Modelling and measuring attributes influencing agile implementation in an enterprise using structural equation modelling

Abhishek Srivastava and
Deepti Mehrotra*

Department of Information Technology,
Amity Institute of Information Technology,
Amity University, Noida,
Uttar Pradesh, India
Email: abhishek.sri13@gmail.com
Email: dmehrotra@amity.edu
*Corresponding author

P.K. Kapur

Amity Centre for Interdisciplinary Research,
Amity University, Noida,
Uttar Pradesh, India
Email: pkkapur1@gmail.com

Vijay Kumar

Department of Mathematics,
AIAS, Amity University, Noida,
Uttar Pradesh, India
Email: vijay_parashar@yahoo.com

Abstract: Today's software applications deployed in an enterprise caters to the complex business processes, integrate various business units and address requirements of global customer base. Critical characteristics of success of Agile development has to be taken into account in order to make this process of software development a rewarding one. In this paper, a significant perception is show cased for the adept to analyse and measure the specified Agile attributes using statistical analysis. A thorough analysis of 16 attributes is done in order to find vital independent factors and measure them steer the maturity of Agile employment in a business. For this purpose, a framework has been suggested which accelerates the process. An observation of the association among 16 attributes was done. With the help of Exploratory and Confirmatory Factor Analysis, a structural model was set up with affirmation using Structural Equation Modelling. These important attributes were measured and the maturity of Agile execution in a project was found with the help of Explanatory factor analysis. The 16 attributes were categorised in four latent variables: Planning, Execution, Tools and Infrastructure and People. This was done using exploratory and Confirmatory Factor Analysis. Moreover, Structural Equation Modelling was done to deduce thirteen key independent attributes that have an impact on other attributes. For the purpose of measuring these key attributes, Structural equation modelling was implemented using AMOS and it was determined that three of the attributes were working below the threshold.

Keywords: agile; continuous delivery; exploratory and confirmatory factor analysis; structural equation modelling; enterprise applications.

Reference to this paper should be made as follows: Srivastava, A., Mehrotra, D., Kapur, P.K. and Kumar, V. (2022) 'Modelling and measuring attributes influencing agile implementation in an enterprise using structural equation modelling', *Int. J. Applied Management Science*, Vol. 14, No. 1, pp.3–22.

Biographical notes: Abhishek Srivastava did BTech and MTech from GGSIPU and is currently pursuing the PhD from Amity University and working as Asst Prof in Amity School of Engineering and Technology. He has more than ten years of experience in research, teaching and industry. He had published more than 30 papers in international refereed journals and conference proceedings. His current research interests include software reliability growth modelling, Agile testing and cyber security. He is a life member of Society for Reliability Engineering, Quality and Operations Management (SREQOM).

Deepti Mehrotra did PhD from Lucknow University and currently she is working as Professor in Amity School of Engineering and Technology. She has more than 20 year of experience in research, teaching and content writing. She had published more than 100 papers in international refereed journals and conference proceedings. She is editor and reviewer for many books, refereed journals and conferences. She is regularly invited as resource person for FDPs and invited talk in national and international conference.

P.K. Kapur is a Professor at Amity International Business School, Amity University, Noida. He is the former Head of the Department of Operational Research and former Dean of the Faculty of Mathematical Sciences, University of Delhi. He has been the President of Society for Reliability Engineering, Quality and Operations Management since 2000 and former President of Operational Research Society of India. He has published extensively in Indian journals and abroad in the areas of marketing, hardware reliability, queuing theory and maintenance and software reliability (more than 350 papers). He has recently been awarded with the 'Best Teacher in Operations Management' by Bloomberg; UTV; Lifetime Achievement Award for contributions to the field of Software Reliability Engineering.

Vijay Kumar received his MSc in Applied Mathematics and MPhil in Mathematics from IIT Roorkee, India in 1998 and 2000, respectively. He has completed his PhD from the Department of Operational Research, University of Delhi. Currently, he is an Associate Professor in Department of Mathematics, Amity University, Noida, India. He has published more than 40 research papers in the areas of software reliability, mathematical modelling and optimisation in international journals and conferences of high repute. His current research interests include software reliability growth modelling, optimal control theory and marketing models in the context of innovation diffusion theory. He has edited special issues of *IJAMS* and *RIO* journal. He is an editorial board member of *IJMEMS*. He is a life member of Society for Reliability Engineering, Quality and Operations Management (SREQOM) and IEEE.

1 Introduction

One of the greatest challenges software industry faces today is to address the ever-changing need of the customers and to develop a quality product. Agile software development process can handle if any requirement change is proposed during software development. The focus of Agile is on co-operative work for the project lifecycle with an added flexibility and expected changes, which are in contrast to the older traditional approaches. Agile process is considerate of the ability of reacting quickly and effortlessly towards the changes. User's feedback that contains a detailed requirement, containing enough information about the project has been taken so that the developers can implement it. Traditional approaches led a software be undergone through many sequential processes of a cycle involved in Software Development Life Cycle (Kumar et al., 2016a, 2016b, 2017). Steps included in the software development process are: requirement analysis, defining it properly, chalking a plan, start a build, testing and finally deployment. However, Agile is less documentation focused and more code focused, puts the spotlight on only the development and quality assurance aspects. Nowadays, the business environment is constantly and quickly changing and in this hard and fast reality, agile and its usage is becoming a vital need, not only because the customers require faster turnaround but also because of the higher quality needs. The important components of Agile are Continuous Integration, Continuous Testing, Continuous Delivery and Continuous Monitoring. Several attributes affect the implementation of Agile in an enterprise. In agile framework to handle the dynamic software model, developer and tester work hand in hands. The Testing in agile environment needs to strategies such that it can enhance when user stories are added, modified or deleted. Such changes result in running the test case leading to waste of time and manpower involved in testing.

Much of the literature review focuses on how Agile is deployed with less focus done on the factors that affect the implementation of Agile methodology in particular those that may critical for developing , planning an efficient software testing strategy. These factors are key attributes that ensure the quality that is our one of the prime objectives of adopting agile methodology. Thus, recognition of key independent factors that affects other attributes as well as Agile implementation is conjectural.

