RISCOSS: Managing Risk and Costs in Open Source Software Adoption

Oscar Franco-Bedoya[1], Xavi Franch[1], Angelo Susi[2], Maria C. Annosi[4], Ruediger Glott[5], Ron Kenett[3], Lidia López[1], Fabio Mancinelli[6], Pop Ramsamy[7], Cedric Thomas[8], David Ameller[1], Claudia Ayala[1], Stijn Bannier[5], Ron Ben-Jacob[3], Nili Bergida[3], Yehuda Blumenfeld[3], Olivier Bouzereau[8], Dolors Costal[1], Manuel Domínguez[7], Daniel Gross[2], Kirsten Haaland[5], Jorge Martín[7], Mirko Morandini[2], Marc Oriol[1], Alberto Siena[2] and Nikolas Galanis[1]

[1]Universitat Politècnica de Catalunya - BarcelonaTech, Spain
[2]Fondazione Bruno Kessler, Italy
[3]KPA Ltd., Israel
[4]Ericsson Telecomunicazioni, Italy
[5]University of Maastricht (UMM), The Netherlands
[6]XWiki SAS, France
[7]CENATIC, Spain
[8]OW2, France
{ohernan,franch,llopez,dameller}@essi.upc.edu,
{cayala, dolors, moriol, ngalanis}@essi.upc.edu,
{susi, gross18, morandini,siena}@fbk.eu,
{ron, nilib, yehudab}@kpa-group.com,
mariacarmela.annosi@ericsson.com,
{glott.ruediger, kirstenhaaland, stijn.bannier}@gmail.com,
fabio.mancinelli@xwiki.com,
{pop.ramsamy, manuel.dominguez, jorge.martin}@cenatic.es,
{olivier.bouzereau, cedric.thomas}@ow2.org
http://www.riscoss.eu

Abstract. Open Source Software (OSS) has become a strategic asset in software development, and open source communities behind OSS are a key player in the field. By 2016 an estimated 95% of all commercial software packages will include OSS. This extended adoption is yet not avoiding failure rates in OSS projects to be as high as 50%. Inadequate risk management has been identified among the top mistakes to avoid when implementing OSS-based solutions. Understanding, managing and mitigating OSS adoption risks is therefore crucial to avoid potentially significant adverse impact on the business. This chapter introduces the RISCOSS decision support platform. RISCOSS develops a risk management-based methodology to facilitate the adoption of open source code into mainstream products and services. RISCOSS develops a methodology and a software platform that integrate the whole decision-making chain, from technology criteria to strategic concerns. Using advanced software engineering techniques and risk management methodologies, RISCOSS develops innovative tools and methods to identify, manage and mitigate risks of integrating third-party open source software. RISCOSS is the only platform to deliver a complete solution rather than a piecemeal approach to enable mainstream product developers to safely integrate open source software in their developments. Itself an open source project,

MartÃ■n J., Costal D., Dominguez-Dorado M., LÃşpez L., Oriol M., Ramsamy P., Mancinelli F., Bergida N., Blumenfeld Y., Kenett R., Ben-Jacob R., Thomas C., Bouzereau O., Galanis N., Franco Bedoya O., Franch X., Ameller D., Ayala C., Annosi M., Susi A., Haaland K., Bannier S., Gross D., Siena A., Morandini M. and Glott R.
RISCOSS: Managing Risk and Costs in Open Source Software Adoption.
DOI: 10.5220/0006144600560077
In *European IST Projects - The Quest for Excellence Towards 2020 (EPS Vienna 2014)*, pages 56-77
ISBN: 978-989-758-101-4

RISCOSS is open to third party contributions to help the platform grow in functionalities and make the transition to a fully marketable product or service. The platform will be validated against a collection of use cases coming from different types of organizations and emergent small OSS products.

Keywords. Open Source Software, FLOSS, OSS, OS Community, Free Libre Open Source, Business Strategy, Software Adoption, Risk, European Project.

1 Introduction

Open Source Software (OSS) has become a strategic asset for a number of reasons, such as its short time to-market software service and product delivery, reduced development and maintenance costs, and its customization capabilities. Open source technologies are currently embedded in almost all commercial software, by 2016, they will be included in 95% of all commercial software packages [1]. Figure 1 shows the growing trend in OSS projects. In spite of the increasing strategic importance of OSS technologies, still IT companies and organizations face numerous difficulties and challenges when making the strategic move to the open source way of working [2]. In fact, according to the most popular OSS portal, SourceForge, most OSS projects have ended in failure: 58% do not move beyond the alpha developmental stage (22% of them remain in the planning phase, while 17% remain in the pre-alpha phase, and some of them become inactive). Among the roots for these failures, it stands that OSS is about freedom and choice, but freedom and choice introduce risk [3]. The risks that IT companies face when integrating OSS components into their solutions are not to be neglected and incorrect decisions may lead to expensive failures. Insufficient risk management has been recently reported as one of the five topmost mistakes to avoid when implementing OSS-based solutions [4]. Financial institutions are required to manage such risks under the Basel III global regulatory standard and their capital requirements are determined accordingly [5]. With proper risk management and mitigation, failure could be reduced or negative impact and cost minimized. The RISCOSS project was launched to address issues raised by communication equipment manufacturers looking to integrate open source code into their products. While open source software is now recognized as an indisputable industry asset, many mainstream companies and managers are still uncomfortable about making the strategic move to the open source way of working. RISCOSS not only enables users to collect informed intelligence on open source components, but also goes one step further by offering risk analysis that adapts to individual business situations. In the RISCOSS project we are interested in (i) understanding the risks that lie in the selection of software components, and (ii) putting such risks in relation with the decision-maker's business and its objectives. This chapter introduces RISCOSS decision support platform that uses advanced software engineering techniques and risk management methodologies to help managers and decision makers unfamiliar with the underlying mechanisms of the open source world, make informed decisions regarding integrating open source components into their own projects, products or services.

