Journal of Theoretical and Applied Information Technology
15th August 2022. Vol.100. No 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SOFTWARE QUALITY ASSURANCE PERSPECTIVE ON E-COMMERCE SYSTEM DEVELOPMENT

YEE HAN CHUNG[1] , RAJERMANI THINAKARAN[1], MALATHY BATUMALAY[1], NURUL HALIMATUL ASMAK ISMAIL[2]

[1]Faculty of Data Sciences and Information Technology, INTI International University, 71800 Nilai, Negeri Sembilan, Malaysia

[2]Department of Computer Science and Information Technology, Applied College, Princess Nourah bint Abdulrahman University, Kingdom of Saudi Arabia

E-mail: [1]yeehanchung@gmail.com, [1]rajermani.thina@newinti.edu.my, [1]malathy.batumalay@newinti.edu.my , [2]NHismail@pnu.edu.sa

ABSTRACT

COVID-19 has accelerated the growth of E-commerce revenue and market size within a few months, causing the quality of E-commerce systems to be top-notch to gain a competitive edge. This study attempts to maximize feasibility and boost confidence throughout the development of e-commerce systems. This paper presents different types of software development methodologies, including traditional and agile methodologies. This study also presents related works on software development methodologies, comprising Extreme Programming, Scrum, and Kanban, and a particular software quality model, ISO/IEC 25010. Their challenges and advantages are discussed. The sub-characteristics of ISO/IEC 25010 are mapped to the features of e-commerce systems. As a result, Scrum with frequent and effective meetings can minimize technical debt, design failures, stress, miscommunication, ambiguity, improve shared vision, continuous feedback for verification, productivity, team morale, delivery predictability, project visibility, risk reduction, and engineering discipline. The ISO/IEC 25010 contributes to the success of E-commerce systems.

Keywords: *Software Development Methodology, Software Quality Model, Extreme Programming, Kanban, Scrum*

1. INTRODUCTION

COVID-19 global pandemic occurred since 2020 has signified and proved its disruption of business domestically and internationally, which may be severe for business conditions and human lives if there is no e-commerce for enterprises to generate revenue and consumer to purchase goods online [1],[10].

McKinsey reported that in February 2021, companies which adopted E-commerce for their business development experienced five-times-faster revenue growth, between 40 and 60 percent cost reductions, and 30 percent higher acquisition efficiency within sales [2]. E-commerce systems consists of various subsystems which contribute to the overall complexity and performance of the systems [2], [41]. The e-commerce systems can carry out operations, including the acquisition, selling, transfer, and exchange of services, products, and information with the involvement of computer networks and the Internet [3].

Studies have shown that the long-term success of the e-commerce system, which is widely accepted by users, is based more on the quality of the services provided, rather than the concepts of the e-commerce system, because other competitors could take the concepts [4],[5]. Thus, every organization that relies heavily relies on software development prioritizes improving the performance and quality of software [6].

According to statistics reported for online shopping in Malaysia [7], 60 percent of respondents reported that their online purchases increased tremendously compared to pre-COVID levels, and household items and packaged food promotions and sales helped shoppers with great savings. It is also reported that in this year, 2020, the Shopee e-commerce platform experienced 82 percent growth only from quarter 1 to 2 and dominates dominate the rankings because during the Covid-19 pandemic

lockdown era, Shopee made the most of consumers [8]. Compared to last year, revenue generated from the Shopee e-commerce platform alone increased by about 187.7 percent [9]. This evidence further signifies the impact on consumer, especially the convenience of e-commerce systems.

Furthermore, since the lockdown, customers of all ages have desired to purchase things online, and e-commerce makes it more convenient, accessible, and affordable for them, culminating in the loss of ancient behaviours that require customers to physically go shopping [1][10]

The aforementioned assertion is consistent with the findings of the forecast that by 2050, all commerce would be e-commerce [11]. Furthermore, when it comes to online purchasing, females are more reliant on e-reputation, commerce's whilst males are more reliant on levels of perceived trust in e-commerce [12]. As a result, one of the key variables in the e-commerce system is client trust [13].

2. LITERATURE REVIEW

2.1 Types of Software Development Methodologies

Traditional Methodology - The following is a description of the variations: Traditional development follows a top-down approach that makes any transition difficult, follows architectural leadership style, ensures pre-planning for executing subsequent phases, only requires customer involvement in the initial requirements gathering phase, builds project plan before beginning the system development process, allows only project manager to govern project ownership, follows the one-time dispatch of product discipline, and obeys mechanical requirements [16].

Agile Methodology - Leaders from diverse working backgrounds developed the Manifesto for Agile Software Development to include values and principles to optimize the method of software development [18]. The four Agile Manifesto basics are described below:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

As with the agile development approach, it makes it easier to experiment with various strategies to identify the best possible solution, enables the free flow of communication that welcomes team members' ideas, improves flexibility and adaptability of workflow for adjustment, actively engages customers to ensure success, prepares each module with the demonstration to allow the customer to receive project work incrementally with the correct direction, allows a sense of shared ownership of project contribution, follows the incremental dispatch of the product, and obeys organic structure that encourages cooperation with flexibility and participative that commonly utilized in the small and medium organization [16].

The agile method will aid crisis management during software development if certain techniques in the study [19] are followed. According to a study [20], the agile approach experience had no effect on the degree of stress, and that every new agile practitioner enjoyed the change in implementing the agile approach that follows collaborative practices, which helped practitioners avoid stress and provide a better overall project experience.

According to an annual report [21] from 2020, the top 5 reported advantages of implementing agile practices include (1) ability to manage changing priorities, (2) improve project visibility, (3) enhance business or IT alignment, (4) reduce time-to-market, and (5) improve team morale, which can be summarized as a trend for improving speed and adaptability. Moreover, the two-key metrics for assessing the success of the use of the agile methodology recent years have been consistently comprising of customer satisfaction and business value [21].

As for the challenges, a study [23] shows that from 8 papers published from 2017 onwards, the 7 most universal challenges faced during decision-making for requirements in agile development environment consist of (a) rapid evolving market trends, (b) low drive from team members in knowledge contribution for requirement engineering, (c) minimal of development time because of frequent disagreement from each stakeholder with different experiences, (d) difficulty in ascertaining requirements during the preliminary phase of development, (e) miscommunication of project goals between stakeholders and development teams, (f) dominance of High Paid Person Opinion (HiPPO) during development, and (g) unforeseeable drawbacks from customer involvement for the development team.

