CROWDSOURCING IN SOFTWARE TESTING: OPPORTUNITIES FOR SOFTWARE COMPANIES

by

Paramjit Singh Aujla

DISSERTATION

Presented to the Swiss School of Business and Management Geneva

In Partial Fulfillment

Of the Requirements

For the Degree

DOCTOR OF BUSINESS ADMINISTRATION

SWISS SCHOOL OF BUSINESS AND MANAGEMENT GENEVA

<MONTH OF GRADUATION, YEAR>

CROWDSOURCING IN SOFTWARE TESTING

by

Paramjit Singh Aujla

APPROVED BY

____

Chair

RECEIVED/APPROVED BY:

____

SSBM Representative

Dedication

Dedicated to the pioneers of crowdsourcing, who showed us that innovation can bloom from the collective power of the open web. This work is for the next generation of developers and testers, empowered by the boundless possibilities of collaboration in the digital age.

Acknowledgements

I extend my heartfelt gratitude to all those who contributed to the realization of this dissertation on the dynamic field of crowdsourcing in software testing.

I am deeply thankful to the participants of the crowdsourced testing initiatives whose valuable insights and dedicated efforts have played a pivotal role in shaping this research. Your willingness to engage in this collaborative endeavour has been instrumental in shedding light on the complexities and possibilities within the realm of software testing.

I express my sincere gratitude to my academic advisor Dr. Mario Silic for their guidance, unwavering support, and invaluable feedback throughout the research process. Your expertise and mentorship have been instrumental in refining the focus and methodology of this dissertation.

I also acknowledge the SSBM community for fostering an environment conducive to research and providing resources essential for the completion of this study.

Lastly, I want to thank my parent, wife, daughter and friends for their encouragement and understanding during the demanding phases of this academic journey. Your belief in my capabilities has been a constant source of motivation.

This dissertation is a testament to the collaborative spirit of crowdsourcing and the collective efforts of those who believe in the power of diverse perspectives in advancing the field of software testing.

ABSTRACT
USING CROWDSOURCING FOR SOFTWARE TESTING

Paramjit Singh Aujla
2021

Dissertation Chair: Jaka Vadnjal
Co-Chair: Anna Provodnikova

This research is being conducted to explore and understand how effectively crowdsourcing techniques are utilised in the Information Technology industry today to build software products. The core question posed is: What are the business strategies and practices of IT Product development companies to build high-quality software products at a low cost?? The goal is to analyse data from various industry papers, organisations and products in the Information Technology industry and formally define the crowdsourcing techniques and practices beneficial for software product development.

The research utilized a qualitative approach, using an online interview questionnaire and semi-structured interviews. The interview questionnaire was shared with a wider audience working in the IT Industry with specialization skills in software testing, or development over social media platforms like LinkedIn.com. The Semi-structured interviews were conducted through using channels such as Zoom, and face-to-face interactions. The study focused on the software project experiences of various individuals having experience in the IT Industry, ranging from Testers to mid-senior and executive positions. All interviewees were carefully selected from the global locations to learn the variety of experiences and build an understanding of the usage of Crowdsourcing in Software Testing. Data were collected through interview questionnaires, interviews and web data scrapping. Abductive reasoning methods were employed in exploring Crowdsourcing

awareness, such as the adoption and use of Crowdsourcing, the impact of Crowdsourcing on individuals and organizations, and the emergence of testing strategies based on Crowdsourcing.

The study revealed that Crowdsourcing for software testing is still a new field for many people. Software companies still prefer direct vendor contracts or full-time people to get the job done. It is observed that key concerns are falling in the areas of security, team management and infrastructure. Based on their experiences, crowdsourcing will evolve as a field: With the increasing demand for software products and services, the adoption of crowdsourced testing is expected to grow significantly. More companies are likely to embrace crowdsourced testing as a viable alternative to traditional testing methods, especially for large-scale or complex projects., Advanced testing tools: Crowdsourced testing platforms are likely to incorporate advanced testing tools and technologies such as AI, machine learning, and automation to improve testing accuracy and efficiency., Improved quality control: Crowdsourced testing platforms are likely to implement better quality control mechanisms to ensure that testing is conducted accurately and thoroughly. This could include improved tester selection processes, better testing guidelines, and more comprehensive reporting and feedback mechanisms.

TABLE OF CONTENTS

LIST OF TABLES

List of Figures

CHAPTER I:

INTRODUCTION

This section aims to give a comprehensive introduction to the topic of the study, highlighting its background and key points, and emphasizing the significance of the research. It will also outline the motivation and expected outcomes of the research, providing a rationale for why the study is necessary and must be conducted. In addition, the section will present the problem statement that serves as the basis for the proposed research questions. Finally, the section will discuss how this study will contribute to the existing literature by addressing a knowledge gap.

1.1 Research Background and Scope

This study examines the strategies and usage of Crowdsourcing in the field of Software Testing and how these insights could be relevant to other companies in the same industry, regardless of their size and scope. The findings of this research offer practical insights, implications and future growth that can help businesses mitigate risks and address challenges, thereby increasing the likelihood of success in the future.

Based on the data collection across experienced people working in various roles within the Information Technology industry, including junior, mid to senior, Executive and founder positions, the data suggests that awareness of Crowdsourcing is still in nascent stages and yet to reach the long-term slope of sustainable growth. Organisations are reluctant to utilise Crowdsourcing platforms primarily due to Security and operations aspects. Furthermore, as more Crowdsourcing products enter this space, create more innovative platforms addressing the challenges and Crowdsourcing becomes more widespread, it will become an integral part of the Software Product development.

Software development is a complex process. Today companies are still struggling with production bugs, which are expensive to fix and dent the organisation's reputation in the market. In today's connected world, software validation & testing have become even more challenging. A single piece of software might mean various inter-connected technologies like Cloud, Omni-channel, big data, or BI. Validation of such a wide array of technologies becomes very challenging & cumbersome (Sathe, A. and Kulkarni, R., 2013). The objective is to analyse crowdsource techniques and areas of application for software testing which can help organisations lower the cost of quality, increase efficiencies, lower defect seepage ratio and Increase the return on investment."

Since (Howe, J., 2006) invented the term "crowdsourcing" in 2006, it has quickly become a mainstream innovation avenue for businesses. Pierre Levy mentioned in his book on Collective Intelligence that "No one knows everything, but everyone knows something, and mankind has all knowledge." (Pierre Levy, 2010). The crowd appears to have solutions to all kinds of innovation difficulties — they can come up with fresh toy concepts and solve urgent scientific concerns (Acar, 2019). Crowdsourcing activities may be found in a variety of fields, including medical (Saez-Rodriguez et al., 2016), media (Fitt, 2011), art (Casal, 2011), finance (Belleflamme et al., 2010), and government (Belleflamme et al., 2010). (Bommert, 2010). Because of its capacity to collect information quickly, cheaply, and precisely, crowdsourcing has much promise, especially in software testing. Because of technological advancements, businesses, organisations, and the government may now involve vast numbers of Internet users in executing organisational activities more efficiently. It also gives a low-cost, scalable approach to getting ideas that would otherwise be difficult or expensive to obtain internally (Cox, 2011). Due to the crowd's diversity of perspectives, assumptions, and beliefs can help lessen bias in collective decision-making when compared to small teams (Bonabeau, 2009). Businesses use crowdsourcing as a way to spot trends, understand client demands, get new viewpoints, and validate business plans.

Companies mostly employ crowdsourcing to fulfil client needs, product design and development processes, and quicker product and innovation cycles.

Product engineering requires a lot of input from a wide array of end users during different phases of the product development life cycle. Right from ideation to MVP (Minimal viable product), inputs or feedback is needed from the wide array of users. These users can be from technical or non-technical backgrounds based on the type of feedback required. It is mostly not feasible to gather this volume of information or feedback internally. Crowdtesting leverages customised testing teams sourced from a globally dispersed community of digital experts (*By Applause, they are a worldwide leader in enabling digital quality*) (Applause App Quality, Inc. (2021)) who can help in surveys, product ideas, understanding user preferences, evaluating new locations for products, product naming, testing new features, or new business ideas.

1.2 Research Problem

All sorts of technology-rich devices, be it a smartphone, a smart speaker, virtual reality, updated laptops, or others we are using daily, and all of them are driven by software. Because of the fast advancement of technology, more people are needed at desks to generate the code or software that drives it. "The process of executing a programme to identify mistakes" is how testing is defined (Mayers, 2019). As a result, it is a destructive process of looking for faults that are supposed to exist in a programme. Its primary purpose is to instil trust in a program's ability to perform as intended (Alyahya, 2020). Web applications are frequently made up of complex, multi-tiered, heterogeneous infrastructures that contain, among other things, Web sites, apps, database servers, and client PCs. Because all of these components continue to multiply, guaranteeing their high quality and dependability is essential (Alyahya, 2020). As organisations, individuals, and governments become more reliant on the Internet for day-to-day operations, low-reliability software can negatively impact them. Because software testing is labour and resource-demanding, it is challenging to produce dependable software; hence, crowdsourcing is used to meet these demands. Whether it is online application testing, game beta testing, workflow design, or bug reporting, crowdsourcing has a wide range of applications in the information technology business. Crowdsourcing is now being used to achieve corporate objectives through mass participation made possible by Web 2.0 technology. Functional testing, user acceptability testing, contract acceptance testing, user experience testing, and beta testing are examples of collaborative crowdsourcing testing. In recent years, several tech startups and e-commerce enterprises have been using crowdsourcing to cut capital investment and capture speedy new ideas.

The decision to use crowdsourcing is based on the tester's skill set, experience, and ability, which might affect the quality of work, timing, and product value (Mariani and Wamba, 2020). Humans can conceive, discriminate, and filter in crowdsourcing and learn, adapt, and use common sense and experience in ways that machines cannot. So, in this section of the dissertation, I will look at software testing in terms of issues, theories, platforms, trends, viewpoints and advancements.

A comprehensive study is needed to formally define the crowdsourcing techniques and practices beneficial for software product development. This paper will focus on developing an end-to-end crowdsourcing framework focused on testing which can easily integrate with existing software development models and practices

1.3 Purpose of Research

The purpose of this research is to investigate the role and potential of crowdsourcing in software testing. Crowdsourcing has emerged as a promising approach to harness the collective intelligence and diverse skills of a large group of individuals, often referred to as the "crowd," to accomplish tasks or solve problems. In the context of software testing, crowdsourcing offers unique opportunities to augment traditional testing approaches, enhance test coverage, and improve software quality.

The specific objectives of this research are as follows:

- Understanding the concept of crowdsourcing in software testing: This research aims to provide a comprehensive understanding of crowdsourcing as it pertains to software testing. It will explore the different types of crowdsourcing models, such as open crowdsourcing, closed crowdsourcing, and hybrid models, and analyze their suitability for software testing processes.
- Evaluating the effectiveness of crowdsourced testing: The research will assess the effectiveness of crowdsourced testing in terms of various factors, such as cost, speed, scalability, and quality. It will compare crowdsourced testing with traditional testing approaches to determine its advantages and limitations.
- Identifying the challenges and risks associated with crowdsourced testing: Crowdsourcing introduces unique challenges and risks that need to be carefully considered. This research will identify and analyze these challenges, including factors such as task decomposition, crowd reliability, coordination, and security concerns. It aims to propose strategies and best practices to mitigate these challenges effectively.
- This research will explore the broader implications of integrating crowdsourcing into software development processes. It will examine how crowdsourcing can facilitate early feedback, accelerate the release cycles, and enable continuous improvement through rapid bug detection and validation.

- Providing guidelines and recommendations for implementing crowdsourced testing: The research will offer practical guidelines and recommendations for organizations considering the adoption of crowdsourced testing. It will provide insights into the selection of suitable crowdsourcing platforms, effective crowd management strategies, and ways to integrate crowdsourced testing seamlessly into existing testing frameworks.

By accomplishing these objectives, this research aims to advance the understanding of crowdsourcing in software testing and provide valuable insights to software development organizations, researchers, and practitioners. The findings of this study will contribute to the body of knowledge on software testing methodologies and enable informed decision-making regarding the adoption and implementation of crowdsourcing as a viable testing approach.

1.4 Significance of the Study

The proposed research on crowdsourcing in software testing holds significant importance due to the following reasons:

- Advancing Software Testing Practices: By exploring the role and potential of crowdsourcing in software testing, this study aims to advance the field of software testing practices. Traditional testing approaches often face challenges in terms of resource limitations, test coverage, and scalability. Crowdsourcing has the potential to address these challenges by harnessing the collective intelligence and diverse skills of a large crowd, enabling organizations to achieve higher-quality software products.
- Enhancing Test Coverage and Quality: Crowdsourcing offers an opportunity to expand test coverage by engaging a diverse group of testers with varying backgrounds, experiences, and perspectives. This study will investigate the

effectiveness of crowdsourced testing in improving test coverage and software quality. The findings can help organizations identify the appropriate situations and tasks where crowdsourcing can have a positive impact on overall testing outcomes.

- Cost-Effectiveness and Efficiency: Cost and time efficiency are critical considerations in software development projects. Crowdsourcing has the potential to provide cost-effective and efficient testing solutions by tapping into a global talent pool. This research will evaluate the cost-effectiveness of crowdsourced testing and provide insights into its potential to reduce testing costs and time while maintaining high-quality standards.
- Mitigating Testing Challenges: Crowdsourcing introduces unique challenges in terms of task decomposition, crowd reliability, coordination, and security concerns. Understanding these challenges and proposing effective strategies to mitigate them will be a significant contribution of this study. It will equip organizations with knowledge and guidelines to effectively leverage crowdsourcing while minimizing potential risks.
- Practical Implementation Guidance: This research aims to provide practical guidelines and recommendations for organizations considering the adoption of crowdsourced testing. These guidelines will help organizations select suitable crowdsourcing platforms, manage crowds effectively, and integrate crowdsourced testing seamlessly into their existing testing frameworks. Such implementation guidance will be valuable for practitioners seeking to leverage crowdsourcing in their software testing processes.
- Enabling Continuous Improvement: By exploring the impact of crowdsourcing on software development processes, this study will investigate how crowdsourcing can facilitate early feedback, accelerate release cycles, and enable continuous

improvement through rapid bug detection and validation. This understanding will support organizations in embracing agile development methodologies and enhancing their software development lifecycle.

Overall, the significance of this study lies in its potential to advance software testing practices, improve test coverage and quality, provide cost-effective solutions, mitigate testing challenges, offer practical implementation guidance, and enable continuous improvement in software development processes. The findings of this research will benefit software development organizations, researchers, and practitioners seeking to leverage crowdsourcing as an effective testing approach in their projects.

1.5 Research Purpose and Questions

The objectives of the research are predetermined to demonstrate what the study will achieve when completed. The study will seek to achieve this by answering the central question and the relevant sub-questions posed below.

Core Question:

What are the business strategies and practices of IT Product development companies to build high-quality software products at the low cost of quality using Crowdsourcing techniques?

Sub-questions:

What primary software testing challenges do product development organisations face in building quality software products?

What are the key factors which impact software quality and quality costs?

Which Crowdsourcing practices are utilised today in the software industry?

What are the effects of Crowdsourcing practices on the quality and costs of software products?

CHAPTER II:

REVIEW OF LITERATURE

2.1 Theoretical Framework

Crowdsourcing is used to address the difficulty of outsourcing by completing work more precisely and swiftly and improving value-generating activities and services. Using crowdsourcing techniques, projects are assigned to an unspecified group of people who will be compensated for their efforts in completing the tasks. As a result, there are two critical distinctions between crowdsourcing and outsourcing: an open call and a crowdsourced solution (Burger-Helmchen and Penin 2010). Instead of depending on a single or small number of specified suppliers or third parties, crowdsourcing allows everyone to respond to an open request (Pe'nin 2012). The goal behind crowdsourcing is to use the so-called "wisdom of crowds" and the benefits that come with it (Ponsonby and Mattingly, 2015).

The core premise is that a group of individuals may produce better outcomes than anyone alone. As a result, crowds are far more capable of addressing problems than any expert (Jeppesen and Lakhani 2010; Leimeister S. Zogaj et al. 2015). Estelles-Arolas et al. (2012) compiled 40 definitions from 32 papers published between 2006 and 2011 and offered a unified definition compatible with all 165. I can identify four common elements of crowdsourcing based on the preceding definitions. Software testing may be crowdsourced in several contexts, such as when a software company lacks testers specialising in a particular form of testing, such as security testing. This type of testing may be challenging to accomplish, even for highly experienced internal testing teams. Because hackers' techniques are constantly evolving, traditional security solutions may not eliminate all security flaws.

Furthermore, the ideal people to do usability testing are end-users. Consequently, end users' representations and domain knowledge suggestions can be leveraged to improve crowdsourced usability testing outcomes (Alsayyari & Alyahya, 2018). Aside from the benefit of collective intelligence, the literature lists many other benefits for businesses when it comes to crowdsourcing: access to a large pool of resources and capabilities, as well as ideas and solutions; outsourcing of failure risks due to performance-based remuneration; and (Burger-Helmchen and Penin 2010). However, crowdsourcing has many disadvantages, including the risk of disclosing valuable information such as intellectual property or proprietary information (Rayna and Striukova 2010) and the risk of receiving insufficient or low-quality contributions from the crowd (Leimeister et al. 2015; Hoßfeld et al. 2012).

The phrase 'crowdsourced software testing' refers to crowdsourcing techniques to create the software (in its broadest sense). Despite its widespread use in various software engineering jobs, "crowdsourced software engineering" is rarely defined precisely. According to our research, 68 per cent of the journals I studied utilise (or repeat) the notion of crowdsourcing without providing a description (Mao et al., 2017). Howe's definition (18%), which I discussed previously, is the most often cited among the 34 per cent who cite a definition. Only two of the 203 papers I looked at provide a precise description of what crowdsourcing entails when it comes to software engineering (Mao et al., 2017). Employers (also known as requesters) who have software development work that needs to be completed; workers who participate in the development of software; and platforms that provide an online marketplace where requesters and workers can meet are the three types of actors (or stakeholders) involved in crowdsourced software engineering.

In general, task processing in crowdsourcing may be classified as either dependently or autonomously driven. For example, when a group of people work together to complete a job, such as Wikipedia, I can detect the dependence, but independently driven crowdsourcing, such as Amazon Mechanical Turk or oDesk, is an example of independently driven crowdsourcing. Another example of independent crowdsourcing is when a group of individuals collaborate on an inventive solution to a problem and agree among themselves to create the best option out of many (Afuah and Tucci 2012; Tung et al. 2013).

