

Inner Source (Software Development)

Prof. Dr. Dirk Riehle

Friedrich-Alexander University Erlangen-Nürnberg

Inner Source Summit – Renningen – 2018-05-16

Professorship of Open Source Software

- Professor of Computer Science
 - For software engineering and open source software
 - At the computer science department of the engineering faculty
- Previously held research positions at ...
 - SAP Labs (Silicon Valley) leading the open source research group
 - UBS (Swiss Bank, Zurich) leading the software engineering group
- Previously worked in development at ...
 - Skyva Inc. (supply chain software, Boston) as software architect
 - Bayave GmbH (on-demand business software, Berlin) as CTO





Engagement Models

	Student Projects	Shared Projects	Exclusive Projects	Consulting (Limited)
Recruiting	Access to students, student jobs	N/A	Researcher	N/A
Out-sourcing	Non-exclusive rights to results	Benefitting from current project, non-exclusive license	Sponsorship of project, possibly exclusive IP rights	Consulting and projects, exclusive IP rights
Innovation	Non-exclusive rights to results	Benefitting from current project, non-exclusive license	Sponsorship of project, possibly exclusive IP rights	Consulting and projects, exclusive IP rights
Startups + ecosystem	Startup-Informatik: Improved Eco-system	Benefitting from current project, non-exclusive license	Sponsorship of project, possibly exclusive IP rights	Consulting for an improved ecosystem

The Other Definition of Open Source

“ Open source enables a **development method** for software that harnesses the power of **distributed peer review** and **transparency of process**. The promise of open source is **better quality, higher reliability, more flexibility, lower cost, and an end to predatory vendor lock-in.**” [O18]

Definition of Inner Source

“ Inner source refers to **the application of the open source approach** and benefits to developers **within the corporate environment** [...]” [D+02]

[D+02] Dinkelacker, J., Garg, P. K., Miller, R., & Nelson, D. (2002, May). Progressive open source. In Proceedings of the 24th International Conference on Software Engineering (pp. 177-184). ACM.

Open all artifacts!
Welcome visitors!

Reported Benefits of Inner Source [CR17]

- **Higher quality (code) components**
 - More users, more eyes, more perspectives: Less bugs faster
 - **Ultimately, higher product quality**
- **Improved knowledge sharing**
 - Collaboration across org. boundaries: More weak network ties
 - **More innovation**
- **Higher employee satisfaction**
 - More self-determination, broader recognition
 - **More attractive employer, better employee retention**
- **Faster, better, cheaper software development**

[CR17] Capraro, M., & Riehle, D. (2017). Inner source definition, benefits, and challenges. ACM Computing Surveys (CSUR), 49(4), 67.

Example Practitioners of Inner Source



Adobe

NETFLIX

facebook



Microsoft



HUAWEI

NOKIA



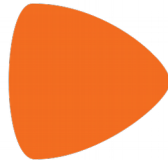
PHILIPS

Google



BOSCH

Continental



zalando

PayPal

SIEMENS

Collaboration in Open Source [R+09]

- **Traditional Work**

- Hierarchical
 - Closed and hidden silos
 - Assigned to project
- Status-oriented
 - Public and private discussions
 - Hierarchical status decides
- Assigned tasks
 - Prescribed process
 - Prescribed jobs

- **Open Collaboration**

- Egalitarian
 - Open for contribution
 - Everyone can contribute
- Meritocratic
 - Public discussion process
 - Decisions based on merit
- Self-organizing
 - People find their own process
 - People find their best project

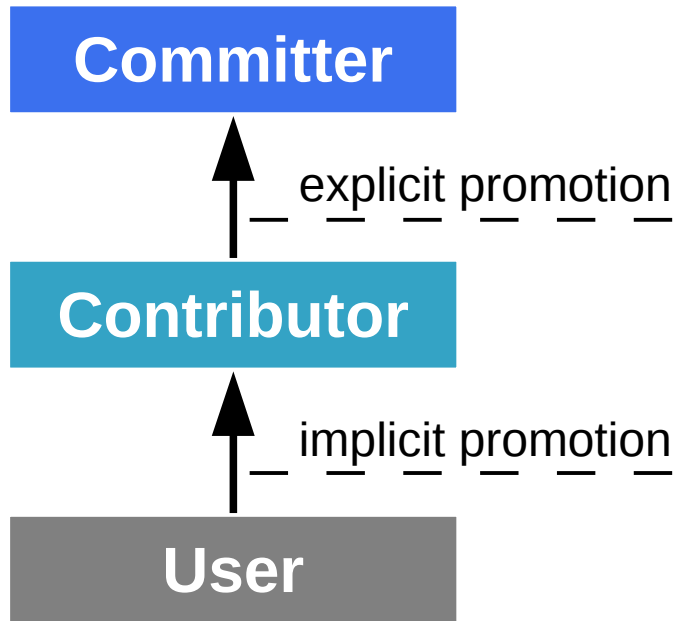
[R+09] Riehle, D., Ellenberger, J., Menahem, T., Mikhailovski, B., Natchetoi, Y., Naveh, B., & Odenwald, T. (2009). Open collaboration within corporations using software forges. IEEE software, 26(2), 52-58.

