

Evaluation of Agile Methods for Quality Assurance and Quality Control in ERP Implementation

Jawad Javed Akbar Baig, Atif Shah, Faisal Sajjad

Information and Computer Science Department

King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.

Email: [g201507890, g201408760, g201409220]@kfupm.edu.sa

Abstract—Most organizations are now adopting Enterprise Resource Planning (ERP) solutions to standardize, formalized and automate their processes. The organization usually hire IT Consulting Firms (ITCF) for ERP implementation due to its complexity, large scale, and domain knowledge. ERP brings radical changes in the environment, daily processes, and interactions of the organization. To make these changes gradual and incremental, ITCF are moving towards agile methodology. Quality assurance and quality control play a pivotal role in the success of an ERP implementation, with the use of agile methods the implementation becomes more complex. In this paper, grounded theory is used to study the rapid parallel processes of testing, validation, and verification for different phases of ERP implementation through using agile methods. In our study, 23 industry professionals participated from six different organizations. Each participant had experience of various ERP implementations in diverse geographical locations in various domains with different roles. This paper discuss and analyze different agile methods, which can reduce challenges in quality control and assurance during ERP implementation. The results show that agile methods (daily scrum meeting, pair programming and frequent reviews) reduce complexity and increase the quality of ERP implementation.

Index Terms—Agile Methods, ERP Implementation, Grounded Theory, Quality Assurance, Quality Control

I. INTRODUCTION

Enterprise resource planning (ERP) solutions are popular information technology systems from 1990s. Today they are categorized among big IT investment in the world [1] [2]. They are special type of commercial enterprise software, design to cover end to end needs of a specific domain. They provide standard and flexible processes in order to integrate and automate daily operations of an organization. For implementation of an ERP solution, organizations hire experienced IT Consulting Firms (ITCF). Risks like delay in schedule and over budget are very common in ERP implementations due to miscommunication, late client feedback and change in requirements [3] [15] [16]. So ITCF are now adopting agile methods to increase visibility and agility [4].

Agile methods have gained popularity in last two decades [5]. Agile increases communication, collaboration and progress visibility between team members and client. Agile also increase the ability to adapt changes. Although initially agile methods specially Scrum [6] and XP [7] were proposed for lightweight development, but with the passage of time the benefits of agile have gained the attention of enterprise software teams as well [8] [9] [10]. Enterprise software project

is complex due to big budget, long schedule, huge scope and large geographically distributed teams. Still companies implementing and developing enterprise software, are using agile methods and taking benefits from it [11] [12] [13].

Normally 25% - 50% of total budget is spent only on testing in ERP implementation [17] and total quality control and assurance will take more budget. ERP implementation can be subdivided into phases like business process reengineering, configuration, customization, integration, data conversion, and maintenance. For quality control and assurance of each phase, there are different effective ways. For example in business process reengineering best ways are reviews, walkthrough, and inspections but for data conversion best way is to auto-reconcile data using code. However, introducing agile methods in these phases will be surely an effect. Therefore, a study is required to analyze these effects. This paper discusses changes and effects in quality control and assurance in an ERP implementation while using agile methods.

Based on collected and analyzed information during interviews, we found out that there are three distinguish opinions exist in the industry. According to 70% of practitioners selective agile methods (daily scrum meeting, pair programming, and frequent reviews) based on needs and situation is more beneficial instead of applying whole agile framework (Scrum [6] , XP [7] and Lean [20]). 25% of practitioners share the opinion that use of selective methods will limit benefits. According to them selection of agile methods is not easy and require a lot of experience. Therefore, they suggest to use a particular agile framework as whole. Remaining 5% of practitioners say that agile and ERP implementation cannot work together because agile is client oriented and ERP implementation is product centric. According to majority of 95% agile is helpful for quality control and assurance of ERP implementation but selection of correct agile methods is difficult. This paper focuses on this problem and analyze a list of agile methods.

