Scrum Adoption Challenges Detection Model: SACDM

Ridewaan Hanslo
University of South Africa, School of Computing,
College of Science, Engineering and Technology
Pretoria, South Africa
Email: ridewaan@gmail.com

Professor Ernest Mnkandla
University of South Africa, School of Computing,
College of Science, Engineering and Technology
Pretoria, South Africa
Email: mnkane@unisa.ac.za

Abstract—Scrum has been the most widely adopted Agile methodology over the past decade with Scrum and Scrum variants offering alternatives to the old software development methods. While Scrum plays an important role in the success of Agile development, it does come with its own challenges. In previous research challenges have been analyzed at the organizational and team level, primarily via case studies. However, fundamentally, Scrum needs to be adopted at the individual level. Furthermore, challenges such as inexperience, poor communication, specialization, lack of teamwork, low-quality, organizational culture and Scrum compatibility, have been identified as contributors. This paper therefore discusses the Scrum and Agile adoption challenges faced both globally as well as within the South African borders, from the findings of a narrative review. Secondly, a custom model adapted from the Diffusion of Innovation theoretical model was developed to detect the Scrum adoption challenges experienced within software organizations at the individual level. The custom model referred to as the Scrum Adoption Challenges Detection Model (SACDM) consists of four constructs, namely; individual factors, team factors, organizational factors and technology factors. The constructs are composed of nineteen independent variables that assists in understanding which factors contributes towards an individual either adopting or rejecting Scrum within a software organization. SACDM is therefore used to detect the adoption or rejection of Scrum as the dependent variable based on the independent variables being tested within the four constructs. The model can further be used with a survey questionnaire to provide generalized awareness of Scrum adoption challenges allowing software organizations to make more informed decisions when adopting Scrum. Future research is to allow the model to contribute towards Scrum adoption challenges predictive analysis.

Index Terms—Adoption Challenges, Agile Methodologies, Scrum, Software Engineering, Software Organization.

I. INTRODUCTION

"SOFTWARE development has become one of the world's most important practices. The software we produce today is rapidly becoming the embodiment of much of the world's intellectual property. Simply put, our modern world depends on software" [1].

Scrum, in the context of this paper, refers to an Agile methodology with emphasis on project management structure and communication between all stakeholders including clients and business representatives, regularly setting sprint time limits for software completion, reviewing changes and applying retrospection before working on the next product backlog requirements [2]. The software organization we refer to in this paper is any company, firm or organization that has a software development division or group of two or more individuals responsible for developing and maintaining software, for the benefit of the software organization or the client they service.

Scrum was developed in the early 1990's by Ken Schwaber and Jeff Sutherland [3]. Scrum is currently the most widely adopted Agile methodology, based on the 2017 VersionOne survey [4]. The reason for its high adoption rates could be its simplicity, as it can be easily understood by both business and Scrum teams alike.

Agile adoption (which includes Scrum) has its challenges such as work specialization, organizational culture, resistance to change, and lack of communication to name a few, however, what is certain is the fact that successful adoption improves numerous aspects of the business operation. Business operations include project visibility, manage change priorities, better aligned Information Technology (IT), increased productivity and enhanced software quality [4]. As stated by Rogers [5], "A technological innovation usually has at least some degree of benefit for its potential adopters".

What is evident from the reviewed literature is that, whilst there are common problems and challenges identified, there are very few empirical studies on the Scrum adoption challenges experienced by individuals. Most research focused on qualitative methods with emphasis on case studies [6]. A descriptive and explanatory case study done by Noruwana and Tanner [7] on Agile processes with emphasis on Scrum alludes that there is a knowledge base to unearth on adoption challenges.

To get an overview of adoption challenges faced by individuals within software organizations, it is necessary to get to the core of the challenges, i.e. what are the Scrum adoption challenges experienced in practice? Is there a relationship to be discovered with the adoption challenges and Scrum adoption outcomes? Will the knowledge and understanding point to a potential correlational or causal outcome?

Multiple theories, models and frameworks such as Diffusion of Innovation (DOI), Technology Acceptance Model (TAM), Perceived Characteristics of Innovations, and Theory of Planned Action, have been used to better understand the adoption and implementation of methodologies in software development [8].

We looked at the Scrum adoption challenges experienced through the lens of the DOI theoretical model. The DOI theoretical model was chosen instead of alternate theories because it was the only theory at the time to have been used both at the individual level and organizational level of IT adoption research, which meets the author's requirement. The custom model was divided into four constructs identified as individual factors, team factors, organization factors, and technology factors. The four constructs combine to form a holistic representation of the individual's belief, its relation to people, how they perceive management, and their perception of the methodology being used [6].

The aim of this paper is to propose a model which can be used to detect Scrum adoption challenges experienced by individuals within software organizations. The constructs of the custom model are an adaptation of the DOI theoretical model and the conceptual framework of the object-orientated technology (OT) study by Sultan and Chan [31]. The independent variables within the model's constructs are generated using the narrative review method. The proposed model will be used to differentiate adopters from non-adopters of the Scrum methodology, respectively.

Section II provides a brief background on Scrum as an Agile methodology, followed by the global and South African (SA) Scrum adoption challenges compiled from extant literature. Section III lists the constructs with a discussion of the variables with its hypothesized relationship to Scrum adoption and section IV explains the composition of the proposed SACDM. Section V concludes the paper.