RISCOSS is a collaborative project with funding support by the European Commission as part of the FP7 program under contract 318249. It develops both a methodology

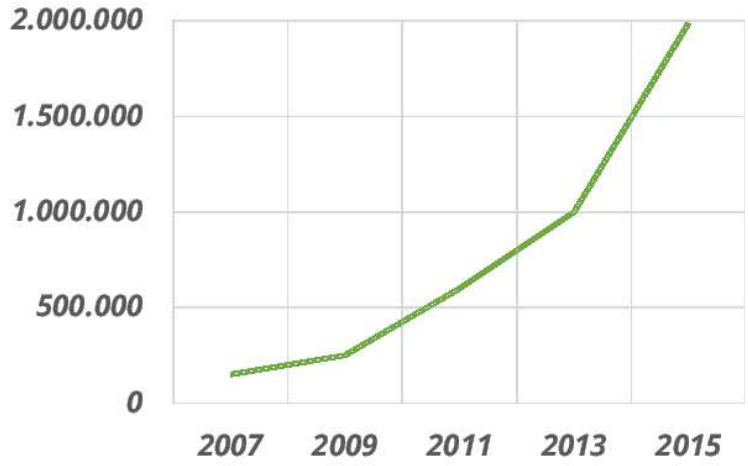

Fig. 1. Open Source Software; number of projects trends. Source: Black Duck Management webinar 2014 in IEEE OSS webcasts series.

and a software platform that integrate the whole decision-making chain, from technology criteria to strategic concerns to help managers identify, manage and mitigate risks of integrating third-party open source software

2 Objectives of the Project

The RISCOSS project is articulated around five main objectives:

- *Objective O1*. Strategic modelling and analysis of OSS-based ecosystems. Comprehensive representation of the elements of an OSS-based ecosystem and analysis techniques to discover relevant properties of this ecosystem with the aim of reusing it in designing new and more efficient ecosystem
- *Objective O2*. Risk management of OSS projects. Support to the establishment of practices and processes, based on innovative software engineering and statistical assessment and measurement techniques, for the management of risk in a continuous and incremental way.
- *Objective O3*. Business models and services for OSS solutions. Support firms capacities and resources to identify and evaluate the impact of OSS-related risks on their business models and for the management of OSS across the different business model components.
- *Objective O4*. Deployment of a software engineering platform for supporting decision making. Construction of an open-source platform integrating the methods, models and techniques developed in the context of the project.
- Objective O5. Industrial validation of project results. Demonstrate the results of the project in a number of representative use cases, drawn from different application domains, involving the appropriate actor.

The attainment of these objectives should bring as main benefit the reduction of the risks, costs and time needed to construct and evolve software systems in todays

European society. In the long term this is expected to improve the competitiveness of European industry, ensuring Europes global leadership in ICT.

3 Business and Software Ecosystem

Our approach basically elaborates around the idea of business and software ecosystems. Moore [6] coined the term business ecosystem to describe “an economic community supported by a foundation of interaction between organizations and individuals — the organisms of the business world. This economic community produces goods and services of value to customers, who are themselves members of the ecosystem. The member organizations also include suppliers, lead producers, competitors, and other stakeholders. Over time, they co-evolve their capabilities and roles, and tend to align themselves with the directions set by one or more central companies. Those companies holding leadership roles may change over time, but the function of ecosystem leader is valued by the community because it enables members to move toward shared visions to align their investments and to find mutually supportive roles”.

Business ecosystems have their equivalent at the technological level. Messerschmitt and Szyperski [7] used the term software ecosystem to describe the broader commercial, legal (regulatory) and market context in which traditional software systems operate. Companies such as Apple and Google have embraced a network centric view of software ecosystems, and developed novel business models, with varying degrees of openness from the adoption of selected open web standards, to the promotion of key web APIs as ad-hoc standards, to the (more or less) full embrace of open source software to encourage the emergence of massive global hardware/software ecosystems surrounding their products and services (e.g. iPhone, Android, etc.). Key arguments why companies adopt a software ecosystem approach to support their products and services offerings include [8], [9]: increase value of the core offering to existing users; increase attractiveness for new users; accelerate innovation through open innovation in the ecosystem; collaborate with partners in the ecosystem to share cost of innovation; platformize functionality developed by partners in the ecosystem (once success has been proven), and decrease total cost of ownership for commoditizing functionality by sharing the maintenance with ecosystem partners.

When it comes to OSS, both types of ecosystems have their peculiarities. As mentioned before, OSS based business ecosystems require business models that take account of the potential impact of OSS specifics on the production, distribution, costs and revenues aligned with or derived from OSS-related value propositions. OSS-based software ecosystems should address licensing problems, component interdependencies and frequency of releases, for instance. Helander and Rissanen [10] focus on the co-creation of value in OSS value networks, thus highlighting an aspect of OSS-based ecosystems that is important especially for businesses. The authors define value-creating networks “...as entities consisting of several directly or indirectly connected individual or organizational actors that transform and transfer different kinds of resources in order to create value not only for the networks end customer but also to themselves.” Each actor within the value network performs those tasks in which he has specific expertise, and together all partners create added value that finally benefits the end user. There are a number

of diverse actors that can form an OSS value network, starting from OSS projects and developer communities and ending with various end users, and mediators in between. Each actor is assumed to pursue common as well as particular interests. The links between more strategic business ecosystems and more IT-oriented software ecosystems is one of the focal points of our approach [2].

3.1 Modeling the Business and Software Ecosystems

While considering open source adoption in terms of risk management, it becomes clear that code itself is just the tip of the iceberg. Open source software is defined by the stakeholders that support it: contributors, communities, users, open source organizations, etc. The RISCOSS risk-based perspective highlights the role of the business and software ecosystems. It takes into account both the point of view of the communities providing the software components and that of the companies looking to integrate them. And here is one of the key innovation provided by the RISCOSS solution: it leverages advanced academic research to create a model of the business and software ecosystems relating to a target open source component. RISCOSS is based on a foundational ontology that links key concepts in business and (open source) software ecosystems [11]. In this context, our first objective is to support the risk assessment processes of OSS adoption by using i* models as a basis for the analysis of the strategic perspective of the OSS-based ecosystem that involves the assessment of the OSS project ecosystem and the adopting organization ecosystem. Figure 2 displays an excerpt of the ecosystem diagram for an OSS ecosystem called XWiki (www.xwiki.org), which is one of the case studies of the RISCOSS project. It is an i* strategic diagram and this shows the ecosystem dependencies, (e.g. resources, tasks, goals) between the actors in the business ecosystem. These models could be a valuable tool for the adopting organizations, for having an overview on the OSS project ecosystem and evaluating the risks implied in a potential adoption [12].

i* Framework. The i* framework [13] is a goal-oriented and agent-oriented modeling framework. i* is currently one of the best organizational modeling techniques. Its main feature is its ability to represent intentional social relations among stakeholders. It provides the equired infrastructure to model concepts such as actors, roles and agents, and to develop reasoning mechanisms with them. The i* framework defines two key models at different level of abstraction: the strategic dependency model and the strategic rationale model. A set of modeling primitives defines the model components and the relationships among them, where each business element is labeled according to its description.

