Inclusion of e-Assist to increase Agile Adoption

Radha Shankarmani[1], Dr. S. S. Mantha[2], Dr. Vinaya Babu[3]

[1]Information Technology Department, Sardar Patel Institute of Technology, Mumbai, India
[2]Chairman, AICTE, New Delhi, India
[3]Principal & Professor of CSE, JNTU, Hyderabad, India

Abstract- Agile methodology promises quality deliverables in frequent intervals. In Agile project development, user stories (requirements) entered in Product backlog is prioritized and according to priority few of them are selected for the current iteration (sprint). These user stories have to be implemented within that sprint. Hence the project team demands experts in the relevant domain for successful completion. Software development organizations have limited number of experts with them and hence the number of projects that follow agile methodology is restricted. This paper first discusses the risk involved if moderately skilled members are chosen for agile software development and then the methods to mitigate the risk so that more number of projects can adopt agile methodology

Index Terms- Agile adoption; Scrum; Risk; Mentors; Expert

I. INTRODUCTION

In software development change in requirements are inevitable. Changes can occur because of business conditions, global market, and technology growth to name a few. In traditional software development methodology, risk management is based on avoidance of requirement changes whereas agile methodology adapts to requirement changes. In agile the requirements are taken as user stories. All the user stories are added to product backlog and are prioritized. According to the priority user stories are taken for implementation in the current iteration (sprint) planning. User stories low on information is not taken into the current plan. The benefit of agile is early delivery of partial product to customers. Customer feedback is taken once a week to incorporate changes. And so "Defects" in requirement are found early in agile and late in other software development methodologies which make it expensive to fix.

Defect leads to rework. The metric of rework is defined as time spent on fixing the defects as a percentage of time spent on its development [3]. The survey says amount of rework generally lies between 20 to 30% in other methodologies [4] whereas in agile the amount of rework is much less than 20%. Agile also reduces retrofit because the customer representative (a.k.a. PO-Product Owner) is part of the agile team and PO is well informed about what the end product is going to be, at the time of building the product itself. Therefore "good agile practices" not only avoid rework but also reduces retrofit.

II. CHALLENGES IN AGILE ADOPTION

In the beginning Agile has a lot of challenges because of uncertainty and so most of the time Software quality is not visible in the early sprints [1]. Agile is self-organizing hence as the team progresses towards more sprints, it learns and adapts in attaining the required quality. Code testing cannot find all the defects because defects can exist in the product, specifications, design and plans. So other testing methods need to be used to find defects as soon as they are created.

The "test first" concept in agile makes the Quality Assurance Analyst work with Product Owner to analyse the user story point in many ways and inform developer where the code might break even before the developer writes the code. Everyday code is run through "automated build and test system" to verify whether it runs correctly on the required operating environment. During the sprint release if the code does not have the required quality it is placed in the product backlog and taken in the next sprint. The team collects data about risk mitigation over the course of sprints, and work out their strategy for managing risks more effectively.

Scrum is one of the agile project management techniques. Scrum insists on daily stand-up meeting to help the team identify the road blocks (impediments) to the work completion planned for the previous day. It is the duty of scrum master to get the impediments resolved. All these factors make agile follow inherent risk mitigation methods.

Strong communication exists between developers and stakeholders from the start of the project. Agile process consists of design and development, daily discussion of road-blocks (impediments) to development if any, deliver completed work to stakeholder and get their weekly feedback, define the work and deliverables for the next week and repeat the entire process [9]. Early release of partial product for customer evaluation is the biggest positive about agile methodology. For early release the necessary development and test environment is mandatory.

Whether it is traditional or agile the challenges in project management related to people are (i) effect of human error (ii) departure of key personnel (iii) impact of incompetent personnel [2]. In agile all the above have to be rectified without exercising control because the principle of agile is "people over process" and having a "self- organizing team". Needless to say agile ensures that the team has all the necessary skills for successful completion of the project.

Source of barriers leading to risks in agile discussed here are: (a) Lack of development and test environment, (b) Team members are neither expert in their domain nor generalists (take multiple roles). For example, developer takes the role of a tester for a peer developer. Having generalist in the team without proper guidelines can lead to increased risk as they tend to create more technical debts (extra efforts required to complete the work in design or code). The reason behind this is, most of the code is peer-reviewed and few are unit tested (with no marking on which

of them are unit tested!) and so there are chances that defects may go unnoticed in review.

