Scientific Software Engineering:
Mining Repositories to gain insights into BACARDI

Lynn von Kurnatowski[1], Martin Stoffers[1], Martin Weigel[2], Michael Meinel[1], Yi Wasser[2], Kathrin Rack[1], Hauke Fiedler[2]

German Aerospace Center (DLR)
[1]Institute for Simulation and Software Technology
[2]Space Operations and Astronaut Training

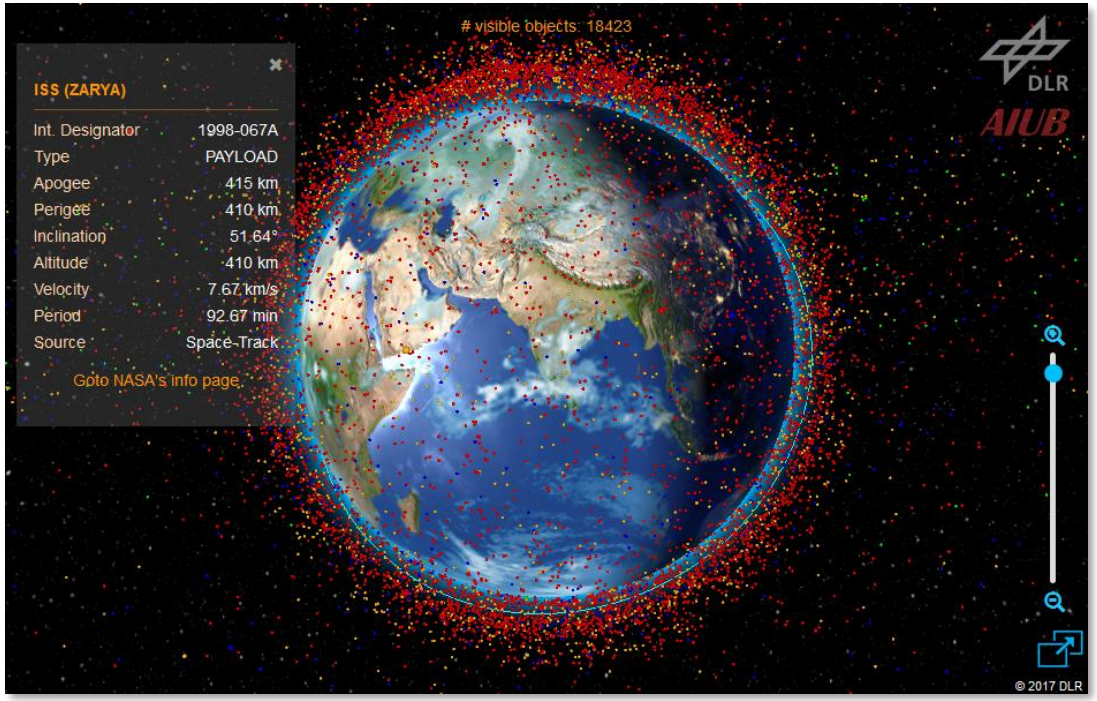
BACARDI – Backbone Catalogue of Relational Debris Information

BACARDI is a software platform to register and track orbital objects like space debris and satellites.

Background information:

- software was initially developed as a demonstrator
- during this phase the software was developed without a formal development process in place

- after the successful development of a prototype
  => the project has been continued as a long-term project
- to balance the character of a research project and operational requirements



