Requirement Barriers to Implement the Software Projects in Agile Development

Deepak Kumar and Saru Dhir

Abstract The success of an organization is to deliver the good-quality products as per the client's needs. But few organizations are unable to deliver the successful product due to number of software barriers. The research is based on the study and analysis of different requirement barriers, which causes problem in agile implementation. Several authors identified the barriers for successful implementation of software, but none have found the barriers at the initial stage of requirements. The motivation behind this work is to classify the main requirement barriers to the effective implementation of software projects in agile development. For the study, interviews were conducted with developers and testers. The results recognized the key barriers and it will deliver the roadmap to managers to take suitable steps to overwhelm the major barriers to effective software implementation.

Keywords Software requirement · Agile project implementation · Interpretive structural modelling (ISM) · Requirement barriers · Agile methodology · Requirement engineering

1 Introduction

Software development is a teamwork where each member has different roles such as software development, testing, project analysis etc. A project quality, delivery time and cost of delivered product specify its success and failure rate. Agile software development is a leading approach in software organizations during last few years; to fulfil the client's need of producing quick, better and cost-effective solutions. In agile development, client's have direct interaction with team members of project to improve the communication among them. As the concept of quality is relative, the aim of this work is to comprehend the factors that affect the failure of software project and its quality with regards to Agile Software Development (ASD).

D. Kumar (✉) · S. Dhir
Amity University, Noida, Uttar Pradesh, India

A. G. Aggarwal et al. (eds.), *Optimization Models in Software Reliability*, Springer Series in Reliability Engineering,
https://doi.org/10.1007/978-3-030-78919-0_15

The popularity of agile development is increased during the last years; despite of that, agile methods are also criticized for successful delivery on functional requirements and on neglecting the quality requirements. Ignorance of quality requirements becomes the result of non-satisfaction of user's requirements.

In common practice, all the individuals or groups faces challenges during the implementation, which reduces the performance of the system (Boehm and Turner 2005). For a successful service workflow, requirement management should be considered as an initial point. Different success and failure factors were identified in the project implantation (Dhir et al. 2017, 2019). For a successful implementation of the software project, requirement elicitation and management are a significant task (Dhir and Kumar 2015; Dhir et al. 2019).

Software complexity and their issues are fully involved in requirement and design factors. Project requirements are chosen rendering to the client's end product's need (Rai and Dhir 2014; Rajagopal et al. 2005). An approach was planned to collect the requirements and appropriate steps were taken to eradicate the barriers (Rajagopal et al. 2005). It is necessary to remove the barriers and ensured about the software requirements should be according to the customer's requirements to maintain the complete software quality.

The topical survey of Standish Group (2014) represents the success reason of the project is: requirements statement, user contribution and management support. This survey report considered the standards of a project that are eventually based on the requirement management.

During agile implementation, requirements are adaptive in nature and it is also not easy to maintain the requirement specification documents. Requirements are continuously change in agile development, due to different reasons such as: missing requirements, customers lack knowledge, market change and bug fixing.

Barriers in software requirements would affect the budget and quality of product. Number of barriers means number of risk factors increases for the failure of software project. So, it is essential to recognize the different barriers during implementation to improve the functionality, including efficiency, performance, quality and security of the system.

Requirement documentation is the key deliverable of requirements for the implementation of software. Lack of documentation is also another barrier as the documentation is not possible in agile.

A survey was directed with the help of industry experts and identifies the barriers during software implementation. Industry experts were the software developers, testers and leaders who implemented the software using agile methodology (Dhir and Kumar 2015).

The research recognizes the most significant barriers that affects the effectiveness and quality of software implementation. This work signifies the ISM practice to classify the connection between the diverse barriers and find the most significant barrier that affect the software implementation using agile methodology. The research work presents a roadmap for managers of an organization to resolve the issues influencing during the agile development, so that the management or senior members can take appropriate actions to resolve these issues.

2 Literature Survey

There are different studies have been focused on the impact and acceptance of agile development in different organizations (Rai and Dhir 2014; Aggarwal and Dhir 2013). Authors discussed about the agile principles and the impact on current software development.