(a) Lack of development and test environment

Deployment testing which is testing the infrastructure or operating (production) environment for the application takes almost the same effort as coding. Design patterns are chosen taking into account the scale of operation of business or business condition, the technical constraints, skill set of the team etc. Infrastructure used for early integration testing is smaller than those required during production. The early sprint releases may fail in later sprints during build.

Solution: Infrastructure based on virtualization platforms introduces the possibility of environments that can be re-scaled based on the needs [5]. Build if it happens on cloud can expedite the integration process and the commitment (also investment) to the architecture need not be done initially as architecture maturity comes in later sprints. Developers can take build images existing on cloud and test [17].

(b) Team members are neither expert in their domain nor generalists.

Agile has small time-boxed delivery, say 4-6 weeks. Hence there is no time to train people if they lack required skill. To develop people into being better than they were, a generic strategy need to be followed to suit everyone's personality. Some work under pressure, some work if patted on their back and some are good if they are left alone. Apart from this the team has to be risk-taking and need to know how to deal with failure.

A. Quality Metrics in Agile

Quantitative metrics in Agile are velocity, burn-rate, defects, unit test coverage, and technical debt, work in progress and story cycle time. Velocity is the measure of amount of work completed in that sprint and story cycle time is stories started in one iteration and completed in a subsequent iteration. [8].

Solution: Knowledge sharing and networking helps to a great extent In order to reduce the story cycle time, e-Assist or e-Mentors can be included in the project.

Qualitative Metrics: Propositions are made and stated as Hypothesis to increase agile adoption. If experimentation conducted gives an expected outcome or the required credibility then hypothesis is not rejected [6].

Hypothesis 1: Increase agile adoption with e-Mentors if the reason for low agile adoption is "enough experts are unavailable".

Hypothesis 2: Select team with closest skill if experts are not available. Identify developers with closest skill since it is easy to train them to the required skill - Comparison of known technology to unknown and delivering required material in order to raise the level of average developer to expert level. This also increases work satisfaction of developer, creates a sense of achievement and in a way reduces attrition. An increase in IT assets is an additional benefit.

Hypothesis 3: Provide online resources to suit multiple roles – Developers will have more test-cases when they play the role of a tester to find errors in unit testing. An integration environment set up is done on cloud and code is run there to find build errors. Use of patterns can reduce the architectural errors.

Hypothesis 4: Use e-Mentors to resolve impediments. The experts having required skill but unavailable can act as e-Mentor from outside the team. Their identity can be withheld and bonus points can be given to them for sustaining their assists. This has two benefits there will not be any ego clash since the identity is not revealed and bogus request will not be created just to earn points.

Hypothesis 5: Product owner to work with QA to refine user stories – product owners when they work with QAs can find all possible scenarios of user story, a kind of test harness can be done.

Hypothesis 6: Additional resources by way of e-Mentors can reduce the effort in rework.

Hypothesis 7: Recognizing the assist by mentors to create team spirit. The online mentor can be one within the team or from any other team to help in solving the impediment. In agile code is owned by the team and not by an individual. Performance assessment in agile can be twofold one for the individual task completion and second for the assist made to their team

Improving Quality using e-Assists in Agile

Create / provide resources as references to assist the project development team to have a better understanding of the domain, to foresee issues related to customer and tools, to solve technology related issues.

Improving Quality using e-Mentors in Agile

The subject matter experts (SMEs) who are presently busy with other projects but have the required skill set for this project can act as an e-Mentor.

III. METHODOLOGY

Acquire resources and extract information using NLP, semantic tagging etc. Create patterns using error / defect – log, customer related UI specifics, best practices, lessons learned from retrospection meetings. Knowledge from this can be shared and reused. Have a support team to find patterns, tool compatibility, customer related issues etc. An alternate way is to automate the process using semantic agent.