The analysis done in this paper by performing the explanatory analysis of the factors significantly influences the agile software development and deployment. A skeletal model by means of statistical analysis has been proposed and its impact in the real-time environment is done to verify the model. A skeletal model would make it easy for practitioners to identify the connection among these factors and recognise significant factors, which affect other factors in Agile execution. The experts will be able to find out the attributes that necessarily be needed for improving the Agile process. These improvements would help prevent disappointment of customers, financial losses and complete enhancements of quality. This research supports significantly in the area of Agile to address the research gaps.

1 *Selecting the most critical factors*: This manuscript describes about the factors that influence the structural context of software testing in agile development.

2 *Lack of validation and authentication of Agile attributes*: This manuscript focuses on analysing 16 Agile attributes and identifying the important Agile attributes by using Exploratory and Confirmatory analysis of factors and SEM (Structural Equation Modelling).

3 *Measurement of maturity of Agile implementation*: The manuscript focuses on the result of Structural Equation modelling results assessing approach to analyse significant Agile factors in a process. Our main goal for choosing the SEM is to explain the variance on dependent variable. By Using SEM, we are not concern about the specific agile factor that contributes to the prediction. Rather, we are willing to find agile factors in aggregate and try to maximise the variance accounted for predicting the success.

Firstly, the factors that significantly affect the software development process were identified using extensive literature review to strengthen the study further. Industry expert opinions were taken so that holistic set of factors were selected which provided both theoretical and practical insight of agile development process. The 16 Agile attributes were shortlisted to analyse in detail to understand their influence on each other. To find meaningful patterns within a data set, Exploratory Factor Analysis is useful.

This motivated us to extend our research by EFA that is used to understand the hierarchical importance of determinants of defects in software development. In this study, we deliberate on unnoticed attributes or the hidden factors to form the hypothetical co-relation on the selected factors. The hypothetical model proposed is confirmed using CFA. To know and analyse the structural theory and authenticate the structural model, Structural Equation modelling is done. The relationship among selected attributes is formed through the statistical analysis. It also determines the key independent attributes. The strategy decides the maturity level of the Agile attributes as far as arithmetic score and features the low performing attributes.

With the intensive literature review, we have found that there is a need to find the comprehensive evaluation of 16 Agile testing attributes to decide the key independent factors that measure them to decide the reliability of Agile execution. This paper gives essential bits of knowledge in the field of Agile by leading point-by-point evaluation of 16 credits and utilising factual investigation to decide the key independent factors and estimating to use CFA and EFA to decide the reliability of Agile. Section 2 gives the background study of the subject, Section 3 clarifies Agile and explains the properties that impacts Agile execution, Section 4 displays the planned procedure in depth, and Section 5 represents the results got after the utilisation of planned technique pursued by the discourses and Conclusion in Section 6.

2 Background

Software excellence in an organisation has become the need of the hour for any organisation, as faults in software not only adversely affect financial drops never the less it might similarly adversely distress social life. This manuscript is motivated around calculating and measuring numerous attributes of excellence in an origination thus allowing consultants to succeed developments in the quality. Numerous case studies have been reported previously to extent the excellence of attributes pre-dominant in policy, construction, code, testing and processes features of software. However, limited evidence that published work promotes specific objectives of Agile testing attributes. This report leads the wide-ranging idea of software engineering, software quality, enterprise applications in perspective with agile and the necessity of calculating the quality. It also offers the outline of previous works and current study by previous scholars in the area of calculating quality. Lastly, the manuscript exhibits the gap analysis in the current research and lists the purposes of the study.

2.1 Review of past work done in the field of Agile

Agile flexibility provides ability to the organisations to adapt the changes in inputs and prosper in a competitive atmosphere of continuous delivery by reacting quickly and effectively to dynamical markets (Cho et al., 1996) within the product development method. There are many agile factors that impact the launch of a replacement product into production, like provide capability, producing capability, and technologies capability, etc. Since final product encompasses components from many alternative suppliers, provider involvement could increase the gracefulness of transferring new product into production. Practitioners identified several attributes in agile environment that offer capability within the provider integration method and permits the provider to set up for future product-development efforts and speed up the launch of a replacement product into production. Wynstra and Ten Pierick (2000) declared that sanction native provider involvement in New Product Development (NPD) has become a standard approach in rising project effectiveness (product prices and quality) and project strength (development prices and time). A number of the authors have compared Agile by different procedures like waterfall method, they had recognised the actual events that the ideas of Agile are inter-related. Cockburn (2002) declared that empirical studies into agile characteristics are required in order that sensible tools can be applied to improve design documentation. (Ambler, 2008) suggested a vision for minimising the effort for detailed planning that was required for a product. Practioners discovered that the ideas of Agile are comparatively new and therefore the main four elements of Agile are collaboration, automation, activity and watching. Ambler (2011) talked about the necessity of additional controlled agile approach in an organisation by a severe concentrate on preparation and collaboration among groups. Several authors gave varied explanations for implementation of additional controlled agile and cooperative Agile in an organisation that ends up in a victorious enterprise execution.

Patwardhan et al. (2016) implemented agile approach in enterprise project management and attained prospering results in relations of quality and execution approach that can increase project's prospects for success. Hsieh and Chen (2015) explored that Continuous Integration is one among the main features of agile. Avison and Fitzgerald (1991), Nandhakumar and Avison (1999) shared the idea that traditional development methods are control-oriented and hypothetical in nature, and not suitable for the dynamic environment. They accompanied testing tool and attained outcomes with smart excellence and quicker turn around. Bruneo et al. (2014) presented employment of cloud surroundings and mentioned varied problems and challenges sweet-faced. It was found that the system gives additional excellence in less time. Dyck et al. (2015) stated that Software testing attributes are not sufficiently explored in the past and literature clearly lacks a uniform definition. (Fowler and Highsmith, 2001) stated that empirical studies into agile characteristics are needed so that practical tools can be applied to achieve agility. Srivastava et al. (2017, 2019) discussed a layout based on measuring and analysing the comulative errors in testing process of agile. They enlightened how this information, services and capabilities reinforced the Agile employment in an enterprise.