II. SCRUM ADOPTION CHALLENGES

The Scrum Guide written by Schwaber and Sutherland [1] states the following about Scrum: "A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. Scrum is:
- Lightweight
- Simple to understand
- Extremely difficult to master"

Scrum is a value-driven method (as opposed to a plan-driven method such as the waterfall method) which is iterative and incremental development [9]. The Scrum value-driven method continuously reassesses the problem while making small software feature increments in short time blocks within small teams [10]. Scrum is so flexible and abstract in its definition and implementation that it is often used outside of the Software Engineering (SE) practice [2].

Adoption challenges, in the context of this paper, refers to the challenges faced by software organizations when choosing and following an Agile methodology [11]. As mentioned in the introduction we used the narrative review method to generate the custom model's independent variables. Before we were able to generate the independent variables, we first had to identify the adoption challenges faced within the global and SA context. The Scrum and Agile adoption challenges were acquired through the narrative review method, which was as follows;
- Data sources was relatively recent i.e. all except one paper was less than ten years of age.
- It has been cited in other literature.
- If not cited, the source must have been published by an accredited publisher, e.g. Springer, Pearson, Institute of Electrical and Electronics Engineers (IEEE), International Journal of Environmental Science and Technology (IJEST), etc.
- If not published by an accredited publisher, the source must have been presented at a known institution, e.g. Agile Africa conference, Johannesburg Centre for Software Engineering (JCSE).
- Alternatively, the source is a recent dissertation or thesis paper.
- The primary search terms were 'Scrum', 'Scrum adoption', 'Scrum challenges', 'Agile challenges', 'Scrum South Africa', 'Agile South Africa' and 'Agile adoption'.
- The sources where carefully perused and relevant literature was ear marked for further investigation.
- These pre-selected literature sources were filtered based on the content it provided, i.e. Do the literature sources contain challenges and issues experienced during Scrum and Agile adoption? Or is the literature describing adoption challenges on irrelevant Software Development Methodologies (SDM)?
- Identified challenges within the literature was collated and the frequency of occurrence was recorded.

Table I is a consolidated list of global Scrum and Agile adoption challenges taken from twenty-one literature studies Stray et al. [12], [32], Asnawi et al. [13], Santos et al. [14], Fægri [15], Marchenko and Abrahamsson [16], Overhage et al. [17], Heikkila et al. [18], Kapitsaki and Christou [38],

Bjarnason and Regnell [39], Irrazabal et al. [40], Hoda et al. [41], [42], Dorairaj et al. [43], Senapathi et al. [44], Ressin et al. [45], [46], Santos and Goldman [47], Kim and Ryoo [48], Ihme [49], and Allisy-Roberts et al. [4], with publication years ranging from 2008 to 2017. Of the listed challenges, the one that is very peculiar comes from the mixed mode study by Heikkila et al. [18], which recorded that cross functional generalist teams were not plausible in the environment. This is contradictory to the Scrum philosophy of well-balanced redundant knowledge teams with the ability to work on various aspects of projects without the dependency of team member specialization.

TABLE I.
GLOBAL SCRUM AND AGILE ADOPTION CHALLENGES

No.	Global Scrum and Agile Adoption Challenges	Frequency
1	Lack of knowledge/training/skills	11
2	Organizational culture/mindset	9
3	Teamwork/communication issues	9
4	Lack of documentation	5
5	Budget and schedule constraint	2
6	Escalating commitment	2
7	Hard to scale	2
8	High management overhead	2
9	Lack of senior support	2
10	Work specialization	2
11	Cross functional generalist teams	1
12	Increase stress and workload	1
13	Lack of quality	1
14	Lack of top management support	1
15	Long time to market	1
16	Low user satisfaction	1
17	Over engineered solutions	1
18	Over optimistic task estimates	1
19	Project team size	1
20	Requirements creep	1
21	Retrospective inadequacy	1
22	Too many meetings	1

Top Management Support (TMS) has been found to significantly affect the user's perception of an IT technology, and the organizations IT adoption and diffusion, respectively [19], [20]. Therefore, the inclusion of lack of TMS is probably expected, considering the impact management support have on IT adoption [21]. It should be noted that although TMS is important for the adoption and diffusion of a methodology, it cannot save a project that is failing, and too much support might hinder the adoption and diffusion success [19].

Table II is a consolidated list of SA Scrum and Agile adoption challenges taken from six literature studies Mnkandla and Dwolatzky [22], Du Toit [23], Tanner and Khalane [11], Tanner and Mackinnon [24], Tanner and Wallace [31], and Noruwana and Tanner [7], with publication years ranging from 2004 to 2013. The rational for displaying the global and SA adoption challenges separately was to allow the authors to compare the two tables and identify if region had a role in challenges experienced. Most of the challenges within the SA literature was experienced globally, however, within the SA literature, organizational challenges were prevalent within the top challenges (by frequency), and while globally team challenges were more predominant.

It should be made clear that when referring to communication problems, it includes clients, and not just the individuals within the software organization. Especially in Scrum, clients are expected to be more collaborative, knowledgeable and representative, and committed towards the projects [26]. The importance of customer's active involvement in the development process is crucial to the success of Agile development [52]. The greater the involvement of customers during the development process the greater the chance of success [6].