Communication in Open Source [R15]

- 1. Public**
- 2. Written**
- 3. Complete**
- 4. Archived**

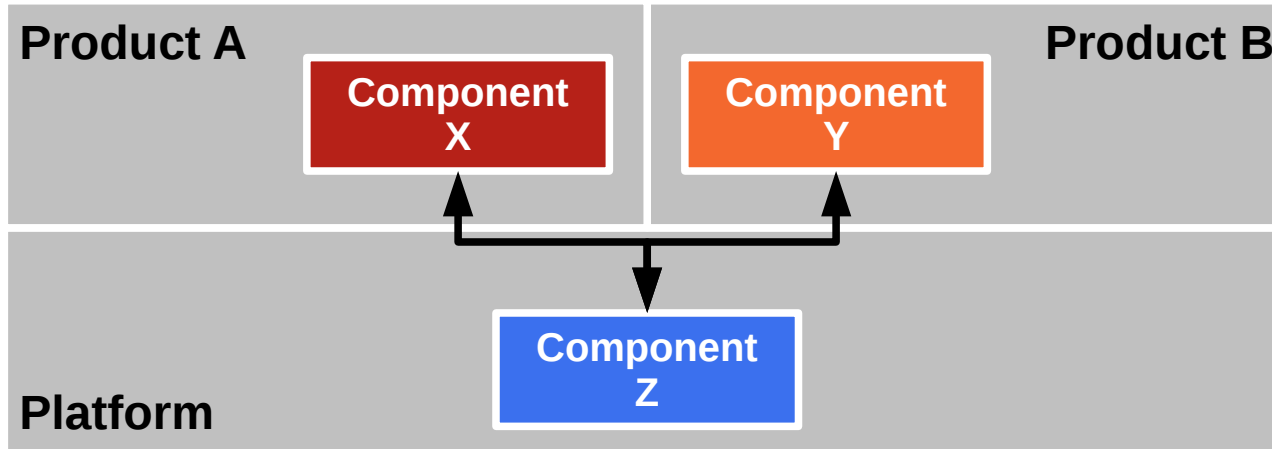
[R15] Riehle, D. (2015). The five stages of open source volunteering. In *Crowdsourcing* (pp. 25-38). Springer, Berlin, Heidelberg.

Quality Assurance in Open Source



- Formally: Has commit (write) rights
- Performs bulk of work; quality assurance
- Provides small features, bug fixes
- Submits patches (no commit rights)
- Knows and uses software
- Helps with comments, feedback

Example Scenarios of Inner Source Collaboration



- Bug fix
- Refactoring
- New component

Inner Source at SAP

- Inner source (then “firm-internal open source”)
 - Established to improve openness mindset, tear down silos
 - Established to simplify development using a software forge
- Set-up a software forge called SAP Forge
 - Set-up forge in 2006 (joint work of SAP Labs Silicon Valley and Israel)
 - Managed growth from 2006-2009 from 0 to 1500 registered developers
- Evolution of SAP Forge
 - Continued linear growth even after I left
 - Documented experiences in [R+09]

SAP Forge: Welcome - Mozilla Firefox

File Edit View History Bookmarks Tools Help del.icio.us

https://forge.sap.corp/

SAP FORGE

Software/Group Search

Logout My Account

Quick Jump To...

Home My Stuff Wanted

Welcome to SAP Forge!

SAP Forge is a collaboration system that was set up in order to encourage cooperative work within SAP. Why? Because the success of most projects nowadays depends on the quality of collaboration between people. This includes internal collaboration among project members and external collaboration with other groups.

SAP Forge provides tools for increased transparency and improved collaboration. It makes it easier for people to collaborate and accumulate shared knowledge, even when they are spread across multiple sites, countries, and time zones.

Specifically, SAP Forge provides:

- A single point of entry for all project-related information
- Message boards / discussion forums
- Issue trackers for: bugs, feature requests, patches, etc.
- Task management
- Mailing lists
- Website hosting
- Permanent archival for file releases
- Document management
- News & announcements
- Surveys
- Source Code Management (SCM) repositories like CVS and Subversion (Perforce integration is being planned)
- Full backup and administration

Participation

SAP Forge Statistics

Top Project Downloads

[More]

Highest Ranked Users

Explore GitHub



All



Showcases



Trending

Explore

Browse interesting projects solving all types of interesting problems.

[Sign up for free](#) to get started

Writing

Music

Programming languages

A list of programming languages that are actively developed on GitHub.

25 21

Game Engines

CSS preprocessors

Developers use CSS preprocessors to build CSS faster. You can browse som...

Projects that power GitHub for Windows

NoSQL databases

Projects that power

Properties of a Software Forge

- Projects are
 - Available in one place
 - Findable / searchable
 - Better documented
 - Available forever

The Mobile Retail Demo Project

- **Benefits to project**

- Found 18 volunteers (18/27)
- These volunteers
 - Were motivated
 - Had broad expertise
 - Improved requirements
 - Brought broad support
 - Helped with testing
 - Increased project visibility
 - Aided display of significance