XWiki. XWiki is an open source platform for building collaborative applications. It is developed by a communit. XWiki is publicly developed on the GitHub forge. The company also has a public wiki, several mailing lists, a bug tracking system and IRC channels where the community exchange information about the product. Internally we deal with different clients using almost the same means, but in a private way. The idea is to provide data coming from these sources as an input to the methodology. Public data

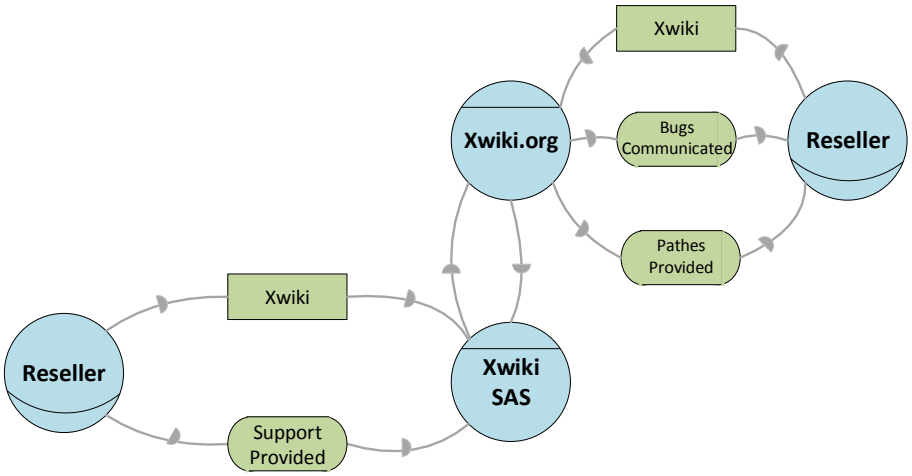

Fig. 2. XWiki business ecosystem model.

is already available. Client data might require some approval before being examined. This use case will be run together with the OW2 case, since XWiki is member of the OW2 association.

4 RISCOSS Consortium

RISCOSS combines knowledge and expertise from universities and companies with strong background in risk and costs management. The RISCOSS consortium is coordinated by Universitat Politècnica de Catalunya. By bringing together organisations from five countries (France, Israel, Italy, Spain and Netherlands), the consortium will have an effective impact over Europe and beyond. Figure 3 shows the members of the RISCOSS consortium.

5 RISCOSS Use Cases

As the composition of the projects consortium shows, it is our firm goal to demonstrate the feasibility of the RISCOSS approach in a significant number of industrial use cases. This will help to collect empirical data for assessing the objectives of the project and thus to further feed scientific work as the project progresses: (1) Ericsson TEI[9] is producing regulatory products for the Ericsson Corporate, including state-of-the-art products for data retention and lawful interception, belonging to the regulatory solutions product family. (2) CENATIC[10] is the national center for the application of information technologies and communication based on open source. It is a government of Spain to promote awareness and use of free software. (3) OW2[11] is an independent,

[9] www.ericsson.com/it
[10] www.cenatic.es/
[11] www.ow2.org/

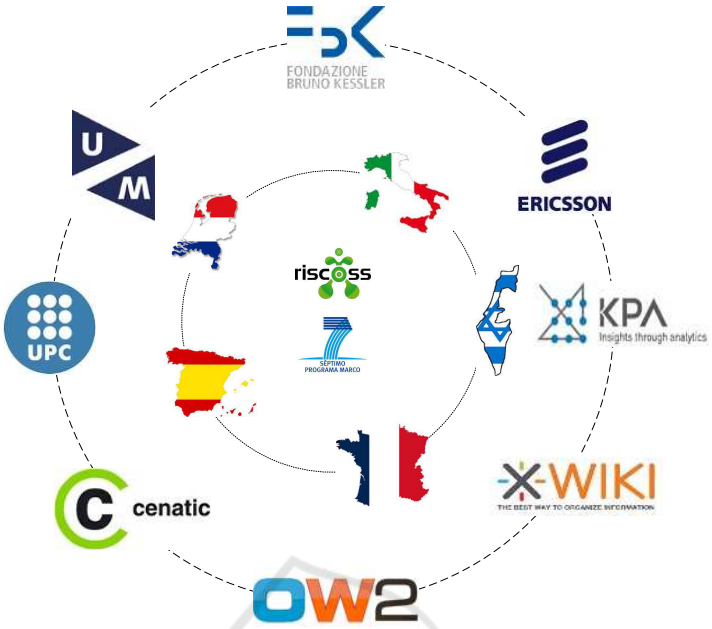

Fig. 3. RISCOSS Consortium.

global, open-source software community. The mission of OW2 is to promote the development of open-source middleware, generic business applications, cloud computing platforms technical infrastructure. (4) XWiki[12] is a for profit organization. Its business model is based on: open source software strategy and selling high-added value services to customers and partners installing, deploying and operating XWiki solutions. XWiki offers both a genericplatformfor developing collaborative applications using thewiki paradigm and projects developed on top of it. (5)Moodbile[13] open source project aims to enable mobile learning applications (and other kinds of applications for education) to work together with LMSs (Learning Management Systems)

5.1 A Reference Scenario

To better comprehend the nature of the problem, consider the case of a commercial company manufacturing communication equipment. These products call on a lot of software manufacturing companies are in a complex ecosystem, their product lines manage many different products composed of thousands of software components including commercial and open source components and components developed in-house. They need to maintain several versions of each product at the same time for each of the variants, for instance, two versions in maintenance and another in development. There may be different versions or variants of the same product when products are customised to meet the requirements of specific markets. While a given component is generally used through

[12] www.xwiki.com
[13] www.moodbile.org

the different releases of the different products, it may not incorporate all the patches currently available within the community perhaps because of the interdependencies between components that need to be properly handled. Such is the situation that managers need support in order to manage all the risks that appear during the lifetime of the projects regarding whether to adopt or not a particular component, whether to open source it or not and whether to upgrade or not at a particular moment or wait for the next release, etc. In this perspective, we recognize that understanding, managing and mitigating OSS adoption risks is crucial to avoid potentially significant adverse impacts on the business, in terms of time to market, customer satisfaction, revenue and brand image. The objectives that the company would want to achieve through RISCOSS could be:

- To gain full control over the OSS components that are integrated into its products.
- To design a proper business plan and strategy with open source strategic intents to achieve wanted effects. Strategic and business perspectives of open source, including decision processes as well as support for external community activities, are greatly underdeveloped not to say inexistent today.
- To be perceived as an active community leader/opinion maker and regarded as a trusted long term player in the OSS arena.
- To drive and make contribution in a controlled manner as well as support the use of OSS code.

However, This company is managing a complex ecosystem where several questions emerge, e.g.:

- How to design the possible viewpoints from which one can look at an ecosystem in order to collect relevant information for managing the evolution for the OSS products embedded in the companys applications.
- How to secure those specific features of OSS do not harm business strategies and their underlying business models?
- How to implement a systematic approach towards understanding and representing dependencies that involve OSS components for assessing all kinds of risk?