Encourage experts to register as e-Mentor. Assess the assist and award them accordingly to sustain their involvement. Identify mentors based on three factors (i) Rank, depending on the number of times he /she solved the query (ii) Criticality of the project in which the mentor is currently working and (iii) Period of association, the time he/she has to complete the ongoing project. If the time is more, they can render more help [10].

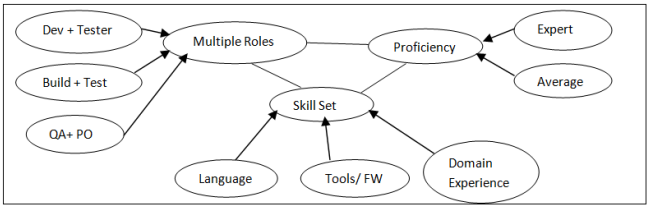


Fig. 1 Attributes of team members

Fig. 1 shows various attributes of team members namely - Skills, Proficiency and multiple roles. Skill set required for a

developer are implementation language, framework, development tools, domain expertise (experience in the domain). The proficiency level in these skill set can be average to expert. Developer needs to play multiple roles, for example, test peer code; a QA can verify completeness of user story (use case in OOAD terminology).

Agile works well if the team has

- Proficiency in skill-set -Number of experts to average has to be a minimum ratio of 2:1 in the team[16]
- Team selection -Team has to have experts in language, domain, framework and tools
- Team member can play multiple roles.

Agile team normally has 7 ± 2 members. Let us take the team consist of six (PO, Architect, 3 Developers and a QA) members. Traditionally out of six there has to be minimum 4 experts and maximum 2 moderately proficient members. Assuming the expert team completes the user story implementation for release in one sprint. A moderately proficient member completes the user story within one sprint duration or later depending on where from he/she has to rework. Velocity is a measure of the amount of work the team completes in one sprint [7]. Story Cycle-Time, will expose problems with stories started in a sprint and completed in a subsequent sprint [8]. The phases of user story implementation are given in the Fig. 2 below.

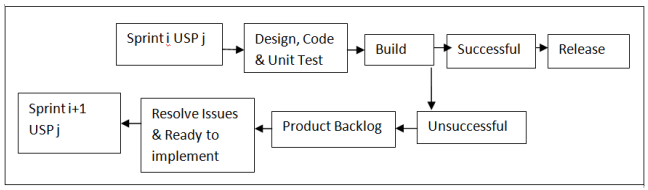


Fig. 2 Phases of User Story implementation

A. Sample Calculation to find expert equivalent for 4:2

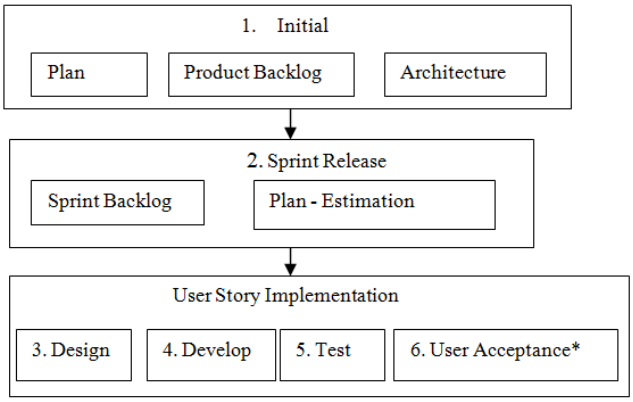
The six phases of testing required in agile software development from user story prioritization in Product backlog -> sprint backlog -> user story implementation (Design, Code, Test) to release of that user story as shown in Fig. 3.

In phase 1 User story description, prioritization and initial architecture are tested. Changes in user story description can change the initial architecture.

In phase 2 estimation of current sprint (user stories) is done. Since agile is time-boxed, if estimation goes wrong story(s) might remain incomplete at the end of the sprint.

In phase 3 design and phase 4 coding have to be tested, both at unit level and build level by peers and then in phase 5 by quality assurance person.

Phase 6 demands the approval of Product owner who is the customer representative. Then the user story implementation is said to be accepted (Definition of Done – DoD in scrum) [19].



*Incomplete story is added to the product backlog and included in the next or later sprint.

