Results of a Survey on Improving the Art of Semantic Web Application Development

Emanuele Della Valle[1], Gaetano Niro, and Catalina Mancas[2]

[1] Politecnico di Milano, Dipartimento di Elettronica e Informazione, Milano - Italy
[2] University of Craiova, Software Engineering Department, Craiova - Romania
emanuele.dellavalle@polimi.it, gaetano.niro@gmail.com, Catalina.Mancas@ucv.ro

Abstract. While the demand for solutions based on Semantic Web Technologies (SWTs) is rapidly growing, engineering a Semantic Web Application (SWA) is still largely an art. We believe that the lack of an end-to-end methodology is a major barrier for industrial uptake of SWTs. In order to empirically prove our thesis, we run two surveys: a preliminary one in summer of 2010 with 26 participants and a broad one in winter and spring 2011 with 111 participants. In this paper, we report our main findings: we identified the the most frequent barriers in applying SWTs, we analyse the impact of SWT adoption on the various phases of a software project, we examined how risky it is to apply SWTs, and we report on the reaction of both the developers team and the customers towards SWTs. We believe that the analysis herein presented can cast some light on the art of engineering SWAs and help in understanding which methods and tools the community should focus on for fostering Semantic Web industrial uptake.

1 Introduction

In the last decade, a growing amount of Semantic Web Applications (SWAs) have been identified [11,6,16] and developed using the Semantic Web Technologies (SWTs) standardized by the World Wide Web Consortium (W3C).

The recent industrial uptake Linked Data [5] can lead to the fake assumption that every future software project employing SWTs will succeed. However, in software project management, successful projects are the result of the right processes correctly monitored. Unfortunately, ontology engineering [13,21,20] is the only deeply studied process in SWA development.

Our thesis is that the peculiarities of SWAs require a specific end-to-end methodology. What we see, when we observe the Linked Data successes, are projects based on the *art and craft* of few early (and very skilled) adopters. In the past years, we discussed our thesis with several Semantic Web expert, and they generally disagree. The Semantic Web community appears to believe that a project manager leading a SWA project should chose an appropriate standard software process and integrate ad-hoc the ontology development process.

In order to empirically prove our thesis, we run two surveys: a preliminary one in summer of 2010 with 26 participants chosen among a carefully selected group of project managers that lead SWA projects and a broad one in winter and spring 2011 open to every body. The participants to the second survey were 111. The topics of the two surveys were similar. The main difference (apart from the number of interviewees) among the two surveys is that the most frequent answers to the open questions of the first survey, became possible choices for the close questions of the second one.

In this paper we present step by step the research methodology adopted and we report our main findings: we identified the most frequent barriers in applying SWTs, we analyse the impact of SWT adoption on the various phases of a software project, we examined how risky it is to apply SWTs, and we report on the reaction of both the developers team and the customers.

The rest of the paper is organized as follows. In Section 2, we present the research process. In Section 3 we briefly review the literature that we used to formulate our two surveys. Section 4 presents how the surveys were created the content of the two surveys. In Section 5, we describe how the two surveys were conducted and the relationships among them. In Section 6, we report the results of the second survey and we discuss our findings. Finally, in Section 7 we briefly conclude sketching a methodology supported by a case tool.

2 The Research Process

The research process followed in conducting our work consists in four steps illustrated in Figure 1. In 18 months, we repeated those steps twice.

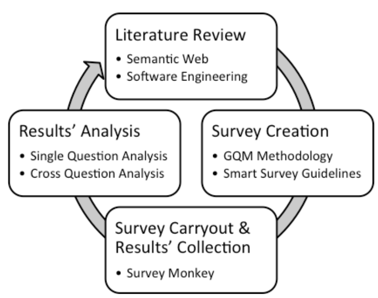

Fig. 1. Our Research Process

The very first step in our research process consists in **reviewing the literature** of Semantic Web and Software Engineering from the late '90s until today, in order to understand the current state of the art of SWA development in terms of available technologies, process, and tools.

In the second step, the **survey creation**, we use the *Goal, Question, Metric (GQM) Approach* [2] to identify the goal, i.e., check whether people that lead SWA projects perceives the need for an end-to-end methodology for SWA development, formulate the questions and decide the metric to use to collect the answer for an effective analysis.