There are different uncertainties occurred during project planning such as uncertain estimates, requirements management and prioritization, ignorance of non-functional requirement. Other studies have been evaluated directing on how the acceptance of agile issues can be resolved (Misra et al. 2009). Qu et al. (2012) identified different risk factors for project management. Result analysis were analysed to evaluate the perplexing possible relationship between risk factors using interpretative structural modelling.

Alsaqaf et al. (2018) identified the nine challenges faced by the agile developers in large distributed projects that harmed the implementation of quality requirements. There are different challenges in software development such as: organization environment, communication and time differences in distributed environment.

Srinivasan and Lundqvist (2009) faced the challenges, that the agile team was not involved in the initial estimation of project due to which ambiguous requirements become poor in quality and schedule overruns. The literature lacks the obtainability of framework that can identify the barriers of requirement elicitation in agile software development, where the software requirements are changed very frequently (Srinivasan and Lundqvist 2009).

Conboy et al. (2011) conducted a study focusing on the challenges by the people in the agile development such as transparency by the team members, lack of business knowledge among team members etc. (Conboy et al. 2011). The selection of accurate requirement is a big challenge. Overall quality of the product depends upon the selection of requirements. A survey analysis was done by different practitioners to identify the prevalence's and challenges using agile software development. Statistical analysis was executed to identify the significant value of agile implementation over the traditional development (Dhir et al. 2017).

There are different decision-making techniques are applied for improvement of selection, such as Analytic Hierarchy Process (AHP), Paired Comparison Analysis, Game Theory, Multiple Criteria Decision Analysis and Interpretive structural modelling (ISM). ISM is an interpretive because it decides the findings of the group, how the variables are associated (Dhir et al. 2017).

Researcher studied and analysis among the barriers on a case study in 'just in time' production using ISM. Paper described the hidden barriers to 'just in time' production using ISM (Jadhav et al. 2015).

3 Methodology

ISM methodology has been introduced by prof. John Warfield with an objective to examine socio economics system issues and understand the complex relationship in different areas.

It has been demonstrated that ISM is a fixed decision-making tool that permits entities, groups and organizations to develop a connection for determining the complex situation and signifies the relationship through binary matrix (Huang et al. 2005; Warfield 1974). This method is used to understand composite structure with a simple geometric model articulating the complex relationship between numerous elements.

The usage of this method ranges for modelling systems interconnected to planning, decision making, competitive analysis, process re-engineering and many more. Statistical techniques provide the quantitative results using the large number of variables, whereas ISM provides the relationship among qualitative variables. The relationships between different variables are established with the repetition of questions, such as: 'Does target A supports to accomplish target B?', 'Does target A supports to accomplish target B?' for all pair of elements. According to the established relationship, a structure is created which is modelled through a digraph.

Steps for model development using ISM methodology are (Jadhav et al. 2015):

- Identify the barriers.
- Establish the relationship between barriers by conducting the interviews. Generate a self-structural interaction matrix (SSIM) of variables showing pairwise connection among barriers.
- Generate an initial reachability matrix (RM) and eliminate transitive relations in final reachability.
- Partition reachability matrix into various levels in different iterations.
- Plot a directed graph (digraph) in view of relations.
- Modify the digraph into an ISM model.
- Analyse an ISM model and examine hypothetical inconsistency.

Step 1: Recognize the Barriers

A wide analysis study is finished to recognize the barriers for the agile implementation in software development. The survey is directed during agile implementation to recognize the barriers for avoiding the effective and productive implementation. Survey data is collected by conducting the interviews with many industry experts. Interviews and discussions are done to control the barriers during the agile adoption and implementation. The main concern includes ever changing requirements, communication gap, undefined goals, incomplete requirements, lack of plan, shortage of expert members, requirement management, budgetary constraint, lack of documentation and ignorance of non-functional requirement. The barriers are listed as shown below in Table 1.

