Chapter 8
Conclusion and Recommendations
8.1 Introduction

A debate is shaping about the compatibility of two different software development approaches: agile software development and distributed software development. The IT industry, in general, is embracing distributed software development, as major firms scramble to build and recruit highly skilled, low-cost talent in the technology centers across nations. Executing a software project with a distributed team is not a simple proposition. There are many challenges injected into an already complex process, many of which are impossible to quantify and difficult to control. As a result, many companies are looking at using agile methodologies in distributed environments. Many agile practitioners claim that agile methodologies are not compatible with such a distributed development, since they are based on practices like pair programming, face-to-face communication, customer on place and so on. Another debate is grooming around the development of documentations for knowledge sharing and maintenance of software. Solely relying on tacit knowledge (as recommended by agile manifesto) is not going to serve the purpose, especially in distributed software development. The goal of this research was to understand how the agile development practices manage the KM practices, especially knowledge sharing in distributed environment and what parameters are applicable for use of agile in distributed environment.

The aim of this chapter is to discuss the findings in terms of each of the four research questions outlined in Chapter 1. As well as discussing the outcomes of the hypothesis tests, this chapter provides context and meaning to the study by qualitative analysis of the survey responses.

The scheme of the research work carried out during the course of this thesis can be summed up in the following:

- Investigation of the KM practices in software engineering organisations working in agile environment through literature and developing theoretical and conceptual foundations.
- Empirical analysis of KM and KS processes in software engineering organisations in India working in agile methodologies through exploratory methods.
- Identification of parameters of applicability of agile software development in
Firstly, the discussion focuses on the findings related to the hypotheses associated with the first research objective: Identification and assessment of KM practices in software engineering organisations working in agile methodologies. In perusal of this objective, the research also explores the extent of KM adoption related to particular organisational characteristics.

Diving deeper, we discuss knowledge sharing strategies in distributed agile development and come up with some guidelines for effective knowledge sharing when agile software development is applied in distributed environment. Then in next section, we will discuss the parameters of applicability of agile in distributed environment.

Finally, the last and important objective of the study is to design an information radiator for distributed software development teams.

**8.2 Summary of Finding of KM Practice Adoption Survey**

As discussed in chapter 1 and 2, it was felt that it is crucial to examine knowledge factor in agile methodologies because many researchers believe that many KM practices are already embedded in agile methodologies. For better understanding, we summarise and explain the findings from survey data analysis. The first research question explored the extent of adoption of KM practice, and the relationship between the extent of KM practice adoption and particular organisational characteristics.

**8.2.1 Adoption of Individual KM Practices**

We find adoption level of 29 KM practices individually. Mean of 29 KM practice is found to be 75.09% with standard deviation of 12.1. Almost all KM practices have adoption level of more than 50 percent. Which shows that more than half of the respondents are using listed KM practices. We summarise five least and maximum used KM practice in Table 8.1.
Table 8.1: Comparison of adoption of least and most used KM practices

<table>
<thead>
<tr>
<th>Top 5 KM Practices</th>
<th>%age</th>
<th>Least 5 KM practices</th>
<th>%age</th>
</tr>
</thead>
<tbody>
<tr>
<td>There is a program of active participation in discussion forums to share and learn ideas and experiences</td>
<td>90.51</td>
<td>Employees at all levels are encouraged to participate in formulating business policy in this organization</td>
<td>63.7</td>
</tr>
<tr>
<td>The organisation fosters development of “human-centered” information technology</td>
<td>89.99</td>
<td>A formal process of transferring best practices, including documentation and lessons learned exist in the organisation.</td>
<td>60.1</td>
</tr>
<tr>
<td>Knowledge acquisition and dissemination is not restricted to hierarchies</td>
<td>88.88</td>
<td>Groups and individuals and routinely document and share information about their expertise</td>
<td>51.14</td>
</tr>
<tr>
<td>Your organisation encourages experienced employees to transfer their knowledge to new or less experienced employees.</td>
<td>88.51</td>
<td>Your organisation has a written knowledge management policy or strategy.</td>
<td>50.89</td>
</tr>
<tr>
<td>There are lesson learned and best practices repositories within my organization</td>
<td>88.16</td>
<td>A high proportion of our internal knowledge sharing is achieved through documents and database</td>
<td>50.64</td>
</tr>
</tbody>
</table>

We have divided KM practices into seven groups. These all groups comprise the overall knowledge management practice. We asked different questions to respondents to check how that particular practice has been implemented in their respective organisations (cf. Table 8.2).