- **Finally, transfer of project to product division**

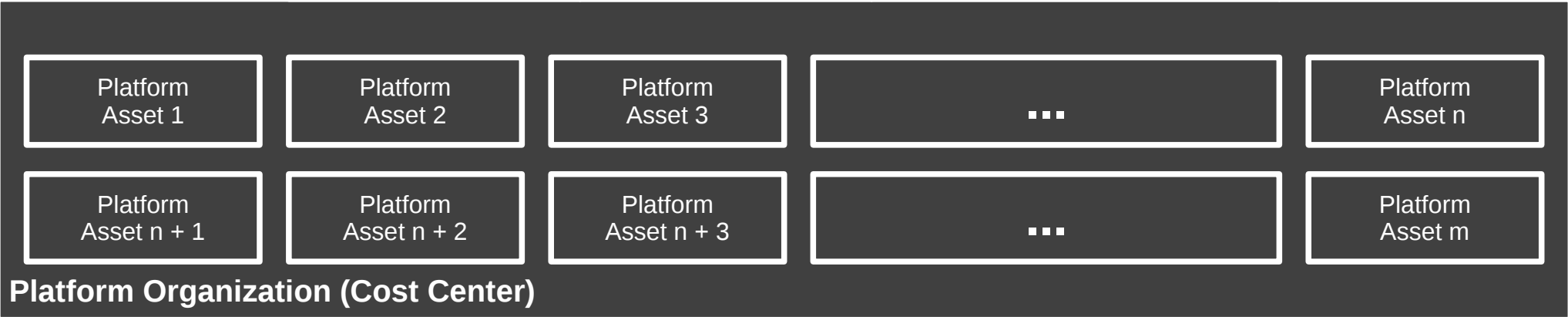
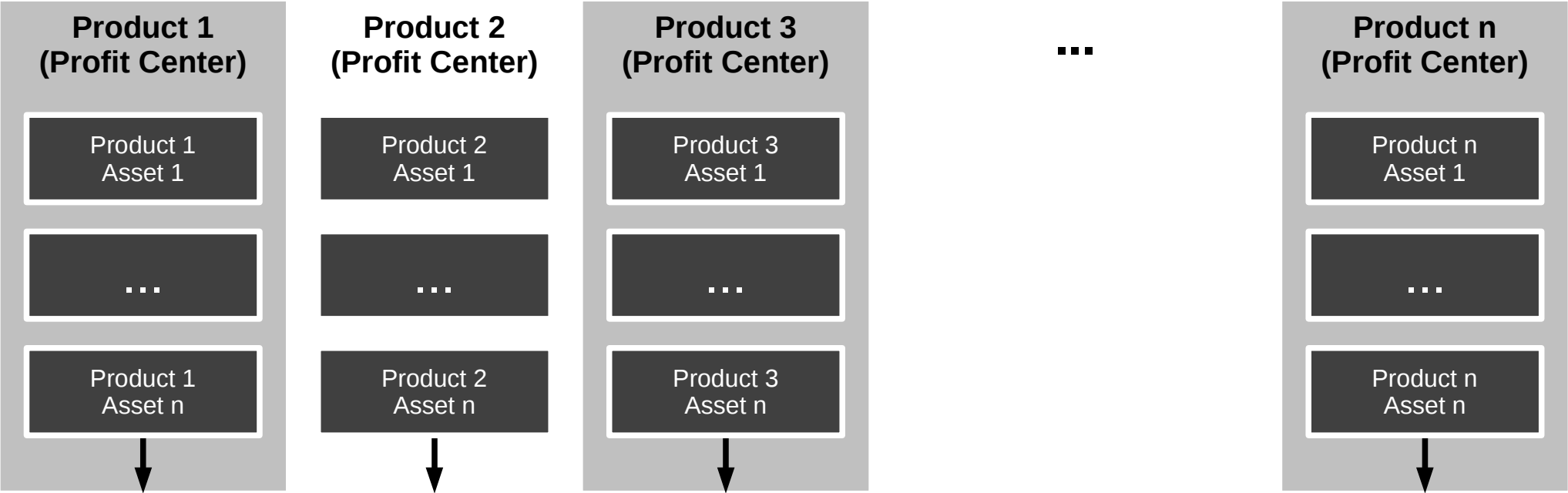
- **Benefits to company**

- Project uses common tools
- Tool set-up is understood
- Project can be found
- Project is archived

Final Reflections on SAP Forge / Inner Source at SAP

- Steady progress and interest
 - From 0 to 1500 developers in three years
- Developers often just wanted the tools
 - At the click of a button rather than lots of forms
- Approach worked well for research-to-product transfer
 - SAP Forge gave research early exposure to products
- A tool alone cannot create cultural change
 - But can be very helpful to it

Product Line Organization



Key Facts about Cases

- Product lines
 - Organization
 - Platform = cost center
 - Products = profit centers
 - Domain
 - Business software
 - Healthcare software
 - Telco carrier software
 - Developers
 - More than 500 developers each
 - Culturally and socially homogeneous
 - All in one location, i.e. no GSD problems
- Companies
 - Large international firms
 - All older than 20 years
 - Diversified, many product lines