In this paper, grounded theory [18] [19] is used to conduct empirical study. We first collected information from literature by conducted small literature review, to enable ourselves and create an initial set of questions for interviews. We found very less number of studies in the literature which discuss the quality of an ERP implementation using agile methods. So that is why we adopted grounded theory to form new study and theory.

Section 2 contains details of the literature review. Details related research design are explained in section 3. Section 4 analyze agile methods and their benefits. We concluded by sharing threats to validity in section 5 and our final thoughts in section 6.

II. RELATED WORK

We conducted literature to collect substantial information for interviews and enable ourselves to understand and analyze information provided by practitioners. There is very less literature available for our area, so we studied papers on related topics like ERP implementation, factor to impact ERP success, agile methods in ERP and, quality and testing in ERP and agile.

Kristi and Ahmad [21] conducted a literature review to collect critical factors and best practices for ERP implementation. They applied these critical factors and best practices on a case study. They shared that testing is very critical and costly part of ERP implementation. Chung and Snyder [1] shared that ERP systems have become the largest IT investment. They analyzed ERP solutions by collecting data from industrial reports and ERP advocates. AlMashari [2] discussed ERP topics like ERP adoption, technical aspects of ERP and ERP in IS curricula. He also studied reasons why organizations are investing huge budget to adopt ERP solutions.

Umple et al. [22] discussed that ERP systems are very complex systems and require big cost and resources for implementation. They provided factors for success and implementation procedures. Authors also shared a case study of successful ERP implementation to support his arguments. Al-Mashari et al. [23] discussed critical factors of success of ERP implementation based on literature review and organizational experience. They shared that system testing is one of the critical factors for success. Bokovec et al. [24] conducted research on complete lifecycle of global ERP implementation. They discussed each phase in detail including testing of different types. Stensrud and Myrteit [25] performed an empirical study on 30 ERP projects and shared how to identify which ERP project performed well.

Huo et al. [26] conducted research on agile methods and their effects on quality. They compared effects on quality using agile methods with waterfall methodology. They shared that in agile the frequency of QA practices is high, which improve final quality but also increase cost. Itkonen et al. [27] discussed that in agile there are constructive quality building practices but quality assessment base on destructive attitude is very less. They suggested that quality in agile development can be improved by adding some quality control techniques. Bhasin [28] shared that agile methods have built-in quality management mechanism but still in industry practitioners are facing some issues. They studied various challenges in quality management while using agile methods and proposed some suggestions to overcome challenges.

Arachchi et al. [29] shared that one of the most important factors in ERP success is quality. They concluded that reactive measures of quality control are not enough and we should take

proactive measure by focusing on quality assurance. Wiczorek et al. [30] discussed that by manually testing we cannot systematically test an ERP system, so we required automatic testing. For automatic testing, we require test data. They discussed how to generate test data and proposed multiple solutions. Malik et al. [31] studied performance testing in enterprise system after addition of new feature. They proposed an automated approach to identify performance deviation in the subsystem. Asgar et al. [32] analyzed that testing the data conversion in complex ERP is very difficult. They proposed a test-first approach to identify wrong mapping at the early stage. In order to reduce time, they proposed parallel execution of data conversion approach. They also shared industrial case study which gave positive results using proposed methods.

Patel and Kollana [33] presented experience report of test case reuse in enterprise software implementation. They shared that their company created standard and specific test cases with the intention of reusing over multiple enterprise software implementations. Authors analyzed it using goal question metric approach. Talby et al. [34] analyzed and shared that agile testing practices improve quality and productivity. Stolberg [35] conducted research on continuous integration in agile testing. He explained that continuous integration automatically increase frequent and early testing and as result, it improves quality. Sangwan and Laplante [36] conducted the study on test driven development in geographically distributed development. They proposed a scale up test driven development for the large-scale geographically distributed project.

