Quantifying the Impact of OSS Adoption Risks with the help of *i** Models

Dolors Costal[1], Daniel Gross[2], Lidia Lopez[1],
Mirko Morandini[2], Alberto Siena[2], Angelo Susi[2]

[1] GESSI Research Group, Universitat Politcnica de Catalunya
Barcelona – Spain
{dolors,llopez}@essi.upc.edu
[2] Fondazione Bruno Kessler
I-38123, Trento – Italy
{gross,morandini,siena,susi}@fbk.eu

Abstract. Adopting Open Source Software (OSS) components in organisational settings requires evaluating the possible impact of adoption decisions on business goals. Measures available in OSS, capturing indicators such as the quality of open source code and the activeness of the developing community, can be used as a driver to assess various risks in component adoption. In this paper we illustrate how risk and impact models are used to relate measures obtained from the component under analysis to business goals in *i**-based OSS business strategy models.

Keywords: Risk Assessment, Open Source Software, Business Strategy

1 Introduction

During the development of complex software systems, the choice of adopting pre-existing software components can have a significant impact on the technical, organizational, as well as high level business objectives of the developing organisation. More and more, open source software (OSS) components are adopted by organisations in the development of commercial software systems, considering possible advantages in license cost, time to market, security, maintenance efforts, or other factors, depending on the domain. However, OSS adoption also carries various technical and business risks, due to the lack of contract partners, lack of guarantees for future component maintenance and support efforts, the distributed, open development process, and the heterogeneous development community. OSS component selection and maintenance thus claim for a thorough analysis of technical aspects of the components and their possible impact on the high level strategic objectives. The understanding and management of risks is a key factor for contributing to the success of the development project or to ratify its failure.

In this paper we use and extend the *i** modelling framework to support the evaluation of the possible impact of OSS component on the business objectives of a company. The framework was developed in context of the European FP7

project RISCOSS (www.riscoss.eu). It offers a risk model composed of a measurement framework for available OSS component data as well as a conceptual modelling language capable of representing risk events, indicators and situations. The risk model is linked to i^* models that capture and represent high-level business goals and related business model strategies, by means of an impact model. Propagating the available OSS measures and indicators and the organisational data in these models, we are able to quantify the impact to business objectives, a prerequisite for undertaking proper mitigation activities. The work is also inspired by various works [2, 1, 6, 3] that introduced the concepts of situations, risk events and goal analysis in the i^* framework to deal with problems in business intelligence, risk analysis and norms compliance.

We present the models in Section 2 and discuss their application in Section 3. We conclude with a summary of the current state of work and the short- and long term perspectives.

2 Modelling OSS Adoption Risks

To quantify and evaluate the impact of risky events on business objectives, we make use of a number of interrelated models and techniques. In particular, we need to model (i) the (possible) strategy of the adopter, expressed in terms of goals to be achieved and tasks to be performed; (ii) the ecosystem, naively intended as the inter-relation among a collection of actors; and the conditions, that make risks more or less likely to happen, or increase or reduce their significance. Business strategies are represented using i^* models, due to its support for modelling goals, actors and strategic dependencies. Risk conditions are modelled using an ad-hoc language, which uses in a new way concepts already available in existing literature.

Underlying the goal models we developed an OSS ontology. The consortium has chosen an existing ontology, OFLOSSC [4], to define a standard set of relevant concepts, relationships and terminology to describe a terminology for OSS adoption, related in particular to OSS development communities and socio-technical interactions between OSS community members.

Additionally, a Systematic Literature Review (SLR) was conducted by members of the consortium, on OSS adoption risks, available measures and mitigation activities. This SLR allowed us to gather knowledge used to build the initial taxonomy of risks and risk indicators for creating risk model instances [5].

2.1 Goal Models

The i^* model in the upper side of Figure 1 includes two main actors: the organisation that adopts an OSS component and a prototypical OSS community that produces the OSS component. The reason why we only include the SR diagram for the adopter's organisation is because the evaluation is from the adopter point of view. The higher goals, inside the organisation actor, represent the organisation strategic goals, for example to get involved with OSS ("OSS

Involvement") and benefit from co-creating software assets with an OSS community ("Benefit from co-creation taken"). The low-level goals and tasks, which represent requirements for an adequate application of the organisation OSS business strategy, appear further below and include for example "Select component" and "OSS Community contributed". Overall the purpose of the goal model is to help understand how such lower level goals and tasks which are susceptible different kinds of risks are linked to higher level business goals and qualities.