The answer to these questions requires the clear understanding of OSS-based ecosystems from a strategic perspective, with clear identification of relevant strategic dependencies (not just software dependencies) in order to control and mitigate all the risks coming from the adoption of OSS components along the lifetime of the different products being part of the ecosystem.

6 RISCOSS Analytic Process

According to [14] [15] [2], The RISCOSS analytic process is based on a three-layered approach to risk management in OSS projects: the first one for the gathering and aggregation of data both from communities and projects by collecting and summarising available data from different data sources and defining risk drivers. The second layer, for converting these risk drivers into risk indicators [2]. Finally, the third layer for assessing how these risk indicators impact on the fulfilment of the business goals of the adopting organisation [16]. Figure 4 summarize this view.

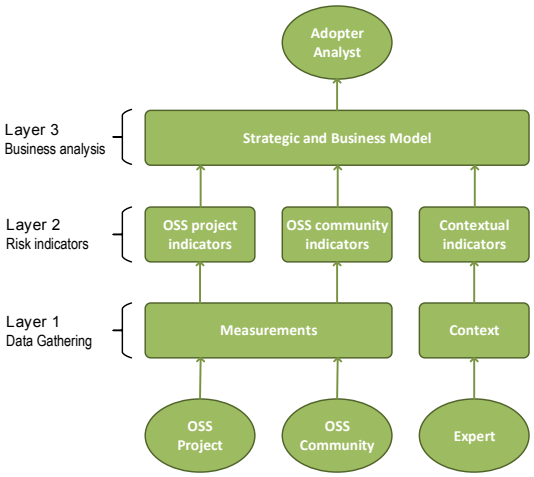

Fig. 4. A red approach for risk assessment in OSS adoption.

6.1 Layer I: Raw Data and Risk Driver Measures

In this layer, we deal with data collected from OSS communities and projects that determines the risk drivers. The data has a twofold nature. On the one side it refers to the characteristics of the OSS components developed by the communities, e.g. *Number of open bugs*, *Mails per day*, *Forums posts per day* [14]. We gathered the measures from the available literature using a Systematic Literature Review (SLR) and statistical expert assessment. On the other hand, measures that highlight the structure of the community in terms of its evolution (e.g. changes in its roles and members and in the quality and quantity of relationships between them) which are obtained mainly via social network analysis techniques [12]. Figure 5 shows an example where centrality measure is visualized . The data sources for these measures are community repositories, versioning system, mail lists, bug tracker and forums and from third-party tools such as FOSSology, Sonar, Antepedia among others [14].

The raw data collected from communities and projects are aggregating into what RISCOSS calls risk drivers. These represent summarized data over a specific time frame. For example, a series of bug report data (raw data) is aggregated into a distribution of the *number of bugs* over a given period of time (risk driver). The corresponding measurement instruments are designed to implement a continuous monitoring process to report data to the statistical and reasoning engines that are used in the other layers. However, human intervention may be eventually needed because of: 1) data sources that may be unavailable for a particular component or community; 2) values that can eventually be calculated but require a dedicated activity to do so (e.g., evaluation of some quality aspect like security or performance); 3) values that are not directly accessible or are very costly to compute (e.g., level of expertise of the organization's OSS development team) [14].

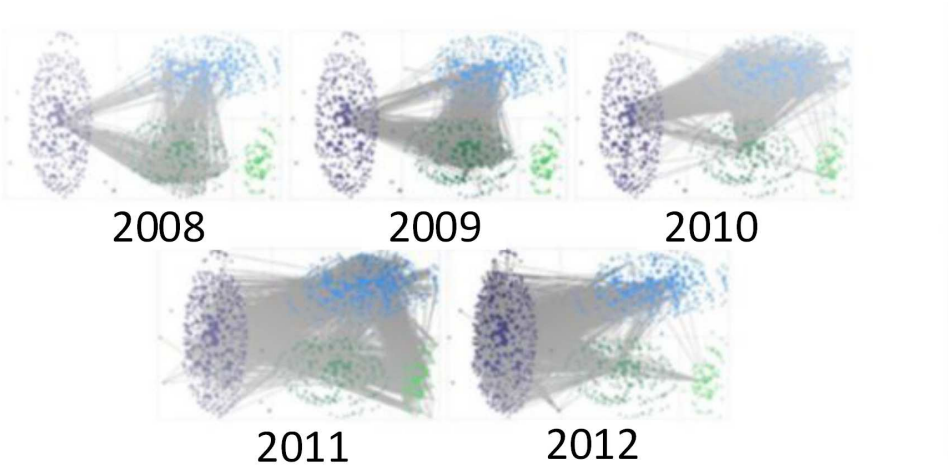

Fig. 5. Social network analysis of the *structure* of the OSS communities and of their *evolution*.

6.2 Layer II: Risk Indicators and Models

In this layer, we define the set of indicators of possible risks and models that allow linking these risks to the possible objectives of the adopting organization. The indicators are variables extracted via the OSS community data analysis obtained from OSS project measurements and OSS community measurements as described before or via expert assessment [14]. Several categories of indicators can be observed e.g.:

- *Project indicators:* Related to the particular OSS project can be grouped following some criteria such as Reliability and Maintainability of the code
- *Community indicators:* Related to the communities that may be extracted thanks to the community measures. These allow to build indicators such as Community activeness, or Community cohesion.
- *Contextual indicators:* Can be that mainly depends on the objectives of the organization, such as OSS business strategy, or the Type of project in which the OSS component hast to be introduced.

Here, statistical analysis, Bayesian networks and social network analysis are exploited to determine risk indicators. In particular:

- Statistical analysis of data from OSS communities allows determining the trends and distributions of data.
- A Bayesian network (BN) is essentially a directed acyclic graph, together with the associated conditional probability distributions. BN are efficient models for (automated) reasoning under uncertainty by fusion of data and domain expert knowledge [17]. In RISCOSS project, Bayesian networks are used to link the community data gathered from the community data sources and the community measures to the risk indicators using data generated by experts assessment based on their experience in OSS adoption and community context. In particular, simulated or real scenarios, described via the values of the OSS measures, are presented to experts and they

