

A Framework for Reducing Impediments in Agile Projects

Radha Shankarmani
Information Technology
Department
S. P. I. T., Mumbai, India

S. S Mantha, PhD
Chairman, AICTE New Delhi,
India

Dr. Vinaya Babu, PhD
Department of CSE,
JNTUH, India

ABSTRACT

Agile methodologies are at present widely used in industry. Many agile software development metrics are irrelevant to practitioners since they are dependent on the scope or content [Fenton and Neil, 2000]. Currently estimation is done by the team for each of the prioritized features [1] in order to commit to the number of features that can be implemented in each sprint. Each sprint is time-boxed; meaning the time to deliver is fixed. Risk in agile development is failing to deliver planned features in each sprint. There could be many reasons for not being able to deliver. Two of the reasons are discussed in this paper; one being incorrect time estimation and the other reason could be technical impediments. Actually the time taken to complete the task (feature implementation) depends on the proficiency of the development team and there has been no publication on estimation based on how and when to measure this. This is one of the gaps the research paper is trying to address. The scrum master assists the team in removing the impediments that obstructs the progress. In other words scrum master helps in facilitating productivity [2]. Hence, preparedness to resolve impediments as and when it arises is essential. This second risk is addressed by using semantic agents which is discussed in this paper.

Keywords

Agile project; impediments; semantic agents; estimation; ontology

1. INTRODUCTION

Agile project development involves incremental and iterative approach. In scrum, the project progresses via a series of iterations of fixed duration called sprints that get the project closer to the goal. At the end of each sprint, the team has to produce or demonstrate the working, potentially shippable and high quality product increment [11]. That means each and every day of sprint is important. To set the context for each day's work and to help the team to stay on the track, a quick status update is done during the daily stand-up meeting where the team members discuss about accomplished previous day work, today's work to be done and any roadblocks or impediments causing delay in accomplishing committed work.

1.1 Estimation Uncertainty

According to Boehm's "Cone of uncertainty", estimation uncertainty decreases as the project progresses [5]. Todd Little [3] in his paper, brings out the conflict with the Boehm's "Cone of uncertainty" by examining three years of historical project data representing 120 projects developed by a market-leading oil and gas software company. The ratio of actual over estimate according to the data collected follows a log-normal distribution. This behavior and uncertainty remains at every

stage in the project lifecycle. Hence estimation using productivity measure is thought of.

1.2 Agile Risks

When teams have the discipline to look for and track risks and impediments as well as changing to adapt to lessons learned, then they are putting themselves in a virtuous cycle [15]. If they see the risks they have a chance of avoiding the impediment. If they see the impediment, they have a chance to quickly mitigate the impact. And if they take time to learn the lessons from the impediments they encounter, then they are much less likely to encounter those impediments again - and if they do, they will recognize them earlier and be better equipped to deal with them.

1.3 The Scrum Master

The scrum master acts as a facilitator to resolve impediments. For this, scrum master conducts daily stand-up meetings that last for 10-15 minutes to know the status of work. The common pitfalls of daily meeting are discussed in [4]. The most common issues mentioned are either the team members do not want to discuss the difficulties in front of the entire team or the scrum master does not take the initiative to solve the problem in time. This is another point this paper is trying to address by providing a virtual scrum master using semantic agents.

2. USE OF SEMANTIC AGENTS

2.1 Agile in Practice

In agile project management, not much of project related data and documents are stored; they are merely exchanged in daily stand-up meetings [7]. The team shares information by posting on dashboards. The issues faced are later analyzed for understanding the cause of delays during retrospection meetings [8]. Work flow is often stagnated due to lack of resources in the collaborative team [6]. All these issues are very common to agile and can be resolved by using semantic agents in development environment and metadata management.

2.2 Semantic Agents

Understanding the role of Semantic agents in agile Project Management: The semantic agent search extends, but does not replace any search engine.

- It creates an additional layer for extending the query with the keyword synonyms. Look up tables are created for storing the synonyms.

- The main agent passes the updated query to the corresponding Error message agent to retrieve all possible questions and their related answers. The results are displayed using ranking algorithm.
- The ontology layer creates metadata which is used for storage and retrieval of such queries and accepted answers for future search.

The ranking algorithm uses various parameters to sort the most popular (accepted) answers first. This enables the team to share knowledge with each other and execute the tasks with higher efficiency.

