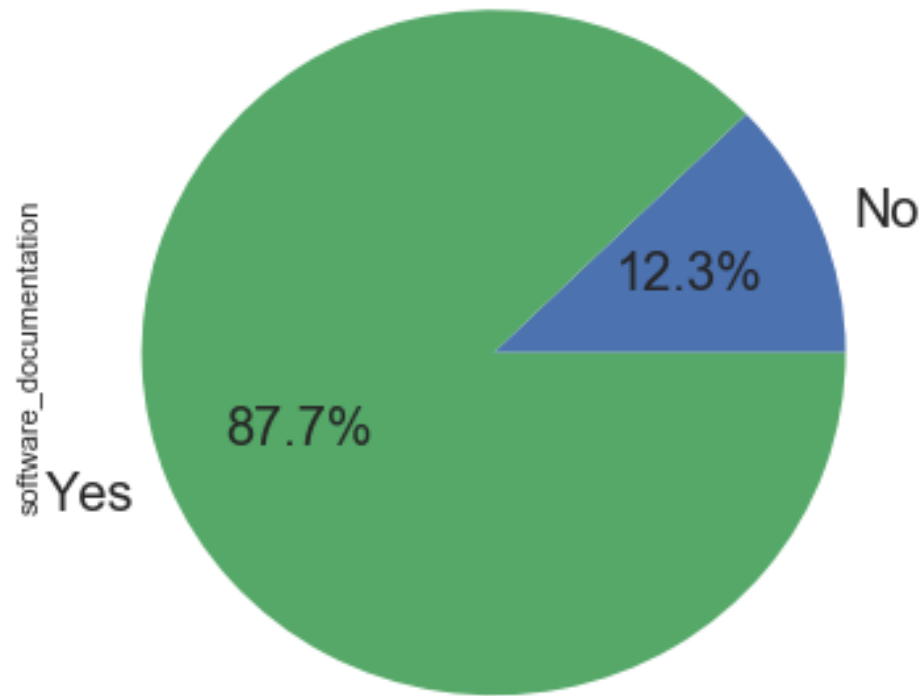
...die Probleme sind jedoch eher klassisch.

Softwareentwicklung im DLR



Dokumentation

Are you documenting your software?

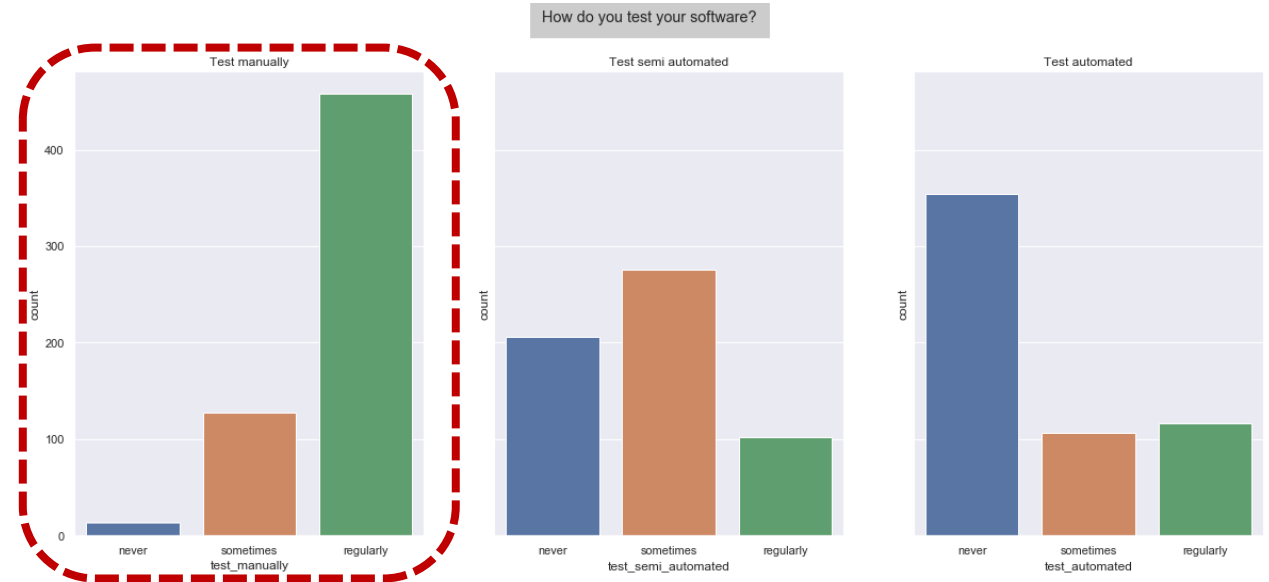
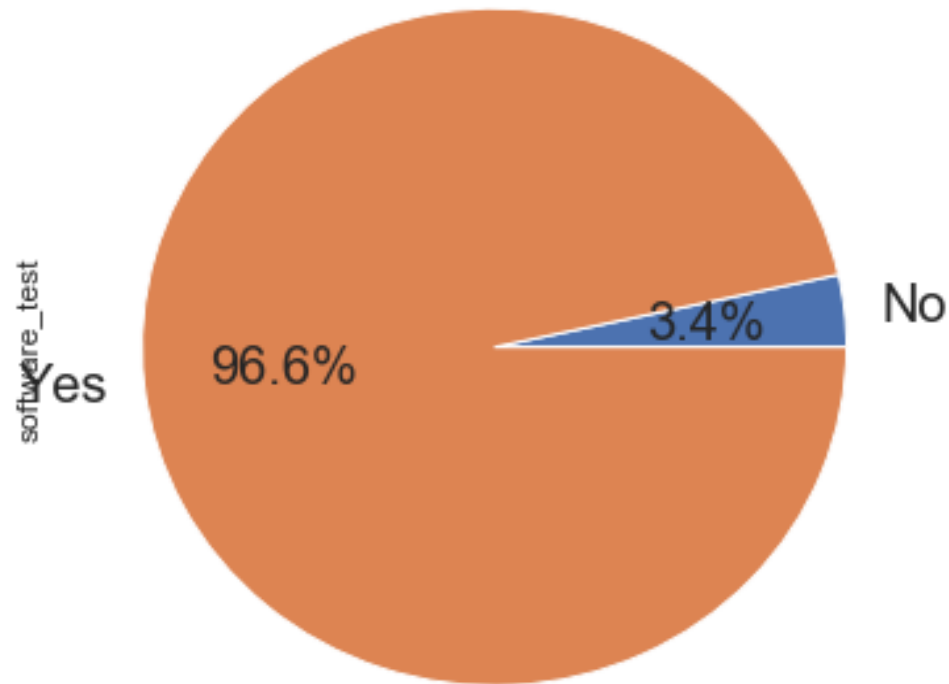


Softwareentwicklung im DLR



Testing

Are you testing your software?



