Aligning Business Goals and Risks in OSS Adoption[1]

Dolors Costal[1], Lidia López[1], Mirko Morandini[2], Alberto Siena[2], Maria Carmela Annosi[3], Daniel Gross[2], Lucía Méndez[1], Xavier Franch[1], Angelo Susi[2]

[1]Universitat Politècnica de Catalunya (UPC)
c/Jordi Girona, 1-3, E-08034 Barcelona, Spain
{dolors, llopez, franch}@essi.upc.edu, emendez@lsi.upc.edu
[2]Fondazione Bruno Kessler (FBK)
I-38123, Trento, Italy
{morandini, siena, dgross, susi}@fbk.eu
[3]Ericsson Telecomunicazioni S.p.A., Italy (TEI)
I-84016, Pagani, Italy
mariacarmela.annosi@ericsson.com

Abstract. Increasing adoption of Open Source Software (OSS) requires a change in the organizational culture and reshaping IT decision-makers mindset. Adopting OSS software components introduces some risks that can affect the adopter organization's business goals, therefore they need to be considered. To assess these risks, it is required to understand the socio-technical structures that interrelate the stakeholders in the OSS ecosystem, and how these structures may propagate the potential risks to them. In this paper, we study the connection between OSS adoption risks and OSS adopter organizations' business goals. We propose a model-based approach and analysis framework that combines two existing frameworks: the *i** framework to model and reason about business goals, and the RiskML notation to represent and analyse OSS adoption risks. We illustrate our approach with data drawn from an industrial partner organization in a joint EU project.

Keywords: risk analysis, open source software, *i** framework, i-star

1 Introduction

Open Source Software (OSS) has become a driver for business in various sectors, namely the primary and secondary IT sector. Estimates exist that in 2016, a 95% of all commercial software packages will include OSS components [1].

OSS adoption impacts in fact far beyond technology, because it requires a change in the organizational culture and reshaping IT decision-makers mindset. Hence, the way in which organizations adopt OSS affects and shapes their businesses. At the same time, OSS software components introduce various risks that may not be visible at the time of the adoption, but can manifest in later development and maintenance

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

phases, causing unexpected failures. These risks may have an impact on the business goals of the adopter organization.

In this paper, we study the connection between risks and the business goals of the OSS adopter organizations, and how risks propagating through OSS ecosystem structures may compromise stakeholders' strategic goals. The present work builds upon the results of two previous papers:

(1) In [2] we proposed goal-oriented models using the *i** approach [3] to model the different existing OSS adoption strategies. The models describe the consequences of adopting one such strategy or another: which are the business goals that are supported, which are the resources that emerge, which are the dependencies that exist between the different actors of the OSS ecosystem, etc.

(2) In [4] we presented a framework for risk modelling and risk evaluation, which is tailored to assess OSS adoption risks. The framework is comprised by a risk modelling language (RiskML) and a quantitative reasoning algorithm that analyses risk models.

The present work proposes to align RiskML models and *i** models to analyse the propagation of the risk impact towards the business goals of the OSS adopter and the rest of actors of an OSS ecosystem. It is guided by two main research questions:

- **RQ1: What is the conceptual relationship between OSS adoption risks and the adopter organization business goals?**
- **RQ2: How do OSS adoption risks affect the adopter organization business goals?**

RQ1 explores how the risk and business goal-oriented modelling approaches can be integrated into a single modelling framework. This will be done by formulating an integrated metamodel that will serve as a basis for the design of risk-aware OSS ecosystems (RQ1.1), and then by offering means to examine the relationship between risks and business goals at the instance level (RQ1.2). RQ2 explores how existing risk and business model analysis techniques can be combined to propagate the results of risk analysis to business goals, considering the relationships that may exist among actors which collaborate in OSS ecosystems.

This research is part of an ongoing European FP7 project (RISCOSS, www.riscoss.eu), which aims to support organizations in understanding, managing and mitigating risks during OSS adoption [5]. The preliminary validation of our research results was performed at one of the industrial partners (Ericsson Italy at Pagani, TEI), where the approach helped illustrating how risks during the adoption and maintenance of OSS components may impact on its business goals.

As research method we adopted a design science approach following the engineering cycle described in [6]. This fits well with the research aim to create new metamodel artefacts, while also acquiring new knowledge. Fig. 1 illustrates the cycle, which includes problem investigation, solution design and solution validation.

The rest of the paper is organized as follows. Section 2 briefly provides additional background which is illustrated through a running example drawn from Ericsson's business and development environment. Sections 3 to 5 present the integrated risk and business modelling framework, the steps taken to align both risk and business models

and illustrates the proposed analysis engine. Section 6 presents related work, while Section 7 concludes and points to future work.