III. RESEARCH DESIGN

We used grounded theory [18] to conduct our research. Main reason for selecting grounded theory was that it helps to create a new theory based on collected data instead of using data to validate any existing theory. Second reason was that very few information related our topic was available in literature and grounded theory helps in such situation. Third reason was that agile focus on individuals and communications. So grounded theory helped us to directly interact with individuals and collect ideas which can improve collaboration and communication. Last reason is that grounded theory is a successful method of conducting an empirical study on software engineering topics. Figure 1 shows the flow of grounded theory activities. Below are research questions of our study.

- Does agile methods help to increase quality of ERP implementation?
- Which agile methods are currently used in industry for ERP implementation?

A. Applied Grounded Theory

We used standard grounded theory techniques to perform our research. Applied grounded theory techniques are explained below:

1) *Minor Literature Review*: Minor literature review was conducted to enable ourselves for discussion and communication with practitioners and to formulate an initial draft of questions.

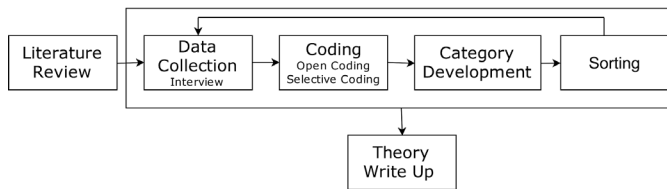


Fig. 1. Grounded Theory

2) *Interview with Practitioners*: For data collection interviews with 23 practitioners from six companies were conducted. They had experience in different ERP implementations and had worked in various geographies, roles and companies.

3) *Open Coding*: After interview Open coding was performed on collected data. It is the first step of analysis in which we extract important concepts and ideas from raw data and then assign codes. Codes are the combination of two to three words which represent an idea in a short way. In this collect all ideas even if they does not belong to our research questions. This helps to reduce the risk of ignoring good points.

4) *Selective Coding*: In order to identify concepts and ideas related to the quality of ERP only, selective coding was performed. It is similar to open coding but in this, we focus on a specific area or domain. It is also called focused coding.

5) *Category Development*: In this activity we combine the collected codes into groups based of similarity and remove duplicates. This help to form basic structure of our theory.

6) *Sorting*: In sorting we sort the codes and category logically. This is done to make theory more understandable and logical.

7) *Write Up*: In this step we convert our theory into final report or paper.

B. Companies

To conduct our research we selected companies who had diverse experience of multiple ERP implementations in different geographical locations. We also interviewed with one of the university who recently implemented an ERP system. In table I the dynamics of companies are explained in detail.

C. Participants

We conducted interviews with 23 professional who had good and diverse experience in ERP implementation. Interviews were open-ended because we wanted to gather ideas and concepts. Practitioner shared their experience and good practices they used. Practitioner had good experience in the domain, table II provides details related this. Practitioners worked on various ERP implementations of different domains, table II contain information about this. The practitioner also performed different roles in ERP implementation. Table III contain details of their current roles.

IV. RESULTS AND DISCUSSION

There exist three different opinions among practitioners about the success of agile methods for improving the quality of ERP implementation. The majority of 70% believe that

TABLE I
LIST OF COMPANIES FROM WHICH PRACTITIONERS TOOK PART IN STUDY

| Company | Dynamics of Company |
|---------|--|
| A | Software house from Pakistan, which performs ERP implementation of PeopleSoft Oracle, Flexcube Oracle, and locally developed ERPs. It has three offices in three different cities of Pakistan. It has performed 38 implementation of PeopleSoft in Pakistan, Philippines, Malaysia, Indonesia, Kingdom of Saudia Arabia and Turkey and 9 implementation of Flexcube in Pakistan and Qatar. There are many projects of support too. |
| B | Software house which performs ERP implementation by outsourcing major part of implementation. It has three office in three Countries (Two in Pakistan and One in Kingdom of Saudia Arabia). Company has performed 6 implementation of PeopleSoft in Kingdom of Saudia Arabia and Pakistan. |
| C | Software house resides in Pakistan and have only one office. It works as offshore office for many different ITCF for PeopleSoft implementation. It has performed implementation of MS Dynamics for more than 11 corporations in Pakistan. |
| D | Software house from Malaysia, which performs ERP implementation of PeopleSoft in Malaysia and Indonesia. Company has performed 6 implementations and have offices in both countries Malaysia and Indonesia. |
| E | Software house from Saudia Arabia, which performs SAP implementation. |
| F | IT team of University which recently implemented PeopleSoft campus solution. |

