A MECHANISM FOR TASK DECOMPOSITION OF MICROTASKING IN CROWDSOURCED SOFTWARE DEVELOPMENT

By

MAIRA ZULFIQAR

NATIONAL UNIVERSITY OF MODERN LANGUAGES

ISLAMABAD

January, 2022

A Mechanism for Task Decomposition of Microtasking in Crowdsourced Software Development

By

MAIRA ZULFIQAR

BSSE, National University of Modern Languages, Islamabad, 2017

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

In **Software Engineering**

To
FACULTY OF ENGINEERING & COMPUTER SCIENCES

NATIONAL UNIVERSITY OF MODERN LANGUAGES ISLAMABAD

THESIS AND DEFENSE APPROVAL FORM

The undersigned certify that they have read the following thesis, examined the defense, are satisfied with overall exam performance, and recommend the thesis to the Faculty of Engineering and Computer Sciences for acceptance.

THESIS TITLE: A Mechanism for Task Decomposition of Microtasking in Crowdsourced Software Development

Submitted by: Maira Zulfiqar **Registration #:** 001 MSSE/Ibd/F18

Master of Science in Software Engineering

Degree name in full

Software Engineering

Name of Discipline

Dr. Huma Hayat _____

Name of Research Supervisor Signature of Research Supervisor

Dr. Basit Shahzad _____

Name of Dean (FE&CS) Signature of Dean (FE&CS)

Prof. Dr. Muhammad Safeer Awan _____

Name of Pro-rector Academics Signature of Pro-rector Academics

January 21st, 2022

Date

AUTHOR'S DECLARATION

I Maira Zulfiqar

Daughter of Ch. Zulfiqar Ali

Registration # 001 MSSE/Ibd/F18

Discipline Software Engineering

Candidate of **Master of Science in Software Engineering (MSSE)** at the National University of Modern Languages do hereby declare that the thesis **A Mechanism for Task Decomposition of Microtasking in Crowdsourced Software Development** submitted by me in partial fulfillment of MSSE degree, is my original work, and has not been submitted or published earlier. I also solemnly declare that it shall not, in future, be submitted by me for obtaining any other degree from this or any other university or institution. I also understand that if evidence of plagiarism is found in my thesis/dissertation at any stage, even after the award of a degree, the work may be cancelled and the degree revoked.

Signature of Candidate

Maira Zulfiqar

Name of Candidate

21st January, 2022

Date

ABSTRACT

A Mechanism for Task Decomposition of Microtasking in Crowdsourced Software Development

Software crowdsourcing has reshaped the form of software development and gained more importance in recent years. Microtasking is one of the models of crowdsourcing which involves the decomposition of complex task into the number of simple, short and self-contained tasks (microtasks). It has been observed that it is difficult for microtasking platforms to define the criteria for decomposition of task into microtasks. Less research has focused to microtasking in software crowdsourcing for the development of task decomposition mechanism. This research fills this gap by focusing on enlisting the task decomposition methods and microtasking activities which exist in crowdsourced software development. Systematic Literature Review has been conducted to identify the task decomposition methods and microtasking activities. A total of 70 research articles are reviewed in systematic review, 72 microtasking activities and 14 task decomposition methods are found. Expert review has been conducted to validate the naming conventions and positioning of identified microtasking activities in their specific microtasking categories. Identified task decomposition methods are compared to check if these are suitable to decompose dependent, independent and hybrid tasks into microtasks. Literature has revealed that no existing task decomposition method can decompose all types of tasks i.e., dependent, independent and hybrid tasks. This study fills this gap by proposing the Hybrid model which can decompose dependent, independent and hybrid tasks into microtasks. Proposed task decomposition model is iterative in nature which can be useful for crowdsourcing and microtasking platforms

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AMT	-	Amazon Mechanical Turk
BDD	-	Behavior Driven Development
CSD	-	Crowdsourced Software Development
OPQ	-	Optimal Priority Queue
SEBOK	-	Software Engineering Body Of Knowledge
SLADE	-	Smart LArge scale task DEcomposer
SLR	-	Systematic Literature Review
UI	-	User Interface

LIST OF APPENDICES

ACKNOWLEDGEMENTS

First of all, I wish to express my gratitude and deep appreciation to Almighty Allah, who made this study possible and successful. This study would not be accomplished unless the honest espousal that was extended from several sources for which I would like to express my sincere thankfulness and gratitude. Yet, there were significant contributors for my attained success and I cannot forget their input, especially my research supervisors, Dr. Noman Malik and Dr. Huma Hayat, who did not leave any stone unturned to guide me during my research journey. Their thought-provoking ideas, scholarly supervision and attention towards research have contributed a lot in successful completion and enrichment of this thesis in time and best manner.

I am extremely thankful to the respected coordinator Dr. Raheel for his support, encouragement and contribution. He has always a constant source of inspiration and his ever-pushing motivational factor was driving factor for thesis completion.

I would like to pay special tribute to my Parents, it’s because of them whatever and wherever I am, it’s just because of them and their selfless sacrificial life and their great efforts with hard work, pain and unceasing prayers have enabled me to reach at this point in life.