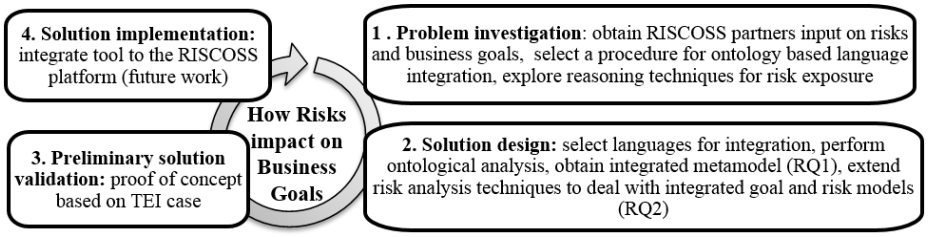

Fig. 1 Steps of the engineering cycle following [6].

2 Background

In this section we present the two modelling frameworks upon which we build our proposal, namely the *i** framework and the RiskML modelling language. In order to illustrate their concepts, we will use a running example from the RISCOSS project.

Running example. TEI is part of Ericsson, one of the world's leading telecommunication corporations. Ericsson produces hardware (telecommunications infrastructure and devices) as well as the software to run it. The company's mission is to empower people, business and society at large, guided by a vision of a sustainable networked society. One of TEI's roles within the Ericsson ecosystem is to provide OSS alternatives to support efficient third party products handling. However, adopting OSS components also exposes TEI to risk, because OSS comes typically without legal contracts that guarantee the adopter over time about software functionalities and qualities, so the company may suffer towards its partners and customer for lacks in the adopted software. It must therefore undertake adequate actions to analyse, assess and possibly mitigate potential risks.

2.1 Business Goal Models: *i**

In OSS ecosystems, actors pursue their goals while interacting with other actors. The *i** framework [3] was formulated for representing, modelling and reasoning about socio-technical systems. Its modelling language (the *i** language) is composed by a set of graphic constructs which can be used in two diagrams. Firstly, the Strategic Dependency (SD) diagram, including the organizational *Actors*. Actors have *Dependencies*, one actor (*Depender*) depends on other (*Dependee*) for the achievement of some intention (*Dependum*). The main intentional elements are: *Resource*, *Task*, *Goal* and *Softgoal*. Softgoals represent goals with no clear criteria for their satisfaction.

Secondly, the Strategic Rationale (SR) diagram represents the internal actors' rationale. The rationality of each actor is represented using the same types of intentional elements described above. Additionally these intentional elements can be interrelated by using relationships such as *Means-end* (e.g., a task can be a mean to achieve a

goal), *Contributions* (e.g., some resource could contribute to reach a quality concern or softgoal) and *Decompositions* (e.g., a task can be divided into subtasks).

Fig. 2 shows an excerpt of the TEI business model related to the maintenance of products including some OSS component, and how its business goals impact on Ericsson's goals. Besides the typical business goal of any organization for reducing costs (goal *Cost reduced*), the main objectives for TEI are fulfilling the Ericsson's customers' requirements (softgoal *Product requirements achieved*) using a *Maintainable code* in order to secure the quality (*Quality of code*). For TEI it is crucial to use *Mature technology* and *Secure code*. When TEI decides to use an OSS component, there are three possibilities for maintaining this code: they can assume the activity (*Provide in-house maintenance*), rely to the community behind the OSS component (*Rely on the OSS community for maintenance*) or rely to a third party organization (*Contract 3PP organization for maintenance*). For this portion of the TEI's business model, the impacted Ericsson business goals are *Time-to-market reduced* and *Reputation kept*. Ericsson expects from TEI that the *Development time is reduced*, *Responsiveness* and *Reliable products* for achieving its business goals. Notice that the model only includes the third party organization (*3PP OSS Provider*) in order to illustrate that the maintenance is outsourced; for the sake of brevity, the description of this relation is not exhaustive and not all dependencies between both organizations are included in this model.

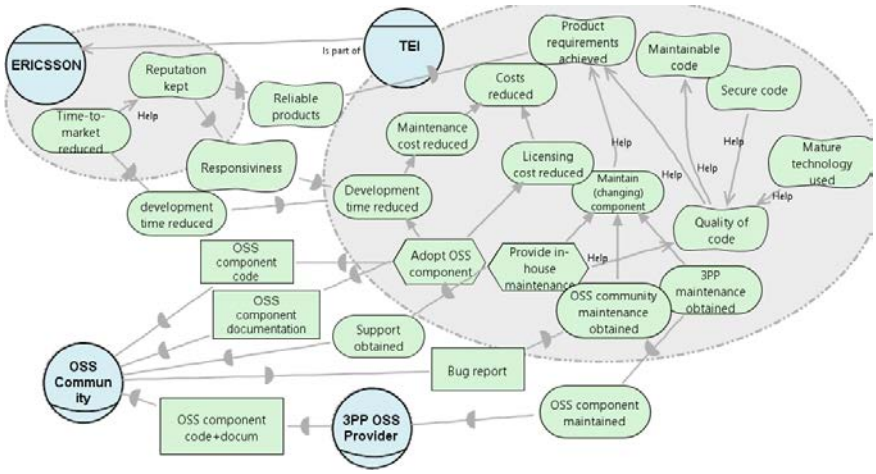

Fig. 2 TEI business goal model

2.2 Risk Models: RiskML

OSS ecosystems rotate around the production and use of OSS software components. RiskML is a modelling language introduced in [4] to capture knowledge related to risks of software components and to support automated analysis. RiskML uses the concepts of *Event* - a change in states of affairs, which may harm goals [7], with a certain *likelihood*, and *significance; Goal* - anything, which is of interest for a stakeholder to obtain or to maintain; and *Situation* - states of affairs, under which risks are