Fig.3. Six levels of Testing required in Agile Software Development

Assuming an expert gets a 6 on 6 (100%) on user story completion within time-box. For EVERY moderately proficient member, effort depends on his/her rework. Agile methodology needs 4 experts (minimum) + 2 average (maximum) members. The last two scenarios appear if the two average members are Product Owner and Architect.

Table I find EXPERT EQUIALENT OF TEAM

Rework From	**Rework To**	**Team Expertise**	**Expert Equivalent**
Code & test	Debug/Refactor	(5/6) * 2 + 4	5.7
Build	Debug/Refactor	(4/6) * 2+ 4	5.3
Build unsuccessful or system test fails	Cause identified/ Resolved, implementation completed in next Sprint	-(1/6) * 2+ 4	3 .7
Implementation issues	Unresolved, Added to Product Backlog and taken in later sprints	-(2/6) * 2+ 4	3.3

B. Sample Calculation to find expert equivalent for 3:3

Instead of 2:1, assume1:1 ratio initially i.e., 3 experts and 3 average + e-Mentor to Assist (adding e-Mentor can reduce the rework efforts to half, therefore 0.5 factor is added for each type of assist viz., code, test/tool and domain are added). The factor of 0.5 was taken as it was seen with reference code students' effort in programming reduced to half. Further experimentation needs to be done assuming the effort will follow normal distribution; the standard deviation can be taken as a correction factor.

Table 2 fIND EXPERT EQUIALENT OF TEAM

Rework From	Rework To	Team Expertise	Expert Equivalent
Code & test	Debug/Refactor	(5/6) * 3 + 3+0.5	6
Build	Debug/Refactor	(4/6) * 3+ 3+1.0	6
Build unsuccessful or system test fails	Cause identified/ Resolved, implementation completed in next Sprint	-(1/6) * 3+ 3+1.5	4
Implementation issues	Unresolved, Added to Product Backlog and taken in later sprints	-(2/6) * 3+ 3+1.5	3.5

C. When an Employee Leaves the organization, it is important to capture the Knowledge he had diligently developed during his tenure[15]

Brain drain is what happens when the talent that has been so diligently cultivated leaves the organization. Capturing valued employee's critical knowledge can play a key role in increasing project success. Knowledge has to accessible from anywhere hence the learning repository can be placed on cloud. By using knowledge capture efficiently training costs can be reduced. This can also supplement job-shadowing and to some extent peer to peer mentoring. Capturing useful information can bring knowledge about platform, content and solutions together. Knowledge capturing is considered to be extremely useful for creating seamless employee transition.

About 80% of learning comes through peers [18]. This centralized repository of learning material can bring the assistance of globally located peer group to the team members.

IV. IMPLEMENTATION

o implement the methodology in order to increase the suitability of agile adoption for candidate project by providing e-Assist / e-Mentors in the form of semantic agent is given in Fig. 4. Agile principle stresses on face to face communication. The references provided by e-Assist and e-Mentor are no substitute for face to face communication rather it creates confidence in a team member during daily standup meeting. This is very useful especially if the team is distributed where face to face communication is not possible.

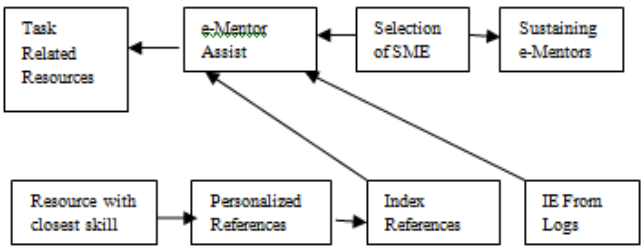


Fig 4 Solving impediments using semantic search

The role of the semantic agent is to analyze and collect relevant resources suitable for the agile team for executing their task successfully in the given time period. Normally team is selected based on the skill set and proficiency required for the project in hand. Agile projects require highly proficient skill set since it commits to quality product delivered at regular intervals. The resources who have the required skill and proficiency but are unavailable can act as an external mentor to the team. In this paper we have suggested to select moderately proficient members and provide them enough assistance to make them work like an expert. The reference material to be provided for them can be in the form of documents related to domain, tutorial and demos for tools, sample and reusable code for technology. Depending on the chosen members' skill set and proficiency various references are attached to the task. Apart from the references for domain, tool and technology QA can provide test cases and test data which will help them do peer testing. Understanding the domain with the given resource material can make them derive test cases on their own. Further all these customized references can be saved and indexed for future use. Thus searching time for the second time for similar requirement is reduced.