Table 8.2: Summary of KM Practice adoption levels by KM Practice groups

<table>
<thead>
<tr>
<th>KM practice Group</th>
<th>Number of practices in the Group</th>
<th>Mean Adoption</th>
<th>No. of Practices adopted in Range</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>75.01% -100%</td>
</tr>
<tr>
<td>Knowledge Management Environment</td>
<td>5</td>
<td>81.45</td>
<td>4</td>
</tr>
<tr>
<td>KM Culture, Policies and Strategies</td>
<td>6</td>
<td>69.43</td>
<td>2</td>
</tr>
<tr>
<td>Knowledge Capture and acquisition</td>
<td>4</td>
<td>70.11</td>
<td>2</td>
</tr>
<tr>
<td>Knowledge Sharing</td>
<td>5</td>
<td>76.64</td>
<td>4</td>
</tr>
<tr>
<td>Training and Mentoring</td>
<td>5</td>
<td>72.99</td>
<td>2</td>
</tr>
<tr>
<td>KM Technologies</td>
<td>4</td>
<td>81.33</td>
<td>3</td>
</tr>
<tr>
<td>Total</td>
<td>29</td>
<td>75.325</td>
<td>17</td>
</tr>
</tbody>
</table>
8.2.1.1 Knowledge Management Environment

We present a new dimension to KM practices by including KM environment. It is important to know before implementing any knowledge management process, whether organisations provide a healthy KM environment so that it can help other KM processes to grow. The study asks five different questions from the organisations to provide input regarding the KM environment. It was found that agile organisations, by default, provide a healthy environment where KM can grow. Mean adoption level of this practice was found to be 81.45%. It was found that organisations maintain lesson learned and best practices repositories in more than 88% of the firms. An important aspect was found that 85% respondents agreed that organizations have adopted the culture of knowledge sharing instead of hoarding of knowledge. Organisations provide support for communication (82.2%) and networks among employees (79%) so that maximum knowledge transfer can take place.

8.2.1.2 KM Culture, Policies and Strategies

Group mean of adoption of KM practices is 69.4%. Climate of openness, trust and new learning was found dominant in the organisations which are accepted by more than 85% of organisations. It was found that principles of agile manifesto of openness, trust and learning are actually followed by organisations working in India. The minimum accepted practice in this group is that organisations do not favour written knowledge management policy, only 50% of the respondents agreed to this which is minimum adopted practice in all. 66.5% of organisations have clearly mentioned the processes and rules through which knowledge should be managed. Strengthening the concept of self-organising and cross-functional teams organisations encourage their employees to participate in formulation of business policies also.

8.2.1.3 Knowledge Capture and Acquisition

Mean adoption average of this group is 70.11%. Main outcome which comes from this group is that knowledge acquisition and dissemination is not restricted to hierarchies which are agreed to by 88% respondent organisations. Majority of the respondent

---

7 see principle 6 and 7 of agile manifesto (www.agilemanifesto.com)
organisations (80%) agreed that knowledge gaps are systematically identified and well-defined processes are used to close them. A little more than half of the organisations routinely document and share information about their expertise among groups and individuals.

8.2.1.4 Knowledge Sharing

Over 90% of the respondent organisations have programs for active participation in discussion forums where people can share their ideas and learn new ones. Over-all adoption mean of this group of practices is 76.6%. To increase networks among employees, more than 83% of the respondent organisations use rotation of people among projects. Approximately 80% of the organisations are using self-organising and cross functional teams and direct people-to-people contact in organisations to help in sharing of internal knowledge among employees. More than half of the respondent organisations use documents for internal knowledge sharing, which is comparatively very less than knowledge sharing via people-to-people contact. Results also acknowledge people-to-people contact mentioned in agile manifesto.

8.2.1.5 Training and Mentoring

Overall adoption mean of this group was recorded as 72.99%. In most of the organisations (88%) experienced employees transfer their knowledge to new employees which can prevent ‘experience walk away home’ syndrome. 78% of the organisations offer offsite training to employees in order to keep their skills up-to-date. When asked whether organisation provides formal or informal training to their employees, we got a mixed response with 63.8% and 68.6% responses respectively.

