A Framework of Challenges and Key Factors for Applying Agile Methods for the Development and Customisation of Software Products in Distributed Projects

Abdulrahman M. Qahtani, Gary B. Wills, Andrew M. Gravell
School of Electronics and Computer Science, University of Southampton, UK

Abstract

Organisational and cultural boundaries are two of the main aspects of the software development process for distributed customers. Many organisations use traditional approaches such as the waterfall model to develop and customise software products for different customers. Agile software development, however, has departed from traditional approaches, and a great deal of research has been conducted on adopting agile software development for either distributed or collocated projects. However, little research has discussed the agile development approach across organisational boundaries between developers who use an agile software development approach and their customers. The purpose of this paper was to introduce agile software development to those organisations which create and customise software products for different distributed customers and to study the potential impact of introducing agile development across those organisational boundaries. We proposed a framework for developing and customising software across organisational boundaries. This would ultimately address and accommodate key factors in software development and customisation across organisational boundaries using an agile development approach. This research will help the development society and serve as a guide for project managers, developers, stakeholders, and users to understand agile software development and the impact of adopting an agile approach across organisational and cultural boundaries in their projects.

1. Introduction

Software development attempt to be faster and lighter development process for business community [1]. To enable this effort, organisations need to change the method of software development from a traditional to an agile approach. Agile software development is a reaction against traditional models such as the waterfall model [2], and aims to reduce development time and costs, as well as to accommodate changes in requirements at any time without a significant effect on the duration of the development period. The agile method represents a paradigm shift in the software development process and community [2]. It is not merely a development approach or methodology; rather, it is a philosophy of software development and a new way of thinking in the development process and project management [3].

A great deal of research has been conducted on the adoption of agile software development for both distributed and collocated projects [4]. In last two decades, agile development has become a faster way to develop, produce and deliver software products [1]. The main reason for this significant change is that agile methods are a collection of software practices which promote the iteration and flexibility of software development. The successful adoption of agile methods for one team has led many researchers to investigate the use of agility principles in distributed software development (DSD) projects in order to combine the advantages of agile development and DSD. A number of research studies have been conducted on various aspects of distributed development and organisational boundaries [6,7,8]. Although there are many positive features involved in applying agile methods to DSD, there are also some challenges to face. For instance, with communication, an agile manifesto emphasises face-to-face communication [8], and this is not something that is certain in DSD. In addition, there are many other factors that affect any software development process in general and the geographical distribution of projects in particular, such as project management, integration management and communication and configuration.

In this study, we have focused on these factors in terms of crossing organisational boundaries, especially for software vendors, in order to investigate and analyse the main components and propose a framework to accommodate the key components in the context of applying agile methods for the development and customisation of software products in distributed projects (see Figure 1).

This rest of this paper is structured as follows: In Section 2 and Section 3, we briefly give the background of distributed agile development (DAD) and the type of software customisation used in this research, as well as the scope of this paper. In Section 4, we review the proposed models and framework in terms of DSD and DAD, as well as distributed software development challenges and issues. In Section 5, we state the research method used to conduct this research. In
Section 6, we discuss the proposed framework and its components. In Section 7, we set out the uses of the proposed framework and its potential benefits; finally, in Section 8, we discuss future work.

![Business model between vendor and customers](image)

**Figure 1. Business model between vendor and customers**

2. **Distributed Software Development (DSD)**

In recent years, the software industry has seen a trend to relocate the development process to geographically distributed centres. This movement is driven by several factors, such as the need to optimise the avenues for creating high-quality software products by highly qualified teams [9]. Furthermore, the DSD suitable for the organisations involved have distributed geographical locations. Although DSD solves a great deal of the problems involved in software development, many challenges and issues remain in that context. For instance, communication between distributed teams and differentiation in time zones, especially for globally distributed projects, and project management across distributed locations [10].

3. **Distributed Agile Development (DAD)**

In industry, there are many collocated projects (where the whole project and team are present at one site) that have successfully adopted the agile approach through the use of agile methods. On the other hand, there have also been some research and industrial experiences related to adopting agile software development for distributed software development (DSD) projects or global development projects (GSD) [10,11]. However, using the agile approach in DSD projects involves many challenges and issues [9], such as the lack of face-to-face interactive communication between people at distributed sites. In this paper, we focus on those organisations that develop and provide software products for different distributed customers, then customise those products.