2.2 Theories and Model Foundation

To aid Crowdsourced Software Engineering, various theories and models have been offered. Kazman introduced the Metropolis Model for Crowdsourced Software Engineering, and Chen (Kazman and Chen, 2009, 2010) argued that traditional software development approaches such as waterfall, spiral, and more recent agile models are not suitable for CSE. Platform (kernel), applications created on the kernel (periphery), and end-users are among the roles outlined under the Metropolis Model (referred to as masses). The seven management principles of the approach were introduced.

Saxton et al. created a taxonomy of nine crowdsourcing models (2013). Crowdsourced Software Engineering is supported by intermediary and collaborative software development models. (Tsai et al., 2014) Tsai presented it as a way to summarise the commonalities among various Crowdsourced Software Engineering approaches. The platform provides options for worker ranking and recommendation. Many stakeholders have collaboration choices. A few studies analysed crowd developer competition using game-theoretic crowd formulations (Wu et al., 2013; Hu and Wu, 2014; Xu and Wang, 2014b). Wu et al. emphasise the 'min-max' (defence-offence) nature of crowdsourcing software development competitions (Wu et al., 2013). For evaluating TopCoder developer competition, Hu and Wu (2014) proposed a game-theoretic model. The significant hypotheses were addressed in further depth.

2.2.1 Cultural Values for Technology Adoption

The most crucial component in crowdsourcing is the acceptance of cultural values and technological advancements. Several combinations of psychology and information science theories were employed by the researchers to investigate the possible aspects that may impact users' behavioural intentions to use technology (Zhao and Zhu, 2014). Theories that have been widely used in the acceptance of technology between individuals include psychological constructs such as work motivation, extrinsic and intrinsic motivation, hedonic motivation, self-efficiency, planned behaviour and reasoned action (Locke & Latham, 1990; Ryan & Deci, 2000; Bandura, 1977; Van der Heijden, 2004). Since crowdsourcing is different from the outsourcing and technology-based approach so there may be chances that everybody or every organisation will not join it earlier but later on upon getting any kind of motivation (work motivation, extrinsic and intrinsic motivation, hedonic motivation, self-efficiency, planned behaviour and reasoned action) will help to accept it.

Digital crowdsourcing is yielding higher research output and applications. Digital devices provide the opportunity to engage people's interest and commitment further to cooperatively share their efforts in gathering data and information for the benefit of the greater community by utilising digital devices. In contrast, if a technology involves less effort, it may be adopted more extensively since people will feel more comfortable working with it and will be less annoyed as a result (Srite and Karahanna, 2006). This attribute is referred to as perceived ease of use (PEU), and it is useful in the development of crowdsourcing modules because it allows a larger number of people to participate easily in the testing. As a result, in order to achieve better results, the software testing module should be designed with the convenience of use of the population in mind.

2.2.2 Theory of Reasoned Action (TRA)

The "Theory of Reasoned Action" is built on the presumption that the behaviour being evaluated is one that individuals feel they can carry out at any time they want to. This hypothesis describes the extent to which people feel they are capable of doing the activity because they have sufficient capabilities and/or opportunities or that they lack these. Fishbein and Azjen presented the A model known as the TRA model in 1975. It is concerned with the development of a system of observation for two categories of variables, which are as follows:

1. Attitudes are defined as positive or negative feelings in relation to the achievement of an objective.
2. Subjective norms are the very representations of the individual's perception in relation to the ability to reach those goals with the product.

So, if the software product teams apply this in crowdsourcing in software testing, it will be like outsourcing to those who know its fundamentals or have expertise, training, or skills.

According to the notion, the average response of many individuals, even amateurs, to a question is typically more correct than the perspective of a small number of specialists. In this regard, a community of persons with the same interests and who are confronted with the same challenges might produce better goods and solutions than professionals working in isolation in the sector. According to their research, Jean-Fabrice Lebraty and Katia Lobre-Lebraty, both information systems academics, affirmed that the "variety and impudence of the members of a crowd" is a valuable asset to crowdsourcing operations (Lebraty & Lobre-Lebraty, 2013).

2.3 Claimed Advantages and Growth Trends

Compared to typical software development approaches, crowdsourced software engineering offers many potential advantages to provide. When it comes to software development, crowdsourcing may assist organisations in integrating elastic, external human resources to lower costs associated with internal employment while also using the distributed production model to accelerate the development process.

According to TopCoder, as compared to conventional software development, the process of crowdsourced software development can offer customer-requested software assets with a lower defect rate at a lower cost and in less time (Mao et al., 2015). If you compare crowdsourced development to in-house development or outsourcing, TopCoder claims that it can save you between 30 and 80 per cent on expenditures (Lydon, M. (2012)). According to Boudreau and Lakhani (2016), the defect rate was observed to be 5 to 8 times lower in the TopCoder American Online case study when compared to traditional software development approaches. The Harvard Medical School used Crowdsourced Software Engineering to enhance DNA sequence gapped alignment search methods, according to a paper published in Nature Biotechnology (Boudreau and Lakhani, 2016). When compared to the MegaBLAST system developed by the National Institutes of Health in the United States, the top crowd solution achieved greater accuracy and a three-order-of-magnitude gain in performance in just two weeks.

Crowdsourced Software Engineering is becoming increasingly popular among businesses and organisations ranging from the military to academic institutions to huge information technology corporations. Crowdsourced Formal Verification (CSFV) is a programme developed by DARPA for software formal verification, and the Veri games website was founded to make this approach more accessible (Yilmaz, 2015).

NASA and Harvard Business School collaborated to build the NASA Tournament Laboratory, which is tasked with developing crowdsourced software solutions for NASA systems. For Office 2010, Windows 8.1, and Windows 10, Microsoft used crowdsourcing to complete portions of the software development process. AppStori and Mob4Hire are two examples of crowdsourcing platforms that have been specifically designed for the software development industry.

2.3.1 Crowdsourcing Practice in Software Testing

This section describes the most popular crowdsourcing platforms and typical crowdsourced software engineering and testing methods. Crowdsourcing has been used for usability testing, performance testing, GUI testing, test case development, and the Oracle problem (Schneider and Cheung, 2011; Alsayyari and Alyahya, 2018). Since most case studies are based on one (or more) of these commercial platforms, we give pertinent case studies. Several frameworks have been suggested to facilitate crowdsourcing for software testing (Yan et al., 2014; Liang et al., 2014)

2.3.2 Commercial Platforms

Table 1 lists commercial crowdsourcing platforms for software engineering. These platforms employ numerous open-call techniques, such as on-demand matching, which connects workers with registrations, and online bidding, which allows developers to bid on projects before starting work on them. Also, the platforms cover a wide range of tasks in software development. Many different sorts of software development jobs may be accomplished using platforms such as TopCoder and GetACoder. Others are more precise in their requests. For example, uTest and BugCrowd are both software testing and security analysis tools that are created specifically for their respective purposes. There are other broad crowdsourcing markets, such as Amazon Mechanical Turk, oDesk, and Freelancer, which are not specially created for software engineering, but may be used to support a variety of software development jobs regardless of their intended usage (Mao et al. 2017).

Table 1. A List of Commercial Platforms for Crowdsourced Software Testing (Mao et al. 2017)

Platform	**URL**	**Task Domain**	**Open call Form**
TopCoder	www.topcoder.com	Software Development	Online Competition
GetACoder	www.getacoder.com	Software Development	Online Bidding
uTest	www.utest.com	Software Testing	On-demand Matching, Online Competition
Passbrains	www.passbrains.com	Software Testing	On-demand Matching
99Tests	www.99tests.com	Software Testing	On-demand Matching
TestBirds	www.testbirds.com	Software Testing	On-demand Matching
Testbats	www.testbats.com	Software Testing	On-demand Matching
Pay4Bugs	www.pay4bugs.com	Software Testing	On-demand Matching
CrowdTesters	www.crowdtesters.com.au	Software Testing	On-demand Matching
TestFlight	www.testflightapp.com	Mobile App Testing	On-demand Matching
Mob4hire	www.mob4hire.com	Mobile App	Testing Online Bidding
Testin	www.itestin.com	Mobile App Testing	On-demand Matching
Ce.WooYun	ce.wooyun.org	Software Security Testing	On-demand Matching

Bugcrowd	www.bugcrowd.com	Software Security Testing	Online Competition

2.3.3 Case Studies

In recent years, a large number of case studies involving crowdsourced software testing and engineering have been published. The majority of them are built on one or more of the commercial platforms indicated above. Among these, the TopCoder platform has the greatest number of case studies that have been published in the literature (Nag et al., 2012a; Li et al., 2013; Tajedin and Nevo, 2014; Stol and Fitzgerald, 2014). Stol et al. (2014) provided an in-depth case study with a client firm that had crowdsourced software development expertise utilising TopCoder, which was conducted in collaboration with the client company. Through conversations with representatives from the client firm, several difficulties relevant to the TopCoder development process were uncovered (Stol and Fitzgerald, 2014).

Another in-depth case study in the form of interviews was carried out by Tajedin and Nevo (2014), although this time from the perspective of TopCoder's management team rather than from the perspective of the customer. During the case study, it was discovered that two sorts of value-adding activities may be performed on the crowdsourcing platform: macro, market-level actions and micro, transaction-level actions. Wu et al. (2013) provided an overview of the lessons learnt from their data collection on software crowdsourcing. The techniques used by TopCoder and AppStori, two crowdsourced software development platforms, were investigated. Specifically, the authors contend that the competitive behaviour of' min-max competitions help the quality and inventiveness of crowdsourced software development.

For the SPHERES Zero Program, sponsored by NASA, DARPA, and Aurora Flight Sciences, Nag et al. (2012) used TopCoder to crowdsource software development. Nag's master's thesis has further details (Nag, 2012). Lakhani et al. (2010) reviewed the progress of TopCoder from 2001 to 2009, covering the platform and community, customer advantages and issues, and management responsibilities and obstacles. Archak (2010) studied TopCoder developers' strategic behaviour.

A phenomenon known as cheap talk (Farrell and Rabin, 1996) occurs during the contest registration phase when highly ranked developers register early to prevent competitors from entering the market. Archak argues that cheap talk and TopCoder's reputation system help make simultaneous online contests more efficient. Focussed regression analysis was used to analyse the elements affecting the contest's output quality. The final submission quality was predicted by payment and requirement criteria. Li et al. (2013) studied TopCoder. The project and platform elements yielded 23 quality factors.

2.4 Limitations

Other unresolved concerns in Crowdsourced Software Engineering include collaboration and communication. Coordination and communication are required for both resources and development. Another concern is intellectual property and data security. Because crowdsourcing is an open call, every one may see the tasks. Transferring task deliverables may cause IP difficulties. For example, the crowd developers may utilise non-commercial code, while the client firm demands the work for commercial use

2.4 Summary

In this survey, I looked into the usage of crowdsourcing in software testing and research. An analysis of the research development in this domain from the viewpoints of theories practises, and applications found a growing rate. These include crowdsourced software development models, large commercial software engineering platforms and case studies, as well as applications to software testing and engineering research. The report also discusses challenges in Crowdsourced Software Engineering, as well as past research and solutions. Finally, the literature is utilised to identify gaps and potential research challenges.

CHAPTER III:
METHODOLOGY

3.1 Overview of the Research Problem

The research problem addressed in this study revolves around the utilization of crowdsourcing in the context of software testing. Crowdsourcing has gained attention as a potentially effective approach to augment traditional testing methods and overcome challenges related to resource limitations, test coverage, and scalability. However, there is a need to investigate the practical implementation, effectiveness, challenges, and best practices associated with crowdsourcing in software testing.

The research problem can be broken down into several key aspects:

- Implementation and Adoption: One aspect of the research problem involves understanding how organizations adopt and implement crowdsourcing in their software testing processes. This includes exploring the selection of appropriate crowdsourcing platforms, defining the scope of testing tasks suitable for crowdsourcing, and integrating crowdsourced testing seamlessly into existing testing frameworks.
- Effectiveness and Quality Assurance: Another aspect is to evaluate the effectiveness of crowdsourced testing in terms of test coverage, defect identification, and overall software quality assurance. This includes comparing crowdsourced testing with traditional testing approaches to determine its advantages and limitations and investigating its impact on improving software quality.
- Challenges and Risks: The research problem also entails identifying the challenges and risks associated with crowdsourcing in software testing. This includes examining factors such as task decomposition, crowd reliability,

coordination, and security concerns. Understanding these challenges is crucial for devising strategies and best practices to mitigate them effectively.

- Integration with Software Development Processes: The research problem further involves investigating how crowdsourcing can be integrated with software development processes. This includes exploring the impact of crowdsourcing on agile development methodologies, continuous integration, and continuous delivery. Understanding how crowdsourcing can facilitate early feedback, accelerate release cycles, and enable continuous improvement is essential for maximizing its potential benefits.

The research problem aims to address these key aspects by conducting a comprehensive investigation into crowdsourcing in software testing. By exploring the implementation, effectiveness, challenges, and best practices, this research will contribute to the existing body of knowledge and provide valuable insights to researchers, practitioners, and organizations seeking to leverage crowdsourcing as a viable approach in their software testing endeavours.

3.2 Research Purpose and Questions

The objective of this qualitative data sampling is to gather in-depth insights into the practical implementation and challenges of crowdsourcing in software testing. The objectives of the research are predetermined to demonstrate what the study will achieve when completed. The study will seek to achieve this by answering the central question and the relevant sub-questions posed below.

Core Question:

- What are the business strategies and practices of IT Product development companies to build high-quality software products at the low cost of quality using Crowdsourcing techniques?

Sub-questions:

- What primary software testing challenges do product development organisations face in building quality software products?
- What are the key factors which impact software quality and quality costs?
- Which Crowdsourcing practices are utilised today in the software industry?
- What are the effects of Crowdsourcing practices on the quality and costs of software products?

3.3 Research Design

To investigate the practical implementation and challenges associated with crowdsourcing in software testing, a qualitative research design will be employed. Qualitative techniques offer a deeper understanding of participants' experiences, perceptions, and practices, allowing for in-depth exploration of the research problem. The following outlines the research design using qualitative techniques for the topic of crowdsourcing in software testing:

- Research Approach: This study will utilize an exploratory research approach. The goal is to gain insights into the practical implementation of crowdsourcing in software testing, the challenges faced, and the best practices employed by organizations.
- Sampling: Purposeful sampling will be used to select participants who have relevant experience and knowledge in implementing crowdsourcing in software testing. Participants may include software testers, project managers, and decision-makers in organizations that have utilized crowdsourcing for testing purposes. The sample size will be determined based on data saturation, where new insights and themes no longer emerge from additional participants.

- Data Collection:
  - Interview questionnaire: In-depth, structured interview questionnaire will be conducted with the selected participants. The interviews will provide an opportunity to delve into their experiences, perceptions, and practices regarding crowdsourcing in software testing. The interview questions will be developed based on the research objectives and literature review, covering topics such as the selection of crowdsourcing platforms, task decomposition, coordination, crowd management, and challenges faced during implementation. The responses will be recorded using an online platform and transcribed for subsequent analysis.
  - Document Analysis: Relevant documents, such as project reports, guidelines, and communication records related to the implementation of crowdsourcing in software testing, will be collected and analyzed. These documents can provide additional insights into the challenges and best practices observed in real-world contexts.

- Data Analysis:
  - Thematic Analysis: The recorded interview questionnaire responses and document analysis will be subjected to thematic analysis. This process involves identifying recurring themes, patterns, and categories within the data. Initially, a coding framework will be developed based on the research objectives and emergent themes from the data. The data will then be coded, organized, and analyzed using qualitative data analysis software or manual techniques. Themes and patterns will be refined and interpreted to generate meaningful findings.

- Ethical Considerations: Ethical guidelines for research involving human participants will be followed. Informed consent will be obtained from participants, ensuring their confidentiality and privacy. The study will also comply with ethical standards regarding data storage, usage, and reporting.
- Validity and Reliability: Strategies such as member checking, peer debriefing, and keeping an audit trail of the research process will be employed to enhance the validity and reliability of the study.

By employing qualitative techniques, this research design will provide a rich and nuanced understanding of the practical implementation and challenges associated with crowdsourcing in software testing. The findings will contribute to the existing body of knowledge, inform best practices, and offer valuable insights to researchers, practitioners, and organizations aiming to leverage crowdsourcing in their software testing endeavours.

3.4 Population and Sample

In qualitative research, the population refers to the group of individuals or organizations that possess the characteristics relevant to the research topic. For the topic of crowdsourcing in software testing, the population would consist of individuals and organizations that have experience or knowledge in implementing crowdsourcing for software testing purposes. According to Ziebland et al, the sampling technique in qualitative research is primarily intended to represent a wide range of opinions and experiences, rather than to mimic their frequency in the general population (Ziebland & McPherson, 2006).

The sample, on the other hand, refers to the subset of the population that will be included in the study. Given the specific nature of qualitative research, purposeful sampling techniques are commonly employed to select participants who can provide rich and diverse insights into the research topic. Here is an example of population and sample for the topic of crowdsourcing in software testing:

Population:

The population for this study includes:

- Software testers who have experience with crowdsourced testing or have been involved in implementing crowdsourcing in software testing projects.
- Project managers or decision-makers in organizations that have utilized crowdsourcing for software testing.
- Researchers or experts with knowledge and expertise in crowdsourcing and software testing.

Sample:

The sample will be purposefully selected from the population to ensure a range of perspectives and experiences. The following factors will be considered when determining the sample:

- Job Roles: Include software testers, project managers, and decision-makers who have direct involvement in software testing activities or decision-making related to crowdsourced testing.
- Experience: Select participants with a diverse range of experience in implementing crowdsourcing in software testing, including both successful and unsuccessful cases.
- Organization Types: Include participants from different types of organizations, such as startups, large corporations, and outsourcing companies, to capture varied perspectives.
- Geographic Diversity: Consider selecting participants from different geographical locations to account for any potential cultural or contextual differences.

While the sample size may be small, it can still be information and enable the researcher to obtain information that is meaningful, and derive useful perceptions from the interviewees (Creswell, 2003). Further, because the sample size is small, it enables the interview method to shine, since it only requires a few participants to gather rich and detailed data (Genise, 2002).

The sample size will depend on the principle of data saturation, where new insights and themes no longer emerge from additional participants. It is essential to ensure that the selected sample is representative enough to provide a comprehensive understanding of the research topic.

3.5 Participant Selection

When conducting qualitative research on the topic of crowdsourcing in software testing, selecting participants who can provide rich and diverse insights is crucial. Qualitative techniques allow for an in-depth exploration of participants' experiences, perceptions, and practices. Here are some considerations for participant selection using qualitative techniques:

- Relevant Experience: Select participants who have direct experience with crowdsourcing in software testing. This includes software testers, project managers, decision-makers, and researchers who have been involved in implementing or overseeing crowdsourcing initiatives. Participants with firsthand knowledge and practical experience can provide valuable insights into the challenges, best practices, and outcomes of crowdsourcing in software testing.
- Diversity of Perspectives: Aim for a diverse sample to capture a range of perspectives. Consider factors such as job roles, years of experience, organization types (e.g., startups, large corporations, outsourcing companies), and geographic locations. Including participants from different backgrounds and contexts can

reveal variations in implementation approaches, challenges faced, and lessons learned.