Softwareentwicklung im DLR



Es fehlt an...

Ressourcen



Motivation



Wissen



Es braucht Personal & Infrastruktur, Incentives & Erfolgsgeschichten sowie Schulungen & Beratung.

Sustainable Software Engineering



Unsere Mission

Wir verbessern die Softwarequalität im DLR, in Helmholtz und weltweit.

Wir erforschen, entwickeln und implementieren Software Engineering-Lösungen.

Wir setzen uns dafür ein, dass Software als eigenständiges Forschungsergebnis anerkannt wird.



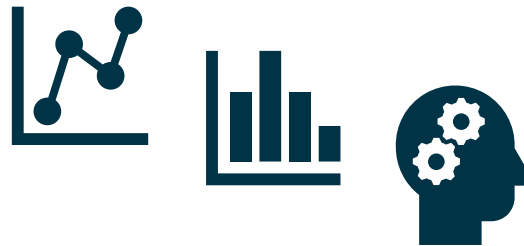
Ein wesentlicher Teil unserer Arbeit dreht sich um Research Software Engineering (RSE).

Unsere Forschungsbereiche



Research Software Engineering

- Wissensvermittlung
- Richtlinien
- Konsultation
- Softwareentwicklung
- Werkzeuge



Softwareanalytik

- Repository Mining
- Knowledgegraphen
- Provenance



Open Science

- Open Source
- (Forschungs-) Datenmanagement
- Offene (Meta-) Datenformate
- Reproduzierbarkeit

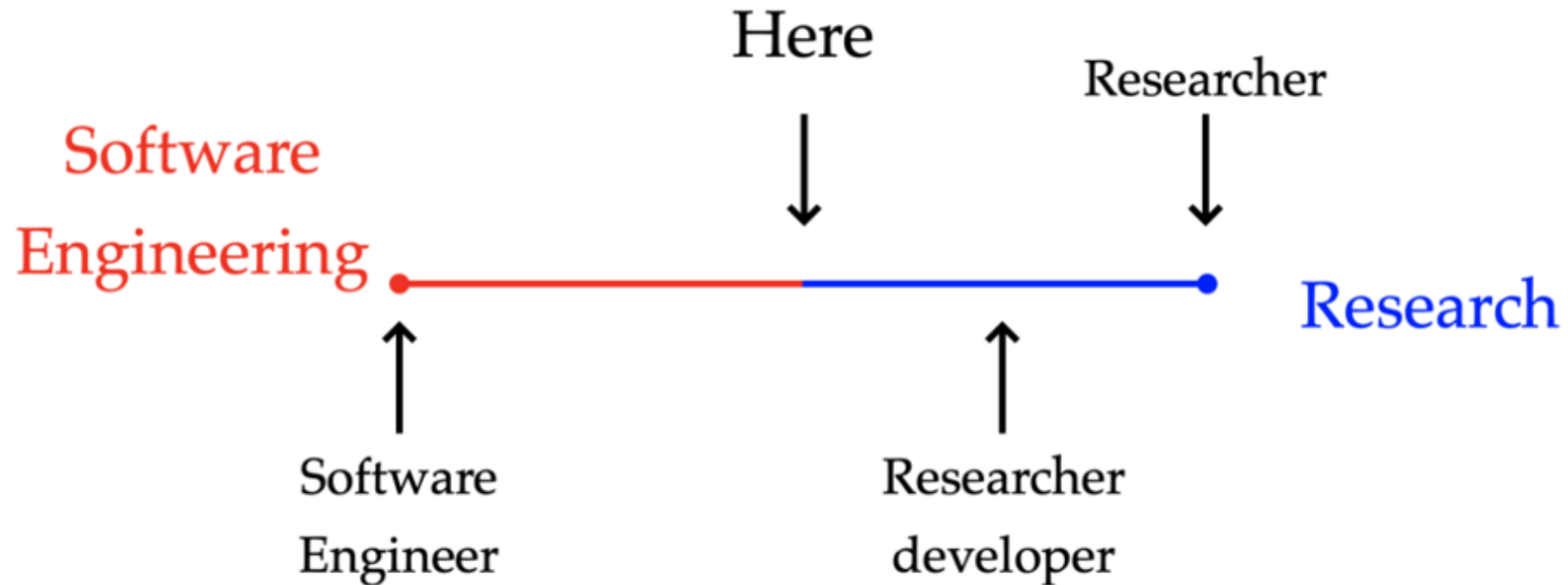
Research
Software
Engineering

DEFINITION

Research Software Engineering (RSE)