The third step consists in the **survey carry-out and results' collection**. For the preliminary survey, which we conducted in summer of 2010, we collected the answer of the 26 interviewees by sending the questionnaire via email and, where possible, personally interviewing them. For the broader survey, which we conducted in winter and spring 2011, we used SurveyMonkey, a Web based survey tool. The interviewees to the second survey were 111.

The fourth and final step consists in the **analysis of the results**. We used *pivot tables* to perform both single-question and cross-question analysis. We used the results of the first survey to refine and focus the second one. For instance, the most frequent answers to the open questions of the first survey, become possible choices for the close questions of the second one.

3 Literature Review

In this section, we present the results of our literature review clustered by the following topics: *a*) domains in which SWAs have been developed, *b*) expected benefits of SWTs, *c*) barriers, problems and risks commonly encountered in developing SWAs, *d*) impact on Software Development Lifecycle (SDLC) of the introduction of SWTs in a software project, *e*) resource consumption and reusability of SWTs, *f*) techniques for costs and effort estimations in software projects targeting the development of a SWA, *g*) impact on the developers' team of the adoption of SWT, and, *h*) last but not least, customer satisfaction in software projects targeting SWAs.

Domains. In our literature review we found many papers that attest successful results at applying SWTs in a large variety of domains. The trace starts back in the late '90s when papers like [10] recognized a *common thread* binding the industrial and research communities: the urge need to share meaning of terms and common understanding of certain domain. Given that such a sharing can be easily achieved by conceptualizing a domain in an ontology, any domain enough *knowledge intensive* can be a potential target for SWA development. According to an authoritative study [12] conducted in 2009 in Europe, the four prototypical application fields for SWAs are: Healthcare and Biotechnologies, Knowledge Management, eCommerce and eBusiness, and finally, Multimedia and Audiovisual Services.

Benefits. The literature abounds in papers claiming that the Semantic Web is going to embrace the same success as the Web [3,6,16,8]. These claims are in many cases supported by industrial demonstrators. A rich collection of them is available in W3C Semantic Web Case Studies and Use Cases [1]. By analysing them we extracted the following list of often claimed benefits: improving search relevance, enabling automatic annotation and tagging, offering better control over data and documents, and easing information and data integration.

Barriers, Problems and Risks. Few papers report on investigations focused on encountered barriers, problems and risks in developing SWAs. One of the most authoritative papers under this perspective remains [10] that identifies: lacks in usability; lack of tools; lack of skilled programmers; lack of methods and of best practices; low capability to estimate costs; and inability to assess benefit in a clear way. While we are not saying that no progress at all has been made (see, for instance, the introduction of Linked Data bests practices [4]), we believe that the Semantic Web community is still under-estimating how high the barriers for broad industrial uptake are.

Impact on SDLC. As we already discussed in Section 1, methodologies supporting the development of ontologies for SWAs [13,21] and general-purpose software development methodologies already exist [20]. To our best knowledge, no end-to-end methodology for developing SWAs has been proposed and no one has conducted a systematic study on the impact of SWTs on the different phases of development of a software project.

Resource Consumption and Reusability. The literature supports the idea that SWTs produce cost savings. For example, the authors of Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering [22], identify two positive aspects which lead to a cost decrease: a reduced maintenance overhead achieved through increases in consistency and an increased potential for reuse, substitution or extension via content discovery on the Semantic Web. However, this claim is based on few use cases developed in research project. An in-the-field survey has not been conducted.

Cost Estimations. The Software Engineering literature [15] correlates successful software projects with rigorous estimation of time, costs and resources. The Semantic Web community has not proposed a cost estimation tool for SWA development, yet. The most advanced tool, proposed so far, is ONTOCOM [19], a parametric tool that estimates the costs of ontologies.

People Dimension. As books like Peopleware [7] remind us, people are the most valuable asset in a software projects. Even if the Software Engineering literature abounds of research on the topic we were not able to find any study tailored to Semantic Web related literature on this topic.