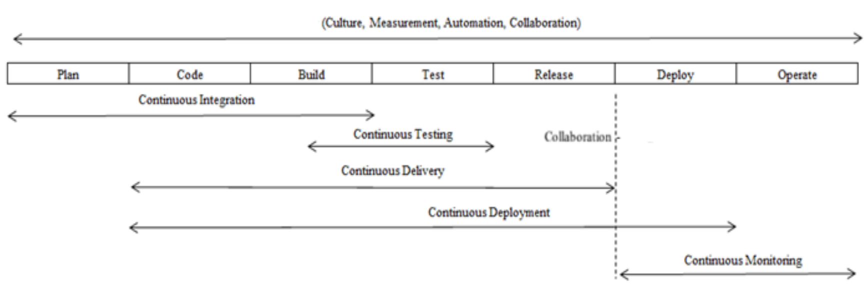
After doing this intensive research in agile testing literature we found that large enterprise applications have the significant role of development methodology. Ulrich and Eppinger (2008, pp.13–15) describe the software development process consists of the six stages i.e. concept development, planning, detail design, system-level design, testing and production phases. Some obtainable development approaches are Realistic test Estimates, Automated Test Cases (Schaefer et al., 2013), Agile Method Used (Scrum, Kanban, XP), Supporting tools (Erich et al., 2014), Clearly defined requirements, Test driven devilment, Adequate documentation, Test Environment Availability, Availability of Test Data, Supporting Tools, Feedback Mechanism, Customer Involvement (Patwardhan et al., 2016); (Lwakatare et al., 2015), Daily Startup Meetings (Hsieh and Chen, 2015), Agile Testing Mindset, Code Maintainability (Patwardhan et al., 2016; Somerville, 2001), Whole Team Tests (Whole Team Approach) and Realistic test Estimates (Mohamed et al., 2016).

2.2 Agile and its attributes that influence its implementation

Agile research has gained success due to the shift in business priorities that contributed escalation to the prerequisite of improved amount of release. The further most significant objective of Agile is to deliver additional regular releases that needs a controlled principles, operative partnership among participants, capacity and mechanisation that effects in quicker period to marketplace, condensed faults in the novel releases, and lesser lead-time among solutions. Agile testing can produce good quality software, achieve stakeholder satisfaction, motivate teams, assures quick and effective responses to stakeholders requests. Majorly, it allowed the software group to produce quick responses on the software's opportunity. Quick responses and readiness to modification curved out to be the key structures of the agile association. Uncertainty the software team is not assured in accepting whatever the user requests, it provides a leading estimate and then listens to response. After the literature, we dig out 16 significant testing attributes (As discussed above) of the faults that impact the defect prioritisation. We consulted 10 industry experts from the software industry, having more than twenty years of knowledge employed in creating, advancement and testing of enterprise applications. A case study is accompanied in an enterprise application defect prioritisation process.

The Agile and its core modules are demonstrated in the Figure 1. The core modules of Agile are Continuous Integration, Continuous Testing, Continuous Delivery, Continuous Deployment and Continuous Monitoring. These modules require prospect to embrace the tenure "Continuous Everything" in the subject of Agile (Akela, 2016).

Figure 1 Agile and its components



The basic benefits offered through Agile testing are quicker dispatch or shorter period to promote, value upgrading, higher client satisfaction, extreme lyco-operative plus inspired groups, enhanced efficiency and improved efficiency. Presently there are numerous attributes that impact the Agile employment in an organisation. Table 1 demonstrates 16 Agile attributes, that stand recognise dafter the review of different papers and deliberated in this analysis for the exhaustive validation.

Table 1 Details of Agile attributes

Attribute name	*Attribute description*	*Reference*
Test driven development	In Test Driven Development, the test is written first that is explained in a short way. The requirements and specifications are focused upon. Then run the test to check if there is any fault in the code. A new code is written and again it is checked until it passes with the requirements needed and then code refactoring is done	Patwardhan et al. (2016), Somerville (2001)
Continuous integration	The Continuous Integration (CI) is created for agile development where the development is arranged according to the functional description of requirements. It integrates the work of the team members and then checks for errors.	Schaefer et al. (2013), Stillwell and Coutinho (2015), Perry (2006)
Test environment availability	The main focus of Test Environment Availability is to improve the quality or making defect free software by making it more constructive and well organised. The development cycle involves four methods that include defect prevention, defect removal, defect tolerance and defect forecasting.	Patwardhan et al. (2016), Artač et al. (2016)
Feedback mechanism	Through feedback loops, the productivity increases as it helps one to improve along with providing feedback. According to the feedback provided, the loops re-examine the path in order to achieve the desired objective. These loops take place at every step all over the release and are iterative in nature.	Erich et al. (2014)
Adequate documentation	Documentation plays key role during software development. It focuses on WHAT and WHEN to document. The documentation effort must be baked into agile testing process. Everyone contributes to the documentation efforts, for each sprint. There should be a “Point person” who manages the documentation effort.	Mullaguru (2015), Saaty (2008)
Supporting tools-	The tools used in agile methodology supports the project in various ways. Managing Agile Testing can be Tough, So niche tools for agile projects can be super helpful to interface with the project, ensuring everyone is on the same page with regards to requirements, prioritisation, budget, and timeline as well as reporting.	Appleton et al. (1998), Shihab et al. (2012)
Availability of test data	Test data helps in creating larger data. The main purpose is not only to provide the data but also to provide it so that it can be used again and again. This requires a storage area of test data where various types of tests can take place that makes testers to search more about the data and information is available and the testers can use that data for testing. Hence the test environments are also restored.	Hsieh and Chen (2015)

Table 1 Details of Agile attributes (continued)

Attribute name	*Attribute description*	*Reference*
Daily stand-up meetings	Daily Stand-up Meetings gives updates about what an individual did, his/her plans for the day and if they are facing any problem in order to achieve the desired aim. Working day usually is chosen for these kinds of meetings. Every team member has to attend the meeting. It is ideally not last more than 15 minutes.	Ambler (2011)
Clearly defined requirement	Agile Testing requirement should be point to point and it has to be defined less strictly. Some processes like feedback mechanism and definition should be incremental in nature for this agile testing requires potentially shippable increments and there should be demo ready after one to three sprint cycles.	Mohamed (2016)
Automated test cases (unit, functional, regression)	Regression testing remains a black box testing method achieved via performing components of code frequently to guarantee that the continuing code alterations do not influence the systems functionality.	Artač et al. (2016)
Agile method used-repeated (Scrum, Kanban, XP)	Agile Methods break errands into little augmentations with insignificant arranging called Iterations. Each iteration includes a team employed over a complete software design enhancement cycle, comprising positioning, prerequisites examination, plan, writing codes, component testing, and responsive testing. It limits general hazard therefore enables the venture to adjust to variations swiftly. The majority of agile usage utilises an appropriate day by day eye to eye correspondence among colleagues.	Lwakatare et al. (2015), Sacks (2012)
Customer involvement	Continuous integration, Testing, Placement and Detecting is the precise component of the Agile methodology that needs response in all the component of agile. The outcome of every achievement that is similar to the good appraisal testing and placement is usually common with the suitable investor in authentic timeline which usually facilities the earlier problem determination by improve fitness. The comeback from each achievement similarly Code Appraisals, Testing, placement, issue is mutual with the suitable investors in authentic time that facilities the earlier problem determination by improved fineness.	Patwardhan et al. (2016), Lwakatare et al. (2015)
Defect management process	To identify the defects is a major objective in Agile testing process. It has to be reported and corrected on time. Our primary goal is to identify and prevent the defects. The whole process is risk driven, these involves strategies and priorities.	Patwardhan et al. (2016), Hsieh and Chen (2015)