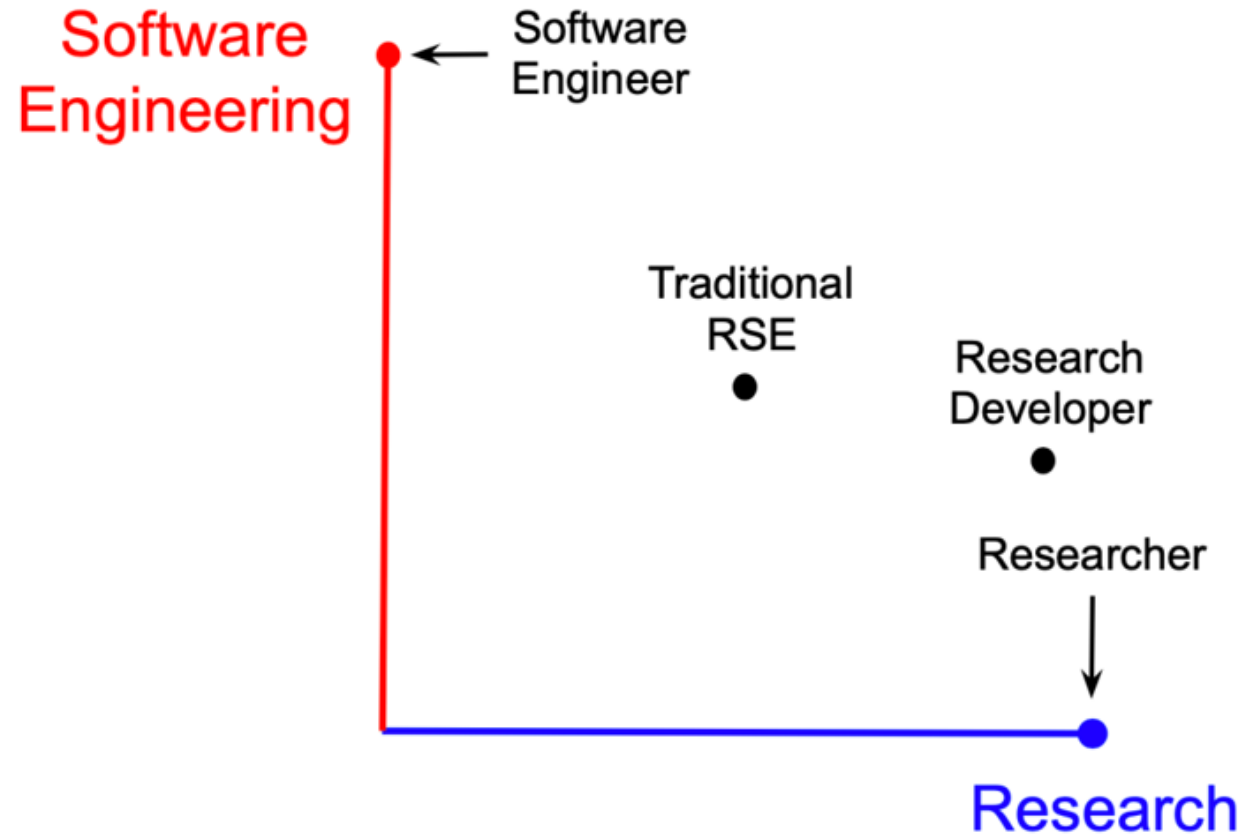
Definition (erster Versuch)



Research Software Engineering (RSE)



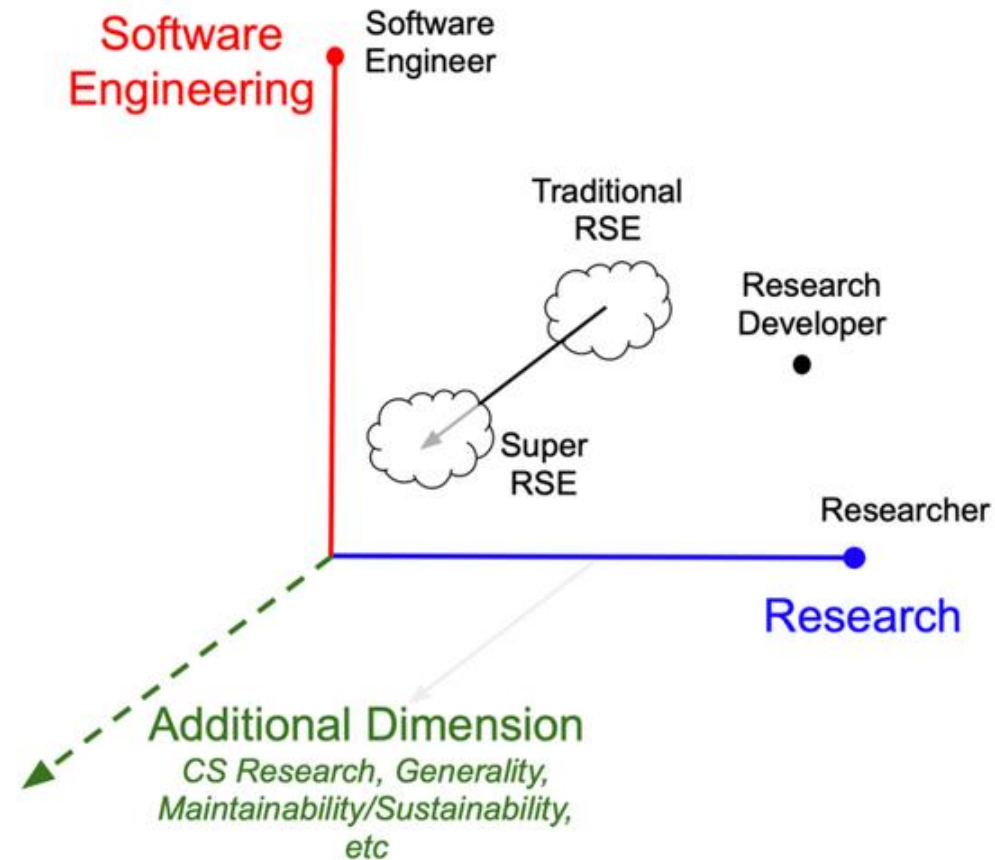
Definition (zweiter Versuch)



Research Software Engineering (RSE)



Definition (dritter Versuch)



Research Software Engineering (RSE)



In der Praxis

- Understand research (projects)
- Develop
- Operate
- Train



Research
Software
Engineering

PRAKTISCHE UMSETZUNG

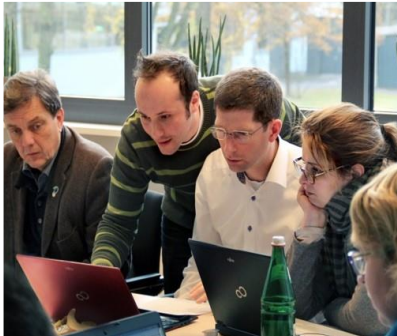
Software Engineering Initiative des DLR




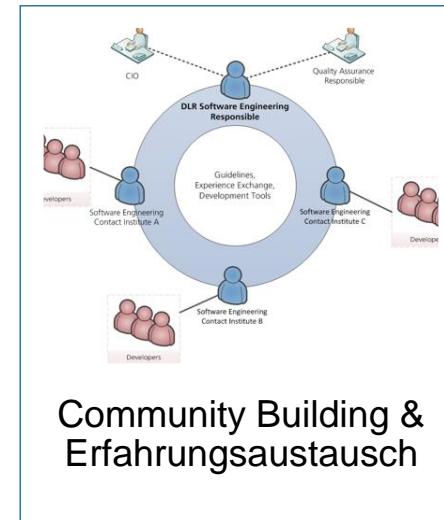
Change Management

Recommendation	Comment
EÄM.2: The most important information describing how to contribute to development are stored in a central location. <i>(from application class 1)</i>	Build steps are missing
EÄM.5: Known bugs, important unresolved tasks and ideas are at least noted in bullet point form and stored centrally. <i>(from application class 1)</i>	
EÄM.7: A repository is set up in a version control system. The repository is adequately structured and ideally contains all artifacts for building a usable software version and for testing it. <i>(from application class 1)</i>	
EÄM.8: Every change of the repository ideally serves a specific purpose, contains an understandable description and leaves the software in a consistent, working state. <i>(from application class 1)</i>	