Mohan and Ahlemann [27] says that the use of the Information Systems Development (ISD) process is determined by the rational and hierarchy of the organizational culture. Often the needs, beliefs and values of the users of the methodology are not considered, which is like the subjective norm situation, whereby the developer's views are not always the determinant to the Agile methodology adoption decision. As Hardgrave et al. [28] puts it; "Developer's intentions are directly influenced by their perceptions of usefulness, social pressure, compatibility and organizational mandate". Chan and Thong [26] indicates that prior SDM studies focused on the developer views of the SDM such as perceived ease of use and perceived usefulness, however, failing to realize the importance of management (e.g. management style) and people-related (e.g. competency levels) challenges.

TABLE II.
SA SCRUM AND AGILE ADOPTION CHALLENGES

No.	SA Scrum and Agile Adoption Challenges	Frequency
1	Lack of knowledge/training/skills	6
2	Organizational culture/mindset	5
3	Lack of structure/planning	5
4	Requirements creep/story changes	5
5	Communication issues	4
6	Motivational issues	4
7	Lack of resources (labor and non-labor resources)	3
8	Management inefficiencies	3
9	Workload	3
10	Team distribution	2
11	No/lack of individual recognition	1
12	Team size	1

Due to the nature of software development being a social phenomenon, and Agile being at the forefront of this complex human interaction activity [25], expectations that noise or disturbance by team members would have been identified as one of the challenges encountered, are aroused. However, surprisingly this is not the case, and the study by Eccles et al. [29] states, on the contrary, employees welcome it.

The next section discusses the independent variables and their hypothesized relationship with Scrum adoption, which forms part of the custom model's four constructs.

III. THE CUSTOM MODEL CONSTRUCTS AND VARIABLES

The custom model was constructed with DOI being the theoretical base, but we tailored the model to match the context of the application, i.e. Scrum adoption challenges [31]. Not all DOI constructs were used in the study, the three that have been included, due to it being consistently relevant in innovation studies are compatibility, complexity, and relative advantage [28]. The custom model is discussed in section IV. Scrum Adoption Challenges Detection Model (SACDM).

The narrative review method produced the independent variables which were collated and coded, and subsequently used as the input to the model. These independent variables were assigned to either one of the four custom model factors, namely; individual, team, organization and technology. Therefore, the independent variables were tailored towards the specificity of the innovation [30], [31]. The nineteen independent variables affecting Scrum adoption are discussed in this section.

A. Individual Factors

The first set of variables found in the literature deals with the Scrum challenges experienced by individuals within the organization.

Escalation of Commitment: Escalation of Commitment in the software industry context, its defined as continuously assigning resources to projects that indicates signs of failure. Statistics of 30 to 40 percent of software projects that experiences escalation of commitment have been recorded [32]. We have included escalation of commitment to the individual factors construct because it has often been caused by individual developers within Scrum teams who persist with a task even though it is not adding value to the project. The sooner the Scrum team notices this problem (usually in daily stand-ups) the greater the chances of limiting resource wastage.

H1: Thus, it has been hypothesized that escalation of commitment negatively affects the adoption of Scrum within IT organizations.

Experience: While experience may be seen as being knowledgeable and skilled on an event or subject, it also refers to the project team member having mastery of multiple skills sets such as programming languages, management skills etc. The mastery of multiple skill sets is usually obtained by working on various tasks, projects, and teams over a period of time [26]. Experience has also been identified as a contributor to performance of programmers [33].

H2: Experience is therefore hypothesized to have a positive influence on individual willingness to adopt new innovations.

Over Engineering: Over engineering or over engineered solutions can be summarized as software that has more features and functionality added to it than what was required from the client. Reasons that could lead to software being over engineered are lack of communication with stakeholders, bad planning or limited domain knowledge by the Scrum team [14]. This variable has been included as an individual factor because the developers within the development team are responsible for completing the sprint backlog. The development team is included in the sprint planning meeting and if anything related to the backlog item is unclear to the developer during the sprint he or she may liaise with the Scrum team to clear any confusion. We therefore think that over engineering affects innovation adoption negatively.

H3: Over engineering is negatively related to adoption of Scrum. Over engineering will be lower for adopters.

B. Team Factors

The second set of variables is concerned with the individual's perception of team related challenges based on the literature.

Communication: Communication is the act of exchanging information from one individual or group to another using a common system of behavior [26].

H4: Lack of communication therefore have a negative impact on adoption. Communication will be higher for adopters.

Teamwork: Teamwork is the process whereby individuals work together as a team to complete tasks and achieve a common goal or objective [26]. However, teamwork challenges within Agile development methods is a reoccurring problem. Activities which have been documented as important to increase team as well as organizational performance, are, recognizing other's achievement, responding constructively to team member opinions, assisting and supporting others, and showing greater leniency towards team members [12].

H5: More teamwork amongst team members will affect Scrum adoption positively. Adopters will be in a team that show greater signs of teamwork than non-adopters.

Specialization: The term specialization is the process of an individual having a high degree of knowledge and skills within a domain of interest, improving the individual's

proficiency and expertise within his or her role. Agile software development teams prioritize the idea of self-organizing teams in which team members share overlapping skills which improves flexibility. The problem with work specialization is that it doesn't make provision for interchangeable roles [15].