In case the query is not a repeated one then answers may not be found in the blogs/forums and agents cannot retrieve answers. In such cases e-Mentoring is suggested. For mentoring to be effective the team member should have a certain level of understanding in the concerned domain. This has to be taken care of during team selection.

A semantic agent is used to select project team which has the closest match to the required skill set for the project [19]. Employees have required skill-set but unavailable for the project can be mentors to available team [13]. Mentors are chosen based on parameters like ranking (success factor in providing solutions), Criticality of the mentor's project and period of association the mentor can have with the mentee.

Using NLP the assist made by mentors can be found. A mathematical model is then used to reward the assist made by mentors for sustainability.

3. METHODOLOGY

The errors and queries of the team can be used to find the productivity of the team which interns helps to estimate the implementation time of the user stories.

3.1 User Story State & Earned Points

Depending on the state of user story where the error has occurred, the points earned by the team is calculated. For example, from specification to customer acceptance testing done in one run – points earned is 6. Other scenarios are described along with points earned are given in the Table 1.

R or FS – Requirements or functional specification; TS- Technical specification

C&UT- Code & Unit Testing; B- Build, FT- Functional Testing; CAT- Customer Acceptance Testing

Scenario 2 – Minor coding changes (no effect on interfaces or system testing) and Scenario 6 – Architecture Error (is less if agile principles are followed)

Earned points are given by seeing the similarity of team performance of football (FPL-Fantasy Premier League) team and agile team. The rules for awarding points in FPL are used to map similar scenario found in agile team [12]

Table 1 User story states and earned points

Scenario	Description	Diagram	Points
1	Correct Reqmts.		6
2	Minor coding changes		5
3	Build error		4
4	Technical Debt		4
5	Rework		-3
6	Architectural Error		-2
7	With e-Assist		3

3.2 Team Productivity

Productivity = size / effort;

We assume the team experts know how to split the tasks so that it's practical to finish within the sprint time box. An error in breaking the task may cause spill over. If the tasks remain beyond the time box then the productivity is less than 100% and the value depends on the amount of remaining work to be done.

Hence the productivity percentage is calculated as 1 minus the remaining percentage of work to be done.

Productivity = 1- Remaining % of work

In general, Percentage of efforts for Code and unit testing (CU) is twice that of each the other phases namely Requirement (Functional Specification from user story- FS), Technical specification (TS-Design + Test cases), Build (B), FT -(Functional and System testing) and Customer Acceptance Testing (CAT)

Total work duration= Σ time for each phase

$$= FS + TS + CU + B + FT + CAT$$

Transforming FS to TS = 1 week; TS to Code = 1 week; Code & UT to Build = 2 weeks; Build to FT = 1 week;

FT to CAT = 1 week. Total = 1 + 1 + 2 + 1 + 1 = 6 weeks

Assumption: Assuming expert performer earns all 6 points per user story the productivity for all the scenarios are calculated and shown in the Table 2.

Remaining work = ((6- earned points) / 6) / (Σ duration for revisited phases / total duration)

If earned points are negative-

Productivity in case of rework = work done + (Rework + Phases to execute / total duration)

Where, Rework = 1,

Work done = (earned points/ total duration) / (duration of phases revisited / total duration)

Productivity in case of Technical Debt = previous productivity + work done

Table 2 Productivity calculation from earned Points

Scenario	Earned points	Remaining work	Work done	Productivity
1	6	0		100%
2	5	((6-5)/6)/(3/6)=0.08		1-0.08=0.92=92%
3	4	((6-4)/6)/(2/6)=0.11		1-0.11=0.89=89%
4	-1		((-1/6)/(4/6))=-0.25	-0.25+1=0.75=75%
5	-3		(-3/6)/(6/6)=-0.5	-0.5+1=0.5=50%
6	-2		((-2/6)/(4/6))=-0.5	-0.5 + 1+ (1/6)=0.67=67%

Generalized formula for calculation of productivity:

Effort is measured in terms of User Story Points (USP) per sprint. The effort required for all User Stories (US) taken for implementation per sprint is given by

$$\text{Total effort in user story point} = TE = \sum_{i=1}^{i=n} USP_i \quad (1)$$

The completed User Stories (DoD) will have j=1 else j=0

Effort for completed user stories is given by E_DoD.

$$E_DoD = \sum_{i=1}^{i=n} j_i \times USP_i \quad (2)$$

$$\text{Total effort for incomplete user stories} = TE - E_DoD \quad (3)$$

User story points are relative estimates for size and complexity. They do not directly relate to number of hours. For example 1 user story point can be 4 hours or 8 hours as desired by the organization. According to the organization experience for each of the scenario, productivity percentage can be varied.