TABLE II
PRACTITIONER'S EXPERIENCE

| Sr. | Year of Experience | Practitioner Count | Sr. | No of ERP Implementations | Practitioner Count |
|-----|--------------------|--------------------|-----|---------------------------|--------------------|
| 1 | 3 to 5 | 9 | 1 | 3 to 5 | 6 |
| 2 | 6 to 10 | 9 | 2 | 6 to 10 | 12 |
| 3 | 11 to 15 | 2 | 3 | 11 to 15 | 2 |
| 4 | 15+ | 3 | 4 | 15+ | 3 |

selective agile methods according to situation and need, can improve the quality of ERP implementation. Whereas according to 25%, whole agile frameworks should be used because the selection of agile methods according to the situation is very difficult. So this study provides a list of agile methods for quality improvement of ERP implementation. Figure 2 contains the details of our theory.

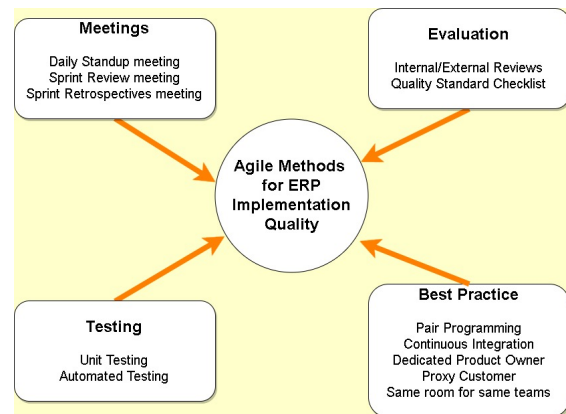


Fig. 2. Agile Methods for ERP implementation Quality

TABLE III
PRACTITIONERS' ROLES STATISTICS

| Sr. | Role | Total Count |
|-----|---|-------------|
| 1 | CTO | 1 |
| 2 | Senior Vice President | 1 |
| 3 | Director | 3 |
| 4 | Project Manager | 5 |
| 5 | Solution Architecture | 4 |
| 6 | Functional/Technical Consultant and Tead Lead | 7 |
| 7 | Client side IT team manager | 2 |

A. Meetings

Some meetings play important role in reducing miscommunication among team members and client. Below are some meetings which we found were helpful to increase quality.

1) *Daily standup meeting*: The purpose of daily standup meeting is to increase the progress visibility. But it can also be used to increase communication and collaboration. Everyone in daily standup answer three questions, "What I did yesterday?", "What will I do today?" and "What are dependencies?". Some practitioners say that it's kind of quick review of the whole team by the whole team and it can be helpful in all phases of ERP.

"For me, daily standup calls are a daily short review of team progress, individual progress, and dependencies, by the whole team." - Project Manager

2) *Sprint review meeting*: Mainly sprint review meeting is part of Scrum methodology in which after every sprint (iteration) a meeting is arranged with client in which work is reviewed. This helps in getting early client feedback. It also clarifies many misconception from client's mind.

"Sprint review meeting is quick, iterative and early user acceptance testing." - Director

3) *Sprint Retrospectives meeting*: Sprint retrospective meeting is also part of Scrum methodology which is conducted after Sprint review meeting. The main purpose of this is to consider and analyze areas of improvement for future iterations. This meeting corrects problem at early stages.

"I believe that most important meeting among all meetings is Sprint Retrospectives meeting. It makes require changes to reduce problems at the early stage." - CTO

B. Evaluation

Evaluation of each task assigned to resources is important to control quality.