a software engineering process has been designed and applied

Motivation

Highly distributed project and working environment

- 2 Institutes
- 4 Departments
- 3 physical locations
- 4-6 team members

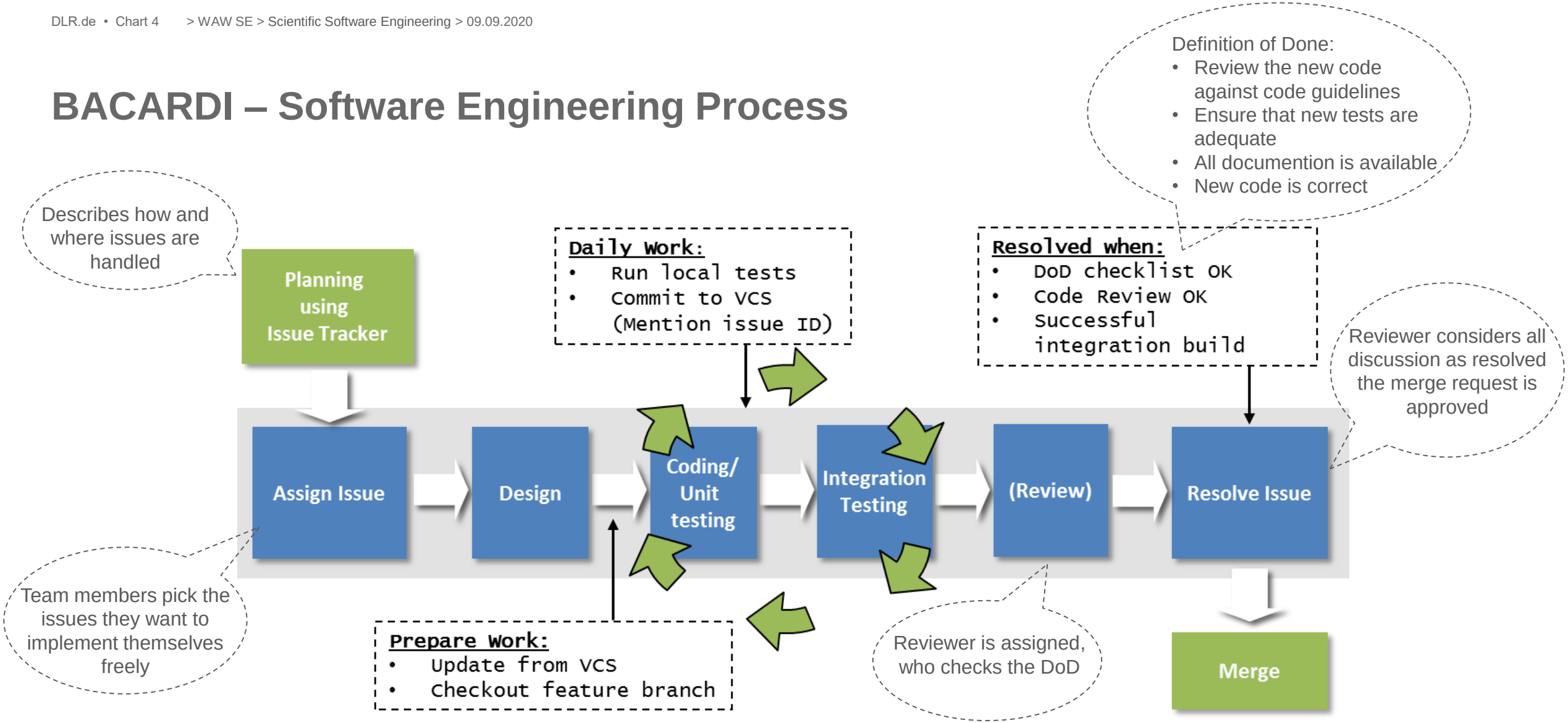
Different personal focus on the project based upon:

- research goals
- personal background

Communication difficulties due to:

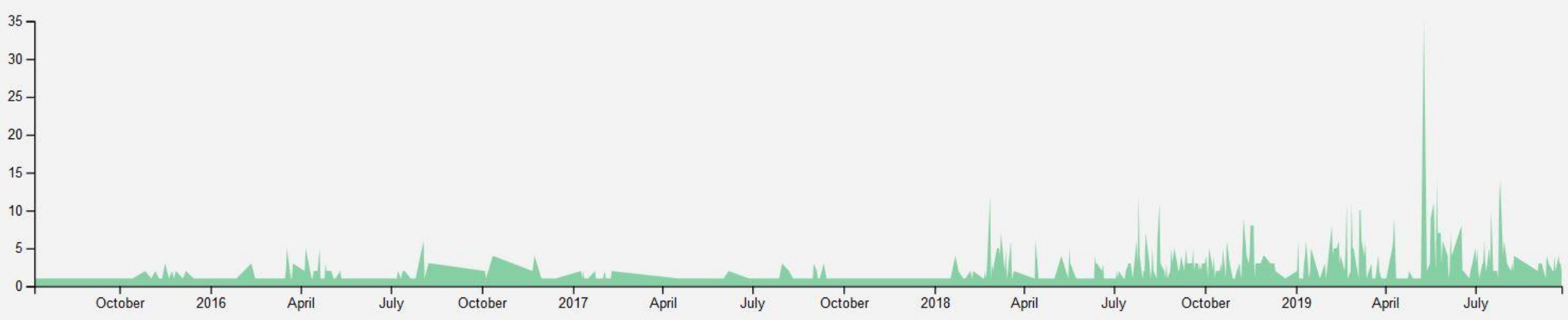
- technologies available to team members
- distributed workplaces of team members

