

From Proprietary to Open Source: Building a Network of Trust

Petri Sirkkala, Imed Hammouda and Timo Aaltonen
Tampere University of Technology
P.O. Box 553, FI-33101 Tampere, Finland
firstname.lastname@tut.fi

Abstract. When a corporation is about to release a product as open source a large network of trust must be built and maintained. Open source and commercial domains have radically different aspects of trust. Still, trust is vital in products survival in both settings. This paper focuses on building cognitive, or rational, trust in both commercial and open source domains. We set the view angle so that trust can be approached via the various relationships between the stakeholders involved in the community building process. Towards this goal, the paper focuses on the first steps of the process by proposing a set of best practices.

1 Introduction

Trust has different meanings in the commercial and open source domains. For instance, especially in a commercial context, companies build trust in their software products through quality, brand, and reputation. On the other hand open source communities rely on trust of merit. A skilled programmer enjoys high levels of trust based on her merits of coding and other forms of contributions. The setting in this paper is such that a company decides to publish a product as open source. The product has originally been developed as open source.

The question addressed in this paper is how to balance the different dimensions of trust in the commercial and open source worlds. In particular, we focus on the problem of building the necessary trust around a new open source community. We identify the entities involved and the trust bonds between them. The goal of the process is to ensure a vital new community around a newly opened product. Only the initial steps of the process are covered in this paper.

Trust is defined in the literature as “the extent to which a person is confident in, and willing to act on the basis of, the words, actions, and decisions of another” [3]. Trust implies a party’s willingness to accept vulnerability, but with an expectation or confidence that it can rely on the other party [1]. Any evidence of lack of dependability provides a rational basis for withholding trust [3]. These definitions

lead to our view angle of trust. This paper uses dependability of different dimensions to explain and build a network of trust.

2 From Dependability to Trust

When a corporation publishes a new product as open source, the product needs a community to survive. It is very important to build a network of trust between the entities involved in the community. Figure 1 illustrates the general setting when releasing a new product or platform as open source. The model has been developed based on our first experiences and results of an industrial case study taken from a Finnish context. The solid elements represent existing relationships and the dashed elements represent new links that should be built during the release process (these will be explained in detail in Section 3).

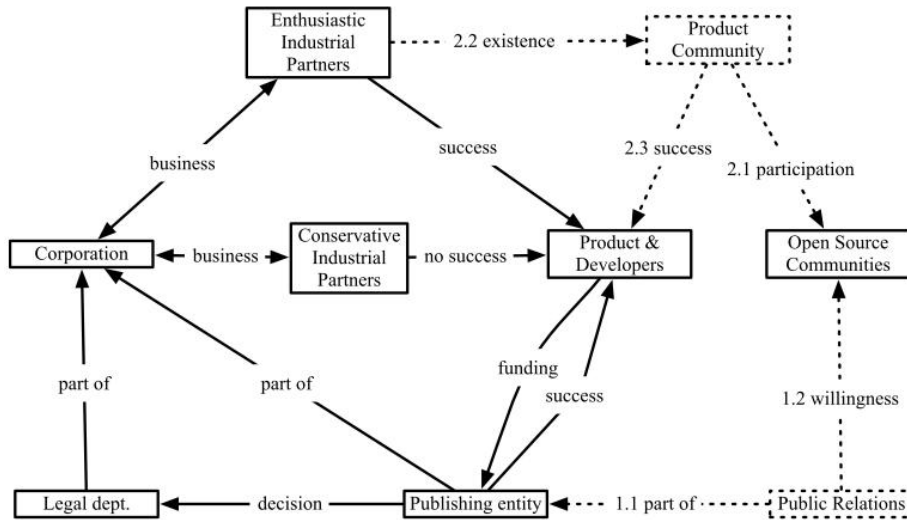


Figure 1. Dependability graph of a product to be open sourced

In Figure 1 the entities in boxes represent the stakeholders in the graph of dependability. The arrows between entities represent different dimensions of dependability. The Publishing entity is a *part of* the publishing corporation, for example a project team. The publishing entity depends on the *success* of the platform or product being published. It also depends on the *decisions* taken by the corporation’s legal department. Enthusiastic industrial partners, which are keen to see the product *succeed*, depend on *business* relationships with the releasing corporation. Conservative industrial partners, on the other hand, would prefer to keep the current *business* operations as usual and therefore would rather see the new product and its open source community *not to succeed*.