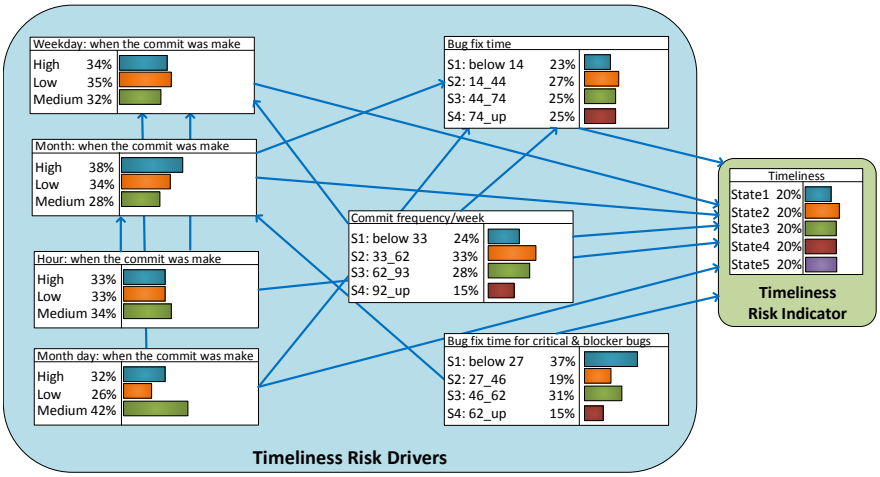

Fig. 6. Example of a Bayesian network data-to-risk linkage.

are asked to rate risk indicators with respect to each scenario [12]. Figure 6 illustrates how the RISCOSS process leverages the Bayesian network approach with a highlight, on the community timeliness risk indicator [16].

6.3 Layer III: Business Goals

The same second layer approach is used in the third layer to derive any potential impact on business risk/goals, mapped from risk indicators through a second type of qualitative validation called the strategic workshop. By linking the resulting distribution of business risk/goals with the different *i** models of the business and software ecosystems, RISCOSS enables adequate business-driven risk management of adopting open source software [14].

Business goals of the organization that adopts OSS are exposed to several types of business risks. In general, they are summarized in four categories:

- *Strategic risks:* Mainly related to the company's strategy and plan, such as Pricing pressure, Failures in comply regulation, industry or sector downturn, or Partner issues.
- *Operational risks:* Such as Poor capacity management or Cost overrun.
- *Financial risks:* Such as Assets lost, Debts or Accounting problems.
- *Hazard risks:* Related to, for example, Macroeconomic conditions or to Political issues.

These generic risks need to be tailored to specific application instances in order to provide added value to decision makers. Business goals are included in models that represent the ecosystem that blends together communities, OSS adopter organizations and other key actors. The key relationships between these actors are represented through dependencies in goal-oriented models expressed in i* Reasoning is based upon different techniques, and in our layered context, we are particularly interested in bottom-up evaluation, since the leaves are directly linked to the risk model.

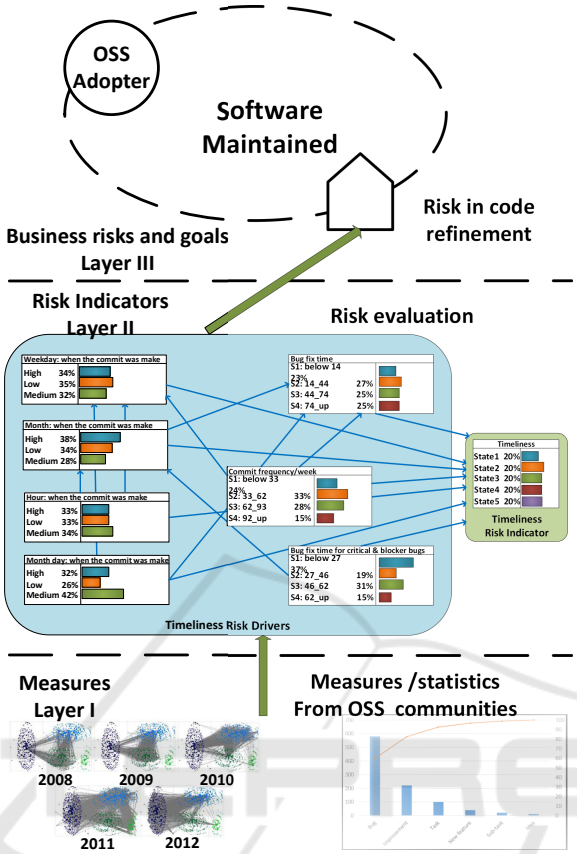

Fig. 7. The 3-layers: in layer i the measures from the oss project and communities, in layer ii the bayesian network to identify the business risks and in layer iii the buisness goals.

6.4 An Integrated View

Figure 7 provides an integrated view that synchronizes the three different layers. In Layer I the raw measures are retrieved and the basic distributions of these data are computed. Layer II will use the Bayesian networks that exploits the measures from Layer I and produces the evidence of the existence of business risks. Finally, in Layer III the Business risks are connected to the Business goals of the organization represented via the i* diagrams. The overall strategy of the RISCOSS work plan is described in Figure 8 [18].

7 The RISCOSS Platform

This section introduces the architecture of the RISCOSS platform. The platform is the embodiment of the methodology into a open source software. Software implementation of the methodology is designed to support the whole process, from the collection of

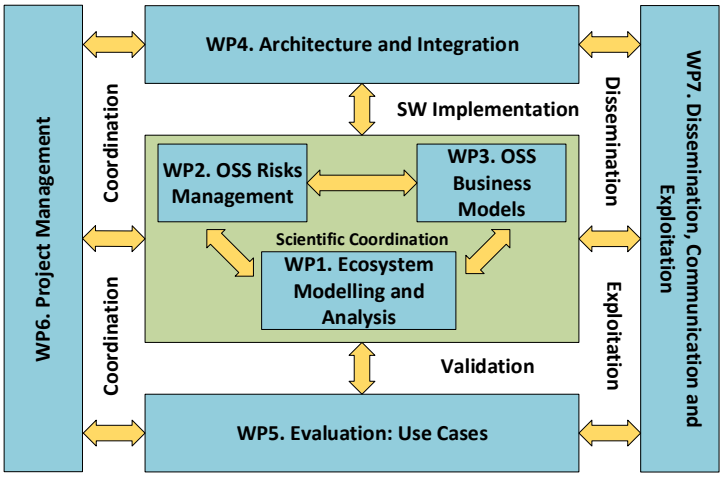

Fig. 8. Overall methodology of the RISCOSS work plan.

open source projects raw data to the provision of decision guidelines for a given adopter. The platform is a web application available for download and that can be deployed in-house or accessed as an online decision support service.

7.1 Platform Scope

The RISCOSS platform is intended to be operated within an ecosystem of resources and stakeholders. RISCOSS takes into consideration the perspective of the project ecosystem, including communities of developers and contributors and that of the adopter, most likely to be a company looking to integrate an OSS component into a product or service.

The major functions include modeling of the ecosystems and the risk profile of the adopter organization, collecting relevant data from the projects, processing models and data by applying innovative statistical and risk processing techniques and then delivering results to users in a useful manner and providing support for selecting between options.

