Modelling and Analysis of OSS Ecosystems

Xavier Franch, Angelo Susi, Eric S.K. Yu (UPC) (FBK) (UdT)

franch@essi.upc.edu, susi@fbk.eu, eric.yu@utoronto.ca







Agenda

- An introduction to ecosystems
- The case of OSS ecosystems
- Ecosystem modeling with i*
 - Short background on i*
 - Modeling OSS ecosystems with i*
- Ecosystem analysis with i^*
 - i*-based analysis
 - An application: risk analysis in OSS ecosystems
- Final discussion



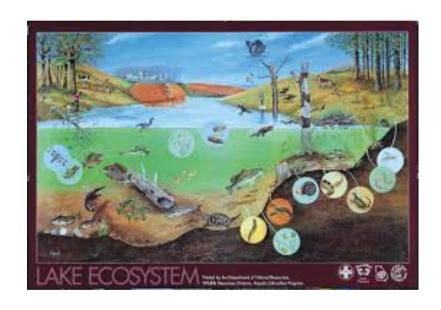
Agenda

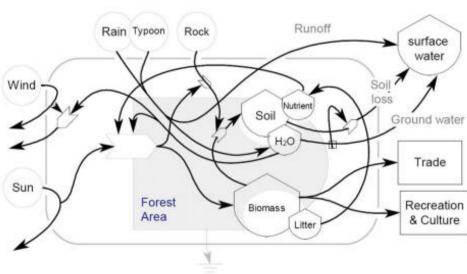
- An introduction to ecosystems
- The case of OSS ecosystems
- Ecosystem modeling with i*
 - Short background on i*
 - Modeling OSS ecosystems with i*
- Ecosystem analysis with i^*
 - i*-based analysis
 - An application: risk analysis in OSS ecosystems
- Final discussion



Ecosystem

A system formed by the interaction of a community of organisms with their environment





What type of organisms and environment





Ecosystems relevant to IS

- Business ecosystem
 - Focus is on organizations that co-create value
- *Software* ecosystem
 - Focus is on the context of software systems
- Not independent



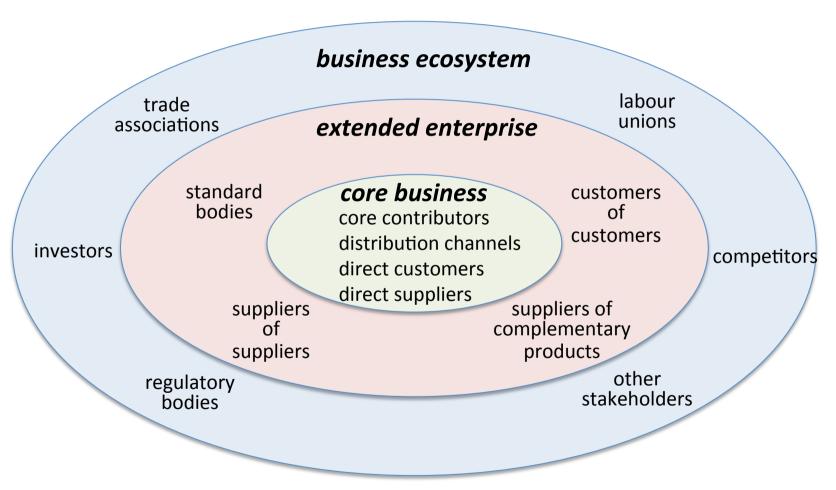
Business ecosystem (BECO)

An *economic community* supported by a foundation of *interacting organizations and individuals* -- the organisms of the business world (Moore, 1993)

- Produces (co-creates) goods and services of value to customers
- Work cooperatively and competitively to support new products, satisfy customer needs, and eventually incorporate the next round of innovations

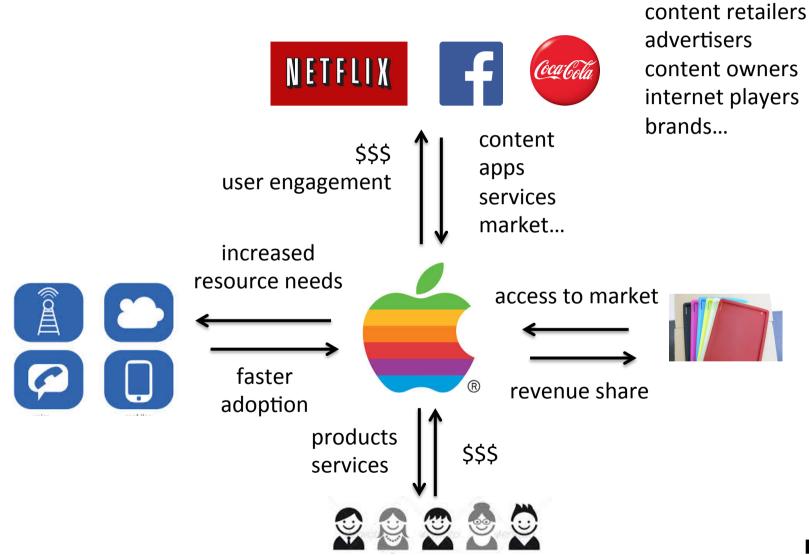
It's competition among business ecosystems, not individual companies, that's largely fueling today's industrial transformation

Business ecosystem actors



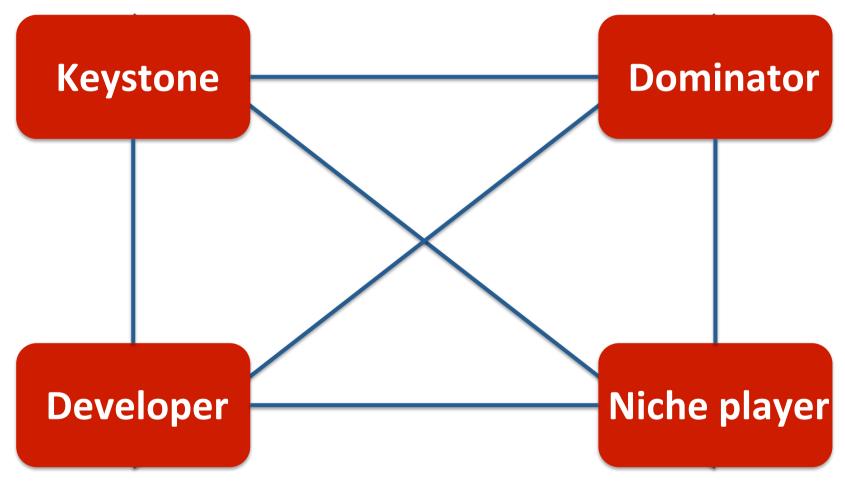


Example





Roles in ecosystems





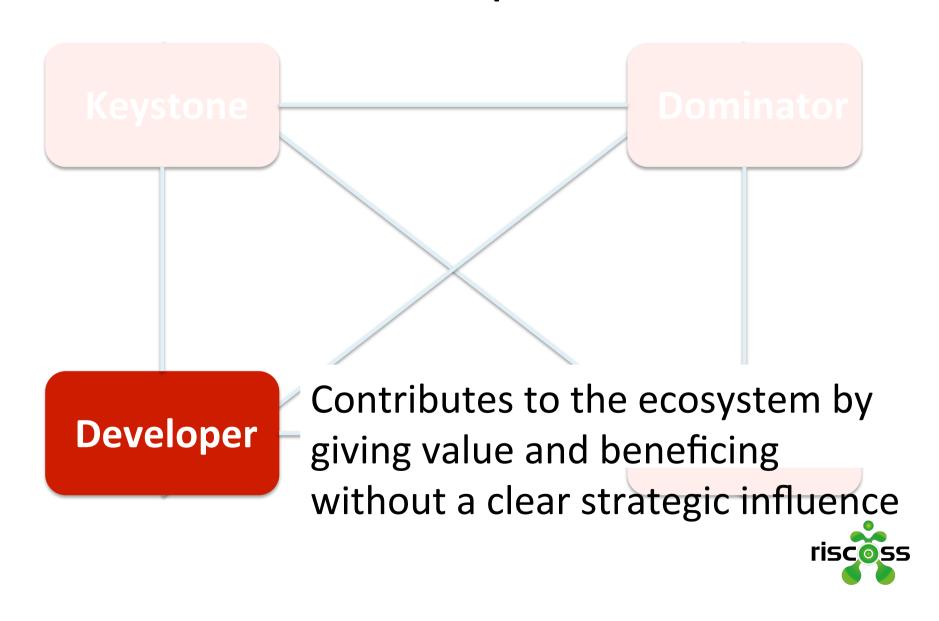
Keystones

Keystone

Improves the health of the ecosystem as a whole acting as regulator

- Boosts productivity of the ecosystem
 - keeps focus; solves problems
 - create technological platforms to be exploited by the rest of the ecosystem
- Ensures sustainability
 - prevents and bridges gaps
 - attract new members and customers

Developer



Dominator

Strongly influences growth directions and business opportunities



- Physical dominator: integrates with the purpose of owning and managing (aggressive)
 - large size and market power; can't change fast
 - blocks and absorbs changes
- Value dominator: creates little value while extracting as much as possible
 - Short-term tactic of maximum value extraction, without attending to ecosystem health
 - Extreme: parasite

Niche players

- Innovation drivers
 - value creation
- Provide complementary assets
 - collaboration, not competition
- Invest in *interactions*
 - positions product as extension vs. standalone

Develops specialized capabilities that differentiate them from others





Software ecosystems (SECO)

Set of actors functioning as a unit and interacting with a shared market for software and services (Jansen and Cusumano, 2013)

A *collection of software projects* which are developed and evolve together *in the same environment* (Lungu et al., 2010)

Collection of organizations that are *related through software or a software related concept* (a standard –
XML, a product -OpenOffice, a hardware –Playstation 3,
a platform –Android)

BECOs and **SECOs**

A BECO is an economic community supported by a foundation of interacting organizations and individuals-- organisms of the business world (Moore, 1993)

A SECO consists of the set of software solutions that enable, support and automate the activities and transactions by the actors in the associated BECO, and the organizations that provide these solutions (Bosch, 2009)

- A SECO is part of a BECO
 - A SECO is not a different ecosystem than a BECO, but it is an ecosystem with additional concepts and consequences
- A BECO talks about organizations and individuals, not about products
- In a SECO, products play a major role as actors in the ecosystem