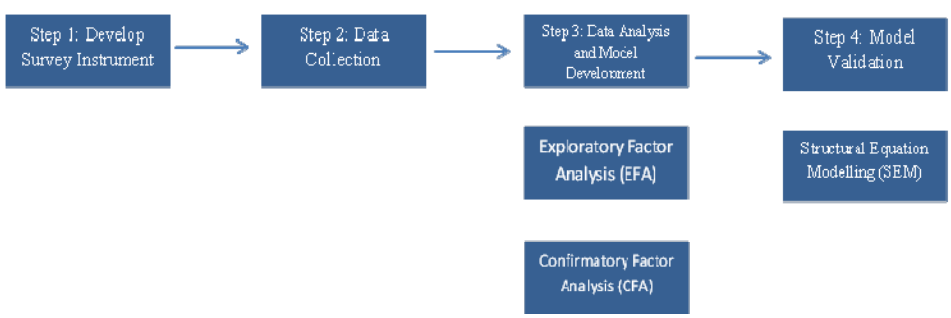
Table 1 Details of Agile attributes (continued)

Attribute name	*Attribute description*	*Reference*
Supporting tools	To support in applying the recommended method a supportive tool is designed. It focuses on handling the requirements deviations in dispersed agile improvement. The tool plays important role into deliberation the nature of distributed agile such as the dissimilar geographical settings of the expansion teams, communication among development team and information management. It is a web-based tool proposed to expedite and accomplish the necessities modelling and deviations in all phases. It starts apprehending requirements from the development commencement phase and keeps track of necessities during all next steps and iterations.	Larman and Basili (2003), Hsieh and Chen (2015)
Agile testing mindset (no policing but improvement)	Test primary and repeatedly signifies to the testing of the reduced modules of the code as rapidly as they fit out to be arranged to tested.	Mullaguru (2015)

3 Projected approach and experimental set up

This segment describes the projected agenda containing the statistical analysis tools specifically EFA, CFA besides SEM for accompanying the exhaustive validation of Agile individualities and validation for determining the Agile factors in an organisations. The projected procedure is implemented in a large organisation real-time project. Figure 2 illustrates the four-step procedure of the projected approach.

Figure 2 Stages for projected approach



3.1 Data collection

This segment describes the phases described in the exhaustive calculation of Agile factors by emerging and authenticating the structural equation model of the Agile Factors.

3.1.1 Agile survey

Sampling methods were given to 480 high-ranked professionals presently employed in Agile with experience more than twenty years in Agile applications were nominated from 10 organisations that has Capability Maturity Model (CMM) level 5. Table 2 establishes a sample of questions containing all 16 factors. We need to verify the authenticity of the questionnaire. For this purposes we have selected five-domain specialist from the industry to accompany our first experiment. All of our specialists have rich experience in Agile. The specialists we selected reviewed the study in sense of descriptive and objective questions and gave their response onto the same. These specialists were requested to see any uncertainty or deceptive info that could lead the study in biased condition. The valuable feedback taken from experts helped us in designing the questionnaire. We have sent these questionnaire to 480 respondents (those are experts of Agile) on Email, LinkedIn to fill the survey. To get the speedy response, we have sent the follow up emails that comprising of strategies to fill the survey.

Table 2 Sample questionnaire

Agile factors	*No impact*	*Low impact*	*Medium impact*	*High impact*	*Very high impact*
	1	*2*	*3*	*4*	*5*
Realistic test estimates	–	–	–	–	–
Agile method used (Scrum, Kanban, XP)	–	–	–	–	–
Clearly defined requirements	–	–	–	–	–
Test driven devilment	–	–	–	–	–
Automated test cases (Unit, Functional)	–	–	–	–	–
Continuous integration	–	–	–	–	–
Adequate documentation	–	–	–	–	–
Test environment availability	–	–	–	–	–
Defect management process	–	–	–	–	–
Availability of test data	–	–	–	–	–
Supporting tools	–	–	–	–	–
Feedback mechanism	–	–	–	–	–
Daily start-up meetings	–	–	–	–	–
Customer involvement	–	–	–	–	–
Whole team tests (Whole Team Approach)	–	–	–	–	–
Agile testing mindset (No Policing)	–	–	–	–	–

3.1.2 Data analysis and model development

The authenticity of replies was tested thoroughly, so that there are no missing or inappropriate responses. We got overall 400 responses among the 480 mails that we had sent and from that, we had chosen 378 replies. The validity of the survey analysed with the help of *Cronbach's Alpha* (Schmitt et al., 1996), depicts how objects are related to one another. It processes the inner reliability of these substances in a data set. The value of 0.7 or more of *Cronbach's Alpha* measured satisfactory as the factors are closely related and is much reliable.

The EFA analysis was done to determine the user's records, which are in detailed measure, and to find the correctness of these records is intended for Factor Analysis. The correctness can be checked with the Barlett's test of sphericity and Kaiser–Meyer–Oklin (KMO) amount of sampling (Williams et al., 2010). With the studies, it was observed that if the outcome of KMO is larger than 0.5 then it will be considered appropriate for Factor Investigation (Williams et al., 2010). The outcome of Barlett's test of sphericity taking less than p i.e. 0.01 has been considered perfect for attribute Analysis (Williams et al., 2010).

