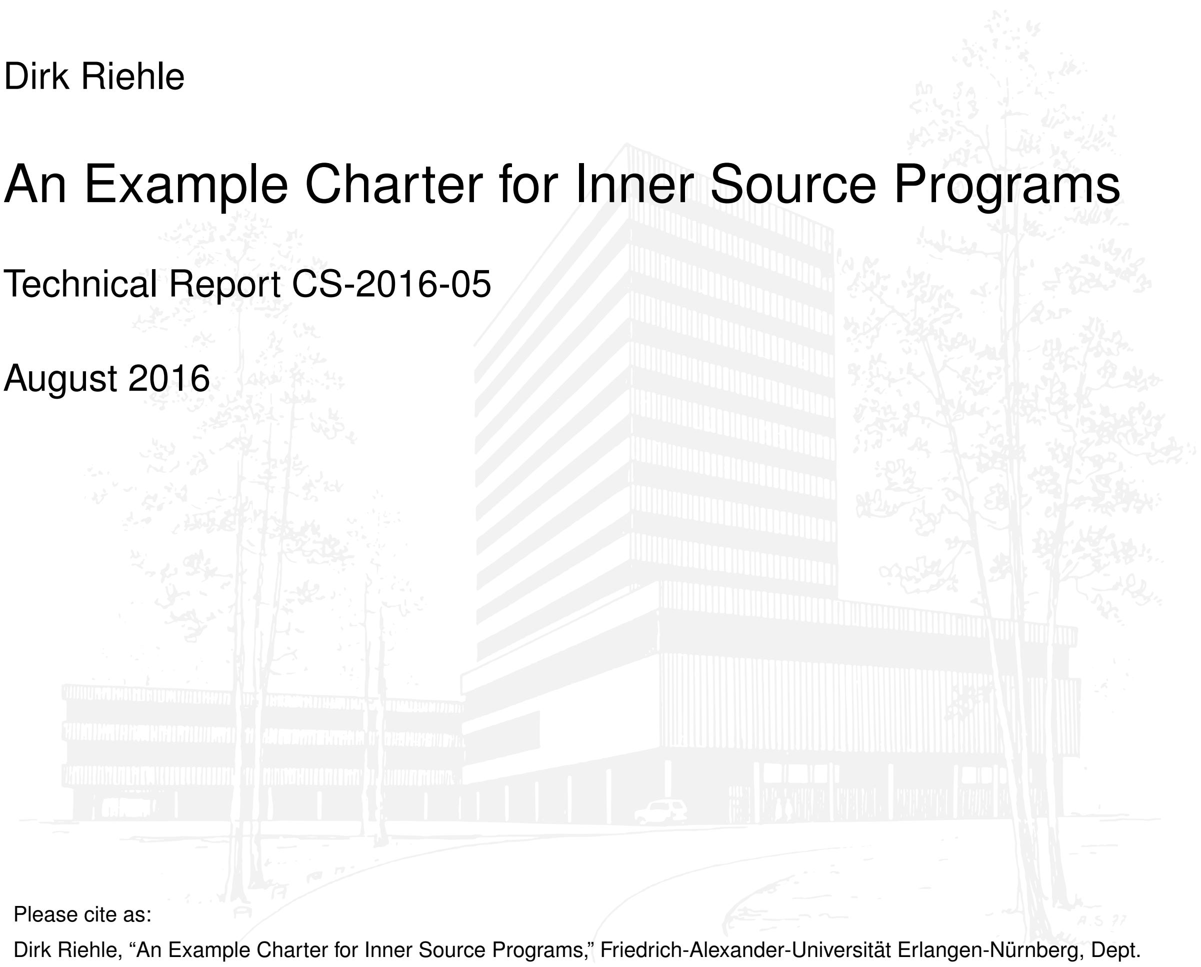
Department Informatik

Technical Reports / ISSN 2191-5008



Dirk Riehle

An Example Charter for Inner Source Programs

Technical Report CS-2016-05

August 2016

Friedrich-Alexander-Universität Erlangen-Nürnberg
Department Informatik

Martensstr. 3 · 91058 Erlangen · Germany

www.cs.fau.de

An Example Charter for Inner Source Programs

Prof. Dr. Dirk Riehle, http://osr.cs.fau.de, dirk.riehle@fau.de

Computer Science Department, Friedrich-Alexander-University Erlangen-Nürnberg

Abstract

Inner source software development is firm-internal software development that uses the principles of open source software development to collaborate across intra-organizational boundaries that would otherwise hinder any such collaboration. Inner source breaks down the barriers to collaboration across development silos by setting up an internal ecosystem of readily available software components. To get started with inner source, companies need to define their goals and then set up a governance structure for an inner source program and the projects within to reach those goals. This governance structure is often codified in the form of a charter document. This technical report presents an example charter for an inner source program. The goal is for companies to be able to copy and adjust this charter for their own needs. Towards this purpose, the charter leaves open the many decisions to be made, but outlines the options that any company needs to decide upon when establishing an inner source program.