Agenda

- An introduction to ecosystems
- The case of OSS ecosystems
- Ecosystem modeling with i*
 - Short background on i*
 - Modeling OSS ecosystems with i*
- Ecosystem analysis with i^*
 - i*-based analysis
 - An application: risk analysis in OSS ecosystems
- Final discussion



OSS ecosystems

Ecosystems in which the co-creation of software is basically open but regulated by:

- strategies
- licenses

A community of developers play an important part in the evolution of the ecosystem



Example: Xwiki BECO

- SME deploying the XWiki OSS CMS with several licenses
- Supported by an OSS community (XWiki.org)
- Integrated in the OW2 association
- Strategic partnership with several companies
- Active in FP7 projects (\$\$ + technical excellence)
- Relying in several infrastructure providers
- Organizations providing some kind of support

Example: Xwiki SECO

- Importance of the community (XWiki.org)
- Software development infrastructure: Maven, Eclipse, Git, Jenkins, XWiki, Nexus, JIRA
- Server development infrastructure: Linux Debian, Puppet (configuration), Vserver (virtualization), Nagios (monitoring/reporting)
- Libraries used: dozens (e.g., from Java, Apache, ...)
- Deployment infrastructure: JVM, Tomcat, MySQL, Postgres



Agenda

- An introduction to ecosystems
- The case of OSS ecosystems
- Ecosystem modeling with i*
 - Short background on i*
 - Modeling OSS ecosystems with i*
- Ecosystem analysis with i^*
 - i*-based analysis
 - An application: risk analysis in OSS ecosystems
- Final discussion



Model-based representation of ecosystems

Several approaches for modeling ecosystems:

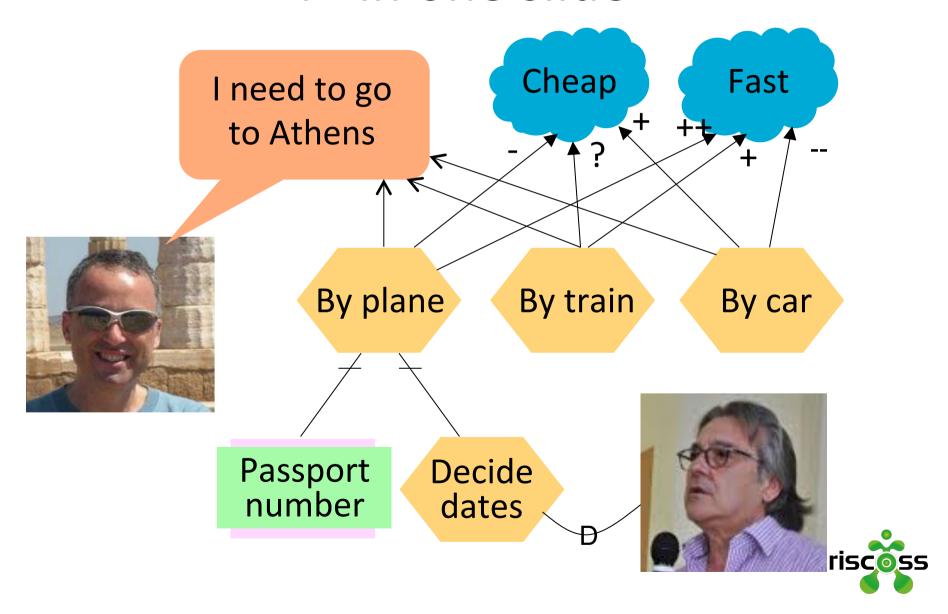
- value model perspective
- software product architecture in the running environment
- quantitative network models
- mathematical models to study particular strategies of software vendors.

What about intentionality and sociality?

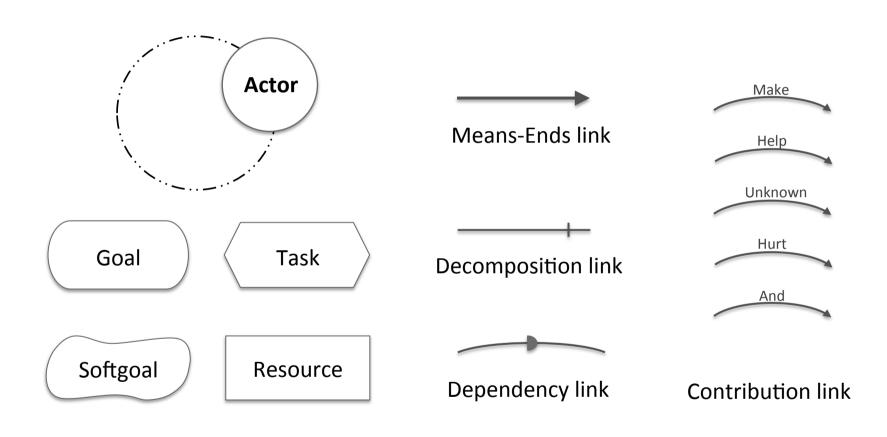
- what are the intentions of the ecosystem actors?
- what are they expectations on each other, and their dependencies?



i* in one slide



*i** quick guide

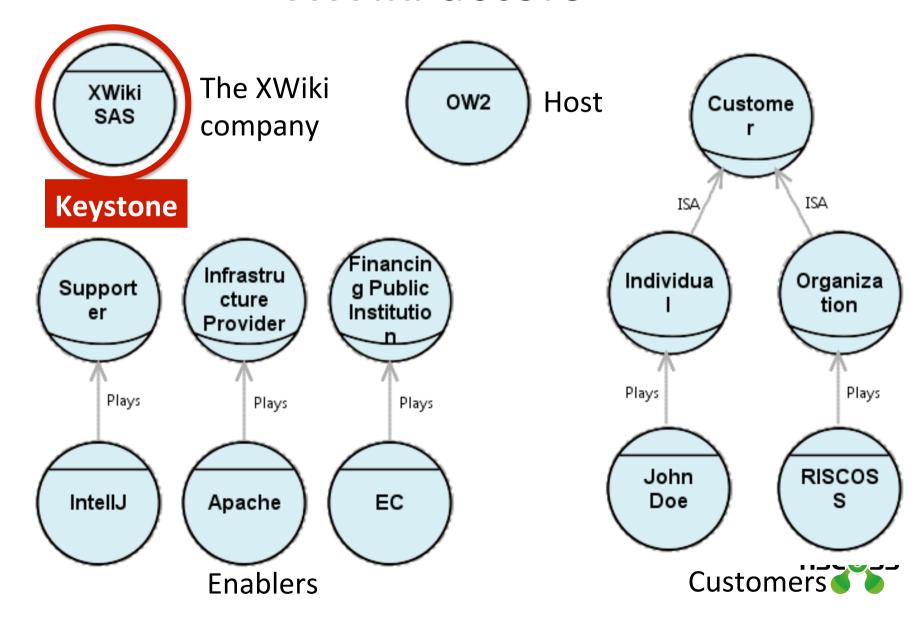


Activities in the modeling process

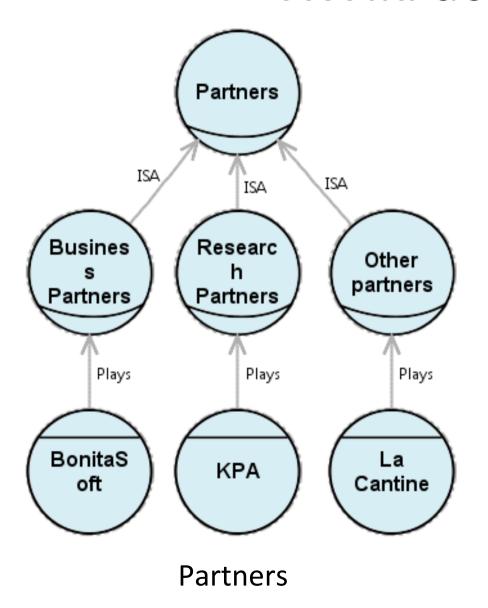
- Identify the main actors in the ecosystem
 - roles and agents
 - others may emerge as we progress
- Identify big groups of dependencies
 - enablers of keystone, their consumers, partnerships, ...