possible [8][9]. Additionally, *Indicators* represent one (simple indicator) or more (composite indicator) gathered measures about a certain property of a software component [8]. By means of transformation functions, indicators inform about the evidence of being in certain situations. The *Indicate* relation represents such transformation, propagating the value of an indicator to the evidence that a situation is satisfied. *Expose*, *Protect*, *Increase* and *Reduce* relations raise a target event's likelihood, lower it, raise an event's significance or lower it, respectively. The *Impact* relation represents the negative effect of a certain event on the satisfaction of a given goal. The higher the impact, the higher is the severity of the negative impact. The *exposure* to a certain risk is defined as a combination of the risky event's likelihood, its significance, and the severity of its impact to goals. For details on RiskML see [4].

Fig. 3 shows an excerpt of a risk model related to OSS code maintainability (indicators in relationship to situations are not displayed for space reasons, as well as information like likelihood and significance). The model is based on (1) interviews with managers, (2) a literature study on OSS risks [10] and (3) various metrics for code quality, such as complexity metrics, lines of code, and test coverage; such metrics have been shown in [11] to serve as measures of the maintainability of the source code. These measures do not offer absolute measures of maintainability, but assist in identifying where code exhibits properties that are known to decrease (or increase) maintainability, showing correlation e.g. with the time and skills needed to maintain the code. For example, code that is more complex takes more time to be understood by the analysts; this can lead to delays in bug fixing and code maintenance and evolution activities.

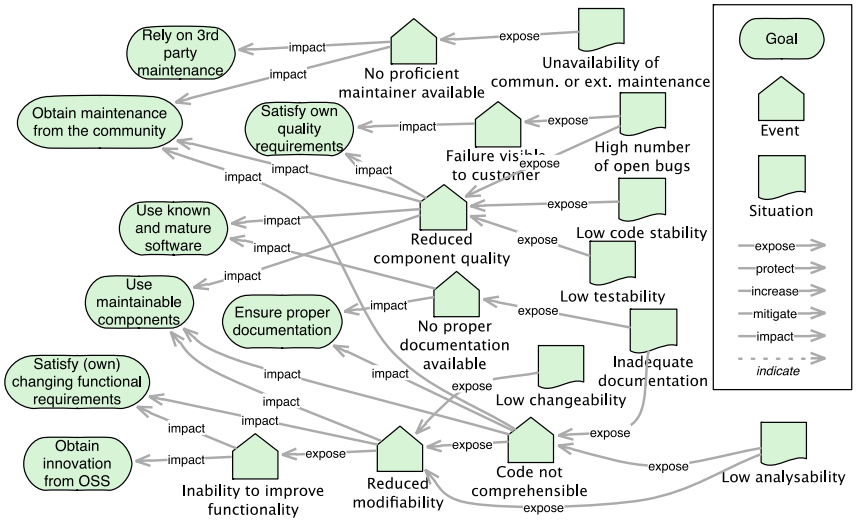

Fig. 3 RiskML risk model for OSS code maintainability

Following the definition of *risk* as a combination of event likelihood and the severity of its impact, the risk model comprises risk events, situations as a means to capture risk indicators, and the goals that are directly impacted by the risk events. For example, the central goal of this risk model is *Use maintainable component* (regarding the

quality and modifiability of the code and the availability of documentation), there are high level goals like *Satisfying own quality and functional requirements* throughout the time of maintenance, and lower level goals such as to continuously *Obtain innovation from the OSS* community, to have access to a *proper documentation*, or to rely on a proficient *3rd party for maintenance.*

3 An Integrated Model for Risks and Goals

3.1 Analysis of Overlapping Concepts

Integrating the product models of two methods requires identifying concepts that have the same semantics, and to merge them afterwards. Following the approach proposed in [12], we have opted for using ontological analysis to identify these concepts, mapping the concepts of the two methods to the concepts of a reference ontology. Among possible options (e.g., BWW, Chisholm's, DOLCE, etc.), we have chosen the UFO ontology [13][14]. UFO is a foundational ontology that has been used to analyse, redesign and integrate language models in a large number of domains.

The starting points are the mappings between both modelling languages concepts (*i** and RiskML) and those of UFO. The concepts which are mapped onto the same or related UFO concepts are considered candidate overlapping concepts for an integrated *i**-RiskML model. Finally, we analyse the overlap and decide whether to map the concepts unconditionally (i.e. in all cases) or under certain conditions. Fig. 4 presents our initial mappings and the candidate overlapping concepts obtained from them. Only the mappings involved in our analysis are included.