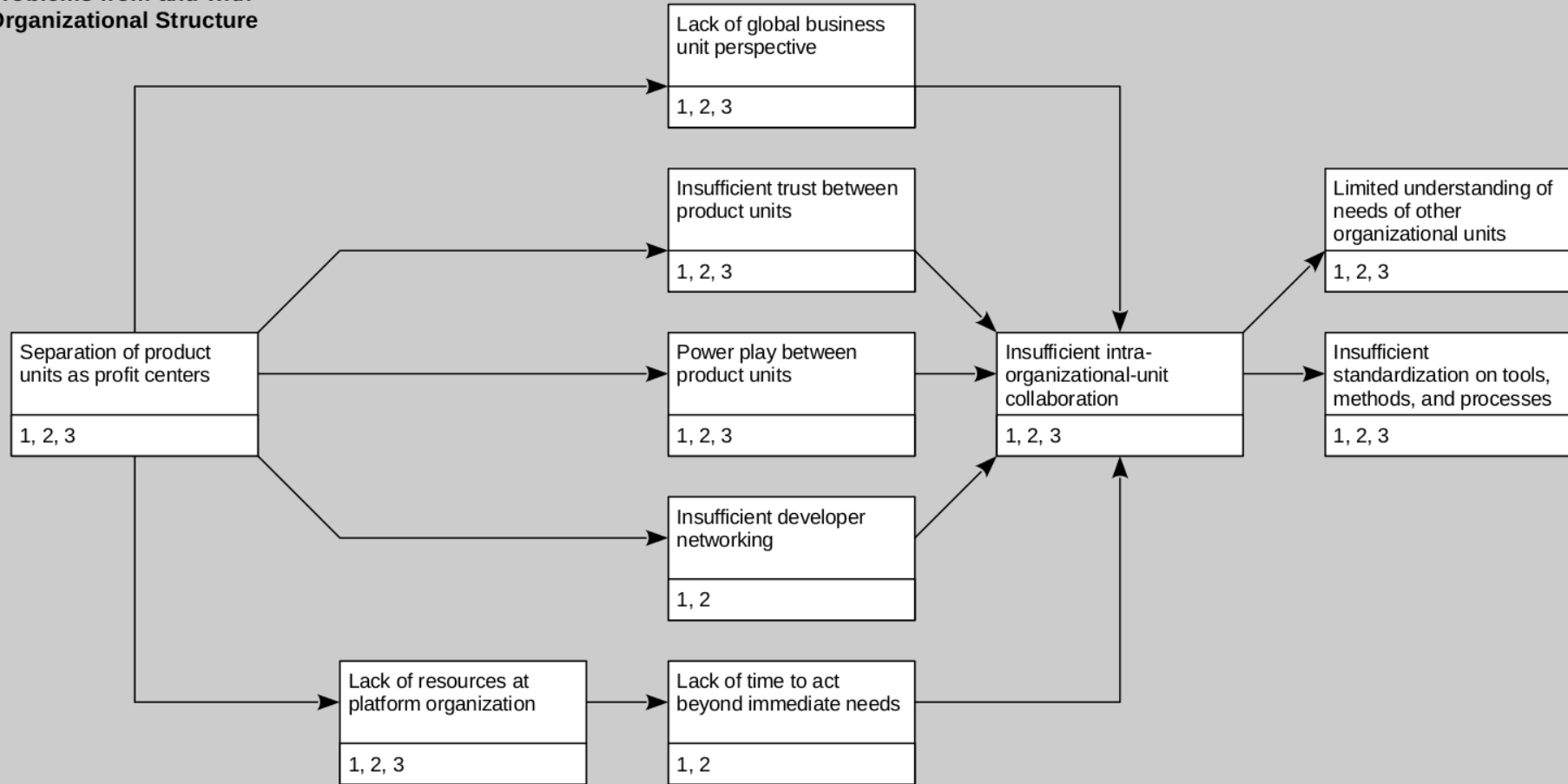
Case Access and Materials Gathered

	Case 1	Case 2	Case 3
Unit of analysis access	Direct access to all units of analysis	Direct access to all units of analysis	Mediated by sponsor
Subject access	Interview partners selected by consensus	Interview partners selected by consensus	Mediated by sponsor
Types of data collected	Collateral materials, interview notes	Collateral materials, interview notes	Collateral materials, interview notes
Researchers	Two researchers (one interviewer, one scribe)	Two researchers (one interviewer, one scribe)	Single researcher taking his own notes

Case	Year	No. Interviews	Workshops	Supplemental Materials
1	2012	11	5	Yes
2	2013	6	None	Yes
3	2013	4	3	Yes