1) *Internal/External Reviews*: Review is preemptive quality control activity. Interval reviews are conducted by experienced resources generally solution architects from inside team. Validation and verification both can be done in it. External reviews can be conducted by experienced resources from a different team, external consultant or even client. Including client in external reviews depend on their availability and domain

knowledge. If the client does not have domain knowledge it will just progress update meeting.

"For a manager reviews are small building block for quality prediction of whole end product. It helps to locate problems." - Project Manager

2) *Quality Standard Checklist*: Organizations create checklists for each task to maintain quality standards. These are generally used as input of reviews but we can share these checklists with resources as guidelines. Checklists can vary in different phases and task according to need.

"We create a combination of standard guidelines in the form of a checklist for our employee and reviewer to maintain quality." - Solution Architect

C. Testing Methods

1) *Unit Testing*: Unit testing is done by the developer himself before reviews. Good unit testing can result in a good final product. It highly depends on the quality of unit tests. Unit testing gives initial and quick feedback of program to the developer. It is more helpful for phases like configurations, customizations and in the development of data conversion utilities.

2) *Automated Testing*: Agile suggest having as much automated testing as possible. Because in agile requirements change a lot and we have to repeat testing at the end of every iteration or change. So automating the frequent process can reduce cost and the same time. Automated testing can be used for configurations, customizations, integrations, data conversion and maintenance. In data conversion, automated reconciliation of data helps to reduces time and data errors.

D. Best Practices

1) *Pair Programming*: In pair programming, two resources sit side by side or on the same PC and work on the same task. Mostly pair programming is used to execute complex tasks. For the success of pair programming, the understanding of pairs is very important. Resources passively perform continues review during whole activity and as result, it increases quality and reduces errors.

"When a task is complex and has a pivotal role then we use pair programing." - Solution Architect

2) *Continuous Integration*: Continuous integration is a practice of software engineering in which work of the team is integrated into one location multiple time on same day. Sometime workable components fail when merged together. So this practice detects bugs early which were ignored in unit testing. It also reduces the cost of integration and last minute chaos. This practice can be used in configurations, customizations, integrations, data conversion and maintenance. But practitioner should be careful in taking backups.

3) *Dedicated Product Owner*: Client availability in agile methods is very crucial. So that is why Scrum suggest to have dedicated product owner from client side who should help in taking decisions and provide feedbacks. Product owner can help in external reviews and setting direction. Product owner

can warn us about some important points and can give early feedback. We can have multiple product owners for different modules of ERP.

4) *Proxy Customer*: A proxy customer is a person from vendor side who can act on behalf of a client in case of unavailability. The proxy customer should be experienced resource and must have good knowledge of client needs. Most of the time practitioner use onsite senior resources as proxy customer.

5) *Same room for same teams*: Agile suggests locating a team in the same room can increase communication and collaboration. If it is not possible then frequent visit can be helpful. Good communication and collaboration result in good quality and reduce ambiguities. For geographically distributed teams, organizations are using skype groups.

Only continuous integration, unit testing, automated testing test-driven development are restricted to configuration, customization, integration, data conversion and maintenance. Remaining methods can be used in any phase of ERP implementation to improve quality.

V. LIMITATION OF STUDY

Theory based on grounded theory cannot assure hundred percent validity. There can exist threads to its validity. Below are some threads we aspect can limit our observations.

A. Data Gathering Method

We mainly used semi-structured interviews because we did not want to restrict interviewee to particular topics. Interview time was limited and if we had used structured interviews then we could have extracted more knowledge related our area. Due to different geographical locations, we conducted most of the interviews using skype. This can limit results of our study.

B. Restricted ERP Product

We conducted interviews with practitioner who worked on four ERP products, PeopleSoft, Flexcube, SAP and Microsoft dynamics. We were unable schedule interviews from practitioner of other ERP products.