- Variation in Success: Select participants who have experienced both successful and unsuccessful crowdsourcing projects in software testing. This variation will enable a comprehensive understanding of the factors that contribute to success or failure in different contexts. Participants who have overcome challenges and achieved positive outcomes can provide insights into effective strategies, while those who have faced difficulties can highlight potential pitfalls to avoid.
- Expertise and Knowledge: Include participants who possess expertise in crowdsourcing, software testing, or related domains. Researchers, consultants, or practitioners who have studied or worked extensively in the field can offer valuable insights, theoretical perspectives, and industry trends.
- Availability and Willingness to Participate: Ensure that selected participants are available and willing to contribute to the study. Obtaining informed consent from participants is essential, and they should understand the research purpose, potential risks, and benefits of participation. Consider any constraints, such as time availability or organizational policies, which may impact participants' ability to engage in the research.
- Data Saturation: The sample size should be determined based on data saturation, which is reached when collecting additional data does not yield new insights or themes. The researcher should continue sampling until reaching this point of saturation to ensure comprehensive coverage of the research topic.

Overall, the participant selection process for qualitative research on crowdsourcing in software testing should prioritize individuals with relevant experience, diverse perspectives, and a willingness to share their insights. By selecting participants thoughtfully, researchers can gather in-depth and nuanced data that contributes to a comprehensive understanding of the research topic.

3.6 Instrumentation

For this research, there were two instruments employed to ensure that the data being collected for the research was relevant, valuable and could potentially assist in the research. The primary instrument was the semi-structured interviews using an interview questionnaire that was conducted by the researcher. In qualitative research on the topic of crowdsourcing in software testing, an interview questionnaire can be an effective research instrument for gathering rich and in-depth insights from participants. For secondary input, online research and organisational public data will be utilised. Details for each of the instruments are given below sub-sections.

3.7 Data Collection Procedures

The Interview Questionnaire

The Qualitative researchers collect data themselves by examining documents, observing behaviour, and interviewing participants. They may use an instrument, but it is one designed by the researcher using open-ended questions. (Creswell, J. W. ,2013).

The research interview questionnaire process involves several steps to ensure a systematic and effective data collection process. Here is an outline of the typical process for developing and administering a research interview questionnaire:

- Research Objectives: Clearly define the research objectives and questions that were aimed to address through the interview questionnaire. Identified the specific areas or themes I wanted to explore related to crowdsourcing in software testing.

- Questionnaire Design: Developed a set of closed and open-ended questions that align with the research objectives. The questionnaire had general introductory questions to establish a relationship with the participant, then the questionnaire progressed to more specific and probing questions. The questions were organized in a logical and sequential order that flowed naturally during the interview.
- Pilot Testing: Conducted a pilot test of the interview questionnaire with a small group of individuals who have similar characteristics to the target participants. This allows me to identify any ambiguities, potential issues, or areas for improvement in the questionnaire. Revised and refined the questionnaire based on the feedback received.
- Obtain Ethical Approval: Ensured that the research follows ethical guidelines and safeguards participant confidentiality and privacy.
- Participants Selection: Utilised a participant identification strategy based on who met the desired criteria like Job Role, experience, and Crowd-sourcing awareness. This involved contacting individuals or organizations with expertise in crowdsourcing in software testing through professional networks like LinkedIn, or referrals.
- Informed Consent: Before conducting the interview, explain the purpose of the research, the voluntary nature of participation, and the confidentiality of data. Obtained informed consent from each participant, ensuring that they understand their rights as participants.
- Online Interview Form: The interview questionnaire was shared with the participants using the Google Forms platform. A unique link was sent to all the participants. The responses were automatically saved in an Excel sheet.

- Transcribe and Analyze Data: Transcribed the recorded responses. Organized the data systematically and analyze it using qualitative data analysis techniques such as thematic analysis or content analysis. The data was used for identifying patterns, themes, and recurring concepts within the data.
- Data Interpretation and Reporting: Interpreted the findings based on the data analysis. In the subsequent sections, the main themes, insights, and patterns that emerged from the interviews. Present the findings clearly and coherently, using quotes or excerpts from the interviews to support the interpretations.
- Ensure Confidentiality: Ensured the confidentiality and anonymity of participants by assigning pseudonyms or coding their identities during the reporting stage. Follow the data protection guidelines and ethical requirements when handling and storing data.
- Reflect on Limitations: Reflect on the limitations of the interview questionnaire process, including any biases or limitations inherent in qualitative research. Consider the implications of the findings and identify opportunities for future research.

This process ensured systematic data collection, ethical considerations, and meaningful analysis of the obtained qualitative data.

The complete list of interview questions is written in Appendix D.

3.8 Data Analysis

Data analysis for research interview questionnaires in qualitative research involves a systematic and rigorous examination of the collected data to identify patterns, themes, and insights. Here is an outline of the typical data analysis process for research interview questionnaires:

- Transcription: Transcribed the recorded interview responses or notes to create a written record of the data. Transcriptions captured participants' responses, including verbatim quotes, to ensure accuracy during the analysis process.
- Familiarization: Read and re-read the transcriptions to become familiar with the data. Developed an understanding of the context, participants' experiences, and the overall content of the interviews.
- Coding: Used coding techniques to categorize and organize the data. Begin with open coding, where you identify and label meaningful units of information, such as concepts, themes, or patterns. Apply codes to segments of the text that represent these units. This process can be done manually or using qualitative data analysis software.
- Theme Development: Grouped similar codes together to develop themes. Themes are overarching patterns or concepts that emerge from the data. Look for connections, commonalities, or differences in participants' responses and assign descriptive labels to each theme.
- Data Reduction: Analyzed the themes further by condensing, summarizing, or synthesizing the data. Seek to identify the most significant and relevant aspects of each theme and extract meaningful insights from the data.
- Interpretation: Interpreted the data within the context of the research objectives and theoretical frameworks. Analyze the relationships between themes, identify sub-themes, and explore any contradictions or divergent perspectives that may have emerged during the analysis.
- Triangulation: Enhanced the rigour and credibility of the analysis by comparing and contrasting the data with other sources or methods. Triangulation involves

validating the findings by examining consistency or convergence across multiple data sources, such as interviews, observations, or documents.

- Reflexivity: Reflect on your own biases, assumptions, and preconceived notions throughout the analysis process. Maintain reflexivity by continuously considering how your perspectives may influence the interpretation of the data. Document your reflections and include them in the analysis process.
- Reporting: Presented the findings comprehensively and coherently. Used quotes or excerpts from the interviews to illustrate key points and support the interpretations. Provide a rich and detailed description of the themes, sub-themes, and insights that emerged from the data.

3.9 Research Design Limitations

While research interview questionnaires are valuable for gathering qualitative data, there are several limitations that researchers should be aware of. These limitations can impact the data collection process and the overall validity and generalizability of the findings. Here are some common limitations associated with research interview questionnaire data collection:

Sample Bias: The participants selected for the interview questionnaire may not represent the entire population or a diverse range of perspectives. There may be inherent biases in the sample, such as participants who are more willing to participate or have specific characteristics or experiences that differ from the broader population. Researchers should carefully consider the representativeness of the sample and acknowledge any limitations in generalizing the findings.

Recall Bias: Participants may struggle to accurately recall past experiences or events, leading to potential inaccuracies or incomplete information in their responses. The accuracy and reliability of participants' memories may vary, particularly for complex or distant events. Researchers should be aware of this limitation and consider using prompts or probing questions to help participants recall specific details.

Interviewer or Questionnaire Bias: The presence and interaction style of the interviewer can influence participants' responses. Interviewers may inadvertently introduce biases through their questioning techniques, body language, or personal beliefs. Researchers should strive to minimize interviewer bias by following a standardized interview protocol, receiving training in interview techniques, and maintaining objectivity throughout the data collection process.

Limited Scope: The interview questionnaire may focus on specific aspects or dimensions of the research topic, limiting the breadth and depth of the data collected. It is essential to clearly define the scope and objectives of the interview questionnaire and acknowledge any areas that are not addressed or explored in depth.

Time and Resource Constraints: Conducting research interview questionnaires can be time-consuming and resource-intensive. Researchers may face limitations in terms of available time, budget, or personnel to conduct and analyze the interviews. It is important to carefully plan and allocate resources to ensure an effective data collection process.

Subjectivity and Interpretation: Qualitative data analysis involves interpretation and subjective judgment. Different researchers may interpret the data differently, potentially leading to variations in the analysis and findings. Researchers should strive for transparency and rigour in their analysis, documenting their analytical process and seeking peer input or intercoder reliability checks to enhance the trustworthiness of the findings.

Limited Generalizability: Qualitative research, including research interview questionnaires, typically aims for in-depth understanding rather than generalizability to a larger population. The findings may be context-specific and may not be directly applicable to other settings or populations. Researchers should clearly define the scope and context of the study and acknowledge the limitations in generalizing the findings beyond the specific research context.

The researcher acknowledges these limitations and discusses them in the research report to provide transparency and ensure the appropriate interpretation and application of the findings. By being aware of these limitations, researchers can make informed decisions about the design, implementation, and analysis of research interview questionnaires.

3.10 Conclusion

The research methodology employed in this study provides valuable qualitative data that adds depth and richness to our understanding of crowdsourcing in software testing. The insights gained from the participants' experiences and perspectives can contribute to the development of best practices, guidelines, and recommendations in the field.

It has provided a comprehensive and in-depth exploration of crowdsourcing in software testing. The interview questionnaire data collection process, combined with rigorous data analysis techniques, has enabled us to uncover valuable insights and generate meaningful findings. These findings will contribute to the body of knowledge on crowdsourcing in software testing and inform future research, industry practices, and decision-making in this domain.

CHAPTER IV:
RESULTS

4.1 Summary of Findings

The results and findings section of this research study presents the outcomes of the data analysis process and highlights the key findings that emerged from the analysis of the collected data. This section aims to provide a comprehensive and detailed account of the patterns, themes, and insights identified from the research interview questionnaires conducted with participants in the field of crowdsourcing in software testing.

4.1.1 The research case

This section details the information about the respondents. Here, I will delve into the 32 responses received during the online questionnaire campaign. The online questionnaire hosted via Google Forms received responses from 32 individuals across different countries, domains and roles. All 32 respondents filled the mandatory questions and among the thirty-two respondents, only twenty-two expressed interest or willingness to provide their names. Additionally, only seventeen respondents shared their email IDs, while the four respondents provided information about their optional personal LinkedIn profiles.

4.1.2 Experience in software testing field

Almost all of the respondents have good experience in the software testing field based on the years of experience it has been found that most of the respondents (twelve) have experience of more than 15 years while six respondents have experience of about 15 to 8 years while remain respondents have experience of about 8 to 4 years and 4 to 1 years respectively in equal proportion i.e. seven respondents each (Figure 4.1).

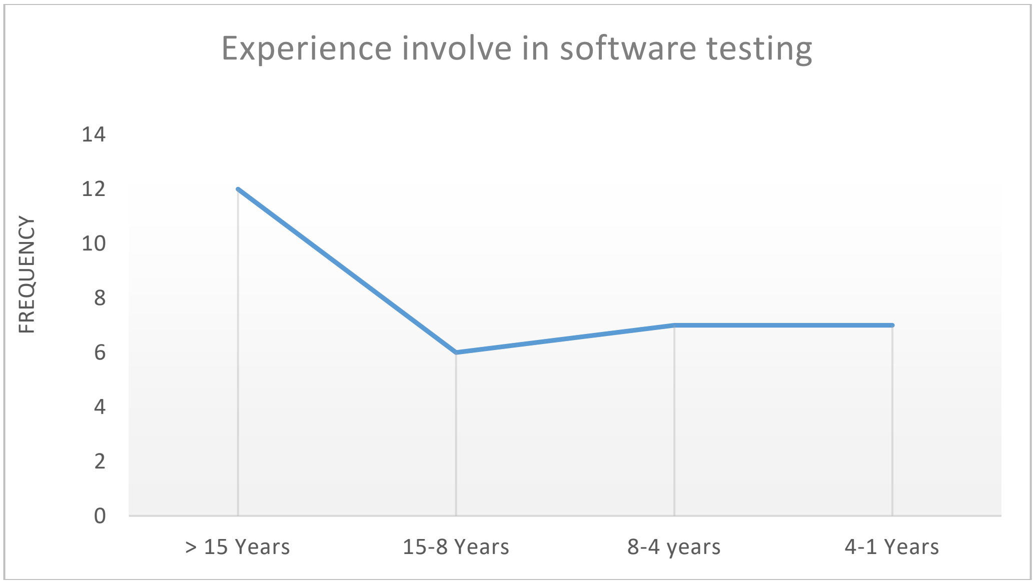

Figure 4.1 Experience of respondents in the software testing field.

Except this most of the respondents about sixty percent never used crowd-sourcing for software testing while the remaining thirty-seven per cent of respondents used crowd-sourcing for software testing (Table 4.1).

Table 4.1: Percentage and frequency of respondents that have ever used crowd-sourcing for software testing.

Response	Frequency	Percentage
No	20	62.50%
Yes	12	37.50%

Table 4.2: Summary of participants in the research

Code	**How long have you been involved in software testing?**	**Role**

P01-SM	> 15 Years	Senior Management
P02-SM	> 15 Years	Senior Management
P03-SM	> 15 Years	Senior Management
P04-SM	> 15 Years	Senior Management
P05-SM	> 15 Years	Senior Management
P06-SM	> 15 Years	Senior Management
P07-SM	> 15 Years	Senior Management
P08-SM	> 15 Years	Senior Management
P09-SM	> 15 Years	Senior Management
P10-SM	> 15 Years	Senior Management
P11-SM	> 15 Years	Senior Management
P12-SM	> 15 Years	Senior Management
P13-JT	1-4 Years	Jr. Tester
P14-JT	1-4 Years	Jr. Tester
P15-JT	1-4 Years	Jr. Tester
P16-JT	1-4 Years	Jr. Tester
P17-JT	1-4 Years	Jr. Tester
P18-JT	1-4 Years	Jr. Tester
P19-JT	1-4 Years	Jr. Tester
P20-ST	4-8 years	Sr. Tester
P21-ST	4-8 years	Sr. Tester

P22-ST	4-8 years	Sr. Tester
P23-ST	4-8 years	Sr. Tester
P24-ST	4-8 years	Sr. Tester
P25-ST	4-8 years	Sr. Tester
P26-ST	4-8 years	Sr. Tester
P27-MM	8-15 Years	Middle Management
P28-MM	8-15 Years	Middle Management
P29-MM	8-15 Years	Middle Management
P30-MM	8-15 Years	Middle Management
P31-MM	8-15 Years	Middle Management
P32-MM	8-15 Years	Middle Management

4.2 Data Analysis

This section covers the analysis and findings from the screener surveys and interviews.

4.2.1 Findings from Interviews

The thirty-two participants chosen for this research were first introduced to the research via communication mediums like LinkedIn messaging, emails or an initial overview call with the respondent. They were shared a link to an online questionnaire hosted using Google Forms. A total of 32 respondents filled the forms and 4 out of 32 were given further clarifications using a phone call to fill the form. Each participant was chosen with care to ensure that there was a fair representation of the Information Technology and

Crowdsourcing industry in its current state globally. Using the online questionnaire, the researcher asked a series of open-ended and closed-ended questions to each participant, followed by deeper questions based on theirexperiences. The questions asked using the online questionnaire are listed below.

Table 4.3: List of the questions

S. No.	Question
1	If you have utilized crowdsourcing, which domain it was used for?
2	What types of testing are you currently involved with?
3	Have you utilised any of the following platforms?
4	In what types of testing approaches do you think crowdsourcing is most effective for software testing?
5	What are the main benefits of using crowdsourcing for software testing, in your opinion?
6	What are the main challenges or drawbacks of using crowdsourcing for software testing?
7	How do you think the use of crowdsourcing for software testing will evolve in the future?
8	Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?
9	Do you have any recommendations for organizations considering using crowdsourcing for software testing?
10	What criteria will you use to select a crowdsourcing platform for testing purposes?

Among these questions, question number 1 to 4 represent the crowdsourcing practices that are utilised today in the software industry. Questions number 5, 7, 8, and 9 represent the effects of crowd-sourcing practices on the quality and costs. Question number 6 represents the primary software testing challenges do product development organisations face in building quality software products of software products while Question number 10 represents the key factors that impact software quality and quality costs.

4.2.2 Coding system for interviewees

A unique code was assigned to each response from every response. The code assigned was as follows: -

Table 4.4: Coding system for interviewees

Response	Codes	Response	Codes	Response	Codes
2nd income benefit for both the Employer and the Tester.	A	Healthcare	AA	Quality control mechanisms	BA
99tests	B	Helps Agile & ensures the ability to test in production not just in Sandbox i.e. real testers, real devices, real payment methods, etc.	AB	Rainforest QA	BB
Look at the likely quality and value of the data and the cost of processing it.	C	Helps accelerate the Testing cycle	AC	Reduces lag related to the Talent management cycle	BC

Logistics	D	High-Tech	AD	Resources	BD
SRE	E	I built the crowd-testing platform for my ex-employer	AE	Retail	BE
Accessibility	F	Improved quality control	AF	Security	BF
Advanced testing tools	G	In-house or contractor-driven teams	AG	Security risks	BG
ALM	H	Increased adoption	AH	Selenium	BH
Any other channels	I	In-store	AI	Takes too long	BI
Anything else we need - e.g. Integration	J	Insurance	AJ	Test coverage	BJ
API Testing	K	Integration	AK	Testbirds - www.testbirds.com, Crowd Sprint, Passbrains, TesterWork, QAProvider,	BK
Banking	L	ISO25000	AL	Testers work	BL
Bugcrowd	M	Jira	AM	Testing coverage and scalability	BM
BugFinders	N	Lack of control	AN	Testing expertise and experience	BN
Consumer Goods & Distribution	O	Life Sciences	AO	Testlio	BO
Depending on company's security protocal it can be a	P	Localization/ Multilingual testing	AP	Travel & Logistics	BP

challenge to give access to outside testers but mostly this can be overcome e.g. with VPn etc.					
Diversity	Q	Lower down Cost of Quality / Cost-effectiveness	AQ	Unit Testing	BQ
Easy availability of testing resources	R	Manufacturing	AR	Usability	BR
Education	S	Mobile	AS	User Feedback	BS
Energy	T	N/A	AT	UserTesting	BT
ETL	U	No	AU	uTest	BU
Faster scalability	V	Not enough value for the cost; takes too long	AV	Utilities	BV
Functional	W	Omni-channel coverage - web	AW	UX	BW
The global pool of resources	X	Performance	AX	Your data is at risk expecially if you are looking at sensitive sectors such as financial services	BX
Hackerone / Intigriti	Y	Public Services	AY		
hard to say. Die out?	Z	Quality control	AZ		

4.2.3 Findings from the Research questionnaire

A total of thirty-two participants participated in this research. The research, its aim, and the questions in the questionnaire were shared withwith participants using Google forms based questionnaire.. Below is a deeper analysis carried out on each survey participant and their responses.