DEDICATION

This thesis work is dedicated to my parents and my brother who have been a constant source of support and motivation during the challenges, who have always loved and believed in me. This work is also dedicated to my teachers throughout my education career, who have always encouraged me and whose good examples have taught me to work hard for the things that I aspire to achieve.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Software crowdsourcing has reshaped the form of software development and gained more importance in recent years [1]. It is a process in which different stakeholders (requirement engineer, designer, developer, tester etc.) work parallel on various tasks, for the development of software project [2]. Crowdsourcing term was first used by journalist Jeff Howe in 2006 and he defined this term as "the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call. This can take the form of peer-production (when the job is performed collaboratively), but is also often undertaken by sole individuals" [3].

Crowdsourcing has been used for wide range of applications i.e. online social health network (Sickweather), image tagging applications of Google, online Filipino and English dictionary Bansa.org, online dictionary Wordnik, natural language manipulation application of OpenMind.org and Google Earth [4]. Different platforms e.g. Kaggle, Topcoder, Amazon Mechanical Turk (AMT), freelancer etc are utilizing crowdsourcing concept [3], [5].

Crowdsourcing platforms play an important role to distribute the tasks among stakeholders according to their expertise over the internet. The experts in any perspective, who involved in the development of a system or software project, are generally known as crowd [2]. Crowdsourced software development can be achieved by accomplishing any of its four models that is: (i) peer production, (ii) competitions, (iii) investments and (iv) microtasking [5].

The *peer production* model supports the collective effort of large network of knowledgeable workers with no limit of amount of discussion or changes made. Different

sources can contribute to make the project successful, without focusing on any financial reward. Open source projects (e.g. LINUX), online collaborative softwares (e.g. Stack Overflow) are classic examples of peer production model of crowdsourcing [6] [7].

Another way to outsource a project can be accomplished by use of *competition* model of crowdsourcing. In essence, crowdsourcing platform advertises different tasks for their completion by the effort of best available crowd. According to their expertise, crowd registers their submission in order to compete with others. In this manner, completion of project can be achieved by giving the large amount of money to the contestant who provides the best solution [8].

The *investments* model associates the concept initiator working in collaboration with external parties or sponsors to accomplish their task. The initiator might have an idea that sponsors are interested in funding. At the successful achievement of the project, the collaborator or sponsor might get the product or solution for free. This type of a model is common in crowdfunding initiatives [9].

The *microtasking* model involves the decomposition of complex tasks into "set of self-contained microtasks that could be completed by a diverse group of individuals for a small incentive". Micro-workers participate in this type of crowdsourcing in terms of minutes and hours, hence paid by the crowdsourced platform. In other words, people referred this approach as paid crowdsourcing [10].

Microtasking is a type of crowdsourcing in which tasks are decomposed into short and autonomous units of task [11]. It anticipates crowdsourced software development in which large crowd of temporal workers participates into short and self-contained tasks, reducing barriers to on-boarding, human mobility, compelling organizations and geographical participation and remoteness [11], [12].

Distribution of digital micro tasks (e.g. information elicitation, information categorization, developing test case) lies in the category of microtasking, one of the model of crowdsourcing [6]. Literature has revealed that microtasking in crowdsourcing has been

spreading enormously in different developing economies, in order to minimize an unemployment [6], [13].

Microtasking process is achieved through specialized platform which is designed for generating, assigning, displaying competition and integrating fragments of task [14]. For example, in CodeOn [15] a client developer call for their IDE, and crowd workers utilize this request and related data to acknowledge in the code. In Apparition, client developer delineates the statement for user interface in natural language, and crowd workers narrate those descriptions into user interface behavior, elements and visual styles [16]. In microtasking environment, in order to rate the success of the process, choice of work flow has much importance. It elaborates the time limit of the task, context of offered task, and the methods to integrate the tasks form each contribution to develop the final software product [17].

Accusatively, microtasking encourages the completion of substantial digital tasks by splitting the complicated tasks into simple and minor tasks that can be achieved by diversified micro-workers available on crowdsourced platform [18]. Different software crowdsourcing platforms e.g. Topcoder, Amazon Mechanical Turk, Freelancer and Kaggle are using the microtasking technique, and billions of micro-workers has been registered in different categories, scale and from different economies, in order to complete the micro-tasks on a small fee [19], [3], [6].

1.2 Problem Statement

As microtasking is one of the model of software crowdsourcing in which micro-workers from different economies participate in the completion of digital tasks [20]. It has been observed that it is difficult for crowdsourced platform to define the criteria for decomposition of tasks into microtasks [14]. Limited research has conducted to identify the existing microtasks and their categories. Less research has focused to microtasking in software crowdsourcing for task decomposition [21]. Due to these uncovered areas, quality of developed software projects and social recognition of crowdsourcing platforms being compromised. Therefore, this research fills this gap by focusing on enlisting the microtasking activities and methods of task decomposition.

Moreover, there is a need to develop a mechanism for task decomposition for crowdsourced platforms.

1.3 Research Questions and Objectives

This research opens following research questions:

RQ 1: What microtasking activities exist in crowdsourced software development?

The objective of this research question is to identify the microtasking activities which exist in crowdsourced platform.

RQ 2: What are the existing methods for task decomposition of microtasking in crowdsourced software development?

The objective of this research question is to identify the task decomposition methods of microtasking which exist in crowdsourced software development.

RQ 3: How task decomposition can be improved for effective microtasking in crowdsourced software development?

The objective of the research question is to propose the model for effective task decomposition in crowdsourced software development.

1.4 Aims of the research

This research is aimed to scrutinize the microtasking activities which exist in crowdsourced software development. After listing down those activities, research will explore the existing criterion, methods or models which have been used for decomposition of complex task into microtasks. Furthermore, research will explore the gaps among the decomposition of complex tasks. This research aims to develop the model for effective task decomposition, in order to achieve better results in microtasking related crowdsourced software development.

