Towards Risk-aware Scheduling of Enterprise Architecture Roadmaps

Christophe Ponsard, Fabian Germeau and Gustavo Ospina
CETIC Research Centre, Charleroi, Belgium

Keywords: Enterprise Architecture, Risk Management, Meta-modelling, Optimisation.

Abstract: Enterprises need to keep their organisation aligned with their business objectives. Enterprise Architecture provides a way to identify the current and target states as well as defining the evolution roadmap taking the form of a complex project portfolio. Conducting changes requires to deal with the occurrence of several risks at the project level which can affect strategic decisions. This paper investigates how to optimally deal with such risks in the global scheduling process. We start by identifying and structuring risks and projects concepts based on a domain model. We then identify a set of risks and related management scenarios. Experiments in risk control are carried out using two local search optimisation tools able to consider risk information in addition with inter-project and resource dependencies. We show the feasibility of efficiently anticipating risks and even dynamically adapting the scheduling. A dedicated focus is set on specific characteristics of IT projects.

1 INTRODUCTION

Enterprise Architecture (EA) is discipline for proactively and holistically leading enterprise responses to disruptive forces by identifying and analysing the execution of change toward desired business vision and outcomes (Gartner, 2014). EA supports a set of well-defined practices for conducting enterprise analysis, design, planning, and implementation, using a comprehensive approach at all times, for the successful development and execution of strategy (FEAPO, 2013). EA can also be viewed as a knowledge hub integrating and sharing information on various structures across the enterprise and the business transformation value chain, as depicted in Figure 1.

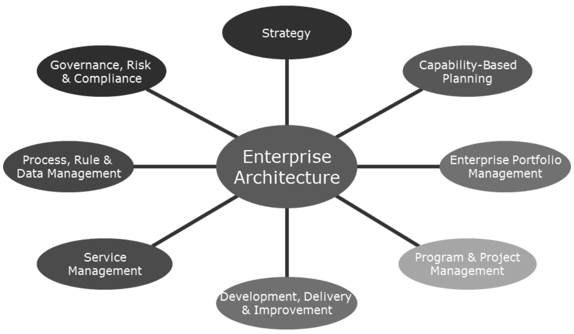

Figure 1: Enterprise architecture as knowledge hub.

The execution of change is carried out through a set of projects. Each project is a temporary endeavour undertaken to create a unique product, service, or result (PMI, 2008). A key role of EA is thus to define and manage a global roadmap of related projects to conduct the required change for keeping the enterprise aligned with its business objectives. Managing such a project portfolio is not an easy task for many reasons:

- the need to ensure the business continuity of the enterprise while transforming it.
- usually complex dependencies between projects reflecting the current state of its organisation and IT infrastructure (including technical debt).
- priority constraints for reaching specific milestones in due time.
- limited internal resource/budget to staff projects.

Projects can face many problems that can cause failures, delays or budget overruns. Project Management (PM) is especially recognised as severe in the IT sector where about half of the projects are challenged and 20% still fail (Standish Group, 2015). Project risk management is a well-defined discipline with standards or frameworks like PRINCE2(Murray, 2009), PMBOK(PMI, 2008) and ISO31000(ISO, 2009) are available.

Managing risks at the portfolio level is more complex. Enterprise Architecture provides a set of complementary viewpoints: business view (strategy, governance, organisation, and key business processes),

Ponsard, C., Germeau, F. and Ospina, G.
Towards Risk-aware Scheduling of Enterprise Architecture Roadmaps.
DOI: 10.5220/0007799806960702
In *Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS 2019)*, pages 696-702
ISBN: 978-989-758-372-8

data view (logical and physical data assets), application view (IT component and their interrelationships), and technical view (hardware, software, and network infrastructure). From those viewpoints, a risk approach can be carried out so that risks can be identified and addressed. Some risks can be directly managed, e.g. to address the challenge of the digital transformation more IT experts might be needed which is done through the hiring/training process. Other risks will be addressed by the specific project, which means that the risks landscape is itself evolving across projects, e.g. an obsolete component needs to be replaced but in the meanwhile other projects may need to deal with performance or reliability issues of that component.