Inhaltsverzeichnis

1. Introduction

Inner source software development is firm-internal software development that follows the three principles of open collaboration [2]. Software development is

- **egalitarian:** Everyone can contribute (rather than people are locked out)
- **meritocratic:** Decisions are merit-based (rather than status-based)
- **self-organizing:** People adjust processes to their needs (rather than the other way round)

As a consequence, software artifacts are not hidden but are made available to the whole company for reading, typically by using an internal software forge. Anyone may potentially contribute, and development ideally proceeds by drawing on a large and diverse contributor base across the company.

Practitioner reports show that inner source software development can benefit development productivity significantly. Among others, HP reports about improved software quality [1], IBM reports about more effective software reuse [4], and SAP reports about increased development speed [2].

Inner source software development can be quite different (and hence stressful) from a traditional perspective. For example, leaning on open source, communication should be open [3], that is

- **public (within the company):** Everyone can see all communication
- **written:** All communication takes place in written form, for example on mailing lists
- **complete:** Nothing that needs saying is left unsaid
- **archived:** All communication is archived for documentation and later review

To reap the expected benefits, companies have started to establish inner source programs to create and govern inner source projects. A first step towards a successful inner source program is to create an inner source charter that outlines the governance of the inner source program and the projects within.

This technical report presents an example inner source charter loosely based on the Apache Software Foundation[1] and the Eclipse Foundation governance documents. The example charter provides decision points, where companies have to make strategic decisions about their program.

This report is based on our experiences with companies seeking to establish inner source programs.

Our research group provides tools and methods to help companies understand their needs and goals for an inner source program. We help companies make the right choice at the decision points in the charter and when following up with more fine-grain process definitions.

1 See http://www.apache.org/foundation/governance/

A. Example Inner Source Charter

A.1 Introduction

This charter document defines the governance structure and processes for the inner source program of **COMPANY**.

A.1.1 Motivation

After plan-driven and agile software development, inner source software development promises faster and cheaper software development of higher quality software components [1] [2] [4]. Inner source is particularly good at fostering new innovative components.

Our **COMPANY** faces a competitive situation in which software is a **KEY_DIFFERENTIATOR | OTHER REASON** to win in the marketplace.

MORE_BUSINESS_MOTIVATION.

For this reason, we are adopting an inner source program and this is its charter.

A.1.2 Inner source

Inner source is firm-internal software development that follows the three principles of open collaboration [1]. Inner source software development is

- **egalitarian:** Everyone can contribute (rather than people are locked out)
- **meritocratic:** Decisions are merit-based (rather than status-based)
- **self-organizing:** People adjust processes to their needs (rather than the other way)

Inner source utilizes and judiciously applies best practices of open source software development. No open source software is being developed, however. Rather the result of inner source software development are traditional closed source software components. The only change is the way how the software is being developed.

A.1.3 Program history

This program was initiated by **COMPANY_BOARD | SOMEONE_ELSE**.

The **OFFICE_OF_CTO | SOMEONE_ELSE** was tasked with setting up the program.

A.2 General Principles

A.2.1 Collaboration

A.2.1.1 Project access

Inner source projects, the program, and their artifacts should be firm-readable by default.

Interested parties can find all artifacts at **INNER_SOURCE_FORGE | OTHER_ACCESS_PATH**.

A.2.1.2 Decision making

Decisions should be based on the merits of the argument and not on a person's status.

Inner source requires consensus building for decision making. This applies to projects as well as program governance.

Before making any important decision, like making a software design change or implementing a major feature, a proposer needs to build consensus. He or she does so by first gather opinions from the other involved people. Officially, they ask for a vote on a proposal. Traditionally,

- +1 indicates agreement,
- 0 indicates neutrality / abstainment, and
- -1 indicates disagreement on the proposal.