4. **Customisation software products**

The term ‘customisation software products’ refers to the customisation and specialisation process of software sold to customers and then changes in the product based on the customer’s requirements. The level of customisation varies between code and configuration customisation. The scope of this paper comprises customisation and development processes involving changes in the software code.

5. **Related Work**

5.1. **Proposed Frameworks And Models In Terms Of Dsd And Dad**

Little research has proposed frameworks and models to provide a guide for developers and managers in the agile development process for distributed projects. Šmite and Borzovs [12] conducted a study to investigate the impact of risk management on GSD, which is called global risk. In addition, they designed a framework to address the key risk management in GSD across organisational and cultural boundaries. Wahyudin et al. [13] proposed a framework for communication and information exchanges between development team members in GSD. This paper focused on the communication aspect of GSD with agile software, as well as the development process. The researchers proposed a concept to formalise the key communication between teams in agile projects to reduce the challenge and cost of communication, and to benefit from communication in a DAD system.

Akbar et al. [14] proposed a model for software companies developing web applications for distributed client locations. Their proposed model emphasises the support needed for communication between developers and offshore clients to complete their projects with minimal documentation. Hossain et al. [4] conducted a survey to investigate and identify the challenges of applying an agile method called Scrum on GSD. Furthermore, they proposed a conceptual framework that presented the key challenges of using Scrum in GSD projects. Their framework could help project managers using Scrum in GSD to consider the challenges and risks their project could face in order to reduce them. Moreover, Lee and Yong [10] conducted a study that examined the main issues of global DSD and the challenges facing distributed localisation teams of software products. They suggested a framework to map the challenges of project management in the globalisation of DAD to practices. Mudumba and Lee [15] found that there was a lack of studies conducted on the risk management of DSD. As a result of this, they proposed an agile risk management framework that supported an identification process of dynamic risk management for DSD. Interestingly, they discussed multi-organisations, multi-teams and other multiplicities in DSD. They reported that several researchers...
had recommended this type of agile method in project management to mitigate the dynamic risk of software development projects. Phalnikar et al. [11] also carried out a study to investigate the benefits of using agile methodology like Scrum in DSD projects. They presented some of the challenges of DSD, such as communication, configuration management, project estimation and cultural challenges. In addition, they demonstrated the benefits of agile distributed development. The scope of their study covered projects using a traditional development approach and the agile adoption of those projects. Furthermore, they proposed two models for a team structure in DSD.

Table 1 shows the frameworks and models that have been proposed to address some of the challenges and issues related to DSD and apply agile concepts on DSD projects. Furthermore, the table lists the main contribution and discusses aspects of each study.

5.2. Review Of Challenges And Issues Related To DSD and DAD

In this section, we review the studies that have discussed the main issues and key factors involved in DSD or using an agile approach in DSD.

Coram and Bohner [8] examined the impact of agile methods on software project management. Their study discussed the impact of applying agile methods in the project process, as well as the people involved in the project process such as developers, testers, project leaders and customers. They also discussed some management and development processes (e.g. planning and documentation). Some researchers have focused on one aspect of project management, such as risk management, and then examined the impact of distributed development or agile development on this aspect. Jiménez et al. [9] conducted a systematic review of the literature relating to the challenges and issues of DSD. In addition, their study shows the proposed solutions in meeting those challenges. They addressed the challenges of DSD projects, such as communication, group awareness (relationship between people in the project), configuration management, knowledge management, coordination, collaboration, project and process management, process support, risk management, cultural differences and quality measurement. In addition, they presented a proposed solution or way of meeting each challenge at that time.

Sengupta et al. [16] have done research initiated by study at the IBM research centre to investigate the challenges of DSD. They identified four areas in DSD, which are collaborative software tools, knowledge acquisition and management, testing in a distributed set-up and process and metrics issues. In addition, they addressed the issues and difficulties of each area as well as presenting the research gaps for those areas, such as inadequate communication, trust, system integration and knowledge management. Damian and Zowghi [17] investigated requirement engineering challenges and issues in distributed software development, especially across cultural boundaries and those existing in stakeholder organisations. These authors were able to construct a model on the requirement gathering process, including negotiation and specification. They illustrated the difficulty of the development process in DSD projects in terms of requirements engineering. Fowler [5] also wrote about his experience in adopting agile principles in an offshore development project. In this report, he discussed the importance of some factors in agile development (e.g. communication, cultural changes and documentation).