Risk and risk-related activities like capability and project management is part of the global EA vision as depicted in Figure 1. Some EA frameworks also provide direct support and guidance to capture risks in specific EA viewpoints. The Open Group Architecture Framework (TOGAF) has a dedicated chapter for risk management, including risk classification, initial/residual assessment, and monitoring (The Open Group, 2009). ArchiMate does not support risk as first class concept but provides language customisation mechanisms allowing one to capture specialised concepts (The Open Group, 2017). A risk mapping is given as informative example and more elaborated mappings are also available (Jonkers, 2014).

Taking decision on how to best anticipate and manage risks when they materialise is quite difficult in the context of enterprise architecture given the complexity of the associated roadmap. Enterprises, especially smaller ones, usually use only simple and short-term strategies with reduced tool support. The purpose of this paper is to explore a better methodological and tool support to cope with this problem. This work goes beyond EA which is not about the timing but only on the sequence to progress from an as-is to a to-be situation. Hence, more information is required about available resources and budget to drive the planning. In order to maximise the chance to reach the to-be state within resource constraints, the planning need to take all relevant risks into account, especially those inferred from the EA model and the resources capabilities. Our paper takes the following approach:

- capture relevant information for carrying out a risk-aware roadmap execution, i.e. both risk and project information. We propose a domain model combining these dimensions. This provides the reference scheme for reasoning and later integrating tool support.
- identify a series of risks potentially impacting roadmap execution and defining some measures to address them.
- perform experiments about practical tool support that can be applied to deal with risks by adapting scheduling tools.

Our research is still in an early phase. We focus here on the meta-model and do not claim to be exhaustive about the studied risks scenarios. Our technological assessment is mainly oriented to technical feasibility in order to manage risk-based constraints.

The paper structure reflects our approach. Section 2 reviews some existing meta-models for project and risk management. Section 3 proposes a unified domain model over those aspects. Section 4 discusses some risks scenarios while Section 5 shows some results using scheduling tools. Finally, Section 6 concludes and presents our next steps.

2 OVERVIEW OF EXISTING PROJECT META-MODELS

Different domain models or meta-models have been defined in the literature, mainly from two sources: Project Management and Enterprise Architecture frameworks. They have slightly different scopes although the global structure is quite similar: they consider portfolio/program that gather projects and go in the details of projects at different levels of granularity.

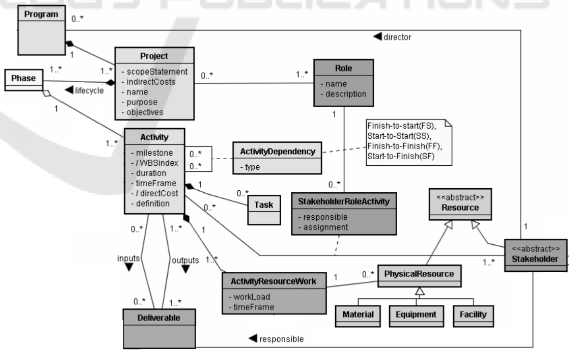

Figure 2: Partial projet meta-model from PMBOK.

As expected, PM frameworks focus more on the project concept and provide details in terms of operational concepts like task/activities, resources involved and products/deliverables. As example, Figure 2 shows the meta-model proposed by PMBOK (PMI, 2008). Other meta-models quite similar in nature have been proposed by other frameworks like RUP and OPEN. An attempt to define a common global meta-model is described in (Martins and da Silva, 2005). However, they tend to become unnecessarily complex and are mainly useful for interoperability.

Another characteristic of those meta-models is that they do not make explicit reference to risks although they capture risk factors.

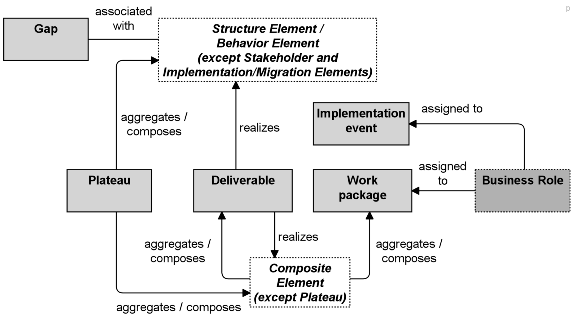

Figure 3: Project concepts of the ArchiMate meta-model.