Regarding the *i** and UFO mapping, we adopt the interpretations presented in [15] (see Fig. 4). We use the same notion for softgoal as [15] since it does not have a uniform treatment, as the paper points out. We consider that an *i** *Softgoal* is a goal for which it is possible that two rational agents differ in their beliefs to which situations satisfy it. Conversely, for an *i** *Goal* (or hardgoal), the set of situations that satisfy it is necessarily shared by all agents.

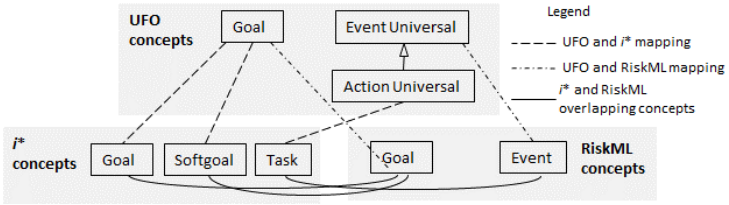

Fig. 4 *i** and RiskML overlapping concepts according to UFO mappings

Next, we provide the mapping between RiskML and UFO[2]. A *Goal* in RiskML is defined as anything which is of interest for a stakeholder to obtain or maintain. As such, it is satisfied if the corresponding state of affairs is achieved. Goals in UFO are related to sets of intended states of affairs of an agent. UFO contemplates a relation

[2] UFO concepts appear underlined in the text whereas RiskML and *i** ones appear in italics.

between Situations and Goals such that one or more Situations may satisfy a Goal. In other words, a Goal is a proposition and a particular state of affairs can be the truth-maker of that proposition. Consequently, the RiskML *Goal* concept can be interpreted as a Goal in UFO. *Events* in RiskML model changes in states of affairs. From the UFO ontology, we have that an Event (instance of Event Universal) is a perduring entity, i.e., entities that occur in time, accumulating their temporal parts. Events are triggered by certain Situations in reality (termed their pre-situations) and they change the world by producing a different post-situation. Consequently, the RiskML *Event* can be interpreted as an UFO Event Universal.

Next we describe the *i**-RiskML candidate overlapping concepts identified after analysing the mappings (see Fig. 4). The RiskML *Goal* overlaps both the *i* Goal* (also called hardgoal) and the *i* Softgoal*. These concepts map unconditionally, meaning that any *i* Goal* and any *i* Softgoal* maps into a RiskML *Goal* and also any RiskML *Goal* maps into either a *i* Goal* or a *i* Softgoal*, since they all map into the same UFO concept. The RiskML *Event* overlaps the *i* Task*. These two concepts map only under certain conditions because: (1) they map into UFO Event Universal and UFO Action Universal, respectively and, (2) according to UFO, only Events deliberately performed by Agents in order to fulfil their Intentions are Actions. Therefore, all *i* Tasks* map into RiskML *Events* but not all RiskML *Events* map into *i* Tasks*.

3.2 Analysis of the *Impact* Relation

The *Impact* relation in RiskML relates an *Event* E and a *Goal* G such that the occurrence of E has an effect on the satisfaction of G (e.g. the event post-situation does not satisfy the goal). Since RiskML *Goals* map into *i* Goals* and *i* Softgoals*, it follows that goals and softgoals are the only *i** elements that can be the target of an *Impact* relation. However, we argue that *i* Tasks* and *i* Resources* could also be involved in impact relations for shortcutting purposes.

As mentioned above, *i* Tasks* are interpreted as UFO Action Universals. According to UFO, Actions are intentional Events, i.e., events with the specific purpose of satisfying some goals. Therefore, any Action Universal (*Task*) implies the existence of an underlying Goal, explicit or not (i.e., it is a hidden goal). If the hidden Goal of a *Task* is impacted by an *Event*, as a shortcut, we allow to use *Task* as target of the *Impact* relation to avoid making the goal explicit and thus to cause excessive model growth).

The case of *i* Resources* is quite similar. Intuitively, for any *i* Resource* we may assume the existence of the underlying Goal on getting the resource available to an agent, explicit or not in the *i** model. Again, for shortcutting purposes, we propose to specify *Resource* as the target of the *Impact* relation.

3.3 Metamodel Integration: the *i**-RiskML Metamodel

We start from the *i** metamodel presented in [16] and the RiskML metamodel in [4]. To integrate them, we take into account the overlapping concepts (Section 3.1) and the shortcuts for the *Impact* relation (Section 3.2). For each set of overlapping con-

cepts and assuming that each one has a corresponding metaclass in the initial metamodels, we need to decide whether to keep all the corresponding metaclasses of both metamodels, or just the metaclasses from one metamodel. To make such decisions, we slightly adapt the heuristics proposed in [12]. In cases where the concepts are totally equivalent, the simplest solution is to keep only the metaclasses from one metamodel (it must be decided which one); the others are removed. Clearly, the associations in which the removed metaclasses participated need to be reconnected accordingly. In the cases where the mapping of concepts is qualified with a condition specifying under which circumstances they can be mapped, all metaclasses should be kept and relations should be defined between them. Table 1 summarises the application of these heuristics to our case and Fig. 5 shows the resulting metamodel.