Furthermore, to facilitate highly complex product development, companies must scale up their agile approach. A case study [24] for a world-leading Swedish telecommunication company in Sweden, implementing a large-scale agile approach for their product management and found technical ability challenges and contextual difficulties.

Additionally, the former difficulties include poor ability to forecast due to a vast amount of content, decreased ability to prioritise because of disputes between multiple stakeholders involved, and complex planning due to vague requirements and ambiguous position of operational product owners [24]. Secondly, since the work environment is an open space where regular meetings, the contextual difficulties consist of distracting other departments [24]. Thirdly, because of the capabilities of some teams due to different opinions on team capabilities, it will be poor team build-up [24]. For example, the teams are unable to deliver what the operating product owner has promised, and the teams feel incapable of doing so, resulting in dissatisfaction and stress from both sides [24].

2.2 Related Work of Software Development Methodologies

Extreme Programming Methodology - From 1999 to 2009, Extreme Programming was the most well-known methodology for software development [25], which overlaps with the claim [26] that there were 9 cases of "XP-Distributed Team" participation during that time. Furthermore, according to a survey [6], about 80% of companies agree that a comprehensive model of software quality is crucial for the XP process and product.

Difficulties: According to a study [27], the rapid development and delivery of codebase by team-based Extreme Programming found that it has potential to be detrimental to the long-term maintenance of software quality, resulting in technical debt because there is too little time for proper design to be considered and is consistent with the results of this study [28]. It could also due to the unawareness of the context of meaning for the term technical debt, which caused the team to be thinking it as a trivial issue [27]. A study [29] states that among other characteristics, design is the most common technical debt, including test, project convention, performance, documentation, build, and security. The report [30] shows that the impact of technical debt would be harmful to quality of systems. If the practices or modifications in phases increase, the complexities of XP increase a XP is comprised of poor architectural structure with minimal documentation [28]. In addition, there is a lack of sufficient guidance provided to resolve XP's limitations [28]. The XP software development approach was also found to be the least preferred approach among the 50 respondents in the context of very small organizations [31]. The most frequent agile methods implemented in Extreme Programming (XP) during the period between 1999 and 2009 were continuous integration and pair-programming [25]. The pair-programming practice, however, poses controversial concerns and is challenging from the viewpoint of implementation [28].

Advantages: According to a study conducted [27] between 2015 and 2016, the Extreme Programming approach was found to be the second most efficient team-based development technique with the lowest effort needed for the unstructured process between Scrum, Scrum with Kanban, and Banana. Furthermore, the use of Extreme Programming followed by Scrum with Kanban, and Scrum, contributed much to user story decomposition [27]. Importantly, the frequent use of user story decomposition allows the team to minimize stress [20]. First, the outstanding functionality provided by Extreme Programming (XP) comprises of Requirement as Story Cards. Secondly, Simplicity. Thirdly, Continuous Interaction. Fourthly, Test Driven Development, and Fifthly is the Refactoring [32]. An empirical study of the impact of technical debt awareness [27][33], has found that not only Extreme Programming can provide teams with certain benefits, including improved communication and collaboration, and facilitating improvements and feedback to continually enhance the development and maintenance of system.

Scrum Methodology: From 2010 to 2016, the Scrum technique was the most often used agile methodology in all remote teams [25]. Importantly, according to an annual report [21], the Scrum approach has consistently implemented the Scrum agile methodology, with at least 75% of respondents practising Scrum and 58 percent of respondents' firms using it.

Difficulties: According to a study [33], it was not easy to observe at least 100 practitioners who worked on past projects utilizing traditional development methodology and their mindset transition to agile development methodology. Furthermore, a particular team was initially unable to meet the Sprint's timeframe, causing frustration and animosity throughout the entire team.

However, after a few Sprints, the teams began to adjust to the protocols in a logical manner, and

everything became normalised [33]. Furthermore, the conference call was packed, resulting in lengthy meetings with minimal participation from attendees, poor meeting outcomes, and little progress [34]. Furthermore, several project teams had to work together to keep track of project progress by describing the system to ensure everyone was on the same page and encouraging other teams to check the system right away [34]. Finally, because multiple teams were working on the same project and the code average was low, the implementation of units and the required end-to-end tests took an inordinate amount of time [34].

Advantages: According to a research study [35], the value of the Scrum agile methodology suggested by 136 Agile practitioners claimed that the customer collaboration, shared vision, and continuous feedback for verification were the advantages of the Scrum approach. The Scrum software development approach assisted in incrementally identifying flaws during feature delivery [34], resulting in a project with a small number of faults. The incremental delivery approach also aided the quality assurance team in assessing the project early on, allowing the developers to address any flaws [34].

Furthermore, the Scrum approach has numerous benefits, including early detection and correction of defects, reduced exposure to costly rework, increased emphasis on quality assurance activities, and participation in a sustainable pace situation [34], all of which may reduce the risk of working on the edge of the timeline. It also increases adaptability for easier handling. For example, all further requests submitted to the product backlog will be prioritised and enforced in succeeding Sprints [34].

According to a survey [34] on the effectiveness of using the Scrum approach, project team members have improved in a variety of ways, including team collaboration, team understanding, the business value of the finished product, time wasted on tasks, daily productivity, optimal effort allocation, avoidance of extremely stressful times, and better team spirit and relationships between team members. In addition, the agile Scrum strategy boosts team productivity, project predictability, and project risk reduction [21]. The aforementioned benefits, such as product reliability and delivery speed decrease, are in line with a study [22] on lead-time in the product design and development process. These findings show how agile approaches to software development can have a significant impact. In addition, the agile Scrum strategy boosts team productivity, project predictability, and project risk reduction [21]. The aforementioned benefits, such as product reliability and delivery speed decrease, are in line with a study [22] on lead-time in the product design and development process. These findings show how agile approaches to software development can have a significant impact.

Kanban Methodology According to the annual agile report [21], in 2019, 76 percent of respondent organizations (the largest percentage) are using Kanban board as their agile project management method. The workflow of software project development can be managed using Kanban boards for planning, designing, development, and delivery [36].