Productivity of all the completed user stories (DoD) remains 100%. For incomplete user stories partial effort has to be calculated:

For this the average productivity of incomplete user stories is multiplied with the sum of USP of incomplete user stories as shown in the formula below

The completed User Stories (DoD) will have j=1 else j=0

$$\text{Avg. productivity} = (1/n) [C \times 100]_{j=1} \times [\sum_{i=n-C}^{i=n} SC_i]_{j=0} \quad (4)$$

- Number of user story completed =C
- Number of user story incomplete = (n-C)
-

Actual effort = Total efforts x Average productivity

$$\text{Actual effort} = AE = TE \times (4) \quad (5)$$

For 100% productivity -> Estimated USP = TE =AE

i.e all the stories are completed (DoD) in time

If in case incomplete user story remains the actual effort is found as shown in equation 5 to be used as the reference for estimating future USP (efforts). If the number of errors in each scenario is more than one during the implementation of user story then the formula can be generalized as shown below-

For 100% productivity:

Estimated USP = Total estimated effort TEE = ACTUAL effort TAE i.e all the stories taken for the sprint are completed (according to DoD) in time.

3.3 Need for Semantic Agent

The survey states over 90% of the errors and issues are repeat ones [21]. Extracting information from the error log can resolve or reduce most of the impediments. The answers for the queries which are not found in blogs or its archives are obtained from experts (e-Mentors). Taking error details as input to the framework estimation, solution, technical document creation and knowledge management can be done.

Removal of technical impediments needs assistance from experts. The solution given by experts is not logged sometimes and also the log is unstructured. All these assists made should be saved in a structured manner and so in future it could be used without waiting for a response from experts!

Most organizations resort to Pair-Programming (PP) to improve quality [9]. PP is followed in companies where number of employees is more and pairing is possible. But in most of the IT companies, team members are allowed to work from home so pairing is not possible. Pairing is the conversation. There is no log of how the errors did get solved. No documentation for future reference. The implementation using multi agent system can solve this problem [16].

The Main agent gets the solution from other semantic agents namely, Error Message Agent and Domain Agent stores the metadata in the repository and fetches the solution according to the request posted in the client interface.

3.4 MainAgent / Virtual Scrum Master

The role of the Scrum Master is completely dedicated to resolving impediments [10]. Virtual Scrum Master accepts the impediment from the team member to resolve and then it checks that whether the impediment is already resolved before. If it is then it will quickly offer the exact problem and its solution and logs that information for future requests. The Architecture diagram of Virtual Scrum Master is shown in Figure 1.

If the impediment has occurred for the first time then with reference to the meta-ontology on the RDFTP Server shown in Figure 2, it identifies the respective agent and forwards the query to it [16].

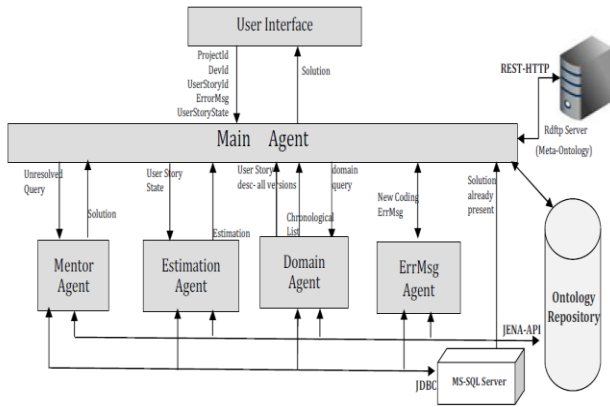


Figure 1: Virtual Scrum Master Architecture

The Virtual Scrum Master would use the meta-ontology shown in Figure 2, to identify the agent depending on the type of impediment.