H6: Specialization within a Scrum team negatively affects adoption. Specialization will be less for adopters.

Sprint Management: Sprint management is defined as a time boxed activity that monitors and manages the progress of a sprint. Events that prevents sprint cycles from operating optimally includes scope creep, lack of timeous feedback, lack of planning and lack of team cohesion [11], [24].

H7: The better the sprint management within the team the more likely there will be adoption. Adopters will have better sprint management.

Change Resistant: Resistance to change within the context of the work environment is a process in which the employee sees change as disruptive and intrusive [34]. With Agile process introduction, developers tend to display signs of cautious optimism, skepticism, and enthusiasm with the problem of some developers not welcoming the change, resisting it without much thought put into it [35].

H8: Teams that are reluctant to change their ways of doing things are more likely to be non-adopters. Adopters will have a lower degree of change resistance.

C. Organization Factors

The third set of variables deals with the individual's perception of organizational challenges encountered within literature.

Training: Training is the acquisition of skills and knowledge through teaching and learning which improves the competency areas of the individual or group. The training within this research study applies to employees going for training to achieve the goals and objectives of the organization they represent [26].

H9: It is hypothesized that staff training is higher in organizations that adopted Scrum. Lack of staff training is hypothesized to negatively affect adoption.

Recognition: Recognition from a business point of view, is seen as matching remuneration, rewards and benefits with the productivity levels of the workers [36]. The study by Noruwana and Tanner [7] identified that individuals were unhappy with the lack of recognition for their contributions within the team because the recognition was given on a team level which does not distinguish between team member productivity levels.

H10: Therefore, individual recognition is hypothesized to improve the likelihood of adoption. Recognition contributes positively towards adoption success.

Quality: The quality that is being referred to is that of software quality and how its correctness contributes toward software projects meeting the business requirements and user expectations. There have been many attempts to improve the quality of software project throughput, yet many software projects continue to fail [11].

H11: Higher degree of throughput quality is positively correlated to Scrum adoption.

Resources: Resources in the context of this study refers to any asset or service, whether it is staff, materials, money etc. that allows the organization to operate sufficiently in producing products and services requested by clients. An exploratory case study conducted by Noruwana and Tanner [7] on a SA company identified lack of labor resources ranging from Agile experience, skillsets, and team members having to perform more than their fair share of responsibilities.

H12: Supply of labor and non-labor resources are more for adopters than non-adopters. Lack of resources is hypothesized to have a negative impact on adoption.

Collaboration: Included in the Agile Manifesto is the statement "Customer collaboration over contract negotiation". What this suggests is that individuals, teams and organizations need to work closely together with clients to achieve a common goal instead of spending most of their effort on securing the deal. Research indicates that many organizations and customers within Agile environments do not abide by this principle. Some of the challenges faced by the lack of collaboration are Agile teams being overly committed, loss of business and productivity, products and user requirements not aligning, and poor feedback mechanisms [37].

H13: Adopters have more collaboration with their clients than non-adopters. Collaboration is positively correlated to adoption.

Management Support: Management support allow organizations to look at innovation adoption from a positive perspective, and this creates a conducive environment for innovativeness [26]. Two findings that are of interest for this study is firstly, management that penalizes employees for mistakes made does not encourage innovativeness, and secondly, management support has a direct effect on the adoption of innovation [31].

H14: Therefore, management support is hypothesized to be a crucial contributor to the adoption of Scrum. Management support will be higher for adopters.

Organizational Culture: Organizational culture which is so eloquently defined by E.H. Schein (1990) is quoted as saying "a pattern of basic assumptions invented, discovered or developed by a given group as it learns to cope with its problems of external adaptation and integration that has worked well enough to be considered valid and, therefore, is to be taught to new members as the correct way to perceive, think, and feel in relation to those problems" [26].

H15: A supportive company culture is positively related to Scrum adoption. Individuals who adopt Scrum will be in firms with a supportive organizational culture.

Organizational Structure: The organization structure is a system with defined activities which governs how individuals within roles, and procedures are coordinated to achieve the goals and objectives of the organization. Evidence from previous studies indicates that organizations that allow for an open and integrated environment with a less hierarchical structure improves the innovation adoption rates [31]. Whilst previous studies have broken up organizational structure into the three components of centralization, formalization and integration, the authors however kept it as a single variable for reasons of simplicity.

H16: Organizational structure contributes negatively towards Scrum adoption. Lack of structure is hypothesized to be higher in organizations who adopt Scrum.

D. Technology Factors

The fourth set of variables relates to the DOI theory and the individual's perception of the Scrum methodology as an innovation.

Relative Advantage: Relative advantage is measured (within the context of this study) as the degree to which Scrum has made a positive contribution to the existing conditions of the individual and organization [31].

H17: There is a linear relationship between perceived relative advantage and Scrum adoption. Adopters will perceive higher relative advantage in the methodology.

Complexity: Complexity is the degree of difficulty experienced by individuals and organizations when adopting Scrum as an innovation [31].

H18: There is a linear relationship between perceived complexity and the adoption of Scrum. Non-adopters will perceive a higher degree of complexity in Scrum than adopters.