BACARDI – Software Engineering Process



Can we measure a change?

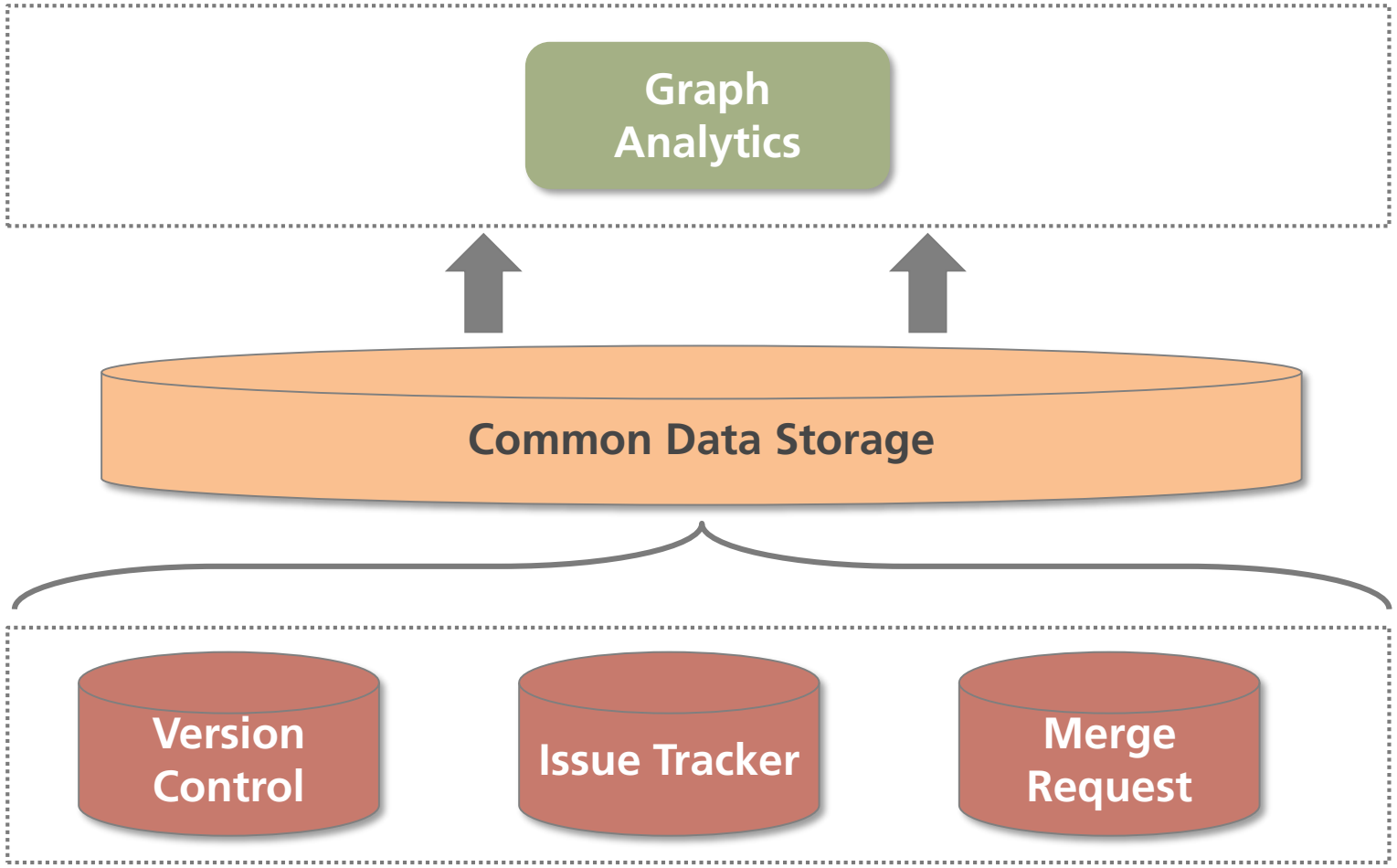
A first Pattern



Number of commits from July 2015 to September 2019

Solution:
Mining Repositories

Mining Software Repositories



Analysis and Usage

Data Integration and Provisioning

Repositories

Results Merge Request

```
MATCH (n:MergeRequest)
WITH* WHERE '2018-01-01' <= n.created_at <= '2018-03-31'
WITH* WHERE
    ((n)-[:IS_ASSIGNED]->(:User)) and
    ((n)-[:APPROVED_BY]->(:User)) and
    ((n)-[:WAS_ASSIGNED]->(:User)) and