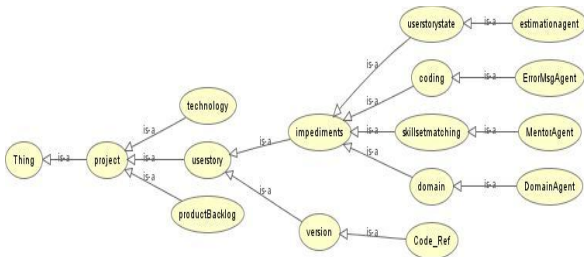


Figure 2: Meta-Ontology on RDFTP Server

If the query is related to the coding error then it forwards it to the ErrMsgAgent that will guide the developer to refine his query which may suggest the root cause of aroused error/exception which can be attributed to many causes. Then it provides the possible ranked answers. The Virtual Scrum Master would then update the ontology repository [17] with all the semantic information about that impediment for future use.

Ontology Repository

Ontology Repository is referred by the agents to get all the relevant information about an impediment raised by the team member. The meta-ontology for user story versions and Meta_ErrorMsgQuery Ontology are shown in Figure 3 and Figure 4 respectively.

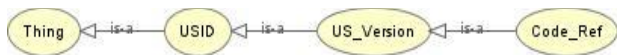


Figure 3: Meta-Ontology for User story versions

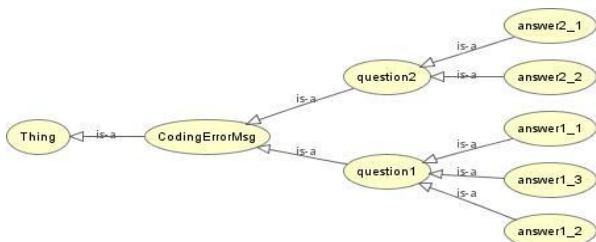


Figure 4: Meta ErrorMsgQuery Ontology

Log Analysis:

All the queries along with their solutions are saved as shown in Question Table 3 . Also all the queries / errors related to each user story (along with the user id) is stored in a sequence (timestamp) in the log table as shown in Table 4 below.

Table 3 Question table

Qid	Query	Is Answered	QSolution

Table 4 Log Table

USId	QId	QSolution	tmstmp

4. IMPLEMENTATION

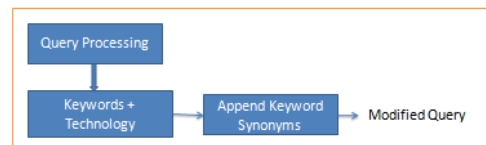
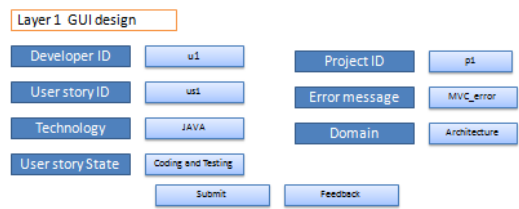
4.1 Framework for reducing impediments in agile projects using Semantic Agent

Taking error details as input, the framework to reduce impediments provides sprint estimation, solution to errors, technical document creation and knowledge management. The layered architecture diagram for implementing the multi-agent system to reduce the impact of impediments is shown in the Figure 5.

4.1.1 ErrorMessageAgent

In case of any coding error impediment, the developer may search the solution at different sources or blogs over the Internet where the results optimized aren't based on problems but other unrelated factors like page visits and keyword count. Older pages or search results that aren't on top aren't usually checked. Getting answers to these questions is very time consuming.

If it is a coding error impediment encountered for the first time, the MainAgent/VSM forwards it to the ErrorMessageAgent. ErrorMessageAgent crawls the Internet for various blogs, tutorials, documentation and searches for the exact questions and their answers related to that error message and then it generates an ontology that represents this semantic information about the query error message.