Compatibility: The compatibility of Scrum against the existing values of the company and the individuals whom it employs provide an indication to the likelihood of individuals adopting or rejecting it [31].

H19: There is more compatibility with adopters of Scrum than companies and individuals that rejects it. The higher the compatibility the more likely there is the potential of adoption.

The following section is dedicated to proposing a practical application model to detect the presence of Scrum adoption challenges encountered by individuals within software organizations. The aforementioned hypotheses are tested with the use of this model.

IV. Scrum Adoption Challenges Detection Model (SACDM)

While DOI as a theoretical model covers both the individual and organizational aspects of IT adoption studies [31], it is not enough though for complex methodologies within Agile, such as Scrum.

According to Chau and Tam [30], diffusion variables are not sufficient enough as a predictor of complex organizational innovation adoption, as the independent and control variables it provides might be of limitation. Bayer and Melone [50] provides a few failures of DOI due to its limitations, two of the failures being the lack of theoretical justification for the five adopter categories without sufficient empirical support for the classifications used, and not taking the interactions between various social systems into account.

Because the Scrum methodology is a social phenomenon with strong emphasis on project management, it is important that we develop a model that include behavioral aspects to it, which unfortunately, haven't received much attention by previous IS adoption studies [51]. As Chan and Thong [26] so eloquently puts it, "There is an urgent need to conduct a critical review of the extant literature to develop a conceptual framework (CF) for Agile methodologies acceptance."

We used the idea of Senapathi et al. [44], who developed a CF based on a synthesis of past research in DOI, agile implementation, and IS implementation literature. Their five factor groups are agile innovation, organizational, sociological, team, and technological factors, which have been adopted from agile, XP, DOI, and IS frameworks and literature.

With a similar approach this study uses a CF which is a synthesis of research composed of theoretical models, DOI, Agile adoption, Scrum adoption, SDM adoption, and IS innovation literature.

As indicated in the introduction of this paper, a model will assist to detect Scrum adoption challenges within software organizations. The detection of the challenges can be generalized with the help of a quantitative survey research design. In future studies the model can contribute towards predictive analysis of Scrum adoption. The custom model is adapted from the study by Sultan and Chan [31], which looked at the adoption of OT in software companies. The authors propose a Scrum adoption challenges detection model, incorporating the DOI theoretical model, by extending the theoretical model to include constructs such as organizational factors, team factors and individual factors (SACDM) as displayed in Fig. 1.

Fig. 1 displays the custom model with the four constructs comprising of nineteen independent variables, and one dependent variable. The model for this research indicates the variables which are hypothesized to have an influence in adoption of a new methodology such as Scrum by individuals, and the proposed directionality of these relationships.
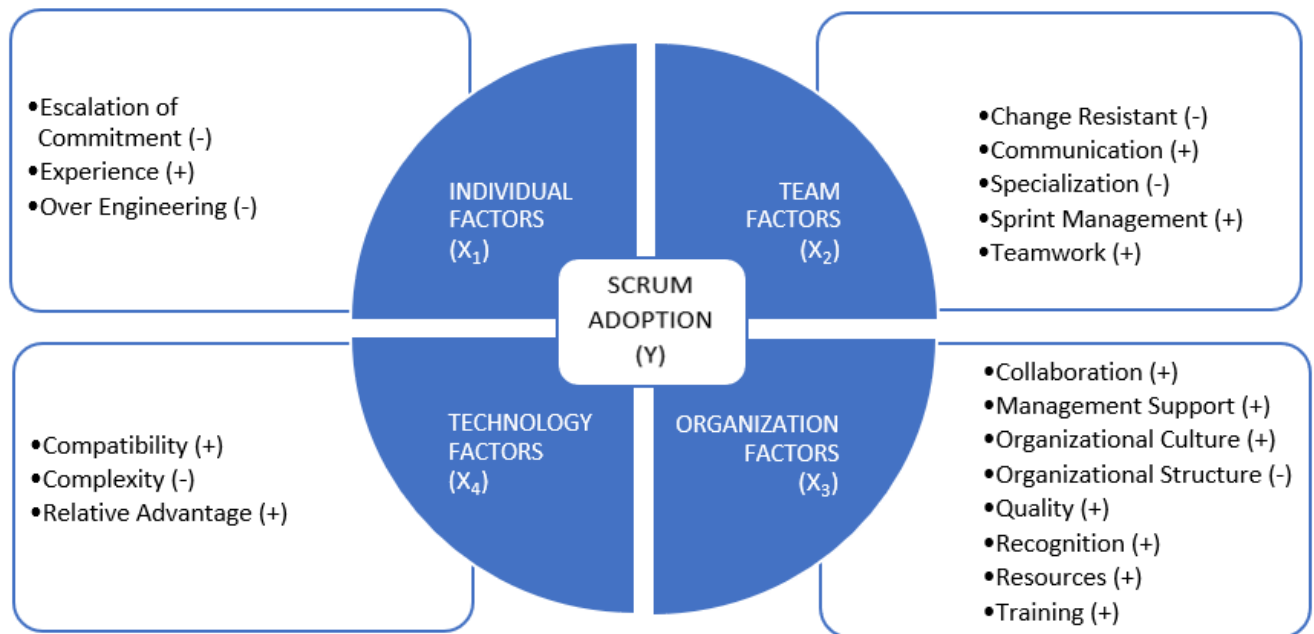
Independent variables are depicted as X_1, X_2, X_3 and X_4.