Problems in PLE 1 / 2

Problems from and with Organizational Structure



Problems in PLE 2 / 2

Problems from and with Organizational Structure

Lack of resources at platform organization
1, 2, 3

Power play between product units
1, 2, 3

Limited understanding of needs of other organizational units
1, 2, 3

Insufficient intra-organizational-unit collaboration
1, 2, 3

Insufficient standardization on tools, methods, and processes
1, 2, 3

Problems with Domain Engineering Process

Delayed domain artifact realization
1, 2, 3

Poorly prioritized domain requirements
1, 2, 3

Unclear [to platform] reusable asset requirements posed by product units
1, 2, 3

Rework and wasted effort
1, 2, 3

Insufficient reusable asset quality
1, 2, 3

Insufficient reusable asset extraction process
1, 2, 3

Too much copy and paste
1, 2, 3

Undesired Effects

Delayed product delivery
1, 2, 3

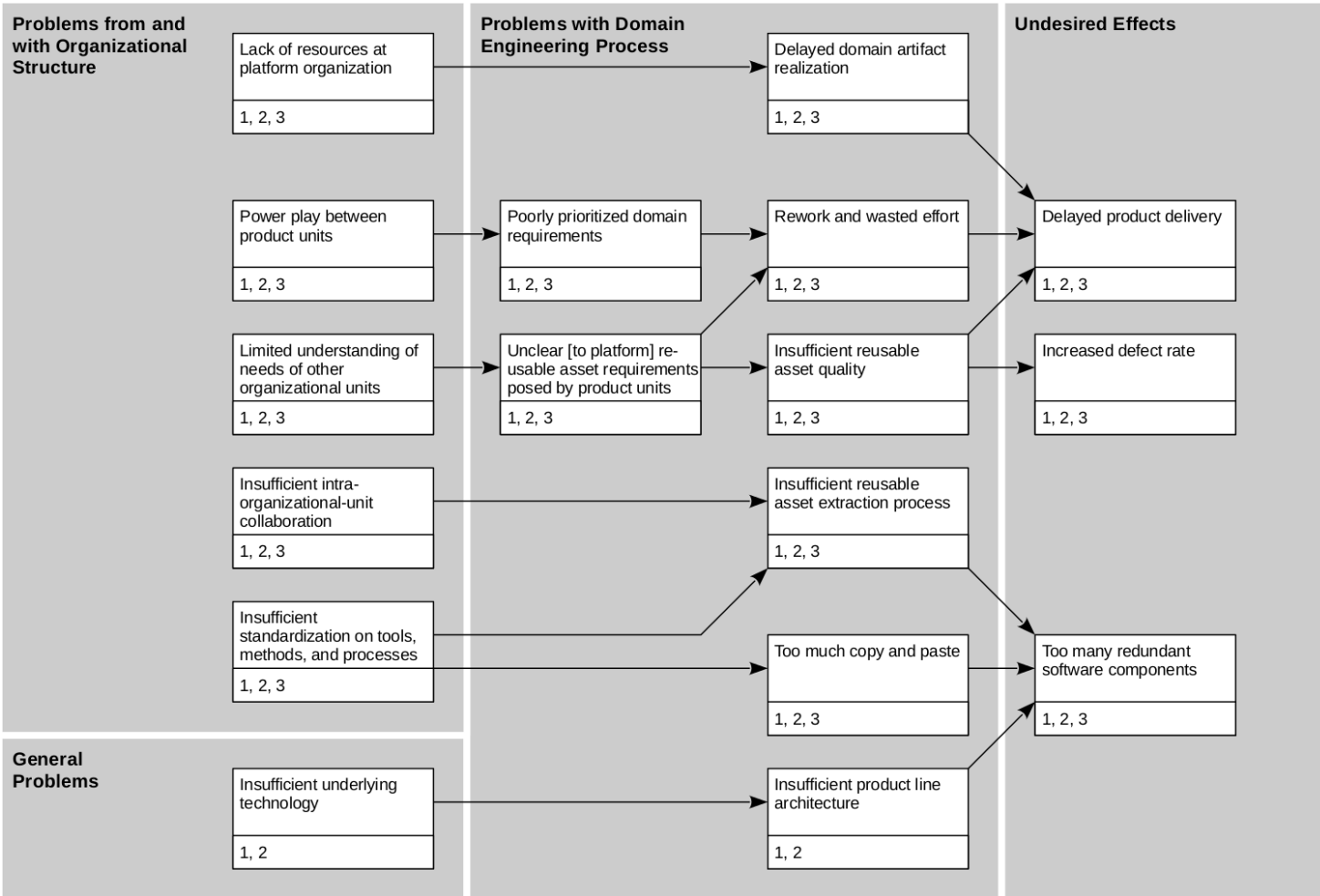
Increased defect rate
1, 2, 3

Too many redundant software components
1, 2, 3

General Problems

Insufficient underlying technology
1, 2

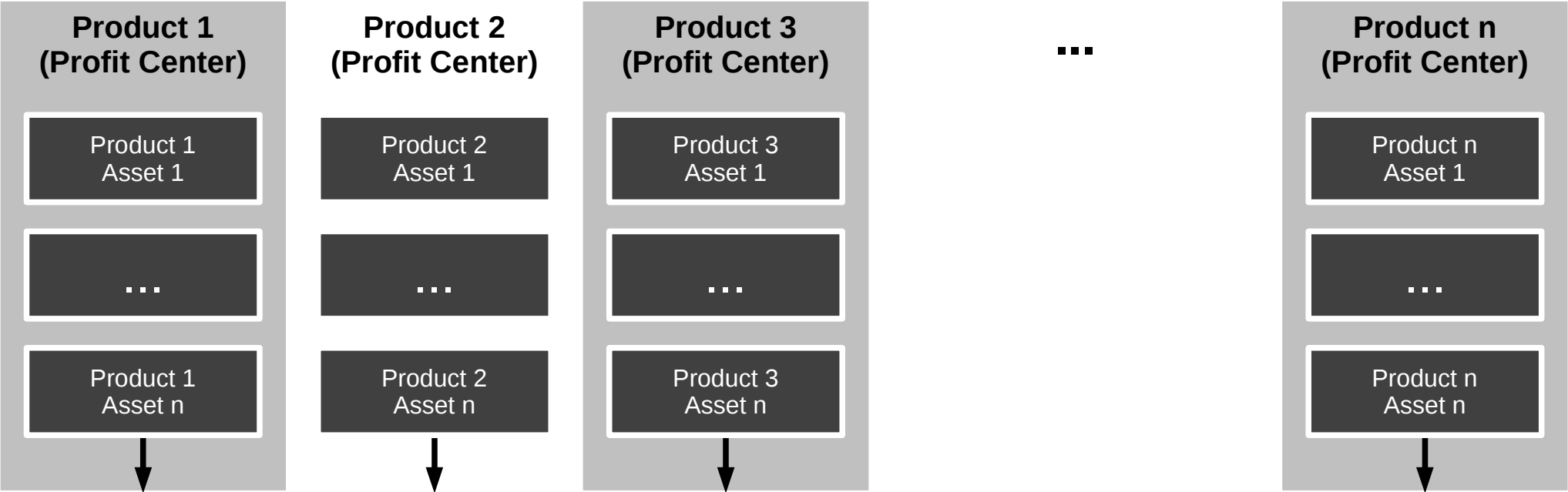
Insufficient product line architecture
1, 2



The separation of product units as profit centers from a platform organization as a cost center, leads to delayed deliveries, increased defect rate, and redundant software components.

[R+16] Riehle, D., Capraro, M., Kips, D., & Horn, L. (2016, December). Inner Source in Platform-Based Product Engineering. IEEE Transactions on Software Engineering vol. 42, no. 12, pp. 1162-1177.