Customer Satisfaction. As we learn from software project management literature [15] minding and negotiating customer's expectations is the key to a successful software development. Nevertheless SWA customers' expectations, assumptions, needs and reactions are only recently getting the relevance that they deserve. The best quote we found on this topic is by Richard Zlade who, in "Inventing the Semantic Web ... again" [23], states: *The Semantic Web [...] potential is enormous and the enthusiasm around it is fully justified. But lets not fool ourselves. We still need legions of users to make the Semantic Web successful. To win them over, were going to need to speak to their goals [...].*

4 Creating the Surveys

The literature review attests the rapid success of SWAs, but, at the same time, the lack of several typical software engineering researches targeted to SWA development. To the best of our knowledge no empirical study was conducted on the impact of SWTs on SDLC and development team, on the actual reusability of Semantic Web artefacts, and on the satisfaction of the costumer.

To formulate the questionnaires of the two surveys we applied the Goal, Question, Metric (GQM) approach [2]; one of the most suitable method in *defining measurable goals*. Defining measurable goals is, indeed, a crucial step in creating the questionnaire because we cannot simply ask: *do you feel the need for an end-to-end methodology for SWA development?* We have to ask questions that reveal such a need (if present) and we need to define what exactly we want to analyse, making sure that a high percentage of the interviewees will understand the questions and will be able to answer them.

The GQM Approach has a hierarchical structure presenting three levels of refinement. The upper conceptual level is the **Goal** level, where the aim of the survey and the viewpoint are decided. It is followed by the **Question** or the operational level, where a set of questions is used to characterize the way the assessment of a specific goal is going to be performed. The questions try to characterize the object of measurement with respect to a selected quality issue and to determine its quality from the selected viewpoint. The third and last quantitative level is the **Metric** level. Once the questions have been developed they need to be associated with the appropriate metrics. The same metric can be used to answer different questions under the same goal.

We identify as **goal** of our survey the *analysis of the life cycle of a SWA development trying to understand if the interviewees perceive the need for an end-to-end methodology*, and as **view point** the one of an industrial software project manager.

The space we can dedicate in the paper to **questions and metrics** allow us only to present the seven areas they cover[3]:

- **Personal information** – we used this information to check that the interviewees matches our viewpoint and to perform some cross-question analysis (e.g., breaking down the answers on the domain the interviewees developed their SWA for).
- **Barriers in applying SWTs** – our intention was to assess the most often encountered obstacles in applying SWTs. As metric, we propose to choose among the barriers attested by the literature (e.g., the redundancy with regard to other systems; the lack of tools, methods and best practices; or the fear of losing control over one's data) and a list of SWTs (i.e., reasoners, SPARQL engines, ontologies, RDF stores, Linked Data, or other specified by the interviewee).
- **Impact of SWTs on each phase of the SDLC** – our intention was to assess how much each software development phases is positively or negatively affected by the adoption of SWTs in the software project. As metrics, we propose interviewees to choose among two lists. The first includes the usual project phases, i.e., concept identification, requirements' collection and analysis, design, coding and debugging, integration, testing, deployment and

[3] At [14,17], we provide interested readers with the text of the two surveys.

maintenance. The second focuses on processes specific of SWA development, i.e., SWA design, ontology engineering, Linked Data integration, reasoner integration, RDF store integration, querying reasoners and RDF stores, and providing a tailored visual interface for such knowledge intensive application.