Table 1 Barriers to successful agile implementation

Barriers	Description
B_1	Lack of communication
B_2	Unclear goals
B_3	Incomplete requirement
B_4	Lack of planning
B_5	Ever changing requirement
B_6	Lack of expertise
B_7	Requirement management
B_8	Budgetary constraint
B_9	Lack of documentation
B_{10}	Ignorance of non-functional requirement

Step 2: Structural Self Interaction Matrix (SSIM)

A team of five members containing of industry experts especially developers, testers were interviewed to recognize the relationship between barriers. Following keywords signify the connection among barriers:

"V" signifies 'a' and helps to attain 'b'.

"A" signifies 'b' and helps to attain 'a'.

"X" signifies 'a' and 'b' helps to attain each other.

"O" signifies both 'a' and 'b' are not linked.

SSIM was implemented according to the known relationships in Table 2 and shows, 'a' and 'b' signifies rows and columns and it also represents the interrelationships between those 10 barriers. Table 3 is then used to generate reachability matrix.

Table 2 Structural self interaction matrix

Attribute (B_i)	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}
B_1	1	V	A	A	O	A	V	V	V	V
B_2		1	A	V	O	V	V	V	V	A
B_3			1	V	A	V	V	V	V	V
B_4				1	A	V	V	V	V	V
B_5					1	V	V	V	V	V
B_6						1	V	V	V	V
B_7							1	V	A	V
B_8								1	A	V
B_9									1	X
B_{10}										1

Table 3 Initial reachability matrix

Attribute (B_i)	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}
B_1	1	1	0	0	0	0	1	1	1	1
B_2	0	1	0	1	0	1	1	1	1	0
B_3	1	1	1	1	0	1	1	1	1	1
B_4	1	0	0	1	0	1	1	1	1	1
B_5	0	0	1	1	1	1	1	1	1	1
B_6	1	0	0	0	0	1	1	1	1	1
B_7	0	0	0	0	0	0	1	1	0	1
B_8	0	0	0	0	0	0	0	1	0	1
B_9	0	0	0	0	0	0	1	1	1	1
B_{10}	0	1	0	0	0	0	0	0	0	1

Step 3: Generate initial reachability matrix (IRM)

Generate IRM Table 3 from Table 2 SSIM.

If (a, b) in SSIM i.e. in Table 2 is "V"; then reachability matrix converts into 1 and (b, a) converts 0.

If (a, b) in SSIM i.e. in Table 2 is "A"; then reachability matrix converts into 0 and (b, a) converts 1.

If (a, b) in SSIM i.e. in Table 2 is "X"; then reachability matrix converts 1 and (b, a) converts 1.

If (a, b) in SSIM i.e. in Table 2 is "O"; then reachability matrix converts 0 and (b, a) converts 0.

Table 3 represents an IRM of 1's and 0's and currently it contains transitive relations.

The transitive relationship of Table 3 matrix is planned by squaring the matrix and will be transitive, if the resulting value of squared matrix has 1 which was earlier value 0. The transitive relation for reachability matrix is verified. Final reachability matrix (FRM) after verifying the transitive values is given in Table 4 and demonstrated in rows and columns, wherever rows specify driving power and columns indicate dependence power (Table 5).

Row wise barriers are the driving power to each barrier. Dependence power is sum of barriers (Tables 6, 7, 8, 9, 10 and 11).

Step 4: Dividing into levels

Both reachability and antecedent set to each barrier is estimated. Later, the connection of reachability and antecedent sets is estimated to all barriers. If the connection and reachability set are similar, then allot the level in the ISM Model. This process is iteratively estimated for whole barriers till the level for each barrier is recognized. Table 5 validates the early iteration with barriers B8 and B10 creating the first level. The whole levels to each of the barriers are signified in Table 12. Level I barriers have the lowermost whereas level VII has the uppermost driving power.