8.2.1.6 KM Technologies

Mean adoption level of this group was 81.33%. In this group, the study asked for practices pertaining to the use of technology for knowledge management activities. 83% of the organisations are using e-mail, bulletin boards, on-line forums and databases for transfer of knowledge. 67% of the respondents agreed that technology brings the organisation closer to its customers.
8.2.2 Adoption of KM Practice across Agile Software Development Industry

The present study did not find any variation in adoption of individual KM practice but we found a wide variation in overall adoption of KM practice by organisations. Responding organisations have adopted, on an average, 73% of the KM practices put forth in the survey. The standard deviation of 16% and 21.83% of coefficient of variance indicates variation across organisations in the level of adoption of the 29 KM practices. More than 11% of organisations are using less than half of the practices, 35% of organisations are using KM practices ranging from 50 to 75%. The organisation with minimum adoption has implemented only 30% of practices and organisation with highest adoption rate has adopted more than 97% of the KM practices. The large deviation provides further support for the hypothesis that there is wide variation in the level of organisational KM practice. Having established that wide variation in overall adoption of KM practices exists across the software engineering organisations, the next four hypotheses considered organisational characteristics thought to be associated with KM adoption: organisation size in terms of employee strength, domain of the software engineering, team distribution and type of organisation.

8.2.2.1 Organisation Size

As discussed, the present study has divided software engineering organisations working with agile practices into three groups on the basis of number of employees working in that firm in their Indian offices. These groups were identified as size I, II and III organisations. The lowest rate of adoption of KM practice was found for Size II (medium-sized) organisations (69.0%) with the highest variance (19%) in the adoption of KM practices. The highest rate of adoption of KM practices (77.13%) was found for Size I (large-sized) organisations. The variance in the adoption of KM practices in Size III (small-sized) organisations (7.5%) is the lowest. The result of the analysis of variance calculation indicated that the KM Practice levels do not vary by size e.g. ANOVA and Robust Tests of Equality of Means are well above the significant level (p>0.05), which means organisations that are using agile methodology do not vary in adopting KM practices.
8.2.2.2 Core Area

The present study has divided software engineering organisations working with agile methodologies on the basis of core area of their work. Three core groups are identified: organisations working in product development (PD) alone; organisations providing consultancy alone; organisations developing software as well as providing consultation both. Description analysis shows where PD shares the highest adoption mean of 81.2% whereas consultancy and both type of organisations have almost same level of mean adoption of 68%. Also, the variation in the adoption of KM Practices (minimum 30.58%, maximum 95%) is more in the case of companies engaged in both the areas of software engineering. Analysis of ANOVA (p = 0.083) found that there is no variation in adoption of KM practice regardless of the core area of the company. Further analysing the results, Tukey HSD test also confirms that there is no significant association among different dimensions of organisations.

8.2.2.3 Team Distribution

Today almost every company is working in distributed environment. So this study tries to find out KM practice level of teams rather than organisations because every company has many teams working in distributed setting as well as co-location. Therefore team is the best available option to know KM adoption level team wise co-located and distributed. An independent samples t-test (p=0.156) confirmed that there is statistically no significant difference between the means of the KM level of the co-located and distributed teams working in the area of agile software development. Therefore, it can be concluded that KM practices do not vary significantly for teams working in co-location or distribution across the globe.

8.2.2.4 Type of Organisation

Software engineering firms have been distinguished on the basis of agile technology: Fully agile organisations (organisations that are using agile practices in all of their software development projects) and partially agile organisations which are using agile practices to only some of their projects. Fully agile companies show greater adoption of KM practices (80.95%) compared to partially agile organisations (69.05%). Partially agile firms show greater variation of 17.81% as compared to 11.3% of fully agile
organisations. The result of the analysis of variance indicated that the KM Practice level vary by companies adopted agile practices for all of their product development and organisations which use agile practices for some of its projects. An independent samples t-test confirmed (p=0.031) that there is statistically significant difference between the means of the KM level of the fully and partially agile organisations.