Dependent variable is Y with $Y = f(X_1.X_2.X_3.X_4)$.

When $Y = 1$, the individual within a software organization is an adopter of Scrum.

When $Y = 0$, the individual within a software organization is a non-adopter of Scrum.

Note: The hypothesized relationships are shown by the symbols in parenthesis.

Fig. 1. Scrum Adoption Challenges Detection Model.

The final list of independent variables is displayed below, in ascending order.

- Change Resistant
- Collaboration
- Communication
- Compatibility
- Complexity
- Escalation of Commitment
- Experience
- Management Support
- Organizational Culture
- Organizational Structure
- Over Engineering
- Quality
- Recognition
- Relative Advantage
- Resources
- Specialization
- Sprint Management
- Teamwork
- Training

This model will be used to differentiate adopters from non-adopters of Scrum, which is important to understand which constructs and variables significantly contributes towards the acceptance or rejection of Scrum. The dependent variable in this study (Y) is the adoption of Scrum. This will be related to the independent variables included in the four sets of variables shown in Fig. 1: (X_1) individual factors; (X_2) team factors; (X_3) organizational factors; and (X_4) technology factors.

This paper concludes by providing a summary of the SACDM and the potential advantages it may have for software organizations looking to improve on their project management operations.

V. CONCLUSION

Scrum is the most widely used SDM at present, providing many organizations with a simple to understand methodology to complete project management tasks. While the advantages to using this methodology are easily noticed by adopters, the challenges during the adoption stage are currently not quantitatively detected. Most research on the

adoption challenges primarily focused on qualitative measures for detection with case studies being the most implemented strategy. The inability to easily detect these adoption challenges can lead to teams and individuals within software organizations not using Scrum correctly or not adopting Scrum altogether, which could potentially limit the successful outcomes of a project.

It is proposed that a practically applied Scrum adoption challenges detection model such as SACDM, will aid in the awareness of the challenges faced by software organizations, and thus potentially limit the negative effects these adoption challenges might have on the individuals and organizations using Scrum. The extant Scrum adoption challenges were acquired through a narrative review of Scrum adoption challenges, both within the global and SA context. The SACDM was developed to detect Scrum adoption challenges with the objective of equipping adopters with the knowledge and awareness to overcome them.

Future research will aim to improve the SACDM, by designing an automated Scrum adoption challenges self-evaluation questionnaire. This questionnaire will allow the authors to gather and analyze the response data, which will be used to create a generalized result-set for the benefit of potential adopters to improve their awareness of Scrum adoption challenges and the correlation to Scrum adoption. The long-term vision of the SACDM is to allow individuals and organizations to predict Scrum adoption with the help of a research database and algorithms used to perform Scrum adoption predictive analysis.

REFERENCES

[1] Leffingwell, D. 2011. Agile software requirements: lean requirements practices for teams, programs, and the enterprise. Boston: Pearson Education, Inc.

[2] Schwaber, K. & Sutherland, J. 2011. The Scrum Guide. Scrum.org, October, 2: 17. http://www.Scrumalliance.org/.

[3] Pressman, R.S. 2005. Software Engineering. A Practitioner's Approach. 6th ed. New York, USA: McGraw-Hill.

[4] Allisy-Roberts, P., Ambrosi, P., Bartlett, D.T., Coursey, B.M., DeWerd, L.A., Fantuzzi, E. & McDonald, J.C. 2017. The 11th Annual State of Agile Report. Journal of the ICRU, 6(2): 7–8. https://academic.oup.com/jicru/article-lookup/doi/10.1093/jicru/ndl025.

[5] Rogers, E.M. 2003. Diffusion of Innovations, 5th Edition. Free Press. https://books.google.co.za/books?id=9U1K5LjUOwEC.

[6] Chan, K.Y. & Thong, J.Y.L. 2007. An Integrated Framework of Individual Acceptance of Agile Methodologies. PACIS 2007 Proceedings: 154.

[7] Noruwana, N. & Tanner, M. 2012. Understanding the structured processes followed by organisations prior to engaging in Agile processes: A South African Perspective. SACJ, (48): 8.

[8] Vijayasarathy, L. & Turk, D. 2012. Drivers of Agile software development use: Dialectic interplay between benefits and hindrances. Information and Software Technology, 54(2): 137–148.

[9] Anderson, D.J., Concas, G., Lunesu, M.I., Marchesi, M. & Zhang, H. 2012. A Comparative Study of Scrum and Kanban Approaches on a Real Case Study Using Simulation. In C. Wohlin, ed. Agile Processes in Software Engineering and Extreme Programming. Malmö: Springer Berlin Heidelberg: 123–137.

[10] Blankenship, J., Bussa, M. & Millett, S. 2011. Pro Agile .NET Development with Scrum. Berkeley, CA: Apress. http://link.springer.com/10.1007/978-1-4302-3534-7.

[11] Tanner, M., Khalane, T. 2013. Software Quality Assurance in Scrum: The need for concrete guidance on SQA strategies in meeting user expectations. IEEE ICAST 2013: 6.