Table 4 Final reachability matrix

Attribute (B_i)	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	Drive power
B_1	1	1	0	1	0	1	1	1	1	1	8
B_2	1	1	0	1	0	1	1	1	1	1	8
B_3	1	1	1	1	0	1	1	1	1	1	9
B_4	1	1	0	1	0	1	1	1	1	1	8
B_5	1	1	1	1	1	1	1	1	1	1	10
B_6	1	1	0	0	0	1	1	1	1	1	7
B_7	0	0	0	0	0	0	1	1	0	1	3
B_8	0	0	0	0	0	0	0	1	0	1	2
B_9	0	1	0	0	0	0	1	1	1	1	5
B_{10}	0	1	0	1	0	1	1	1	1	1	7
Dependence power	6	8	2	6	1	7	9	10	8	10	67

[a]Entries are included to incorporate transitivity

Table 5 RM partitioning iteration 1

Attribute (B_i)	Reachability set	Antecedent set	Intersection set	Level
B_1	B_1, B_2, B_4, B_6, B_7, B_8, B_9, B_{10}	B_1, B_2, B_3, B_4, B_5, B_6, B_7	B_1, B_2, B_4, B_6, B_7	
B_2	B_1, B_2, B_4, B_6, B_7, B_8, B_9, B_{10}	B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9, B_{10}	B_1, B_2, B_4, B_6, B_7, B_9, B_{10}	
B_3	B_1, B_2, B_3, B_4, B_6, B_7, B_8, B_9, B_{10}	B_3, B_5	B_3	
B_4	B_1, B_2, B_4, B_6, B_7, B_8, B_9, B_{10}	B_1, B_2, B_3, B_4, B_5, B_{10}	B_1, B_2, B_4, B_{10}	
B_5	B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9, B_{10}	B_5	B_5	
B_6	B_1, B_2, B_6, B_7, B_8, B_9, B_{10}	B_1, B_2, B_3, B_4, B_5, B_6, B_{10}	B_1, B_2, B_6, B_{10}	
B_7	B_7, B_8, B_{10}	B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_9, B_{10}	B_7, B_{10}	
B_8	B_8, B_{10}	B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9, B_{10}	B_8, B_{10}	I
B_9	B_2, B_7, B_8, B_9, B_{10}	B_1, B_2, B_3, B_4, B_5, B_6, B_8, B_9, B_{10}	B_2, B_9, B_{10}	
B_{10}	B_2, B_4, B_6, B_7, B_8, B_9, B_{10}	B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9, B_{10}	B_2, B_4, B_6, B_7, B_8, B_9, B_{10}	I

Table 6 RM partitioning iteration 2

Attribute (B_i)	Reachability set	Antecedent set	Intersection set	Level
B_1	B_1, B_2, B_4, B_6, B_7	$B_1, B_2, B_3, B_4, B_5, B_6, B_7$	B_1, B_2, B_4, B_6, B_7	
B_2	$B_1, B_2, B_4, B_6, B_7, B_9$	$B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_9$	$B_1, B_2, B_4, B_6, B_7, B_9$	II
B_3	$B_1, B_2, B_3, B_4, B_6, B_7, B_9$	B_3, B_5	B_3	
B_4	$B_1, B_2, B_4, B_6, B_7, B_9$	B_1, B_2, B_3, B_4, B_5	B_1, B_2, B_4	
B_5	$B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_9$	B_5	B_5	
B_6	B_1, B_2, B_6, B_7, B_9	$B_1, B_2, B_3, B_4, B_5, B_6$	B_1, B_2, B_6	
B_7	B_7	$B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_9$	B_7	II
B_9	B_2, B_7, B_9	$B_1, B_2, B_3, B_4, B_5, B_6, B_9$	B_2, B_9	

Table 7 RM partitioning iteration 3

Attribute (B_i)	Reachability set	Antecedent set	Intersection set	Level
B_1	B_1, B_4, B_6	B_1, B_3, B_4, B_5, B_6	B_1, B_4, B_6	
B_3	B_1, B_3, B_4, B_6, B_9	B_3, B_5	B_3	
B_4	B_1, B_4, B_6, B_9	B_1, B_3, B_4, B_5	B_1, B_4	
B_5	$B_1, B_3, B_4, B_5, B_6, B_9$	B_5	B_5	
B_6	B_1, B_6, B_9	B_1, B_3, B_4, B_5, B_6	B_1, B_6	
B_9	B_9	$B_1, B_3, B_4, B_5, B_6, B_9$	B_9	III