- **Risks in SWAs development** – our purpose was to identify the most frequently encountered problems in developing SWAs. We collected the answers using three metrics. The first one is the simple choice between the risk of running a project with higher costs, longer duration and lower quality of the resulting application. The second metric is a list of SWTs (i.e., the same metric used in the question about the barriers). The third metric is a list of processes specific of SWA development (i.e., the same metric used in the question about the impact of SWTs on SDLC).
- **Team's acceptance of SWTs** – our aims was to identify the team's behaviour when they have to deal with SWTs. In the first survey, we left this as an open question. In the second version, we were able to propose a metric based on the most frequent answer of the first questionnaire (e.g., *teams requires additional training*, or *difficulties in meeting deadlines*) as true/false statements. In addition to those true/false statements we also used as metrics the list of processes specific of SWA development (i.e., the same metric used in the questions about the impact of SWTs on SLDC and risks in developing SWAs) and the list of SWTs (i.e., the same metric used in the questions about the barriers and risks in developing SWAs) to understand the level of effort required to perform the processes and use the SWTs.
- **Cost and effort estimation in SWAs** – our intention was to understand if interviewees perceive the usefulness of a cost estimation tool specific to SWAs. In particular, we asked the interviewees whether they estimate costs or not and what methods they use when performing the cost estimation. Moreover, the question asks whether there is a clear separation between the cost estimation for the part of the application that involves usage of SWTs and the rest of the application.
- **Customer's satisfaction about uses of SWTs** – our intention was to study the customer reaction to the use of SWTs in the beginning and in the end of the project.
- **Final remarks** – our purpose was to understand the general perception of the interviewees about the survey and collect indications about how to improve it.

5 Conducting the Surveys

As a target for our first survey we chose team coordinators with at least 10 years of experience (i.e., project managers in industry and main investigators in academia) and developers with at least 5 years of experience. We specifically targeted people active in technology transfer, i.e. people from academia with a record in technology transfer (e.g., launch of academic start-ups), and people from industry with an attitude in transferring research results. As main source of potential interviewees we used the authors of the W3C Semantic Web Use Cases and Case Studies [1].

In summer 2010 we contacted 51 people and we collect 26 questionnaires. The number of interviewees was small, but sufficient to understand strong and weak points of our first survey. Moreover, we were able to use the answers to the open questions to formulate close questions in the second version of the survey.

For the second version of the survey, we used SurveyMonkey, a web based tool that allows to formulate complex questionnaires with switches and that randomizes order of answers in close questions. The survey was open in winter and spring 2011. In total we collected 111 answers, 52,6% of which were provided by people that exactly match our target (i.e., team coordinators with at least 10 years of experience) and 26,3% of which were closely matching our target (i.e., chief developer with at least 5 years of experience). We believe that this numbers are sufficient to claim a good level of generality for our findings.

6 The Results Analysis

In this section, we run through the findings of our second survey. Since some of the questions were optional, we do not report results of analysis made on questions that received little attention by the interviewees.

6.1 Barriers in applying SWTs

As shown in Figure 2), according to our interviewee target the top-4 barriers are: lack of tools, lack of skilled programmers, lack of usability of the available tools and lack of standard ontologies. The bottom-2 barriers are: redundancy with regard to other systems and lack of scalability. In terms of SWTs, reasoners and ontologies are largely the most complex to put at work, while RDF stores and Linked Data appears easy to include in a software project.

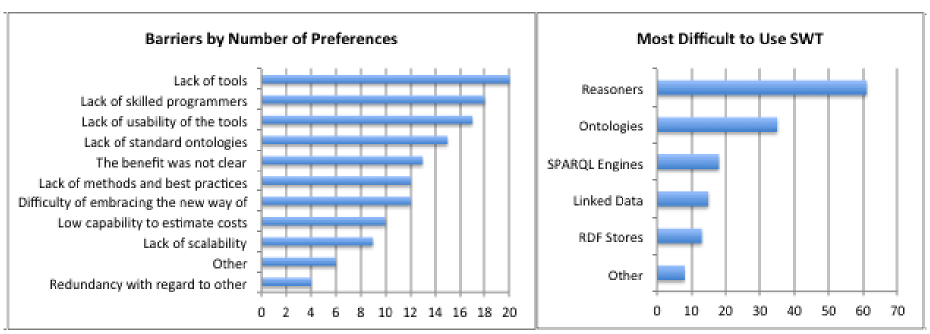

Fig. 2. Barriers perceived by the target group of our survey.

Figure 3 allows for a cross-analysis of the answers by breaking down the results according to the domain the SWA under analysis was developed for.