In the following section, we see there are 16 attributes that are carefully determining the required pattern amongst these factors. EFA was applied on selected factors to designate the latent attributes or the elementary factors. The latent attributes are the extensive classifications that constitute the agile factors. EFA encompasses three phases that are attribute extraction, variation and factor clustering. Factor clustering dispenses with analysis of the elementary attributes that adequately expresses the experimental correlation between the factors. The attributes acquired from factor extraction are altogether hard to disclose and thus converted in to a manageable form using the method of rotation in SPSS. SPSS supports two types of rotation that is varimax and oblique. Varimax is helpful in case of non-dependency of factors and oblique is helpful when we have relation in the attributes or they are dependent on each other. There are many methods obtainable for factor extraction. We have taken Principal Component Analysis (PCA) in this script, since our criteria was to reduce attributes that shares many similarities to exploratory factor analysis. We have explained EFA as per the below equation.

$$X_j = a_{j1} * F_1 + a_{j2} * F_2 + a_{j3} * F_3 + \ldots + a_{jm} * F_m + e_j \tag{2}$$

where p = Entire sum of experiential factors

$J = 1, 2, 3, \ldots, p$

m = Total sum of primary variables

F_i = Influenced/Latent attributes

a_{j1} = Factor loading meant for j-th experiential variable on the factor F_1

e_j = Autonomous p precise errors

Principal Component Analysis (PCA) (Hair et al., 2006) changes set of lines called principal components. It was oblique rotation since factors are not independent and are correlated. They are followed by Eigen values than screen plot. The overall elements that are lying beyond the section of *Scree Plot* and taking Eigen values more than 1 are restrained as the essential factors (Williams et al., 2010). The greater value of factors related to each other signifies that the factor is extremely connected (Thompson, 2004).

Table 3 Measures used in CFA to test the goodness of model fit

Measure	*Full name of measure*	*Acceptable values*	*Reference*
RMSEA	Root mean square error of approximation	Close to Zero	Hair et al. (2006)
RMR	Root mean square residual	<0.08	Hu and Bentler (1999)
GFI	Goodness of fit index	Close to 1	Dawes et al.(1998)
AGFI	Adjusted goodness of fit index	Close to 1	Mittal and Sangwan (2014)
CFI	Comparative fit index	>0.9	Byrne (2001)

RMSEA (Hoe, 2008) is widely used to analyse the inconsistence among the hypothesised standard. RMSEA are chosen between 0 and 1 (Hair et al., 2006). Values less than 0 implies the improved model-fitting comparative to the value which are closer than or higher than 1. For acceptance of the model the RMR value of 0.6 or less is acceptable. RMR is predominantly the square root of inconsistent sample of covariance matrix and model covariance matrix. RMR value less than 0.08 (Hu and Bentler, 1999) is said to be adequate .Goodness of Fit Index (GFI), which signifies the hypotheses model and pragmatic covariance matrix. Many indicators of latent variable affect AGFI. For better results, we should keep the value of AGFI and GFI between 0 and 1. In the literature, we observed that the value AGFI and GFI over 0.9 is acceptable for the model. The evolution of discrepancy among the data and the hypothesised model is done by relative fit index.

4 Measures and outcomes

The result from the proposed methodology explained in the previous part, used to achieve the outcomes in this section. The primary part of this segment shares the outcomes of statistical analysis and establishes the model of Agile attribute.

4.1 EFA (exploratory factor analysis)

Here we obtained the results by using statistical analysis tools for detailed assessment of agile attribute. Below Table 4 demonstrates the outcomes from EFA. The standards of *Cronbach Alpha* remained greater than 0.7 for whole factors that stated dependability of data (Schmitt, 1996). The high value of *Cronbach's Alpha* implies higher consistency and closer relationship between the attributes in the groups. The outcome of *Kaiser Meyer Oklin (KMO*) remained 0.78 and the outcome of *Barlett's test of sphericity* specified value of p less than 0.000, which depicts that the data was appropriate for the factor investigation (Williams et al., 2010).

Table 4 Explanatory factor analysis results

Agile factors	*Mean*	*Std. deviation*	*Cronbach's alpha*	*Factor loadings*				*Regression weights*		
				Planning	*Execution*	*Tools and infrastructure*	*People*	*Estimate*	*SE*	*CR*
1) Realistic test estimates	4.13	0.61	0.86	0.79	–	–	–	0.74	0.06	17.27
2) Agile method used	4.14	0.66	0.84	0.84	–	–	–	0.79	0.05	21.80
3) Clearly defined requirements	4.10	0.63	0.83	0.92	–	–	–	0.72	0.04	21.54
4) Test driven development	4.10	0.68	0.82	0.83	–	–	–	0.60	0.03	15.06
5) Automated test cases	4.14	0.70	0.89	0.76	–	–	–	1.05	–	–
6) Continuous integration	3.42	0.68	0.82	–	0.95	–	–	0.99	0.16	6.62
7) Adequate documentation	3.35	0.65	0.81	–	0.87	–	–	1.59	0.29	5.55
8) Test environment availability	3.37	0.64	0.86	–	0.85	–	–	1.00	–	–
9) Defect management process	3.40	0.68	0.89	–	0.84	–	–	1.08	0.12	10.66
10) Availability of test data	3.66	0.68	0.81	–	0.71	–	–	1.02	0.12	11.21
11) Supporting tool	3.09	0.69	0.88	–	–	0.83	–	1.20	0.08	15.40
12) Feedback mechanism	3.01	0.55	0.87	–	–	0.87	–	1.67	0.07	13.95
13) Daily start-up meeting	2.78	0.56	0.80	–	–	0.77	–	1.00	–	–
14) Customer involvement	3.29	0.62	0.80	–	–	–	0..83	1.07	0.15	8.21
15) Whole team test	3.31	0.66	0.89	–	–	–	0.75	0.87	0.13	6.99
16) Agile testing mind-set	3.50	0.54	0.81	–	–	–	0.90	1.65	0.19	9.57

Preliminary Eigen principles identify the primary total variance, extraction sums of squared loadings. It was noticed that four of the latent variables had a value more than 1 for Eigen standards and their total percentage value was 69.29%. This signifies that for four latent variables there was a natural variance of 69.29% by 16 attributes. Primary total variance is identified by Preliminary Eigen principles, which includes extraction sums and rotation sum of squared loadings. Catell's Scree plot is another technique to extract the primary latent variables. A graphical representation of Eigen values contradicting each other is called Scree plot. The Scree Plot completes presence of four latent variables. Four latent variables categorised as exhaustive analysis are Planning, Execution, Tool and People.