1.5 Scope of the research

This research is the description of microtasked related software crowdsourcing. It identifies the microtasking activities and their categories related to software crowdsourcing which exist in literature. It also identifies the methods which can decompose the complex tasks into the microtasks. Furthermore, this research presents the generic model which can decompose dependent, independent and hybrid tasks into microtasks.

1.6 Contributions of the research

This research has following contributions.

- This research will contribute by providing the list of microtasking activities that exist in the context of crowdsourced software development.
- This research will contribute by the identification of existing methods, models or approaches for task decomposition of microtasking in crowdsourced software development.
- This research will contribute by the development of an effective model for task decomposition of microtasking in crowdsourced software development.

1.7 Significance of research

- It extends the existing Software Engineering Body of Knowledge (SEBOK) by providing the microtasking activities and task decomposition methods for crowdsourced software.
- The study will also help software developers to perform task decomposition in CSD by using developed task decomposition model.

1.8 Layout of Thesis

The rest of thesis is organized as follows:

Chapter 2 provides literature review on crowsourcing, usage of crowdsourcing in softwre engineering, crowdsourcing plarforms and models, detailed description of microtasked crowdsourcing, utilization of microtasking in crowdsourced software development and existing microtasking platfroms. Chapter 2 also covers the existing studies related to microtasked related task decomposition in crowdsourcing.

Chapter 3 covers the research procedure, methodologies adopted for each research question, phases of the research and detailed desription of each research phase. It also describes the steps followed to conduct SLR. Comparative review and expert review are also described in this chapter.

Chapter 4 presents the overview of the studies which were found from four databases including IEEE Xplore, Springer Link, ACM and Science Direct. Microtasking activities and task decomposition methods are identifed from SLR which are reported in the chapter. Findings of comparative review and expert review are also described in the chapter.

In chapter 5, Hybrid model is proposed for effective task decomposition in crowdsourced software development.

Chapter 6 briefly concludes the research. It summarizes the findings of the research and presents the limitations and future work.

1.9 Summary

Chapter 1 covers the details of crowdsourcing, usage of crowdsourcing in different fields, brief description of models of crowdsourcing, utilization of microtasking in software development and importance of task decomposition in case of complex software development.

It describes the problem formulation, research questions and their objectives, aims and scope of the research. Contribution and significance of the research is also elaborated in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

In this chapter, comprehensive review of crowdsourcing, utilization of crowdsourcing in software engineering, crowdsourcing platforms and models is described. Furthermore, task decompositions in crowdsourcing have also reviewed. Review of microtasking (one of the models of crowdsourcing), how to utilize microtasking in crowdsourced software development, microtasking models and its platforms is also described. In the last section of the chapter, existing studies are reviewed and their analysis is reported.

2.2 Crowdsourcing

The practice of crowdsourcing has been emerged as problem solving approach to perform various tasks rapidly in a parallel manner [22]. It uses different mechanisms that integrate all participants at a single platform to accomplish the task. It is an outsourcing approach, first coined by Jeff Howe in 2006, who defined the term as "the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call. This can take the form of peer-production (when the job is performed collaboratively), but is also often undertaken by sole individuals" [23].

With the frequent emergence of utilization of world wide web; academia, industrial and government sectors have been harnessing the individual's intellectual capabilities; which makes the crowdsourcing as topical example [24]. It is being used in different fields, and for various purposes i.e., development of taxonomies, health and medicine, information exchange, solution

of complex projects, consensus, mapping outer space, content development, evaluation of products, biomolecule design, product and/or services development, crowdfunding, knowledge dissemination, purveying of geographic information, software development and software testing [25], [26].

As crowdsourcing has proven itself as important approach for integrating the best available cognitive abilities, so 85 percent of the top global brands are utilizing this approach to enhance their business, including Apple, Amazon and Microsoft on the top [27]. Moreover, different brands e.g., Unilever, Coca-Cola, LEGO, IBM, Google, Google Earth, McDonald's, Boeing, Samsung, Porsche, NASA, DARPA, Starbucks, Asus, Olive Union and PepsiCo are using crowdsourcing approach for their product design and other purposes [28], [29].

2.3 Crowdsourcing in Software Engineering

Software crowdsourcing is fast-expanding, distributed problem-solving approach by which software project develops by mutual effort of the online stakeholders, utilizing an open call format. It facilitates the software engineering process by integrating the various tasks (requirement elicitation, prototyping, designing, coding, testing etc.) performed by globally hired software engineers [30]. It anticipates crowdsourced software development in which large crowd of temporal workers participates into various tasks aiming to reduce barriers to on-boarding, geographical participation and remoteness, human mobility, compelling organizations and time-to-market by expanding parallelism [11], [30].

Crowdsourcing approach has been successfully applied to a wide range of applications such as Stack Overflow, Linux, YouTube, Wikipedia, Recaptcha, GoogleEarth and Yahoo Answers! [25], [31]. Another example is the creation of encyclopedia, which was developed by a pool of 70,000 participants which supports 290 languages with appropriately 35 million articles [5]. Moreover, famous software companies Facebook, Google, Microsoft, Apple and Netflix provides streamlined bug bounties regarding suspicious activities [32].