[12] Stray, V.G., Moe, N.B. & Dingsøyr, T. 2011. Challenges to Teamwork: A Multiple Case Study of Two Agile Teams. In A. Sillitti, O. Hazzan, E. Bache, & X. Albaladejo, eds. Agile Processes in Software Engineering and Extreme Programming. Madrid: Springer Berlin Heidelberg: 146–161.

[13] Asnawi, A.L., Gravell, A.M. & Wills, G.B. 2011. Empirical Investigation on Agile Methods Usage: Issues Identified from Early Adopters in Malaysia. In A. Sillitti, O. Hazzan, E. Bache, & X. Albaladejo, eds. Agile Processes in Software Engineering and Extreme Programming. Madrid: Springer Berlin Heidelberg: 192–207.

[14] Santos, R., Flentge, F., Begin, M.-E. & Navarro, V. 2011. Agile Technical Management of Industrial Contracts: Scrum Development of Ground Segment Software at the European Space Agency. In A. Sillitti, O. Hazzan, E. Bache, & X. Albaladejo, eds. Agile Processes in Software Engineering and Extreme Programming. Madrid: Springer Berlin Heidelberg: 290–305.

[15] Fægri, T.E. 2010. Adoption of Team Estimation in a Specialist Organizational Environment. In A. Sillitti, A. Martin, X. Wang, & E. Whitworth, eds. Agile Processes in Software Engineering and Extreme Programming. Trondheim: Springer Berlin Heidelberg: 28–42.

[16] Marchenko, A. & Abrahamsson, P. 2008. Scrum in a Multiproject Environment: An Ethnographically-Inspired Case Study on the Adoption Challenges. In Agile 2008 Conference. IEEE: 15–26. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4599449.

[17] Overhage, S., Schlauderer, S., Birkmeier, D. & Miller, J. 2011. What Makes IT Personnel Adopt Scrum? A Framework of Drivers and Inhibitors to Developer Acceptance. In 2011 44th Hawaii International Conference on System Sciences. IEEE: 1–10. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5718964.

[18] Heikkila, V.T., Paasivaara, M. & Lassenius, C. 2013. ScrumBut, But Does it Matter? A Mixed-Method Study of the Planning Process of a Multi-team Scrum Organization. In 2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement. IEEE: 85–94. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6681341.

[19] Dong, L. 2008. Exploring the impact of top management support of enterprise systems implementations outcomes: Two cases. Business Process Management Journal, 14(2): 204–218.

[20] Dong, L., Neufeld, D. & Higgins, C. 2009. Top management support of enterprise systems implementations. Journal of Information technology, 24(1): 55–80.

[21] Hardgrave, B.C. & Johnson, R.A. 2003. Toward an information systems development acceptance model: the case of object-oriented systems development. IEEE Transactions on Engineering Management, 50(3): 322–336.

[22] Mnkandla, E., Dwolatzky, B. 2004. A survey of Agile methodologies. THE TRANSACTIONS OF THE SA INSTITUTE OF ELECTRICAL ENGINEERS, 3(December): 236–247.

[23] Du Toit, R. 2013. Enterprise Agile Adoption with Parental Guidance - PG. http://Agileafrica.jcse.org.za/sites/default/files/Riaan-du-Doit-Agile-Adoption-with-Parental-Guidance.pdf 12 August 2014.

[24] Tanner, M. & Mackinnon, A. 2013. Sources of Disturbances Experienced During a Scrum Sprint. ICIME2013.

[25] Tanner, M., Wallace, C. 2012. TOWARDS AN UNDERSTANDING OF THE CONTEXTUAL INFLUENCES ON DISTRIBUTED AGILE SOFTWARE DEVELOPMENT: A THEORY OF PACTICE PERSPECTIVE. In European Conference on Information Systems (ECIS). Association for Information Systems AIS Electronic Library (AISeL): 13.

[26] Chan, F.K.Y. & Thong, J.Y.L. 2009. Acceptance of Agile methodologies: A critical review and conceptual framework. Decision support systems, 46(4): 803–814.

[27] Mohan, K. & Ahlemann, F. 2013. Understanding acceptance of information system development and management methodologies by actual users: A review and assessment of existing literature. International Journal of Information Management, 33(5): 831–839.

[28] Hardgrave, B.C., Davis, F.D. & Riemenschneider, C.K. 2003. Investigating determinants of software developers' intentions to follow methodologies. Journal of Management Information Systems, 20(1): 123–151.

[29] Eccles, M., Smith, J., Tanner, M., Van Belle, J.-P. & van der Watt, S. 2010. The Impact of Collocation on the Effectiveness of Agile IS Development Teams. Communications of the IBIMA, 2010: 1–11.

[30] Chau, P.Y.K. & Tam, K.Y. 1997. Factors affecting the adoption of open systems: an exploratory study. MIS quarterly: 1–24.

[31] Sultan, F. & Chan, L. 2000. The adoption of new technology: the case of object-oriented computing in software companies. IEEE transactions on Engineering Management, 47(1): 106–126.

[32] Stray, V.G., Moe, N.B. & Dybå, T. 2012. Escalation of Commitment: A Longitudinal Case Study of Daily Meetings. In C. Wohlin, ed. Lecture Notes in Business Information Processing. Malmö: Springer Berlin Heidelberg: 153–167. http://link.springer.com/10.1007/978-3-642-30350-0_11.