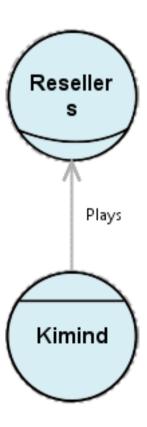


XWiki actors



XWiki actors

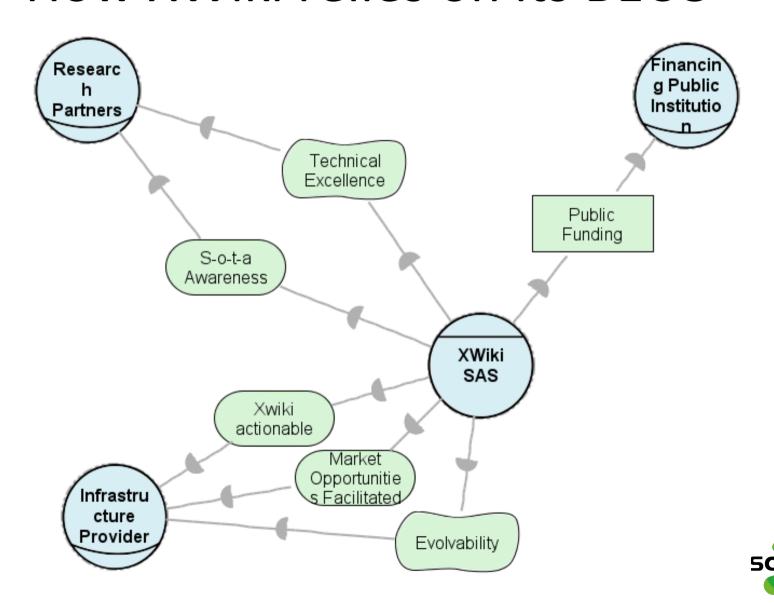




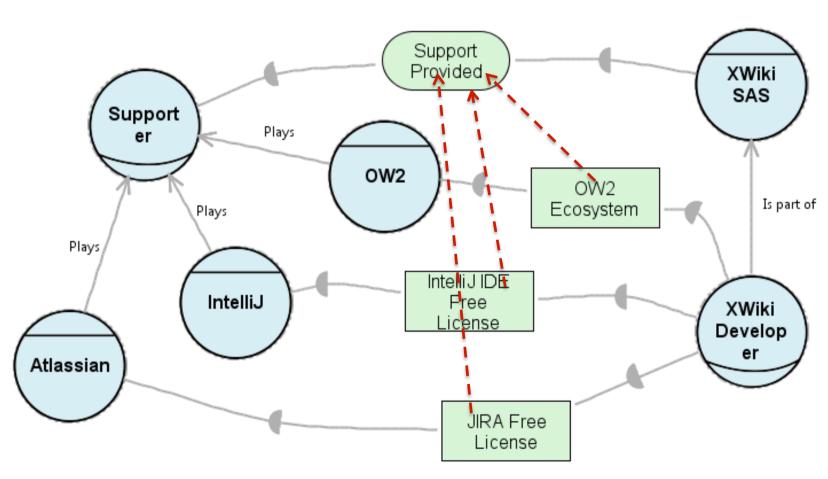
Resellers



How XWiki relies on its BECO

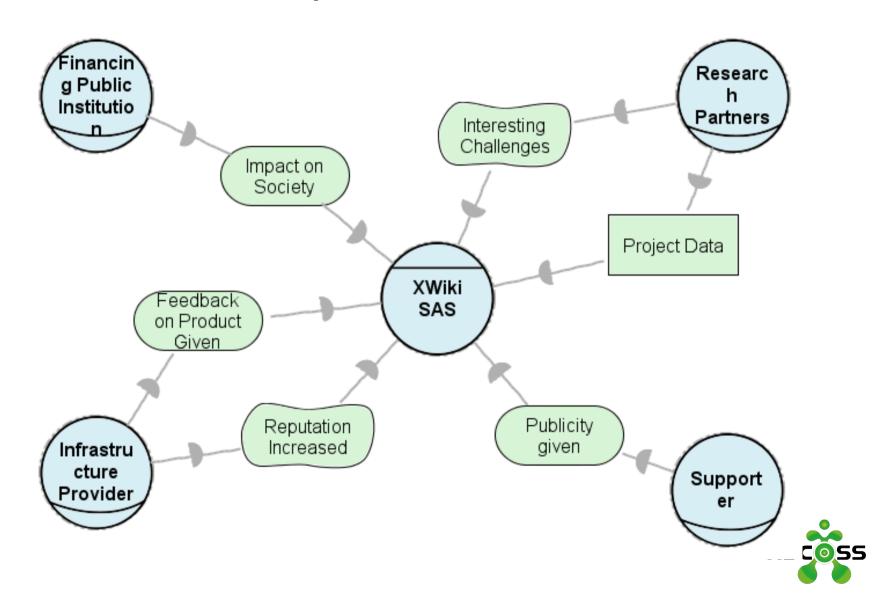


How XWiki relies on its BECO

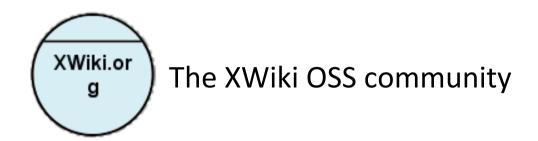




What XWiki provides to its BECO



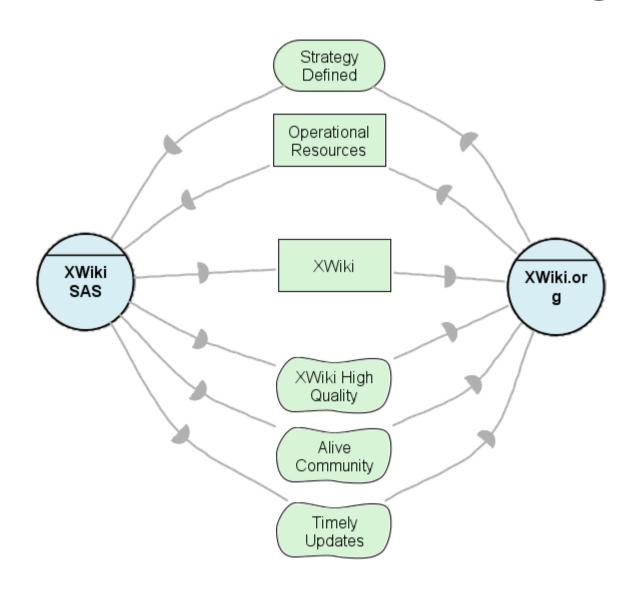
The role of the community



The relationships between the community and XWiki adopters greatly determine the relationships in the ecosystem

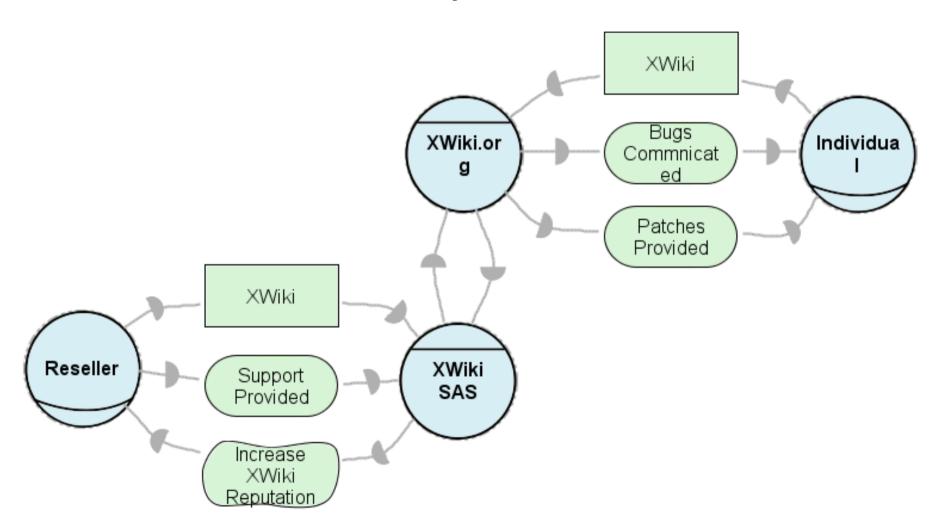


XWiki SAS and XWiki.org





Community in the SECO



Organization adopting XWiki

Let's imagine an organization that needs a CMS

Why it may select XWiki:

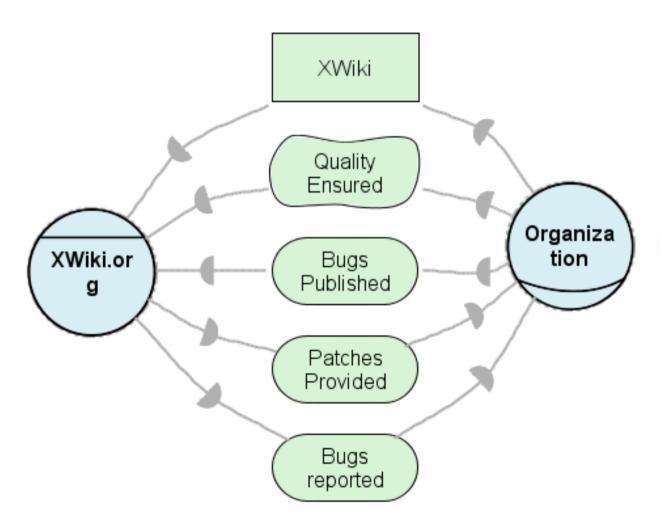
- 1. Best tool in an evaluation process
- Best-fit functionalities
- ...
- 2. Going OSS is a conscious decision
- Reduce costs (development, maintenance, ...)
- Benefit from community work
- "OSS label"
- Strategic move due to context