    ((n)-[:MERGED_BY]->(:User)) and
    ((n)-[:CREATED_BY]->(:User))
WITH* WHERE n.approved = true and n.state = 'merged'
WITH*
MATCH (u1:User)<-[:CREATED_BY]-(n)-[:APPROVED_BY]->(u2:User)
WHERE u1.name <> u2.name
WITH*
MATCH (n)-[:IS_RELATED]->(:Issue)
WITH*
MATCH (n)-[:HAS_MILESTONE]->(m:Milestone)
WHERE n.merged_at <= m.due_date
RETURN count(*)
```

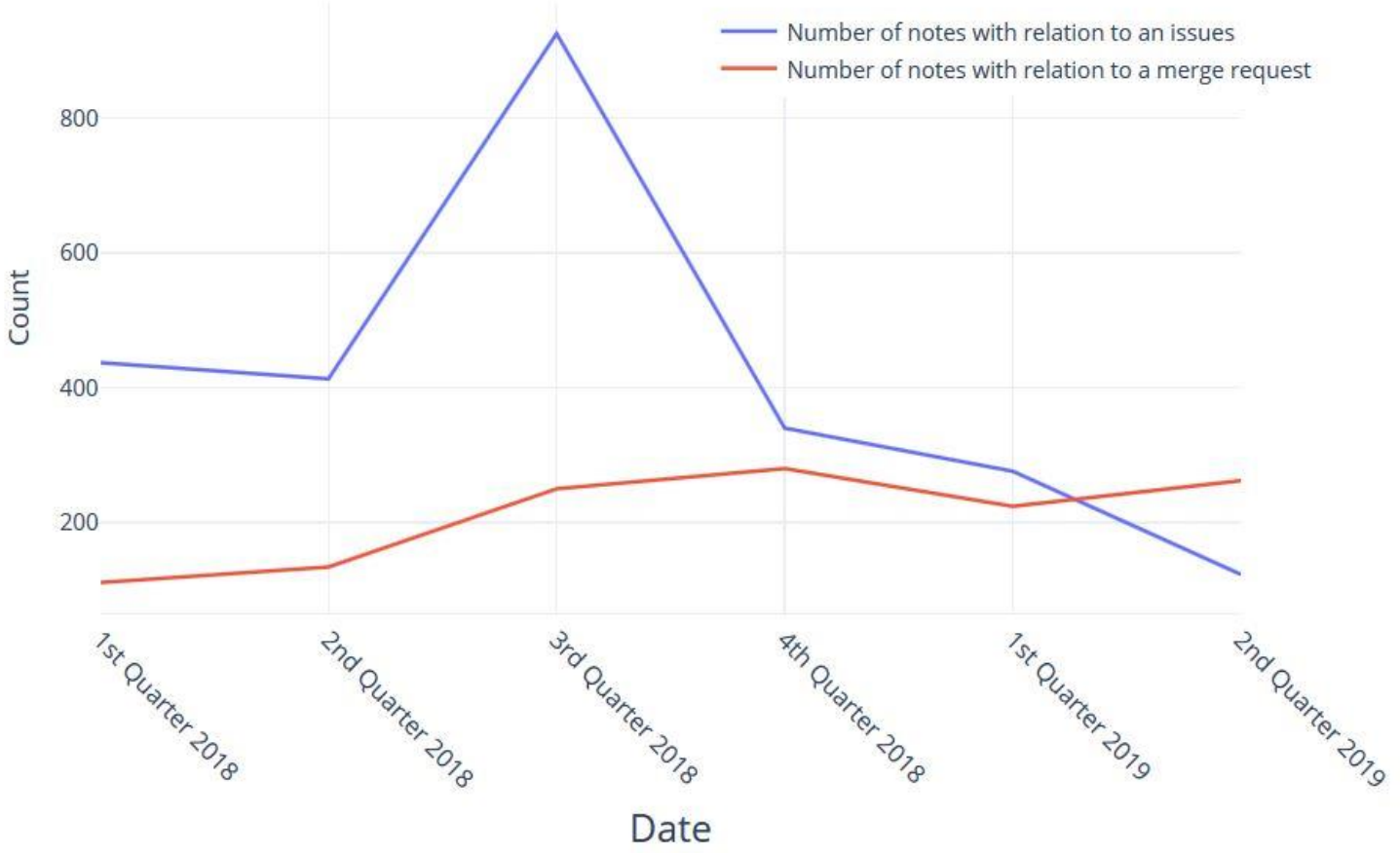
Results Issue Tracker

```
MATCH (n:Issue)
WHERE
    '2018-01-01' <= n.created_at <= '2018-03-31'
WITH*
WHERE
    n.weight  IS NOT NULL
RETURN count(*)
```