2.4 Crowdsourcing Platforms

As the usage of crowdsourcing has been expeditiously increased in the past few years, different crowdsourcing platforms have been developed to facilitate the individuals in various aspects i.e., programming and development, software testing, product design, template design, photography, research, content writing and user experience etc. [33], [34]. In the context of content development, Instagram, TED Translator/Open translation project, Twitter, Flickr, Facebook, Digg.com, Emporis.com, Amazon, Wikipedia, National Library of Austria and Project Gutenberg are the top crowdsourcing platforms [25].

In order to provide the solutions of complex software projects, some mature crowdsourcing platforms are TopCoder, Innocentive, IBM Innovation Jam, GetACoder, Kaggle, Crowdflower, NineSigma, Eyeca, Thinkfortigers, Nokia Idea Poject and OpenIdeo [25], [35]. In the context of provision of digital services, Amazon Mechanical Turk (AMT), Guru.com, Crowdmed, Turkit, Taskcn, FundingTree, Crowdforge, FundingCircle, Gigawalk and Elance are the topical examples [36]. Some of the platforms are developed to provide the software testing facilities e.g., 99Tests, Testbats, CrowdTesters, Pay4Bugs, uTest and Passbrains. Few platforms are providing the software security testing facility for their clients, amongst them Ce.WooYun and Bugcrowd are famous [37].

Crowdfunding is an approach of obtaining assets and funds a business or venture by the mutual effort of intellectual crowd in order to plan a business, accomplish a project or research the market trends [38]. Crowdfunding can be accomplished by the execution of any one of its business models i.e., investment-based crowdfunding and reward & donation-based crowdfunding. Investment-based crowdfunding is based on equity, lending and royalty where funders used to invest in a campaign in order to acquire fiscal advantages. Crowdcube, Indiegogo, Equitynet, Invesdor, Smart Angels and Kiva are the examples of investment-based crowdfunding platforms. Prosper.com, Spear, Babyloan are topical examples of lending-based crowdfunding platforms [39].

In contrast to investment-based crowdfunding, reward & donation-based crowdfunding includes the small-scale investors who are not predominately interested in financial rewards, takes apart in campaign, and receive a reward in the form of motivation for joining the crowd-

funding communities, to learn the skills, final product or service (mp3-files, a book). Kickstarter and Ulule are famous reward based crowdfunding platforms. Donation-based crowdfunding supports philanthropic, altruistic and artistic projects, and similar to the charities and NGOs which ask for contribution for a specific purpose. Famous platforms for donation-based crowdfunding are goFundme, DonorsChoose, MyLocalProject and UnitedDonations [40].

2.5 Task decomposition in crowdsourcing

As crowdsourcing is one of the most remunerative paragons of harnessing the collective effort of intellectual crowd. In order to achieve the fruitful results in crowdsourcing, it depends on the methods of decomposition of complex tasks into the number of autonomous and self-contained subtasks. After successful decomposition, subtasks are assigned to the crowd to find their solutions. In order to break the task into the small chunks, two decompositions are discussed in literature i.e., i) Horizontal task decomposition ii) Vertical task decomposition [41].

2.5.1 Horizontal task decomposition

It is a type of decomposition in which decomposed tasks do not depend on each other, hence one worker can perform each task independently, but quality of results can be compromised through this decomposition. In order to perform horizontal task decomposition, only one crowd-worker has to perform all the microtasks which are related to a specific complex task (say proof read a document). Initially, crowd-worker has to check the spelling, style and grammatical errors, then he has to fix those errors to proof read the document. In this decomposition, no other crowd-worker will verify the changes made by first crowd-worker; therefore, quality of the output can be compromised [42].

2.5.2 Vertical task decomposition

It is a type of decomposition in which decomposed tasks are dependent on each other, hence multiple workers are hired to perform each task individually, and therefore high-quality results can be achieved through this decomposition. For a particular complex task i.e., proof read a paragraph, decomposed microtasks are find-fix-verify which have to be performed sequentially [43]. To accomplish the task, each crowd-worker has to perform a microtask sequentially i.e., one crowd-worker will identify the spelling and grammatical errors, checking of writing style and coherence. After highlighting the errors, another crowd-worker has to fix those error which are identified by the first worker. For the 'verification' microtask, usually two or multiple crowd-workers are hired to verify the results performed by the second worker, hence quality results can be achieved.

2.6 Crowdsourcing Models

As crowdsourcing distinguishes itself from the other outsourcing models on the basis of following factors. i) The tasks are allocated by the utilization of an open call method, ii) the crowd workers are unknown to the platform owners or the organization and iii) the pool of crowd workers can be small as well as large, dependent on the nature of the complex task [44]. Four types of crowdsourcing models are used to elaborate: when and how open call method is used, what type of platform is best fit for the specific project, number of crowd workers to be chosen for the specific project and the expertise of the crowd workers [5].