Product Line Organization



Platform Organization (Cost Center)



Problems with Inner Source 1 / 2

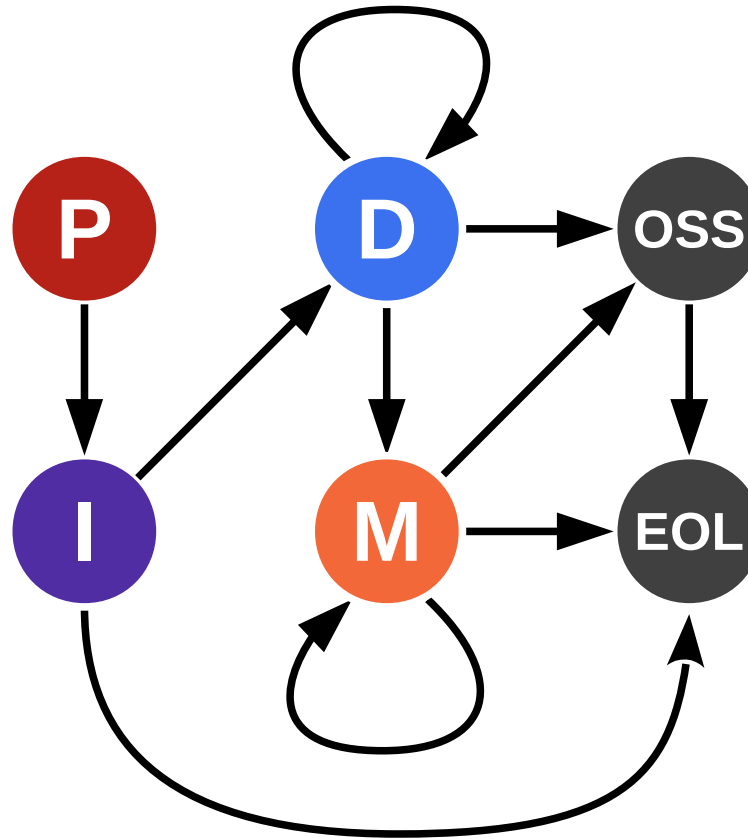
- **(Some) middle managers**
 - **Fear transparency and loss of control**
 - **Fear not meeting performance goals**
- “Most managers dislike showing their planning documents widely; it might open them up for critique.” Manager (product unit)
- “Allowing developers to contribute to inner source projects may feel like losing control to some managers.” Developer (platform)
- “Managers may disallow contribution to inner source if they feel their own product is not benefiting enough.” Developer (platform)

Problems with Inner Source 2 / 2

- **(Some) software developers**
 - **Dislike performing quasi-public work**
 - **Fear follow-on and maintenance work**
- “Inner source leads to [public] mistakes, and [some] developers fear mistakes because they lead to reputation loss among colleagues.” Manager (platform)
- “Most developers hate maintenance of important components because it makes them responsible for fixing high-priority bugs; this creates too much stress.” Developer (product unit)