	Knowledge Management	Enterprise App. Integration	Information Retrieval
Redundancy with regard to other systems	17,5% (7)	17,6% (3)	20,0% (6)
The benefit was not clear	35,0% (14)	35,3% (6)	33,3% (10)
Difficulty of embracing the new way of thinking and working	35,0% (14)	29,4% (5)	20,0% (6)
Lack of methods and best practices	**55,0% (22)**	52,9% (9)	46,7% (14)
Lack of tools	47,5% (19)	29,4% (5)	**53,3% (16)**
Lack of skilled programmers	40,0% (16)	35,3% (6)	33,3% (10)
Lack of usability of the tools	37,5% (15)	47,1% (8)	43,3% (13)
Lack of scalability	45,0% (18)	52,9% (9)	**53,3% (16)**
Lack of standard ontologies	52,5% (21)	**58,8% (10)**	50,0% (15)
Low capability to estimate costs	20,0% (8)	5,9% (1)	20,0% (6)

Fig. 3. A cross-analysis of the perceived barriers broken down by the domain the SWA under analysis was developed for.

While the top-4 barriers are (sort of) natural for innovative technologies, the bottom-2 sounds as an alert for all the ongoing research on scalable reasoning. We advance the hypothesis that to foster industrial uptake of reasoners and ontologies research should focus on support tools.

6.2 Impact of SWTs on each phase of the SDLC

As readable in Figure 4, the target of our survey, in general, believes that SWTs have a positive impact on the early phases of a software project (with the exclusion of the requirement gathering phase) and on integration, while they negatively impact on code & debugging and testing phases.

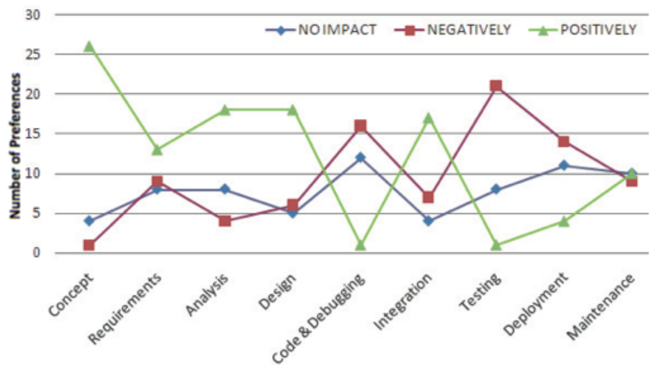

Fig. 4. Impact of SWTs on the different phases of a software project.

The results for the requirement and maintenance phases are controversial, our target group appears to have experienced both positive and negative impacts. It is worth to note that this results contradicts evidence from literature [22]. If we have the opportunity to run a third version of this survey, we will include specific questions on those points.

In terms of SWA development specific processes, consensus appears to be that ontology engineering, reasoning integration and developing appropriate user interfaces are the most complex process, while integrating RDF stores and developing the SWA are the least complex. The results in general confirm those of the question on barriers. A particular attention point emerges for user interfaces.

6.3 Risks in SWAs development

.

As visible in Figure 5, according to our target, the higher risks for a software project aiming at developing a SWA are to overrun (the opinion of the 44,8% of the target) and exceed budget (the opinion of the 37,9% of the target). Only

	Knowledge Management	Enterprise App. Integration	Information Retrieval	Response Totals
More TIME	**38,9% (7)**	35,7% (5)	**50,0% (11)**	**44,8% (13)**
Higher COSTS	33,3% (6)	**42,9% (6)**	31,8% (7)	37,9% (11)
Lower QUALITY	16,7% (3)	21,4% (3)	9,1% (2)	10,3% (3)
No risks	11,1% (2)	0,0% (0)	9,1% (2)	6,9% (2)

Fig. 5. Risks perceived by the target group of our survey.

10,3% of the target believes that the adoption of SWTs can expose the software project to the risk of lower quality of the resulting software artefact. The results are similar even when we take a closer look at the data breaking them down per domain. However, it is worth to note that our target group believes the risk of lower quality of the solution is significant (21,4%) in the enterprise application integration domain.

As shown in Figure 6, In terms of SWA development specific process the top-3 risky ones are: ontology engineering, developing user interfaces and integrating Linked Data.