Empirical studies and experience reports - benefits: From the field of software engineering, there are 15 types of benefits reported [37] in 23 studies, and the studies are listed in the appendix A as P1, P2, P3, P4 and so forth. Interested readers may refer to the paper [37] for the complete version. The benefits are synthesized and described in the table below:

Table 1. Reported benefits of Kanban from empirical studies and experience reports

Category	*Number of benefits*	*Reported benefit*
Process (18 studies)	16	Poor understanding and lack of guidance for Kanban implementation
	12	Performance assessment using metrics for lead-time, and vice versa
	10	Improve identification of impediments
	7	Better workflow
	5	Reduce time-to-market
	4	Better tasks prioritization
	4	Reduction of defects
	4	Improve quality
	4	Increase intuitiveness
People (14 studies)	7	Better collaboration and communication
	6	Increase team motivation
	5	Better team building and cohesion
	6	Increase customer satisfaction
Organization (8 studies)	6	Cultivate continuous learning
	3	Better strategic cooperation

Empirical studies - challenges: According to an empirical study [37], it summarized the challenges reported from 14 studies and they are listed as P4,

P5, P6, P7, P9, P10, P12, P14, P15, P16, P17, P18, P22, P23 in the appendix A.

As shown in the table below, the 14 research indicated the problems in using Kanban in software engineering, and the 8 key challenges have been recognized and divided into three broad categories: process, people, and organization:

Table 2. Reported challenges of Kanban from empirical studies

Category	*Number of challenges*	*Reported challenge*
Process (6 studies)	1	Setting up and retaining the use of Kanban
People (8 studies)	5	Hesitant on using Kanban from Management
	4	Poor understanding of Kanban core practices and concepts
	2	Communication handling between teams and customers
Organization (11 studies)	6	Continuous adaption of evolving organizational culture
	5	Insufficient practices for Kanban
	5	Insufficient training for Kanban
	1	Poor knowledge management of Kanban
	1	Setting up and retaining the use of Kanban
	5	Hesitant on using Kanban from Management
	4	Poor understanding of Kanban core practices and concepts
	2	Communication handling between teams and customers
	6	Continuous adaption of evolving organizational culture
	5	Insufficient practices for Kanban
	5	Insufficient training for Kanban
	1	Poor knowledge management of Kanban

Experience reports - benefits: The table below is based on the reference of a study [37], and the table indicates the benefits of Kanban from 22 experience reports which are massively context specific. The experience reports were published between 2006 and 2016. Furthermore, the 22 reports are listed as ER1, ER2, ER3, ER4, and so forth in the appendix B. However, ER4 does not show any benefit in this case.

Table 3. Reported benefits of Kanban from experience reports

Category	*Number of benefits*	*Reported benefit*
Process (22 reports)	15	Assist decision-making process and facilitate visibility
	15	Improve workflow and cultivate continuous improvements
	6	Facilitate entire development process
	6	Improve predictability in the delivery of final products and greater work estimation
	4	Reduce lead time and cycle time
	2	Improve workload balance
People (8 reports)	8	Guarantee cohesiveness of teams and skills improvement
Organization (12 reports)	8	Establish self-organization and facilitate coordination
	7	Drive and support organizational change management

Experience reports - challenges: A total of 17 experience reports between 2006 and 2016 revealed the challenges related to the use of Kanban for the practitioners in this case [37]. The 17 experience reports include ER1, ER3, ER4, ER5, ER6, ER7, ER8, ER9, ER12, ER13, ER14, ER16, ER17, ER19, ER20, ER21 and ER22, and they are similarly listed in the appendix B, and the recorded challenges of the usage of Kanban is summarized in the table below:

Table 4. Reported challenges of Kanban from experience reports

Category	*Number of challenges*	*Reported challenge*
Process (5 reports)	4	Poor understanding and lack of guidance for Kanban implementation
	1	Performance assessment using metrics for lead-time, and vice versa
People (8 reports)	7	Kanban adopting motivation required for staff
	1	Unpredictable workflow and task switching
Organization (20 reports)	11	Costly, complex and time-consuming due to integration required by Kanban
	4	Continuous adaption of changing organizational culture
	4	Lack of expertise and training
	1	Deep understanding of Lean is required for Kanban

2.3 Related Work of Software Quality Model

ISO/IEC 25010: The E-commerce system is a software-intensive computer system that includes multiple stakeholders who use, procure, develop, and improve the computer system [38]. The essence of ISO/IEC 25010 is meant to be used to carry out a full specification and assessment of the efficiency and quality of software-intensive computer systems in order to assure value to stakeholders [38]. The ISO/IEC 25010 quality model can be used to describe significant quality characteristics that are helpful in achieving certain specifications' criteria, allowing high-quality software products to increase their ability to prevent negative consequences and deliver value to stakeholders [38].

According to study in 2018 [39], ISO/IEC 25000 series standards implemented in 2008 were the renewal edition of ISO/IEC 9126, and the ISO/IEC 25000 is the most well-known quality standard amongst Boehm, McCall, ISO (International Organization for Standardization), and FURPS (Functionality-Reusability-Portability-Security). Furthermore, the improvements of the quality model include compatibility, security factors, satisfactions, freedom of risk, context coverage, reusability, and restructuring capabilities that comprise interoperability and portability [39]. ISO/IEC 25010 is a more comprehensive software quality in use than ISO/IEC 9126 [40] and the readers who are interested in learning more about the history of ISO/IEC 25010 can refer to Đorđević's study [40].

The three key stages of the product life cycle that can be measured using the ISO/IEC 25010 quality model [38] are during product development, during product assessment, and throughout product use. According to the study [6], a quality product can be obtained by prioritizing the method's quality first and assessing the product's quality metrics second, using the ISO/IEC 25010 quality model.

In addition, the ISO/IEC 25010 quality model will direct the design, development, and assessment of the E-commerce system [2][41]. The system design element is also consistent with the findings of another study [42], which found that design or ergonomics was ranked as the top-quality discipline for E-commerce system by practitioners and scholars, followed by information diversity, reliability, security, ease of use, responsiveness, service quality, performance or efficiency, reputation, and vice versa. For interested readers, can refer to Stefani's other work [2][41] on ecosystem indicators for the design of quality E-commerce systems.

3. DISCUSSION

The implementation of the recommended agile development approach and software quality model, as defined above, will be discussed in the subsections that follow. In general, according to a study [43] on the framework for evaluating e-commerce websites, the e-commerce system must take into account security assurance, intuitive navigation, real-time assistant, content quality, and learnability.