EA frameworks target a broader perspective making links with goals, business services, infrastructure and have a more abstract notion of project. As shown in Figure 3, ArchiMate use the notion of "Implementation Events" to describe the transition between successive states named "Plateau" (The Open Group, 2017). However, some frameworks like Alfabet (SoftwareAG, 2015) do have an explicit concept of project. All EA frameworks include some form of risk management. Surprisingly, the risk concept itself is seldom explicitly present in the framework meta-model. The ArchiMate 3.0 specification states risks as an important EA aspect but does not cover them. TOGAF requires a specific meta-model extension to deal with risks(The Open Group, 2009).

3 PROPOSED UNIFIED DOMAIN MODEL

The previous section has highlighted the need to be able to characterise projects in relation with risks in order to cover classical risks (typically by addressing the quality/cost/delay equation) but also to be able to cope with the introduction of domain specific risks.

In order to provide a strong rationale when building our meta-model and making also sure it would enable to express risks in a quantitative way, we applied the obstacle refinement technique borrowed from Goal Oriented Requirements Engineering(van Lamsweerde, 2009). The starting point is a set of key project goals which are challenged by risks (modelled as obstacles) that are refined down to the point to be measurable on model elements. At this point, mitigation strategies can also be deployed, but our focus here is more on identifying key model elements that needs to be captured. We shortly highlight the process in structured text *on the delay goal, i.e. the risk of project overrun*:

↪ project deadline unrealisable → *project.deadline* (attribute in meta-model)
↪ reduced margins between tasks
↪ some tasks could to be late
  ↪ unreliable estimate of task duration → *task.plannedDuration*
  ↪ required resource unavailable → *resource.availability*
   ↪ *task plannedResource* vs *actualResource*
   ↪ possible unavailability rate of resource
   ↪ etc.

The resulting meta-model is shown in Figure 4. Projects are structured steps at different abstraction levels: portfolio, phase and tasks. Those can be used at the required level of granularity. Each step has a planned time interval for its completion and will eventually be completed in an actual time interval. Each step targets the production of some deliverable that is the concrete achievement of some operational goal which is aligned with strategic goals of the organisation. It also requires a set of resources of some type (e.g. monetary, time of material type) to guarantee its completion at some cost of which an actual set will be devoted. Risks can affect steps at various levels and obstruct the related goal. They are characterised by both their likelihood and impact. Impacts categories are defined and can be populated with generic risk types (resource unavailability) and more domain specific ones like production machine breakdown. Risks

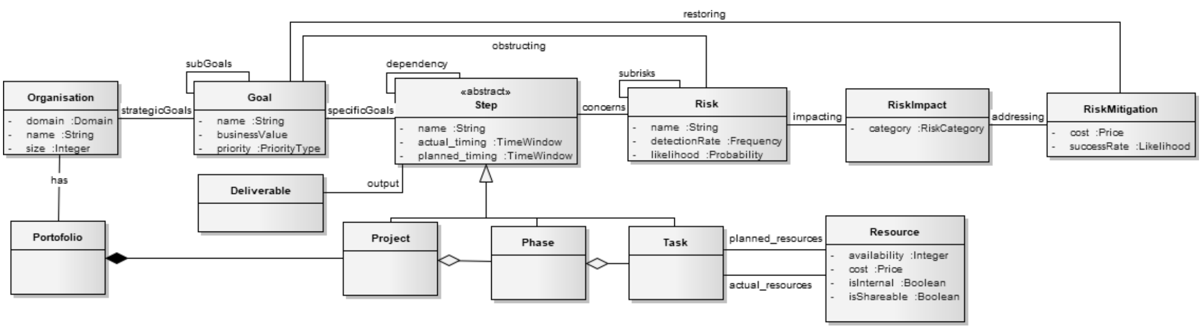

Figure 4: Resulting meta-model.

can be refined into sub-risks and specific mitigation actions can address them in order to restore the related goal. Like risks, different categories of mitigation actions can be foreseen and evolved based both on generic or domain specific strategies. In the process, costs can be considered, including for deploying some risk mitigation to guide its selection.