In addition, he presents the challenges as well as benefits of applying agile in offshore projects. He also discusses the current and future trend of agile offshore development, stating ‘Offshore development is very fashionable’. Rodríguez et al. [18] have conducted a study to investigate the tools and technologies that are used by distributed teams. They discussed the collaboration and integration of these technologies and the tools involved in software processes, such as IBM Jazz, Microsoft SharePoint and Google Apps. Their study included a comparison between these technologies and the benefits of tools and technologies in the software development process, like tracking systems, management features and calendar management. Table 2 shows a summary of the research that has been conducted to investigate the challenges and issues in different factors like communication, knowledge and requirements.

<table>
<thead>
<tr>
<th>Study</th>
<th>Contribution</th>
<th>Main challenges identified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Šmite and Borzovs [12]</td>
<td>Framework</td>
<td>Risk management of GSD</td>
</tr>
<tr>
<td>Wahyudin et al. [13]</td>
<td>Framework</td>
<td>Communication and exchange of information</td>
</tr>
<tr>
<td>Akbar et al. [14]</td>
<td>Model</td>
<td>Communication between people in DSD projects</td>
</tr>
<tr>
<td>Hossain et al. [4]</td>
<td>Framework</td>
<td>Risks of applying Scrum in GSD</td>
</tr>
<tr>
<td>Lee and Yong [10]</td>
<td>Framework</td>
<td>Project management and GSD</td>
</tr>
</tbody>
</table>
Table 2. Studies discussing the main challenges and factors related to DSD development

<table>
<thead>
<tr>
<th>Study</th>
<th>Challenges and factors discussed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coram and Bohner [8]</td>
<td>Project management, people, planning, documentation, development process.</td>
</tr>
<tr>
<td>Jiménez et al. [9]</td>
<td>Cultural differences, group awareness, configuration management, knowledge management, coordination, collaboration, project and process management, process support, risk management, quality and measurement.</td>
</tr>
<tr>
<td>Sengupta et al. [16]</td>
<td>Collaborative software tools, knowledge acquisition and management, testing in DSD, process and metrics issue</td>
</tr>
<tr>
<td>Damian and Zowghi [17]</td>
<td>Requirement engineering and its challenges in DSD, such as technology, culture and informal communication.</td>
</tr>
<tr>
<td>Fowler [5]</td>
<td>Cultural changes, requirements, documentation, costs, project management and future of DSD.</td>
</tr>
<tr>
<td>Rodríguez et al. [18]</td>
<td>Collaborative technologies like IBM Jazz, Microsoft SharePoint, Google Apps and IBM Lotus.</td>
</tr>
</tbody>
</table>

Table 3. Collected factors and issues related to using agile methods in DSD

| Collected factors’ effect on development in DSD |
|-----------------------------------------------|------------------------------------------------|
| • Communication                               | • Testing                                      |
| • Documentation                               | • Tools and technologies                       |
| • Project management                          | • Requirements                                 |
| • Risk management                             | • Time zone                                    |
| • Configuration management                    | • Knowledge management                         |
| • Trust                                       | • Process management                           |
| • Culture                                     | • Coordination                                 |
| • Collaboration                               | • Planning                                     |
| • Group Awareness                              | • People                                       |
| • Measurement                                 | • Integration                                  |
|                                              | • Quality                                      |

6. Research Method

There are many benefits to conducting a literature review. Hart in [19], mentioned that some reasons for reviewing the literature include:

- Identifying what has been done by other researchers and gap that require more research;
- Gaining a new perspective on the topic; and
- Finding out the important factors and variables relevant to the specific topic.

In this study, we reviewed the literature in the following fields: DSD, DAD and customisation software products. Our goal was to review studies that discuss agile methods with distributed software development and the main challenges and key factors involved in using agile software development to specialise software products across organisational boundaries.