3.1 Adopting Scrum Methodology

Scrum methods, including stand-up meetings, backlog, and Sprint or iteration, are the most effective agile practises in the Scrum methodology in all situations, according to a literature analysis [25] on agile practises in global software development. Successful meetings and predictable delivery, which the Scrum methodology enabled, also helped the project team to reduce stress [20]. According to the same annual agile report [21], 1,121 complete survey responses show that the top 5 agile strategies embraced are agile methods such as (1) Daily Stand-Ups, (2) Retrospective, (3) Sprint Planning, (4) Sprint Review, and (5) Short Iterations, and their percentages are 85, 81, 79, 77, and 64 respectively.

Scrum's agile tactics will benefit the project team in a variety of ways, including all Scrum meetings save the Sprint Retrospective, which will improve planning and Sprint Review. Additionally, Stand-Ups can be positively impacted by communication and transparency, while Sprint Retrospective facilitates the transmission of information and input of feedback. Furthermore, Continuous Integration reduces the product time-to-market and increases transparency of the product. Scrum-of-Scrums improves the aspects of communication, including planning, structuredness, and transparency [44]. Scrum roles also will increase the empowerment and satisfaction of the team [44]. The Scrum strategics that should be followed for the growth of e-commerce system are outlined:

Scrum-of-Scrums: This is a weekly meeting with a large team or a project lead to ensure that the team is on track to complete the project successfully.

Daily Stand-Ups: This is a weekly small group gathering where everyone can catch up on work progress. It can be divided into three categories:

(1) only for the purpose of status monitoring,

(2) an hour-long meeting, and

(3) an unprecedented decline in the frequency of the meeting, but with the effect that it will be a long meeting.

Sprint planning takes place during the Sprint shift. All departments will be given a breakdown of the backlog, and each team will be accountable for its own preparation.

Sprint Review: This is a meeting with team leaders and representatives to discuss not only work progress but also what has been accomplished. Any of the evaluation sessions might be casual and unstructured if there is no agenda.

Retrospective: This is a meeting that will vary in length depending on the time. At the meeting, the teams review their progress and discuss how to overcome any obstacles. If the goal is clearly established, it may be beneficial to carry it out within each team.

Backlog: It's akin to a to-do list for objects. The items defined in the Backlog can be changed by priority, which is dependent on the current situation. Each team or project phase may have many backlog items to attend to.

3.2 Adopting ISO/IEC 25010 Quality Model

As shown in the table, the quality goals identified in the ISO/IEC 25010 quality model are mapped with the sub-characteristics and recommendations for improving the production of modern e-commerce systems. To reinforce the discussion, the initial report of ISO/IEC 25010:2011 by ISO is used [38].

Table 5. Mapping of quality goals, sub-characteristics, and recommendations

Quality Goal	*Sub-characteristic*	*Recommendation*
Usability	Appropriateness recognizability	System or product appropriateness
	User-interface aesthetics	Satisfying design and interaction
	Learnability	Simplicity, effectiveness, efficiency, freedom of risk, and satisfaction
	Operability	Ease of operate, control, simplicity, efficiency
	User error protection	Constraints against errors
Functional suitability	Functional completeness	Focus of specified tasks, and user objectives
	Functional correctness	Correct results with precision
	Functional appropriateness	Facilitation of tasks accomplishment
Performance efficiency	Time behaviour	Throughput rate of system
	Capacity	Capacity limit of system
Reliability	Maturity	Reliability under normal operation
	Availability	Constant operation, and accessible when required
	Fault tolerance	Operate in the presence of faulty
	Recoverability	Re-establish desired state in the event of failure(s)
Security	Confidentiality	Data accessibility, and privacy concerns
	Integrity	Modification rights
	Non-repudiation	Proves of events, and transaction security
	Authenticity	Genuine products, and services
Maintainability	Modularity	Discrete components, and minimal impact on changes
	Analysability	Failure diagnosis, and impact assessment
	Modifiability	Stability, changeability, and minimal impact on modification
	Testability	Test criteria establishment, and test case for test criteria established

Discussion on the mapping of each of the sub-characteristics following their quality goals, and recommendations is presented below:

Usability Appropriateness recognizability: Appropriateness of the system or products for needs [38]. It determines whether customers can find products to fulfil their demands when using an e-commerce system. The higher the recognizability, the better the e-commerce system's conversion rate, and hence the more money the e-commerce platform may make. Customers would spend less effort and time locating their perfect products by minimising product selections in the e-commerce system, according to a research study [38]. It also decreases confusion and cognitive strain among customers.

User-interface aesthetics: Pleasant and satisfying interaction [38]. An e-commerce system's user interface can have a negative impact on the overall user experience, especially in terms of perceived customer information overload [45]. The manager and product designer play a crucial role in the aestheticism of user-interface design, and the design

of an E-commerce system must be approached from psychological, business, and technological viewpoints [2][41]. The non-ambiguous design of user interfaces will certainly enhance the motivation of customers towards the e-commerce system [45]. In addition, research studies [29][46][47] argue that design problems were a major factor in the reason for system failures.

Learnability: Ease of use and effectiveness [38]. End-users' ease of understanding, particularly during their initial visit to the e-commerce system. The more user-friendly an e-commerce system is, the easier it is to learn and the less cognitive burden it requires. Information ambiguity has a bigger impact on user motivations for utilising the e-commerce system than does information complexity [45]. Similarly, quick access to information coincides with user engagement and can have a substantial impact on consumer happiness and E-commerce performance [13]. The E-commerce system's poor design increases the number of steps required to achieve consumer goals and, as a result, diminishes its learnability. As a result, it's critical to keep the proportion of total steps to fulfil an objective versus job completion as low as possible [48]. Color palettes, for example, would also contribute to efficacy [48].

Operability: Error tolerance, controllability, and conformance to user expectations are all factors that affect operability and simplicity of operation and management [38]. The ease of use of online stores will positively influence customers' attitudes toward e-commerce, and customers who have a good user experience are more likely to offer the E-commerce system higher feedback ratings [49]. The attributes to be met include the speed of page loading to provide customers with a smooth experience because slow page response times are the key contributor to page abandonment and decrease sales order in online shops [15], and the slow access to content in the e-commerce portal would increase customers' perceptions of information overload [45].