4 IDENTIFICATION OF RISK SCENARIOS AND RELATED MEASURES

This section reviews a few risks that can arise in the execution of a complex project portfolio and identifies some measures that can be used to address them. We especially identify measures that can be automatically explored by a scheduling process. As usual, the two risk factors requiring to be analysed are likelihood and impact.

4.1 Inherent Risk of an Specific Task

A task considered alone independently of its links with other tasks or the required resources can be more or be less risky given its nature and the enterprise context. Information provided by different viewpoints can directly help in estimating the risks, mainly at the impact level: are they bound to a strategic goal of the company? Do they impact many processes, applications or technologies? Are there alternative paths to reach the same result?

Reducing risks can be done in designing the roadmap itself to provide some backup solution or alternatives. A typical classification of risks will address the following dimensions:

- scope risks: new requirements, strategic change.
- scheduling risks: related to estimation, internal/external dependencies.
- resource risks: capability and availability.
- technology risks: related to hardware and software availability, reliability, security issues.
- environment risks, e.g. regulation, government, market, suppliers' issues.
- management risks: related to organisation complexity, available skills, bureaucracy.

Scheduling and resource risks directly affect the planning and will be detailed in the rest of the section. Other risks need to be managed at the organisation level. The residual risks should be considered in terms of possible delays or failures w.r.t. other tasks, depending on the correct achievement of this task.

4.2 Portfolio Structure Risks

A few risks can be inferred from the project portfolio structure. Our meta-model can capture those dependencies at different levels. At least project level dependencies need to be analysed and of course the related critical path to meet the target deadline should be identified. Specific constraints on project start time might also be imposed due to external constraints (dependency on external project/resource/technology) or financial issues (e.g. yearly budget approval).

Dependency impact of a task can be quantified based on indicators related to its presence on the critical path and on the number of downstream dependencies. A riskier task should be scheduled in priority to enable better recovery in case of delay. However, the global duration of each project and effort splitting across project should also be kept under control.

4.3 Resource Related Risks

A resource level characteristics (not all reflected in Figure 4) is the level of reliability of the resource and characterise the risk likelihood. For a process/machine, it is the mean time between failures (MTBF). For people, it can be related to the level of expertise w.r.t the task or some evaluation of their competency. The global capacity should also be taken into account, e.g. if a risky task needs to be reinforced, some resources should be available or another less important project should accept to be delayed. It is also possible to explore different alternatives to reach the same result by different combinations of task durations and intensity of allocations. However, the functional dependency can be complex, as discussed in the next point.

4.4 Domain Specific Risks (IT projects)

In many fields of project management and especially in the IT domain, it is well-known that adding manpower to a task will not result in a proportional speedup and is even totally ineffective at some point in the project, i.e. "adding human resources to a late software project makes it later" (Brooks, 1978). This means that adding more resource in this case is not an option. Such known impact needs to be modelled adequately to rule out the generation of such schedules. For a risky project, it is better to have secured the right level of resource and to have some safety margins.

5 RISK SCHEDULING EXPERIMENTS

5.1 Tooling Used

In order to carry out risk scheduling experiments, two Open Source tools where investigated. Both are based on local search techniques which is the recommended approach to deal with complex scheduling problems, although it does not provide guarantee to find an optimal solution. These tools are:

- OptaPlanner, a lightweight, embeddable, planning engine written in Java (Smet, 2006). It solves constraint satisfaction problems with construction heuristics and meta-heuristic algorithms. It provides support for scheduling with resources and modes which can be enhanced to deal with risks.
- OscaR is Scala toolkit for solving Operations Research problems with two main engines: Constraint Programming (CP) and Constraint-Based Local Search (CBLS) (OscaR, 2012). The second engine can efficiently solve complex scheduling problems through incremental evaluation of neighbourhood search(Van Hentenryck and Michel, 2005). OscaR also provides a high level description language for models and search procedures but has limited built-in support for scheduling.