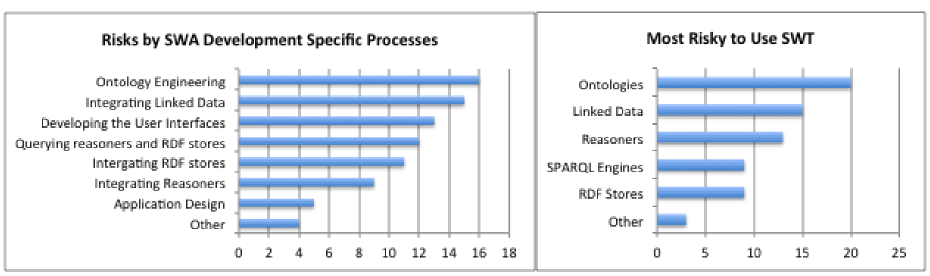

Fig. 6. Risks perceived by the target group of our survey.

While the first and the third risk confirm the results of the answer on barriers and impact of SWTs on SDLC, the second one (i.e., integrating Linked Data) appears to contradict the results of the question about barriers, where Linked Data are believed to be easy to integrate. This point requires further investigation. The good news is that integrating reasoners and designing the SWA are perceived as low risk tasks.

6.4 Team's acceptance of SWTs.

As shown in Figure 7, among the true/false statement, which we proposed to our interviewees, the target of our survey believes that it is difficult to find the developers having enough knowledge about SWTs and that, thus, the team requires additional training. In general, they do not believe that the morale of the team is negatively effected by the adoption of SWTs. They have mixed feeling about the statement that the team is not able to meet deadlines due to the adoption of SWTs.

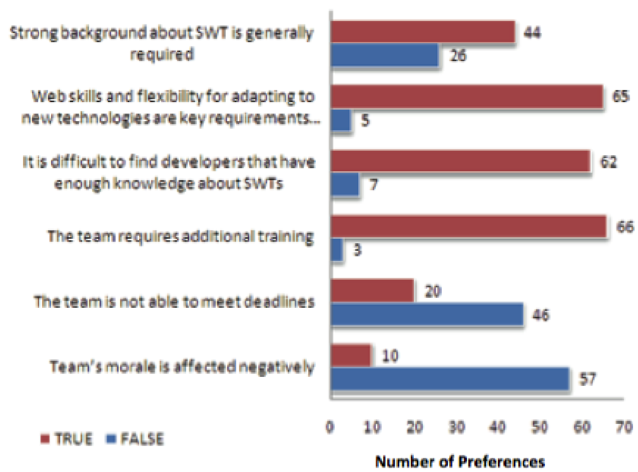

Fig. 7. Number of interviewees in the target group of the second survey that agreed or disagreed with the statement about team acceptance of SWTs, which we collected in the first survey.

Moreover, as visible in Figure 8, the results of the question of the effort spent on the SWA development specific processes and the list of SWTs confirms the finding of the question about barriers and risks with a small difference on *querying reasoners and RDF stores* for which a medium/high effort is highlighted.

6.5 Cost and effort estimation in SWAs.

The question about cost estimation was answered only by 26 interviewees and even less of them answered the subquestions. Apparently only few people in our target group estimate costs (i.e., 13 interviewees). We believe this to be a concrete example of the general immaturity of the processes about SWA development from a software engineering point of view, but we cannot claim the generality of this statement and we omit the analysis of the results.

