Inner Source Explained (for Transfer Pricing)

Prof. Dr. Dirk Riehle
Friedrich-Alexander-University Erlangen-Nürnberg

BMF Workshop 2021

Professorship for Open Source Software

- A professorship (organizational unit) at the computer science department of
    - The engineering faculty of
    - Friedrich-Alexander-University Erlangen-Nürnberg
- We perform research and teaching on
    - Software engineering and product development
    - For both computer science and information systems students
- Strong industry collaboration focus
    - In both basic and applied research projects
    - Including multiple resulting startups
- Led by Prof. Riehle
    - 15 years of industry experience, mostly software product development
    - In Switzerland (Zurich) and the U.S. (Boston and Silicon Valley)

Inner Source Software Development

The use of open source best practices (of collaboration) within a company

The goal is to break-down and collaborate across org. silo boundaries

Expected benefits of inner source are [1]

- More reusable components
- Higher code quality
- Better knowledge sharing
- Higher employee satisfaction

[1] See https://dirkriehle.com/2016/06/09/inner-source-in-platform-based-product-engineering/

The Practice of Inner Source [1]

Make all artifacts company-readable

Market your work, don’t be quiet

Welcome visitors and engage

Welcome contributions

[1] See https://dirkriehle.com/2018/07/05/ten-years-of-inner-source-case-studies-video/

Inner Source Components and Collaboration

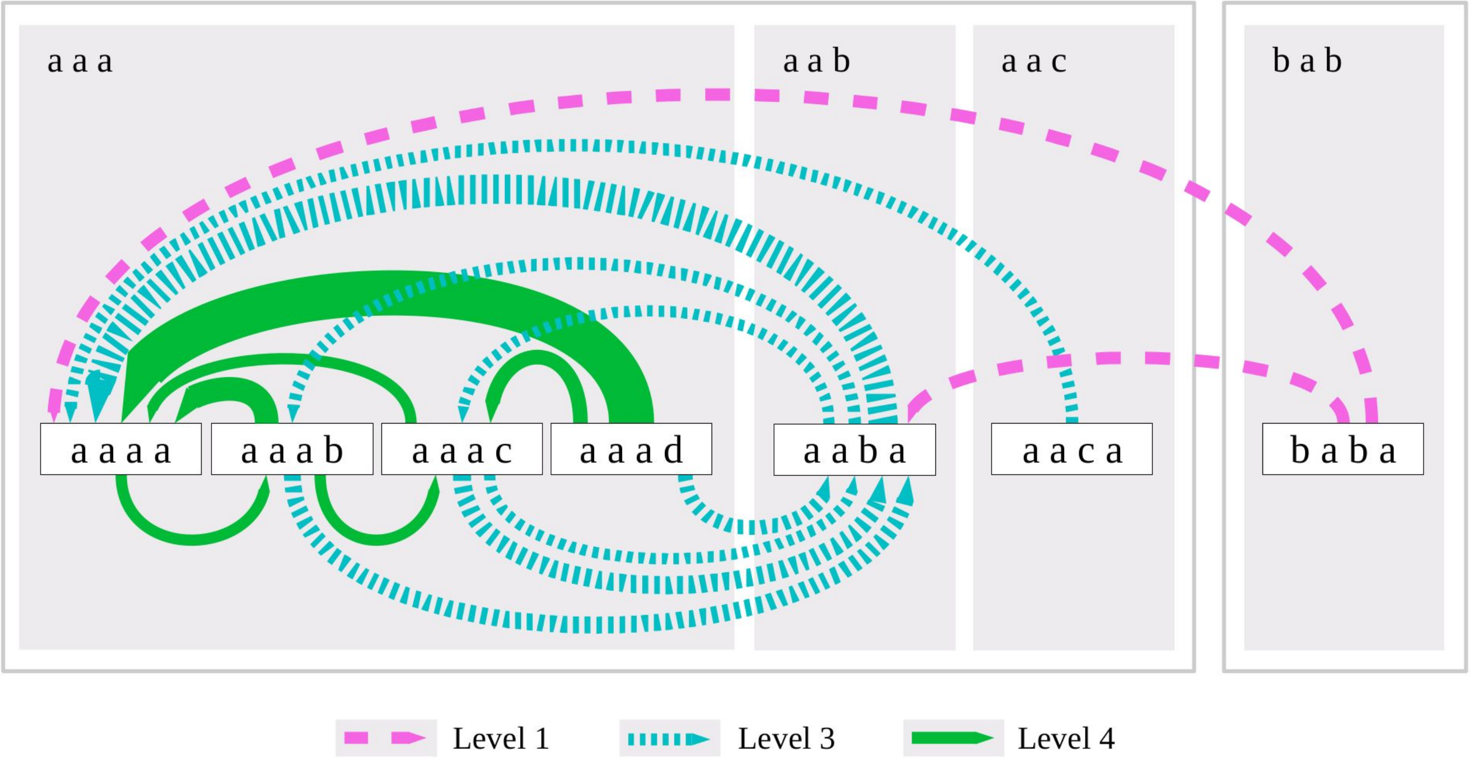
Inner source **components** (the artifact)