8.2.2.5 KM Adoption and Number of Employees

Association between organisational KM level and the number of employees is investigated from different demographic dimensions: organisation size, core area and type. Pearson’s product moment correlation is used to test the association. After analysing this study did not find any association between all sizes of software engineering organisations that are working in agile methodologies. Analysis of association between KM Practice levels and core area of the software engineering domain reveals that there is a significant positive relationship between the organisation KM level and the number of software developers in each organisation for only product development companies (p = 0.039). However the association between KM Level and the number of software developers was not found to be significant for companies dealing in consultancy services alone and organisations working in product development and in consultancy both (p = 0.596 and 0.187 respectively). Exploring the association between KM Practice levels and type of organisations it was found that there is no significant positive relationship between fully and partially agile organisations (p = 0.137 and 0.931 respectively).

8.3 Agile Practice Used for Software Development

For the design of Agile Information Radiator (AIR), it was important to find out the methodology used by Indian software industry. Eventually, it was extracted from data that Scrum is the most common agile practice used by Indian industry followed by XP-Scrum hybrid and finally XP alone (cf. Fig. 8.1). Still, there are 12.6% of the respondents who have marked that they are using other agile practices which are not mentioned in the list. This describes the large number of tailor made agile practices which organisations have developed for their own working.
8.4 Agile Practices used for Knowledge Capture and Acquisition

Respondents were asked for agile practices which they feel are used for Knowledge capture and acquisition. We summarize five top and the least used agile practices. It was clear from the results that the mostly used practices are part of Scrum methodology which is dominant among all the agile methodologies in Indian subcontinent.

Table 8.3: Summary of Agile Practice used for Knowledge Capture and Acquisition

<table>
<thead>
<tr>
<th>Top 5 KM Practices</th>
<th>%age</th>
<th>Least 5 KM practices</th>
<th>%age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standup meetings</td>
<td>73.8</td>
<td>Close collaboration</td>
<td>12.6</td>
</tr>
<tr>
<td>Sprint review</td>
<td>57.1</td>
<td>Virtual scrum wall</td>
<td>9.7</td>
</tr>
<tr>
<td>Burn down charts</td>
<td>54.1</td>
<td>Collected code ownership</td>
<td>8.2</td>
</tr>
<tr>
<td>User stories</td>
<td>53.2</td>
<td>System metaphor</td>
<td>5.9</td>
</tr>
<tr>
<td>Release planning</td>
<td>51.8</td>
<td>Proxy customer</td>
<td>1.8</td>
</tr>
</tbody>
</table>
8.5 Outcomes and Recommendations of the Agile KM Survey

- Indian software industry working with agile practices lacks in providing any formal head for KM positioning (e.g. Chief Learning Officer, Chief Knowledge Officer). Only 3.2% of organisations have chief Knowledge Officer. Half of the respondents responded either by saying it is everyone’s job or no formal role exists in their organisations.

- It is seen that software organisations working with agile methodologies have particularly focused on two aspects of KM, first, providing healthy knowledge management environment and second, providing right set of technology which can enhance the overall knowledge management adoption. Both of the aspects have shown more than 81% of adoption level.

- Learning and sharing through discussion forums is the highest used practice among all respondents of organisations with adoption rate more than 90% whereas the least accepted practice is dependence upon documents for transfer of knowledge which is around 50%. This has strengthened many practitioners’ belief that most of the knowledge in agile software development is tacit in nature and agile approach heavily relies on this tacit knowledge sharing.

- The lack of a clearly defined authority and an independent department or functional unit to manage the systems and processes related to KM function in software organisations is an indication of the neglect and poor long-range planning of the knowledge management activity.

- It was found in survey that dominant agile practices, that are used by agile organisations is scrum, followed by extreme programming. So scrum master or agile coach should be appointed as formal head for agile practices whereas in practice, authorities in the organisations are providing guidelines which are not certified and cannot implement agile in its spirit.

- This study found a lack of commitment in software engineering firms to give away financial reward for knowledge sharing-only 14% of the firms acknowledge this
practice. Majority of the firms (69%) only provide social recognition to the employees who promote knowledge sharing.

- Discussion forums increase the networks among employees which has emerged an important factor for knowledge sharing with WAS score of 0.88. Rotation of employees among projects and direct people-to-people contact has been used more by agile organisations (WAS>1.0) as compared to partially agile organisations (WAS<0.3) for building networks among employees.