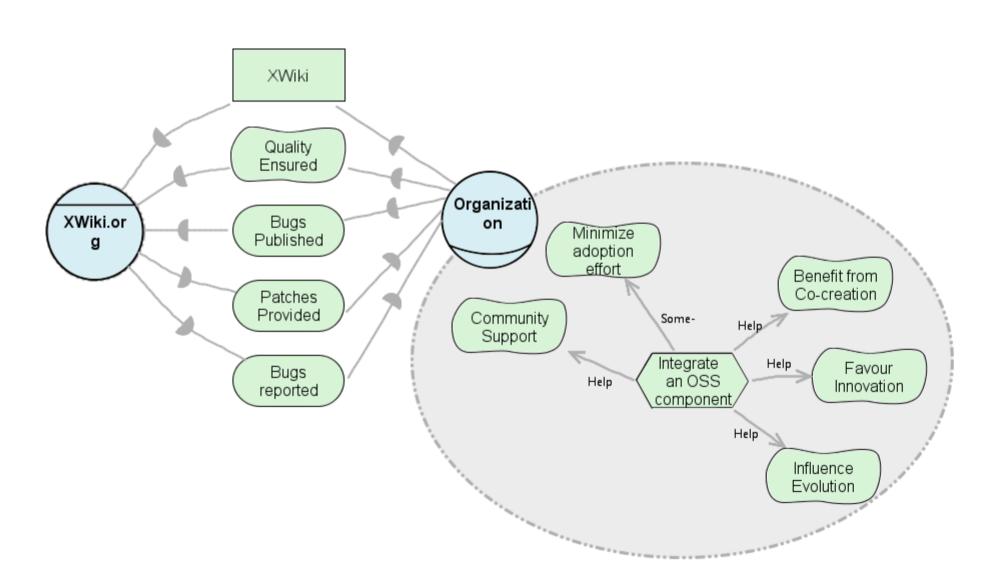
The organization in XWiki's SECO

Assume that it wants to contribute back to XWiki.org

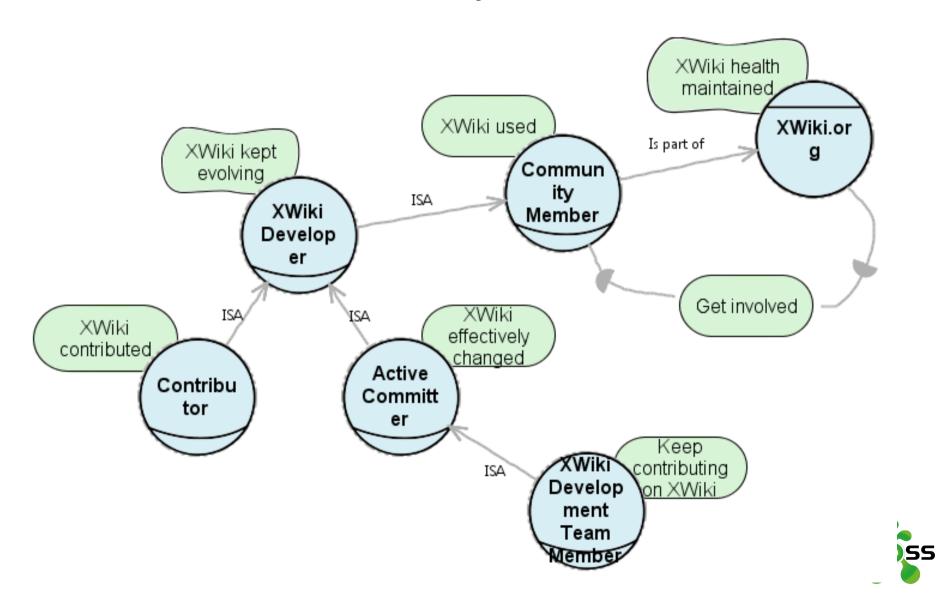




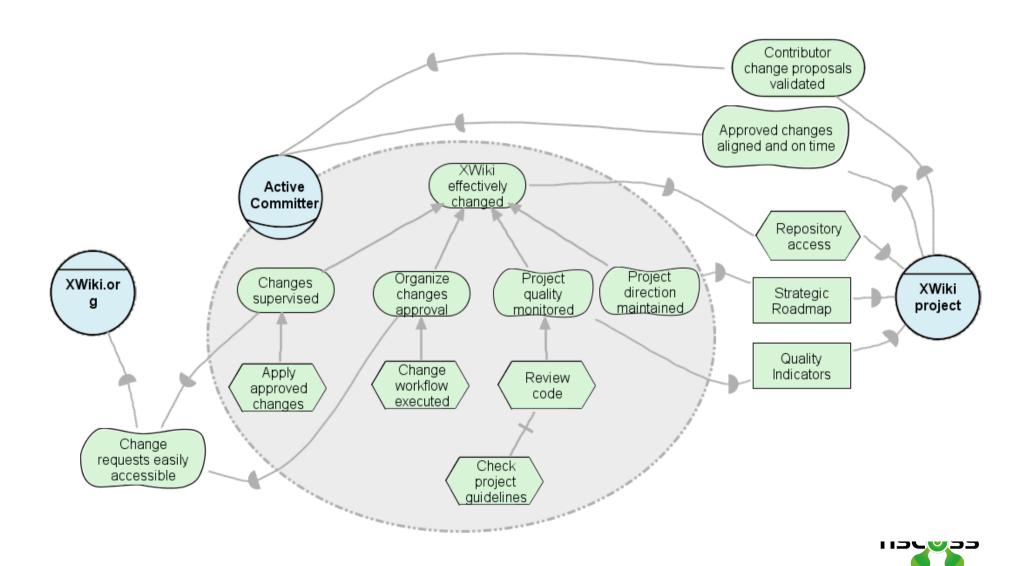
Goals for the organization



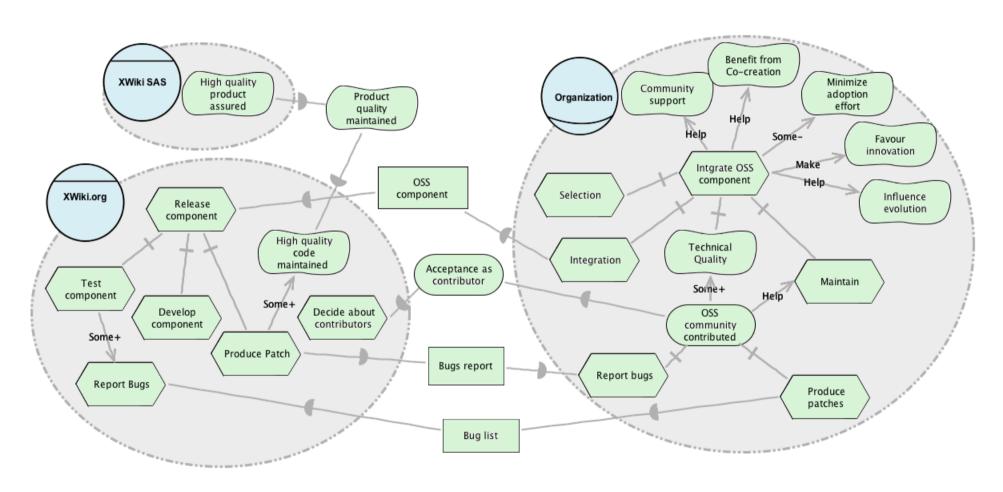
Community structure



Active Committers: Detail

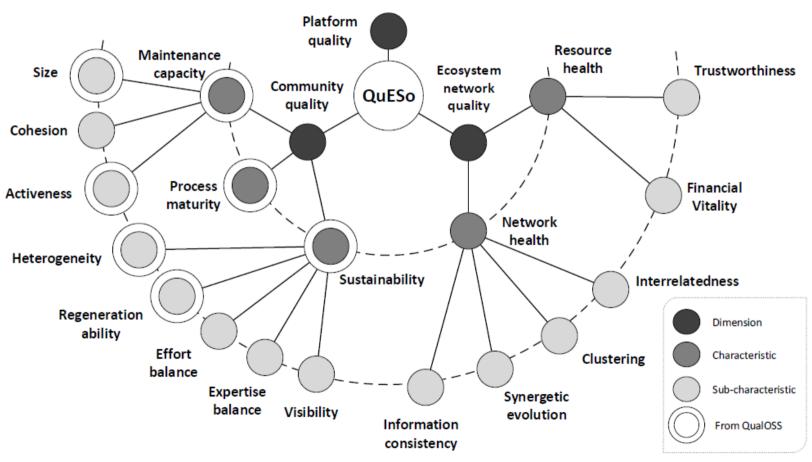


Refining the SECO actors





Ecosystem quality model





OSS Adoption Strategies

- Initiative: start an OSS project and establish a community around it
- Release: code is made public as OSS without a community around
- Acquisition: to use the code without contributing to the OSS project
- Integration: participation in an OSS project to share and co-create
- Fork: to launch a branch of an OSS project
- *Takeover*: to take control over an existing OSS project



Analysis of OSS adoption strategies

	Initiative	Release	Fork	Acquisition	Integration	Takeover
Departing project	No	No	Yes	Yes	Yes	Yes
New project	Yes	Yes	Yes	No	No	No
Effect on community	New	Not formed	New / Split departing	No effect	Enlarge	No effect
Co-creation	Yes	No	Yes	No	Yes	Yes
Influence in the project	Much	No	Much	No	Quite	Maximum

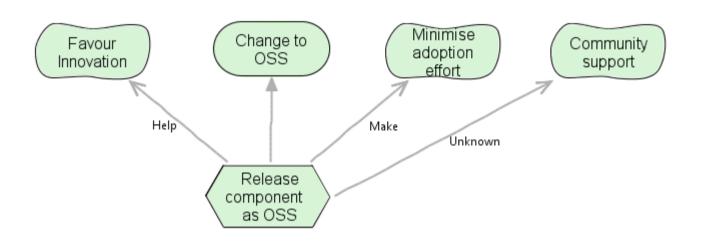


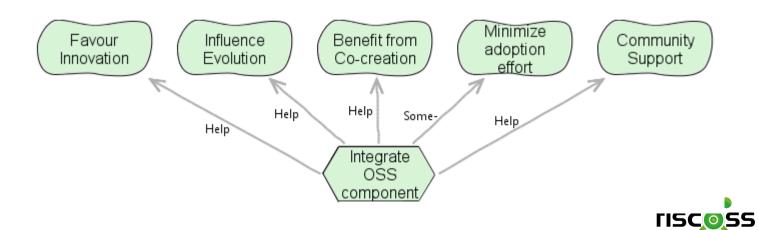
Relation to Business Goals

	Initiative	Release	Fork	Acquisition	Integration	Takeover
Make the change to OSS for a product	Make	Make				
Get new evolution lines for existing software	Some+	Help	Make		Help	Make
Influence on evolution of component	Some+		Some+		Help	Make
Benefit from co- creation	Some+		Some+		Help	Some+
Minimise adoption effort	Hurt	Make	Hurt	Make	Some-	Break
Community support	Make	Unknown	Make		Help	Unknown