4.2.3.1 Crowdsourcing practices are utilised today in the software industry

Each participant had multiple choices to select from in the questionnaire. If a participant has worked on multiple domains, I am counting each domain as an independent response. Out of 32 participants about 10 have no information regarding the domain in which crowdsourcing can be utilization. Maximum responses were ten for education, insurance, retail, and healthcare domains. Eight responses for energy, utilities, resources, and high-tech domains, six for public services, banking and life sciences while 5 responses for consumer goods & distribution and travel & logistics. The minimum responses were four for the manufacturing domain (Figure 4.5).

Table 4.5: Breakdown of participant's responses in different domains for crowdsourcing utilization.

Domain	Number of response
Education	10
Insurance	10
Retail	10
Healthcare	10
Energy	8
Utilities	8
Resources	8
High-Tech	8

Consumer Goods & Distribution	5
Public Services	6
Banking	6
Life Sciences	6
Manufacturing	4
Travel & Logistics	5
N/A	10

Out of 32 participants, 30 suggested that functional is the type of testing while 22 and 23 respondents suggested that performance and usability respectively are the important type of testing for any software. For localization/ multilingual testing, security, unit testing and accessibility, the numbers of responses received were 14, 11, 10 and 2 respectively. The minimum response was one for manufacturing UX, ETL, ISO25000, Integration, API Testing and anything else we need - e.g. Integration testing (Figure 4.6).

Table 4.6: Breakdown of participant's responses for the type of testing.

Type of Testing	Number of response
Performance	22
Functional	30
Usability	23
Security	11
SRE	4
Localization/ Multilingual testing	14
UX	1
Unit Testing	10

Accessibility	2
Anything else we need - e.g. Integration	1
ETL	1
ISO25000	1
Integration	1
API Testing	1

Out of 32 participants, 6 participants use no testing platforms while 11 use, user testing platforms. Eight participants use uTest platforms while 5 use bug finder and Testbirds - www.testbirds.com, CrowdSprint, Passbrains, TesterWork, QAProvider respectively. Similarly, 3 participants use Testlio platforms while 2 used Bugcrowd platform. Other platforms such as 99tests, rainforest QA, Testers work, HackerOne / Intigriti, Selenium, ALM and Jira each by one respondent only. One respondent suggested that he built the crowd-testing platform for my ex-employer while another one suggested that he utilize the software for in-house or contractor-driven teams (Figure 4.7).

Table 4.7: Breakdown of participant's responses for the utilization.

Utilization	Number of response
User Testing	11
Testlio	3
uTest	8
99tests	1
I built the crowd testing platform for my ex employer	1
In house or contractor driven teams	1

Bug Finders	5
No	6
Rainforest QA	1
Testbirds - www.testbirds.com, CrowdSprint, Passbrains, TesterWork, QAProvider,	5
Bugcrowd	2
Testers work	1
Hackerone / Intigriti	1
Selenium	1
ALM	1
Jira	1

Of 32 participants, 16, 21, 20, 15 and 8 gave responses for performance, functional, usability, localization/ multilingual testing and security as a testing approach for software testing. Only two and one responses came for unit testing and SRE respectively as a testing approach for software testing (Figure 4.8).

Table 4.8: Breakdown of participant's responses for the types of testing approaches for software testing.

Types of testing approaches for software testing	Number of response
Performance	16
Functional	21
Usability	20
Security	8
SRE	1
Localization/ Multilingual testing	15

Unit Testing	2

4.2.3.2 Crowdsourcing practices on the quality and costs of software products.

The responses from the respondents suggested that the most important benefits of using crowdsourcing for software testing are the easy availability of testing resources, their diversity, global pool of resources, test coverage and their help in accelerating the testing cycle. About 15 respondents suggested that omnichannel coverage – web, mobile and in-store is the benefit of using crowdsourcing for software testing. While minimum responses came for responses such as 2nd income benefit for both employer and the tester, helps agile & ensures the ability to test in production not just in sandbox i.e. real testers, real devices, real payment methods etc, test coverage and user feedback (Table 4.9).

Table 4.9:Breakdown of participants' responses for the benefits of using crowdsourcing for software testing.

Benefits of using crowdsourcing for software testing	Number of response
Diversity	16
Omni-channel coverage - web	15
Mobile	15
In-store	15
Any other channels	14
Easy availability of testing resources	17
Global pool of resources	16
Faster scalability	9
Test coverage	16
Reduces lag related to Talent management cycle	7
Helps in accelerating the Testing cycle	16
2nd income benefit for both Employer and the Tester.	1

Helps Agile & ensures the ability to test in production not just in Sandbox i.e. real testers, real devices, real payment methods etc.	1
Test coverage	1
User feedback	1
Lower down Cost of Quality / Cost effectiveness	13
Helps in accelerating the Testing cycle	8

Table 4.9 suggests that the maximum use of crowdsourcing for software testing will evolve in the future for advanced testing tools and then for increased adoption and improved quality control. Only one respondent suggested that it is very hard to say about the use of crowdsourcing for software testing will evolve in the future.

Table 4.10: A breakdown of participants' responses to the use of crowdsourcing for software testing will evolve in the future.

Use of crowdsourcing for software testing will evolve in the future	Number of response
Improved quality control	17
Increased adoption	20
Advanced testing tools	21
hard to say. Die out?	1

Table 4.10 suggested that about 17 respondents recommend crowdsourcing for organizations considering using crowdsourcing for software testing while 12 respondents were in confusion towards its recommendation while remaining either didn't know or gave no response to the recommendation of crowdsourcing for organizations considering using

crowdsourcing for software testing. Except for all this, 12 respondents out of 32 gave recommendations for organizations considering using crowdsourcing for software testing. They recommend it by giving responses such as “To make a proper selection as per their type of operation, always looking into the budgeting and resources needed to put in place”; “List the problems you are trying to solve through crowd testing. Identify your use case(s) well. Identify the target market and potential users and choose your testers from your potential user base for more accurate feedback.”; Create a robust platform and governance model, redact the data and test units.”; “Start with WHY? Why would you use crowdsourcing? What does success look like and how would you recognize it? How is the business case for crowdsourcing better than for alternative solutions?”; “I still am in ambiguity about the ethnicity of crowd-sourcing concept in software testing. Each software is different and the way of testing methodology and testability is different. Every crowd who will be sourced might not have the same idea or Thoughts to perform testing in a particular area of testing. Which might impact the main idea of testing that product needs and development for the end user.”; “Define the objective clearly and share use cases you want crowd testers to validate. Clear and easy way to report issues”; “uTest (Functional / Usability / Localisation) Hackerone, Bugcrowd, Intigriti (Security)”; “Ubertesters”; “Should be mentioning the tasks based on certain features or flows. so that user can be focus centric”; “Use Testbirds :)”; “Global reachability, Gain hand on experience” and “the major arguments are "the major arguments are: available target group (UX), available devices (functional) and get more speed in development by avoiding long-term internal discussions about the right approach”.

Table 4.11: Breakdown of participants’ responses for the recommended crowdsourcing for organizations considering using crowdsourcing for software testing.

The use of crowdsourcing for software testing will evolve in the future	Number of response
Improved quality control	17
Increased adoption	20
Advanced testing tools	21
hard to say. Die out?	1

4.2.3.3 Primary software testing challenges do product development organisations face in building quality software products.

Of 32 participants, about 20 respondents believed that the most important challenge is security risks, 13, 11 and 10 respondents believed that quality control, lack of control and logistics respectively. While only one response came for each response such as your data is at risk especially if you are looking at sensitive sectors such as financial services; not enough value for the cost; takes too long and depending on company security protocol it can be a challenge to give access to outside testers but mostly this can be overcome e.g. with VPn's etc as primary software testing challenges do product development organisations face in building quality software products (Table 4.12).

Table 4.12: Breakdown of participants’ responses for the primary software testing challenges product development organisations face in building quality software products.

Primary software testing challenges do product development organisations face in building quality software products.	Number of response
Lack of control	11
Quality control	13
Logistics	10
Security risks	20

Your data is at risks especially if you are looking at sensitive sectors such as financial services	1
Not enough value for the cost	1
Takes too long	1
Depending on company security protocal it can be a challenge to give access to outside testers but mostly this can be overcome e.g. with VPn's etc.	1

4.2.3.4 The key factors which impact software quality and quality costs.

From 32 participants, about 21 responses came for quality control mechanisms. About twenty and 17 respondents believed that testing expertise and experience and testing coverage and scalability respectively are the important key factors which impact software quality and quality costs. Only one respondent believed that the quality and value of the data and the cost of processing is a key factor which impacts software quality and quality costs.

Table 4.13: Breakdown of participant's responses for the key factors which impact software quality and quality costs.

Key factors which impact software quality and quality costs	Domain Code	Number of response
Quality control mechanisms	BA	21
Testing expertise and experience	BN	20
Testing coverage and scalability	BM	17
Look at the likely quality and value of the data and the cost of processing it.	C	1

CHAPTER V:

DISCUSSION

5.1 Discussion of Results

Software development companies strive to deliver high-quality products while minimizing costs. In recent years crowdsourcing has emerged as a valuable technique for achieving this goal. By harnessing the collective intelligence and efforts of a crowd these companies can tap into a vast pool of resources to perform software testing efficiently and effectively. This paper explores the business strategies and practices employed by IT product development companies to build high-quality software products at a low cost of quality through crowdsourcing techniques.

Crowdsourced software testing (CST) is a nascent field within the realm of software engineering. Crowdsourced software testing refers to the practice of conducting software testing operations by leveraging a diverse and extensive collection of individuals, typically characterized by their large numbers and varied composition. The CST (Crowdsourced Software Testing) methodology facilitates the development of software of superior quality by involving a large number of individuals in the process of software testing across a wide range of scenarios. The aforementioned methodology can complement conventional software testing practices, which usually involve a limited number of testers operating within a software development entity (Alyahya & Alsayyari, 2020; Alyahya, 2022). However, only 37% of the respondents reported using crowdsourcing for software testing. The utilization of crowdsourcing approaches is observed in diverse domains today, extending beyond the software sector (Hosseini et al., 2015; Wang et al., 2018). Similar findings were also observed in this study, where a majority of participants indicated that crowdsourcing practices were predominantly employed across a wide range of domains. These domains include, but are not limited to, education, insurance, retail, healthcare, energy, utilities, resources, high-tech, public services, banking, life sciences, consumer goods and distribution, travel and logistics, as well as manufacturing.

Software testing plays a crucial role in ensuring the quality of software development. It serves as a means of verifying software quality and facilitating its attainment by addressing the needs and concerns of various stakeholders involved in the application, including end-users, developers, software designers, and software testers. Cognitive Task Analysis (CTA) was applied to evaluate graphical user interfaces and mobile applications, demonstrating their technical feasibility and a satisfactory level of reliability (Leichts & Leimeister, 2016).

The predominant forms of testing employed by CST (Computer Science and Technology) encompass localization/multilingual testing, security testing, unit testing, accessibility testing, manufacturing UX testing, ETL testing, ISO25000 testing, integration testing, and API testing.

Usability testing holds significant importance in ensuring the effectiveness of a website. However, the associated costs and time requirements often pose barriers, limiting its implementation in the continuous creation and maintenance of websites (Liu et al., 2012). Nevertheless, there has been a shift towards adopting a cost-benefit analysis framework in the usability field, placing greater emphasis on return on investment (ROI) when considering the integration of usability evaluations into the product or website development process (Bias and Mayhew, 2005). Balancing limited resources between usability evaluation and design necessitates a deliberate compromise, as evidenced by the costs associated with usability tests (Spool & Schroeder, 2001).

CST has demonstrated its effectiveness in various forms of testing, including usability testing, functional testing, and performance testing. The execution of these tasks can be achieved by utilizing workflow systems like those provided by uTest and Figure Eight. These platforms act as intermediaries between clients, such as enterprises seeking software testing services, and a registered crowd of individuals ready to perform testing tasks. The platforms receive requests from clients and then distribute these requests to the registered crowd, streamlining the task allocation process. Therefore, the majority of participants in this research use uTest platforms, while others utilize bug-finding services like Testbirds (www.testbirds.com), CrowdSprint, Passbrains, TesterWork, QAProvider, and various other platforms.

Previous research has also highlighted that software platforms like uTest and CrowdFlower have proven their ability to facilitate Crowd-sourced software testing. These platforms enable requesters, such as software organizations seeking assistance with product testing, to submit their testing needs to a large workforce. The employees then carry out testing activities and provide test reports, typically receiving relatively lower compensation (Alyahya, 2022; Imtiaz et al., 2019). The proliferation of these platforms has led to an increase in CST activities, piquing the interest of the academic community to further investigate CST, introduce enhancements, and establish reliable evaluations, thus advancing this emerging software testing technique.

In the field of Computer Science and Technology (CST), several distinct testing methodologies are employed for software evaluation. These methodologies encompass performance testing, functional testing, usability testing, security testing, Site Reliability Engineering (SRE), localization/multilingual testing, and unit testing. Consequently, an increasing number of organizations have turned to crowdsourcing as a means of constructing software systems in a cost-effective and time-efficient manner, while also addressing concerns related to liability, quality, and compliance with legal requirements (Sarı et al., 2019).

5.2 Discussion of Research Question One

What are the business strategies and practices of IT Product development companies to build high-quality software products at the low cost of quality using Crowdsourcing techniques?

Crowdsourcing in software testing offers several distinct advantages over traditional testing methods, ultimately leading to significant cost savings in IT product development.

Reduced Labor Costs:

- Pay as you go: You only pay for the number of tests you run on crowdsourcing platforms, instead of hiring costly in-house testers and paying for their salaries and benefits.
- Scale on demand: You can adjust the level of testing expertise you need according to your project needs. You don’t have to keep permanent resources when they are not needed. No need to maintain permanent resources during lean periods.

- Global talent pool: Access to a vast pool of talented and experienced freelance QA professionals worldwide, enabling you to find the right skillset for specific testing needs at competitive rates.

Faster Time to Market:

- Parallel testing: Conduct tests simultaneously on various devices and platforms, thanks to the readily available crowd of diverse professionals. This significantly reduces overall testing time compared to sequential in-house testing.
- 24/7 availability: Leverage the global nature of the crowdsourcing pool to continue testing around the clock, maximizing available hours and accelerating your release cycle.

Additional Cost Savings:

- Lower travel and infrastructure costs: Avoid the need for physical infrastructure such as testing labs and equipment, which can be costly to upkeep. Easier project management: Crowdsourcing platforms usually take care of tester hiring, training, and project management, lowering your workload and related expenses.
- Early bug detection: Identifying and fixing bugs early in the development process saves time and resources compared to fixing them post-release.

Crowdsourcing in software testing isn't just about cutting costs; it can also significantly contribute to higher-quality IT products. Here's how:

Diverse Testing Pool:

- Real-world user perspectives: Accessing testers from diverse backgrounds, demographics, and locations ensures your software is tested by the actual users you're targeting. This uncovers edge cases and usability issues that might be missed by in-house teams with similar backgrounds.
- Broader device and platform coverage: With a global pool of testers, you can test your software on a wider range of devices and platforms than your in-house

resources might possess. This catches compatibility issues and ensures smooth functioning across diverse setups.

- Minimized bias and blind spots: In-house teams can accidentally develop blind spots and biases in favor of their own preferences. Crowdsourcing introduces new viewpoints and removes such biases, resulting in more complete and impartial testing.
- Targeted expertise: Get access to expert testers for different domains like security, accessibility, or mobile testing, achieving thorough coverage without the need to employ in-house specialists.
- Data-driven improvements: Examine data from user interactions during testing to comprehend usage patterns and pinpoint potential areas for enhancement, resulting in better software quality and user satisfaction.

Increased Testing Depth and Breadth:

- Larger tester volume: Crowdsourcing allows you to scale your testing team up quickly, enabling simultaneous testing on various features and scenarios. This leads to increased coverage and a deeper dive into specific functionalities.
- Specialized expertise: Platforms offer access to testers with specific expertise in areas like security, accessibility, or performance testing. This ensures thorough testing in critical areas that might be neglected with limited in-house resources.
- Continuous feedback loop: With testers readily available on-demand, you can get continuous feedback throughout the development cycle. This enables quick changes and bug fixes in real-time, avoiding issues from piling up and becoming expensive to fix later.

Data-Driven Insights and Improved User Experience:

- Rich user data: Analyze data from user interactions during testing to understand usage patterns, pain points, and feature expectations. This data-driven approach helps you prioritize improvements and tailor your software for optimal user experience.

- Early bug detection and resolution: Identifying and fixing bugs early in the development process minimizes their impact on users and reduces costs associated with post-release fixes. Crowdsourcing's testing depth and breadth lead to earlier bug discovery.
- Reduced user churn and increased satisfaction: When software is well-tested, user frustrations and churn rates decrease. Crowdsourcing helps eliminate bugs and usability issues, leading to a higher-quality product that users enjoy using.
- Remember: Crowdsourcing success relies on choosing the right testing platform, clearly defining test goals, and providing testers with accurate instructions and tools. By implementing these elements effectively, you can harness the power of a diverse testing pool to significantly improve the quality and user experience of your IT product.

While cost reduction is a strong benefit, the impact of crowdsourcing on software quality shouldn't be underestimated. Consider it a double-win strategy: improved quality while optimizing development costs.

5.2.1 Utilisation of a Diverse Pool of Testers

To craft top-notch software, companies tap into the wisdom of the crowds through platforms that connect them with a diverse pool of testers. This mosaic of perspectives – different skills, experiences, and backgrounds – enriches the testing process (Xu 2016). Think of it like examining a diamond from every angle; diverse testers uncover unique issues and perspectives that might elude an individual. This comprehensive testing ensures the software caters to a wider user base, not just a select few.

The digital age poses many challenges for software development. Traditional testing approaches, which rely on small in-house teams, struggle to cope with the dynamic needs of users and the complexity of modern apps. Crowdsourcing offers a solution to this problem. By engaging a large and varied pool of testers from different locations, you can benefit from a wealth of perspectives, expertise, and devices, which results in improved software, and quicker too.