The platform is highly flexible and can be customized depending on user profiles, preferences and business conditions. It draws data from tools used by open source communities of developers and contributors, leverages data analysis tools and takes into account the behavior of community members and users. Tools used by communities to manage their projects are, with some exceptions, publicly accessible. Users of the RISCOSS platform are development teams within companies. The system they use to manage their projects and activities can be considered functionally equivalent to those found in open source communities, with the difference that they are not accessible from outside the company. The RISCOSS Platform provides the interfaces to interact with these systems and to provide outputs to users looking to run risk analysis in a context of open source component adoption.

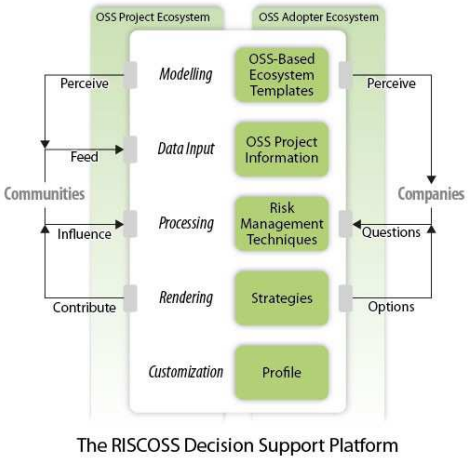

Fig. 9. Architecture overview of the RISCOSS platform.

7.2 Platform Architecture

In this section we describe the architecture of the RISCOSS Platform. The RISCOSS Decision Support platform comprises an extensible application that runs on the XWiki platform. The RISCOSS application is designed to be extensible. In its current version, it is comprised of a number of functional components. There are two main families of components: those that are used across the whole platform and those that are used within a domain. Figure 9 presents the overall architecture and all its major components [11].

Platform Components. These components provide services to all users and all domains of the platform whatever the configuration. On the left side there's everything that exists and that is used by Open Source communities to manage their projects and activities. Usually what we find here is publicly accessible (with some exceptions). In the bottom we find companies, and in general users of the RISCOSS Platform, with the systems they use to manage their project and activities. This can be considered functionally equivalent to what we find in the Open Source communities with the difference that they are not accessible from the outside of the company. In the middle we find the RISCOSS Platform that can interact with these systems and provide functionalities to users in order to perform risk analysis of OSS component adoption. The architecture comprises the following components:

Domain Manager. This components the central element of the RISCOSS Platform; it provides isolation for activities performed in the RISCOSS platform. A domain is a container for all the data related to a given context. An entity using the RISCOSS platform will have its own domain isolated from the others. In each domain, users can specify their own models, layers and roles, and store relevant data for performing specific or customized risk analysis, for example, depending on certain business models or providing certain risk reports. A benefit of the RISCOSS architecture with domains and

Domain Manager is that it allows for interesting deployment alternatives. For example, a service provider can deploy a RISCOSS-as-a-Service platform where it manages different domains for different customers on a single RISCOSS platform while a company that wants to use RISCOSS in isolation will just create a single domain on its privately deployed RISCOSS platform. A third deployment option could be for an open source organization such as the Apache Software Foundation or the OW2 Consortium; in this case, the RISCOSS platform is completely public and each open source project is allocated its own domain where it publishes its own data.

Risk Data Collectors. Risk Data Collectors are stand-alone components able to collect raw data, which RISCOSS calls Risk Data, from a variety of relevant sources. Their task is to post process this Risk Data and prepare it for consumption by the RISCOSS platform. The component is partly outside the RISCOSS platform because it can be controlled by users foreign to the RISCOSS platform. For example, imagine a Risk Data Collector that is able to aggregate data on the distribution of bug-fixing time. In one scenario this component could also be run by an open source community willing to provide this kind of information periodically to a public installation of a RISCOSS platform.

Data Collector Manager. This component manages the Risk Data Collectors available in the platform. Via an API, it provides information on what is available, and the parameters they require. It also manages, for example, the execution, the scheduling of the Risk Data Collectors. The Risk Data Collector Manager also provides the means to link the "entities" created in the Domain Manager (see below) to the data collected by Risk Data Collectors. When the User Layer Manager (see below) creates a component, it is The Risk Data Collector Manager that asks the user how to configure the Risk Data Collectors that must be run to collect Risk Data for that entity. The Risk Data Collector Manager triggers the data collection process and associates the collected data to the right entity in the Risk Data Repository.

Risk Data Repository. This component provides storage and query facilities for storing and retrieving Risk Data. Risk Data Collectors use the exposed APIs to send and store the extracted Risk Data into the RISCOSS platform. It is possible to have multiple Risk Data Repositories deployed within the RISCOSS Platform. Some can store publicly accessible data, while others can be used to store private data accessible only from particular contexts. The former is useful to build a public knowledge base while the latter is necessary for using RISCOSS in the context of a company, for example, where some data must be protected. The Risk Data Repository also allows the user to query historical data in order to understand how risk data evolve overtime.

Risk Analysis Engine. This component contains all the logic for performing the risk analysis using the data available in a given domain. This component also include, for example, business goal impact analysis. In essence, this component provides all the intelligence for computing everything made possible by the RISCOSS Methodology. The Risk Analysis Engine consists in an API that allows configuring and launching the risk analysis and retrieving the results. It loads risk models forming the knowledge base of the engine.

Authorization and Authentication. This component provides authorization and authentication functionalities to the RISCOSS platform. The Domain Manager uses it in order to grant access or not to the resources stored in a domain to a given user. The Authorization and Authentication component is also used within a domain in order to define the access rules to the data stored there. This can be useful within a company, for instance, where not all personnel can access all information.

Event Notification System. This component provides the functionalities for tracking activities and sending notifications when particular events occur, for example, when a new risk report is created. Events can be notified to users either via the RISCOSS platform user interface or email.

Domain Manager Components. The Domain Manager is a macro-component that manages all data to be manipulated in a given domain. It comprises the following subcomponents:

Layer Manager. This component provides the means to define and manage which layers are actually present in a given domain. In particular it provides a way to create a hierarchy of layers, and define the structure of an entity belonging to a layer. A layer in this context, represents the type of a set of entities (e.g., OSS Components, Products, Projects, etc., and the relationships among them).

Role Manager. A role defines a class of users, and is used to define the rules for accessing data, and functionalities. This component provides the means to define and manage which roles are actually present in a given domain. A role defines a class of users, and is used to define the rules for accessing data and functionalities.

Risk Analysis Manager. This component provides the means to define and enforce the process for performing risk analysis in a domain. For example, the Risk Analysis Manager handles controlling and providing information on the status of a risk analysis session in which many users are participating, according to a specific workflow. The Risk Analysis Manager also handles interactions with the Risk Analysis Engine and required interactions with users via the user interface. These interactions can include, for example, requests for further information on certain aspects of the analysis.