The tooling was used in development mode and was also integrated inside Redmine, a complete web-based Open Source PM environment (Lang, 2006) which supports bug/task management and contains a basic Gantt viewer. It also provides a REST API for easing scheduler integration.

5.2 Scenario#1 - Independent Tasks with Increasing Risk Level

This simple scenario just considers a simple portfolio of 5 tasks without any dependency but relying on a single resource. It applies a penalty that is proportional to the risk level and start date of the tasks. This can be stated in OscaR.CBLS using the following objective function:

```
val actRisk = Array.tabulate(size)
  (i => { pb.startTimes(i)*penalty*i})
val globalRisk:Objective = Sum(actRisk)
```

Figure 5 shows the results where the tasks are executed in reverse order so that riskier tasks are executed sooner.

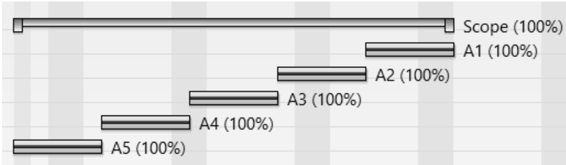

Figure 5: Scenario of independent tasks with increasing risk.

5.3 Scenario#2 - Starting Risky Tasks Earlier

In the three next scenarios, we will consider a portfolio composed of 5 instances of a similar project structure depicted in Figure 6. Task#1 is a blocker for 3 subsequent tasks. At resource level, task#1 and task#2 both need resource R#1 while task#2, task#3 and task#4 require resource R#2. Globally, only one instance of R#1 and three instances of R#2 are available to make the problem a bit constrained. This example contains enough complexity while still easy to depict and to explain using readable Gantt diagrams.

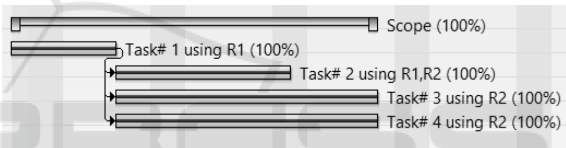

Figure 6: Simple project structure.

Figure 7 shows the result achieved by OptaPlanner with an objective function rewarding for early start of critical task#1 in all projects. The best possible result could be achieved with all projects intertwining to start this task. At the performance level, only basic search operators (insert/retract/swap) were used and reaching this optimal solution required about 30 seconds search time in verbose mode. It could probably be improved with a smarter way to select neighbourhood candidates.

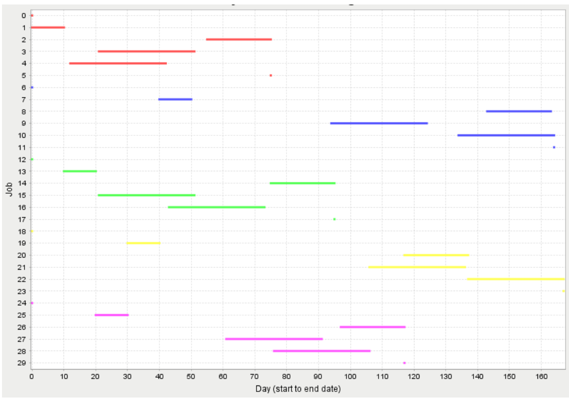

Figure 7: Scheduling to cope with risky tasks early.

5.4 Scenario#3 - Limit Scattering

While controlling the risk of critical tasks, the previous roadmap schedule has the drawback of running many projects in parallel. This results in a longer project duration, which is a known risk because of resource switching overhead and also having projects on hold for quite a while, e.g. project #3 in Figure 7.

A strategy to cope with this is to minimise project duration by using a score that rewards shorter projects. It can be expressed as follows using OptaPlanner (in a inefficient non-incremental evaluation mode). Note that the notion of soft score is for minimisation, while resource allocation is strictly enforced through hard constraints.

```
soft0Score=0;
for(Project p:projEndDateMap.keySet()){
  int ed = projEndDateMap.get(p);
  int sd = projStartDateMap.get(p);
  soft0Score += (sd-ed);
}
```

Figure 8 shows the result easily found by OptaPlanner. The global duration is the same as in the previous scenario. Of course, achieving compact schedules means starting some projects later and is thus not compatible with the previous scenario: in this case, the latest project is started after 90 time units rather than 40 before.