Guidelines & Tools



Trainings



Consulting

- 2005 gestartet
- 2017 Fokussierung auf Entwicklung von Forschungssoftware
- Angetrieben durch Institut für Softwaretechnologie, finanziert durch DLR IT

Guidelines

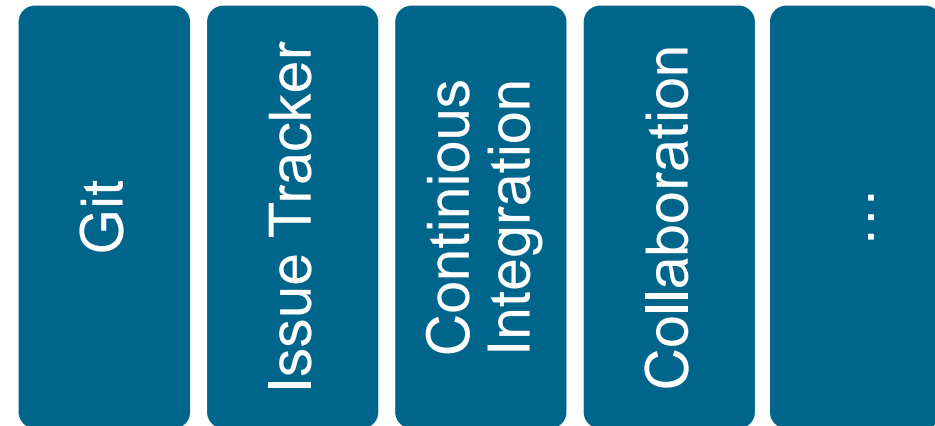
Empfehlungen für Entwickelnde

Change Management

Recommendation	Comment	Status
EÄM.2: The most important information describing how to contribute to development are stored in a central location. <i>(from application class 1)</i>	Build steps are missing	todo
EÄM.5: Known bugs, important unresolved tasks and ideas are at least noted in bullet point form and stored centrally. <i>(from application class 1)</i>		ok
EÄM.7: A repository is set up in a version control system. The repository is adequately structured and ideally contains all artifacts for building a usable software version and for testing it. <i>(from application class 1)</i>		ok
EÄM.8: Every change of the repository ideally serves a specific purpose, contains an understandable description and leaves the software in a consistent, working state. <i>(from application class 1)</i>		ok

Tools

Tools um Empfehlungen umzusetzen



DLR Gitlab Instanz

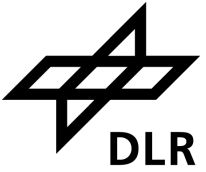
Trainings

- Introduction to Git and Gitlab
- Foundations of Research Software Publication
- Gitlab for Software Development in Teams

<https://codebase.helmholtz.cloud/hifis/software/education/hifis-workshops>

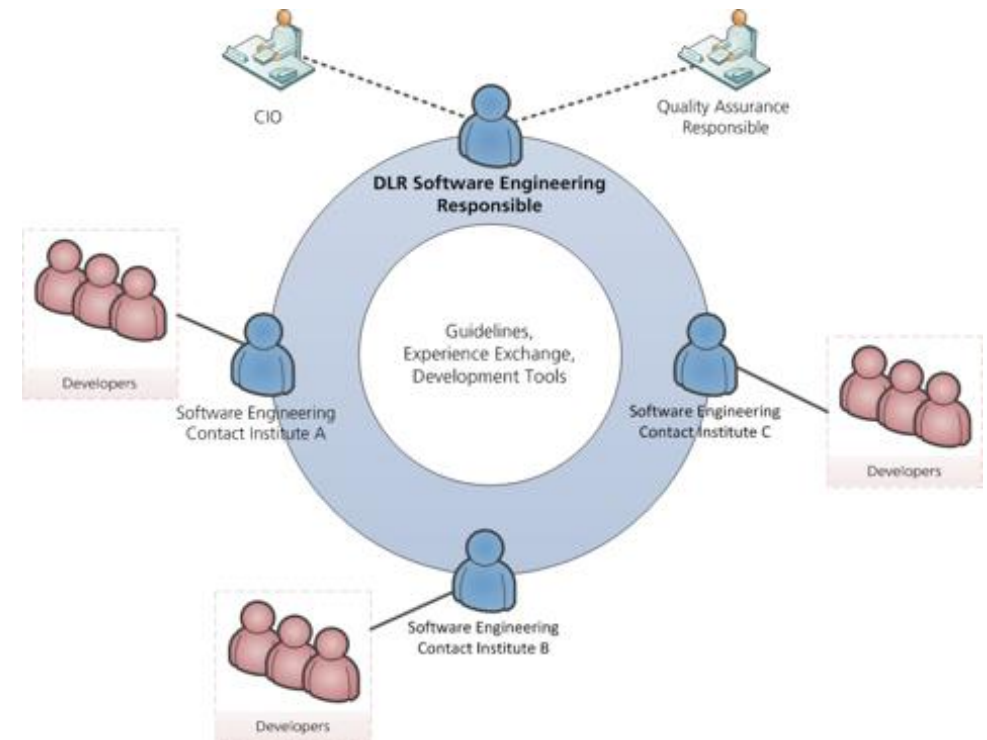


Software Engineering Initiative des DLR



Community Building & Erfahrungsaustausch

- SE Netzwerk
- Wissensaustauschworkshops
- DLR Software Mattermost Team
- SoftwareEngineering.Wiki