However, just opening the floodgates to a multitude of testers isn't sufficient. Creating a strategic approach to leverage a diverse crowd requires thoughtful planning and execution. Here's a human-friendly guide on how to navigate the exciting world of crowdsourced testing:

5.2.1.1 Understand Your Needs:

- What kind of testing do you require? (Functional, usability, compatibility, security?)
- What level of expertise is needed? (Novice for basic tasks, experienced in complex scenarios?)
- Which devices and platforms are relevant? (Crowdsourcing provides a vast array of comprehensive cross-platform coverage.)

5.2.1.2 Assemble Your Crowd:

- Choose the right platform that aligns with your testing needs and target audience.
- Create captivating campaigns and concise test instructions, with attractive rewards to draw and keep high-quality testers.
- Promote diversity by reaching out to testers from various origins, groups, and places for new insights.

5.2.1.3 Manage Your Crowd Effectively:

- Provide detailed test instructions and reporting guidelines, encouraging open communication and prompt addressing of tester queries.
- Implement robust review processes to ensure accuracy and reliability of test results.
- Leverage data insights from test results to identify patterns, prioritize bugs, and improve your software.

5.2.1.4 Enhance Your Strategy:

- Keep improving by monitoring outcomes, collecting responses, and adjusting your method based on lessons.
- Ensure robust security measures and communicate data privacy policies clearly to testers and users.
- Build a community among testers, recognizing top performers, offering growth opportunities, and acknowledging their contributions.

5.2.1.5 Benefits of a Diverse Testing Pool:

- Improved software quality with diverse perspectives uncovering hidden bugs and usability issues.
- Faster testing cycles by conducting tests simultaneously on various devices and platforms.
- Cost-effectiveness, especially for geographically dispersed user bases.
- Increased market reach by gaining insights into user behaviour across different demographics and regions.

Remember, building a successful crowdsourced testing strategy is an ongoing journey. By prioritizing your needs, recruiting and overseeing a varied group of testers, and constantly refining your method, you can tap into the enormous power of crowdsourcing to produce high-quality software, quicker and more effectively. So, welcome the strength of the crowd, and see your product rise to new levels!

5.2.2 Boosting Participation with a Touch of Play

To keep the crowd engaged and motivated, companies turn to gamification. By injecting game-like elements, software testing transforms into a stimulating experience, attracting more participants and sustaining their interest (Gupta & Gupta 2016). Think rewards, leaderboards, and badges - little nudges that encourage testers to provide detailed and accurate feedback. This healthy competition drives participation and ultimately fuels better software quality.

Crowdsourcing gives you a unique chance to access a worldwide group of testers, guaranteeing thorough testing and varied viewpoints. But as with any army, maintaining your crowd's interest and enthusiasm is essential for peak performance. So, how do you turn a potentially scattered legion into a cohesive unit driving software excellence? Here are some key strategies to keep your testers fired up:

5.2.2.1 Embrace Gamification:

- Scores, medals, and rankings: Create a fun and competitive atmosphere with gamified elements. Recognize the best performers, mark achievements, and appreciate individual efforts.
- Missions and tasks: Split testing activities into small and manageable tasks with clear objectives. Offer bonus points for tackling complex bugs or uncovering usability gems.

- Leveling up and specialization: Allow testers to progress through ranks based on their performance and expertise. Offer specialized test paths catering to individual interests and skills.

5.2.2.2 Foster Community and Connection:

- Open communication channels: Create forums, discussion boards, or online chat rooms where testers can share experiences, tips, and even humour. Encourage open communication and collaboration.
- Regular feedback and recognition: Provide timely feedback on tester performance, highlight valuable contributions, and showcase success stories. Recognize top performers publicly and acknowledge individual efforts.
- Virtual events and meetups: Organize online social events or webinars to foster a sense of community and belonging. Facilitate interaction between testers and the development team, building a sense of shared purpose.

5.2.2.3 Make it Meaningful and Rewarding:

- Clear purpose and impact: Explain how their testing efforts contribute to the bigger picture, emphasizing the positive impact on real users. This fosters a sense of ownership and purpose.
- Flexible work arrangements: Offer flexible testing hours and project options to cater to individual schedules and preferences.
- Competitive compensation and incentives: Go beyond basic pay. Offer bonuses for high-quality testing, early bug discovery, or participation in special projects.

5.2.2.4 Ongoing Learning and Improvement:

- Learning and growth opportunities: Offer tools and chances for testers to acquire new abilities, keep abreast of testing techniques, and focus on particular domains. Knowledge exchange and guidance: Motivate seasoned testers to guide newer ones, promoting knowledge sharing and building a culture of constant learning.
- Career advancement paths: Outline potential career progression opportunities within your crowdsourcing platform or even suggest connections with relevant companies.

5.2.2.5 Treat them like Partners, not Outsiders:

- Transparency and open communication: Keep testers informed about project updates, development progress, and key decisions. Value their feedback and involve them in the development process whenever possible.
- Respect and appreciation: Treat testers with respect and courtesy, ensuring a positive and inclusive testing environment. Appreciate their contributions and value their individual perspectives.
- Regular interaction with the core team: Facilitate interactions between testers and the development team. This not only builds trust but also allows testers to understand the context of their work and feel like valued members of the team.

By implementing these strategies, you can transform your crowdsourced testing pool from a scattered group of individuals into a motivated and engaged army, driving software quality and innovation. Remember, a happy and fulfilled testing army leads to a software kingdom free from bugs and user frustrations. So, invest in your crowd, keep them engaged, and watch your software soar!

5.2.3 Providing a Clear Roadmap

Effective crowdsourced testing hinges on crystal-clear guidelines and test cases provided by the company. Think of these as a detailed treasure map for the testers. Well-defined guidelines eliminate ambiguity and ensure everyone understands the software's goals and testing objectives, leading to high-quality feedback (Jain & George 2015). Predefined test cases further guide the evaluation process, making it consistent and efficient, while pinpointing potential bugs and issues.

Clearly established guidelines are essential to the effectiveness of crowdsourcing testing. Equipped with a good understanding of the purpose of the software, test objectives and standards testers are able to provide information that is not only high in quality but also highly relevant and as a result, this for the software as a whole improves and enhances the user experience.

Here's how well-defined guidelines can be advantageous in crowdsourced testing:

- Reduced Ambiguity and Confusion: Testers are less prone to misinterpret instructions, reducing the likelihood of focusing on the wrong aspects of the software. This prevents wasted time and effort on both ends.
- Improved Consistency and Accuracy of Feedback: Clear understanding of expectations increases the likelihood of testers providing consistent and accurate

feedback. This, in turn, facilitates developers in identifying and addressing bugs more effectively.

- Efficient Testing Process: Clear guidelines streamline the testing process by ensuring that testers concentrate on the most crucial tasks. This can result in expedited testing cycles and reduced costs.
- Increased Tester Satisfaction: Testers who comprehend their roles and have the necessary resources tend to be more satisfied with their experience. This heightened satisfaction can lead to increased engagement and retention.

In addition to the mentioned benefits, well-defined guidelines also contribute to:

- Reducing Bias in Testing: Clear criteria in guidelines help mitigate the influence of personal preferences or biases on testers, ensuring a more objective evaluation of the software.
- Improving Communication Between Testers and Developers: Shared practices facilitate effective communication between testers and developers, fostering a collaborative and productive testing process.

In conclusion, clearly defined guidelines play an important role in the success of crowdsourcing research. Taking the time to develop clear and concise tutorials improves testers' success and ensures high-quality information needed for ongoing software development

5.2.4 Keeping the Lines of Communication Open

Clear communication is the golden thread in successful crowdsourced testing. Companies must establish open channels for testers to ask questions, seek clarifications, and offer feedback throughout the process (Yang & Zhang 2015). Think of it as a continuous feedback loop, ironing out any uncertainties and ensuring testers fully grasp the testing requirements and objectives.

Crowdsourcing has become an effective strategy for IT product development companies to build high-quality software products at a low cost of quality. By leveraging a diverse crowd implementing gamification techniques providing clear guidelines enabling continuous communication and implementing quality assurance mechanisms these companies can harness the power of crowdsourcing for efficient and effective software testing. The adoption of these strategies and practices allows companies to tap into a vast pool of resources while minimizing costs resulting in improved software quality and customer satisfaction.

5.2 Discussion of Research Question Two

Crowdsourcing practices on the quality and costs of software products.

The utilization of crowdsourcing offers numerous advantages, which are also relevant in the domain of software development. One of the primary benefits of using crowdsourcing for software testing is the easy access to testing resources, known for their diversity and global reach. This approach allows for extensive test coverage and speeds up the testing cycle. Alongside its advantages in omni-channel coverage, encompassing web, mobile, and in-store platforms, employing crowdsourcing for software testing provides several benefits. It also brings cost savings for both the employer and the tester, enhances agility, and ensures the ability to conduct tests in a production environment rather than solely in a sandbox. This involves engaging real testers, using actual devices, employing genuine payment methods, and ensuring comprehensive test coverage and user feedback.

Furthermore, the objectives of crowdsourcing extend to various aspects, including promoting solution diversity, fostering idea generation, encouraging broader participation, enhancing marketing efforts, and supporting participant education, especially in terms of encouraging individuals to use or gain proficiency in specific tools. To optimize the outcomes of software crowdsourcing, enterprises have the opportunity to leverage cloud infrastructure. This can expedite the establishment of the development environment and facilitate distributed and large-scale collaboration within a highly dynamic community (Asiegbu Baldwin et al., 2017).

Crowdsourcing offers several significant advantages, including cost savings, accelerated Time-to-Market, improved quality, enhanced creativity and open innovation, and talent identification. Cost reduction occurs by lowering development expenses for developers and avoiding additional cost overheads typically associated with hiring private software developers. Crowdsourcing organizers can potentially obtain software at a significantly lower cost, as compensation is only awarded to individuals or teams who successfully complete the software development task, and they may even accept payment below market rates, prioritizing reputation rewards over monetary compensation.

Reducing time-to-market is facilitated by leveraging a network of competent individuals capable of efficiently progressing through various software development phases. Parallel development on projects can occur, with many individuals willing to work during weekends. Enhanced quality is achieved through extensive participation, providing access to a diverse and skilled pool of development professionals who voluntarily engage based on their requisite skills. These individuals participate in competitions where the best software is selected as the "winner." Proficient individuals strive to outdo their peers by presenting innovative ideas, designs, code, or testing. The presence of a vast and diverse network of qualified experts enables the exploration of a wide range of innovative solutions and the adoption of the most optimal outcomes.

Talent identification involves assessing and recognizing individuals with exceptional skills and abilities in a specific domain. In software development, organizers may discover talented developers based on their demonstrated achievements in competitive endeavours (Stol & Fitzgerald, 2014).

The future of crowdsourcing in software testing is expected to witness significant advancements in the utilization of advanced testing techniques, leading to increased adoption rates and enhanced quality control measures. The majority of participants expressed optimism about the future of crowdsourcing in software testing, with only one participant expressing uncertainty regarding its future trajectory.

Primary software testing challenges do product development organisations face in building quality software products

Product development organizations have many issues when it comes to software testing, which is crucial for ensuring the production of high-quality software products. The primary challenges faced by product development organizations in building quality software products include security risks, quality control, lack of control, and logistics. These challenges are particularly significant in sensitive sectors such as financial services, where the risk to data security is high. Additionally, there is often a lack of perceived value for the cost associated with software testing. Furthermore, the time required for testing can be excessive, and granting access to external testers can be challenging due to company security protocols. However, these challenges can be mitigated through the use of virtual private networks (VPNs) and other appropriate measures. The assessment of talent and experience, as well as the evaluation of coverage and scalability, are critical elements that influence the quality of software and its associated expenses. Among the respondents, it was found that only one individual held the belief that the quality and value of the data, as well as the cost of processing, had a significant role in influencing software quality and quality costs.

The key factors which impact software quality and quality costs

Furthermore, it's important to note that our research specifically focused on exploratory functional testing. The applicability of our findings to other forms of testing, such as usability testing or performance testing, remains uncertain and requires further investigation. Our study's results suggest that testing tasks requiring critical thinking and in-depth analysis are more likely to benefit from collaborative testing approaches. On the other hand, the micro-tasking paradigm has shown effectiveness in testing tasks that are divisible and typically exhibit minimal complexity.

To illustrate a scenario where collaborative testing may not be suitable, consider the following example: Many software development firms utilize crowdsourcing methods to ensure the satisfactory functionality of their systems across various devices and geographical regions worldwide. However, in situations where the system relies solely on a user base with specific hardware configurations or residing in specific geographical locations, the use of collaborative testing may not yield noticeable benefits. It's essential to tailor the testing approach to the specific requirements and complexities of the task at hand to optimize its effectiveness.

5.4 Path forward: A base framework for crowdsourcing

I have presented both benefits and challenges associated with crowdsourcing with a focus on Software Testing and opportunities for software product development.

I would like to discuss a base framework that can be utilised by various organisations that seek benefits associated with Crowdsourcing and also address the challenges associated with Crowdsourcing.

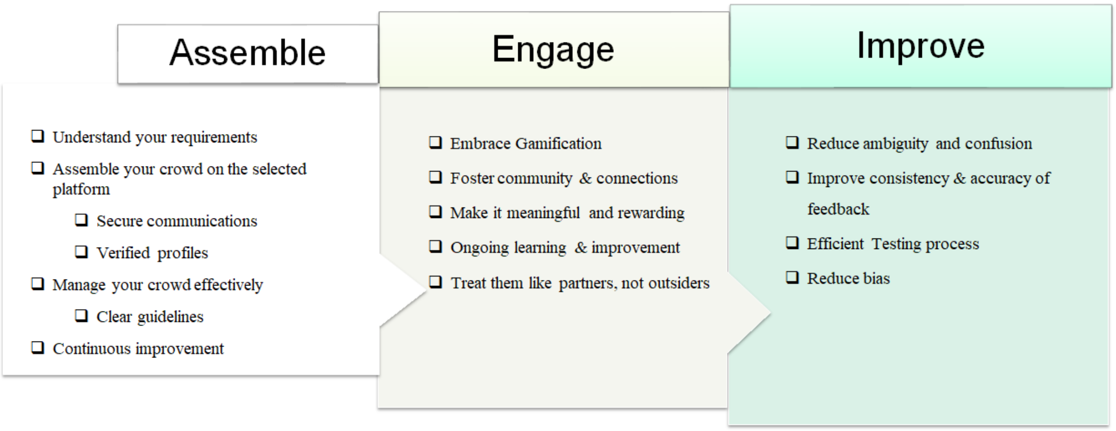
Assemble
Understand your requirements
Assemble your crowd on the selected platform
Secure communications
Verified profiles
Manage your crowd effectively
Clear guidelines
Continuous improvement
Engage
Embrace Gamification
Foster community & connections
Make it meaningful and rewarding
Ongoing learning & improvement
Treat them like partners, not outsiders
Improve
Reduce ambiguity and confusion
Improve consistency & accuracy of feedback
Efficient Testing process
Reduce bias

Figure 5.1 Experience of respondents in the software testing field.

5.4.1 Assemble

- Understand Your Requirements: Before initiating your crowdsourcing project, it's crucial to thoroughly understand your requirements, objectives, and desired outcomes. This involves identifying the specific tasks or challenges you need assistance with and defining clear criteria for success.

- Assemble Your Crowd on the Selected Platform: Choose the appropriate crowdsourcing platform based on your project needs and target audience. Once selected, leverage the platform's tools and features to attract and engage potential contributors who possess the skills and expertise required for your tasks.
  - Secure Communications: Establish secure communication channels to facilitate interaction between project stakeholders, including organizers and participants. Utilize encrypted messaging systems or dedicated platforms with built-in security measures to protect sensitive information and ensure confidentiality throughout the collaboration process.
  - Verified Profiles: Implement a verification process to authenticate the identities and qualifications of crowd members. Require participants to provide verified credentials or undergo screening procedures to ensure they meet the necessary criteria and can contribute effectively to the project.
- Manage Your Crowd Effectively: Employ efficient crowd management strategies to oversee participant activities, monitor progress, and address any issues or concerns that may arise. Utilize project management tools, communication platforms, and team collaboration techniques to coordinate tasks and maintain productivity across the crowd.
  - Clear Guidelines: Provide clear and comprehensive guidelines outlining the objectives, expectations, and rules of engagement for participants. Clearly

communicate project requirements, deadlines, quality standards, and evaluation criteria to ensure everyone understands their roles and responsibilities.

- Continuous Improvement: Foster a culture of continuous improvement by soliciting feedback from participants and stakeholders throughout the project lifecycle. Encourage open communication, actively listen to suggestions, and implement iterative refinements to enhance processes, address challenges, and optimize outcomes over time.

5.4.2 Engage

- Embrace Gamification: Incorporate elements of gamification into your crowdsourcing project to enhance engagement, motivation, and participation. Utilize game-like features such as challenges, leaderboards, badges, and rewards to make tasks more enjoyable and stimulating for contributors.
- Foster Community & Connections: Cultivate a sense of community among participants by encouraging collaboration, networking, and relationship-building. Provide opportunities for individuals to connect, share insights, and form meaningful connections that extend beyond the confines of the project.
- Make It Meaningful and Rewarding: Ensure that tasks and contributions are meaningful and aligned with participants' interests, passions, and aspirations. Recognize and reward individuals for their efforts, achievements, and contributions, whether through monetary incentives, recognition, or opportunities for personal and professional growth.

- Ongoing Learning & Improvement: Promote a culture of continuous learning and improvement by offering resources, training, and support to participants throughout the project lifecycle. Provide access to educational materials, workshops, and mentorship opportunities that enable individuals to enhance their skills, expand their knowledge, and achieve their goals.
- Treat Them Like Partners, Not Outsiders: View participants as valued partners and collaborators rather than mere contributors or outsiders. Involve them in decision-making processes, solicit their input and feedback, and recognize their expertise, insights, and contributions as integral to the success of the project. Foster a sense of ownership and investment in the outcomes of the collaboration.

5.4.3 Improve

- Reduce Ambiguity and Confusion: Minimize ambiguity and confusion by providing clear instructions, guidelines, and expectations to participants. Clearly define tasks, objectives, and criteria for success, and offer clarification or additional support as needed to ensure everyone understands their roles and responsibilities.
- Improve Consistency & Accuracy of Feedback: Enhance the consistency and accuracy of feedback by implementing standardized evaluation criteria and assessment methods. Provide training or guidelines to evaluators to ensure they apply feedback consistently and objectively and offer mechanisms for validation or calibration to maintain quality and reliability.
- Efficient Testing Process: Streamline the testing process to improve efficiency and productivity while maintaining quality and thoroughness. Utilize automation tools,

test management platforms, and standardized protocols to expedite testing procedures, identify defects, and track progress in a systematic and organized manner.