Web logs (blogs) have abundant treasure of hidden information for most of the queries that a developer comes across. Content summarization of the relevant solution for a query can be done from the blog archives and can be used as a reference to solve technological impediments. This can definitely save a lot of time during implementation. Recurrent use of the same solution to the problem can help to create patterns. Using natural language processing (NLP) we can rank the bloggers who give the best solution and add it to the list of experts.

Agile does not recommend heavy documentation. Hence not much of structured documentation is available in case of agile projects. The documentation that gets developed by way of logs is unstructured. NLP is used to extract information from unstructured data.

- Goal: Analyze and collect the available organizational resources (team, reference material, etc) against the required tasks to be implemented.
- Select from available resources, the ones with closest skill set and proficiency using ontology. Calculation of sim_c and sim_p [11][12] can be used for this.
- Generate reference resources (e-Assist) from available documents and logs to resolve impediments. Information from logs can be extracted using GATE (General Architecture or Text Engineering) [13].

- Personalize references according to proficiency, skill-set and role. These reference documents can be stored in a hierarchical tree and tree search can be made faster using signature graphs [14] or ID3 data mining algorithm.
- Contact e-Mentor for solution (provider's identity withheld). A model to reward e-Mentor for sustainability can be developed.
- Capture SME's (subject matter expert) critical knowledge to prevent brain-drain [15].

A. Selection of team with closest skill[20]

Ontology based approach is used to calculate the similarity between two items and retrieve all relevant items. Here ontology is used for storage of actual skill profiles and reference profiles to gain perfect or nearly perfect profile matches. Sample skill – ontology created using Protégé is shown in Fig. 5.

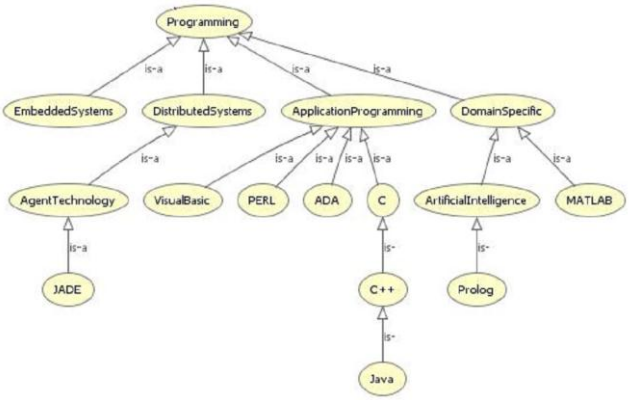


Fig. 5 Sample skill-ontology created using Protégé [20]

Actual skills required for the project is given in the Table 3 below followed by

Table 3 actual skills required for the project

Domain	**Skill**	**Competency**
Programming	JADE	5
Programming	C	5
Programming	C++	5
Scripting	JSP	5
Web Application Design (WAD)	X Query	5
Web Application Design	XPath	5

Table 4 Actual Snapshot OF EMPLOYEE DATABASE

Employee ID	**Domain**	**Skill**	**Domain**	**Skill**	**Domain**	**Skill**
E1	Programming	C, C++	Scripting	Python	WAD	XSD
E2	Programming	C++, JADE, C	Scripting	JSP	WAD	X Query X Path
E3	Programming	C++, Java, JADE, C	Scripting	JSP	WAD	X Query X Path

The similarity measure is given by
$sim_c(c1,c2)=1-d_c(c1,c2)$ [12]

The competency level similarity is determined by the following formula [12]:

$sim_p(cl1,cl2)=1-\alpha(cl1-cl2)$ when $cl1-cl2\geq 0$
$sim_p(cl1,cl2)=1$ when $cl1-cl2<0$

where $0\leq\alpha\leq 0.25$. Using the above formulae on the sample database, we get the following result shown on the Table 5 below.