If there are only +1 or 0s, the proposer can go ahead with the proposed decision.

If there is at least one -1, votes should not be counted, but the disagreement should be viewed as an initial veto and a discussion should be started as to how to reconcile the different perspectives.

Votes should only be counted and decisions should only be made based on a simple majority if everything else failed. Such a situation may indicate a serious problem at hand.

A.2.1.3 Project processes

Project communities decide on their own processes.

An inner source project defines its own engineering processes. These processes should be designed with an eye towards integration into the existing processes of the participating business units.

A.2.2 Communication

Communication in inner source projects and throughout the program should be open.

Open communication is communication that is public (within the company), written, complete, and archived [3]. The goal is to include anyone who cares in the communication and decision making processes and to document those.

The most straightforward way of achieving open communication is to set up archived mailing lists, to which users can subscribe and from which they can unsubscribe, and to follow best practices of effective email communication.

At **COMPANY**, **LEGAL_CONSTRAINTS | CORPORATE_GOVERNANCE** prevent the complete capture of project-related communication. Also, sometimes in-person meetings are more effective than lengthy email discussions. In situations of meetings and other forms of ephemeral communication, a meeting documenter needs to be named. The documenter's job then is to provide a comprehensive summary of the meeting results in written form to the mailing lists or other archives used for project documentation.

A.3 Inner Source Projects

An inner source project develops one or more inner source components; it comprises both the community and the artifacts.

A.3.1 Project communities

A.3.1.1 Project roles

A project community comprises people, who may play one or more of the following roles:

- **Guest.** A guest is a visitor to the project website. Everyone inside the company can be a guest, because the project can be read by anyone within the company.
- **User.** A user is a guest to the project who is actually using the software in some form. A user may not have to declare that they are using the software.
- **Contributor.** A contributor is some who has made a contribution to the project. Contributions range from simple comments all the way to software code.
- **Committer.** A committer is a contributor who has been given commit rights (see committer elections) to some or all of the project artifacts.

In addition, there are project management committees (PMCs), which have members and a chair. For these, see below.

A.3.1.2 Committer elections

Any **CONTRIBUTOR_CAN_APPLY | COMMITTER_CAN_SUGGEST** a contributor for committer status. The existing set of committers vote and in case of at least one positive vote and no veto, the status change is accepted.

A.3.2 Project management committees

Large inner source projects (more than one component) **SHOULD | MUST** be governed by a project management committee (PMC). The PMC has ultimate decision making authority over the project.

A.3.2.1 PMC chair

Each PMC is led by a PMC chair. The **PMC_CHAIR_IS_APPOINTED** by the board | **PMC_CHAIR_IS_ELECTED** by the PMC members; only PMC members can elect a PMC chair.

A.3.2.2 PMC membership

The PMC consists of committers of the project. As soon as someone has been elected a committer, they become eligible to become a PMC member. PMC members can drop their role at any time, but should do so responsibly.

A.3.2.3 PMC member elections

Any **COMMITTER_CAN_APPLY | PMC_MEMBER_CAN_SUGGEST** a committer to become a PMC member. The PMC votes and in case of at least one positive vote and no veto, the committer becomes a PMC member.

A.3.3 Processes and practices

A.3.3.1 Quality assurance

Committers receive and review any contribution made by a contributor, playing the role of quality assurer. If satisfied, a committer will enact the change ("commit" the change to the project).

If not satisfied, a committer should engage in a conversation with the contributor as to why the contribution is not right or how it needs to be changed to be accepted.

A contributor or anyone else who is not a committer has no right to require that their contribution be put into the project. They do have the right for a copy of the project (a fork in open source parlance).

A.3.3.2 Project roadmap

The PMC defines and manages the project roadmap, including all releases.

A.3.4 Intellectual property handling

All project artifacts are available to everyone for reading and use. Use is free and includes all relevant intellectual property rights, including, but not limited to copyright, patents, and trademarks.

NO_CONTRACTS_NEED_TO_BE_SIGNED_FOR_USING | However, before using, a usage contract needs to be signed to avoid future problems with **TRANSFER_PRICING | REVENUE_ALLOCATION | BUSINESS_UNIT_SALE | OTHER_PROBLEM**.

NO_CONTRACTS_NEED_TO_BE_SIGNED_FOR_CONTRIBUTING | However, before contributing, a contributor license agreement needs to be signed to avoid future problems with **TRANSFER_PRICING | REVENUE_ALLOCATION | BUSINESS_UNIT_SALE | OTHER_PROBLEM**.