Table 5 Exploratory factor analysis Eigen values

	Initial Eigen values			*Agile factor*			*Rotation sums of squared loadings*		
	Total	*% of Variance*	*Cumulative %*	*Total*	*% of Variance*	*Cumulative %*	*Total*	*% of Variance*	*Cumulative %*
1	4.348	27.175	27.175	4.348	27.175	27.175	3.698	23.109	23.109
2	3.365	21.034	48.209	3.365	21.034	48.209	3.515	21.967	45.076
3	2.391	14.945	63.154	2.391	14.945	63.154	2.186	13.661	58.737
4	1.410	8.811	71.965	1.410	8.811	71.965	2.116	13.228	71.965
5	0.688	4.301	76.266	–	–	–	–	–	–
6	0.635	3.969	80.235	–	–	–	–	–	–
7	0.547	3.421	83.656	–	–	–	–	–	–
8	0.467	2.919	86.565	–	–	–	–	–	–
9	0.414	2.585	89.160	–	–	–	–	–	–
10	0.366	2.288	91.448	–	–	–	–	–	–
11	0.334	2.087	93.535	–	–	–	–	–	–
12	0.284	1.776	95.311	–	–	–	–	–	–
13	0.278	1.735	97.045	–	–	–	–	–	–
14	0.215	1.343	98.388	–	–	–	–	–	–
15	0.167	1.043	99.431	–	–	–	–	–	–
16	0.091	0.569	100.00	–	–	–	–	–	–

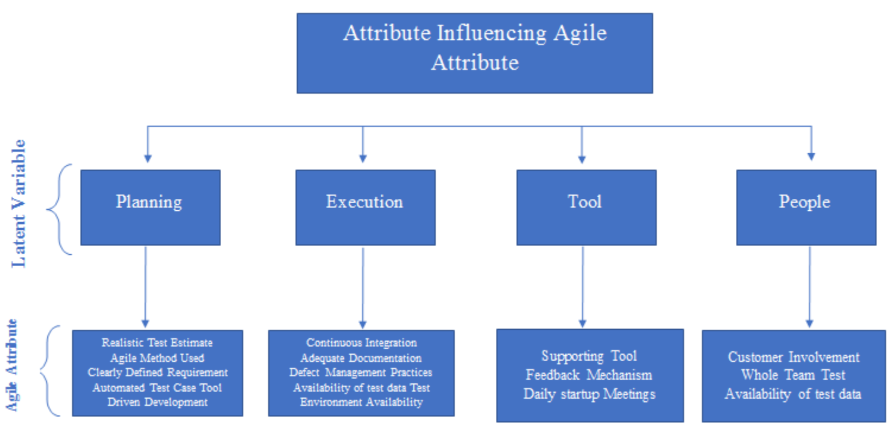
The result of the CFA is demonstrated in the Table 6. CFA validates the attribute structure with the hypothesis that the association among the factors and the fundamental attributes is effective. Numerous statistical procedures that examine the goodness of fit, used to confirm this. The assessment of RMSEA observed as 0.067 that indicated good model fit and acceptance of the model fitting. The assessment of RMR was 0.03, which also showed the satisfactoriness of model. The value of GFI, AGFI and CFI were 0.88, 0.845 and 0.950, respectively. The values for all the statistical measures were in the satisfactory range, which indicated the acceptability of model, and confirms the result provided by the Exploratory Factor Analysis.

Table 6 Confirmatory factor analysis measures

Measures/Indexes	*Estimated values*	*Recommended values*
Chi Square ()	439.361	–
Degree-of-freedom (DoF)	137	–
P-value	0.000	0.0
Chi-Square/DoF	3.207	<5
RMSEA	0.067	Close to Zero
RMR	0.03	<0.08
GFI	0.883	Close to 1
AGFI	0.845	Close to 1
CFI	0.950	>0.9

Figure 3 shows the classification of all Agile traits into four latent variables. Planning variable represents the set of factors associated with it. These attributes are realistic test estimates, agile method used, clearly defined requirement, automated test cases and driven development. Execution variable represents the set of factors, which benefits in executing several processes and techniques. The factors classified under execution are continuous integration, adequate documentation, and defect management process, availability of test data to monitor and test environment availability. Tool variable focuses on the factors i.e. Supporting tool, feedback mechanism and daily start-up meetings where people includes customer involvement, whole team test and availability of test data.

Figure 3 Classification of Agile attributes – exploratory factor analysis



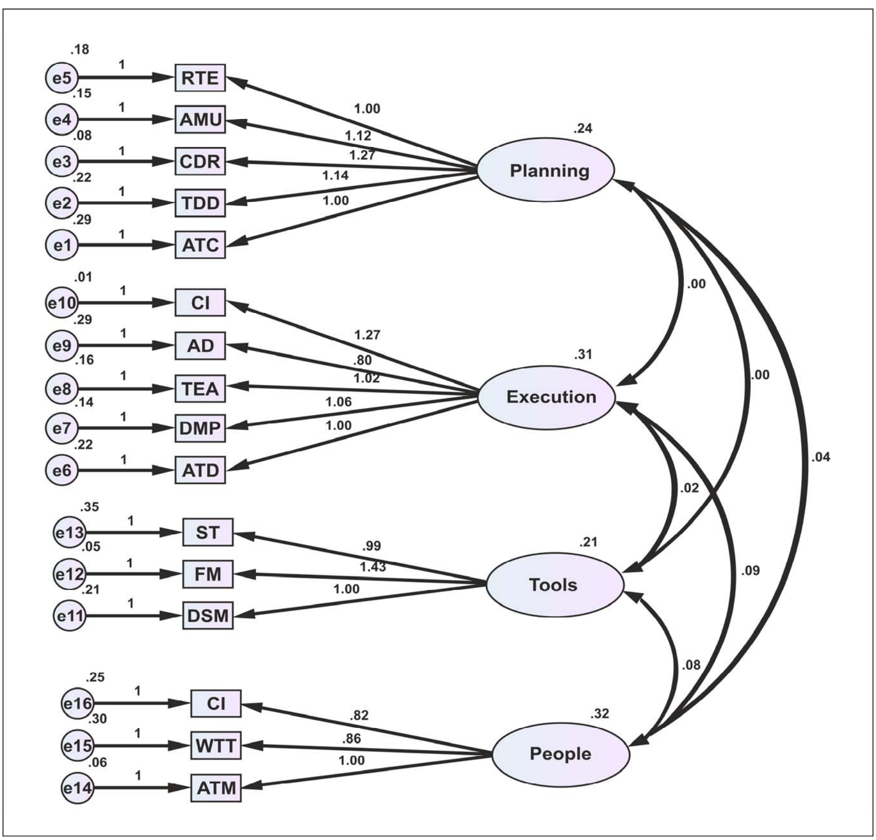
4.2 *Confirmatory factor analysis and structural equation modelling*

In Table 6, we have demonstrated the results of CFA. We can determine the validation of the relationship between the attributes and the factors by CFA, using factor structure and hypothesis test. The signification of statistical values was in the acceptable range, which indicated the acceptability of the model, and approves conclusion given by the EFA.