Table 5 Calculation of sim_c*sim_p

Employee ID	sim_c	sim_p	sim_c*sim_p
E1	0.617	0.6	0.37
E2	1	0.8886	0.886
E3	1	0.8	0.8

The employees having high sim_c*sim_p value are the ones who have the closest skill and proficiency, hence they are chosen for the team.

B. Personalize Resources and Indexing them

Using NLP libraries and NLTK toolkit, a list of keywords and association of keywords table are made. Log traces of references used by experts are used to find the relevant material for reference. If an average skill person does not find it adequate he/she can add more. Semantic agent uses AI rules and filtering mechanism to find which of reference material was really used by the developer and adds it to repository as shown in Fig. 6.

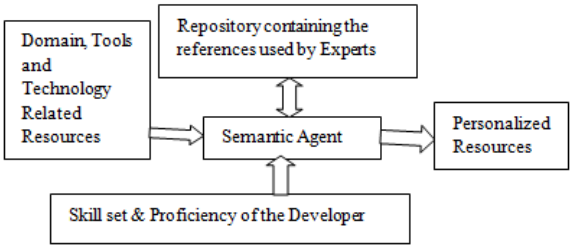


Fig. 6. Extraction of Personalized references from repository.

The resources can be organized into a hierarchy according to the nested structure (called the aggregation hierarchy). Part of this hierarchy structure can be common for a class of developers who have the same skill set and proficiency level. This can be

taken as an index to retrieve files faster from resource repository [14].

- A new index structure i.e. signature file hierarchies and signature graph is proposed for storing information of people and the references they have used.
- A unique signature will be created for each class of resource.
- Using signature graph as the index the resource will be retrieved for each person and stored for reference.

C. Selection of SMEs as e-Mentors [10]

The mentor having the highest sum of the three factors: Rank, Criticality of the project and Period of Association would be selected.

Rank: The rank of the mentor would be determined using two factors Success rate and Help rate; a ratio of 7:3 can be given to Success rate and Help rate

Success rate = NoOfSuccess / NoOfTimesReplied
Help rate = NoOfTimesReplied / NoOfTimesApproached

Criticality of the project in which the mentor is currently working: If the project on which the mentor is currently working has enough of time to complete it, then it is assumed that the mentor as time to spare hence assigned a high rank.

Period of Association: This is given by Module Query Time / Project Time, where Module Query Time is the time that a module that raised the query has for completion and Project Time is the time required by mentor to finish his current work in his project.

V. Conclusions

Agile methodology adoption need not be restricted to a few projects because of lack of skilled employees. It can be extended to moderately skilled people by providing adequate resources to assist and relevant material with information extracted from logs and other documents mentioned. Assist from e-Mentors (SME) can help resolve impediments. Creating deployment infrastructure (production environment) on cloud for testing on various architectures can reduce build errors.

Acknowledgment

I, Radha Shankarmani, am very grateful and indebted to my guide Dr. S. S. Mantha and co-guide Dr. Vinaya Babu for providing me guidance and invaluable suggestions. I would like to thank my students for helping me conduct the experiments on the Empirical formulae that I had used in this paper. Lastly I would like to convey my regards to the contributors of agile articles and their various blogs which gave me an exposure of industries practicing agile methodology.