8.6 Knowledge Sharing Strategies in Distributed Environment

In their original form, almost all the agile practices rely on face-to-face communication which restricts the use of these practices to co-located and small teams. Documents are created only to facilitate communication and are updated only when necessary. But distributed development can not solely relay on tacit knowledge. This raises skepticism about the ability of agile processes to maintain and distribute knowledge among team members in a distributive environment. Therefore, it was important to judge how agile practices cope up with challenges raised by distribution of software development. To meet second objective of the research, this study explored the knowledge sharing strategies from three dimensions. Respondent organisations were asked a few questions on 5-point Likert scale to judge knowledge sharing strategies they follow in distributed environment:

a) Kind of knowledge shared
b) Tool support for knowledge sharing
c) Documentation support for knowledge sharing

8.6.1 Outcomes and Recommendations

- Survey results revealed that cross functional and self organizing teams build trust and knowledge sharing environment. Ambassadors to different sites and short feedback cycles are used to encourage knowledge sharing between sites and members.
• All the organisations are using tools for project management across different dimensions of the agile organisations. Only difference is found in case of fully agile and partially agile companies. 71.3% of fully agile organisations use project management tool as compared to 54.9% of partially agile companies.

• Team members in different time zones should adjust their working hours to reach maximum overlap for synchronous communication between onshore and offshore teams (WAS>0.84). Interviews with different professionals disclose that kickoff-meeting with distributed teams helps in building trust.

• For asynchronous communication, use of document management systems, e.g. Wikis, can be used as central team repository. Documents can include use cases, story cards, tasks, questions and answers, tracking data, calendar of events and team and personal blogs etc.

• Use of the source code as main communication medium between sites creates an environment for effectively sharing documentation and design artifacts. Documents should be managed and updated as an integral part of the software maintenance process. Lightweight documentation should be encouraged, e.g. burn down charts, Code Review docs, Test Cases, Defect Report etc. Asynchronous communication tools are also very helpful in maintaining lightweight documentation and sharing of knowledge.

• Tools for synchronous communication are used to connect the teams. Majority of fully agile organisations (WAS score=1.03) are using collaborative tools to mimic face-to-face communication. Different tools like Integration tool, Bug tracking tool, Source control tool are used to support multi-site synchronization and communication.

• It was found that fully agile organisations (45%) are more regularly updating their documents as compared to partially (24%) agile organisations. Similarly, this practice is not found to be popular in small size organisations where only 23% of the respondent organisations agreed with the statement.
• It was surprising to see that agile organisations are preparing and maintaining more documents than partially agile organisations. More than 60% of fully agile organisations are either writing project reviews or preparing written documents for different development purposes as compared to partially agile organisations.

• Organisations are now moving towards cloud for storage of their code repositories, rather than maintaining them at each location. Placing code repository in a cloud is to make it available for all distributed agile teams so that they can implement collective code repositories. Majority of the respondent organisations (more than 65%) are using cloud for maintaining their source code, so that synchronization can be made among distributed teams, whereas this practice is less popular in small size (WAS =0.53) and partially agile (WAS=0.51) organisations.

• Fully agile organisations facilitate geographically separated teams for performing collaborative work more than the partially agile teams. This means fully agile teams try to maintain their spirit of agile for maximum collaboration of work despite distribution.

• Although agile manifesto disapproves the use of too much documentation which is clearly stated in agile manifesto ‘Working software over comprehensive documentation’, however, organisations working in India in agile practices are maintaining documents for knowledge sharing as well as for future maintenance purposes. This violates the core value of agile manifesto.

• Agile practices heavily rely on informal communication within the employees for knowledge transfer. Software companies should strengthen and promote informal channels of communication so that knowledge sharing and transfer becomes better in organisations.

8.7 Findings and Recommendations of Parameters of Applicability of Agile

As the objective of the research was to identify the parameters of applicability of agile in distributed environment, important factors are extracted using Principal Components Analysis and Varimax Rotation. Factor analysis is a multivariate statistical technique used by researchers to reduce the number of variables or parameters to a manageable
number. Hence, Factor analysis technique was applied to reduce the list of 43 parameters to a smaller set of underlying factors that are significant in understanding the applicability of agile in distributed environment. Eight factors are found after using factor analysis. These factors are given in Table 8.4.