User error protection: Customers should be protected from making mistakes [38]. The e-commerce system should issue smart validation. Customers are asked with a validation notice when they check out or delete an item from their shopping cart, for example. Additionally, if they try to cancel an order during the checkout, a pre-emptive warning should be shown to the customers.. Furthermore, to verify the input of typing characters, such as registration details, and payment details, regular expressions may be implemented to validate them.

Functional suitability: Functional appropriateness was the most generally stated quality target in software projects, according to 50 participants in a poll [31] done in the setting of extremely small firms. Functional appropriateness is defined by three sub-characteristics: functional completeness, functional correctness, and functional adequacy, all of which are explored further down.

Functional completeness: Set of functions covering all the specified tasks and goals of the user [38]. A collection of functions introduced in the e-commerce system are capable to allow users to accomplish their objectives. From the perspective of experts of the e-commerce system [50], the key features of the e-commerce system include the presentation of overviews of products and services, product filtering, product analysis, product and service search, integration with other applications, purchasing of goods and services, and order status monitoring.

Functional correctness: Computational results that are accurate and correct [38]. The e-commerce system must produce dependable results, and this goal is linked to the maturity sub-characteristic of the level of reliability goal, which is also addressed in this research. An invoice might include the buy products, delivery date, product names, unit cost of the product, and gross pay for purchase products, for example. In addition, to produce an order report for the supplier to restock low-quantity items, tracking the inventory of each item for the system administrator is often important to be reliable. For decision-making by the stakeholders, monitoring of system performance should also be accurate. Furthermore, the goods in the e-commerce system should be properly organized and precisely according to their categories.

Functional appropriateness: Facilitate established tasks and user objectives [38]. The e-commerce system is only required to provide clients with the instructions they need to complete specific tasks, without adding any unnecessary steps to their time. According to a study [45], system interface issues such as confused instructions given to customers in the e-commerce environment to reach their aims would increase information overload and have a negative impact on customers' inclinations to utilise the e-commerce platform. In another survey [31] carried out in the context of very small organizations, 49 out of 50 respondents indicated that the most common sub-characteristic considered in software projects was functional appropriateness, which is part of the objective of functional suitability.

Functional efficiency Time behavior: Degree of system throughput rate [38]. The throughput rate is similar to performance, precisely the e-commerce system's processing time while executing a function to accomplish users' defined tasks. This system, for example, needs less memory and runs rapidly. Terms such as memory and speed are categorized as efficiency characteristics [48].

Capacity: Maximum product limit or system parameter of the e-commerce system [38]. The higher the capacity limit, the greater the efforts required by developers for the implementation of the e-commerce infrastructure. For instance, the parameters will consist of the number of concurrent users, the size of database, transaction throughput, and the bandwidth of communication [38]. Furthermore, the restricted capacity that may be severely influenced by media elements may substantially reduce the e-commerce system's ability to efficiently generate a list of product offerings when the website loading time slows, and this problem might impact e-commerce revenues due to page abandonment [15]. Time-to-market and maintenance costs will suffer if the capacity limit is set too high [15]. Importantly, in search results, the large page size that causes slow page loading speed may have a substantial effect on a website's ranking [51].

Reliability Maturity: Under typical conditions, it is dependable [38]. Events like issuing invoices for customers, order reports for suppliers, and system performance monitoring for stakeholders can all be done in the normal course of business. A study [52] demonstrates the importance of reliability in producing highly reliable outcomes. However, the correctness of e-commerce transactions producing outputs is a difficult task to achieve, as there have been many cases where errors in the e-commerce system have caused system downtime. However, the method of applying logic-based formal specification techniques is widely assumed to be capable of creating a coherent and effective e-commerce system [52].

Availability: Accessible and operational when necessary [38]. Maturity, recoverability, and fault tolerance are part of the availability factor [38]. The e-commerce system's design activities, such as behavioural, architectural, and interface designs, may play a significant role in poor design outcomes, such as limited availability [15]. Furthermore, the availability aspect must examine the high cohesion and low coupling to incorporate modularized designs that will aid in future important adjustments to the E-commerce system [15]. During the upstate process, the availability of the e-commerce system will be checked, and the system is expected to be in continuous service for customers to access and purchase products [15]. As this availability attribute is aligned with the next sub-characteristic, the fault tolerance, a related real use case scenario for e-commerce is further addressed.

Fault tolerance: In the event of software or hardware failures, the system will continue to function as planned [38]. Microservice designs can improve the fault tolerance and high availability of an e-commerce system [53]. In comparison, one of the leading European e-commerce platforms, otto.de, and Amazon are presently implementing microservice designs, facilitating non-functional features such as scalability, dependability, and agility of their e-commerce platforms [53].

Recoverability: In the case of a failure or disruption, re-establishment of the ideal state [38]. Following the above fault tolerance, it is possible to tolerate the failure of individual services in the e-commerce systems to restore services after faults have been detected using the microservices [53]. Therefore, in the case of a malfunction, the desired state of the e-commerce system can be easily restored and improves recoverability.

Security: It comprises multiple dimensions for the security objective of e-commerce systems, including safety, repudiation, integrity, authenticity, confidentiality, privacy, availability [54]. To ensure the success of an e-commerce system, more consideration must be paid to usability, privacy, and security factors [55] since the security of personal information would lead to influencing customer trust [13]. As regards the additional resources needed for formal specification methods in e-commerce transactions, the resources [52][56][57] can be referred to by interested practitioners and developers.

Confidentiality: Data can only be accessed by authorized users [38]. Upon completion of registration, each of the customers can connect to the e-commerce system with a dedicated e-mail and password. Some features, such as purchasing history and account configuration, are only available when it is accepted into the e-commerce system. Only the system administrator, on the other hand, has access to the e-commerce system's inventory in order to make necessary adjustments and generate order reports and system monitoring reports. According to a study [58], the impact of online privacy concerns (OPC) on customers' behaviour toward purchasing things online would be directly influenced since

customers would expect to have more control over their personal information. It is also necessary to protect the information of the customer in the e-commerce environment to prevent any personal information fabrication, which can lead to the reluctance of customers to shop online

Integrity: Privileges to access or modify a system [38]. Unauthorized users are prohibited from accessing or changing the e-commerce system in terms of programmes or records. Its purpose is to prevent any malicious intent from disrupting the e-commerce infrastructure. For example, transactions can be safeguarded to ensure data integrity in the e-commerce system [15].