- Are not specific to a particular project or product
- Are (can be) used broadly across the organization

Inner source **collaboration** (the work)

- Is about the inner source component (not the using products)
    - But it is performed by the developers of the using projects or products
- Happens, by definition, across the organization

Example Code Flow Across Org. Levels [1]



[1] See https://dirkriehle.com/2018/04/09/the-patch-flow-method-for-measuring-inner-source-collaboration/

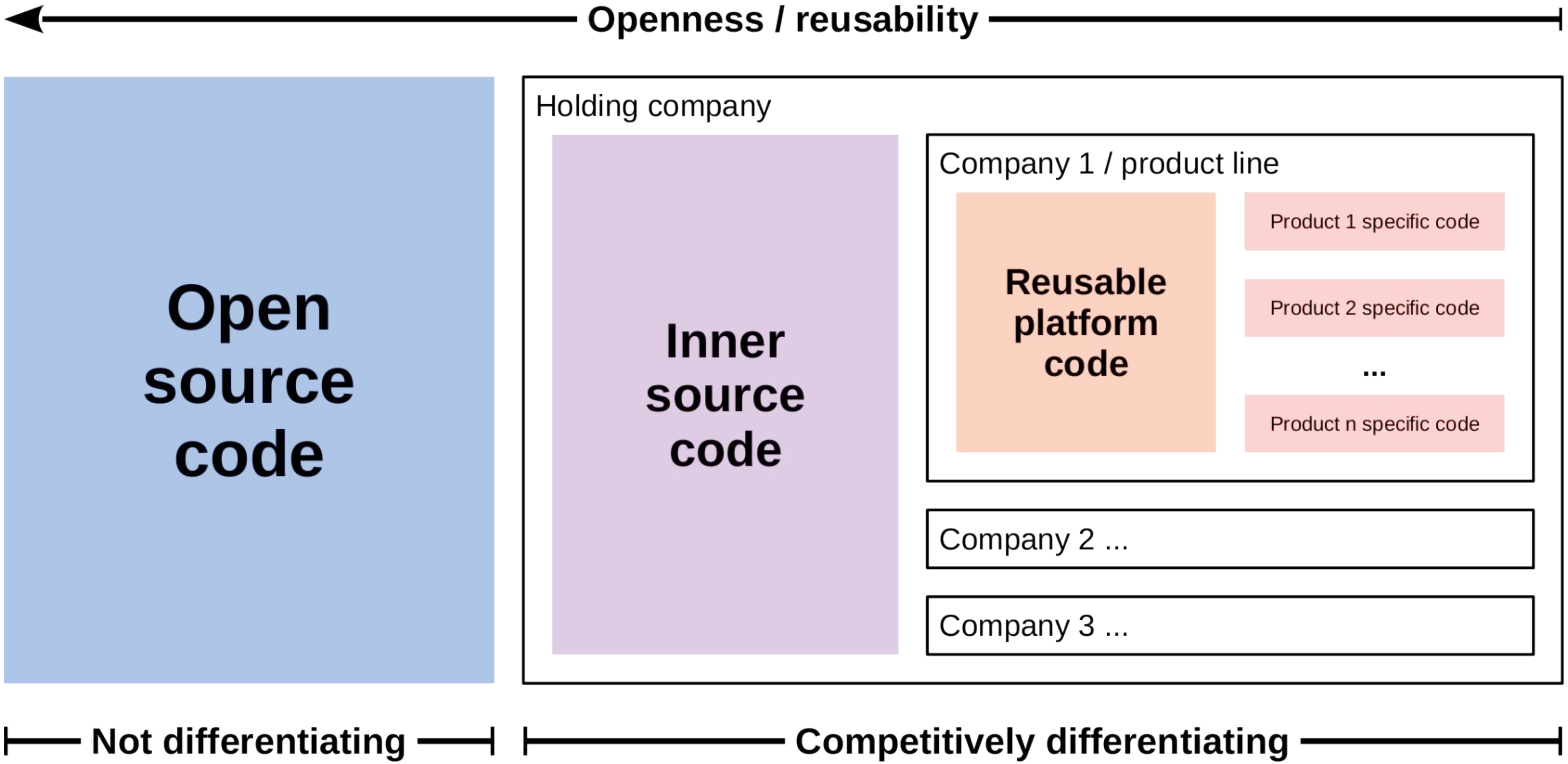
Inner Source from a Transaction Perspective [1]