2.2 Risk Models

The bottom part of Figure 1 shows a risk model with selected OSS component adoption risks. The risk modelling language is built on top of existing i^*-based languages and designed to support modelling and analysing OSS component risks. In particular, the risk modelling language borrows the concepts of **Situation** and **Indicator** from works on business intelligence [2], and the concept of Event from the goal-risk framework [1]. The concepts have been adapted to the purpose of OSS component risk modelling and related to each other by ad-hoc relations. A further key concept is the **Measure**, a raw value obtained from individual observations of a real quantity at particular points in time. Measures are therefore time series. The **Indicator** is another key concept, providing a domain- and context-specific interpretation of measured quantities. More specifically, an indicator supports gathering evidence about states of things, and is defined as an abstract, normalised synthesis of the value of one or more measures. Indicators are in this sense derived metrics at a higher level of domain abstraction.

While a measure is expressed using a metric, for example, 30 days per bug, and can coexist with other measures made in different times, an indicator is an aggregation of measures and is expressed in a neutral way with respect to some reference boundaries (e.g., days per bug: 0.8). An indicator is typically calculated using a function, which takes one or more measures as input, and maps absolute numbers of measures onto an indicator interval [0..1]. The functions can be of different nature: simple mathematical functions, such as $1 - (1/(x + 1))$, depending on single measurements, or more sophisticated functions that use statistical parameters.

A **Situation** is an abstract representation of a state-of-affairs [6], i.e. a partial state of the world. Situations are expressed by means of propositions, which carry an evidence that things are in the state they describe. The evidence values associated with a situation may be numbers in a range [0..1].

An **Event** is the occurrence, at a given place and time, of a change in the state of things [1]. Events are more or less likely to happen in given situations, and can be more or less severe. Events are also expressed through propositions, and describe things that may happen at a certain time in the future, as opposed to situations, which describes things as they are at the time of the observation.

At the bottom of Figure 1, four indicators are depicted: "Commit ratio", "Forum posts per day", "Number of pending feature requests" and "Number of closed feature requests". A measure observed over an OSS component that is of interest, is associated with each one of them. Indicators provide evidence that

situations are true. For example, "Commit ratio" and "Forum posts per day" provide evidence (indicate relation) that the situation of "Low activeness" is true. This situation, in turn, can raise (or lower) the likelihood that some events occur in the future. For example, through the **expose** relation a positive likelihood contribution is established to the "Lack of support" and "Low release frequency" events, while through the **protect** relation a negative likelihood contribution is established to the "Fast API change" event.

2.3 Impact Models

The impact model consists in a set of relations that, in a way similar to [1], describes the connection between the risk model introduced in the previous section and the goal model describing the structure of an organisation. The main difference is that we consider explicitly the role played by an event's *likelihood* and its *significance*, which together determine the event's risk *exposure*. The **impact** relation, from a risk event to a goal, indicates that a higher exposure to the source risk event causes a negative impact on the satisfiability of the target goal. Several examples of impact relationships are given in Figure 1 such as the relationship between an event "e2" ("Loose control on evolution") and the softgoal "OSS Component evolves towards desired feature" meaning that if the Adopter looses control on the evolution of a particular OSS component, this can compromise the possibility to guide the component towards a planned set of features.

3 Risk Assessment

In RISCOSS, risk models are used in conjunction with i^* models to analyse the impact of OSS risk on business goals. The adopted risk analysis is inspired by the goal analysis technique presented in [3]. To each concept, one or more attributes are associated, which represent, for example, the evidence of satisfaction of a situation, the likelihood or severity of a risk event, or the impact on a goal. Starting from the values gathered by indicators, the values are propagated across the model through different relations.

Figure 1 presents an example of impact propagation in a goal model. The model includes two actors "OSS Community" and "Organisation", representing the OSS Adopter, the risk event "Inability to be accepted as contributor" (e3) directly connected to the situation "no possibility of contribution" (s3), and the indicators "Number of closed feature requests" (i2) and "Number of pending feature requests" (i13). We can start the propagation by acquiring data for the two measures from available the OSS community databases, for example 4 closed feature requests per month and 3 pending feature requests per month. These measures can indicate the situation of "no possibility of contribution" that exposes the risk of "Inability to be accepted as contributor" for a given OSS Adopter (see the dashed line). This event negatively impact the goal "OSS