CONSULTING

Autonome optische Qualitätskontrolle von CSP Kraftwerken



QFly

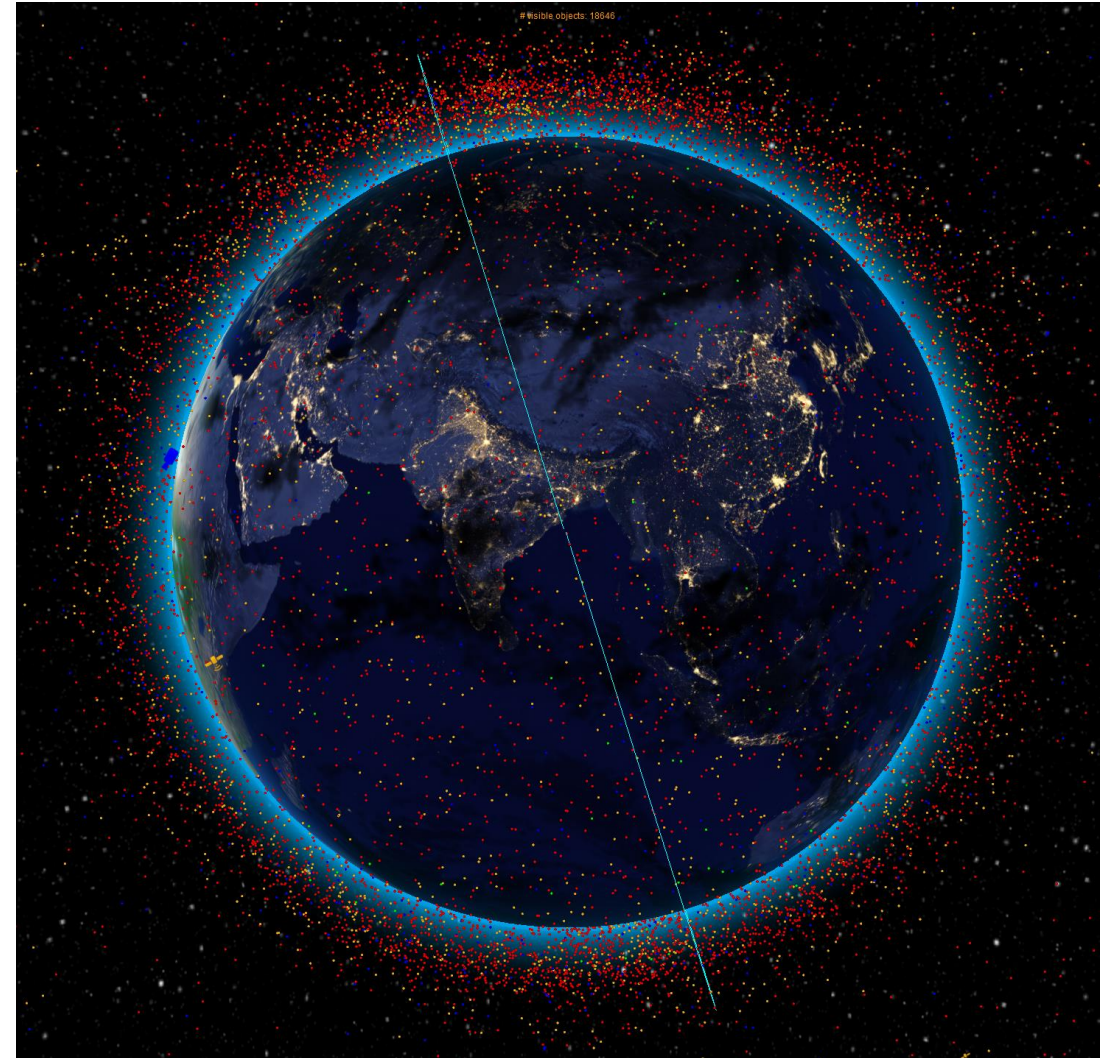
- Entwicklungsumgebung aufsetzen
 - Git
 - Issue Tracker
 - CI
 - Testing Infrastruktur
- Testkonzept einführen und beispielhaft umsetzen
- Entwicklungsprozess entwickeln und implementieren (iterativ)
- ProjektmitarbeiterInnen schulen



Sammlung und Verarbeitung der Orbit Informationen von Objekten im Weltall

Bacardi

- Requirements Engineering (nach ECSS)
- Unterstützung bei Design- und Architekturentscheidungen
- Administrationsstrategien für die Produktivmaschine
- Unterstützung bei der Suche und Behebung komplexer Fehler
- Code Reviews
- Paketierung



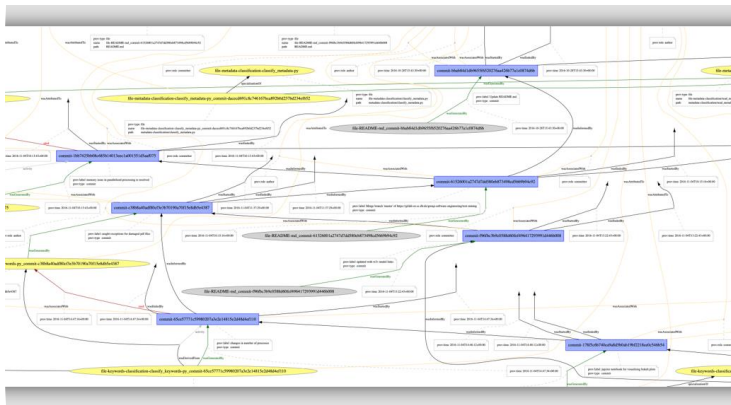
A young man with dark hair is wearing a Microsoft HoloLens augmented reality headset. He is looking directly at the camera with a slight smile. The background is a blurred blue and white gradient with faint, glowing green and yellow lines and text, suggesting a digital or data environment. The text visible in the background includes "array, shape=(n_point, n_feature", "to initialize the clusters", "realize the random state", "om_state is not None:", "m_seed(self.random_state)", "random":", "counts_displs_shape", and "(x.split".