[33] Brooks, R.E. 1980. Studying programmer behavior experimentally: The problems of proper methodology. Communications of the ACM, 23(4): 207–213.

[34] Strebel, P. 1996. Why do employees resist change? Harvard business review, 74(3): 86.

[35] Cohn, M. & Ford, D. 2003. Introducing an Agile process to an organization [software development]. Computer, 36(6): 74–78.

[36] Bishop, J. 1987. The recognition and reward of employee performance. Journal of Labor Economics, 5(4, Part 2): S36–S56.

[37] Hoda, R., Noble, J. & Marshall, S. 2011b. The impact of inadequate customer collaboration on self-organizing Agile teams. Information and Software Technology, 53(5): 521–534.

[38] Kapitsaki, G.M. & Christou, M. 2014. Where Is Scrum in the Current Agile World? In Proceedings of the 9th International Conference on Evaluation of Novel Approaches to Software Engineering. SCITEPRESS – Science and Technology Publications: 101–108. http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0004867701010108.

[39] Bjarnason, E. & Regnell, B. 2012. Evidence-Based Timelines for Agile Project Retrospectives – A Method Proposal. In C. Wohlin, ed. Lecture Notes in Business Information Processing. Malmö: Springer Berlin Heidelberg: 177–184. http://link.springer.com/10.1007/978-3-642-30350-0_13.

[40] Irrazabal, E., Vásquez, F., Díaz, R. & Garzás, J. 2011. Applying ISO/IEC 12207:2008 with SCRUM and Agile Methods. In R. V. O'Connor, T. Rout, F. McCaffery, & A. Dorling, eds. Software Process Improvement and Capability Determination. Dublin: Springer Berlin Heidelberg: 169–180.

[41] Hoda, R., Noble, J. & Marshall, S. 2011a. Supporting Self-organizing Agile Teams. In A. Sillitti, O. Hazzan, E. Bache, & X. Albaladejo, eds. Agile Processes in Software Engineering and Extreme Programming. Madrid: Springer Berlin Heidelberg: 73–87. http://link.springer.com/10.1007/978-3-642-20677-1_6.

[42] Hoda, R., Noble, J. & Marshall, S. 2010. Agile Undercover: When Customers Don't Collaborate. In A. Sillitti, A. Martin, X. Wang, & E. Whitworth, eds. Agile Processes in Software Engineering and Extreme Programming. Trondheim: Springer Berlin Heidelberg: 73–87.

[43] Dorairaj, S., Noble, J. & Malik, P. 2011. Effective Communication in Distributed Agile Software Development Teams. In A. Sillitti, O. Hazzan, E. Bache, & X. Albaladejo, eds. Agile Processes in Software Engineering and Extreme Programming. Madrid: Springer Berlin Heidelberg: 102–116.

[44] Senapathi, M., Middleton, P. & Evans, G. 2011. Factors Affecting Effectiveness of Agile Usage – Insights from the BBC Worldwide Case Study. In A. Sillitti, O. Hazzan, E. Bache, & X. Albaladejo, eds. Agile Processes in Software Engineering and Extreme Programming. Madrid: Springer Berlin Heidelberg: 132–145.

[45] Ressin, M., Abdelnour-Nocera, J. & Smith, A. 2011a. Defects and Agility: Localization Issues in Agile Development Projects. In A. Sillitti, O. Hazzan, E. Bache, & X. Albaladejo, eds. Agile Processes in Software Engineering and Extreme Programming. Madrid: Springer Berlin Heidelberg: 290–305.

[46] Ressin, M., Abdelnour-Nocera, J. & Smith, A. 2011b. Lost in Agility? Approaching Software Localization in Agile Software Development. In A. Sillitti, O. Hazzan, E. Bache, & X. Albaladejo, eds. Agile Processes in Software Engineering and Extreme Programming. Madrid: Springer Berlin Heidelberg: 320–321.

[47] Santos, V. & Goldman, A. 2011. An Approach on Applying Organizational Learning in Agile Software Organizations. In A. Sillitti, O. Hazzan, E. Bache, & X. Albaladejo, eds. Agile Processes in Software Engineering and Extreme Programming. Madrid: Springer Berlin Heidelberg: 324–325.

[48] Kim, E. & Ryoo, S. 2012. Agile Adoption Story from NHN. In 2012 IEEE 36th Annual Computer Software and Applications Conference. IEEE: 476–481. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6340199.

[49] Ihme, T. 2013. Scrum adoption and architectural extensions in developing new service applications of large financial IT systems. Journal of the Brazilian Computer Society, 19(3): 257–274. http://link.springer.com/10.1007/s13173-012-0096-0.

[50] Bayer, J. & Melone, N. 1989. A critique of diffusion theory as a managerial framework for understanding adoption of software engineering innovations. Journal of Systems and Software, 9(2): 161–166.

[51] Jeyaraj, A. & Sabherwal, R. 2008. Adoption of information systems innovations by individuals: A study of processes involving contextual, adopter, and influencer actions. Information and Organization, 18(3): 205–234.

[52] Przybyłek, A. & Zakrzewski, M. 2018. Adopting Collaborative Games into Agile Requirements Engineering. In Proceedings of the 13th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2018), 54-64.