A.4 Inner Source Program

The overall inner source program is governed by its program board. The program comprises an open-ended number of inner source projects. Each project is governed by a project management committee.

The inner source program office manages operations of the program.

A.4.1 Program board

The program board has ultimate decision making authority over the program.

The board decides on all non-project specific issues, including rules for board membership. The board can delegate decision making authority to any employee working within the inner source community.

The board also has authority over the projects. However, projects are expected to work autonomously and coordinate with each other through their project management committees (PMCs). Thus, the board should not intervene in the management of any project except for extenuating circumstances.

A.4.1.1 Executive director

The program board is led by an executive director (ED). The ED is appointed by the **INNER_SOURCE_PROGRAM_OFFICE | PROGRAM_BOARD | OTHER_AUTHORITY**.

A.4.1.2 Board membership

The program board strives for a fair and balanced representation of the program stakeholders. The board of directors comprises

- appointed members from the inner source program office
- appointed members from the business units involved in the program
- elected members from the wider inner source community

Business units should strive to create diversity of its appointed members, drawing on different functional roles like

- Business owners
- Product managers
- Engineering managers
- Software developers
- **OTHER_FUNCTION**

A.4.2 Program office

The inner source program office comprises the staff tasked with

- originally setting up the program, including its charter
- originally evangelizing inner source
- managing and operating technical infrastructure
- managing any other program resource

A.4.3 Processes and Practices

A.4.3.1 Project life-cycle management

Inner source projects have a life-cycle. They are proposed, may then be incubated, might reach maturity and be in wide use, and may ultimately be retired. The program board has to approve any change in life-cycle status (like going from proposed to being incubated).

A.4.3.2 Project proposal

The inner source program office manages a proposal process. Anyone within the company can feed an inner source project idea into the proposal process. The program board decides on whether the proposed project should be incubated.

It is advantageous, but not necessary, if the project proposal has

- at least one developer and
- a PMC member willing to coach the project

If the proposal does not offer this, the board needs to find these people before the project can be incubated.

A.4.3.3 Project incubation

A nascent inner source project must first go through a phase of incubation (the incubator) before it can become a recognized inner source project. The project has at least one developer and one coach.

The job of the coaching PMC member is to introduce the developers of the incubating inner source project to the inner source program and the way inner source projects work.

Once the program board considers the project self-sufficient and the set of committer to be sufficiently diverse (e.g. committers stem from at least three different business units), the project may graduate from the incubator and become a regular inner source project.

The purpose of the incubator is to increase the chances of a project becoming a successful inner source project and at the same time to protect the reputation of the inner source program by withholding the label “inner source project” from projects that are not ready yet.

If the project fails to graduate within **TWO_YEARS | OTHER_TIME_FRAME**, it is put to rest. Its developers can still keep working on it; however, the project is not considered an inner source project.

A.4.3.4 Project retirement

A project is considered inactive, if within the last **SIX_MONTHS | OTHER_TIME_FRAME** no activity took place and when, upon request by the program office, nobody steps forward to explain the situation.

Inactive projects are marked as retired. They may still be in use, so they can’t be deleted. The program office maintains the list of committers involved with the project and monitors the project for any user requests that go unanswered.

In case of unanswered requests, the program office tries to help the requester to get an answer from the last known list of committers. However, the label “retired” should discourage anyone from starting to use the software.

A.5 Program Operations

The inner source program office is financed by the **OFFICE_OF_CTO | OTHER_ORG_UNIT | MEMBERSHIP_FEES**. It uses the funds to finance, manage, and operate all relevant activities and resources.

A.6 References

[1] 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.

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

[3] Riehle, D. (2015). The Five Stages of Open Source Volunteering. In *Crowdsourcing*. Li, Wei; Huhns, Michael N.; Tsai, Wei-Tek; Wu, Wenjun (Editors). Springer-Verlag, 2015, 25–38. Republished from The Five Stages of Open Source Volunteering. Friedrich-Alexander-Universität Erlangen-Nürnberg, Dept. of Computer Science, Technical Report, CS-2014–01, March 2014. Erlangen, Germany, 2014.

[4] Vitharana, P., King, J., & Chapman, H. S. (2010). Impact of internal open source development on reuse: participatory reuse in action. Journal of Management Information Systems, 27(2), 277-304.