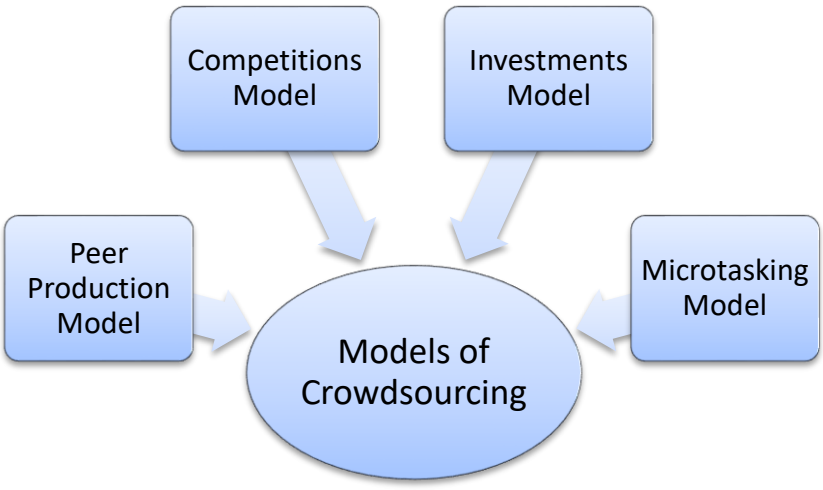

Figure 2.1: Crowdsourcing Models [5]

2.6.1 Peer Production model of crowdsourcing

Peer Production is one of the oldest and best-known models of the crowdsourcing in which crowd workers are known as collaborators, and they make contributions to the project for the sake of knowledge and experience gain, inspite of any monetary reward. Contributors set the goal, scope and boundaries of the project and the control of the project is not centralized to a single contributor [45]. It is the model of transformation and production which sets the goals for social motivation of the contributors, challenges the conformity of property and supports the continued conformity of the organization to the productive and inventive process [46].

In peer production model of crowdsourcing, contributors first understand the project's scope and architecture, project's conventions and design, environment and social values; then contribute to the project for the gain of experience with new technologies and social reputation [5]. Open-source software development is the best-known example of peer production model in which thousands of programmers contribute to develop the updated versions of software. Linux, Rails, Firefox and Apache are the software projects in which tens of thousands of people contribute for their development [45].

2.6.2 Competitions model of crowdsourcing

Competitions model is similar to the traditional outsourcing in which crowd workers are known as contestants and a client post a project on the crowdsourcing platform and pays for its successful completion [47]. A client proposed a project on the crowdsourcing platform, a copilot (an experienced worker paid for the task) distributes the work into multiple jobs known as competitions that might include the requirement elicitation, UI design, modelling, implementation and testing. Copilot then decomposes the competitions into multiple tasks that can be accomplished in couple of days. Each contestant provides a best competing solution, copilot selects the best solution provider as well as runner-up and they get financial reward [5].

Competitions model is suitable when clients need diverse solutions as well as higher quality results, which may increase the budget as well [47]. Topcoder is one of the famous platforms which implements the competitions model, where client posts the development related tasks and contestants compete the competitions [48]. Another platform 99designs (www.99designs.com) implements the competitions model to crowdsource the visual design tasks [49]. TestBirds and uTest are the specialized platforms for usability and system testing while GetACoder and TopCoder provide the entire development related tasks for competitions [50].

2.6.3 Investments model of Crowdsourcing

Investments model is related to the crowdfunding which supports 'the raise of funds by the general public (crowd-workers)' [51]. In this model of crowdsourcing, fundraisers and entrepreneurs raise the funds via crowdsourcing platforms which give them direct access to the interested supporters and market. There are investors, also known as crowd who bear risks to financially support the software project and expect certain recompense. Crowdsourcing platform acts as intermediary party which supports the interaction between fundraisers and investors [9].

Two different types of fundraising have been implemented in investment model of crowdsourcing; direct and indirect crowdfunding. In direct crowdfunding, fundraiser directly makes request to the selected and specific audience by using their own developed crowdfunding platform or website. On the contrary, indirect crowdfunding supports the raising of funds from unknown general public by using the existing crowdfunding platforms [52]. Numerous platforms are available for crowdfunding in which Fundable, Sandawe, SellaBand, Kiva, Kickstarter and Indiegogo are popular [9].

2.6.4 Microtasking model of crowdsourcing

Microtasking is such a model of crowdsourcing which supports the practice of distributed human computation in the form of provision of small, independent, less skill required and autonomous tasks to the workers, who are eligible for that task and available on the network [12]. It supports the decomposition of large and complex task into the number of short and self-contained microtasks that can be performed individually by workers. Complex task, often known as 'macro-task' decomposed manually into the number of independent and short tasks i.e., microtasks [53]. Manual decomposition of the macro-task depends on the nature of task as well as the expertise of information workers [53], [54]. For example, 'proof read a document' is a macro-task that can be transform into multiple microtasks which can be individually performed by the workers.