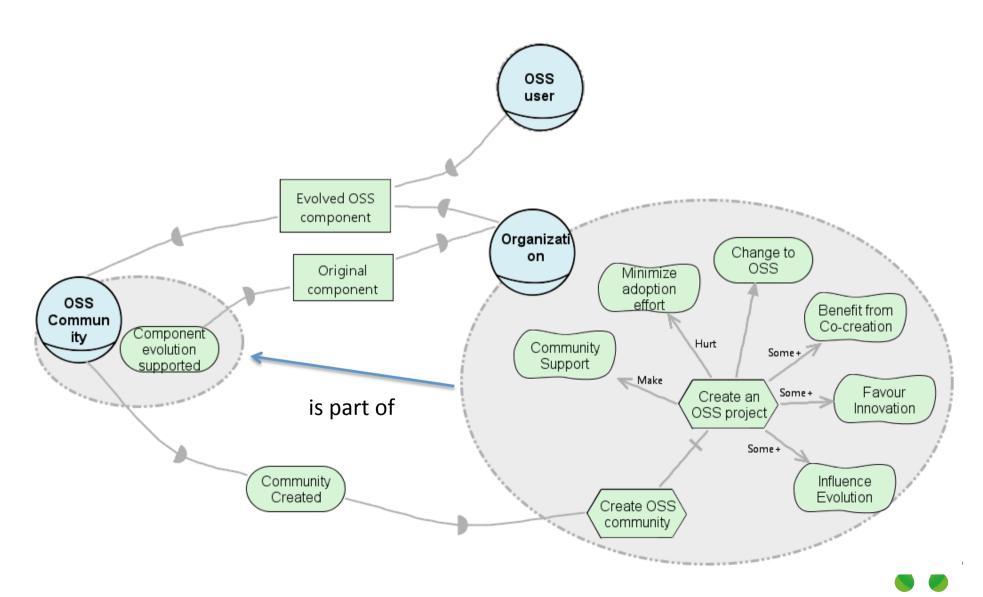


Relation to Business Goals

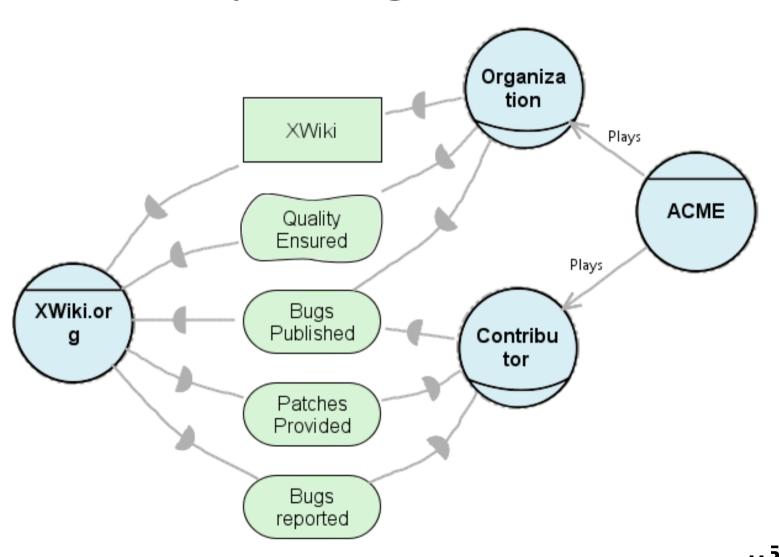




Initiative



Exploiting *i** roles

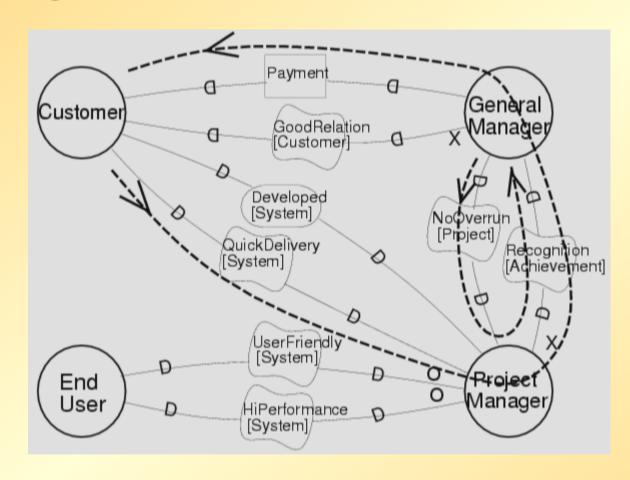


Agenda

- An introduction to ecosystems
- The case of OSS ecosystems
- Ecosystem modeling with i*
 - Short background on i*
 - Modeling OSS ecosystems with i*
- Ecosystem analysis with i^*
 - i*-based analysis
 - An application: risk analysis in OSS ecosystems
- Final discussion

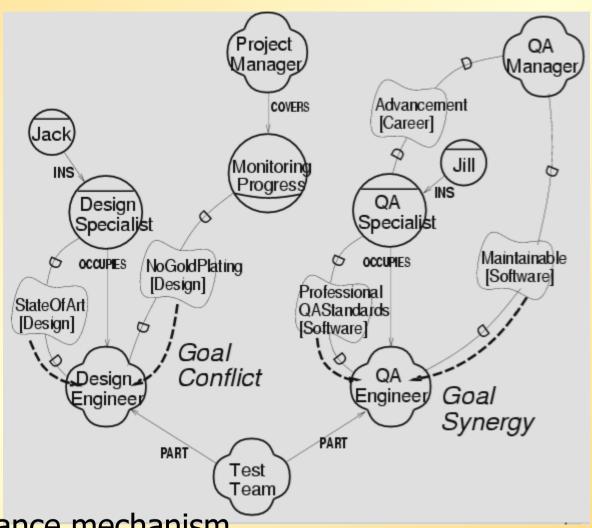


Analyzing vulnerabilities



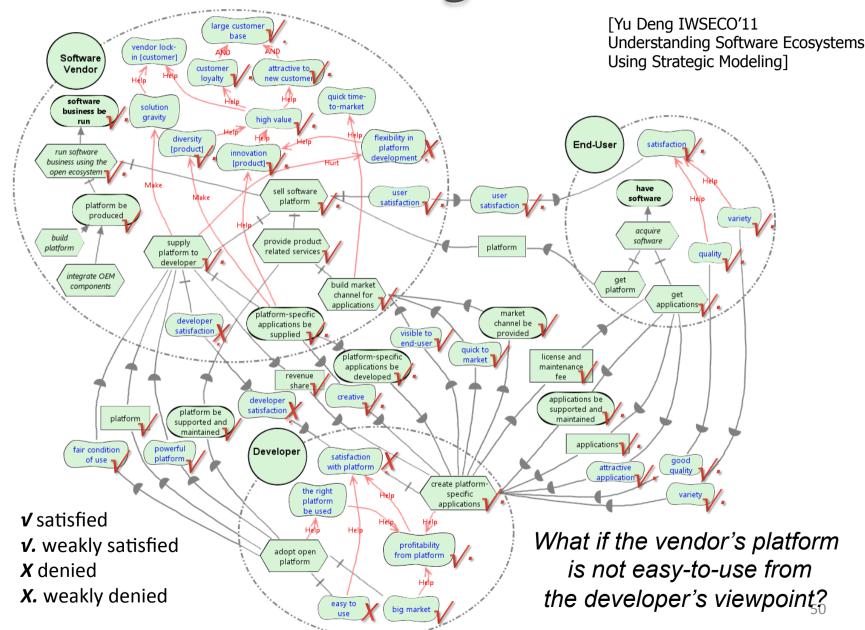
- Example of enforcement mechanism
 - Reciprocal dependency
- Loop analysis

Analyzing vulnerabilities



- Example of assurance mechanism
 - Goal synergy or conflict
- Node analysis

Are Actors' Strategic Goals Met?



Analysis of i^* models

"Qualitative" techniques (e.g., logic based)

 "Quantitative" (e.g., based on propagation algorithms and and feed via statistical evidences...)

RISK ANALYSIS



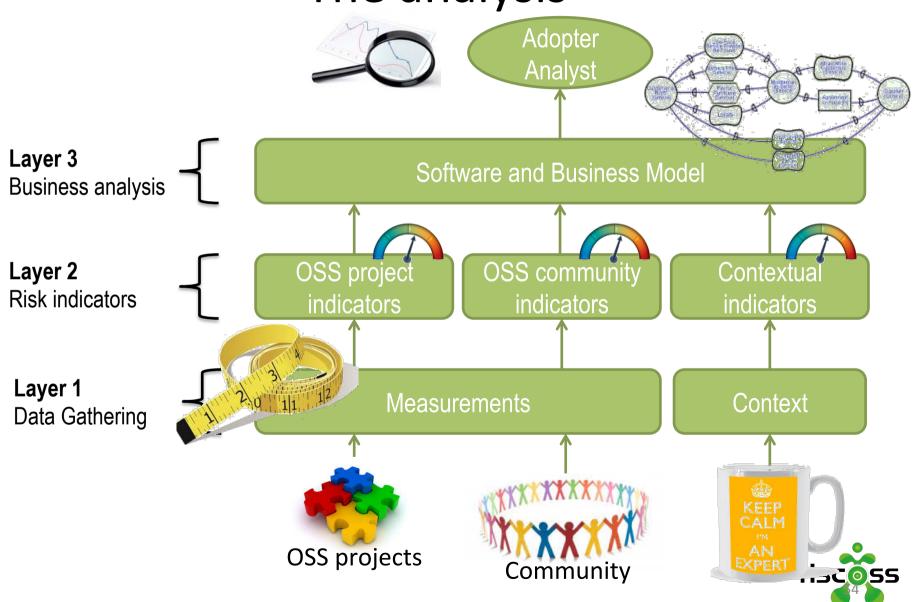
Risk in OSS ecosystem

"Identifying and evaluating the risks of OSS adoption exploiting the information form the OSS strategic and business ecosystems"

- The OSS ecosystem is composed by
 - Adopters (Companies, Public Administrations, OSS communities)
 - OSS communities
- Main steps
 - Modelling risks in the ecosystem via extensions to i^*
 - Reasoning on the models through the reasoning techniques
 - Using data from the "actors" of the OSS ecosystem



The analysis



Examples of OSS adoption Risks

Component selection risks

- Selection effort ill-estimation
- Risk of wrong component selection

Component integration risks

- Integration effort ill-estimation
- Risk of component integration failure
- Security risk

Legal risks

- Intellectual property risk
- Risk of license issues
- Liability risk



Examples of OSS Measures and Risk indicators in OSS ecosystems

Measures

- Long bug fix time: Critical & Blocker
- Long bug fix time: Total
- Commit frequency per week & Number of Commits
- Forum posts per day
- **—** ...

Risk indicators

- Timeliness of the community
- Activeness of the community
- **—** ...



Modeling Risks: entities

- Risk characterized by
 - Event⁽¹⁾; "the OSS component not maintained"
 Situation^(2,3); "the community is not active"

Event Situation

- Measures & Risk Indicators
 - Measure raw and derived evidences;
 "number of bug fixed"

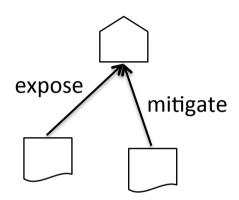
measure □□ & Risk indicator

- 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 *The Practice of Enterprise Modeling*, volume 92 LNBP, pages 82–96. 2011.
- 3. Alberto Siena, Ivan Jureta, Silvia Ingolfo, Angelo Susi, Anna Perini, and John Mylopoulos. Capturing variability of law with Nomos 2. In *ER'12*, LNCS 7532, pages 383–396, 2012.