SE SOFTWARE PROJEKTE

Software Engineering Prozesse analysieren

Software Repository Mining

Gitlab2Prov



Analysing Software Engineering Processes with Provenance-based Knowledge Graphs



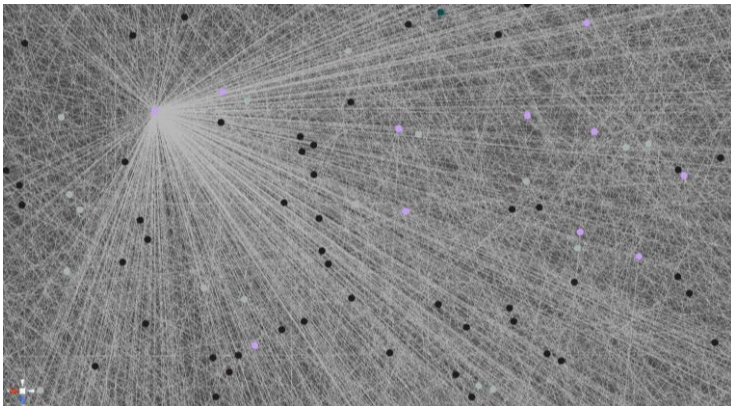
Beispiel Use Case: CWA



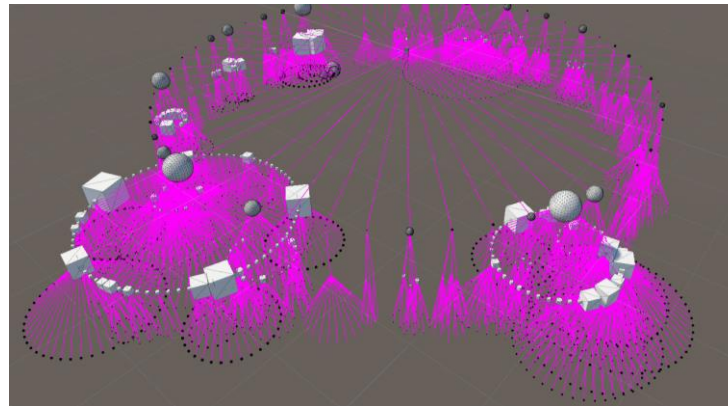
- Tool Entwicklung
- Analyse und Bewertung von Softwareprojekten
- Forschung

Graphvisualisierung

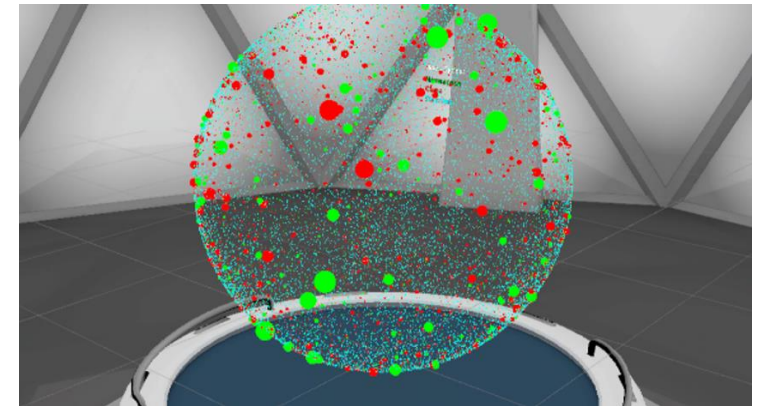
Klassisch



Hierarchisch



Räumlich

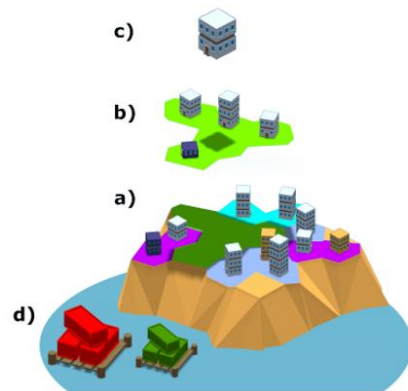


- Abstrakte Daten von Softwareprojekten (Quellcode, Abläufe, ...) liegen häufig in Graphenform vor
- Graphen sind nicht immer die ideale Darstellungsform für solche Daten.

Software Visualisierung

IslandViz

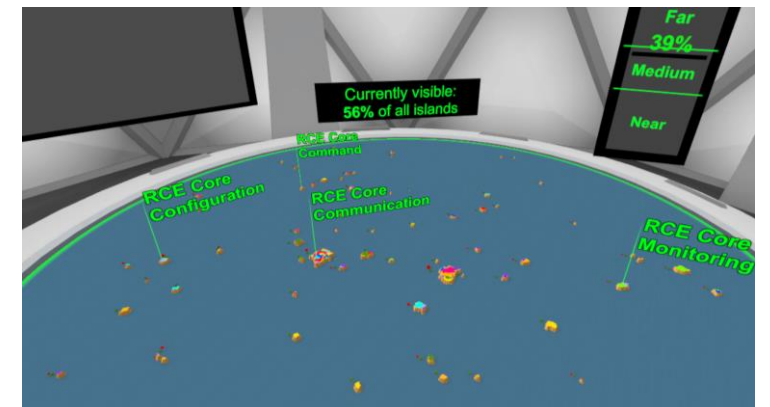
Elemente



Beziehungen



Assistenten



- Darstellung von Softwareprojekten mittels Insel-Metapher
- Interaktiv und Explorativ
- Assistentensysteme (Sprache, Chat, ...)

The background of the slide is a photograph of a city at night. A bright, vertical lightning bolt strikes down from a dark, cloudy sky, illuminating the scene. The city lights are visible in the distance, and the silhouettes of people can be seen in the foreground. A dark blue horizontal bar is overlaid on the bottom half of the image, containing the main title in white text.

WAS WÄRE RAUMFAHRT OHNE SE?