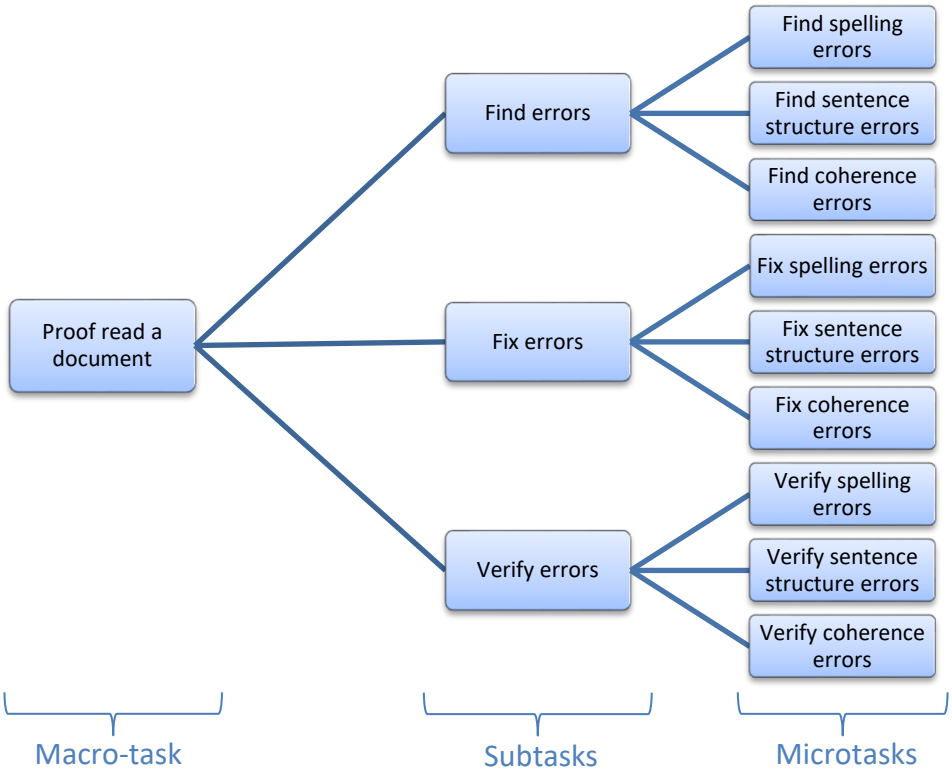

Figure 2.2: Decomposition of Macro-task into Microtasks [41], [42]

Figure 2.2 shows the decomposition of non-technical macro-task i.e., ‘Proof read a document’ into multiple microtasks. It elaborates the detailed decomposition of the given macro-task. Initially, macro-task is decomposed into three subtasks i.e., find errors, fix errors and verify errors. In the second level of decomposition, subtask ‘find errors’ is further decomposed into three microtasks. Workers have to find the spelling errors, sentence structure errors and coherence errors. Similarly, after the identification of all errors from the document, other workers have to fix all types of errors which are highlighted by the previous workers. Similarly, subtask ‘verify errors’ is decomposed into multiple microtasks i.e., verification of spelling errors, sentence structure and coherence errors. Each microtask is performed by individual worker which may take extra time for the accomplishment of task, but higher quality outcomes can be achieved [53]. Microtasking supports the decomposition of large and complex task into the number of short and self-contained microtasks that can be performed individually by workers. Complex task, often known as ‘macro-task’ decomposed manually into the number of independent and short tasks i.e., microtasks [53]. Manual decomposition of the macro-task depends on the nature of task as well as the expertise of information workers [53], [54]. For example, ‘proof read a document’ is a macro-task that can be transform into multiple microtasks which can be individually performed by the workers.

2.7 Microtasking in Crowdsourced Software Engineering

With the utilization of distributed human computation in non-technical tasks, microtasking model of crowdsourcing is often used for the accomplishment of technical tasks in software engineering [24]. In software engineering, microtasks are often known as microservices, which decompose complex web-based dependent tasks into short, independent and single-purpose tasks i.e., microservice [55]. It enables the tasks to build independently, to reuse and deployed individually and quickly.

In the context of programming in crowdsourced software engineering, microtasking can be achieved by two ways i.e., traditional and behavior-driven development (BDD) [11]. In traditional workflow, each crowd worker is assigned a separate task related to programming i.e., write a unit test for each behavior, test, implement and debugging of each test. In the traditional workflow of microtasking, crowd workers need frequent communication to discuss the outcomes of the implemented tasks and to ensure consistency.

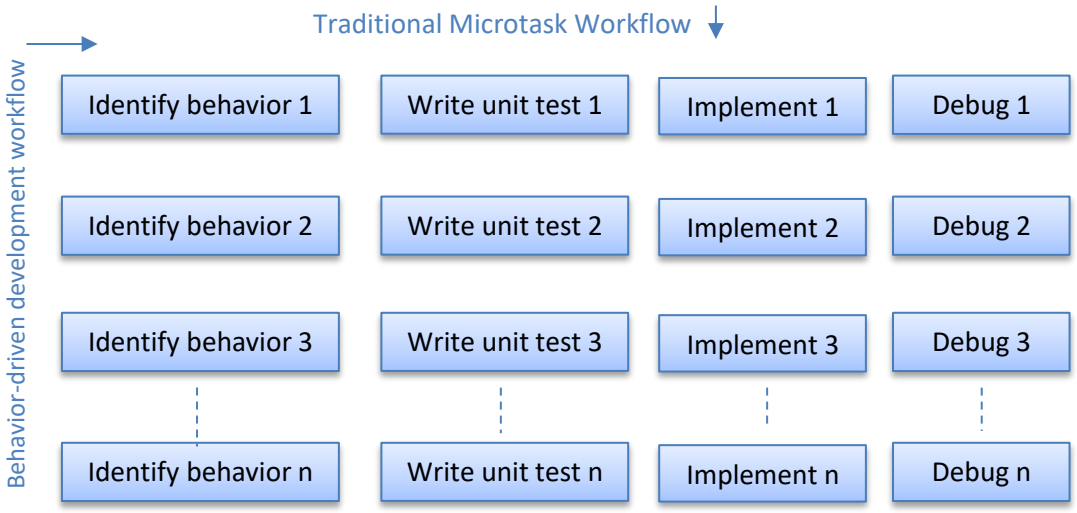

Figure 2.3: Traditional vs BDD workflow for software testing [11], [55]

Figure 2.3 shows the comparison between traditional and behavior-driven development workflows. In BDD, a crowd worker initially writes the unit test for each behavior, then implementation and debugging are performed by the same crowd worker. Behavior-driven development offers multiple advantages; for each behavior, single crowd worker is responsible for the identification, implementation and debugging of behavior within a function. Moreover, it saves the time as crowd workers do not need frequent conversations to understand the functionality of the behavior of function [55].