Risk Configuration Manager. The Risk Configuration Manager provides the means to manage all the artefacts needed for a risk analysis. This component takes into account, and organizes rationally all the artefacts' needed by the Risk Analysis Engine, as described in the RISCOSS Methodology. This includes, for example, goal models, impact models, etc.

Risk Report Manager. This component provides the means to manage the results of Risk Analysis performed in a given domain.

Form/Questionnaire Manager. This component provides the means to define and manage user input forms that help users insert information that is needed by the Risk Analysis Engine. Some of the information can come directly from data previously stored in a Risk Data Repository, but in general, users may be required to manually enter some data - either because it's not possible to retrieve it using Risk Data Collectors or because they want to override some previously collected data. The Form/Questionnaire Manager can also be used to create custom-made forms to support qualitative data collection from field experts.

Feedback Manager. This component provide the means to manage user feedback on data managed in a domain. Such feedback can be provided, for example, on risk analysis reports. Feedback is then used to improve the models used by the Risk Analysis Engine for performing the risk analysis

8 Current Implementation

In this section we shows some screenshots of the current state of the RISCOSS platform implementation. Figure 10 shows the main page of a domain. A domain is a container for all the data related to a given context. An entity using the RISCOSS platform will have its own domain isolated from the others. In each domain, users can specify his own models, layers, roles, and store relevant data for performing risk analysis (e.g., business models, risk reports, etc.). Figure 11 shows the layer manager menu with the functionalities for creating new layers, entities and for displaying the existing layers and entities defined in the system. A layer in this context, represents the type of a set of entities (e.g., OSS components, products, projects, etc., and the relationships among them). Figure 12 shows the risk analysis session definitions. This component provides the means to define and perform risk analysis sessions on an entity in a given domain. A risk analysis session is started by choosing the corresponding item in the risk analysis menu. At this point the user is show a list of entities with the available risk configurations. He can the select which entity to analyse with which configuration. Once this is done, the user can start the evaluation. At this point the RISCOSS Platform will invoke the Risk Analysis Engine to perform the evaluation and presents the results as shown in Figure 13. The results shows the evidences of risk exposure in the form of positive and negative evidences. Depending on the risk model a different representation of the results might be displayed, as shown in Figure 14.

9 Conclusions

The RISCOSS platform is architected from the ground up to help bring the benefits of open source to the entire enterprise development department. RISCOSS not only enables users to collect informed intelligence on open source components, but also goes one step further by offering risk analysis that adapts to individual business situations.

Fig. 10. Domain main page.

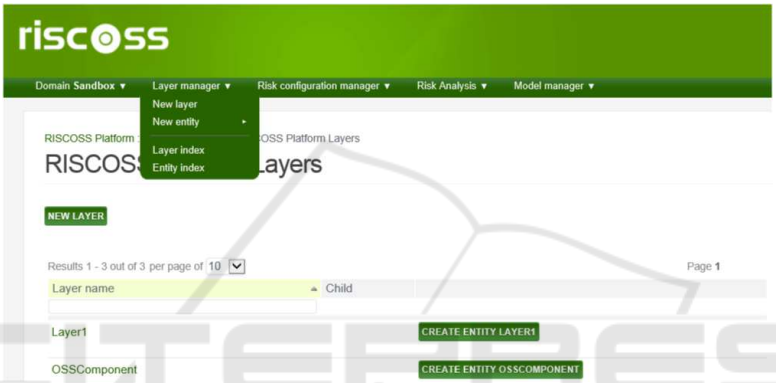

Fig. 11. The layer design window.

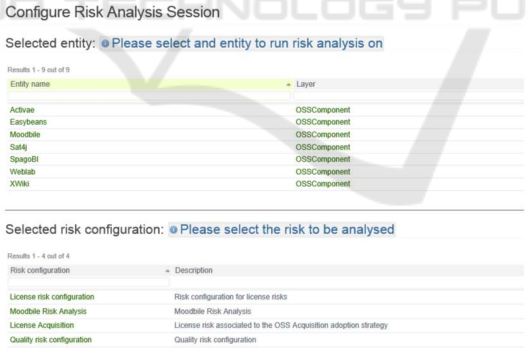

Fig. 12. Risk Analysis Session definitions.

Based on a unique methodology specifically designed to model business and software ecosystems in the world of open source software, and an extensive architecture able to integrate entire portfolios of dedicated functional components such as the Risk Data Collectors, RISCOSS is the only platform to deliver a complete solution rather than a piecemeal approach enabling mainstream product developers to safely integrate open

Fig. 13. Risk Analysis Session results.

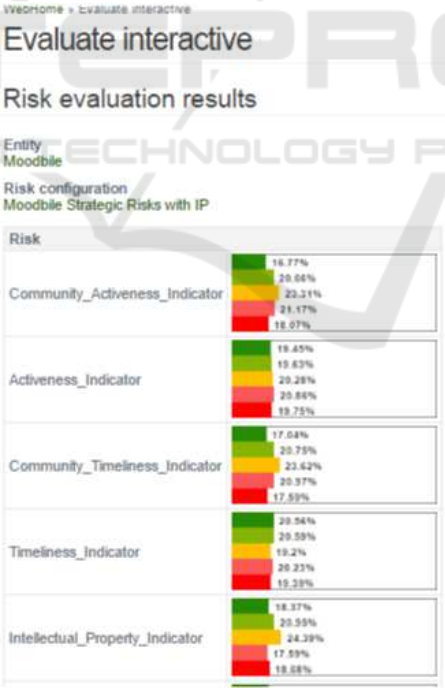

Fig. 14. Risk Analysis Session results.

source software in their developments.

RISCOSS is a research-oriented project that is set to become a reference solution in facilitating industry-wide open source adoption. An open source project itself, RISCOSS is open to third-party contributions to help the platform grow in functionalities and make the transition to a fully marketable product or service.

RISCOSS support the industry ability to correctly manage OSS asoption risks. It is a very important challenge for an OSS community and the earliest these risks are identified and quantified, the better a fighting chance the community will have to guarantee the survival and the success of their product.

Download RISCOSS and join the community at www.riscoss.eu.