Table 1. *i** and RiskML metamodel integration

RiskML	***i****	***i**- RiskML**	
Concept	**Concept**	**Metaclasses kept**	**Rationale**
Event	Task	Both (Task subclass of Event)	Task → Event, not (Event → Task)
Goal	Goal, Softgoal	Goal, Softgoal (from *i**)	Equivalent (*i** more fine-grained representation for goals)
Assoc.		**Metaassociations reconnected**	**Rationale**
Impact		relates Event to IntentionalElement (superclass Goal, Softgoal, Task, Resource)	Goal and Softgoal are equivalent to Goal, Task and Resource for shortcutting

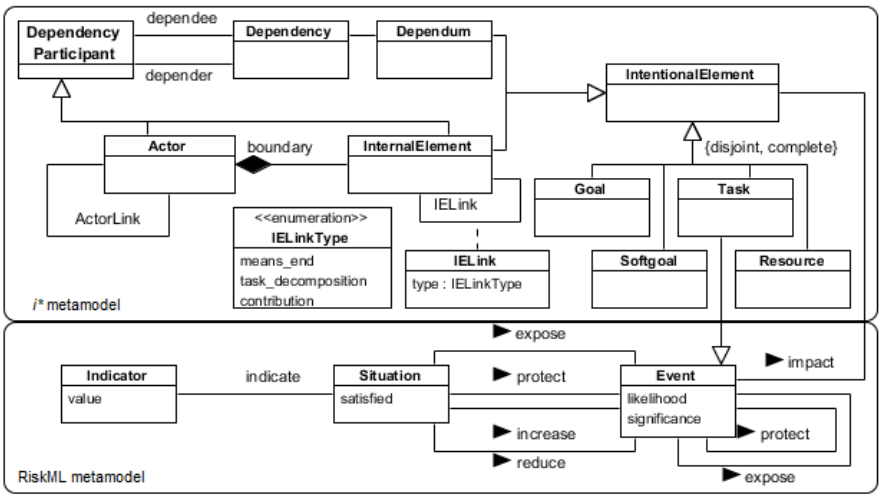

Fig. 5 *i** - RiskML integrated metamodel

4 Aligning Models for an Organization

In this section, we consider the alignment of a business model and a risk model in a concrete organization. We use the TEI case presented in Section 2.

It is worth to mention that the concrete form of the alignment may depend on the business case in which it is done. In this section, we assume that: the risk model is part of a catalogue of reusable models and as such, it cannot be modified. The business model is produced independently of the risk model (either because it already existed, or the business modeller was not aware of the risk model, or even it could have been a conscious decision to avoid bias in the business model). We think that this scenario can be quite usual when analysing the impact of risks in business goals.

4.1 Alignment Method for the *Impact* Relation

According to the results of Section 3, we analyse the possible mapping of every impacted goal in the risk model with some intentional element in the business model. In other words, the alignment problem may be stated as:

> Given a business model B which contains a set I of intentional elements, and given a risk model R which contains a set G of goals, we want to combine them to produce a new model M in which the goals in G are semantically connected to the intentional elements in I according to the ontological framework defined in 3.

We define M as initially including B's actors with their corresponding SR diagram as in B. We analyse next the effect of each goal g in the risk model on the initial model M. Let's call M' the model resulting of this step.

Alignment case 1. There is an intentional element x in B such that g can be considered semantically equivalent to x. In this case, the model M' will keep x and will include the impacts from events in R to g but changing x by g.

Alignment case 2. There is an intentional element x in B which subsumes g. In this case, the model M' includes both related through the appropriate model construct (e.g., means-end if x is a goal, or contribution link if it is a softgoal). The impacts from events in R to g are also included in M'.

Alignment case 3. There is no x in B satisfying cases 1 or 2. This means that there is no obvious impact from the risk to any business goal. In this case, it is necessary to further interact with the business analyst with two possible outcomes:

a. g is in fact pointing out a business goal which has been neglected in the initial version of the business model B. The business model needs to be updated and g finally falls into the cases 1 or 2.
b. g is indicating a risk that is not important for the company. In this case, g is not added in M', as well as any other element in R (e.g situation) related only to g.

4.2 Alignment Application and Results

The alignment between a general risk model (Fig. 3) and an organization business goal model (Fig. 2), follows the iterative process previously defined. The risk model gives awareness on specific issues, which need to be analysed to decide if they may or may not be needed to be addressed in a specific organization, e.g. because a goal is not important in the particular context. The addressed goals are then put in relation to the organization's goals, also joining semantically similar goals from both models.