Non-repudiation: For reassurance, provide proof of happenings [38]. Non-repudiation can be defined as an indisputable transaction and conduct in e-commerce transactions from customers, which occurs in blockchain [59]. Non-repudiation services' objectives are to gather, provide, maintain, and verify transaction evidence from the transmitter to the receiver [59]. To ensure information non-repudiation, all information submitted or retrieved by all clients is incontrovertible. As a result, in order to strengthen confidence between the platform and customers, the non-repudiation service will provide increased security for e-commerce transactions. As evidence, all records are maintained protected. With these, the e-commerce system can enforce more policies to promote stakeholder confidence, such as goods sold are exchangeable but not refundable unless they are spoiled products.

Authenticity: Genuine products and services are available [38]. The e-commerce system must demonstrate that the commodities displayed in the system are genuine in order to exhibit professionalism and promote responsibility. The rules and regulations must be properly set in order for customers to reach mutual agreement and avoid potential litigation and misunderstandings.

Maintainability: In order to keep up with the constant evolving needs of consumers and to compete with other competitors, the e-commerce infrastructure needs to be continually updated and maintained. A series of innovative techniques are necessary to improve software maintainability, adaptability, and scalability [15].

Modularity: Modifications have small effects on discrete components [38]. The e-commerce system's architecture is made up of separate pieces. Any changes to the constituent components should have the least impact feasible. Modularity can be achieved by using Declarative and Component Oriented libraries like React.js or Responsive Frameworks like Angular.js. Modularity contributes to the component-based software engineering paradigm by allowing practitioners and developers to be more agile by relying on reusable components or source codes that may be safely stored for version control on GitHub [60]. These practices have important benefits, including improving performance and cost-effective during the software development.

Analysability: Failure diagnosis and impact assessment [38]. To evaluate the system faults, mechanisms can be implemented into the e-commerce system. In addition, prior to any incident or malfunction, system report may be produced to improve analysability and prevent critical risks, such as unexpected system failures.

Modifiability: Minimal modification impact [38]. It is a measure of modifying the e-commerce system without degrading the system quality or adding flaws to the system. Modularity and analysability can influence modifiability when the system requires a sequence of actions, including coding, designing. documentation, and verification of modifications [38]. In addition, modifiability is composed of stability and changeability [38].

Testability: The establishment of test criteria that requires test case verification [38]. To ensure the safe and secure functionalities of the e-commerce system, the clearly defined test criteria for the e-commerce system need to be tested in test cases. Furthermore, advanced analysis and design following the definition of designs for the e-commerce system to cover factors, such as performance, security, availability, safety, maintainability, and usability requirements are necessary to increase the testability [15].

4. RESULT

According to a quantitative review of the differences between traditional and agile architecture approaches, the agile approach wins out and is consistent with e-commerce growth. The simplified complexities and benefits of software development approaches like as Extreme Programming, Scrum, and Kanban highlight the Scrum methodology as the best ideal strategy for the creation of an e-commerce system. It does not, however, ensure its efficacy while using any particular agile methodology.

After discussing all the recent related works, Scrum software development approach has the benefits of mitigating technical debt, possible design failures, stress, miscommunication, and ambiguity and enhancing shared vision, constant feedback for

verification, productivity, team cohesion, delivery predictability, project visibility, risk reduction, and engineering discipline. Regarding the implementation of the ISO/IEC 25010 software quality model described in the next section, its sub-characteristics are constructively mapped with the main features of the e-commerce system and contribute to the overall progress of the modern e-commerce system development life cycle.

As a result, the above-mentioned software development strategy and software quality model should enable practitioners and developers to produce usable results and comprehend the methodologies discussed. We will discuss the adoption of the Scrum software development approach, as well as its proper execution in accordance with recognised agile practises and research findings. Second, we'll go over how to put the ISO/IEC 25010 quality model into practise, including its quality goals, sub-characteristics, and guidelines for the development of modern e-commerce systems.

5. THREAT TO VALIDITY

In this study, because of the complexities, there is no quantitative method applied to analyze all the decisions or recommendations for the implementation phases of e-commerce systems. Given the significance of this analysis to the e-development of commerce systems, its contribution will not be overlooked. Isolating software from the reality in which it operates, for example, frequently leads to deceptive qualitative evaluations, and this is a common mistake [61].

In addition, during the development of the e-commerce systems, we did not implement the strategies provided to verify the enhancement of the implementation feasibility and the confidence of practitioners and developers. However, in this study, it could be done to assess the above-mentioned criteria by creating a prototype e-commerce platform following the recommended software development methodology and software quality model.

6. CONCLUSION

Observing the differences between traditional and agile software development approaches. The related works and actual use cases of agile software development methodologies such as Extreme Programming, Scrum, and Kanban in terms of their challenges and benefits, and came to the conclusion that the Scrum software development methodology would be the best fit for developing an e-commerce system.

Furthermore, the beneficial impacts of implementing Scrum strategies to improve the feasibility and confidence of practitioners and developers during the system development process. Reviews shows that the related works of the ISO/IEC 25010 quality model to indicate its usefulness in the development of e-commerce systems.

Consequently, to further enhance feasibility and usefulness, the quality goals, sub-characteristics, and recommendations of the ISO/IEC 25010 quality model need to be mapped for the development of modern e-commerce systems. Theoretically, this study contributes to the advancement of e-commerce systems and tries to encourage the future implementation of high-quality e-commerce systems.

7. FUTURE RESEARCH

The evaluation of the implementation of the chosen software development approach and software quality model for the development of the e-commerce system requires further work. Firstly, use a quantitative method, considering the complexities, to avoid developing e-commerce systems that are solely based on intuitive parameters. Secondly, design an experiment to consider other variables and analyse the variables that have an immediate impact on the behaviour of the customer using an e-commerce prototype system.

REFERENCES:

[1] J. Sheth, “Business of business is more than business: managing during the Covid crisis,” Indus. Mark. Manag., vol. 88, pp. 261-264, July 2020.

[2] A. Stefani, “A metrics ecosystem for designing quality e-commerce systems,” Inter. Jou. of Comp. Sci. & Info. Tech., vol. 10, no. 2, April 2018.