Table 8.4: Factors of parameters of applicability of agile

<table>
<thead>
<tr>
<th>Group No</th>
<th>Component</th>
<th>Factors in Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Schedule and Processes Management</td>
<td>Small releases&lt;br&gt;Top management support&lt;br&gt;Medium duration project&lt;br&gt;Domain expert availability&lt;br&gt;Short feedback cycle&lt;br&gt;Upfront modeling&lt;br&gt;Small project size&lt;br&gt;Customer integration&lt;br&gt;Short iteration cycles</td>
</tr>
<tr>
<td>2</td>
<td>Techniques</td>
<td>Source control systems to support multisite synchronization&lt;br&gt;Doing upfront designing&lt;br&gt;Support for status tracking &amp; reporting&lt;br&gt;Overlap working hours in different time zone&lt;br&gt;Collocating onsite and offshore regularly&lt;br&gt;Higher documentation (such as lessons learned, training manuals, good work practices, articles for publication etc.)</td>
</tr>
<tr>
<td>3</td>
<td>Communication and Collaboration</td>
<td>Tools for synchronous and asynchronous communication&lt;br&gt;Increasing communication bandwidth&lt;br&gt;Building Trust between offshore and onshore teams&lt;br&gt;Building central specifications repository&lt;br&gt;Project and process management tools&lt;br&gt;Coordination and collaboration between multiple distributed teams</td>
</tr>
<tr>
<td>4</td>
<td>Resource and Risk Management</td>
<td>Up front planning&lt;br&gt;Building personal relationship networks&lt;br&gt;Employee empowerment&lt;br&gt;Early risk mitigation</td>
</tr>
</tbody>
</table>
Training to employee
Progress monitoring System

5 Technology and Tools
Integrated development tools
Continues integration tool
Maintaining one code repository
Effectively sharing documentation and design artifacts
Educate and awareness of team members on frameworks, tools and strategies

6 User's Adaptability
Awareness of regional languages
Cultural awareness
Experienced staff

7 Organisational and team culture
Offshore team familiar with agile practices and principles
Team commitment
Facilitation of knowledge sharing
Company culture
Self-organization and self-directed teams

8 Financial and Temporal aspects
Increase of productivity
Acceleration of time to market
Reduction of cost

8.8 Findings and Recommendation of Agile Information Radiator (AIR)
Alistair Cockburn coined the term “information radiator” in 2000. An information radiator is something that interested parties can use to find out the state of the project without needing to interrupt or bother the team members. For such tools to be useful for agile teams, they should also provide support for specific agile practices and help overcome some potential drawbacks in agile methods. Therefore, the role of electronic information radiator is more than the one played by it in a co-location. Agile Information Radiator (AIR) has to manage three C’s which come with distribution, i.e. coordination, communication, and collaboration because without these three C’s, distributed development with agile practices is not possible.

On the basis of survey results and interviews with professionals, AIR is logically divided into two parts: Knowledge Management and Project Management. In KM different documents produced during the course of action e.g. requirement documents,
chat logs etc are stored in AIR knowledge base. In project management, this is shown by two different class diagrams, separate for each process.

Although agile recommends minimum production of documentation but Indian agile industry produce and maintain documentation for maintenance phase also. For the need of Indian software industry working in agile, a new KM perspective is added into the tool.

It is found that organisations in India are mostly using scrum practices for software development. Therefore, in the tool most of the scrum practices are implemented in the design of the tool. It is important to note that scrum is a project management methodology. Therefore, researcher includes the majority of practices of scrum methodology.

Apart from modelling approaches, accessibility of relevant knowledge to all the stakeholders concerned is crucial for the exchange of software engineering knowledge among its community. Repository systems serve this purpose by facilitating sharing of the software engineering knowledge and experience across software organisations as well as research community in this field.

AIR helps organisations in systematically capturing, storing and disseminating knowledge among the distributed agile teams, by creating a knowledge-sharing culture. Tool provides online discussion forums so that distributed agile teams can communicate among themselves.

AIR provides for both formal as well as informal ways of sharing of knowledge and effective means of communication, accessible to every member of the organisation. Lightweight knowledge management approaches attempt to capture the informal knowledge that is shared on a daily basis so that it can be disseminated on a larger scale.
8.9 Major Contributions of the Study

The major contributions of this research can be summarised as following:

1. The findings of the KM Assessment survey can be used by the software engineering organisations in getting insights into the current state of knowledge management practices in software engineering organisations working with agile methodologies. In the light of the findings and recommendations of the survey, the software companies interested in the adoption of the KM practices can plan the implementation of their KM strategy in a more systematic manner.