VI. CONCLUSION AND FUTURE DIRECTION

In this study, agile methods were analyzed for ERP implementation quality. We found out that majority of practitioner believe that agile methods can improve the quality of ERP implementation. However, the selection of agile methods according to situation and needs is tricky and require experience. Therefore, we applied grounded theory to form a new theory based on a good combination of agile methods.

Our study shows that most of agile methods can improve quality of all phases of ERP implementation like process reengineering, configuration, customization, integration, data conversion, and maintenance.

We believe that our proposed theory will make the quality control and assurance easy task for an ERP implementation. In future we are planing to implement this theory on an ERP implementation as case study and evaluate the success.

Another interesting area of research is to evaluate challenges and factors which can hinder the success of this theory.

VII. ACKNOWLEDGEMENT

The authors would like to thank King Fahd University of Petroleum and Minerals (KFUPM) for its continuous support of research.

REFERENCES

- [1] S. H. Chung and C. A. Snyder, "Erp adoption: a technological evolution approach," *International Journal of Agile Management Systems*, vol. 2, no. 1, pp. 24–32, 2000.
- [2] M. Al-Mashari, "Enterprise resource planning (erp) systems: a research agenda," *Industrial Management & Data Systems*, vol. 102, no. 3, pp. 165–170, 2002.
- [3] P. Bingi, M. K. Sharma, and J. K. Godla, "Critical issues affecting an erp implementation," *IS Management*, vol. 16, no. 3, pp. 7–14, 1999.
- [4] S. Mahmood and J. J. A. Baig, "Adoption of agile practices in enterprise resource planning implementation: Preliminary results," in *2016 7th International Conference on Computer Science and Information Technology (CSIT)*, pp. 1–5, IEEE, 2016.
- [5] T. Dingsyr, S. Nerur, V. Balijepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," *Journal of Systems and Software*, vol. 85, no. 6, pp. 1213 – 1221, 2012. Special Issue: Agile Development.
- [6] K. Schwaber, *Agile project management with Scrum*. Microsoft Press, 2004.
- [7] R. C. Martin, *Agile Software Development: Principles, Patterns, and Practices*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2003.
- [8] G. Benefield, "Rolling out agile in a large enterprise," in *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, pp. 461–461, Jan 2008.
- [9] A. W. Brown, S. Ambler, and W. Royce, "Agility at scale: Economic governance, measured improvement, and disciplined delivery," in *Proceedings of the 2013 International Conference on Software Engineering, ICSE '13*, (Piscataway, NJ, USA), pp. 873–881, IEEE Press, 2013.
- [10] M. Laanti, O. Salo, and P. Abrahamsson, "Agile methods rapidly replacing traditional methods at nokia: A survey of opinions on agile transformation," *Information and Software Technology*, vol. 53, no. 3, pp. 276 – 290, 2011.
- [11] G. B. Alleman, *Agile Project Management Methods for ERP: How to Apply Agile Processes to Complex COTS Projects and Live to Tell about It*, pp. 70–88. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.
- [12] D. Mishra and A. Mishra, "Complex software project development: agile methods adoption," *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 23, no. 8, pp. 549–564, 2011.
- [13] Y. Li, "Erp adoption in chinese small enterprise: an exploratory case study," *Journal of Manufacturing Technology Management*, vol. 22, no. 4, pp. 489–505, 2011.
- [14] D. Maditinos, D. Chatzoudes, and C. Tsairidis, "Factors affecting erp system implementation effectiveness," *Journal of Enterprise Information Management*, vol. 25, no. 1, pp. 60–78, 2011.
- [15] O. Franoise, M. Bourgault, and R. Pellerin, "Erp implementation through critical success factors' management," *Business Process Management Journal*, vol. 15, no. 3, pp. 371–394, 2009.
- [16] A. Wong, H. Scarbrough, P. Chau, and R. Davison, "Critical failure factors in erp implementation," *PACIS 2005 Proceedings*, p. 40, 2005.
- [17] P. Gerrard, "Test methods and tools for erp implementations," in *Testing: Academic and Industrial Conference Practice and Research Techniques - MUTATION, 2007. TAICPART-MUTATION 2007*, pp. 40–46, Sept 2007.
- [18] B. G. Glaser and A. L. Strauss, *The discovery of grounded theory: Strategies for qualitative research*. Transaction Publishers, 2009.
- [19] K. Charmaz, *Constructing grounded theory*. Sage, 2014.
- [20] A. Agarwal, R. Shankar, and M. Tiwari, "Modeling the metrics of lean, agile and leagile supply chain: An anp-based approach," *European Journal of Operational Research*, vol. 173, no. 1, pp. 211 – 225, 2006.
- [21] K. Wenrich and N. Ahmad, "Lessons learned during a decade of erp experience: A case study," *International Journal of Enterprise Information Systems*, vol. 5, no. 1, p. 55, 2009.
- [22] E. J. Umble, R. R. Haft, and M. Umble, "Enterprise resource planning: Implementation procedures and critical success factors," *European Journal of Operational Research*, vol. 146, no. 2, pp. 241 – 257, 2003.