Layer 2 Main Agent _ Query Modification

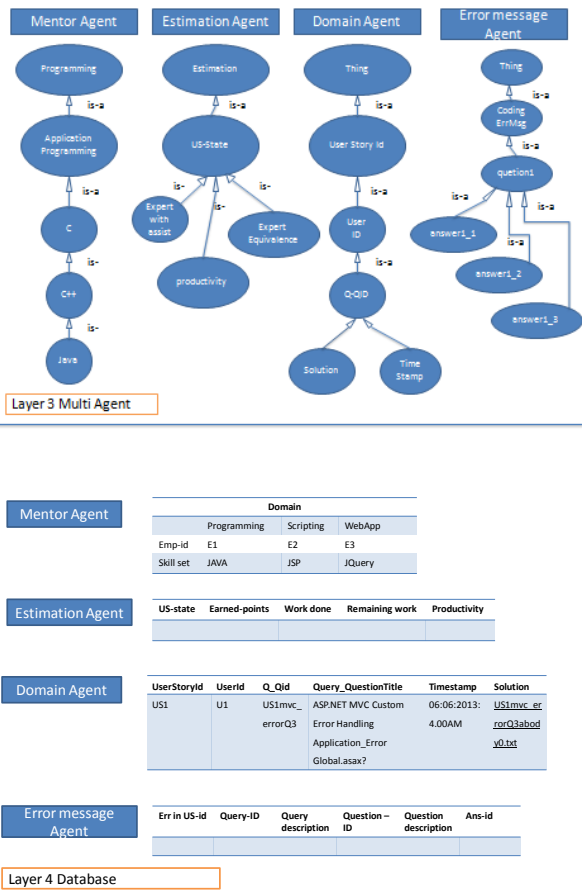


Figure 5: Framework for reducing impediments in agile projects using Semantic Agent

The ontological representation of the query is stored in the ontology repository. Then this agent intelligently goes on asking for a question and suggests the possible solution.

Relevant search results are extracted and their data and other meta-information from the selected trusted sources are recovered. Answers are then sorted according to the sorting algorithm using this information and arranged accordingly. Arranged results with the content of the solution to the problem are returned in JSON format. This is consumed by the different types of clients that make calls for this resource and the results are obtained by the end user.

Semantic Working:

The semantic word search is based on a predefined and a customizable dictionary lookup table. Each entry consists of a list of words that have similar semantic meaning and are interrelated to each other. When a query is fired, the keywords in the query are split and each of the words is compared with the existing lookup table. If a match exists, the other words that are related to the word queried are appended to the search string. This is done for every word in the original search query.

Priority Algorithm:

For sorting answers, a simple priority algorithm is used where points are awarded for certain parameters and the answers are sorted in a descending order at the end. The algorithm works on a weighted basis, where the weights can be customized by the user of the application. These weights are multiplied to the actual parameter on which the ranking is done and added to the sum total for a particular answer.

The parameters on which this ranking is done are the position of the answer in the generic search result, Question score depends on how many users up-voted the question, Answer score is that how many users up-voted the answer, Comment score is that how many users up-voted comments

If the question is actually successfully answered, if the answer is popularly accepted, Accept rate of the user is that how much relevant the user's questions are?

Reputation of the user is that what is the skill or the expertise level of the user, Page views is that how many people have faced the same / similar problem. More the page views, more likely is the problem and more commonly faced it is.

4.1.2 Domain Agent

This model automatically creates technical documentation using semantic agents in addition to online assist. This model works well with agile methodology since each piece of information is stored to support a help system which the developers refer to first when they are to solve an error. This content can be arranged domain-wise, UserStory-wise, developer skill-set-wise and accessed accordingly. This reusable content with additional conceptual information can complement existing help system content. The ontology [14] shown below help in finding the reusable code that is most appropriate for the business scenario in hand.

Alternately, the domain agent can assist the error message agent in refining the query posted by the user. The user will enter the query then domain agent will generate the keywords and append to query and the following steps are done:

ErrMsgAgent will use that appended query and search for answers using ontology file as shown in Figure 6.

In the ontology folder all the owl files are saved. We can search for the match of an ontology query name and filename. As it is a ontology, we can search for keywords in that file.

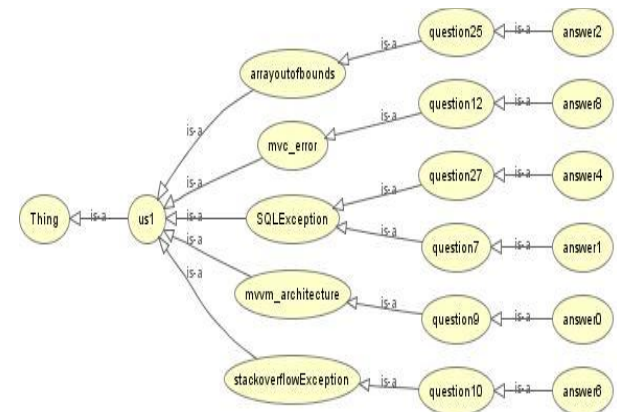


Figure 6: Log Ontology

The error log of the particular user story can link all the errors from its root (first error) till DoD of that user story as shown in Figure 7.