[3] R. K. Rainer, and C. G. Cegielski, Introduction to Information Systems: Enabling and Transforming Business, 3rd ed. John Wiley & Sons, 2011.

[4] E. Turban, D. King, J. K. Lee, T. P. Liang, and D. C. Turban, Electronic Commerce A Managerial and Social Networks Perspective, Springer, 2015.

[5] A. Goldfarb, S. M. Greenstein, and C. E. Tucker, “Economic analysis of the digital economy,” Uni. of Chi. Press, the Nat. Bur. of Eco. Res., 2015.

[6] A. Tabassum, S. N. Bhatti, A. R. Asghar, I. Manzoor, and I. Alam, “Optimized quality model for agile development Extreme

Programming (XP) as a case scenario," Inter. Jou. of Adv. Comp. Sci. and App., vol. 8, no. 4, 2017.

[7] L. Benedict, "Covid-19's impact on Malaysia's ecommerce market," unpublished.

[8] R. Sachitanand, "Shopee tops Singapore ecommerce market, Zalora struggles, Amazon's shadow looms," unpublished.

[9] Businesswire, "Sea limited reports second quarter 2020 results," unpublished.

[10] J. Sheth, "Impact of Covid-19 on consumer behavior: will the old habits return or die?,"Jour. of Busi. Res., vol. 117, pp. 280-283, September 2020.

[11] C. G. Traver, and K. C. Laudon, E-Commerce: Business, Technology, Society, 6th ed., Boston: Addison Wesley, 2002.

[12] P. Oghazi, S. Karlsson, D. Hellström, R. Mostaghel, and S. Sattari, "From Mars to Venus: alteration of trust and reputation in online shopping," Jour. of Inno. & Know., in press.

[13] M. Choshin, and A. Ghaffari, "An investigation of the impact of effective factors on the success of e-commerce in small and medium-sized companies," Comp. in Hum. Behav., vol. 66, pp. 67-74, January 2017.

[14] V. Gatta, E. Marcucci, M. L. Pira, G. Inturri, M. Ignaccolo, and A. Pluchino, "E-groceries and urban freight: investigating purchasing habits, peer influence and behaviour change via a discrete choice/agent-based modelling approach," Trans. Res. Proc., vol. 46, pp. 133-140, 2020.

[15] S. A. Ehikioya, and E. Guillemot, "A critical assessment of the design issues in e-commerce system development," Engi. Report, vol. 2, no. 4, p. e12154, 2020.

[16] C. Tam, E. J. Moura, T. Oliveira, and J. Varajão, "The factors influencing the success of on-going agile software development projects," Int. Jour. of Proj. Manag., vol. 38, no. 3, pp. 165-176, April 2020.

[17] J. M. Ríos, and N. P. Souto, "Comparison of development methodologies in web applications," Info. and Soft. Tech., vol. 119, p. 106238, March 2020.

[18] K. Beck, J. Sutherland, M. Fowler, K. Schwaber, A. Cockburn, A. V. Bennekum, M. Beedle, J. Highsmith, R. C. Martin, R. Jeffries, W. Cunningham, J. Kern, D. Thomas, J. W. Grenning, B. Marick, A. Hunt, and S. J. Mellor, "Manifesto for Agile Software Development," February 2001.

[19] S. V. Zykov, "IT crisisology: the new discipline for managing software development in crisis," Proc. Comp. Sci., vol. 159, pp. 1777-1786, 2019.

[20] A. Meier, M. Kropp, C. Anslow, and R. Biddle, "Stress in agile software development: practices and outcomes," Agile Proc. in Soft. Engi. and Ext. Prog., vol. 314, pp. 259-266, May 2018.

[21] VersionOne, "14th annual state of agile report," May 2020.

[22] P. Joshi, A. Akbari, and R. Berntsson-Svensson, "Impact of usability on process lead-time in information systems: a case study," Jour. of Syst. and Soft., vol. 148, pp. 148-169, February 2019.

[23] R. P. Ghozali, H. Saputra, M. A. Nuriawan, Suharjito, D. N. Utama, and A. Nugroho, "Systematic literature review on decision-making of requirement engineering from agile software development," Proc. Comp. Sci., vol. 157, pp. 274-281, 2019.

[24] F. Evbota, E. Knauss, and A. Sandberg, "Scaling up the planning game: collaboration challenges in large-scale agile product development," Agile Proc., in Soft. Engi., and Ext. Prog., pp. 28-38, May 2016.

[25] R. Vallon, B. J. S. Estácio, R. Prikladnicki, and T. Grechenig, "Systematic literature review on agile practices in global software development," Info. and Soft. Tech., vol. 96, pp. 161-180, April 2018.

[26] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods: review and analysis," Cor. Uni., September 2017.

[27] D. Taibi, V. Lenarduzzi, A. Janes, K. Liukkunen, and M. O. Ahmad, "Comparing requirements decomposition within the scrum, scrum with kanban, xp, and banana development processes," Agile Proc. in Soft. Engi. and Ext. Prog., vol. 283, pp. 68-83, April 2017.

[28] F. Anwer, and S. Aftab, "SXP: simplified extreme programming process model," Int. Jour. of Mod. Edu. and Comp. Sci., vol. 9, no. 6, June 2017.

[29] M. C. O. Silva, M. T. Valente, and R. Terra, "Does technical debt lead to the rejection of pull requests?," vol. 1, pp. 248-254, May 2016.

[30] P. Avgeriou, P. Kruchten, I. Ozkaya, and C. Seaman, "Managing technical debt in software

engineering (Dagstuhl Seminar 16162)," Dags. Rep., vol. 6, no. 4, pp. 110–138, 2016.

[31] G. A. García-Mireles, "Identifying relevant product quality characteristics in the context of very small organizations," Comp. Sci. and Info. Syst., vol. 13, pp. 875-900, 2016.

[32] G. S. Matharu, A. Mishra, H. Singh, and P. Upadhyay, "Empirical study of agile software development methodologies: a comparative analysis," ACM SIGSOFT Soft. Engi. Notes, vol. 40, no. 1, pp. 1-6, Feburary 2015.

[33] G. S. Tonin, A. Goldman, C. Seaman, and D. Pina, "Effects of technical debt awareness: a classroom study," Agile Proc. in Soft. Engi. and Ext. Prog., vol. 283, pp. 84-100, April 2017.