- Reduce Bias: Mitigate bias by implementing measures to ensure fairness, objectivity, and impartiality throughout the testing process. Establish blind or double-blind testing protocols, anonymize data to prevent bias, and provide diversity training to evaluators to promote inclusivity and reduce the influence of personal biases on decision-making.

CHAPTER VI:
SUMMARY, IMPLICATIONS, AND RECOMMENDATIONS

6.1 Summary

Our research demonstrated that crowdsourcing offers significant potential for achieving high-quality software testing at lower costs. By leveraging the advantages of a diverse testing pool, faster testing cycles, and cost-effectiveness, while addressing the challenges of quality control, tester engagement, and security, software development teams can embrace this innovative approach to deliver superior software and enhance user experience. Further research exploring advanced strategies, such as AI-powered crowdsourcing and gamification techniques, can contribute to maximizing the benefits and minimizing the challenges of this promising approach to software testing.

6.2 Implications

There are some constraints associated with this study that may have influenced the outcomes of this research. To begin with, the selection of interviewees for this study was not restricted to any specific number of participants. It was kept open for a longer period in the public domain to take inputs from various industry experts having been exposed to the practices of CrowdSource for the Testing domain. I still could not collect data from 100s of respondents as this is a niche industry and it is not possible to find people with direct exposure and expertise to CrowdSource for the Testing domain. This limitation raises the possibility that alternative perspectives that the researcher was unable to include may exist.

Additionally, the participants were specifically chosen to represent diverse business models prevalent in the IT Industry. It is crucial to note that this study's findings apply exclusively to the Testing areas covered in the interview process.

The geographical scope of this study was intentionally not limited to any country. Consequently, I could collect data relevant to more than one region worldwide. Still, Therefore, it is essential to consider the context before applying the findings of this research in different geographical contexts.

Lastly, this research does not account for or distinguish between factors such as the scale of profitability, operational efficiency achieved or the project size and scope under study. These factors may be deemed significant for future research endeavours.

6.3 Recommendations for Future Research

Crowdsourcing is a method that uses the collective knowledge and know-how of a big, allotted institution of human beings to clear up complex issues or responsibilities Crowdsourcing has been used in areas as diverse as innovation, design, statistics collection, and software program improvement. Crowdsourcing for checking out is largely a new technique that uses crowdsourced specialists to carry out software checking out tasks, inclusive of usability testing, usability trying out, safety checking out, and overall performance testing

Crowdsourced testing has many potential blessings, inclusive of decreased trying out costs, increased test coverage, increased test productiveness, and quicker checking out cycles but it additionally faces many challenges, together with managing the satisfactory and variety of crowd-sourced personnel, ensuring the validity and validity of crowd-sourced testing outcomes Source- Integrating trying out with conventional checking out techniques and therefore crowdsourcing checking out is an emerging and promising place of research that calls for similarly attention and research from students and practitioners.

In this section, I review some existing research studies on crowdsourcing testing and identify some future research directions and opportunities in this area. I organized our discussion into four main themes: crowdsourcing testing information products, crowdsourcing from sources and channels, crowdsourcing quality assurance, and mass demand strategy and social impact.

6.3.1 Crowdsourced Test Reports Processing

One of the main challenges of crowdsourcing for testing is how to effectively process and integrate the large number of test reports submitted by crowdsourced workers. These test reports may contain duplicate, irrelevant, or low-quality information, which makes it difficult for developers to review and use them. Therefore, several research studies have proposed methods and techniques to address this challenge, such as:

- Duplicated reports detection: This aims to identify and remove the test reports that report the same bug or issue, to reduce the redundancy and noise in the test reports set. For example, Wang et al. [1] proposed a method based on natural language processing and machine learning to detect duplicated reports in crowdsourced testing.
- Test reports aggregation and classification: This aims to group and label the test reports based on their similarity or relevance, to facilitate the analysis and prioritization of the test reports. For example, Zhao and Zhu [2] proposed a method based on topic modelling and clustering to aggregate and classify test reports in crowdsourced testing.
- Priority ranking: This aims to rank the test reports based on their importance or urgency, to help developers focus on the most critical or valuable test reports. For

instance, Chen et al. Three proposed a multi-standards choice-based technique for comparing take a look at reviews in crowd-sourced testing.

- Report precis: This targets to offer a concise and comprehensive precis of the take-a-look-at document, providing an outline of key findings and insights from the test reports. For instance, Zhang et al. 4 proposed a technique based totally on information collection and sensitivity evaluation to summarize check reports in crowdsourcing checking.

6.3.1.1 Future research directions in this theme include:

- Developing more advanced and robust methods and techniques for crowdsourced test report processing, such as using deep learning, natural language generation, and semantic analysis.
- Evaluating and comparing the effectiveness and efficiency of different methods and techniques for crowdsourced test reports processing, such as using empirical studies, experiments, and benchmarks.
- Exploring the impact of crowdsourced test reports processing on the software development process and outcomes, such as the bug fixing time, software quality, and user satisfaction.

6.3.3 Crowdsourced Testing Platforms and Methods

Another challenge in crowdsourcing for testing is how to design and implement crowdsourcing processes and processes such as division of labor, job allocation, job creation, and job search. Thus, many research studies have proposed frameworks and models to guide the design and implementation of crowdsourced trial designs and procedures, e.g.

- Task decomposition: This refers to how to break down the software testing task into smaller and manageable subtasks, to match the capabilities and preferences of crowdsourced workers. For example, Mao et al. proposed a framework based on the software testing life cycle to decompose the software testing task into different phases and levels.
- Task allocation: This refers to how to assign the software testing subtasks to the appropriate crowdsourced workers, to optimize the resource utilization and task completion. For example, Li et al. proposed a model based on the worker skill and task difficulty to allocate the software testing subtasks to the most suitable workers.
- Task execution: This refers to how to support and facilitate the crowdsourced workers to perform the software testing subtasks, to improve the task efficiency and effectiveness. For example, Chen et al. proposed a method based on gamification and feedback to motivate and guide the crowdsourced workers to execute the software testing subtasks.
- Task evaluation: This refers to how to assess and reward the crowdsourced workers for their software testing subtasks, to ensure the task quality and fairness. For example, Wang et al. proposed a method based on reputation and incentive to evaluate and reward the crowdsourced workers for their software testing subtasks.

6.3.3.1 Future research directions in this area include:

- To create scalable and adaptable crowd-testing techniques and tactics, such as employing words, statistics, and self-learning techniques.
- Examine and contrast the advantages and outcomes of various crowdsourcing techniques and approaches, including user research, questionnaires, and interviews.

- Examine the ways in which you might integrate and synchronize the tactics and methods of crowd-testing with those of automated, model-based, and agile software testing.

6.3.4 Crowdsourced Testing - Ensuring Quality Assurance

A further challenge of crowdsourcing for testing is how to ensure and improve the quality of crowdsourced testing, such as the quality of crowdsourced workers, test results, and testing process. These quality aspects affect the reliability and validity of crowdsourced testing, as well as the trust and satisfaction of developers and users. Therefore, several research studies have proposed measures and techniques to address this challenge, such as:

- Competence of Crowdsourced Workers: This refers to the abilities and knowledge of crowdsourced workers in performing software testing tasks, including their testing skills, expertise, and experience. For instance, Stolee and Elbaum introduced a technique based on worker profiles and test case generation to assess and enhance the quality of crowdsourced workers.
- Quality of Test Results: This concerns the precision and comprehensiveness of the outcomes produced by crowdsourced workers, encompassing elements such as test cases, test reports, and test feedback. Zhang et al. proposed a technique grounded in the test oracle and test coverage to gauge and enhance the quality of test results.
- The effectiveness of the testing process is examined, taking into account factors like testing time, cost, and effort. It is centred on how well crowdsourcing workers execute the testing methods. In order to assess and enhance the calibre of the testing procedure, Chen et al. introduced a method centred on the test plan and test strategy.

In addressing these dimensions of quality assurance in crowdsourced testing, researchers aim to establish robust methodologies that ensure the competence of workers, the accuracy of results, and the overall efficiency of the testing process. These efforts contribute to refining crowdsourced testing practices and promoting reliable and effective software quality assurance.

6.3.4.1 Future research directions in this theme include:

- Developing more comprehensive and rigorous measures and techniques for crowdsourced testing quality assurance, such as using quality models, quality standards, and quality metrics.
- Evaluating and comparing the impact and trade-offs of different measures and techniques for crowdsourced testing quality assurance, such as using quality analysis, quality evaluation, and quality optimization.
- Exploring the factors and conditions that influence the quality of crowdsourced testing, such as worker diversity, task complexity, and platform design.

6.3.5 Crowdsourced Ethics and Social Impact Assessment

The last hurdle in using crowdsourcing for testing is figuring out how to handle the moral and social concerns that come up, such as the accountability, security, privacy, and fairness of crowdsourced Testers. These problems have an impact on crowdsourced testing's credibility and trustworthiness as well as its social and financial advantages. Therefore, several research studies have proposed guidelines and principles to address this challenge, such as:

- Privacy is the safeguarding and observance of the private and sensitive data of users, developers, and crowdsourcing Testers, including their location, identity, and preferences. To solve the privacy difficulties in crowdsourced testing, for instance, Stol and Fitzgerald provided a guideline based on the privacy rules and privacy controls.

- Security is the avoidance and reduction of dishonest and fraudulent actions by developers, users, and crowdsourcing Testers, including theft, cheating, and sabotage. To solve the security concerns in crowdsourced testing, for instance, Zhao et al. presented a philosophy based on security audits and security measures.
- Fairness: This pertains to giving crowdsourcing Testers, developers, and users fair and just chances and benefits, including participation, contribution, and remuneration. To solve the fairness difficulties in crowdsourced testing, for instance, Mao et al. suggested a concept based on the fairness criterion and fairness evaluation.
- Responsibility: This pertains to the elucidation and performance of the tasks and commitments made by users, developers, and crowdsourced Testers, including their transparency, accountability, and quality. To solve the accountability difficulties in crowdsourced testing, for instance, Zhao and Zhu provided a guideline based on the responsibility roles and responsibility norms.

6.3.5.1 Future research directions in this theme include:

- Creating crowdsourcing testing platforms and techniques that are more morally and socially conscious, such as social feedback, ethical design, and social learning.
- Comparing and assessing, via the use of social network analysis, ethical analysis, and social impact assessment, the ethical and social effects of crowdsourced testing on various stakeholders.
- Examining the potential and ethical issues with crowdsourced testing in many settings and situations, including the legal, cultural, etc.

6.4 Conclusion

I explored how crowdsourcing can improve software testing with the goal of quality testing without breaking the bank. I engaged software developers and QA pros in an online poll. They shared their wisdom and views on testing software using crowdsourcing.

Our study discovered some benefits of using crowdsourcing in software testing:

- Getting access to a pool of testers: Websites that crowdsource bring together testers from all over the world. They have different skills, experiences, and locations which allows testing on various devices and for different kinds of users. This leads to finding issues that might not be seen by company testing teams (Hütter &amp; Jain, 2012).
- Quick testing rounds: Having the chance to find and use many testers at once makes it possible to test different parts and platforms at the same time. This makes the testing time shorter than traditional company methods (Jalote, 2015).
- Saving Money: Crowdsourcing is usually cheaper than doing tests in-house. Why? You only pay per test and can change the number of testers whenever you need to. This eliminates the need for expensive fixed overhead costs for maintaining a permanent in-house testing team (Guo &amp; Cheng, 2015).
- Better User Experience: With crowdsourcing, testers are diverse and understand real users. They can spot problems and things that frustrate users. This results in software that is more intuitive, user-friendly, and ultimately provides a better user experience (Zhou et al., 2020).

Effective crowdsourcing does come with challenges:

- Data quality: For valid test results, there must be good processes for reviews and strong communication with testers. Tools for data analysis can help uncover any inconsistencies and help reduce bias (Feng et al., 2018).

- Keeping testers involved: Consistent tester engagement is vital for crowdsourcing. Having a game-like setting, clear instructions, and timely feedback can make users' experiences better and their performance to improve (Wu et al., 2018).
- Privacy and security: Sensitive data and intellectual property must be safe while using crowdsourcing platforms. It requires strong security efforts and data privacy policies that are transparent (Alqahtani et al., 2020).

So, our study shows crowdsourcing can boost testing quality for software. It costs less too. Using a large, varied group of testers, speedy tests, and good cost control helps a lot. The product teams need to sort out quality checks, tester interest, and safety. Software teams can try this fresh method for better user experiences. Further studies can help. They could look at AI-based crowdsourcing and game-like tests. This could help make the most of crowdsourcing and sort out Issues.

APPENDIX A

ONLINE SURVEY COVER NOTE

Subject: Doctorate thesis data collection

As you are aware, I am currently researching Using Crowdsourcing in the software Industry with a critical focus on Software Testing, and your input is crucial in helping me gather the necessary information to complete my study.

As part of my data collection process, I have developed a questionnaire that aims to capture your thoughts, experiences, and opinions on the subject matter. The questionnaire is designed to be comprehensive yet concise, and it should take no more than 15 minutes to complete.

APPENDIX B

ONLINE INTERVIEW QUESTIONNAIRE

Following the interview data collection was primarily carried out on the LinkedIn platform. The interview questionnaire was hosted on Google Forms and had a total of 15 questions.

S. No.	Type	Question
1	Text	Name
2	Text	Email Address
3	Text	LinkedIn Profile (Optional)
4	Radio Button	How long have you been involved in software testing? • Mark only one oval. • 1-4 Years • 4-8 years • 8-15 Years • > 15 Years
5	Multiple Choice	If you have utilized crowdsourcing, which domain it was used for? Check all that apply. • Banking • Capital Markets • Insurance • Healthcare • Energy, utilities, resources • Education • Life Sciences • Consumer Goods & Distribution • High-Tech • Travel & Logistics • Retail

		• Public Services • Manufacturing • N/A
6	Multiple Choice	What all types of testing are you currently involved with? Check all that apply. • Functional • Performance • Usability • Security • Unit Testing • SRE • Localization/ Multilingual testing • Other ……….
7	Radio Button	Have you ever used crowdsourcing for software testing? Mark only one oval. • Yes • No
8	Multiple Choice	Have you utilised any of the following platforms? Please add any other platformin the other box. *Check all that apply.* • Testbirds - www.testbirds.com • UserTesting • BugFinders • Rainforest QA • Testlio • uTest • Testbats • Pay4Bugs • Testin • Bugcrowd • Other ……….
9	Multiple Choice	In what types of testing approaches do you think crowdsourcing is mosteffective for software testing?

		Check all that apply. • Functional • Performance • Usability • Security • Unit Testing • SRE • Localization/ Multilingual testing
10	Multiple Choice	What are the main benefits of using crowdsourcing for software testing, in your opinion? *Check all that apply.* • Lower down Cost of Quality / Cost effectiveness • Helps in accelerating the Testing cycle • Easy availability of testing resources • Global pool of resources • Reduces lag related to Talent management cycle • Diversity • Faster scalability • Test coverage • Omni-channel coverage - web, mobile, in-store, and any other channels
11	Multiple Choice	What are the main challenges or drawbacks of using crowdsourcing forsoftware testing? *Check all that apply.* • Quality control • Logistics • Lack of control • Security risks
12	Multiple Choice	How do you think the use of crowdsourcing for software testing will evolve in the future? *Check all that apply.*

		• Increased adoption: With the increasing demand for software products and services, the adoption of crowdsourced testing is expected to grow significantly. More companies are likely to embrace crowdsourced testing as a viable alternative to traditional testing methods, especially for large-scale or complex projects. • Advanced testing tools: Crowdsourced testing platforms are likely to incorporate advanced testing tools and technologies such as AI, machine learning, and automation to improve testing accuracy and efficiency. • Improved quality control: Crowdsourced testing platforms are likely to implement better quality control mechanisms to ensure that testing is conducted accurately and thoroughly. This could include improved tester selection processes, better testing guidelines, and more comprehensive reporting and feedback mechanisms. • Other ……….
13	Radio Button	Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing? *Mark only one oval.* • Yes • No • Maybe • Don't know
14	Large TextArea	Do you have any recommendations for organizations considering using crowdsourcing for software testing? ---
15	Multiple Choice	What criteria will you use to select a crowdsourcing platform for testing purposes? *Check all that apply.*

		• Testing expertise and experience: Look for a platform that has a pool of experienced testers who have expertise in the specific types of testing you require, such as functional testing, usability testing, or security testing. Ensure that the platform provides you with access to testers who have experience with testing similar software products. • Testing coverage and scalability: Consider a platform that can provide testing coverage across different devices, operating systems, and browsers. Ensure that the platform can scale up or down quickly based on your testing requirements. • Quality control mechanisms: Look for a platform that has rigorous quality control mechanisms in place to ensure the accuracy and reliability of test results. This could include vetting testers, providing detailed testing guidelines, and providing comprehensive feedback and reporting. • Other ……….

APPENDIX C

ONLINE INTERVIEW QUESTIONNAIRE RESULTS

Respondent: P01-SM

How long have you been involved in software testing?	> 15 Years
Role	Senior Management
If you have utilized crowdsourcing, which domain it was used for?	Healthcare, Energy, utilities, resources, Education, Retail
What all types of testing are you currently involved with?	Functional, Performance, Usability, Security, SRE, Localization/ Multilingual testing, UX
Have you ever used crowdsourcing for software testing?	Yes
Have you utilised any of the following platforms? Please add any other platform in the other box.	uTest, 99tests, I built the crowd-testing platform for my ex-employer
In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Functional, Performance, Usability, Security, Localization/ Multilingual testing
What are the main benefits of using crowdsourcing for software testing, in your opinion?	Lower down Cost of Quality / Cost effectiveness, Helps in accelerating the Testing cycle, Easy availability of testing resources, Global pool of resources, Reduces lag related to Talent management cycle, Diversity, Faster scalability, Test coverage, Omni-channel coverage - web, mobile, in-store, and any other channels
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Security risks, Your data is at risks expecially if you are looking at sensitive sectors such as financial services

How do you think the use of crowdsourcing for software testing will evolve in the future?	Increased adoption: With the increasing demand for software products and services, the adoption of crowdsourced testing is expected to grow significantly. More companies are likely to embrace crowdsourced testing as a viable alternative to traditional testing methods, especially for large-scale or complex projects., Advanced testing tools: Crowdsourced testing platforms are likely to incorporate advanced testing tools and technologies such as AI, machine learning, and automation to improve testing accuracy and efficiency., Improved quality control: Crowdsourced testing platforms are likely to implement better quality control mechanisms to ensure that testing is conducted accurately and thoroughly. This could include improved tester selection processes, better testing guidelines, and more comprehensive reporting and feedback mechanisms.
Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Yes
Do you have any recommendations for organizations considering using crowdsourcing for software testing?	List the problems you are trying to solve through crowd testing. Identify your use case(s) well. Identify the target market and potential users and choose your testers from your potential user base for more accurate feedback.
What criteria will you use to select a crowdsourcing platform for testing purposes?	Testing expertise and experience: Look for a platform that has a pool of experienced testers who have expertise in the specific types of testing you require, such as functional testing, usability testing, or security testing. Ensure that the platform provides you with access to testers who have experience with testing similar software products., Testing coverage and scalability: Consider a platform that can provide testing coverage across different devices, operating systems, and browsers. Ensure that the platform can scale up or down quickly based on your testing requirements., Quality control mechanisms: Look for a platform that has rigorous quality control mechanisms in place to ensure the accuracy and reliability of test results. This could include vetting testers, providing detailed testing guidelines, and providing comprehensive feedback and reporting.