The results were divided into two groups. The first group included studies that proposed frameworks and models to discuss some aspect of agile development with DSD. Table 1 shows the studies, main contributions and aspects or factors that have been discussed. The other group shows studies that have been conducted in terms of discussing challenges and the main aspects involved in using agile
software development and DSD. Table 3 gives list of factors and aspects that have been discussed in the two previous literature sections. While some of these have been presented in the proposed frameworks or models in section 1, such as project management, risk management and communication, no framework has been proposed to address all of these aspects in terms of organisational boundaries in distributed software development. Figure 3 illustrates those factors and how many times they have been mentioned in previous studies.

From the previous literature, we collected the key factors involved in using agile development across organisational boundaries for development and customisation software products from different fields. We then included them in the proposed framework (see figure 2).

<table>
<thead>
<tr>
<th>Communication</th>
<th>Project Management</th>
<th>Configuration Management</th>
<th>Knowledge Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Zone</td>
<td>Risk Management</td>
<td>Integration</td>
<td>Documentation</td>
</tr>
<tr>
<td>Trust</td>
<td>Process Management</td>
<td>User acceptance test</td>
<td>Communicating</td>
</tr>
<tr>
<td>People</td>
<td>Quality</td>
<td></td>
<td>Requirements</td>
</tr>
<tr>
<td>Cultural difference</td>
<td>Collaboration</td>
<td></td>
<td>Group Awareness</td>
</tr>
<tr>
<td>Collaboration</td>
<td></td>
<td></td>
<td>Tools &amp; Technology</td>
</tr>
</tbody>
</table>

7. Proposed Framework

As shown in the previous section, the mainstream of research in adopting an agile methodology in DSD projects has focused on different aspects of software development across organisational boundaries. For example, some researchers have proposed project management models such as [20], while others have proposed risk management models such as [7]. In this study, we put forward a framework (figure 3) for the adaptation of software products across organisational boundaries.

The proposed framework consists of seven components, specifically communication, communicating requirements, project management, knowledge management, documentation, the user acceptance test and configuration and integration management. These components cover management aspects and features of the software development process, such as documentation and testing.

7.1. Communication

Many researchers have described communication as being one of the main issues regarding distributed software development and agile development [1][5]. Communication is an important factor in agile development, emphasising face-to-face communication. To discuss communication as a key factor affecting specialisation software products across organisational boundaries, there are other aspects related to communication that is important so consider. These are described below.

i. Time zone: Time zone is an effective factor in communication, especially for teams distributed across countries and working hours in different organisations. Agile software development promotes people’s interaction during the development process, and this is difficult if there is a difference in time zone.

ii. Trust: During distributed development and development across organisations, the problem of face-to-face communication highlights another issue, which is trust between team members during different stages and forms, such as during requirement negotiations, information exchange, and conveying experiences [6].

iii. People: The manifesto for agile software development places great emphasis on people in the development of software using agile methods, prioritising ‘individuals and interactions over processes and tools’ [6]. Furthermore, the main motivation of the organisation in distributing their development projects is to look for highly skilled human resources [6]. Structures for
people in development or the specialisation process across organisational boundaries, including project managers, stakeholders and developers, are a very important factor in communication.

iv. Cultural differences: This is an important factor for distributing development and developing across organisations. Ref. [5] described cultural change as the ‘hardest’ part of adopting agile methods. In addition, culture can have an effect on communication, especially for GSD projects.

v. Collaboration: One of the four values of the agile manifesto is customer collaboration. Thus, agile software development emphasises and promotes the concept of collaboration with customers and other developers in order to support the software product and the development process. In the specialisation of software products across organisational boundaries, collaboration is very important in meeting the customer’s requirements and avoiding problems of distributed sites, as well as applying agile principles in the domain throughout an organisation.

7.2. Project Management

From the literature review, it can be seen that project management is a hot topic for researchers in terms of applying agile principles in distributed development, due to its effect on the development process [8,9]. These researchers have discussed different aspects of project management, as described below.

i. Risk management: Risk management has become a critical concern for professionals working in DSD [15]. Such concerns increase with the application of agile principles in DSD projects or across organisational boundaries. Thus, risk management has been described by researchers as one of the key challenges regarding DSD and DAD. In the proposed framework we assumed that risk management was a part of project management and defined it as a subcomponent under project management.