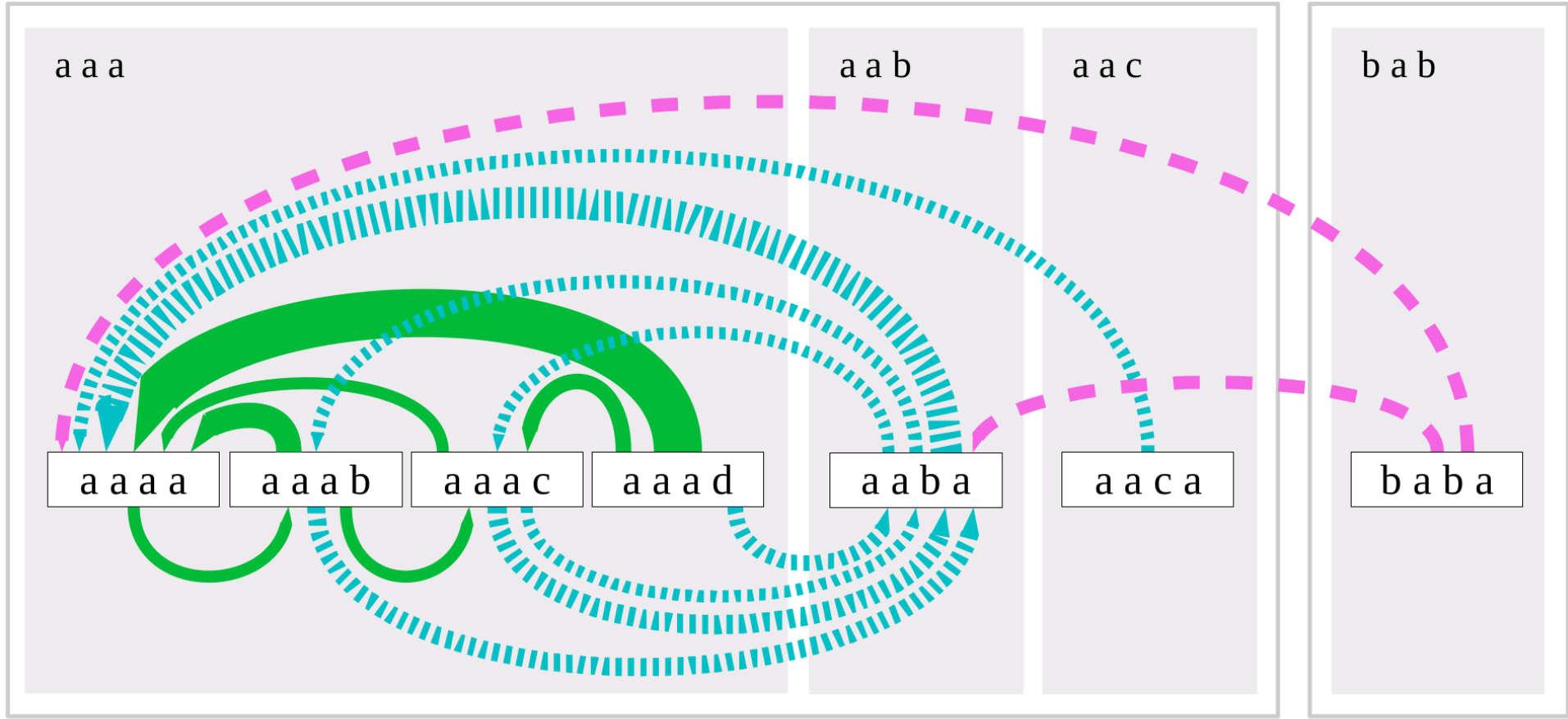
Establishing Successful Inner Source

- Inner source tooling
 - Software forge
 - Peer review
 - ...
- Inner source projects
 - Open collaboration
 - Open communication
 - ...
- Inner source programs
 - Cultural change
 - Managerial incentives
 - ...



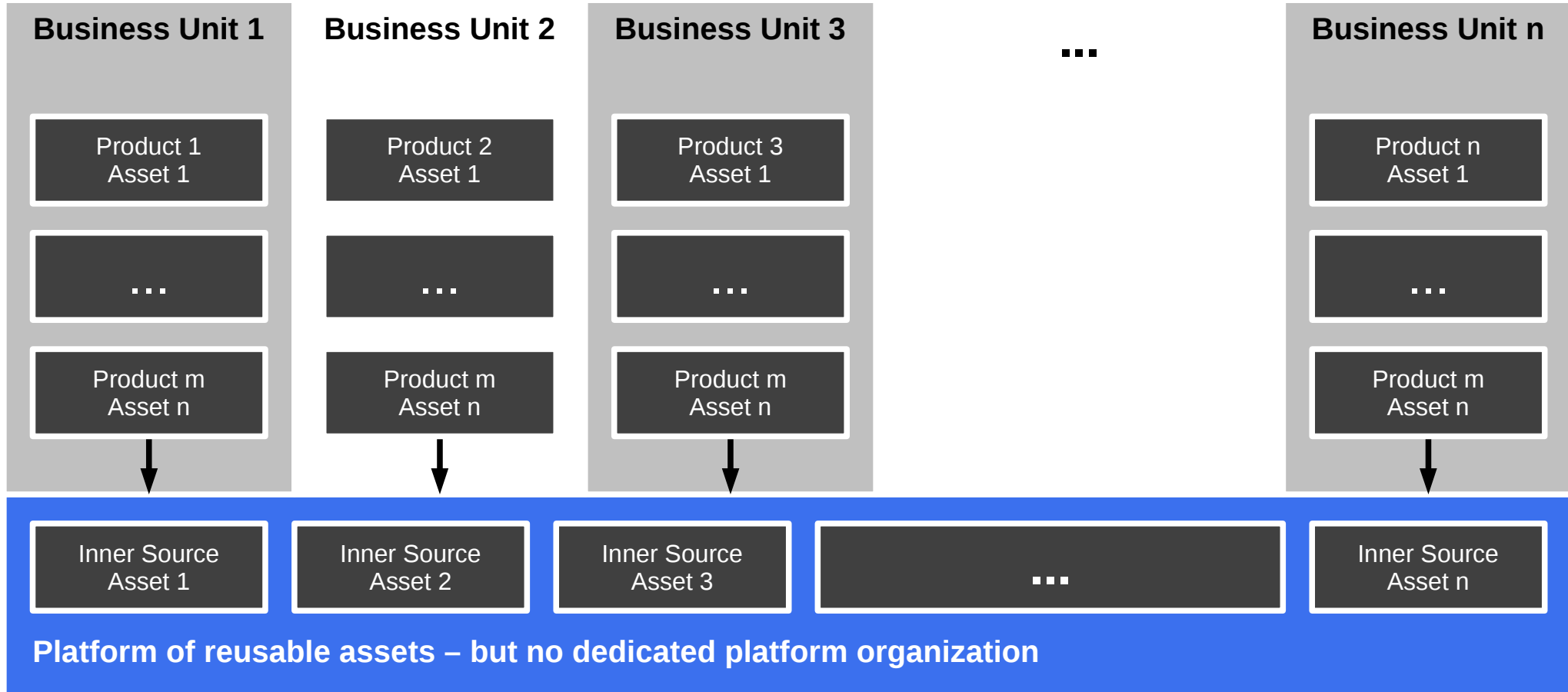
- P (roposal)
- I (ncubation)
- D (evelopment)
- M (aintenance)
- OSS
- EOL

Five Years Later, Revisiting One Case ... [C+18]



[C+18] Capraro, M., Dorner, M., & Riehle, D. (2018). The Patch-Flow Method for Measuring Inner Source Collaboration.