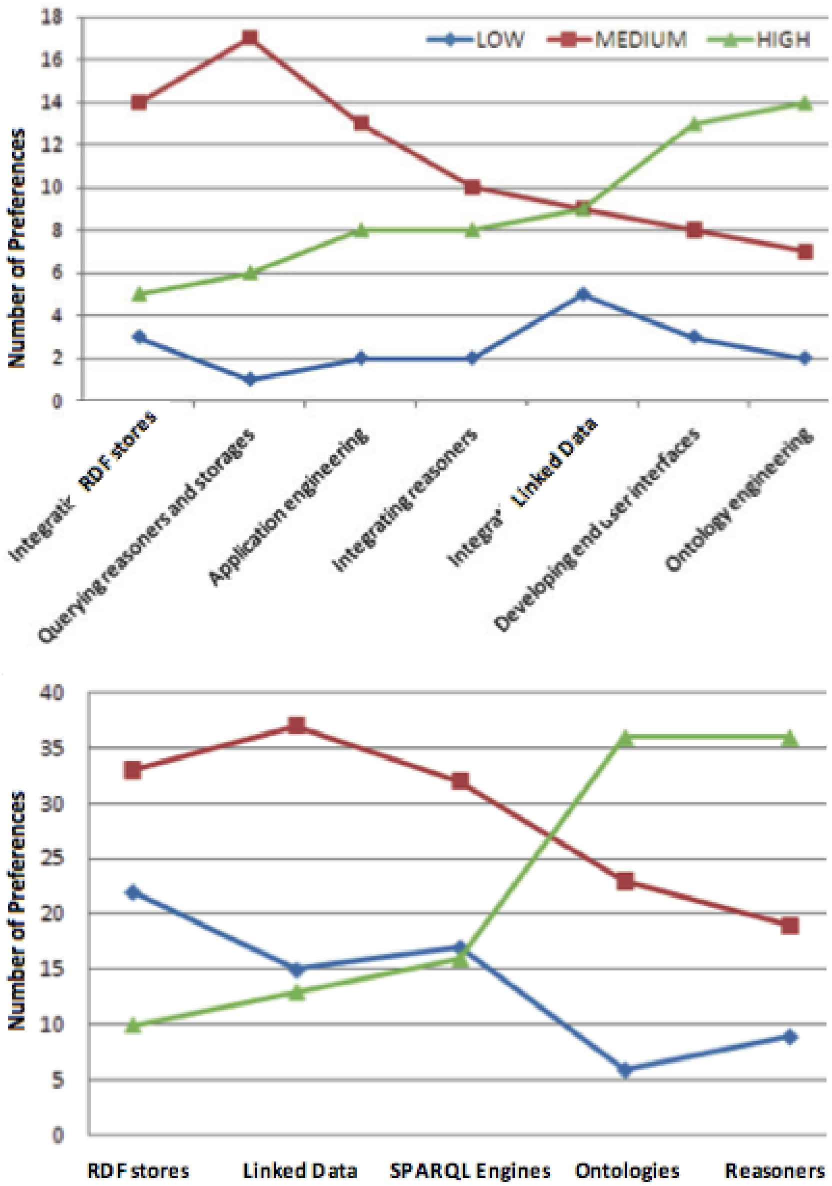

Fig. 8. Effort perceived by the target group of our survey.

6.6 Customer's satisfaction about uses of SWTs.

According to our target, only 48,4% of the customer are aware of the benefits claimed in literature when embracing the development of a SWA. In the beginning of the project only 40% have a positive impression of the SWTs. The good news is that after the project this percentage grows to 70%. These results should rise the Semantic Web community attention on the point of communicating more broadly (i.e., beyond scientific publications) the benefits that SWTs can bring.

7 Conclusions

The SWT industry is rapidly growing, but we believe that a sustainable growth requires mature software development processes.

SWA have their peculiarities, but they are still a software artefacts. If we measure, based on the results of the two surveys, the maturity of the SWT industry in terms of the Capability Maturity Model [18], we asses that it is placed at level 1. At this level, few or no processes are in place, no cost and time estimation tools are used, systematic testing is neglected and success depends on *the heroism of the few*.

We believe that the results of the survey show that the target of our survey perceives the need for an end-to-end methodology. Actually, they tell us that the methodology should come with an appropriate case tool.

The methodology should specifically support the project manager in the requirements gathering, testing and maintenance phases. Moreover, it should include a light-way cost and time estimation tool required to bring the SWT industry to level 2 of the Capability Maturity Model.

We hazard the hypothesis that the best case tool in this historical moment of Semantic Web should exploit component base software development [9]. This would lower the integration barriers, it would assure the correct interoperability of the components in the framework and it can allow for a *light* cost estimation tool based on the selected components. In particular, the components to include in the case are an RDF store, a framework to invoke both local and remote SPARQL Engines, a reasoner and a set of ontologies that can serve as starting point for the development of the ontology at the core of the SWA. Last, but not least, a special attention should be put to tools that support the development of graphic user interfaces.