References

[1] http://blogs.forrester.com/tom_grant/12-05-15-agile_adoption_is_a_trip_down_the_cone_of_uncertainty Posted by Tom Grant on May 15, 2012
[2] www.sei.cmu.edu/reports/08tn003.pdf
[3] http://www.systemdynamics.org/conferences/2007/proceed/papers/CHICH380.pdf
[4] http://it.toolbox.com/blogs/alm-blog/agile-rework-and-retrofit-10467
[5] http://dev2ops.org/blog/2010/2/18/deployment-management-design-patterns-for-devops.html
[6] http://agile.dzone.com/news/hypothesis-based-development
[7] http://www.ambysoft.com/essays/agileProjectPlanning.html
[8] http://www.slideshare.net/ThoughtWorksStudios/agile-metrics-that-matter#text-version
[9] http://journals.tdl.org/jodi/article/view/5874/5880
[10] "Selection of e-mentors and Smartphones application to resolve issues in Agile Projects" R. Shankarmani, IC4e, Jan. 2011.
[11] 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, pp. 1437-1454 (2003)
[12] Bizer, R. Heese, M. Mocho, R. Oldakowski, R. Tolksdorf, R. Eckstein, "The Impact of Semantic Web Technologies on Job Recruitment Processes" *In International Conference workshop on computer science (WI'05),* 2005.
[13] http://www.gate.ac.uk/
[14] "Signature file hierarchies and signature graphs: A new Index method for object oriented databases", by Yangjun Chen and Yibin Chen. SAC'04, ACM 2004.
[15] http://www.trainingindustry.com/webinars/when-an-employee-leaves-how-much-knowledge-just-walked-out-the-door.aspx
[16] http://www.eici.ucm.cl/Academicos/ygomez/descargas/ing_sw1/Using%20Risk%20to%20Balance%20Agile%20and%20Plan-Driv.pdf
[17] http://www.open.collab.net/media/pdfs/CollabNet%20Whitepaper_Reinforcing%20Agile%20Dev%20in%20the%20Cloud.pdf?_=d
[18] www.trainingindustry.com/media/13704834/8_30_12_webinar.material.pdf
[19] http://alaverdyan.com/readme/2011/01/the-definition-of-done-dod/
[20] "A Decision Support System Utilizing a semantic Agent" by R. Shankarmani etal, ICSESS 2010, Beijing, China

Authors

First Author – Radha S. Mani is currently working as an Associate Professor at the Department of Information Technology, Sardar Patel Institute of Technology, Mumbai, India. She is also presently pursuing her Doctorate from JNTU, Hyderabad. Prof Shankarmani holds a Masters degree in Computer Science and Engineering from NIT, Trichy and Bachelors degree from PSG College of Technology in Electronics and Communication Engineering. Her areas of interest include Software Engineering, Software Testing, Software Architecture, Databases, Data Warehousing and Mining, Computer Simulation and Modeling, Management Information System and SOA.

Having more than 14 years of teaching experience and 4 years of industry experience, she has held designations such as Programmer, Software Engineer and Manager. She did her sabbatical for two months in Infosys, Pune in 2005 and has co-authored two books "Middleware and Enterprise Integration Technologies" and "Game Architecture and Programming" for Wiley India. She has published papers in National, International conferences and journals.
Email: radha_shankarmani@spit.ac.in

Second Author – Dr. Sachin S. Mantha is currently working as Chairman, AICTE. He has more than 190 publications in National and International Journals and Conferences to his credit as also 12 Ph D students who have completed their Ph D under him. He has two books published titled "Objet Oriented Programming in C++ Published by Wiley India Aerodynamics of Cars, An Experimental Investigation - A Synergy of Wind Tunnel & CFD, LAP LAMBERT Academic Publishing.

He is also the recipient of INFOCOM CMAI National Telecom award for 2010 and the HR Nexus – the HR & Network, Leadership award in the field of Education, 2010. The .e – governance project implemented in AICTE has also won the Silver Edge Award instituted by United Business Media and Information Week and the Good Governance National Technology Award, 2010 instituted by the Good Governance Magazine.
Email: ssmantha@vjti.org.in

Third Author – Dr. Vinaya A. Babu is currently working as Professor in CSE and Principal of JNTUH College of Engineering, JNTUniversity Hyderabad. He guided 9 PhDs and many more PhD Scholars are working under his Guidance. He had held number of administrative positions in the University including that of Director Admissions, SCDE, and HOD of CSE etc. He has published more than 82 Research publications to his credit so far in his specialization such as algorithms, data mining, parallel algorithms, networks and related areas.

He secured "Best State Teacher Award", State Govt. of Andhra Pradesh in 2011,"Best Computer Science Teacher Award" by ISTE, AP Chapter in Nov 2009, "Eminent Educationist", awarded by National & Inter compendium, New Delhi, India April 2009,"Eminent Citizen of India", awarded by National & International Compendium" New Delhi, India March 2008, "Distinguished Academician" awarded by "Pentagram Research centre Private Limited" Hyderabad, India on Jan 2008 and "Jewel of India 08" awarded by Indian solidarity council, New Delhi, India
Email: dravinayababu@jntuh.ac.in