Table 8 RM partitioning iteration 4

Attribute (B_i)	Reachability set	Antecedent set	Intersection set	Level
B_1	B_1, B_4, B_6	B_1, B_3, B_4, B_5, B_6	B_1, B_4, B_6	IV
B_3	B_1, B_3, B_4, B_6	B_3, B_5	B_3	
B_4	B_1, B_4, B_6	B_1, B_3, B_4, B_5	B_1, B_4	
B_5	B_1, B_3, B_4, B_5, B_6	B_5	B_5	
B_6	B_1, B_6	B_1, B_3, B_4, B_5, B_6	B_1, B_6	IV

Table 9 RM partitioning iteration 5

Attribute (B_i)	Reachability set	Antecedent set	Intersection set	Level
B_3	B_3, B_4	B_3, B_5	B_3	
B_4	B_4	B_3, B_4, B_5	B_4	V
B_5	B_3, B_4, B_5	B_5	B_5	

Table 10 RM partitioning iteration 6

Attribute (B_i)	Reachability set	Antecedent set	Intersection set	Level
B_3	B_3	B_3, B_5	B_3	VI
B_5	B_3, B_5	B_5	B_5	

Table 11 RM partitioning iteration 6

Attribute (B_i)	Reachability set	Antecedent set	Intersection set	Level
B_5	B_5	B_5	B_5	VII

Table 12 Level of requirement barriers

Attribute (B_i)	Reachability set	Antecedent set	Intersection set	Level
B_1	B_1, B_2, B_4, B_6, B_7, B_8, B_9, B_{10}	B_1, B_2, B_3, B_4, B_5, B_6, B_7	B_1, B_2, B_4, B_6, B_7	IV
B_2	B_1, B_2, B_4, B_6, B_7, B_8, B_9, B_{10}	B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9, B_{10}	B_1, B_2, B_4, B_6, B_7, B_9, B_{10}	II
B_3	B_1, B_2, B_3, B_4, B_6, B_7, B_8, B_9, B_{10}	B_3, B_5	B_3	VI
B_4	B_1, B_2, B_4, B_6, B_7, B_8, B_9, B_{10}	B_1, B_2, B_3, B_4, B_5, B_{10}	B_1, B_2, B_4, B_{10}	V
B_5	B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9, B_{10}	B_5	B_5	VII
B_6	B_1, B_2, B_6, B_7, B_8, B_9, B_{10}	B_1, B_2, B_3, B_4, B_5, B_6, B_{10}	B_1, B_2, B_6, B_{10}	IV
B_7	B_7, B_8, B_{10}	B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_9, B_{10}	B_7, B_{10}	II
B_8	B_8, B_{10}	B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9, B_{10}	B_8, B_{10}	I
B_9	B_2, B_7, B_8, B_9, B_{10}	B_1, B_2, B_3, B_4, B_5, B_6, B_8, B_9, B_{10}	B_2, B_9, B_{10}	III
B_{10}	B_2, B_4, B_6, B_7, B_8, B_9, B_{10}	B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9, B_{10}	B_2, B_4, B_6, B_7, B_8, B_9, B_{10}	I

Barriers are classified into four classes:

1. Autonomous barriers.
2. Dependent Barriers.
3. Linkage Barriers.
4. Independent Barriers.

Step 5: Digraph

Digraph is created representing the directed link among various barriers. Here, level VII generates the root node, means B5 is root driving power besides controls further barriers.

Figure 1 illustrate the digraph having the partition into diverse levels through I to VII.