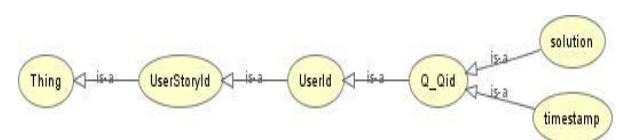


Figure 7: Meta-Ontology Log

4.1.3 Agile Requirements: User story versions

Agile requirements are indispensable, urgent, and barely sufficient but developed in small pieces. They keep on evolving as software is developed.

Requirement specification discussion between the Developers and the Product Owner has a time span; after this time span specification freeze. No more requirements from the Product Owner are considered for that release. If any, they are considered for a later release. We include such cases as different versions and add it to our ontology as shown below.

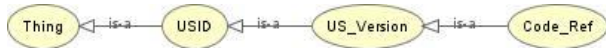


Figure 8: Meta-Ontology for User story versions

5. RESULT ANALYSIS

5.1 Error Message agent (Intellisense)

5.1.1 Comparison with Google

ErrorMsgAgent assists user in identifying the exact question and provides correct answer in seconds while if the same question is posted on Google it could not provide the answer.

5.1.2 Comparison with Stackoverflow blog

Time saved in browsing for the correct question using Error agent (Intellisense) is the time consumed for getting the correct URL. For example, consider the Query: how to toggle a jquery?

Table 5 Manual Search time

Question No.	Description	Page No.	Position (From Top)	No. of url visited	Time Consumed (In min)
1	jQuery text() change on toggle()?	5	3	5	35
2	How to toggle (jquery) with an anchor	2	2	2	20
3	How to toggle attribute in jQuery	1	11	1	9
4	how to use JQuery "toggle" with parameters?	14	14	15	155
5	How do you create a toggle button?	20	4	21	221
6	How to reset toggle in jQuery?	1	14	1	9
7	How to Toggle in jQuery	1	13	1	10
8	How to control jquery toggle?	1	15	1	8

Search time for manual search in stackoverflow.com was recorded and shown in Table5.. Time taken for automated search is calculated using the E-crawler performance metrics published by Oracle [20] and shown below.

After fetching the results from keyword search, the time taken to find the required result depends on

- the page in which the result is displayed,
- the position of the URL in that page,
- average time taken for processing URLs (183ms),
- average time taken for filtering a document (253ms),
- average time taken for fetching the document from cache memory (89ms).

The average time taken for the above said factors is referenced from oracle documentation [20].

Average number of results (URLs) displayed in a page in Google and stackoverflow.com is 10 and 15 respectively

Number of pages visited be = N

Position of URL found in that page be= P

If the desired document/answer found in the first page then, total time taken will be = (Avg. time taken for processing URL + Avg. time taken for filtering a document) x Number of URL visited.

For simplicity we take Number of URL visited = Position of URL found in that page

If the desired document/answer found in any page after the first page then, total time taken will be = (Average time taken for fetching the result from cache memory) x (Number of pages -1) + (Avg. time taken for processing URL + Avg. time taken for filtering a document) x (15 x (Number of pages -1) + Number of URL visited).

Sample calculation for the questions presented in Table 5.1 using the above formula is shown as follows:

$$Q1 = 89 \times 4 + (183 + 253) \times (60 + 3) = 27824 / 3600 = 7.7 \text{ min}$$

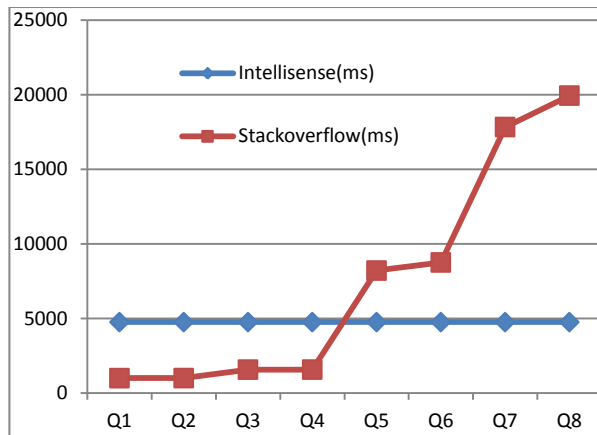
$$Q3 = (183 + 253) \times 14 = 436 \times 14 = 6104 / 3600 = 1.7 \text{ min}$$