Ariane 5 Flight 501 (1996)

Transport von Satelliten in die Magnetosphäre der Erde



By esa

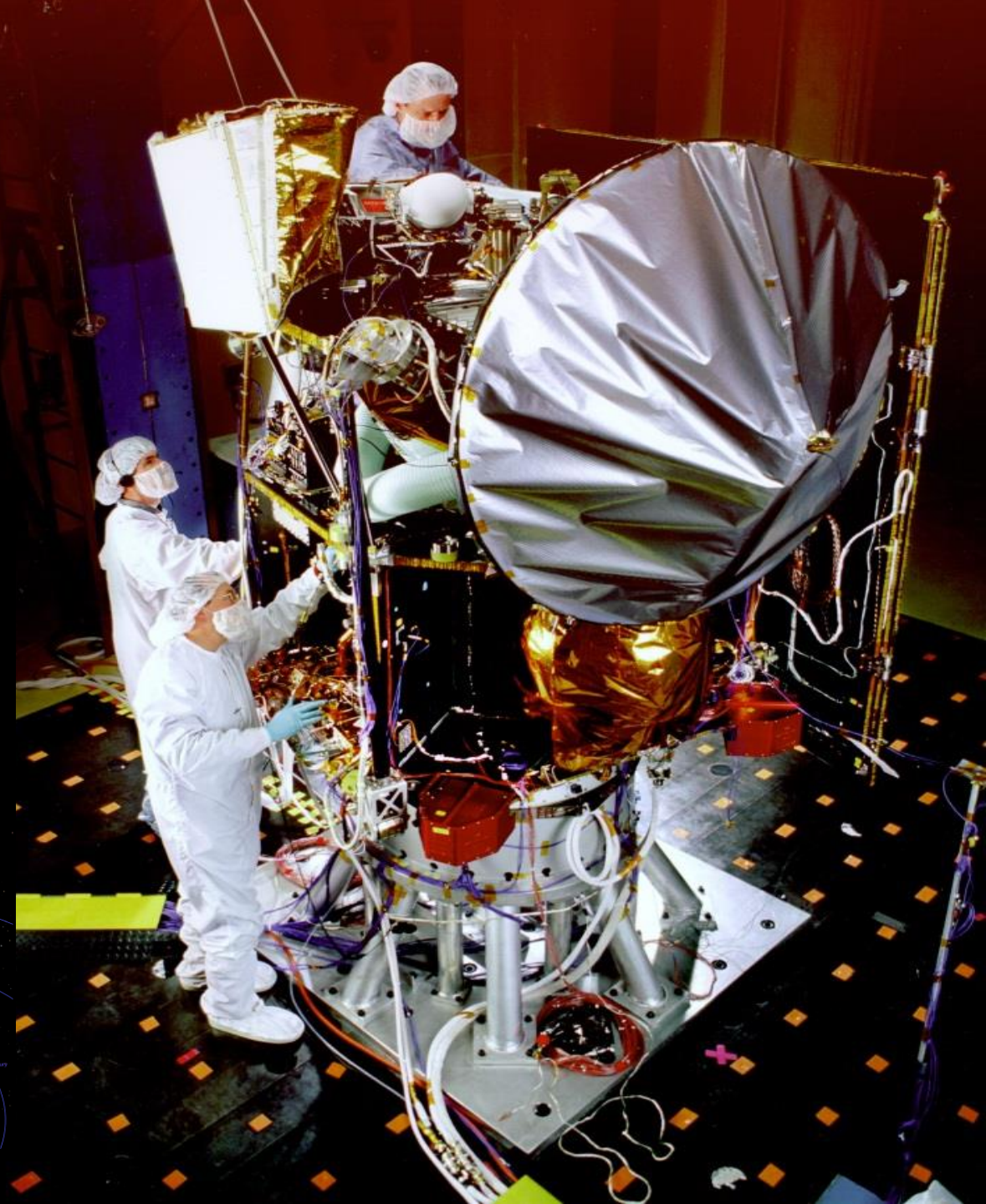
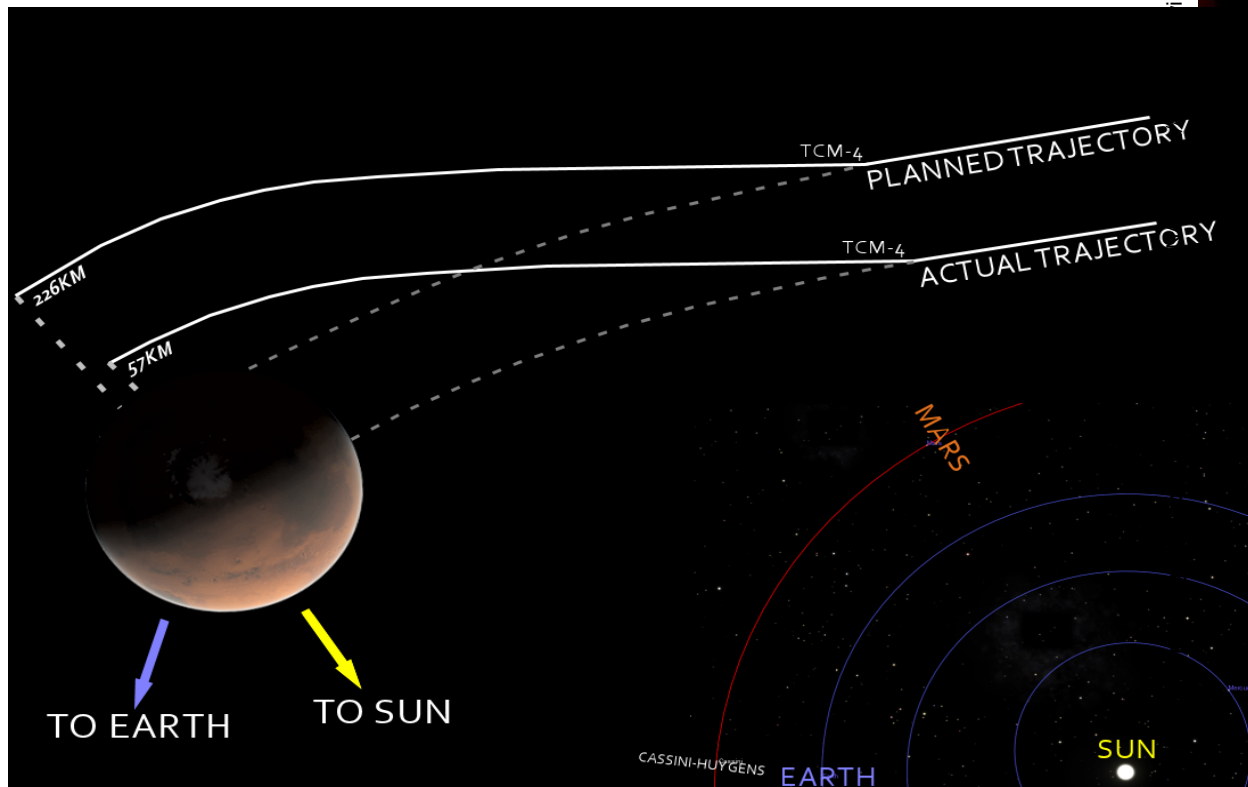