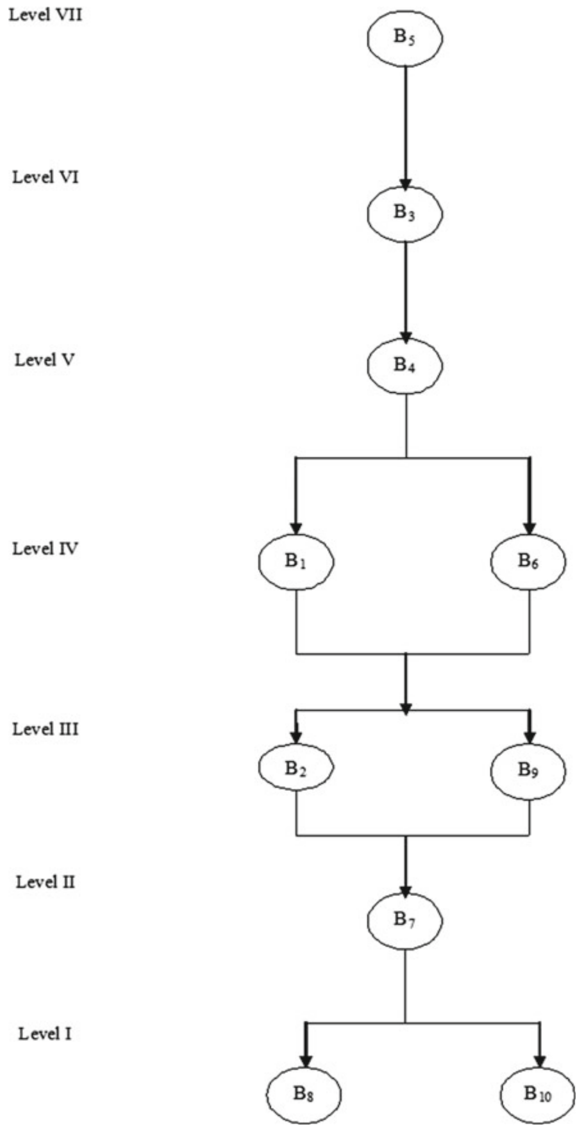

Fig. 1 Digraph representing inter-relationship among barriers

Step 6: Convert the digraph to ISM model

Figure 2 indicates the ISM model to successfully implement agile software. Figure 2 represents the ever-changing requirement is the major barrier followed by incomplete requirement, lack of planning etc.

Step 7: Analyse the ISM model

ISM model was finally reviewed by industry experts and approved the results.