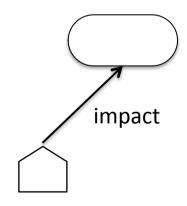


Modeling Risks: relationships

- Relationships between situations and events
 - "expose", "protect"
 Tell when a situation makes it possible (or impossible) an event
 - "increase", "mitigate"
 Tell when a situation makes it critical (or not influential) an event



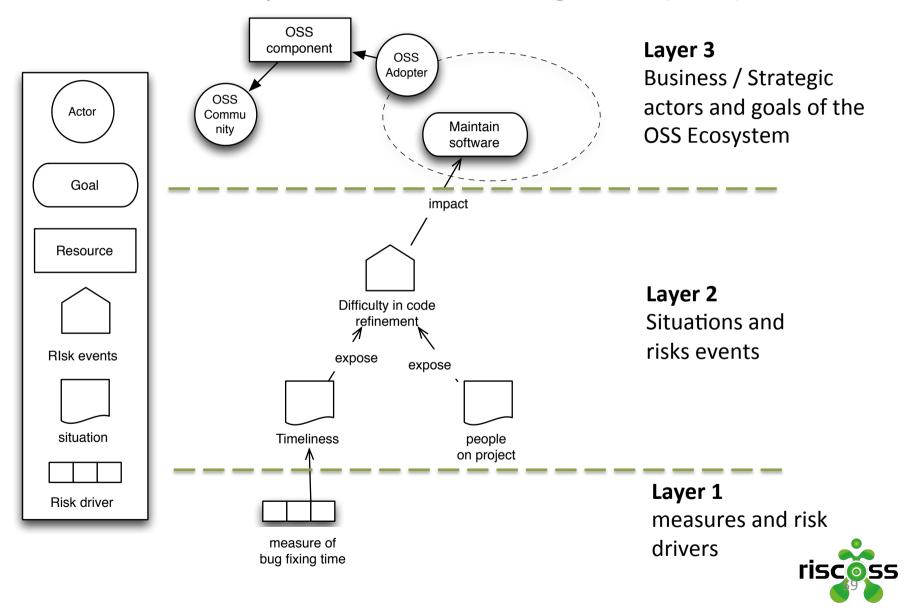
- Relationship between risks events and goals / tasks
 - "Impact" to connect the strategic model with the risk model





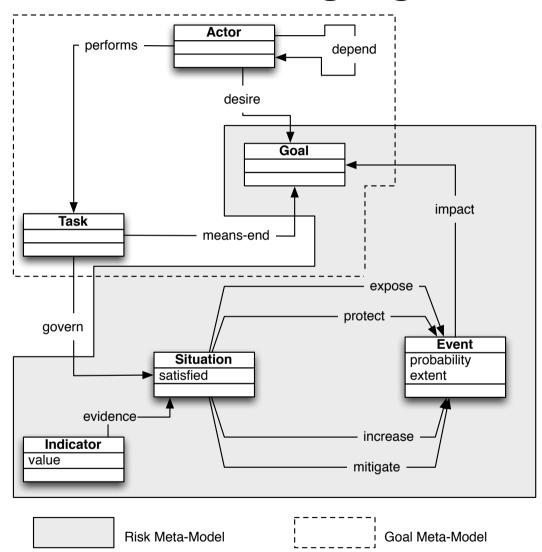
Levels of representation:

OSS ecosystems and risks together (in i^*)



Meta-Model of the risk language

- Connected to the goal-models of the ecosystems
- Allows for the specification of risk impact on goals, activities and other assets





Statistical analysis of OSS projects and communities



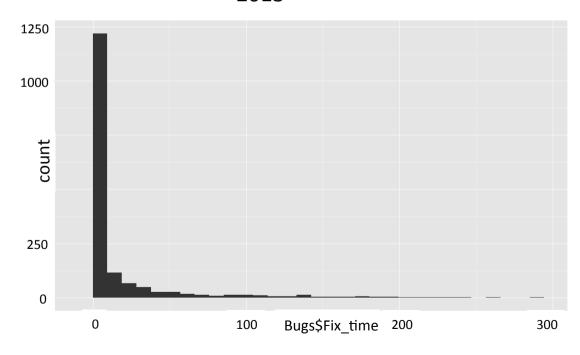
Statistic: "Bug fix time"

Study the "behavior" of the community in the project

n	1645
mean	16.22 days
sd	39.10 days
median	0.73 days
min	0 days
max	284.91 days

Statistical analysis of "Bug fix time" (in Xwiki OSS community)

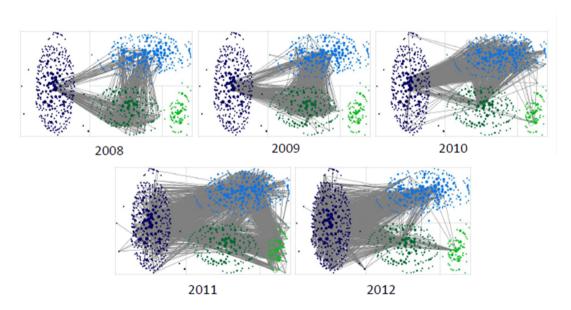
Date Range: August 6th 2012 to August 6th 2013





Community network analysis

- Analysis of the "structure" of the OSS communities and of their "evolution" via Social Network Analysis
 - Centrality measures and Prestige measures to determine the "connectivity" of nodes



e.g., to infer possible "critical" events in the community (such as a fork, a decrease in the activity)



Scenario for expert assessment

(Random) scenarios

Risk drivers and value of the intervals of their distributions

Risk Driver	State 1	State 2	State 3	State 4	State 5
Average bug fix time (days)	0	1	16	55	94
Bug fix time for critical & blocker level bugs	0	2	14	45	76
Commit frequency / week	0	21	44	90	113
Hour: When the commit was made	mostly morning	mid-day	mostly night		
Weekday: When the commit was made	mostly weekdays	mixed pattern	mostly week ends		
Holiday: When the commit was made	never	sometimes	always		
Timeliness	1	2	3	4	5

Scenario 1	Scenario 2	Scenario N
15	21	
3	3	
15	23	
mostly morning	mostly night	
mostly weekdays	mostly weekdays	
never	sometimes	
?	?	?

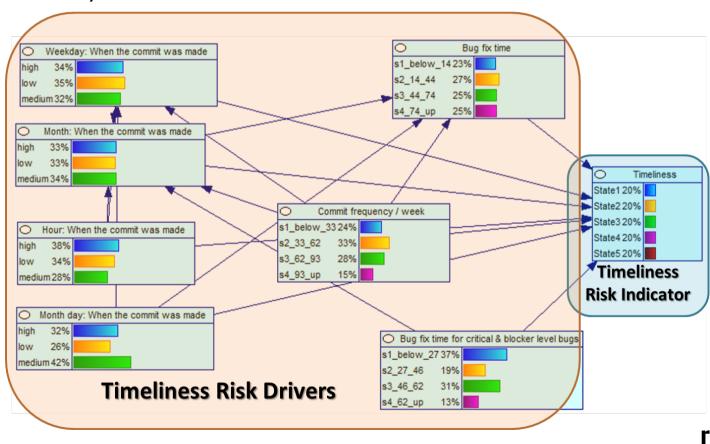
Expert assessment:

Evaluate how much the values of the Risk drivers in the scenario impact the Timeliness of the community (e.g., in the interval [1,5])

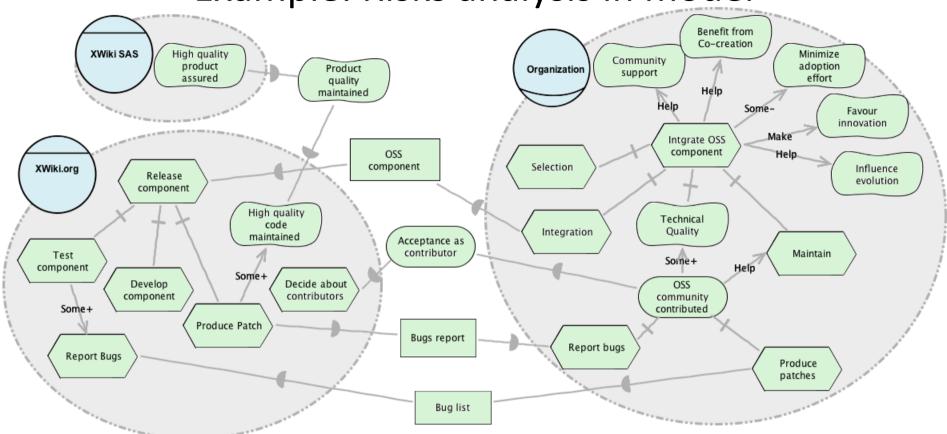


Resulting Bayesian Network

- Bayesian network (BN)
 - BN is a Directed Acyclic Graph (DAG)
 - Enable an effective representation and computation of the joint probability distribution over a set of random variables