Transactions for Component TopSecret

Date, Time	Author, Business Unit	Commit ID	Lines of code
2021-04-06, 09:43:16	Ac432@BU2	#4646546547475757	17
2021-04-06, 09:59:56	Ac432@BU2	#4646546547423221	9
2021-04-06, 11:23:26	Jw112@BU3	#3452222452235113	302
2021-04-06, 14:49:11	Ac432@BU2	#4522266214334287	32
2021-04-07, 08:53:33	Ho563@BU4	#2621445761222598	45
2021-04-07, 10:11:08	Ac432@BU2	#6762836246622111	89

[1] This is mock-up data, used for illustration purposes only

Economic Value of Inner Source Code



Inner Source Contributions as Service Provision

A compliance perspective, not an engineer's

- Service provision is asymmetric
  - One service provider and one or more clients
  - If more than one client, can still be reduced to 1-1 relationships
  - Relationship is set-up with explicit contracts
- Structural properties of work
  - **Economic risk is with client, not with provider**
  - Code flows into one direction only
  - Service provider does not use IP themselves

Software Development as a Service Organization

As software is eating the world, software

- Is in every product
  - https://dirkriehle.com/2018/07/17/contractions-and-expansions-in-organizing-software-development/
- Becomes a key differentiator
  - https://dirkriehle.com/2020/07/20/what-non-software-vendors-should-do-and-not-do-to-take-charge-of-their-software-future/
- Speeds up innovation and competitiveness
  - https://dirkriehle.com/2019/12/09/the-first-derivative-of-software-is-eating-the-world/

If software generates revenue, it should not be moved into a service organization

Inner Source as Collaboration for Mutual Benefit

This is what engineers want and how they look at it

- (If the company is large enough,) many use, some contribute
  - Many use, because they can: It is a marketplace with free offerings
  - Some contribute to manage their dependencies (like in open source)
- Intellectual property ownership
  - Distributed unless there are explicit agreements in place
  - Informally declared “inner source licenses” (e.g. MIT derived)
- There is a home / manager of the inner source project
  - Manager role is one of convenience, not for legal reasons
  - Forking is always possible

The Accounting Side of Inner Source

To manage software development

- Managerial accounting

To satisfy tax authorities

- Tax accounting, transfer pricing

Transfer Price Calculation Methods [1]

English	Deutsch	As service provision [Neumann]	As development collaboration
Comparable uncontrolled price method	Preisvergleichsmethode	N/A (IS ≠ OS)	N/A (no comparison possible)
Resale price method	Wiederverkaufspreis-methode	N/A (no resale)	N/A (no resale intended)
Cost plus method	Kostenaufschlagsmethode	**Applicable** (firm-internal service)	N/A (not an internal service)
Transactional net margin method	Nettomargenmethode	N/A (too far into the future)	**Applicable?** (dev. as joint venture?)
Transactional profit split method	Gewinnaufteilungsmethode	N/A (too far into the future)	**Applicable?** (dev. as joint venture?)

[1] See https://www.oecd.org/tax/transfer-pricing/oecd-transfer-pricing-guidelines-for-multinational-enterprises-and-tax-administrations-20769717.htm

Thank You! Any Questions?

Prof. Dr. Dirk Riehle, M.B.A.

dirk.riehle@fau.de, https://oss.cs.fau.de

dirk@riehle.org, https://dirkriehle.com, @dirkriehle