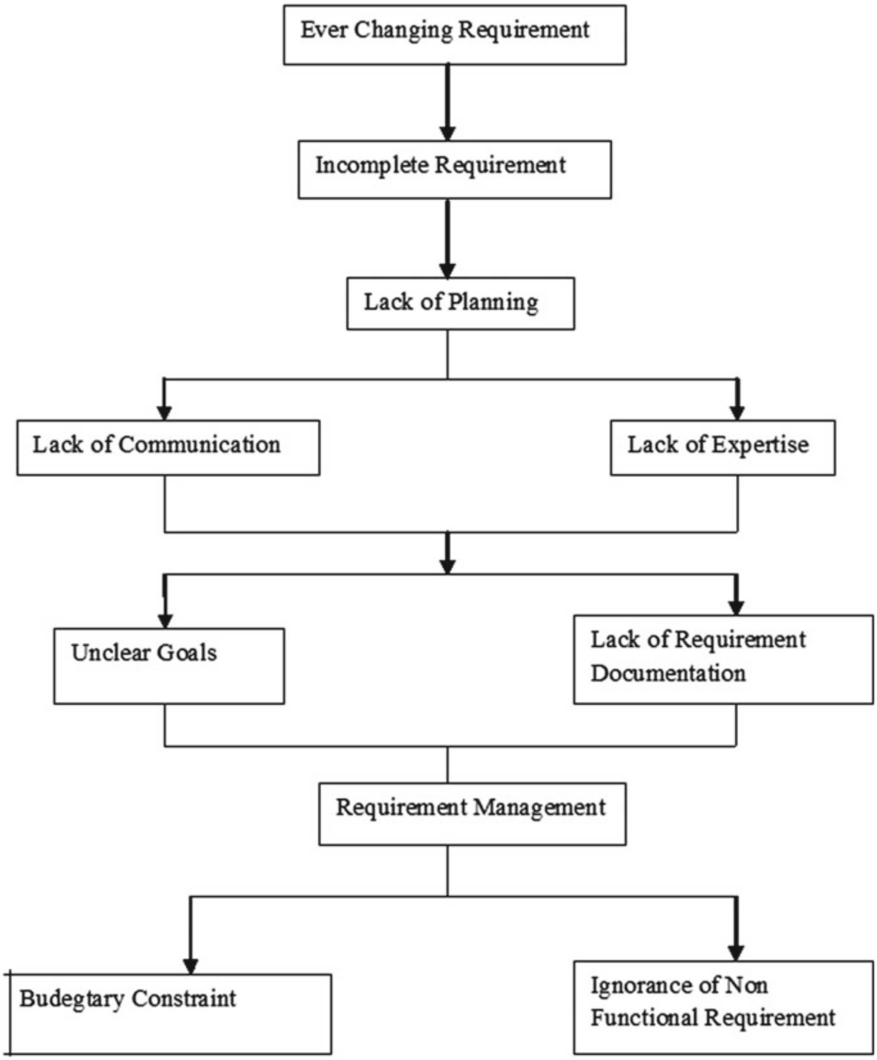

Fig. 2 ISM Model

4 Conclusion

The paper analyzed barriers in agile implementation and ISM model is represented by using Interpretive Structural Modelling (ISM) technique. It is concluded from the ISM model, that the ever-changing requirements (frequent change in project requirements from the customer side) forms the main barrier preventing the successful implementation of agile. Ever-changing requirement as barrier B5 the uppermost driving power besides the lowermost dependence power value creating the main barrier persuading altogether further barriers. Hence, it's vital for all project managers, developers and team to coordinate on the time, so that its complete effect can be reduced, and a quality product can be delivered on time. Thus, it is necessary to provide the resources to complete the client's ever-changing requirements and incomplete requirements on time.

References

Aggarwal S, Dhir S (2013) Ground axioms to achieve movables: methodology. Int J Comput Appl 69(14)

Alsaqaf W, Daneva M, Wieringa R (2018) Understanding challenging situations in agile quality requirements engineering and their solution strategies: insights from a case study. In: 2018 IEEE 26th international requirements engineering conference (RE). IEEE, pp 274–285

Boehm B, Turner R (2005) Management challenges to implement agile processes. Traditional development organizations

Conboy K, Coyle S, Wang X, Pikkarainen M (2011) People over process: key people challenges in agile development

Dhir S, Kumar D (2015) Factors persuading nuts and bolts of agile estimation. Adv Sci Lett 21(10):3118–3122

Dhir S, Kumar D, Singh V (2017) Requirement paradigms to implement the software projects in agile development using analytical hierarchy process. Int J Decis Support Syst Technol (IJDSST) 9(3):28–41

Dhir S, Kumar D, Singh V (2019) Success and failure factors that impact on project implementation using agile software development methodology. Software engineering. Springer, pp 647–654

Huang J-J, Tzeng G-H, Ong C-S (2005) Multidimensional data in multidimensional scaling using the analytic network process. Pattern Recogn Lett 26(6):755–767

Jadhav J, Mantha S, Rane S (2015) Analysis of interactions among the barriers to JIT production: interpretive structural modelling approach. J Ind Eng Int 11(3):331–352

Misra SC, Kumar V, Kumar U (2009) Identifying some important success factors in adopting agile software development practices. J Syst Softw 82(11):1869–1890

Qu Y, Yuan M-J, Liu F (2012) The risk factor analysis for software project based on the interpretative structural modelling method. In: 2012 international conference on machine learning and cybernetics. IEEE, pp 1019–1024

Rai P, Dhir S (2014) Impact of different methodologies in software development process. Int J Comput Sci Inf Technol 5(2):1112–1116

Rajagopal P, Lee R, Ahlswede T, Chiang C-C, Karolak D (2005) A new approach for software requirements elicitation. In: Sixth international conference on software engineering, artificial intelligence, networking and parallel/distributed computing and first ACIS international workshop on self-assembling wireless network. IEEE, pp 32–42

Srinivasan J, Lundqvist K (2009) Using agile methods in software product development: a case study. In: 2009 sixth international conference on information technology: new generations. IEEE, pp 1415–1420

The Standish Group (2014), The Standish group: the chaos report. Proj. Smart. 16

Warfield JN (1974) Developing interconnection matrices in structural modeling. IEEE Trans Syst Man Cybern 1:81–87