Table 2 Alignment of Business and Risk models

	g in R	x in B	New link
***Alignment case 1* (equivalent)**			
g1	Obtain maintenance from the community	Rely on the OSS community for maintenance	
g2	Rely on 3rd party maintenance	Contract 3PP organization for maintenance	
g3	Use known and mature software	Mature technology used	
g4	Use maintainable components	Maintainable code	
Alignment case 2 (subsumes)			
g5	Satisfy (own) changing functional requirements	Product requirements achieved	contribution link
g6	Satisfy own quality requirements	Product requirements achieved	contribution link
Alignment case 3.a (new business goals)			
g7	Ensure proper documentation	Ensure proper documentation (new)	task-decomposition (Adopt OSS component) contribution link (Maintainable code)
Alignment rule 3.b (discarded)			
g8	Obtain innovation from OSS		

Applying the guidelines presented above, Table 2 includes the alignment between the models presented in Section 2, a business model B (Fig. 2) and a risk model R (Fig. 3). In this concrete example, goals from g1 to g4 are equivalent (*Alignment case 1*). In this case, the elements from B are kept. For g1 and g2, the intentional elements in B are tasks, so the underlying goals to these tasks in B are equivalent to the goals in R. Goals g5 and g6 are subsumed by one element of B (*Alignment case 2*), therefore the elements from R are included in B as goals. Finally, for missing goals (*Alignment case 3*), g7 is included in the TEI business model (as a softgoal) and g8 is discarded.

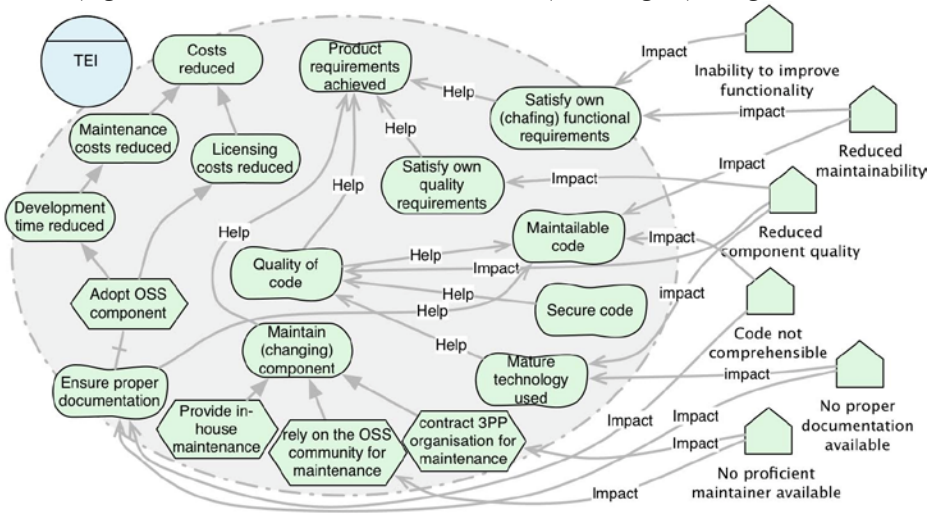

Fig. 6 TEI SR diagram connected to RiskML risk events

Fig. 6 shows the part of M in the TEI running example which includes the alignments above. Concretely, it includes the SR diagram for the actor TEI, including the new elements and the impacts from the Risk Events included in R.

5 Risk Analysis

The alignments described in the previous sections allow us to perform a model-based analysis of OSS ecosystems, linking the metrics of OSS projects to their impact on business goals. Risk analysis in RiskML is a reasoning technique, described in [4], which uses forward quantitative inference algorithms to evaluate risk exposure. The algorithm starts from the gathered indicators about OSS projects, and applies inference rules to derive exposure to risky events: firstly, indicator values are mapped onto the satisfaction evidence of situations; afterwards, situation satisfaction raises or lowers the occurrence likelihood of events (*expose* and *protect* relations) or their significance (*increase* and *reduce* relations). The impact of risk events on the software ecosystems is captured by goal analysis, which is a technique for reasoning on *i** models, described in [17]. In a nutshell, it relies on the idea that actors want their goals to be achieved, so well-engineered goal models should ensure goal satisfiability. In goal analysis, intentional elements hold a satisfiability evidence and a deniability evidence, representing the evidence that the intentional element can be achieved or not achieved. Satisfiability and deniability evidence can hold at the same time for the same intentional element, thus representing the existence of contradictory information. Both value types are propagated across an i* goal model: and-or decompositions propagate satisfiability evidence from operational goals to tactical and strategic goals, while contribution link represent partial or total, positive or negative effect of a source goal to a different one.