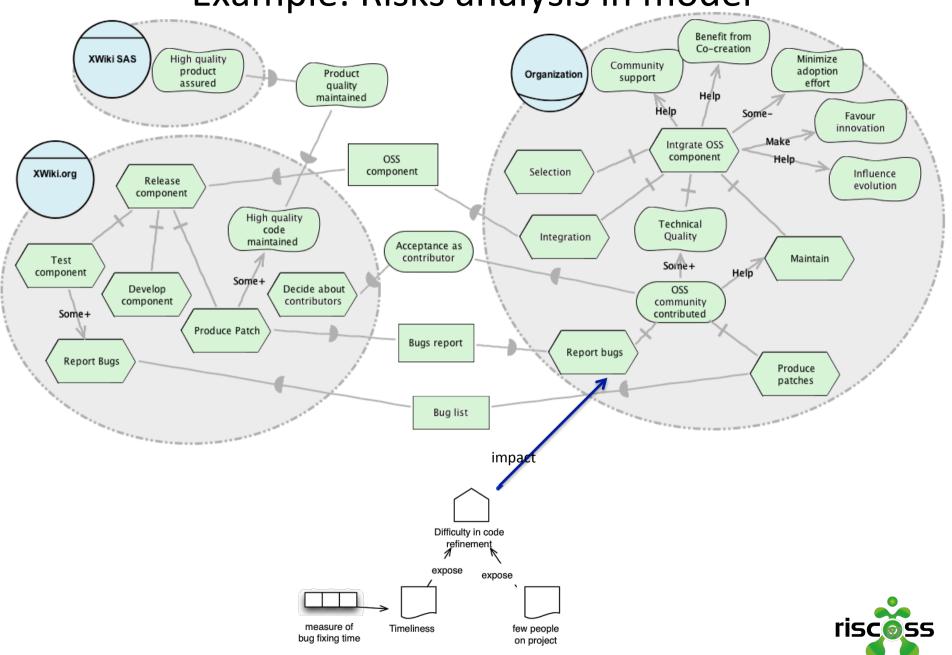


Example: Risks analysis in model

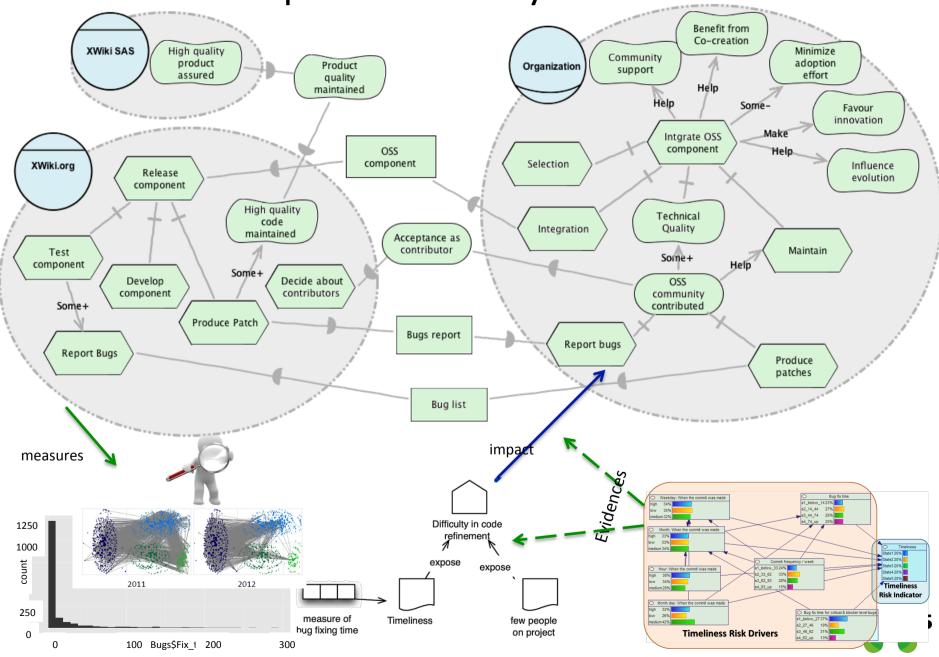




Example: Risks analysis in model



Example: Risks analysis in model

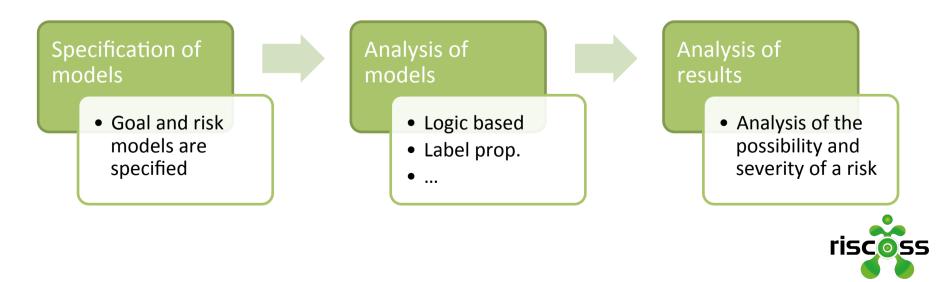


Reasoning on models



Risk and goal model reasoning

- Risk and Goal model analysis
 - starting from the knowledge about values of properties of some nodes of the model (Risk events, Situations, Goals, Activities) infer knowledge about values of properties of other nodes



Reasoning Techniques: logic based

- Model analysis via Disjunctive logic
 - Declarative logic language that offers primitives to represent:
 - disjunctive facts (which introduce alternative truth values of predicates)
 - disjunctive rules (in which disjunctions may appear in the rule heads to allow multiple alternative consequences to be drawn from a rule)
 - Allows for the analysis of alternatives (interesting also for mitigation strategies)
- Via DLV disjunctive logic reasoner

An example of rules facts and models:

Propagation: satisfiability and deniability

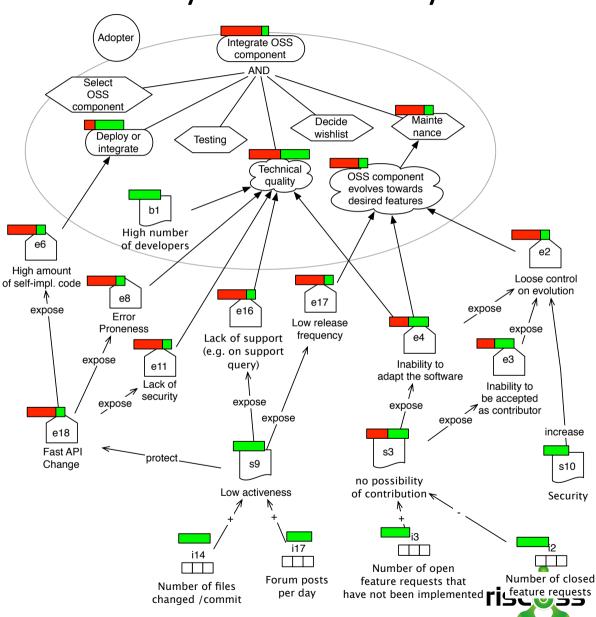
 Propagatoin of the evidences in two channels

Satisfiability

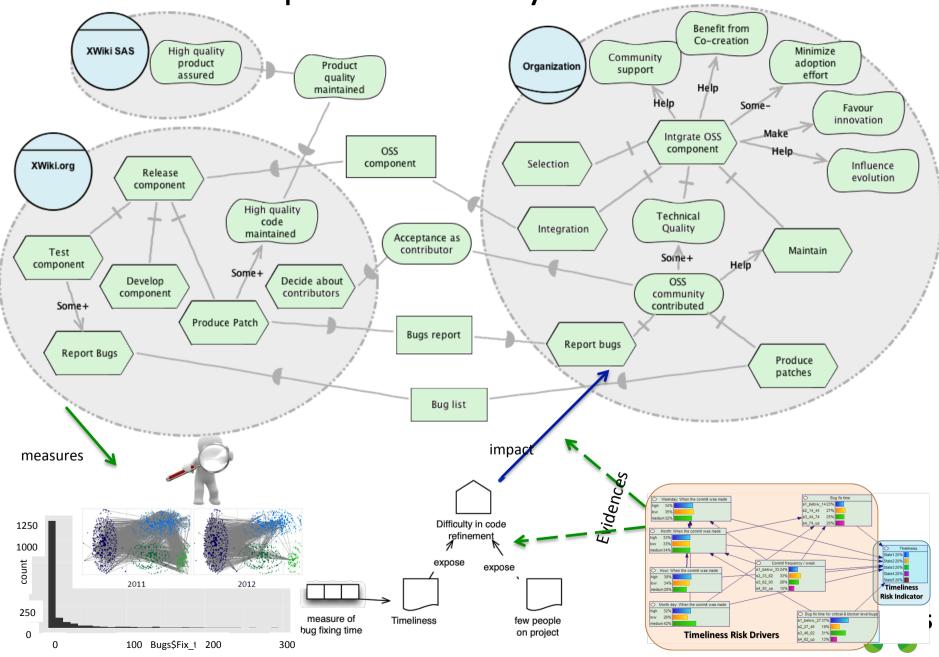
There is some evidence that "there will be lack of support"

Deniability

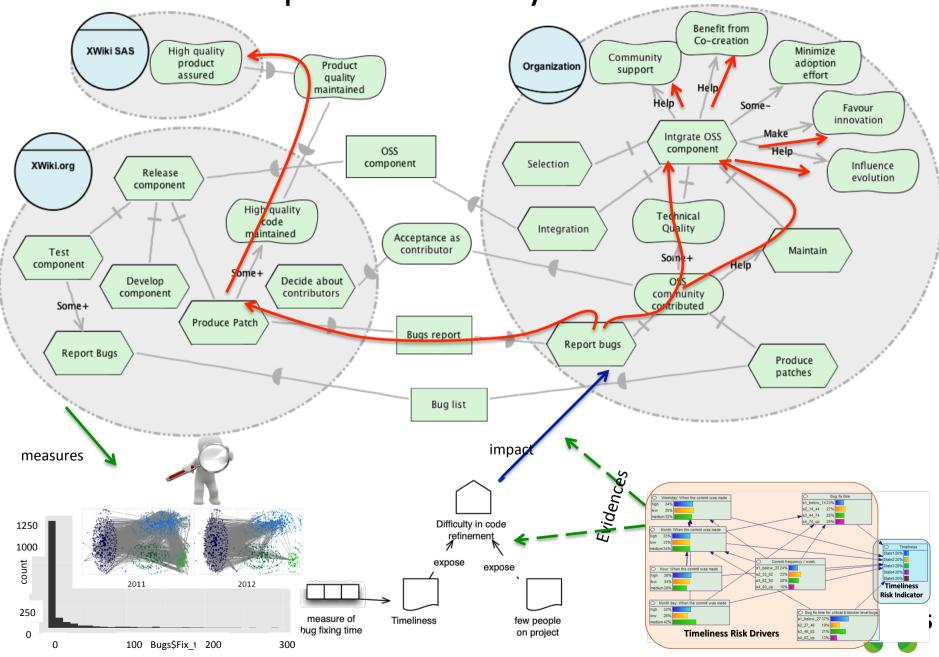
There is some
 evidence that
 "there will be no
 lack of support"