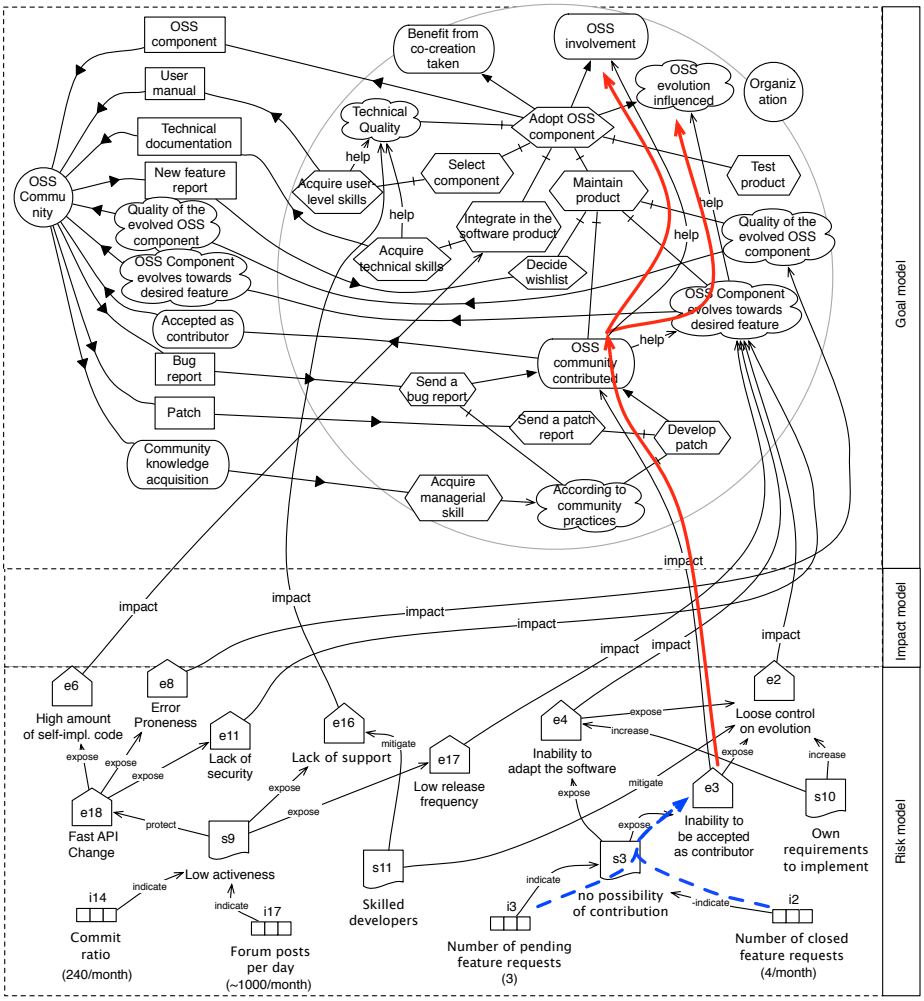

Fig. 1. A model of OSS component adoption risks.

community contributed" whose accomplishment can be compromised, also compromising the goal "OSS component evolves towards desired feature" and the higher level business goals "OSS involvement", and "OSS evolution influenced" that are connected via the "help" relationship (as shown by the continuous red line).

4 Conclusion and Future Work

In this work we described a modelling language, which makes use of i^*, extended by means of a risk model and an impact model to capture the impact of risks on business goals of an organisation. We are now developing a method, which makes use of risk models to perform risk mitigation. Among the available alternatives, the method seeks to find the one that better fits user needs while minimising risk. To this purpose, we need to improve the quality and accuracy of the models, and in particular of the parts related to risk, to have a reliable risk quantification. A tool is currently under development to support risk assessment in OSS component adoption. The tool automatically gathers measures from OSS data available online, and uses the i^* and risk models described here to infer knowledge about risks and to allow for a comparison among various components. The tool will be web-based and, once developed, it is intended to be proposed to OSS communities and adopters as a means to support risk assessment among practitioners. For our point of view, the tool will also serve to gather feedback from in-the-field usage.

Acknowledgement This work is a result of the RISCOSS project, funded by the EC 7th Framework Programme FP7/2007-2013, agreement number 318249.

References

1. Yudistira Asnar, Paolo Giorgini, and John Mylopoulos. Goal-driven risk assessment in requirements engineering. *Requir. Eng.*, 16(2):101–116, 2011.
2. Daniele Barone, Lei Jiang, Daniel Amyot, and John Mylopoulos. Reasoning with key performance indicators. In Paul Johannesson, John Krogstie, and AndreasL. Opdahl, editors, *The Practice of Enterprise Modeling*, volume 92 of *Lecture Notes in Business Information Processing*, pages 82–96. Springer Berlin Heidelberg, 2011.
3. Paolo Giorgini, John Mylopoulos, Eleonora Nicchiarelli, and Roberto Sebastiani. Formal reasoning techniques for goal models. *J. Data Semantics*, 1:1–20, 2003.
4. Isabelle Mirbel. Oflossc, an ontology for supporting open source development communities. In *ICEIS (4)*, pages 47–52, 2009.
5. Mirko Morandini, Alberto Siena, and Angelo Susi. Risk awareness in open source component selection. In *17th Int. Conf. on Business Information Systems (BIS'14)*. Springer, May 2014.
6. Alberto Siena, Ivan Jureta, Silvia Ingolfo, Angelo Susi, Anna Perini, and John Mylopoulos. Capturing variability of law with Nòmos 2. In *ER'12*, LNCS 7532, pages 383–396, 2012.