As a measure of SEM, it was mandatory to develop the hypothesis demonstrating the association among the latent variables. The 5 industry professionals assisted us in outlining these hypotheses and they authorised that the Agile factors are interdependent on every other latent variable like Tools Influences Execution, Tools Influences People, People influences Execution and Planning influences Execution.

We use statistical tool namely AMOS to perform Confirmatory Factor Analysis and hence the results are shown in Figure 4. From the above four Hypotheses were observed firstly hypothesis (H1) that describes as, Tools Influences execution, secondly hypothesis(H2) that states Tools influenced by people, thirdly hypothesis (H3) which states that people influences execution and fourth one is hypothesis (H4) that describes the planning influenced on execution. We have taken three hypothesis namely H1, H3 and H4 that has been accepted as our p-value was 0.7 and *p*-value assigned to H2 was less than 0.5. Hypothesis namely H2 were rejected. Hypothesis H1, H3 and H4 were accepted whereas Planning, execution and tool are the main factors, which will be moderated in a priority order for achieving the high-quality agile implementation. Evaluation of maturity of Agile implementation in an enterprise.

Figure 4 Confirmatory factor analysis



After analysing the outcomes it was found that the reliability of some agile factors was at lower level. The factors which were involved in performing out below threshold level were Customer Involvement, Whole team test, Agile Testing Mind-set.

5 Discussions

Agile Methodology and testing methods have benefitted many organisations with technical excellence and specific time deliverables. Four of the basic factors that derive all of the 16 factors were finalised by Confirmatory Factor Analysis. We have categorised Planning, Execution, tool and people as the underlying factor. It has been analysed by SEM that Planning and Execution are the driving factors that determine Tools. We have also demonstrated that the factors that were represented by driving factors were the key independent factors and to get the speedy delivery with improved quality, the key independent factors has to be mitigated at the high priority in Agile Methodology

Realistic Test Estimates, Agile Method Used (Scrum, Kanban, XP), Clearly defined requirements, Test driven development, Automated Test Cases (Unit, Functional, Regression), Continuous Integration, Adequate documentation, Test Environment Availability, Defect management process, Availability of test data, Supporting tool, Feedback mechanism, Daily Start-up meeting, are the Thirteen key independent factors. We have developed the methodology that only thirteen key independent were need to be measured for the faster turnaround , which was the major goal of Agile.

This result demonstrates to be favourable for agile practitioners of a given methodology. They were inefficient to measure the maturity of agile implementation. The outcome was received and the least performing attributes were observed by the management team of reputed organisation (for security reasons, we do not disclose the name) in particular. For the least performing attributes, some amendments were suggested. There is an increase in advancement of the maturity of agile implementation. The Agile industries were keenly interested in the scores of evaluation to implement rapid turnaround results that were received in Agile implementation.

6 Conclusion

The reference of the Agile factors is provided by few articles in the literature. However, to conclude the reliability level of the Agile implementation, the detailed assessment and measurement of these factors in an enterprise remains speculative. By offering, a structure that empowers experts to deploy the exhaustive validation of the Agile factors this manuscript gives expressively in the era of Agile using SEM and extent of these factors to regulate the maturity of Agile implementation. A structural model for these factors was established after examining the relationship among 16 Agile factors in as using SEM and CFA. These 16 factors were grouped under four latent variables namely Planning, Execution, Tool and People. To authenticate the structural model, Structural Equation Modelling was used. It was determined that the factors categorised under Planning, Execution and Tool are categorised under the key self-regulating factors and they need to be alleviated at precedence to achieve faster turnaround, higher quality and higher maturity level of Agile implementation. The calculation discovered that the reliability of Agile execution was under threshold level for three low performing factors under latent variable people, which required immediate management attention. Corrective actions were taken against these three low performing factors by the management team (of the company). An important enhancement was established by the re-evaluation of performance of the Agile factors in the reliability level of Agile implementation, which was factor due to the acceptance of the projected approach.

References

Akela, A. (2016) *Assembling the key components of continuous delivery*. Available online at: https://devops.com/assembling-components-continuous-delivery/

Ambler, S.W. (2008) 'Tailoring usability into agile software development projects', *Maturing Usability*, Springer, London, pp.75–95.

Ambler, S.W. (2011) 'Disciplined agile delivery and collaborative DevOps', *Cutter IT Journal*, Vol. 24, No. 12.

Appleton, B., Berczuk, S., Cabrera, R. and Orenstein, R. (1998) 'Streamed lines: branching patterns for parallel software development', *Proceedings of PloP*, Vol. 98, p.14.

Artač, M., Borovšak, T., Di Nitto, E., Guerriero, M. and Tamburri, D.A. (2016) 'Model-driven continuous deployment for quality DevOps', *Proceedings of the 2nd International Workshop on Quality-Aware DevOps*, ACM, New York, USA, pp.40–41.

Avison, D.E. and Fitzgerald, G. (1991) *Information Systems Development*, Blackwell Scientific Publications, Oxford.

Bruneo, D., Fritz, T., Keidar-Barner, S., Leitner, P., Longo, F., Marquezan, C. et al. (2014) 'CloudWave: where adaptive cloud management meets DevOps', *2014 IEEE Symposium on Computers and Communications (ISCC)*, pp.1–6.

Byrne, B.M. (2001) *Structural equation modeling with AMOS: Basic concepts. Applications, and Programming*, Mahwah, New Jersey.

Cho, Y.H., McCullough, B.F. and Weissmann, J. (1996) 'Considerations on finite-element method application in pavement structural analysis', *Transportation Research Record*, Vol. 1539, No. 1, pp.96–101.