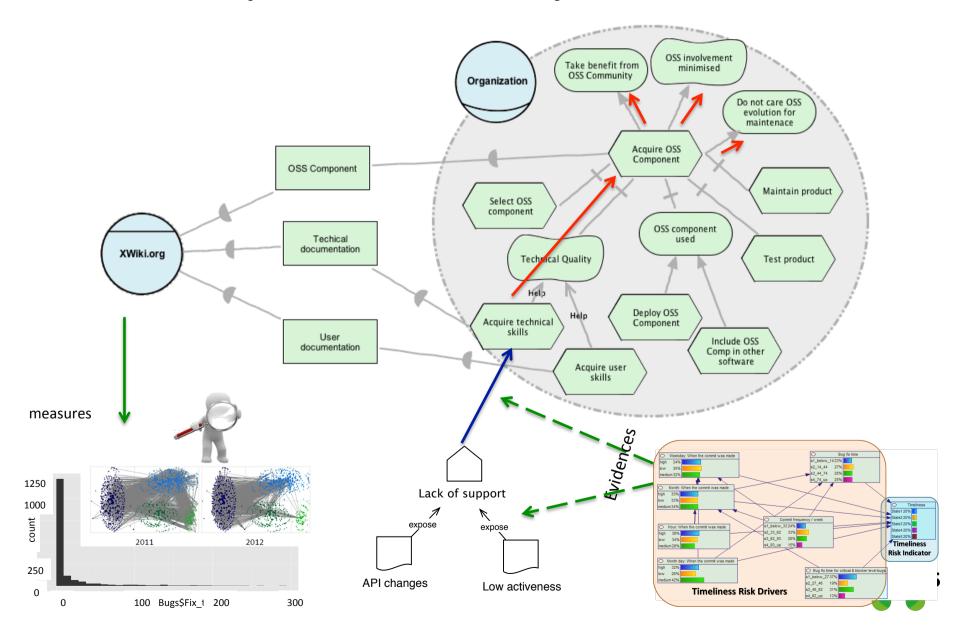
Example: Risks analysis in model



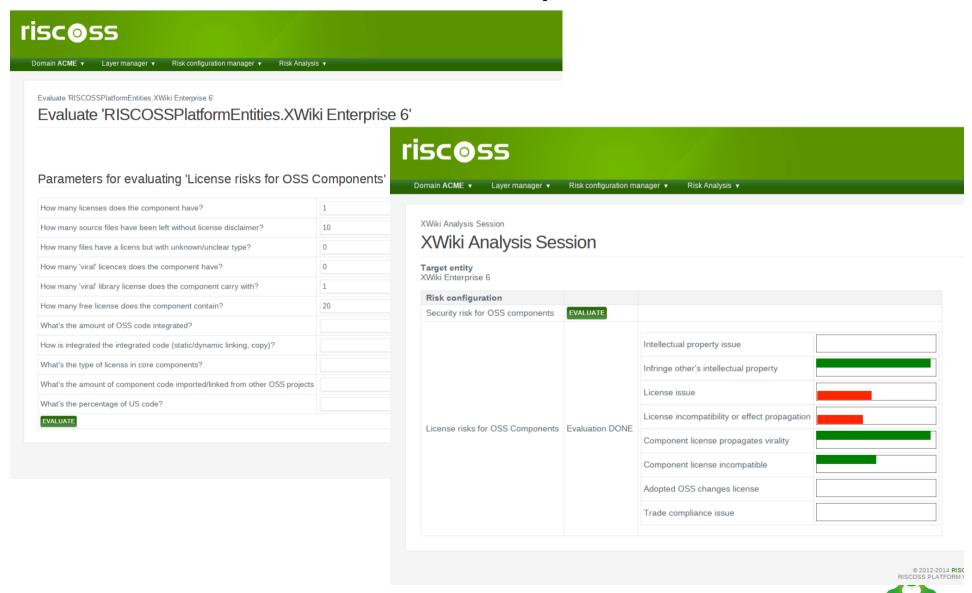
Example: Risks analysis in model

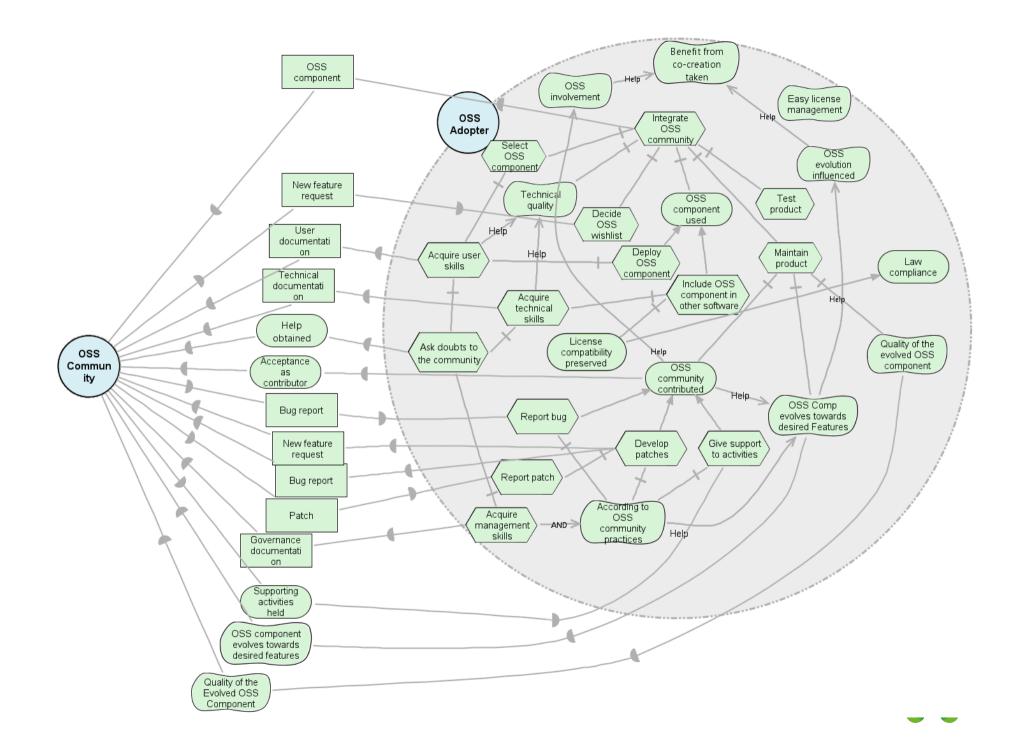


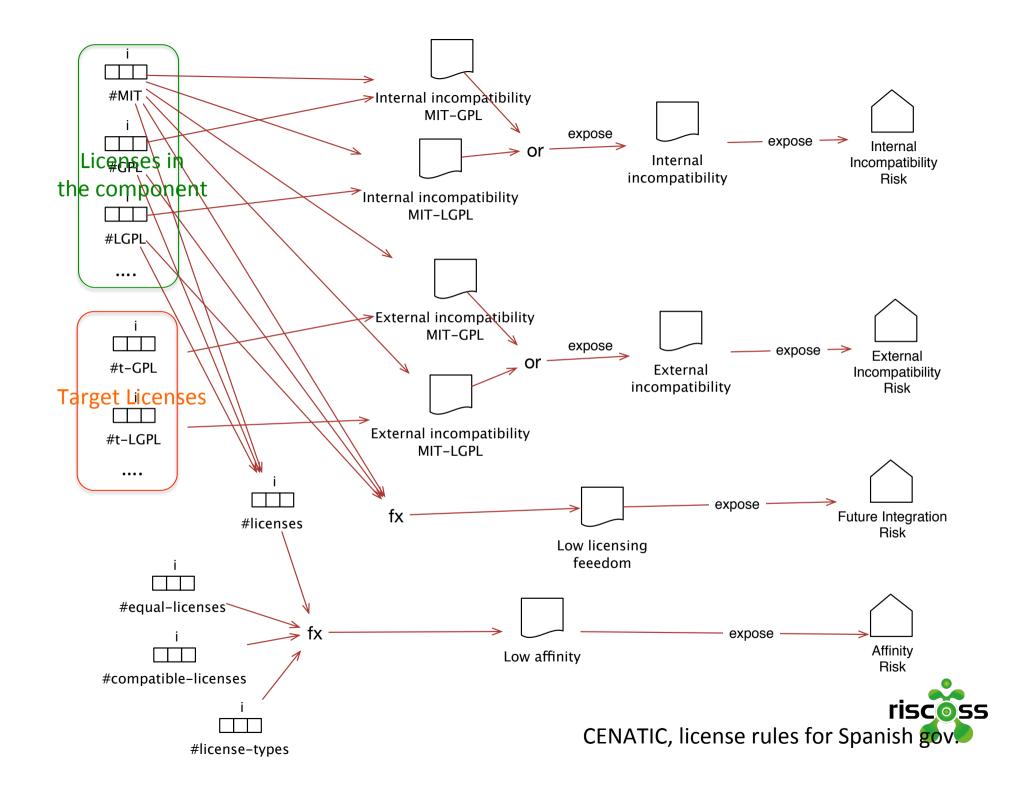
Example: Risks analysis in model



The RISCOSS platform





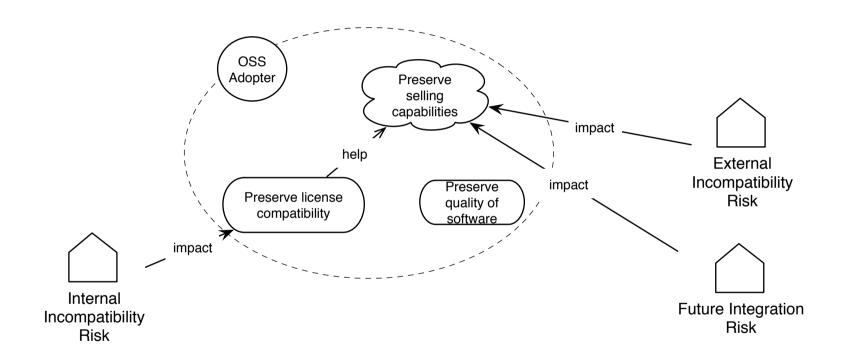


Risks inlicenses

- Internal incompatibility risk. Risk that two (or more) of the adopted components have licenses that are compatible with each other
- External incompatibility risk. Risk that the target license(s, in case of dual licensing) is incompatible with one or more of the component licenses
- Future uncertainty risk. Risk due to the low degree of freedom in the choice of the target license of future components
- Affinity risk. Risk that arises from the need of maintaining a given corporate licensing scheme. It measures how this set of components, although being compatible, deviates from the desired scheme (as specified by the target license)



The impacts





Some scenarios

- Licenses of the components and target licenses:
 - BSD4 and GPL2
 - BSD4 and LGPL2.1
 - BSD4 and LGPL2.1 + target license GPL2



Agenda

- An introduction to ecosystems
- The case of OSS ecosystems
- Ecosystem modeling with i*
 - Short background on i*
 - Modeling OSS ecosystems with i*
- Ecosystem analysis with i^*
 - i*-based analysis
 - An application: risk analysis in OSS ecosystems
- Final discussion



Summary

- Ecosystems provide a perspective that adapts with sociotechnical systems today
- The case of OSS ecosystems is particularly interesting due to the existence of a community behind
- Intentional modeling provide the opportunity of capturing strategic actors and relationships
- The resulting models are adequated for in-depth analysis



Thanks for your attention!