ii. Process management: In the proposed framework, the term ‘process’ refers to the software development process, which is clear in a traditional approach. For example, in the waterfall model, this refers to the analysis and design of the customer’s requirements, implementation, testing, delivery, and the documentation process. All these processes should be considered in terms of agile development and DSD across organisational boundaries. Owing to its importance in the software development process, it is addressed as an important component of project management.

iii. Quality: Although the software development process across organisational boundaries aims to achieve many advantages from using agile principles, such as reducing the time and cost of the development process and increasing productivity, the quality of the software products is also important. Quality is addressed as an important subcomponent of project management in the proposed framework.

iv. Planning: Planning, such as the planning before any iteration in order to sort out a priority list of the customer’s requirements, has an important place in agile development. However, that importance increases across boundaries in terms of arranging the distributed development process and planning across organisational boundaries. Thus, the proposed framework shows the importance of planning in project management.

7.3. Knowledge Management

In the development process of any software project or business, a huge amount of information and knowledge is used. The bulk of the information appears in various forms such as test cases, codes, comments and logs on source codes, project specifications, and developers’ and project team members’ experiences and comments. Furthermore, a high level of accuracy and availability should be ensured in relation to this information through the employment of useful tools. The proposed framework emphasises knowledge management and integration as key components in software development across distributed multi-teams.

i. Documentation: The manifesto for agile software development emphasises working software over comprehensive documentation. However, documentation in DSD and across organisations is required in order to counteract the lack of face-to-face interaction and informal communication. Ref [9] discusses documentation in GSD, and the authors emphasise the documentation process in DSD as being a part of the knowledge management.

ii. Communicating requirement: The proposed framework promotes management and software engineering practices through agile concepts. Agile software development supports face-to-face communication and interaction with customers over the complexity of the process. The framework supports the idea of allocating agile teams in the customer’s location in order to gather the customer’s requirements and using other agile teams in distributed development to deal with these requirements.

iii. Group awareness: Information should be available and equal for the people in the distributed agile development teams, such as developers at different sites. Thus, group awareness is a very important factor, and one of the manifesto’s values is an emphasis on individuals and interactive action.

iv. Tools and technologies: In the development process, using either traditional approaches or agile
methodology, some tools and technologies are employed. These tools can target the communication level or at the development and management level, and include tracking and documentation tools.

7.4. Configuration and Integration Management

The coordination and synchronisation of the source code and software versions is an important step for the development of any iteration. However, the integration and version control of the source code becomes more complex with distributed projects across multi-teams and organisations. Therefore, configuration management is a key component of the proposed framework, and it guides the developers and project managers at the customer’s location so that they consider this step and make sure that the new version of any iteration is integrated with the customer’s needs and the customer’s environment in terms of both platforms and hardware.

i. Integration: For the specialisation of software products across organisational boundaries, there are multiple versions to meet customers’ change requests or new requirements. Thus, the integration process is emphasised in order to ensure that the new version is compatible with the current version or that the correct version can be customised for the organisational system. In addition, emphasis is placed on using the version control concept and technology to move smoothly from one version to another in the specialisation process across customer boundaries.

ii. User acceptance test: Most software testing occurs at the time of development and is performed by the development team, such as unit tests and integration tests. However, user acceptance tests require sharing customers in order to ensure that the software meets all customer requirements. Thus, the user acceptance test is the one of key components of the proposed framework, serving to address the testing process across organisational boundaries.

8. Discussion and Conclusion

While agile methods promote the iterative development of software products as incremental and integrated releases, applying the agile concept to distributed projects produces new challenges and issues. Some of these challenges occur in terms of teams or organisational boundaries. There are different reasons for these challenges, such as cultural differences, distance, or a low level of communication. In fact, the complexity of the development process produces challenges if there are many different customers at different distributed locations supported by a central development team. The framework aims to address these challenges and issues related to developing software across organisational boundaries. In addition, a good framework design accommodates the key aspects of a problem. Thus, we have proposed a framework to accommodate the key aspects of agile development for the specialisation of software products across organisational boundaries.

The proposed framework covers the technical aspects through software development practices, and these include communication requirements, documentation, configuration and integration management and user acceptance test components. Furthermore, it covers management aspects in terms of project management, knowledge management and communications. The proposed framework will serve as a guide for project managers and developers in considering the key aspects and issues of using the agile approach to developing or maintaining software across organisational and cultural boundaries.

9. References