[34] G. Papadopoulos, "Moving from traditional to agile software development methodologies also on large distributed projects," Proc. Social and Beha. Sci., vol. 175, pp. 455-463, February 2015.

[35] M. Ochodek, and S. Kopczyńska, "Perceived importance of agile requirements engineering practices–a survey," Jour. of Syst. and Soft., vol. 143, pp. 29-43, September 2018.

[36] E. Schön, D. Winter, J. Uhlenbrok, M. J. Escalona, and J. Thomaschewski, "Enterprise experience into the integration of human-centered design and kanban," Proc. of the 11th Inter. Joint Conf. on Soft. Tech. (ICSOFT 2016), vol. 1, pp. 133-140, 2016.

[37] O. A. Muhammad, D. Dennehy, K. Conboy, and M. Oivo, "Kanban in software engineering: a systematic mapping study," Jour. of Syst. and Soft., vol. 137, pp. 96-113, March 2018.

[38] International Organization for Standardization (ISO), "Systems and software engineering - systems and software quality requirements and evaluation (SQuaRE) - System and software quality models," ISO/IEC 25010:2011, 2011.

[39] R. Hassan, and S. H. Mohammad, "Software quality models: a comprehensive review and analysis," Jour. of Eletr. and Comp. Engi. Inno., vol. 6, no. 1, pp. 59-76, 2018.

[40] N. D. Đorđević, "Software quality standards," Vojno. Glas., vol. 65, no. 1, pp. 102-124, January 2017.

[41] A. Stefani, "On the design of effective e-commerce applications: an ISO-based lifecycle model," Jour. of Comp. and Comm., vol. 6, no. 5, pp. 15-30, May 2018.

[42] H. Jegham, and S. Ghannouchi, "Proposal of a computer supported collaborative work model for e-commerce web sites based on quality guiding framework," Proc. Inter. Conf. on Soft. Engi., November 2017.

[43] V. Akriti, and K. S. Aman, "Framework for evaluation of e-commerce websites," Inter. Jour. of Sci. Res. Engi. & Tech., vol. 6, no. 9, September 2017.

[44] P. Diebold, and U. Mayer, "On the usage and benefits of agile methods & practices," Agile Proc. in Soft. Engi. and Ext. Prog., vol. 283, pp. 243-250, April 2017.

[45] C. Y. Li, "Why do online consumers experience information overload? An extension of communication theory," Jour. of Info. Sci., vol. 43, no. 6, pp. 835-851, October 2016.

[46] L. Sousa, A. Oliveira, W. Oizumi, S. Barbosa, A. Garcia, J. Lee, M. Kalinowski, R. d. Mello, B. Fonseca, R. Oliveira, C. Lucena, and R. Paes, "Identifying design problems in the source code: a grounded theory," Proc. of the 40th Inter. Conf. on Soft. Engi., pp. 921-931, May 2018.

[47] L. Xiao, Y. F. Cai, R. Kazman, R. Mo, and Q. Feng, "Identifying and quantifying architectural debt," 2016 IEEE/ACM 38th Inter. Conf. on Soft. Engi. (ICSE), pp. 488-498, May 2016.

[48] I. Atoum, "A novel framework for measuring software quality-in-use based on semantic similarity and sentiment analysis of software reviews," Jour. of King Saud Uni. - Comp. and Info. Sci., vol. 32, no. 1, pp. 113-125, January 2020.

[49] G. Agag, and A. A. El-Masry, "Understanding the determinants of hotel booking intentions and moderating role of habit," Inter. Jour. of Hosp. Manag., vol. 54, pp. 52-67, April 2016.

[50] S. Sulova, "A system for e-commerce website evaluation," Proc. 19th Inter. Multi. Sci. Geo. SGEM 2019, vol. 19, pp. 25-32, 2019.

[51] C. Ziakis, M. Vlachopoulou, T. Kyrkoudis, and M. Karagkiozidou, "Important factors for improving Google search rank," Future Inter. 2019, vol. 11, no. 2, p. 32, January 2019.

[52] S. A. Ehikioya, and A. A. Olukunle, "A formal model of distributed security for electronic commerce transactions systems," Inter. Jour. of Net. and Dist. Comp., vol. 7, no. 2, pp. 68-84, April 2019.

[53] W. Hasselbring, and G. Steinacker, "Microservice architectures for scalability, agility and reliability in e-commerce," 2017 IEEE Inter. Conf. on Soft. Archi. Work. (ICSAW), pp. 243-246, June 2017.

[54] S. P. Patro, N. Padhy, and R. Panigrahi, "Security issues over e-commerce and their solutions," Inter. Jour. of Adv. Res. in Comp. and Comm. Engi., vol. 5, no. 12, pp. 81-85, December 2016.

[55] M. Teltzrow, and A. Kobsa, "Impacts of user privacy preferences on personalized systems," Desig. Pers. User Exp. in eCom., vol. 5, pp. 315-332, 2004.

[56] C. Zhang, X. Li, Z. Feng, and J. Song, "Formal modelling and analysis of fairness characterization of e-commerce protocols," Jour. of App. Math., vol. 2014, pp. 1-10, 2014.

[57] S. A. Ehikioya, and B. Ola, "A comparative study of specification methods for electronic commerce systems," The 3rd ACS/IEEE Inter. Conf. on Comp. Syst. and App., June 2005.

[58] I. Anic, V. Škare, and I. K. Milaković, "The determinants and effects of online privacy concerns in the context of e-commerce," Elec. Comm. Res. and App., vol. 36, p. 100868, 2019.

[59] W. Fang, W. Chen, W. Zhang, J. Pei, W. Gao, and G. Wang, "Digital signature scheme for information non-repudiation in blockchain: a state of the art review," EURASIP Jour. on Wire. Comm. and Net., March 2020.

[60] M. D. Papamichail, T. Diamantopoulos, and A. L. Symeonidis, "Measuring the reusability of software components using static analysis metrics and reuse rate information," Jour. of Syst. and Soft., vol. 158, p. 104687, December 2019.

[61] H. Nakai, N. Tsuda, K. Honda, H. Washizaki, and Y. Fukazawa, "A SQuaRE-based software quality evaluation framework and its case study," 2016 IEEE Reg. 10 Conf. (TENCON), pp. 3704-3707, 2016.