2.1 Dimensions of Dependability

Given the context described in Figure 1, one can identify the different ways two entities may depend on each other. While building trust the dimension of the dependability must be taken into account. The specific aspects or the issues of the dimension depend also on the entities involved in the relationship. The dimension of *business* could contain partnership, subcontracting acquisition or licensing each others technology. Dependability on *decision* with the legal department might be about issues of licensing, legal code checking, and about the availability of trademarks and names. The *success* dimension is roughly defined by money, popularity and quality aspects. The *no success* can generally be viewed as failure on any aspect of success. *Funding* is considered to be a dependability dimension where one party depends on the resources of the other. Being *part of* another, or contained, implies that one party is organizationally part of another. *Participation* can have the aspects of development, support and promotion. *Willingness* has the factors of being motivated by the other and perceiving the other in a positive way. *Existence* dimension can have the aspects of vitality, consciousness, skills and being organized.

2.2 Suffer and Joy

A dependability dimension is typically associated with a situation of satisfaction or dissatisfaction. In this paper we measure the satisfaction by *joy* and dissatisfaction by *suffer*. In the dependability graph any dependent reacts to the others' state, resulting in joy or suffer. Take for instance the dashed dependability of success numbered 2.3. The product community has dependability on the success of the product: The level of community's joy depends on the success rate of the product. In the meantime the community then needs to trust the product to succeed. To build such trust the publishing entity needs to take relevant actions, which are described in more detail in Section 3.

3 Building Trust

In the following we describe the steps required to build community's trust for a product released in the open source domain. The dependability dimensions we described earlier are to be taken into account. As a matter of fact, we focus on the dashed portion of Figure 1 (new relationships established). The first step, (1.1) in the figure, is to fill the public relations position. Such person needs to be in close cooperation with the product and developers team to ensure that the insight of the product is properly channeled to the community. Furthermore, the public relations person depends on the willingness of the open source communities (1.2). Therefore, she needs to motivate the community by making positive perception of the product and publishing entity. This requires transparency to make clear that the publishing entity has no hidden motives [2, p.164]. Any monetizing attempts need to be

carefully thought of in order to avoid bashing the community [2, p. 164]. The methods include sending positive tone messages to community members and being prompt in answering any queries. Furthermore, the public relations person should communicate the possible incentives to motivate contributors to join the community. The next step (2.1) is to have the open source community to join in and to actively participate in the product community. This yields to networking of developers which in turn strengthens the product community. This happens by migration of developers or by having developers that act as bridges between projects [4]. Participation can be endorsed by building a sufficient community infrastructure (e.g. for communication, bug reporting, downloads, etc.) and by defining clear processes to incubate projects and to promote developers to leading roles in the community. Existence in (2.2) can be strengthened by having a well defined role and form of participation in the community. Also the motives and the goals of the enthusiastic industrial partners should be well clear. Success of the product (2.3) can be endorsed by the corporation showing keen interest in the product and by allocating resources for community building. Other actions include organizing events, conferences, and competitions as well as writing publications and documentation around the product.

Support for existing trust is needed on solid areas of Figure 1. All dependencies need attention and work to hold. For example business relationships related to the released product needs to be maintained. Furthermore, conservative industrial partners should be encouraged to adapt for changes in the business and development modes of the product. Furthermore, existing bureaucracy (e.g. decisions taken by the legal department) and organization structures of the corporation should be adjusted not to conflict with the “release early, release often” principle of open source. Another important action is to maintain a stable funding channel for the core development of the product, at least during the first phases of the community building process.

3 Conclusions

Trust is essential to the vitality of any open source community. Building and maintaining trust is a major challenge when releasing a proprietary software product as open source. The process involves a wide spectrum of stakeholders with different forces and complex relationships. As trust relates to the dependability between the various stakeholders, the various dimensions of dependability have to be taken into account when building community trust. This paper presented first steps towards building a model of dependability-trust in the context of opening proprietary software. The model has to be further strengthened and evaluated against a number of industrial case studies.

References

- [1] Maria Antikainen, Timo Aaltonen, and Jaani Visnen. The Role of Trust in OSS Communities - Case Linux Kernel Community. In Joseph Feller, Brian Fitzgerald, Walt Scacchi, and Alberto Sillitti, editors, OSS, volume 234 of IFIP, pages 223–228. Springer, 2007.
- [2] Ron Goldman and Richard Gabriel. *Innovation Happens Elsewhere: How and Why a Company Should Participate in Open Source*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [3] Daniel J. McAllister. Affect- and Cognition-based Trust as Foundations for Interpersonal Cooperation in Organizations. *The Academy of Management Journal*, 38(1):24–59, 1995.
- [4] Michael Weiss, Gabriella Moroiu, and Ping Zhao. Evolution of Open Source Communities. In *Proceedings of the second international OSS conference*, pages 21–32, 2006.