Respondent: P02-SM

How long have you been involved in software testing?	> 15 Years
Role	Senior Management
If you have utilized crowdsourcing, which domain it was used for?	Insurance, Energy, utilities, resources, High-Tech
What all types of testing are you currently involved with?	Functional, Performance, Usability, Security, Unit Testing, Localization/ Multilingual testing
Have you ever used crowdsourcing for software testing?	No
Have you utilised any of the following platforms? Please add any other platform in the other box.	In house or contractor driven teams
In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Performance, Security, Localization/ Multilingual testing
What are the main benefits of using crowdsourcing for software testing, in your opinion?	Helps in accelerating the Testing cycle, Easy availability of testing resources, Global pool of resources, Test coverage
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Quality control, Lack of control, Security risks
How do you think the use of crowdsourcing for software testing will evolve in the future?	Increased adoption: With the increasing demand for software products and services, the adoption of crowdsourced testing is expected to grow significantly. More companies are likely to embrace crowdsourced testing as a viable alternative to traditional testing methods, especially for large-scale or complex projects., Advanced testing tools: Crowdsourced testing platforms are likely to incorporate advanced testing tools and technologies such as AI, machine learning, and automation to improve testing accuracy and efficiency.
Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Maybe
Do you have any recommendations for organizations considering using crowdsourcing for software testing?	Create a robust platform and governance model, redact the data and test units.

What criteria will you use to select a crowdsourcing platform for testing purposes?	Testing expertise and experience: Look for a platform that has a pool of experienced testers who have expertise in the specific types of testing you require, such as functional testing, usability testing, or security testing. Ensure that the platform provides you with access to testers who have experience with testing similar software products., Testing coverage and scalability: Consider a platform that can provide testing coverage across different devices, operating systems, and browsers. Ensure that the platform can scale up or down quickly based on your testing requirements.

Respondent: P03-SM

How long have you been involved in software testing?	> 15 Years
Role	Senior Management
If you have utilized crowdsourcing, which domain it was used for?	N/A
What all types of testing are you currently involved with?	Functional, Performance, Security, Accessibility
Have you ever used crowdsourcing for software testing?	No
Have you utilised any of the following platforms? Please add any other platform in the other box.	No
In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Functional, Usability, Localization/ Multilingual testing
What are the main benefits of using crowdsourcing for software testing, in your opinion?	Diversity
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Quality control, Lack of control, Security risks
How do you think the use of crowdsourcing for software testing will evolve in the future?	Improved quality control: Crowdsourced testing platforms are likely to implement better quality control mechanisms to ensure that testing is conducted accurately and thoroughly. This could include improved tester selection processes, better testing guidelines, and more comprehensive reporting and feedback mechanisms.

Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	No
Do you have any recommendations for organizations considering using crowdsourcing for software testing?	
What criteria will you use to select a crowdsourcing platform for testing purposes?	Testing expertise and experience: Look for a platform that has a pool of experienced testers who have expertise in the specific types of testing you require, such as functional testing, usability testing, or security testing. Ensure that the platform provides you with access to testers who have experience with testing similar software products., Quality control mechanisms: Look for a platform that has rigorous quality control mechanisms in place to ensure the accuracy and reliability of test results. This could include vetting testers, providing detailed testing guidelines, and providing comprehensive feedback and reporting.

Respondent: P04-SM

How long have you been involved in software testing?	> 15 Years
Role	Senior Management
If you have utilized crowdsourcing, which domain it was used for?	Banking, Insurance, Life Sciences, Manufacturing
What all types of testing are you currently involved with?	Functional, Performance, Security, Unit Testing
Have you ever used crowdsourcing for software testing?	Yes
Have you utilised any of the following platforms? Please add any other platform in the other box.	UserTesting, Rainforest QA
In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Functional, Performance, Unit Testing
What are the main benefits of using crowdsourcing for software testing, in your opinion?	Lower down Cost of Quality / Cost effectiveness, Global pool of resources, Diversity, Test coverage
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Quality control, Security risks

How do you think the use of crowdsourcing for software testing will evolve in the future?	Advanced testing tools: Crowdsourced testing platforms are likely to incorporate advanced testing tools and technologies such as AI, machine learning, and automation to improve testing accuracy and efficiency., Improved quality control: Crowdsourced testing platforms are likely to implement better quality control mechanisms to ensure that testing is conducted accurately and thoroughly. This could include improved tester selection processes, better testing guidelines, and more comprehensive reporting and feedback mechanisms.
Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Don't know
Do you have any recommendations for organizations considering using crowdsourcing for software testing?	
What criteria will you use to select a crowdsourcing platform for testing purposes?	Testing coverage and scalability: Consider a platform that can provide testing coverage across different devices, operating systems, and browsers. Ensure that the platform can scale up or down quickly based on your testing requirements.

Respondent: P20-ST

How long have you been involved in software testing?	4-8 years
Role	Sr. Tester
If you have utilized crowdsourcing, which domain it was used for?	Education
What all types of testing are you currently involved with?	Performance
Have you ever used crowdsourcing for software testing?	No
Have you utilised any of the following platforms? Please add any other platform in the other box.	UserTesting

In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Performance
What are the main benefits of using crowdsourcing for software testing, in your opinion?	Helps in accelerating the Testing cycle
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Lack of control
How do you think the use of crowdsourcing for software testing will evolve in the future?	Improved quality control: Crowdsourced testing platforms are likely to implement better quality control mechanisms to ensure that testing is conducted accurately and thoroughly. This could include improved tester selection processes, better testing guidelines, and more comprehensive reporting and feedback mechanisms.
Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Yes
Do you have any recommendations for organizations considering using crowdsourcing for software testing?	To make proper selection as per their type of operation, always looking into the budgeting and resources needed to put in place.
What criteria will you use to select a crowdsourcing platform for testing purposes?	Quality control mechanisms: Look for a platform that has rigorous quality control mechanisms in place to ensure the accuracy and reliability of test results. This could include vetting testers, providing detailed testing guidelines, and providing comprehensive feedback and reporting.

Respondent: P21-ST

How long have you been involved in software testing?	4-8 years
Role	Sr. Tester
If you have utilized crowdsourcing, which domain it was used for?	N/A
What all types of testing are you currently involved with?	Functional, Performance, Usability, Security, ETL
Have you ever used crowdsourcing for software testing?	No

Have you utilised any of the following platforms? Please add any other platform in the other box.	No
In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Functional, Performance, Usability, Localization/ Multilingual testing
What are the main benefits of using crowdsourcing for software testing, in your opinion?	Diversity, Omni-channel coverage - web, mobile, in-store, and any other channels
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Lack of control
How do you think the use of crowdsourcing for software testing will evolve in the future?	Advanced testing tools: Crowdsourced testing platforms are likely to incorporate advanced testing tools and technologies such as AI, machine learning, and automation to improve testing accuracy and efficiency.
Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Maybe

Respondent: P22-ST

How long have you been involved in software testing?	4-8 years
Role	Sr. Tester
If you have utilized crowdsourcing, which domain it was used for?	N/A
What all types of testing are you currently involved with?	Functional, Usability
Have you ever used crowdsourcing for software testing?	No
Have you utilised any of the following platforms? Please add any other platform in the other box.	None
In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Functional, Performance, Usability, Security, Localization/ Multilingual testing

What are the main benefits of using crowdsourcing for software testing, in your opinion?	Helps in accelerating the Testing cycle, Easy availability of testing resources, Global pool of resources, Diversity, Faster scalability, Omni-channel coverage - web, mobile, in-store, and any other channels
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Lack of control, Security risks
How do you think the use of crowdsourcing for software testing will evolve in the future?	Increased adoption: With the increasing demand for software products and services, the adoption of crowdsourced testing is expected to grow significantly. More companies are likely to embrace crowdsourced testing as a viable alternative to traditional testing methods, especially for large-scale or complex projects.
Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Maybe

Respondent: P23-ST

How long have you been involved in software testing?	4-8 years
Role	Sr. Tester
If you have utilized crowdsourcing, which domain it was used for?	Banking, Insurance, Healthcare, Life Sciences, Travel & Logistics, Retail
What all types of testing are you currently involved with?	Functional, Performance, Usability, SRE
Have you ever used crowdsourcing for software testing?	No
Have you utilised any of the following platforms? Please add any other platform in the other box.	UserTesting, Bugcrowd
In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Performance

What are the main benefits of using crowdsourcing for software testing, in your opinion?	Lower down Cost of Quality / Cost effectiveness, Helps in accelerating the Testing cycle, Reduces lag related to Talent management cycle, Faster scalability
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Lack of control, Security risks
How do you think the use of crowdsourcing for software testing will evolve in the future?	Increased adoption: With the increasing demand for software products and services, the adoption of crowdsourced testing is expected to grow significantly. More companies are likely to embrace crowdsourced testing as a viable alternative to traditional testing methods, especially for large-scale or complex projects., Advanced testing tools: Crowdsourced testing platforms are likely to incorporate advanced testing tools and technologies such as AI, machine learning, and automation to improve testing accuracy and efficiency.
Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Maybe

Respondent: P24-ST

How long have you been involved in software testing?	4-8 years
Role	Sr. Tester
If you have utilized crowdsourcing, which domain it was used for?	Banking, Insurance, Healthcare, Education
What all types of testing are you currently involved with?	Functional, Usability, Unit Testing, Localization/ Multilingual testing
Have you ever used crowdsourcing for software testing?	Yes
Have you utilised any of the following platforms? Please add any other platform in the other box.	uTest, Testers work

In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Functional, Usability, Localization/ Multilingual testing
What are the main benefits of using crowdsourcing for software testing, in your opinion?	Easy availability of testing resources, Reduces lag related to Talent management cycle, Test coverage, Omni-channel coverage - web, mobile, in-store, and any other channels
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Quality control
How do you think the use of crowdsourcing for software testing will evolve in the future?	Increased adoption: With the increasing demand for software products and services, the adoption of crowdsourced testing is expected to grow significantly. More companies are likely to embrace crowdsourced testing as a viable alternative to traditional testing methods, especially for large-scale or complex projects., Advanced testing tools: Crowdsourced testing platforms are likely to incorporate advanced testing tools and technologies such as AI, machine learning, and automation to improve testing accuracy and efficiency., Improved quality control: Crowdsourced testing platforms are likely to implement better quality control mechanisms to ensure that testing is conducted accurately and thoroughly. This could include improved tester selection processes, better testing guidelines, and more comprehensive reporting and feedback mechanisms.
Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Yes

Respondent: P25-ST

How long have you been involved in software testing?	4-8 years
Role	Sr. Tester
If you have utilized crowdsourcing, which domain it was used for?	Banking, Insurance, Healthcare, Energy, utilities, resources, Education, Life Sciences, Consumer

	Goods & Distribution, High-Tech, Travel & Logistics, Retail, Public Services, Manufacturing
What all types of testing are you currently involved with?	Functional, Usability, Localization/ Multilingual testing, Accessibility
Have you ever used crowdsourcing for software testing?	Yes
Have you utilised any of the following platforms? Please add any other platform in the other box.	Testbirds - www.testbirds.com
In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Functional, Usability, Localization/ Multilingual testing
What are the main benefits of using crowdsourcing for software testing, in your opinion?	Helps in accelerating the Testing cycle, Global pool of resources, Faster scalability, Test coverage, Helps you work more Agile as a business & ensures the ability to test in production not just in Sandbox i.e. real testers, real devices, real payment methods etc.
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Security risks, Depending on company security protocal it can be a challenge to give access to outside testers but mostly this can be overcome e.g. with VPn's etc.
How do you think the use of crowdsourcing for software testing will evolve in the future?	Increased adoption: With the increasing demand for software products and services, the adoption of crowdsourced testing is expected to grow significantly. More companies are likely to embrace crowdsourced testing as a viable alternative to traditional testing methods, especially for large-scale or complex projects.
Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Yes

Respondent: P26-ST

How long have you been involved in software testing?	4-8 years
Role	Sr. Tester
If you have utilized crowdsourcing, which domain it was used for?	High-Tech

What all types of testing are you currently involved with?	Functional, Performance, Usability, Localization/ Multilingual testing
Have you ever used crowdsourcing for software testing?	Yes
Have you utilised any of the following platforms? Please add any other platform in the other box.	Selenium
In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Performance, Usability, Localization/ Multilingual testing
What are the main benefits of using crowdsourcing for software testing, in your opinion?	Lower down Cost of Quality / Cost effectiveness, Helps in accelerating the Testing cycle, Easy availability of testing resources, Diversity, Faster scalability, Omni-channel coverage - web, mobile, in-store, and any other channels
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Logistics
How do you think the use of crowdsourcing for software testing will evolve in the future?	Increased adoption: With the increasing demand for software products and services, the adoption of crowdsourced testing is expected to grow significantly. More companies are likely to embrace crowdsourced testing as a viable alternative to traditional testing methods, especially for large-scale or complex projects., Advanced testing tools: Crowdsourced testing platforms are likely to incorporate advanced testing tools and technologies such as AI, machine learning, and automation to improve testing accuracy and efficiency.
Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Yes

Respondent: P27-MM

How long have you been involved in software testing?	8-15 Years
Role	Middle Management

If you have utilized crowdsourcing, which domain it was used for?	Insurance, Retail
What all types of testing are you currently involved with?	Functional, Performance, Usability
Have you ever used crowdsourcing for software testing?	No
Have you utilised any of the following platforms? Please add any other platform in the other box.	UserTesting, Testlio
In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Functional, Performance, Usability
What are the main benefits of using crowdsourcing for software testing, in your opinion?	Faster scalability, Test coverage, Omni-channel coverage - web, mobile, in-store, and any other channels
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Quality control, Logistics
How do you think the use of crowdsourcing for software testing will evolve in the future?	Increased adoption: With the increasing demand for software products and services, the adoption of crowdsourced testing is expected to grow significantly. More companies are likely to embrace crowdsourced testing as a viable alternative to traditional testing methods, especially for large-scale or complex projects.
Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Maybe

Respondent: P28-MM

How long have you been involved in software testing?	8-15 Years
Role	Middle Management

If you have utilized crowdsourcing, which domain it was used for?	Consumer Goods & Distribution
What all types of testing are you currently involved with?	Functional, SRE
Have you ever used crowdsourcing for software testing?	No
Have you utilised any of the following platforms? Please add any other platform in the other box.	Testbirds - www.testbirds.com, CrowdSprint, Passbrains, TesterWork, QAProvider,
In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Performance, Usability, Security, SRE, Localization/ Multilingual testing
What are the main benefits of using crowdsourcing for software testing, in your opinion?	Easy availability of testing resources, Global pool of resources, Reduces lag related to Talent management cycle, Diversity, Faster scalability, Test coverage, Omni-channel coverage - web, mobile, in-store, and any other channels
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Logistics, Lack of control, Security risks
How do you think the use of crowdsourcing for software testing will evolve in the future?	Increased adoption: With the increasing demand for software products and services, the adoption of crowdsourced testing is expected to grow significantly. More companies are likely to embrace crowdsourced testing as a viable alternative to traditional testing methods, especially for large-scale or complex projects., Advanced testing tools: Crowdsourced testing platforms are likely to incorporate advanced testing tools and technologies such as AI, machine learning, and automation to improve testing accuracy and efficiency., Improved quality control: Crowdsourced testing platforms are likely to implement better quality control mechanisms to ensure that testing is conducted accurately and thoroughly. This could include improved tester selection processes, better testing guidelines, and more comprehensive reporting and feedback mechanisms.

Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Maybe

Respondent: P29-MM

How long have you been involved in software testing?	8-15 Years
Role	Middle Management
If you have utilized crowdsourcing, which domain it was used for?	N/A
What all types of testing are you currently involved with?	Functional, Usability, Localization/ Multilingual testing,
Have you ever used crowdsourcing for software testing?	No
Have you utilised any of the following platforms? Please add any other platform in the other box.	Testlio
In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Functional, Performance, Usability
What are the main benefits of using crowdsourcing for software testing, in your opinion?	Helps in accelerating the Testing cycle, Easy availability of testing resources, Global pool of resources, Diversity
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Quality control, Lack of control, Security risks
How do you think the use of crowdsourcing for software testing will evolve in the future?	Increased adoption: With the increasing demand for software products and services, the adoption of crowdsourced testing is expected to grow significantly. More companies are likely to embrace crowdsourced testing as a viable alternative to traditional testing methods, especially for large-scale or complex projects.

Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Maybe

Respondent: P30-MM

How long have you been involved in software testing?	8-15 Years
Role	Middle Management
If you have utilized crowdsourcing, which domain it was used for?	N/A
What all types of testing are you currently involved with?	Functional
Have you ever used crowdsourcing for software testing?	No
Have you utilised any of the following platforms? Please add any other platform in the other box.	No
In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Performance
What are the main benefits of using crowdsourcing for software testing, in your opinion?	Helps in accelerating the Testing cycle
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Lack of control
How do you think the use of crowdsourcing for software testing will evolve in the future?	Advanced testing tools: Crowdsourced testing platforms are likely to incorporate advanced testing tools and technologies such as AI, machine learning, and automation to improve testing accuracy and efficiency.
Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Maybe

Respondent: P31-MM

How long have you been involved in software testing?	8-15 Years
Role	Middle Management
If you have utilized crowdsourcing, which domain it was used for?	Insurance, Healthcare, Education, Manufacturing
What all types of testing are you currently involved with?	Functional, Usability, Security
Have you ever used crowdsourcing for software testing?	Yes
Have you utilised any of the following platforms? Please add any other platform in the other box.	BugFinders, uTest, Bugcrowd, Hackerone / Intigriti
In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Functional, Usability, Security
What are the main benefits of using crowdsourcing for software testing, in your opinion?	Helps in accelerating the Testing cycle, Easy availability of testing resources, Reduces lag related to Talent management cycle, Test coverage, All the full time employers always interested in 2nd income, hence it'll be benefit for both Employer and the Tester.
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Security risks
How do you think the use of crowdsourcing for software testing will evolve in the future?	Advanced testing tools: Crowdsourced testing platforms are likely to incorporate advanced testing tools and technologies such as AI, machine learning, and automation to improve testing accuracy and efficiency., Improved quality control: Crowdsourced testing platforms are likely to implement better quality control mechanisms to ensure that testing is conducted accurately and thoroughly. This could include improved tester selection processes, better testing guidelines, and more comprehensive reporting and feedback mechanisms.

Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Yes

Respondent: P32-MM

How long have you been involved in software testing?	8-15 Years
Role	Middle Management
If you have utilized crowdsourcing, which domain it was used for?	Healthcare, Travel & Logistics, Retail
What all types of testing are you currently involved with?	Functional, Performance, Usability, Localization/ Multilingual testing, API Testing
Have you ever used crowdsourcing for software testing?	Yes
Have you utilised any of the following platforms? Please add any other platform in the other box.	UserTesting, uTest
In what types of testing approaches do you think crowdsourcing is most effective for software testing?	Functional, Performance, Usability, Localization/ Multilingual testing
What are the main benefits of using crowdsourcing for software testing, in your opinion?	Lower down Cost of Quality / Cost effectiveness, Test coverage, Omni-channel coverage - web, mobile, in-store, and any other channels
What are the main challenges or drawbacks of using crowdsourcing for software testing?	Security risks
How do you think the use of crowdsourcing for software testing will evolve in the future?	Increased adoption: With the increasing demand for software products and services, the adoption of crowdsourced testing is expected to grow significantly. More companies are likely to embrace crowdsourced testing as a viable alternative to traditional testing methods, especially for large-scale or complex projects., Improved quality control: Crowdsourced testing platforms are likely to implement better quality control mechanisms to ensure that testing is conducted accurately and

	thoroughly. This could include improved tester selection processes, better testing guidelines, and more comprehensive reporting and feedback mechanisms.
Will you recommend crowdsourcing for organizations considering using crowdsourcing for software testing?	Yes

REFERENCES

Sathe, A. and Kulkarni, R., 2013. Study of testing as a service (TaaS) – cost effective framework for TaaS in cloud environment. International Journal of Application or Innovation in Engineering & Management (IJAIEM), 2(5), ISSN 2319 – 4847.

Acar, O.A., 2019. Motivations and solution appropriateness in crowdsourcing challenges for innovation. Research Policy, 48(8), p.103716.

Afuah, A. and Tucci, C.L., 2012. Crowdsourcing as a solution to distant search. *Academy of management review*, *37*(3), pp.355-375.

Ajzen, I. and Fishbein, M., 1975. A Bayesian analysis of attribution processes. *Psychological bulletin*, *82*(2), p.261.

Alsayyari, M. and Alyahya, S., 2018, March. Supporting coordination in crowdsourced software testing services. In *2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)* (pp. 69-75). IEEE.

Alyahya, S., 2020. Crowdsourced software testing: A systematic literature review. *Information and Software Technology*, *127*, p.106363.

Archak, N., 2010, April. Money, glory and cheap talk: analysing strategic behavior of contestants in simultaneous crowdsourcing contests on TopCoder. com. In *Proceedings of the 19th international conference on World wide web* (pp. 21-30).

Bandura, A., 1977. Self-efficacy: toward a unifying theory of behavioral change. *Psychological review*, *84*(2), p.191.

Belleflamme, P., Lambert, T., & Schwienbacher, A. (2010, June). Crowdfunding: An industrial organisation perspective. In Prepared for the workshop Digital Business Models: Understanding Strategies, held in Paris on June (pp. 25-26).

Bommert, B., 2010. Collaborative innovation in the public sector. *International public management review*, *11*(1), pp.15-33.

Bonabeau, E., 2009. Decisions 2.0: The power of collective intelligence. *MIT Sloan management review*, *50*(2), p.45.

Boudreau, K.J., Lakhani, K.R. and Menietti, M., 2016. Performance responses to competition across skill levels in rank-order tournaments: field evidence and implications for tournament design. *The RAND Journal of Economics*, *47*(1), pp.140-165.

Burger-Helmchen, T. and Pénin, J., 2010, March. Crowdsourcing of inventive activities: Definition and limits. *International Journal of Innovation and Sustainable Development* 5(2/3):246-263. 2011.

Bankole Awuzie, Peter McDermott, (2017) "An abductive approach to qualitative built environment research: A viable system methodological exposé", Qualitative Research Journal, Vol. 17 Issue: 4, pp.356-372, https://doi.org/10.1108/QRJ-08-2016-0048

Casal, D.P., 2011. Crowdsourcing the corpus: Using collective intelligence as a method for composition. *Leonardo Music Journal*, *21*, pp.25-28.

Cox, L.P., 2011. Truth in crowdsourcing. *IEEE Security & Privacy*, *9*(5), pp.74-76.

Creswe l, John W, 2014. Research design: qualitative, quantitative, and mixed methods approaches / John W. Creswe l. — 4th ed, pp.43-44

Dorst, K. (2011), "The core of 'design thinking'and its application". Design Studies, Vol. 32 No. 6, pp. 521-532.

Estellés-Arolas, E. and González-Ladrón-de-Guevara, F., 2012. Towards an integrated crowdsourcing definition. *Journal of Information science*, *38*(2), pp.189-200.

Farrell, J. and Rabin, M., 1996. Cheap talk. *Journal of Economic perspectives*, *10*(3), pp.103-118.

Fitt, V.A., 2011. Crowdsourcing the News: News Organization Liability for iReporters. *William Mitchell Law Review*, *37*(4), p.1.

Hoßfeld, T., Liers, F., Schatz, R., Staehle, B., Staehle, D., Volkert, T. and Wamser, F., 2012. Quality of experience management for youTube: clouds, foG and the aquareYoum. *PIK-Praxis der Informationsverarbeitung und Kommunikation*, *35*(3), pp.133-143.

Howe, J., 2006. The rise of crowdsourcing. *Wired magazine*, *14*(6), pp.1-4.

Hu, Z. and Wu, W., 2014, April. A game theoretic model of software crowdsourcing. In *2014 IEEE 8th International Symposium on Service Oriented System Engineering* (pp. 446-453). IEEE.

Jeppesen, L.B. and Lakhani, K.R., 2010. Marginality and problem-solving effectiveness in broadcast search. *Organisation science*, *21*(5), pp.1016-1033.

Kazman, R. and Chen, H.M., 2009. The metropolis model a new logic for development of crowdsourced systems. *Communications of the ACM*, *52*(7), pp.76-84.

Kazman, R. and Chen, H.M., 2010, November. The metropolis model and its implications for the engineering of software ecosystems. In *Proceedings of the FSE/SDP workshop on Future of software engineering research* (pp. 187-190).

Lebraty, J.F. and Lobre-Lebraty, K., 2013. *Crowdsourcing: One step beyond*. John Wiley & Sons.

Leimeister, J.M., Zogaj, S. and Durward, D., 2015. New Forms of Employment and IT--Crowdsourcing. In *4th conference of the regulating for decent work network*.

Li, K., Xiao, J., Wang, Y. and Wang, Q., 2013, July. Analysis of the key factors for software quality in crowdsourcing development: An empirical study on topcoder. com. In *2013 IEEE 37th Annual Computer Software and Applications Conference* (pp. 812-817). IEEE.

Liang, H., Wang, M.M., Wang, J.J. and Xue, Y., 2018. How intrinsic motivation and extrinsic incentives affect task effort in crowdsourcing contests: A mediated moderation model. *Computers in Human behavior*, *81*, pp.168-176.

Locke, E.A. and Latham, G.P., 1990. *A theory of goal setting & task performance*. Prentice-Hall, Inc.

Lydon, M. (2012). Topcoder overview. Retrieved from http://www.nasa.gov/pdf/651447main.TopCder_Mike_D1_830am.pdf

Mao, K., Capra, L., Harman, M. and Jia, Y., 2017. A survey of the use of crowdsourcing in software engineering. *Journal of Systems and Software*, *126*, pp.57-84.

Mao, K., Yang, Y., Wang, Q., Jia, Y. and Harman, M., 2015, March. Developer recommendation for crowdsourced software development tasks. In *2015 IEEE Symposium on Service-Oriented System Engineering* (pp. 347-356). IEEE.

Mariani, M.M. and Wamba, S.F., 2020. Exploring how consumer goods companies innovate in the digital age: The role of big data analytics companies. *Journal of Business Research*, *121*, pp.338-352.

Mayer, M., 2019. Examining community dynamics of civic crowdfunding participation. *Computer Supported Cooperative Work (CSCW)*, *28*(5), pp.961-975.

Tejaswi, M. (2021, September 27). Attrition in IT set to cross 1 Million this year. Retrieved from https://www.thehindu.com/business/Industry/attrition-in-it-sector-to-cross-1-million-this-year/article36702884.ece#comments_36702884 . Bengaluru

Nag, S., Heffan, I., Saenz-Otero, A. and Lydon, M., 2012, March. SPHERES Zero Robotics software development: Lessons on crowdsourcing and collaborative competition. In *2012 IEEE Aerospace Conference* (pp. 1-17). IEEE.

Pénin, J. and Burger-Helmchen, T., 2012. Crowdsourcing d'activités inventives et frontières des organisations. *Management international/International Management/Gestiòn Internacional*, *16*, pp.101-112.

Pierre Levy, 2010. Collective Intelligence: mankind's emerging world in cyberspace, Retrieved from https://michaelnielsen.org/blog/collective-intelligence-by-pierre-levy/

Ponsonby, AL and Mattingly, K., 2015. Evaluating new ways of working collectively in science with a focus on crowdsourcing. *EBioMedicine*, *2*(7), p.627.

Rayna, T. and Striukova, L., 2010. Web 2.0 is cheap: supply exceeds demand. *Prometheus*, *28*(3), pp.267-285.

Roozenburg, N. F. (1993), "On the pattern of reasoning in innovative design". Design Studies, Vol. 14 No. 1, pp. 4-18.

Ryan, R.M. and Deci, EL, 2000. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, *25*(1), pp.54-67.

Saez-Rodriguez, J., Costello, J.C., Friend, S.H., Kellen, M.R., Mangravite, L., Meyer, P., Norman, T. and Stolovitzky, G., 2016. Crowdsourcing biomedical research: leveraging communities as innovation engines. *Nature Reviews Genetics*, *17*(8), pp.470-486.

Saxton, G.D., Oh, O. and Kishore, R., 2013. Rules of crowdsourcing: Models, issues, and systems of control. *Information Systems Management*, *30*(1), pp.2-20.

Schneider, C. and Cheung, T., 2013. The power of the crowd: Performing usability testing using an on-demand workforce. In *Information systems development* (pp. 551-560). Springer, New York, NY.

Srite, M. and Karahanna, E., 2006. The role of espoused national cultural values in technology acceptance. *MIS quarterly*, pp.679-704.

Stol, KJ and Fitzgerald, B., 2014, May. Two's company, three's a crowd: a case study of crowdsourcing software development. In *Proceedings of the 36th International Conference on Software Engineering* (pp. 187-198).

Tajedin, H. and Nevo, D., 2014, January. Value-adding intermediaries in software crowdsourcing. In *2014 47th Hawaii International Conference on System Sciences* (pp. 1396-1405). IEEE.

Tsai, P.Y., Ko, C.J., Chia, S.Y., Lu, Y.J. and Tuanmu, M.N., 2021. New insights into the patterns and drivers of avian altitudinal migration from a growing crowdsourcing data source. *Ecography*, *44*(1), pp.75-86.

Tung, Y.H. and Tseng, S.S., 2013. A novel approach to collaborative testing in a crowdsourcing environment. *Journal of Systems and Software*, *86*(8), pp.2143-2153.

Van der Heijden, H., 2004. User acceptance of hedonic information systems. *MIS quarterly*, pp.695-704.

Applause App Quality, Inc. (2021). What is CrowdTesting? Retrieved from https://www.applause.com/crowdtesting

Wu, W., Tsai, W.T. and Li, W., 2013. An evaluation framework for software crowdsourcing. *Frontiers of Computer Science*, *7*(5), pp.694-709.

Wu, W., Tsai, W.T. and Li, W., 2013. Creative software crowdsourcing: from components and algorithm development to project concept formations. *International Journal of Creative Computing*, *1*(1), pp.57-91.

Xu, X.L. and Wang, Y., 2014. Crowdsourcing software development process study on ultra-large-scale system. In *Advanced Materials Research* (Vol. 989, pp. 4441-4446). Trans Tech Publications Ltd.

Yan, R., Gao, M., Pavlick, E. and Callison-Burch, C., 2014, June. Are two heads better than one? crowdsourced translation via a two-step collaboration of non-professional translators and editors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1134-1144).

Yilmaz, M., 2015. *Framework for evaluating loop invariant detection games in relation to automated dynamic invariant detectors*. Naval Postgraduate School Monterey United States.

Zhao, Y. and Zhu, Q., 2014. Evaluation on crowdsourcing research: Current status and future direction. *Information Systems Frontiers, 16*(3), pp.417-434.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. Qualitative Research in Psychology, 3(2), 77-101.

Corbin, J., & Strauss, A. (2014). Basics of qualitative research: Techniques and procedures for developing grounded theory (4th ed.). Sage Publications.

Creswell, J. W. (2013). Qualitative inquiry and research design: Choosing among five approaches. Sage Publications.

Miles, M. B., Huberman, A. M., & Saldaña, J. (2013). Qualitative data analysis: A methods sourcebook (3rd ed.). Sage Publications.

Silverman, D. (2016). Qualitative research (4th ed.). Sage Publications.

Alyahya, S. (2022). Collaborative crowdsourced software testing. *Electronics, 11*(20), 3340. https://doi.org/10.3390/electronics11203340

Alyahya, S., & Alsayyari, M. (2020). Towards better crowdsourced software testing process. *International Journal of Cooperative Information Systems, 29*(01n02), 2040009. https://doi.org/10.1142/s0218843020400092

Asiegbu Baldwin, Oluigbo Ikenna, Ajakwe Simeon, Onyike Gerald. (2017). *Crowdsourcing Software Development: Concept, Benefits and Adoption*. Isroset.org. https://www.isroset.org/pub_paper/IJSRCSE/2-IJSRCSE-0315.pdf

Bias, R. G., & Mayhew, D. J. (2005). *Cost-Justifying Usability: An update for the internet age, second edition*. Elsevier.

Hosseini, M., Shahri, A., Phalp, K., Taylor, J., & Ali, R. (2015). Crowdsourcing: A taxonomy and systematic mapping study. *Computer Science Review*, *17*, 43–69. https://doi.org/10.1016/j.cosrev.2015.05.001

Imtiaz, J., Sherin, S., Khan, M. U., & Iqbal, M. Z. (2019). A systematic literature review of test breakage prevention and repair techniques. *Information and Software Technology*, *113*, 1–19. https://doi.org/10.1016/j.infsof.2019.05.001

Leicht, N., & Leimeister, J. M. (2016). *How to systematically conduct crowdsourced software testing? Insights from an action research project*. Unisg.Ch. https://www.alexandria.unisg.ch/server/api/core/bitstreams/d49eeb44-fc7d-4015-9b3f-1ece10548632/content

Liu, D., Bias, R. G., Lease, M., & Kuipers, R. (2012). Crowdsourcing for usability testing: Crowdsourcing for Usability Testing. *Proceedings of the American Society for Information Science and Technology*, *49*(1), 1–10. https://doi.org/10.1002/meet.14504901100

Sarı, A., Tosun, A., & Alptekin, G. I. (2019). A systematic literature review on crowdsourcing in software engineering. *The Journal of Systems and Software*, *153*, 200–219. https://doi.org/10.1016/j.jss.2019.04.027

Spool, J., & Schroeder, W. (2001). Testing web sites: Five users is nowhere near enough. *CHI '01 Extended Abstracts on Human Factors in Computing Systems*.

Stol, K.-J., & Fitzgerald, B. (2014). *Two's company, three's a crowd: A case study of crowdsourcing software development*. Core.ac.uk. https://core.ac.uk/download/pdf/59353793.pdf

Wang, J., Wang, L., Wang, Y., Zhang, D., & Kong, L. (2018). Task allocation in mobile crowd sensing: State-of-the-art and future opportunities. *IEEE Internet of Things Journal*, *5*(5), 3747–3757. https://doi.org/10.1109/jiot.2018.2864341

Alqahtani, A., Jarada, A., & Zaidan, A. A. (2020). Security and privacy aspects of crowdsourcing software testing: A systematic review. Journal of Systems and Software, 169, 110616.

Feng, X., Liu, Y., Guo, Y., & Zhou, Y. (2018). Bug report classification for mobile crowdsourcing software testing projects. In 2018 IEEE 41st International Conference on Software Engineering (ICSE) (pp. 844-855). IEEE.

Guo, Y., & Cheng, R. Y. (2015). Incentive mechanisms for crowdsourcing software testing: An investigation. Software Quality Journal, 23(3), 499-534.

Hütter, E., & Jain, R. K. (2012). Crowdsourcing software testing: A systematic literature review. Information and Software Technology, 54(5), 437-447.

Jalote, P. A. (2015). An integrated approach to software engineering. Springer.

Zhou, Y., Wu, Z., Yang, L., Guo, Y., & Wang, F. (2020). Test case generation for web applications using user behavior analysis. IEEE Transactions on Reliability, 69(3), 817-834.

Alqahtani, A., Jarada, A., & Zaidan, A. A. (2020). Security and privacy aspects of crowdsourcing software testing: A systematic review. Journal of Systems and Software, 169, 110616.

Feng, X., Liu, Y., Guo, Y., & Zhou, Y. (2018). Bug report classification for mobile crowdsourcing software testing projects. In 2018 IEEE 41st International Conference on Software Engineering (ICSE) (pp. 844-855). IEEE.

Guo, Y., & Cheng, R. Y. (2015). Incentive mechanisms for crowdsourcing software testing: An investigation. Software Quality Journal, 23(3), 499-534.

Hütter, E., & Jain, R. K. (2012). Crowdsourcing software testing: A systematic literature review. Information and Software Technology, 54(5), 437-447.

Jalote, P. A. (2015). An integrated approach to software engineering. Springer.

Jain, N., & George, M. P. (2015). The effect of communication and reward structure on test quality in a service crowdsourcing environment. ACM Transactions on Software Engineering and Methodology, 24(3), 1-32.

Wu, X., Liu, Y., Guo, Y., & He, J. (2018). Gamification for crowdsourced software testing: A comparative study with traditional incentives. Information Systems and Micro-computing, 36(2), 103-134.

Zhou, Y., Wu, Z., Yang, L., Guo, Y., & Wang, F. (2020). Test case generation for web applications using user behavior analysis. IEEE Transactions on Reliability