- [23] M. Al-Mashari, A. Al-Mudimigh, and M. Zairi, "Enterprise resource planning: A taxonomy of critical factors," *European Journal of Operational Research*, vol. 146, no. 2, pp. 352 – 364, 2003.
- [24] K. Bokovec, T. Damij, and T. Rajkovi?, "Evaluating {ERP} projects with multi-attribute decision support systems," *Computers in Industry*, vol. 73, pp. 93 – 104, 2015.
- [25] E. Stensrud and I. Myrteit, "Identifying high performance erp projects," *Software Engineering, IEEE Transactions on*, vol. 29, pp. 398–416, May 2003.
- [26] M. Huo, J. Verner, L. Zhu, and M. Babar, "Software quality and agile methods," in *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, pp. 520–525 vol.1, Sept 2004.
- [27] J. Itkonen, K. Rautiainen, and C. Lassenius, "Towards understanding quality assurance in agile software development," 2005.
- [28] S. Bhasin, "Quality assurance in agile: A study towards achieving excellence," in *AGILE India (AGILE INDIA), 2012*, pp. 64–67, Feb 2012.
- [29] S. M. Arachchi, S. C. Chong, and A. Madhushani, "Quality assurance and quality control in erp systems implementation," *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, vol. 11, no. 1, pp. 70–83, 2015.
- [30] S. Wiczorek, A. Stefanescu, and I. Schieferdecker, "Test data provision for erp systems," in *Software Testing, Verification, and Validation, 2008 1st International Conference on*, pp. 396–403, April 2008.
- [31] H. Malik, B. Adams, A. Hassan, P. Flora, and G. Hamann, "Using load tests to automatically compare the subsystems of a large enterprise system," in *Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual*, pp. 117–126, July 2010.
- [32] T. Asgar, M. Akour, and T. King, "Applying test-first and parallel processing techniques to erp data conversion," in *Information Technology: New Generations (ITNG), 2012 Ninth International Conference on*, pp. 269–274, April 2012.
- [33] S. Patel and R. Kollana, "Test case reuse in enterprise software implementation – an experience report," in *Software Testing, Verification and Validation (ICST), 2014 IEEE Seventh International Conference on*, pp. 99–102, March 2014.
- [34] D. Talby, A. Keren, O. Hazzan, and Y. Dubinsky, "Agile software testing in a large-scale project," *Software, IEEE*, vol. 23, pp. 30–37, July 2006.
- [35] S. Stolberg, "Enabling agile testing through continuous integration," in *Agile Conference, 2009. AGILE '09.*, pp. 369–374, Aug 2009.
- [36] R. Sangwan and P. Laplante, "Test-driven development in large projects," *IT Professional*, vol. 8, pp. 25–29, Sept 2006.