References

1. T. Baker, T. Heath, N. Noy, R. Swick, and I. Herman. Semantic Web Case Studies and Use Cases. Available as http://www.w3.org/2001/sw/sweo/public/UseCases/.
2. V. R. Basili, G. Caldeira, and H. D. Rombach. *Encyclopedia of Software Engineering*, pages 528–532. John Wiley & Sons, 1984.
3. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web: Scientific American. *Scientific American*, May 2001.

4. C. Bizer, R. Cyganiak, and T. Heath. How to publish linked data on the web. Web page, 2007. Revised 2008. Accessed 22/02/2010.
5. C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.
6. M. d'Aquin, E. Motta, M. Sabou, S. Angeletou, L. Gridinoc, V. Lopez, and D. Guidi. Toward a new generation of semantic web applications. *IEEE Intelligent Systems*, 23(3):20–28, 2008.
7. T. De Marco and T. Lister. *Peopleware: Productive Projects and Teams (Second Edition)*. Dorset House Publishing Company, Incorporated, 1999.
8. J. Domingue and D. Fensel. Toward a service web: integrating the semantic web and service orientation. *IEEE Intelligent Systems*, 23(1):86–88, 2009.
9. G. T. Heineman and W. T. Councill. *Component based software engineering : putting the pieces together*. Addison-Wesley, Boston, 2001.
10. R. Jasper and M. Uschold. A framework for understanding and classifying ontology applications. In *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5): Stockholm, Sweden: 1999, August 2nd*, 1999.
11. K. Kozaki, Y. Hayashi, M. Sasajima, S. Tarumi, and R. Mizoguchi. Understanding semantic web applications. In J. Domingue and C. Anutariya, editors, *ASWC*, volume 5367 of *Lecture Notes in Computer Science*, pages 524–539. Springer, 2008.
12. A. Leger, J. Heinecke, L. J. Nixon, P. Shvaiko, J. Charlet, P. Hobson, and F. Goasdou. *Semantic Web take-off in a European Industry Perspective* . Idea Group Inc., 2008.
13. M. F. Lopez, A. G. Perez, and N. Juristo. METHONTOLOGY: from Ontological Art towards Ontological Engineering. In *Proceedings of the AAAI97 Spring Symposium*, pages 33–40, Stanford, USA, March 1997.
14. C. Mancas and E. Della Valle. Improving the art of semantic web application development - survey 2010. Available as http://applied-semantic-web.org/docs/2010/12/survey2010.pdf.
15. S. McConnell. *Rapid development-Taming wild software schedules*. Microsoft Press, 1996.
16. E. Motta and M. Sabou. Next generation semantic web applications. In R. Mizoguchi, Z. Shi, and F. Giunchiglia, editors, *ASWC*, volume 4185 of *Lecture Notes in Computer Science*, pages 24–29. Springer, 2006.
17. G. Niro and E. Della Valle. Improving the art of semantic web application development - survey 2011. Available as http://applied-semantic-web.org/docs/2011/01/survey2011.pdf.
18. M. C. Paulk, C. V. Weber, B. Curtis, and M. B. Chrissis. *The capability maturity model : guidelines for improving the software process*. Addison-Wesley, Boston, 1995.
19. E. P. B. Simperl, I. Popov, and T. Buerger. Ontocom revisited: Towards accurate cost predictions for ontology development projects. In *6th Annual European Semantic Web Conference (ESWC2009)*, pages 248–262, June 2009.
20. I. Sommerville. *Software Engineering (8th Edition) (International Computer Science Series)*. Addison Wesley, May 2007.
21. Y. Sure, S. Staab, and R. Studer. Methodology for development and employment of ontology based knowledge management applications. *SIGMOD Record*, 31(4):18–23, 2002.
22. P. Tetlow, J. Z. Pan, D. Oberle, E. Wallace, M. Uschold, and E. Kendall. Ontology driven architectures and potential uses of the semantic web in systems and software engineering. W3C Working Draft Working Group Note 2006/02/11, W3C, 03 2006.

23. R. Zlade. Inventing the semantic web ... again. Article on Semantic Universe, 01.20.2009, 2009. Available as http://semanticuniverse.com/articles-inventing-semantic-Web%E2%80%A6again.html.