Enterprise



Inner Source Best Practices

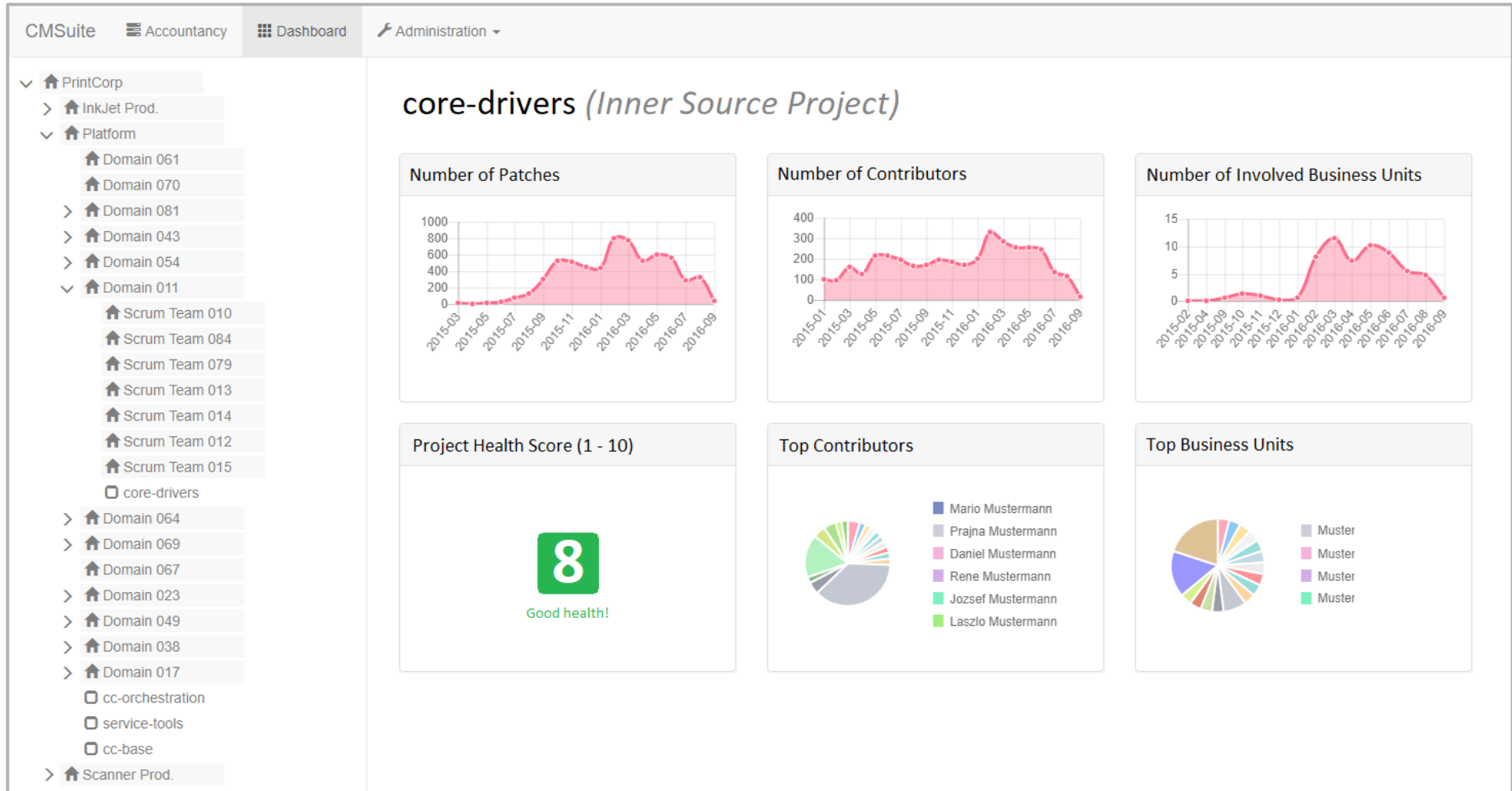
Inner Source Culture

- Open Collaboration
 - Establish principles of open collaboration
 - Make all artifacts available for reading
 - Decide based on the merits of the argument
 - Define and follow your own processes
- Open Communication
 - Establish principles of open communication
 - Communicate publicly
 - Communicate in writing
 - Communicate completely
 - Archive all communication

Inner Source Program

- Establish inner source program
- Inner source piloting
 - Start with an inner source pilot project
 - Choose a serious but not mission-critical pilot project
- Inner source program board
 - Establish an inner source program board
 - Define program board membership rules
- Inner source program governance
 - Define inner source program objectives
 - Define inner source program scope
 - Define inner source engagement rules
 - Incentivize managers to contribute to inner source
- Intellectual property management
 - Define intellectual property rules
 - Define set of acceptable inner source licenses
 - Define an inner source license
 - Include default inner source license
 - Ensure contributor can contribute legally
 - Ensure contributor agrees with project license
 - Require a contribution sign-off
 - Require contributor agreement
- Inner source project life-cycle
 - Define an inner source project life-cycle
 - Inner source project incubation
 - Establish an inner source project incubator
 - Review proposed inner source projects
 - Inner source project retirement
 - Establish an inner source project retirement home
- Inner source program office
 - Establish inner source program office

Management Accounting for Inner Source



Transfer Pricing

- How to value intellectual property on a fine-grain scale (commits)?
- How to optimize (legally) the resulting IP flow for tax purposes?

Typical Inner Source Work Packages 1 / 2

- Pilot project
 - Situation analysis
 - Process definition, tool selection
 - Pilot project selection
 - Project goal definition
 - Pilot project support
 - Lessons learned
- Support measures
 - Educational materials
 - Workshops
 - Literature

Typical Inner Source Work Packages 2 / 2

- Program support
 - Situation analysis
 - Program definition
 - Charter definition
 - Process definition
 - Tool selection
 - Program implementation
 - Educational measures
 - Workshops
 - Information exchange
 - Benchmarking
- Project consulting
 - Situation analysis
 - Project support
 - Lessons learned

Thank you! Questions?

dirk.riehle@fau.de – <http://osr.cs.fau.de>

dirk@riehle.org – <http://dirkriehle.com> – [@dirkriehle](#)