Results Communication

```
MATCH (n:MergeRequest)-[:HAS_NOTE]->(o:Note)
WHERE
    '2018-01-01' <= n.created_at <= '2018-03-31'
RETURN count(o)
```

```
MATCH (n:Issue)-[:HAS_NOTE]->(o:Note)
WHERE
    '2018-01-01' <= n.created_at <= '2018-03-31'
RETURN count(o)
```



Summary

- More regular and direct communication due to process
- Better introspection into different parts of the software for each team member
- General acceptance of the process within the team

And now?

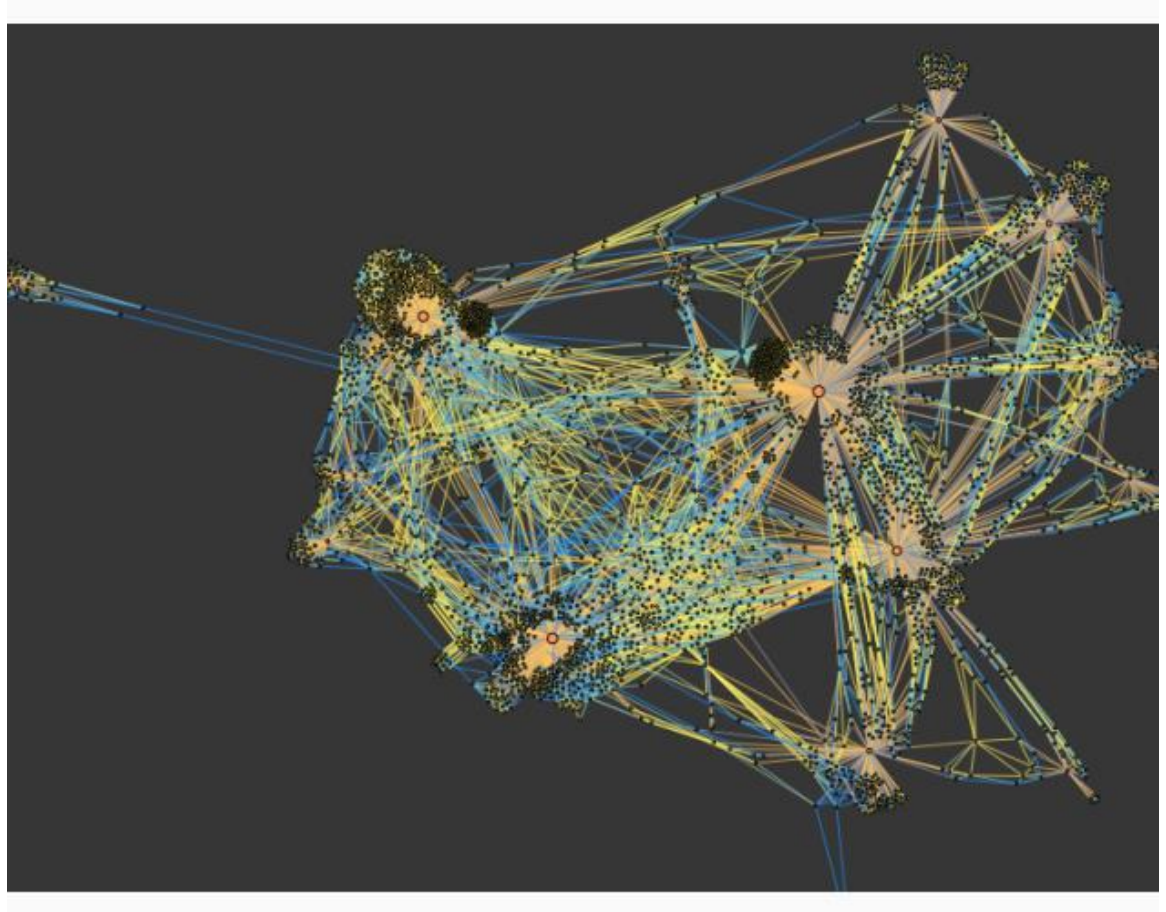
Future Work

- Utilize metrics within CI build to provide developer feedback
- Find methods to raise the acceptance of the software development process
- Extended the new concept of GitLab2PROV