Cockburn, A. (2002) *Agile Software Development*, Addison-Wesley, Boston.

Dawes, J., Faulkner, M. and Sharp, B. (1998) 'Business orientation scales: development and psychometric assessment', *27th EMAC Conference*, Vol. 5, pp.461–478.

Dyck, A., Penners, R. and Lichter, H. (2015) 'Towards definitions for release engineering and devOps', *Proceedings of the 3rd International Workshop on Release Engineering*, IEEE Press, New Jersey, USA, pp.3–3.

Erich, F., Amrit, C. and Daneva, M. (2014) 'A mapping study on cooperation between information system development and operations', *International Conference on Product-Focused Software Process Improvement*, Springer International Publishing, pp.277–280.

Fowler, M. and Highsmith, J. (2001) 'The agile manifesto', *Software Development*, Vol. 9, No. 8, Miller Freeman, Inc., San Francisco, CA, pp.28–35.

Hair, J.F., Black, W., Babin, B. and Anderson, R.E (2006) *Multivariate Data Analysis*, Pearson Education, Upper Saddle River, NJ.

Hoe, S.L. (2008) 'Issues and procedures in adopting structural equation modelling technique', *Journal of Applied Quantitative Methods*, Vol. 3, No. 1, pp.76–83.

Hsieh, C.Y. and Chen, C.T. (2015) 'Patterns for continuous integration builds in cross-platform Agile software development, *Journal of Information Science and Engineering*, Vol. 31, No. 3, pp.897–924.

Hu, L. and Bentler, P.M. (1999) 'Cut-off criteria for fit indexes in covariance structure analysis: conventional criteria versus new alternatives', *Structural Equation Modeling*, Vol. 6, No. 1, pp.1–55.

Kumar, V., Kapur, P.K., Taneja, N. and Sahni, R. (2017) 'On allocation of resources during testing phase incorporating flexible software reliability growth model with testing effort under dynamic environment', *International Journal of Operational Research*, Vol. 30, No. 4, pp.523–539.

Kumar, V., Mathur, P., Sahni, R. and Anand, M. (2016a) 'Two-dimensional multi-release software reliability modeling for fault detection and fault correction processes', *International Journal of Reliability, Quality and Safety Engineering*, Vol. 23, No. 3. Doi: 10.1142/S0218539316400027.

Kumar, V., Sahni, R. and Shrivastava, A.K. (2016b) 'Two-dimensional multi-release software modelling with testing effort, time and two types of imperfect debugging', *International Journal of Reliability and Safety*, Vol. 10, No. 4, pp.368–388.

Larman, C. and Basili, V. (2003) 'Iterative and incremental development: a brief history', *IEEE Computer*, pp.47–56. doi: 10.1109/MC.2003.1204375.

Lwakatare, L.E., Kuvaja, P. and Oivo, M. (2015) 'Dimensions of DevOps', *International Conference on Agile Software Development*, Springer International Publishing, Helsinki, Finland, pp.212–217.

Mittal, V.K. and Sangwan, K.S. (2014) 'Development of a structural model of environmentally conscious manufacturing drivers', *Journal of Manufacturing Technology Management*, Vol. 25, No. 8, pp.1195–1208.

Mohamed, S.I. (2016) 'DevOps maturity calculator DOMC – value oriented approach', *International Journal of Engineering Research and Science*, Vol. 2, No. 2, pp.25–35.

Mullaguru, S.N. (2015) 'Changing scenario of testing paradigms using DevOps – a comparative study with classical models', *Global Journal of Computer Science and Technology*, Vol. 15, No. 2.

Nandhakumar, J. and Avison, D.E. (1999) 'The fiction of methodological development: a field study of information systems development', *Information Technology and People*, Vol. 12, No. 2, pp.176–191.

Patwardhan, A., Kidd, J., Urena, T. and Rajgopalan, A. (2016) 'Embracing Agile methodology during DevOps developer internship program', arXiv preprint arXiv:1607.01893.

Perry, W.E. (2006) *Effective Methods for Software Testing*, 3rd ed., Wiley, Indianapolis, IN.

Saaty, T.L. (2008) 'Decision making with the Analytic hierarchy process', *International Journal Services Sciences*, Vol. 1, No. 1, pp.83–98.

Sacks, M. (2012) 'Devops principles for successful web sites', *Pro Website Development and Operations*, A Press, pp.1–14.

Schaefer, A., Reichenbach, M. and Fey, D. (2013) 'Continuous integration and automation for DevOps', *IAENG Transactions on Engineering Technologies*, Springer Netherlands, pp.345–358.

Schmitt, N. (1996) 'Uses and abuses of coefficient alpha', *Psychological Assessment*, Vol. 8, No. 4, pp.350–353.

Shihab, E., Bird, C. and Zimmermann, T. (2012) 'The effect of branching strategies on software quality', *ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, IEEE, pp.301–310.

Somerville, I. (2001) *Software Engineering*, 6th ed., Addison-Wesley, Boston, MA.

Srivastava, A., Kapur, P.K and Mehrotra, D. (2017) 'Modelling fault detection with change-point in agile software development environment', *Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*, pp.303–308, Doi: 10.1109/ICTUS.2017.8286023.

Srivastava, A., Kapur, P.K., Mehrotra, D. and Majumdar, R. (2019) 'Modelling fault detection using SRGM in Agile environment and ranking of models', *Journal of Cases on Information Technology*, Vol. 21, No. 2, pp.1–20.

Stillwell, M. and Coutinho, J.G. (2015) 'A DevOps approach to integration of software components in an EU research project', *Proceedings of the 1st International Workshop on Quality-Aware DevOps*, pp.1–6.

Thompson, B. (2004) *Exploratory and Confirmatory Factor Analysis: Understanding concepts and applications*, American Psychological Association.

Ullman, J.B. and Bentler, P.M. (2003) *Structural Equation Modelling*, John Wiley & Sons, Inc.

Williams, B., Onsman, A. and Brown, T. (2010) 'Exploratory factor analysis: a five-step guide for novices', *Journal of Emergency Primary Health Care*, Vol. 8, No. 3, pp.1–13.

Wynstra, F. and Ten Pierick, E. (2000) 'Managing supplier involvement in new product development: a portfolio approach', *European Journal of Purchasing & Supply Management*, Vol. 6, No. 1, pp.49–57.