2.8 Microtasking Models

As microtasking is the shared human-computation among different workers, it can be achieved by practicing any of its two basic models. Utilization of its models depend on the nature of the task, expected outcome(s) of the task, skills of the participants, process management and reward given to the participants in terms of reputation, remuneration and experience [56].

The first model of microtasking supports crowd workers to perform small, independent and granular tasks for low remuneration i.e., few cents. This model invites the crowd workers

to perform atomic units of tasks in a parallel manner which require less cognitive effort and minimal skills; hence rewarded low incentive. Atomic unit of tasks are characterized as simple, require less cognitive effort, require single user and non-interactive with other tasks in terms of task execution [57]**.** Samasource is a platform which supports the accomplishment of atomic tasks such as image tagging, color and image identification [58]. These types of tasks are 'well structured' by their definition, 'well mapped-out' in terms of their execution and 'well-defined' in terms of their expected outcome.

The second model invites multiple crowd workers to perform interdependent and interactive tasks sequentially. As in interactive microtasking, approaches to decompose the tasks are limited and yet difficult to develop, hence tasks are interdependent i.e., input of one task is (maybe) the outcome of the previous task [56]. Different platforms e.g., Eyeca, uTest, Bombardier support the accomplishment of collaborative tasks i.e., debug the given piece of code, write a unit test according to the behavior of given function [25]. As these tasks require the special skills, software and environment to be performed, hence; collaborative efforts are required to execute them sequentially. Moreover, those tasks which lie under this model are ill-structured; their execution process and expected outcome are not well-defined.

2.9 Microtasking platforms

As the utilization of microtasking has been increased in recent years, different microtasking platforms have been developed to facilitate the clients in various aspects e.g., content access and development, software development, research, quality assurance and testing, designing and data translation etc. [24], [59]. Some microtasking platforms are famous for specific niche e.g., Quicktate and iDictate are known for microtasks which are related to the call auditing [60]. Similarly, TryMYUI is a microtasking platform which is known for the user experience related microtasks. Most of the platforms give their workers reward in terms of social recognition, self-interest, career opportunities, pleasure/fun, experience and financial reward [61], [62]. Different microtasking platforms and their supported task categories (microtasks related to specific area) are shown in Table 2.1.

Table 2.1: Existing online platforms for microtasking

Task Categories	**Microtasking Platforms**
Data manipulation	CrowdFlower (Figure Eight), Amazon Mechanical Turk (AMT), My little job, Crowd Guru, Click Worker, Fiverr, Field Agent, SmartCrowd, Guru, Freelancer, PeoplePerHour, Upwork, Swag bucks, Lion Bridge, RapidWorkers
Research	CrowdFlower (Figure Eight), Amazon Mechanical Turk (AMT), My little job, Prolific, Crowd Guru, Click Worker, ySense, Fiverr, Field Agent, SmartCrowd, Guru, Freelancer, PeoplePerHour, Upwork, Survey Junkie, Appen, Easy shift, PartTimeClicks
Sentiment Analysis	Amazon Mechanical Turk (AMT), Click Worker, Fiverr, Grinda Buck, Survey Junkie
Tagging and labeling	Crowd Guru, Fiverr, Guru, 99designs, FancyHands
Testing and Quality Assurance	ySense, Fiverr, SmartCrowd, Guru, Freelancer, Upwork, PeoplePerHour, Kashkick, Remotasks, EasyShift, Truelancer
Graphic designing	Fiverr, Skyword, Guru, Freelancer, Designhill, 99Designs, PeoplePerHour, Toptal, Upwork, TaskRabbit, Inbox Dollar, Zeerk, Crowdsource (One Space), SEOClerks, Lion Bridge
Content writing	Fiverr, Skyword, Freelancer, Guru, Writer Access, Humanatic, PeoplePerHour, Toptal, Upwork, Gigwalk, Remotasks
Programming and development	Topcoder, GrindaBuck, Fiverr, Guru, freelancer, Toptal, PeoplePerHour, Upwork
Music	FancyHands, Truelancer, Fiverr, Spare5, PeoplePerHour
Call auditing	Quicktate, iDictate, Scribie, InboxDollars
User Experience	TryMYUI

2.10 Studies related to Microtasked Crowdsourcing

Researcher came across many studies as shown in Table 2.2. The most recent work is done by Huan Jiang et. al in 2020 [43], who presented the guidelines for the decomposition of sequential tasks (tasks which are interdependent). Another recent work has been done by Shinobu Saito et. al in 2020 [54], who conducted the case study of application of microtask programming. The authors presented the concept of insourcing of project from the organization, and developed the web-based application by utilizing the slack time of the crowd-workers of the organization.