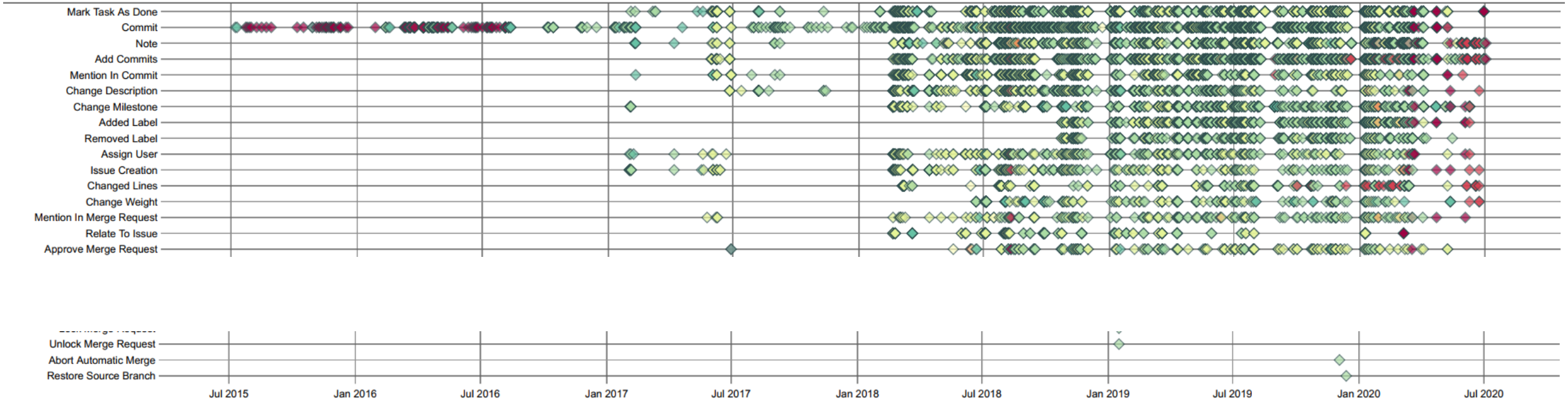
GitLab2Prov

GitLab2PROV generates retrospective provenance of GitLab projects

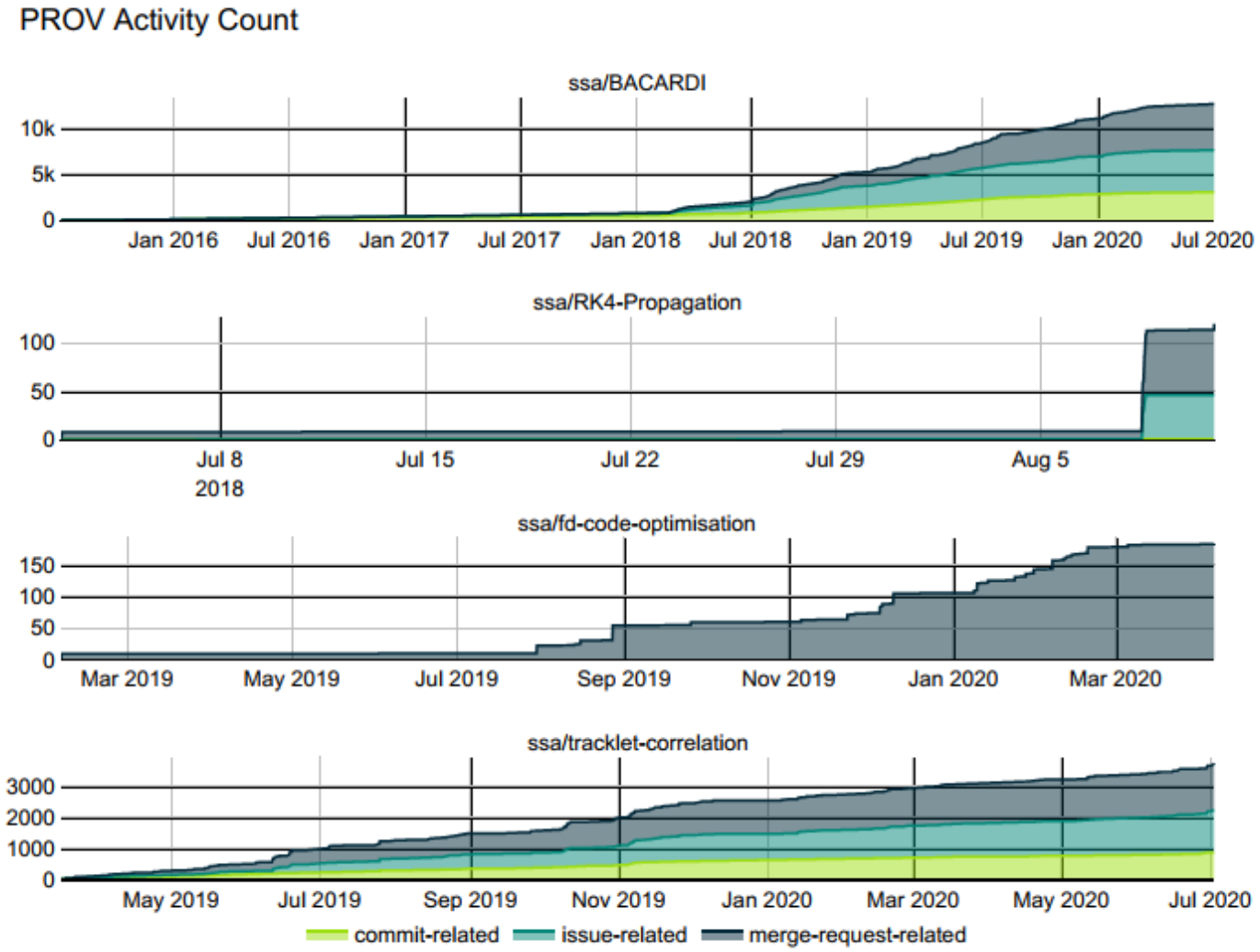
- Utilizes the W3C PROV standard
- Includes provenance models for
  - Git Commits
  - GitLab Issues
  - GitLab Merge Requests



GitLab2Prov



GitLab2Prov



Thanks!

@DLR_software

https://www.dlr.de/sc/
https://www.dlr.de/rb/