Risks affect negatively goals, because they can reduce their chance to be achieved. The *impact* relation represents this negative effect: the more the source event is likely and significant, the more there is evidence that the impacted goal is not achievable. This is depicted in the integrated model in Fig. 7, where the RiskML model concerning maintenance quality risks is plugged into the goal model of an adopter. E.g., the *Reduced component quality* risk event impacts on the *Quality of code* softgoal. If the event is exposed (likely and significant), there is evidence that the softgoal is denied. Once a goal has been impacted, this impact can be propagated across the goal model. The denial evidence of the *Quality of code* softgoal propagates through the model to other goals, having a negative effect on the *Maintainable code* softgoal and on the *Product requirements achieved* softgoal. When an actor is part of an OSS ecosystem, it depends on other actors for having goals fulfilled, and itself fulfils goals for them. When an actor (dependee) fails in fulfilling a goal for another one (depender), the depender suffers consequences. The Ericsson actor is at risk of losing reputation if TEI fails in satisfying the product requirements. Thus the value of the indicators, captured through the situations, raises risks that span through the whole model.

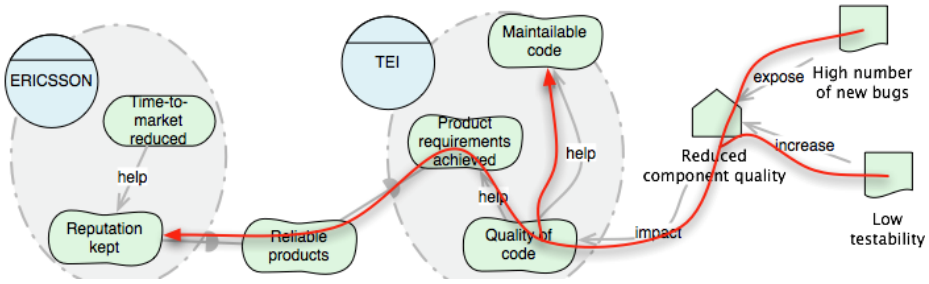

Fig. 7 Example flow of risk propagation in a OSS ecosystem

6 Related Work

Several works dealt with the modelling of risks in the context of organizations through goal-oriented languages. The Goal-Risk framework by Asnar et al. [7] uses *i** to capture, analyse and assess risk at an early stage of the requirements engineering process. They distinguish an asset layer to model business goals, an event layer to model risk events, and a treatment layer to model mitigation actions. We build on top of their approach and extend it in two directions: first, we support complex indicators, connecting risk models to data sources; and, second, we extend the goal analysis support, adding the capability to analyse how risks are propagated across a set of actors. Additionally, the choice of using two different languages - risk and goal modelling - instead of an integrated one, is of help in models reuse, allowing us to study how the same risks may impact on different strategies or different ecosystems.

The KAOS methodology [18] deals with risk management by complementing goal modelling with obstacle analysis that consists in identifying the adverse conditions that may prevent a goal to be achieved [19]. KAOS has a formal representation for goal models, and takes into account partial or probabilistic values for goal satisfaction, allowing quantitative reasoning on the models but does not integrate concrete measures and indicators. Sabetzadeh et al. [20] presents an approach, based on KAOS, which uses statistical reasoning to analyse the problem of introducing a new technology in an organization. Assessing the risk related to the possible non-compliance of the new technology to the fixed standards or internal regulations. CORAS [21] is a model-based approach for security risk analysis and assessment, comprised by a risk modelling language, a process for security analysis, and a tool for reporting risk analysis results. It limits to a defensive risk analysis to protect company assets, and does not rely on a particular reasoning technique. Finally, Grandry et al. [22] integrates an enterprise architecture model and an information system security risk management model by mapping concepts of two metamodels from both domains. A main difference to ours proposal is that it focuses on the management of an enterprise more than on the analysis of a larger organization where the presence of multiple interacting actors and the strategic dependencies among them is of crucial importance. Moreover, our approach also proposes a set of guidelines to align the different models at the level of the model instances.

7 Conclusions and Future Work

In this paper we have presented an ongoing work to analyse OSS adoption from a risk management perspective. The work is motivated by the need of industry actors, represented here by Ericsson, to understand the impact of OSS adoption risks on their business goals, and how that impact can spread over a whole ecosystem. The approach is quite general, though, and could be applied to other kinds of risks, providing the adequate business and risks models. We have chosen two modelling languages that are able to represent the business environment and the underlying risks, and support goal and risk analysis, respectively. To explore the interaction of risks and goals in the OSS ecosystem, we have developed a formal alignment of concepts, and shown how this reflects on the analysis, applying it to the case of OSS adoption.

The main contributions of this work are: (1) the integrated metamodel including goals and risks related concepts (RQ1.1), using the foundational ontology UFO in order to provide an ontological matching between the concepts *Goal* and *Event* from RiskML and *Goal*, *Softgoal* and *Task* of *i**, concluding that a *Risk* can *Impact* on any type of *i** Intentional Elements (*Goal*, *Softgoal*, *Task* and *Resource*); (2) a methodology to plug a risk model to a goal model. (RQ1.2); (3) the propagation techniques in order to show how risk exposure can reflect in an evidence of goal denial, which intuitively means that higher risk causes higher possibility that the goal will not be achieved (RQ2). The Ericsson case has been used as preliminary validation. Besides the validation of the formal framework, this case has shown the adequacy of the proposal in an industrial setting. Although we may expect that models may grow in more complete cases, the business and risk models themselves are not expected to grow proportionally, supporting then scalability of the approach. Of course, further validation of this statement is required.