$$Q4 = 89 \times 13 + 436 \times 224 = 98821 / 3600 = 27.4 \text{ min}$$

The stackoverflow.com blog also could not identify the proper question and answer. The following questions were posted on the blogs due to unavailability of answers which took long duration to get answer. The same eight queries specified in Table 6 is used to compare results with stackoverflow.com. and time taken (in msec) is shown below

Table 6 Compare results with stackoverflow.com

Query	Intellisense(ms)	Stackoverflow(ms)
Q1	4760	1006
Q2	4760	1009
Q3	4760	1577
Q4	4760	1578
Q5	4760	8216
Q6	4760	8759
Q7	4760	17833
Q8	4760	19940



	Intellisense	Stackoverflow
Total Execution time	4760ms	23837ms
Total Memory Utilized	0.23MB	0.25MB

From the above readings, it is justified that Intellisense is giving better performance

Time taken is : 4760.0ms

Used memory is bytes: 241768bytes

Used memory is megabytes: 0.23056793212890625MB

If the solution is not found on Blogs we normally end up writing a post on blogs and depend on other programmers to give us an answer who may have faced a similar problem. We also found the above queries were already posted on the blogs but users are unable to find them as they are archived in blogs and there can be Query Mismatch. The number of times that same query is posted are shown in Table 7.

Table 7 Time taken by blog to post the answer

Problem	Time taken for correct answer	Times posted
Problem displaying custom error page running Tomcat 7 in Eclipse Helios	1 month 15 days	5 Times
How to run jBoss AS 7 with eclipse 3.6(helios)?	29 hours	9 Times
Unable to run Apache server giving error ports already in use	25 hours	11 Times

Agents in turn embed semantics in the search query so that the results can have an extended coverage of relevant answers. Have a look up table with words (thanks, great) which reflect answers were useful. Taking a lot of search results and sorting them using various methods to return the best possible solution to the answers. Weightage can be given to answers as follows; Up votes for questions as well as answers, say 10 points; Page views, say 1 point; Reputation of the author of the answer, say 2 points; Correctness of the solution based on semantics, 2 points. Log is generated when the user submits the feedback.

This agent is uploaded on cloud to enable the developers use it and give their feedback. The questions are taken from StackOverflow itself as they were asked during the time of research and tested again their own and IntelliSense searches.

Table 8 Query from StackOverflow

N o.	Query	First Search	Intelli Sense	Result Analysis
		StackOverflow		
1	fit image to screen properly in javascript	http://stackoverflow.com/questions/17761206/fit-image-to-screen-properly-in-javascript	Relevant	StackOverflow leaves the question unanswered.
2	How to play embedded youtube link by using webview in android	http://stackoverflow.com/questions/17761094/how-to-play-embedded-youtube-link-by-using-webview-in-android		fetch the answer, answered much previously
3	Content Assist Does Not Work	http://stackoverflow.com/questions/17761342/content-assist-does-not-work	Correct result	StackOverflow first few results doesn't have answers at all.
4	Facebook plugin for Phonegap errors	http://stackoverflow.com/questions/17761067/facebook-plugin-for-phonegap-errors	Relevant	First result does not have an answer.
5	ViewPager implementation	http://stackoverflow.com/questions/17761251/viewpager-implementation	Correct result	No answers from StackOverflow search in the first few results.
6	static web method - the risk of shared values	http://stackoverflow.com/questions/17761230/static-web-method-the-risk-of-shared-values-between-users	Correct result	No answer from StackOverflow
7	jQuery validation plugin and ajax post before validating the input	http://stackoverflow.com/questions/17762304/jquery-validation-plugin-and-ajax-post-before-validating-the-input	Correct result	First result doesn't have an answer.
8	how to deploy web service on the web	http://stackoverflow.com/questions/12137191/how-to-configure-the-web-service-wsdl-url	Correct result	First result doesn't have an answer. Also, the question returned is irrelevant.