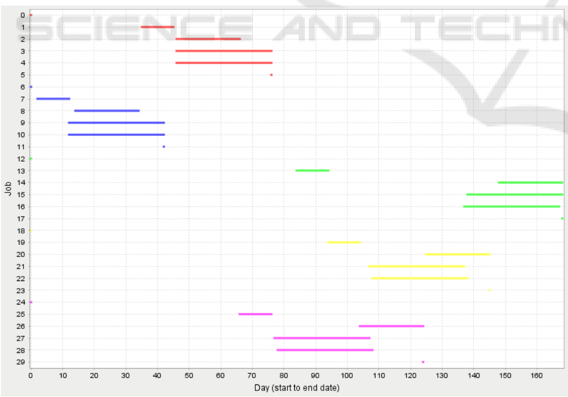

Figure 8: Scheduling to minimise project duration risks.

5.5 Scenario#4 - Redmine Integration

In this scenario, we connected the OscaR-based scheduler with a Redmine application (Lang, 2006). The REST API was used to retrieve the whole project and risk information matching our risk meta-model described in Section 3. For risk support, a dedicated plugin was used. The results are shown in a scheduling page featuring two Gantt diagrams, respectively before and after optimisation as depicted in Figure 9. A view of resource load over time is also available from a separate tab. The platform is available online in read only mode without registration need at this URL: http://prima-q.cetic.be.

Figure 9 shows the Gantt of a two-year project involving mainly six people (actually the project supporting this research). The optimisation reveals that some tasks could have been started earlier and that progress could have been achieved with more tasks in parallel. However, this does not totally reflect all task constraints and resource availabilities. This means that the result of the optimisation should be analysed carefully to make sure it relies on accurate data and to include some margins for the unforeseen.

6 CONCLUSION AND NEXT STEPS

In this paper, we have presented our approach to capture and model risks within the scope of an enterprise architecture roadmap and with the goal to perform risk-aware scheduling. We proposed a reference meta-model inspired from both the project and risk literature. Looking at different risks arising in enterprise operation, we identified a few of them directly impacting the scheduling, as well as measures to address them. Finally, we had some practical experiments with mature open source optimisation tools.

Our results so far are encouraging as we were successful in deploying our risk management strategies. Although our experiments are still of reduced size, the tooling showed the ability to scale, especially the OscaR tool which was also integrated with a pre-operational Redmine environment. In this respect, the developed meta-model proved useful to develop the required integration layer. We believe it can help others to map all relevant concepts into their own framework and tooling.

Going beyond this feasibility step, the next steps of our research are to investigate:

- more elaborated classifications of risks impacting the scheduling of the roadmap.
- the use of alternatives in project roadmaps and different possible implementations of tasks (e.g. duration vs intensity).
- more thoroughly the literature, especially related to Project Portfolio Management, System of Systems (Gillespie et al., 2017) and considering specific SME needs (Marcelino-Sádaba et al., 2014).
- how to make our tooling adequate for use by SMEs for the kind of EA projects they have to manage, in the line with our earlier work in this