2. The survey of the knowledge sharing in distributed environment can act as milestones for planning and implementation of the agile in distributed setting. The aspirant organisation can take the advantage from the guidelines laid by the researcher in order to enhance the adaptability of agile in distributed environment by focusing on good knowledge sharing strategies.

3. This study can also assist the organisations in implementing agile in distributed environment. It provides a list of eight factors through an empirical investigation for implementing agile in distributed environment. These factors will assist software organisations in removing the bottlenecks and hindrances which are faced by the organisations while implementing agile in distributed environment.

4. An extended conceptual model of distributed agile development has been explained in this study. This model explains the management of the project when agile practices are applied in distributed settings. Also the management of documentation produced in the project has been explained in the model.

5. A new schema of knowledge management has been suggested that enables the storage and retrieval of knowledge produced in distributed project. This part of the model has been explained with the help of a separate class diagram.

6. A comprehensive and detailed model has been designed with the help of UML that supports distributed agile development and management of knowledge and documentation produced while developing a project.
7. The proposed model extends and refines the distributed agile development concept by adding needs of Indian software organisations, especially the use of documentation and knowledge management, and thus makes it practical to implement AIR in Indian software engineering environments to reap the benefits proposed by the model.

8. The AIR is not bound to any special infrastructure, making it safe against failures on technical grounds. In fact, the model can be easily integrated into the existing software engineering environment without any major technological revisions or upgradations.

9. The AIR model can be tailored very easily by the organisations for their specific needs. For instance, it can support software project management during the planning phase or it can be implemented in an organisation for supporting and enhancing the specific agile practices, e.g. pair programming etc.

10. Moreover, systematic and well-organised information about the status of the project helps in speeding up the software development with distributed agile teams because instant and accurate status can be displayed with the help of the tool. This is also an important contribution to the existing body of software engineering practice.

11. The AIR can be introduced and extended gradually in an organisation, providing a mechanism and cushion to handle ill-effects caused by distribution in a more stable, consistent manner.

Overall, this research work adds significant contribution to the field of empirical software engineering. This study presents a generalized model for implementing agile in distributed environment and tries to mitigate the ill-effects caused by distribution while applying agile practices. Knowledge management support provided by the tool helps in the maintenance of documentation and knowledge in order to survive in volatile and knowledge based economy. As modelling framework in software engineering enables better communication and comparative analysis of empirical research findings, the present work contributes in the field of modelling in software engineering based on empirical investigations in the industry.
8.10 Limitations of AIR

Though AIR is an important contribution in structuring and managing distributed agile teams, it also helps in managing knowledge generated by different phases of software development, it is not without limitations.

Limiting factor for implementing AIR is the requisite infrastructure for virtual collaboration of distributed agile teams. The AIR requires a corporate intranet as the front-end along with a database in the backend. Success of AIR depends on how organisations are maintaining their code repositories and knowledgebase.

Another important factor for ensuring the success of AIR depends upon the culture of the organisation. Corporate culture supporting knowledge management, experience management and freedom of teams are prerequisite for implementation of AIR.

8.11 Scope of Future Research

There is always scope for improvement, whenever a step has been taken for the refinement of a process.

Important research endeavour in this area could be the focus on some specific agile practices that can be incorporated into agile processes, especially the practices related to extreme programming because a trend has been seen that Indian software industry is now showing interest in hybrid approach of integrating scrum-XP for software development.

Some experience based model can be integrated to AIR when applied in distributed environment so that experienced knowledge of a person is captured and stored for future references. These models can help in management of experience of the person which can reduce the impact of ‘experience walking home syndrome’.

Some of the web 2.0 practices (like wiki pages etc.) can be incorporated with the model for online handling of the documentation. They support collaboration across time and space.

The AIR proposed in this work integrates different kinds of software engineering practices specific to scrum methodologies. The proposed schema can also be effectively
used for designing tools for other agile practices also. This will give scope for future research in other agile software development methodologies.

As companies are moving towards cloud storage of repositories and tools, cloud based architecture can be used for storage of knowledge produced from the tool. Cloud helps in 24x7 availability of data and code, which can help in implementation of specific agile practices like collective code ownership.

---

*Term introduced in the late 1960s to explain the impact of quick increases in computer power and the complexity of the problems. As the programs became more complex more programmers were used to manage the complexity, but it did not work, still results were: late delivery of software, software not meeting the intended needs, and many errors were detected after product delivery, which led to complexity in maintaining software.*