The conceptual alignment described in this paper allowed us to map a formalism for reasoning on risk exposure onto another well-suited to reason on goal satisfaction. While this is good for integration purposes, having a finer-grained analysis technique would help in providing more specific results. Future work goes along several directions. First, we are interested in assigning importance degree to goals, in order to classify the risks impact on the basis of their severity. Also, this will allow us to develop reasoning techniques to select mitigation strategies to reduce risk exposure. Second, it will be important to explore the risk–goal relation in the other way, understanding how OSS adoption can impact the measures and modify the risk exposure. Third, we also need to work further in the alignment between risk and goal models, so that the process that has been depicted in Section 4 becomes more prescriptive (e.g. risks impacting on dependums). Lastly, further validation of the approach is needed in order to evaluate its practical implications such as the effort required to align models.

References

[1] Driver, M.: Hype Cycle for Open-Source Software. Technical Report, Gartner, 2013.

[2] López, L., Costal, D., Ayala, C.P., Franch, X., Glott, R., Haaland, K.: Modelling and Applying OSS Adoption Strategies. ER 2014: 349-362

[3] Yu, E.: *Modelling Strategic Relationships for Process Reengineering*. PhD thesis, University of Toronto, Toronto, Ontario, Canada, 1995.

[4] Siena, A., Morandini, M., Susi, A.: Modelling Risks in Open Source Software Component Selection. ER 2014: 335-348.

[5] Franch, X. et al.: Managing Risk in Open Source Software Adoption. ICSOFT 2013: 258-264.

[6] Wieringa, R.: *Design Science Methodology for Information Systems and Software Engineering*. Springer, 2014.

[7] Asnar, Y., Giorgini, P., Mylopoulos, J.: Goal-driven Risk Assessment in Requirements Engineering. Requirements Engineering Journal, 16(2), 2011: 101–116.

[8] Barone, D., Jiang, L., Amyot, D., Mylopoulos, J.: Reasoning with Key Performance Indicators. PoEM 2011: 82–96.

[9] Siena, A., Jureta, I., Ingolfo, S., Susi, A., Perini, A., Mylopoulos, J.: Capturing Variability of Law with Nòmos 2. ER 2012: 383–396.

[10] Morandini, M., Siena, A., Susi, A.: Risk Awareness in Open Source Component Selection. BIS 2014: 241-252.

[11] Heitlager, I., Kuipers, T., Visser, J.: A Practical Model for Measuring Maintainability. QUATIC 2007: 30–39.

[12] Ruiz, M., Costal, D., España, S., Franch, X., Pastor, O.: Integrating the Goal and Business Process Perspectives in Information System Analysis. CAiSE 2014: 332-346.

[13] Guizzardi, G.: *Ontological Foundations for Structural Conceptual Models*. Ph.D. Thesis. University of Twente. The Netherlands, 2005.

[14] Santos, P.S. Jr., Almeida, J.P.A, Guizzardi, G.: An Ontology-based Semantic Foundation for ARIS EPCs. SAC 2010: 124-130.

[15] Guizzardi, R.S.S, Franch, X., Guizzardi, G.: Applying a foundational ontology to analyze means-end links in the *i** framework. RCIS 2012: 1-11.

[16] Lopez, L., Franch, X., Marco, J.: Making Explicit Some Implicit *i** Language Decisions. ER 2011: 62-77.

[17] Giorgini, P., Mylopoulos, J., Nicchiarelli, E., Sebastiani, R.: Reasoning with Goal Models. ER 2002: 167–181.

[18] van Lamsweerde, A., Letier, E.: Handling Obstacles in Goal-oriented Requirements Engineering. IEEE Transactions of Software Engineering, 26(10), 2000: 978–1005.

[19] Cailliau, A., van Lamsweerde, A.: Assessing Requirements-related Risks through Probabilistic Goals and Obstacles. Requirements Engineering Journal, 18(2), 2013: 129–146.

[20] Sabetzadeh, M, Falessi, D., Briand, L.C, Di Alesio, S., McGeorge, D., Åhjem, V., Borg, J.: Combining Goal Models, Expert Elicitation, and Probabilistic Simulation for Qualification of New Technology. HASE 2011: 63-72

[21] Lund, M.S., Solhaug, B., Stølen, K.: *Model-Driven Risk Analysis - The CORAS Approach*. Springer, 2011.

[22] Grandry, E., Feltus, C., Dubois, E.: Conceptual Integration of Enterprise Architecture Management and Security Risk Management. EDOC Workshops 2013: 114-123.