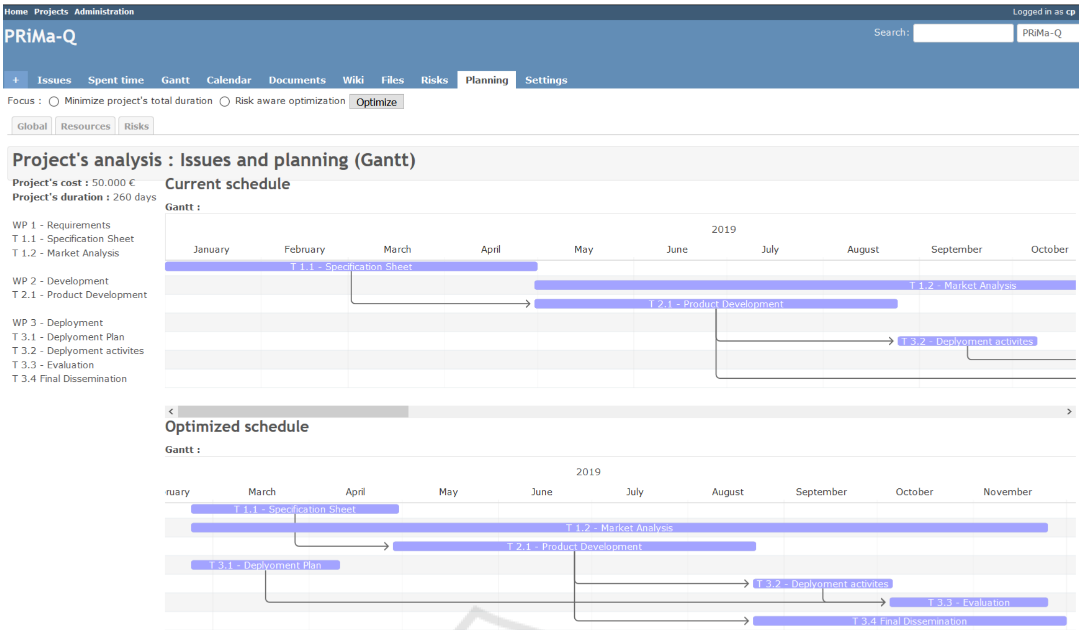

Figure 9: Redmine integration of the risk-aware scheduling.

area (Ponsard and Majchrowski, 2015).

ACKNOWLEDGEMENTS

This work was partly funded by the PRIMa-Q CORNET research project (nr.1610088) from the Walloon Region of Belgium and the German Federation of Industrial Research Associations (AiF). We thank our User Committee, especially Respect-IT and LabNaf.

REFERENCES

Brooks, Jr., F. P. (1978). *The Mythical Man-Month: Essays on Softw*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition.

FEAPO (2013). Common Perspectives on Enterprise Architecture. Federation of EA Professional Organizations, Architecture and Governance Magazine. Issue 9.

Gartner (2014). Gartner it glossary - enterprise architecture. https://www.gartner.com/it-glossary.

Gillespie, S. E. et al. (2017). System of systems architecture feasibility analysis to support tradespace exploration. In *12th System of Systems Engineering Conference, SoSE 2017, Waikoloa, HI, USA, June 18-21*.

ISO (2009). Risk management – principles and guidelines. ISO 31000.

Jonkers, H. (2014). Enterprise architecture-based risk assessment with archimate. https://bizzdesign.com/blog.

Lang, J.-P. (2006). Redmine. http://www.redmine.org.

Marcelino-Sádaba, S. et al. (2014). Project risk management methodology for small firms. *Int. Journal of Project Management*, 32(2):327 – 340.

Martins, P. V. and da Silva, A. R. (2005). PIT-P2M: ProjectIT Process and Project Metamodel. In *Proc. of OTM Confederated Int. Conf. on the Move to Meaningful Internet Systems*. Springer-Verlag.

Murray, A. (2009). *Managing successful projects with PRINCE2*. TSO, Norwich.

OscaR (2012). OscaR: Scala in OR. https://bitbucket.org/oscarlib/oscar.

PMI (2008). *A Guide to the Project Management Body of Knowledge (PMBOK Guide), 4th Edition*. Project Management Institute.

Ponsard, C. and Majchrowski, A. (2015). Driving the adoption of enterprise architecture inside small companies - lessons learnt from a long term case study. In *Proc. of Int. Conf. on Enterprise Information Systems*.

Smet, G. D. (2006). OptaPlanner Open Source Constraint Solver. http://www.optaplanner.org.

SoftwareAG (2015). Alfabet, Enterprise Architecture Management. www2.softwareag.com/corporate/products/aris_alfabet/eam.

Standish Group (2015). CHAOS Report. http://www.standishgroup.com.

The Open Group (2009). TOGAF - The Open Group Architecture Framework V9.

The Open Group (2017). Archimate 3.0.1. http://pubs.opengroup.org/architecture/archimate3-doc.

Van Hentenryck, P. and Michel, L. (2005). *Constraint-Based Local Search*. The MIT Press.

van Lamsweerde, A. (2009). *Requirements Engineering - From System Goals to UML Models to Software Specifications*. Wiley.