P. M.

Mars Climate Orbiter (1998)

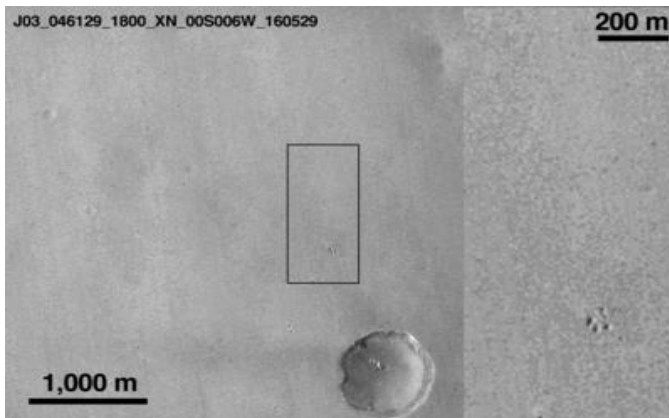
Mars erforschen und
Kommunikationsrelay

ikimedia.org/wiki/File:Mars_Climate_Orbiter_during_tests.jpg



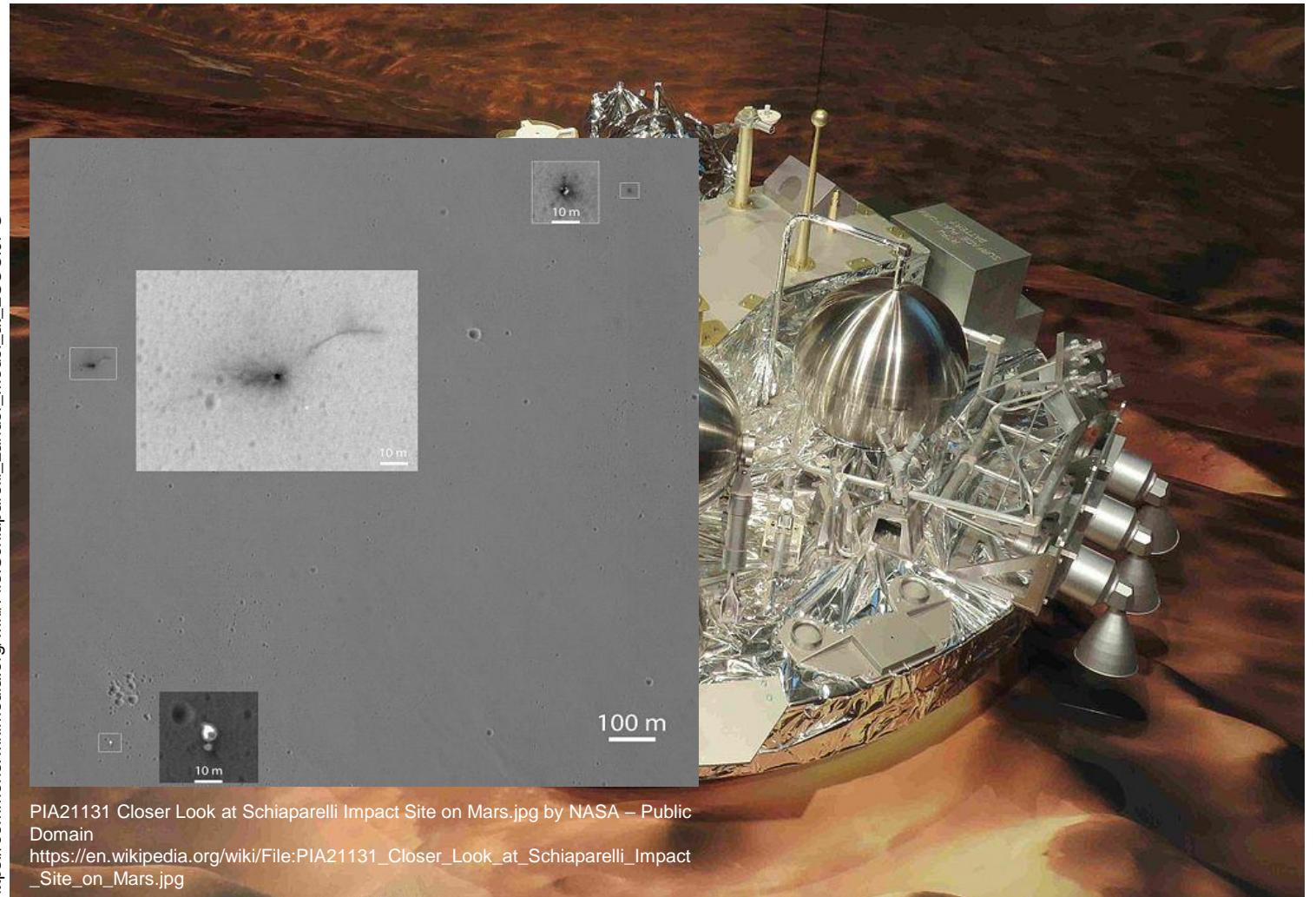
Schiaparelli (2016)

Landung auf Mars



PIA21130 Signs of Schiaparelli Test Lander Seen From Orbit.gif by NASA – Public Domain
https://en.wikipedia.org/wiki/File:PIA21130_Signs_of_Schiaparelli_Test_Lander_Seen_From_Orbit.gif

Schiaparelli Lander Model at ESOC.JPG by Gerbil – Public Domain
https://commons.wikimedia.org/wiki/File:Schiaparelli_Lander_Model_at_ESOC.JPG



PIA21131 Closer Look at Schiaparelli Impact Site on Mars.jpg by NASA – Public Domain
https://en.wikipedia.org/wiki/File:PIA21131_Closer_Look_at_Schiaparelli_Impact_Site_on_Mars.jpg

CARINA HAUPT

carina.haupt@dlr.de
@caha42(@scholar.social)



#WIRROCKENSOFTWARE