6. CONCLUSION

The thesis presented is towards identifying risks due to impediments in agile development process and a methodology to address them. The first risk factor is the deviation in estimation [18]. To correct the estimation the parameter suggested is phase of the lifecycle in which the error or issue happened is taken. A mathematical model is suggested to find the productivity (productivity is taken as 100% in case the user story implementation is completed in time), the team's

capability of solving the problem is understood and accordingly further sprints are estimated. The other way is to find team expertise (expert equivalent) to estimate effort.

The second risk factor is technical impediment. If the team does not have subject matter experts then the query is posted in forum/blog. Writing the query correctly can bring the relevant answer from the blog achieves. Semantic agent technology is used to modify the query so that all relevant questions are displayed along with the answers for each of the question. Ontology is used to save the same in a structured way for future reference. The ranking algorithm is developed by awarding points to the query presenter, the expert who gives the answer and the feedback on the answers if presented earlier. Hence this agent works better than any blog or Google. The results to the query by using the agent (intellisense), Google and the popular blog “stackoverflow.com” shows that the answers that were present in “stackoverflow.com” but could not be retrieved (since they were stored under a different category) were also retrieved by the semantic agent. This agent is uploaded on cloud to enable the developers use it and give their feedback (<http://intellisense.cloudapp.net/Answers/>)

In case the error or query appears for the first time, then the experts with that technical skill are identified by semantic agent and the error message is forwarded to them. These experts act as mentors.

Additional benefit: The error log constitutes a good share of technical documentation.

7. REFERENCES

- [1] [http://msdn.microsoft.com/en-us/library/hh273052\(v=vs.88\).aspx](http://msdn.microsoft.com/en-us/library/hh273052(v=vs.88).aspx)
- [2] www.scrummethodology.com/the-scrummaster-role/
- [3] “Software Schedule Estimation and Uncertainty Surrounding the Cone of Uncertainty,” T. Little, IEEE Software, May 2006, Vol. 23, No. 3
- [4] <http://guide.agilealliance.org/guide/daily.html>.
- [5] Glen B. Alleman,” Agile Development Versus Traditional Development”,2009, http://herdingcats.typepad.com/my_weblog/2009/06/agile-development-versus-traditional-development.html
- [6] <http://xp123.com/xplor/impediments/>
- [7] Mountain Goat Software ,“The Daily ScrumMeeting”, <http://www.mountaingoatsoftware.com/scrum/daily-scrum>
- [8] Scrum Guides ,“The Official Scrum Rule book” , <http://www.scrum.org/Scrum-Guides>
- [9] Extreme programming ,“Daily Stand Up Meeting ”, <http://www.extremeprogramming.org/rules/standupmeeting.html>
- [10] Scrum methodology, “Scrum Impediments”, <http://scrummethodology.com/scrum-impediments/>
- [11] Joe Little,” The Basics of SCRUM ” May 30th, 2011
- [12] <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6139390&queryText%3DPerformance+assessment+of+ASD+team+Using+Football+Rules+As+Reference>
- [13] http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5552340&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D5552340-- A decision support system utilizing a semantic agent.
- [14] Protégè OWL : <http://protege.stanford.edu/overview/protege-owl.html>
- [15] Mike Griffiths,” The Top Five Software Project Risks”, <http://www.projectsmart.co.uk/top-five-software-project-risks.html>
- [16] Vasant honavar,” Tutorials- Intelligent Agents and Multi-Agent Systems”, IEEE Conference on Evolutionary Computation (CEC), 1999
- [17] Jin Pan, Stephen Cranefield and Daniel Carter, “A Lightweight Ontology Repository”, International Foundation for Autonomous Agents and Multiagent Systems, AAMAS'03, 2003
- [18] http://info.thoughtworks.com/rs/thoughtworks2/images/twebbook-perspectives-estimation_1.pdf
- [19] Colucci, S.; Di Noia, T.; Di Sciascio, E. ; Donini, F.M. ; Mongiello, M. ; Mottola, M“ A Formal Approach to Ontology-Based Semantic Match of Skills Descriptions” Journal of Universal Computer Science Vol. 9, No. 12 2003,pp. 1437-1454
- [20] http://docs.oracle.com/cd/E35215_01/admin.11222/e35070/crawler_performance_metrics.htm
- [21] Jørgensen, M. "A Review of Studies on Expert Estimation of Software Development Effort"