10 RISCOSS Related Scientific Publications

1. Ayala, C., Franch, X., López, L., Morandini, M., Susi, A.: Using i* to represent OSS ecosystems for risk assessment. In: Proceedings of the 6th International i* Workshop in CAISE'2013. (2013)
2. Costal Costa, D., Gross, D., López Cuesta, L., Morandini, M., Siena, A., Susi, A.: Quantifying the impact of OSS adoption risks with the help of i* models. In: Proceedings of the 7th International (iStar) Workshop,iSTAR'2014. (2014)
3. Franch, X., Kenett, R., Mancinelli, F., Susi, A., Ameller, D., Ben-Jacob, R., Siena, A.: A layered approach to managing risks in OSS projects. In: Proceedings of the Open Source Software: Mobile Open Source Technologies, Springer (2014), 168-171
4. Franch, X., Susi, A., Annosi, M.C., Ayala, C.P., Glott, R., Gross, D., Kenett, R., Mancinelli, F., Ramsamy, P., Thomas, C., et al.: Managing risk in open source software adoption. In: Proceedings of the 8th International Conference in Software Technologies ICSOFT 2013. (2013) 258-264
5. Franco-Bedoya, O., Ameller, D., Costal, D., Franch, X: QuESo a quality model for open source software ecosystems. In: Proceedings of the 9th International Conference on Software Engineering and Applications,ICSOFT-EA 2014, (2014), 209-221
6. Franco-Bedoya, O., Costal, D., Hidalgo, S., Ben-Jacob, R.: Expert mining for evaluating risk indicators scenarios. In: Proceedings of the 38th Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE , (2014), 205-210
7. Galanis, N., Casany, M., Alier, M., Mayol, E.: Building a community: the Moodbile perspective. In: Proceedings of the 38th Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE, (2014), 211-216
8. Kenett, R., Franch, X., Susi, A. & Galanis, N.: Adoption of Free Libre Open Source Software (FLOSS): a risk management perspective. In: Proceedings of the 38th Computer Software and Applications Conference (COMPSAC), 2014, (2014), 171-180
9. López, L., Franch, X.: Applying Business Strategy Models. In: Organizations Proceedings of the 7th International (iStar) Workshop, iSTAR'14, (2014)

10. López, L., Costal, D., Ayala, C., Franch, X., Glott, R., Haaland, K.: Modelling and applying oss adoption strategies. In Yu, E., Dobbie, G., Jarke, M., Purao, S., eds.: Conceptual modeling. Volume 8824 of lecture notes in computer science. Springer International Publishing (2014) 349-362
11. Morandini, M., Siena, A., Susi, A.: Risk awareness in open source component selection. In: Abramowicz, W., Kokkinaki, A., eds.: Business Information Systems. Volume 176 of Lecture Notes in Business Information Processing. Springer International Publishing (2014), 241-252
12. Morandini, M., Siena, A., Susi, A.: A Context-specific definition of risk for enterprise-level decision making. In: Proceedings of the 8th International Workshop on Value Modeling and Business Ontology (VMBO 2014), (2014)
13. Oriol, M., Franco-Bedoya, O., Franch, X., Marco, J.: Assessing open source communities' health using Service Oriented Computing concepts. In: Proceedings of the 8th Research Challenges in Information& Science (RCIS), 2014 IEEE, (2014)
14. Siena, A., Morandini, M., Susi, A.: Modelling risks in open source software component selection. In Yu, E., Dobbie, G., Jarke, M., Purao, S., eds.: Conceptual Modeling. Volume 8824 of Lecture Notes in Computer Science. Springer International Publishing (2014) 335-348
15. Vergne, M., Susi, A.: Expert finding using markov networks in open source communities. In Jarke, M., Mylopoulos, J., Quix, C., Rolland, C., Manolopoulos, Y., Mouratidis, H., Horkoff, J., eds.: Advanced Information Systems Engineering. Volume 8484 of Lecture Notes in Computer Science. Springer International Publishing (2014) 196-210
16. Yahav, I., Kenett, R., Bai, X.: Risk based testing of Open Source Software (OSS). In: Proceedings of the 38th Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE, 2014, 638-643

Acknowledgements

This work is a result of the RISCOSS project, funded by the EC 7th Framework Programme FP7/2007-2013 under the agreement number 318249. We would also like to thank the contribution of EOSSAC project, founded by the Ministry of Economy and Competitiveness of the Spanish government (TIN2013-44641-P).

References

1. Driver, M.: Hype cycle for open-source software. Technical report, Gartner Group (October-2013)
2. Franch, X., Susi, A., Annosi, M.C., Ayala, C.P., Glott, R., Gross, D., Kenett, R., Mancinelli, F., Ramsamy, P., Thomas, C., et al.: Managing risk in open source software adoption. In: Proceedings of the 8th International Conference in Software Technologies ICSOFT 2013. (2013) 258–264
3. Driver, M.: Critical strategies to manage risk and maximize business value of open source in the enterprise. Technical report, Gartner Group (June-2011)
4. Driver, M.: Five mistakes to avoid when implementing open-source software. Technical report, Gartner Group (November-2011)

5. Kenett, R.S., Raanan, Y.: Operational Risk Management: A Practical Approach to Intelligent Data Analysis. Wiley (2010)
6. Moore, J.F.: Predators and prey: a new ecology of competition. Harvard business review 71 (1993) 75–86
7. Messerschmitt, D.G., Szyperski, C.: Software Ecosystem: Understanding an Indispensable Technology and Industry. MIT Press Books. The MIT Press (2005)
8. Bosch, J.: From software product lines to software ecosystems. In: Proceedings of the 13th International Software Product Line Conference. SPLC '09, Pittsburgh, PA, USA, Carnegie Mellon University (2009) 111–119
9. Qualipso: http://www.qualipso.org, last visited march 14th (2013)
10. Helander, N., Rissanen, T.: Value-creating networks approach to open source software business models. Frontiers of E-Business Research 2005 (2005) 840–854
11. Franch, X.: An overview of the riscoss decision support platform methodology and architecture. Technical report, RISCOSS Project (2014)
12. Ayala, C., Franch, X., Lpez, L., Morandini, M., Susi, A.: Using i* to represent oss ecosystems for risk assessment. In: Proceedings of the 6th International i* Workshop in CAISE'2013. (2013)
13. Yu, E.: Modeling Strategic Relationships for Process Re-Engineering. PhD thesis, Department of Computer Science,University of Toronto. (1995)
14. Kenett, R., Franch, X., Susi, A., Galanis, N.: Adoption of free libre open source software (floss): A risk management perspective. In: Proceedings of the 38th Computer Software and Applications Conference (COMPSAC), 2014. (2014) 171–180
15. Franch, X., Kenett, R., Mancinelli, F., Susi, A., Ameller, D., Ben-Jacob, R., Siena, A.: A layered approach to managing risks in oss projects. In: Proceedings of the Open Source Software: Mobile Open Source Technologies, Springer (2014) 168–171
16. Franco-Bedoya, O., Costal, D., Hidalgo, S., Ben-Jacob, R.: Expert mining for evaluating risk indicators scenarios. In: Proceedings of the 38th Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE. (2014) 205–210
17. Galanis, N., Casany, M., Alier, M., Mayol, E.: Building a community: The moodbile perspective. In: Proceedings of the 38th Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE. (2014) 211–216
18. Consortium, R.: Riscoss: Description of work. Technical report, RISCOSS Project (2012)