Table 2.2: Existing studies related to microtasked crowdsourcing

Year	Author	Paper	Contribution of the paper
2014 [41]	Huan Jiang, Shigeo Matsubara	Efficient Task Decomposition in Crowdsourcing	This paper presented the task decomposition models. Horizontal task decomposition model was developed for independent tasks which are performed in parallel manner. Vertical task decomposition model was developed for dependent tasks which are executed sequentially.
2015 [63]	Muhammad Allahbakhsh, Saeed Arbabi, Masood Shirazi, Hamid-Reza Motahari-Nezhad	A Task Decomposition Framework for Surveying the Crowd Contextual Insights	The paper proposed a new method of polling (questionnaire) for crowd-workers which involves the decomposition of a questionnaire into multiple questionnaires and assigning them to multiple crowd-workers.
2015 [64]	Alessandro Bozzon, Marco	Designing Complex Crowdsourcing	This study proposed the tools and approaches for crowd-based

	Brambilla, Stefano Ceri, Andrea Mauri, Riccardo Volonterio	Applications covering Multiple Platforms and Tasks	workflows. They presented the modelling concepts that used their developed workflow patterns for the decomposition of complex task into the small and simple interactive subtasks.
2018	Yongxin Tong, Lei Chen, Zimu Zhou, H. V. Jagadish, Lidan Shou	SLADE: A Smart Large-Scale Task Decomposer in Crowdsourcing	The study proposed greedy heuristic algorithm for homogeneous task decomposition and efficient (proved by experiments) framework for heterogeneous decompositions.
2018 [14]	Thomas D. LaToza, Arturo Di Lecce, Fabio Ricci, W. Ben Towne, Andre van der Hoek	Microtask Programming	Instead of traditional form of crowdsourcing, the authors have proposed the crowdsourced programming which is done by achieving small tasks i.e., microtasks.
2019 [11]	Emad Aghayi, Thomas D. LaToza, Paurav Surendra, Seyedmeysam Abolghasemi	Implementing Microservices through Microtasks	The authors developed novel workflow by using behavior driven development to decompose the programming related tasks.
2019 [21]	Sebastian Heil, Valentin Siegert, Martin Gaedke	Crowdsourced Reverse Engineering: Experiences in Applying Crowdsourcing to Concept Assignment	Authors described crowdsourced reverse engineering of automatic task extraction, result aggregation & quality control and source code anonymization.

2019 [55]	Emad Aghayi, Thomas D. LaToza, Paurav Surendra, Seyedmeysam Abolghasemi	Crowdsourced Microservices: Behavior-Driven development Applied to Microtask Programming	Authors described the Behavior-Driven Development (BDD) workflow which implemented microservices through microtasks. The term microservices used to decompose complex web-based tasks into the independent, short and simple microtasks.
2020 [43]	Huan Jiang, Shigeo Matsubara	Efficient Task Decomposition for Sequential Crowdsourced Task Solving	The authors gave guidelines for the decomposition of sequential (the tasks which depends on each other) tasks in crowdsourcing.
2020 [54]	Shinobu Saito, Yukako Limura, Emad Aghayi, Thomas LaToza	Can Microtask Programming Work in Industry?	The authors reported case study of application of microtask programming. The authors developed web-based application by utilizing the slack time of workers of the same organization.

Huan Jiang et. al [41] described the vertical and horizontal tsk decompositions which are performed in crowdsourcing. According to the authors, task can be executed vertically as well as horizontally. Vertical decomposition model was developed for dependent subtasks and horizontal decomposition model was defined for independent subtasks. Authors took proofreading as a crowdsourcing example and execute its subtasks (find, fix, verify) by both models i.e., vertical and horizontal. They concluded that in general, vertical task decomposition model outperforms the horizontal decomposition model, in terms of improving the quality of executed tasks [41].

In 2015, Saeed Arbabi [63] and his companions discussed the methods of polling (example of questionnaire) and proposed a new method which is based on decomposition of a poll into multiple sub-polls. Proposed method (framework) has three components; first component was named as decomposition unit which used to decompose a poll into multiple sub-polls (fragments of questionnaire). Second component was the recruitment unit which assigns the decomposed sub-polls to the sufficient number of suitable crowd-workers. Third component was the re-composition unit which includes the recruitment of suitable participants for sub-polls, which remain non-recruited in the recruitment unit due to lack of suitable crowd-workers. The proposed framework was implemented and tested on Stack Overflow platform [63].

Alessandro Bozzon et.al [64] described the approach for prototyping, designing and pattern-based instruction of crowd0based workflows. The proposed workflow model is transcribed into executable instructions which includes control data and event-based notifications. For the designing of complex crowd-based workflow, initially high-level workflow schema is designed in which execution of simple tasks is explained along with their coordination. Furthermore, simple tasks (microtasks) i.e., like, modify, group, classify and choose etc. are identified and their properties are observed. For designing of crowd-based application, six phases are studied which include operation, object, performer, workplan, platform and UI design. For the complex task i.e., proofreading, decomposition pattern was developed which includes subtasks e.g., create, decide, improve, find, compare and fix. The developed concept have been implemented in the CrowdSearcher which is a crowd management system that supports the demonstrative patterns of crowd-based workflows and applications; and harness runtime interface for explicit deployment of crowd-based applications and web-based prototypes to the client networks and crowdsourcing platforms [64].

Yongxin et.al [65] investigated the general crowdsourcing task decomposition problem i.e., Smart Large scale task Decomposer (SLADE) which decomposes the largescale crowdsourcing task. For homogeneous SLADE problem, they proposed greedy heuristic algorithm. However, for heterogeneous SLADE problem, they proposed effective and efficient approximation guideline which is developed by using Optimal Priority Queue (OPQ). The authors have conducted experiments on Mechanical Turk to verify the effectiveness of proposed

solution. Experiments proved that proposed algorithms are efficient, effective and guaranteed their approximation [65].

In 2018, Thomas et. al [14] manually decomposed the programming related tasks into the number of microtasks which were then assigned to the crowd-workers. Decomposed programming related tasks were: implementation of a part of function, write a unit test, updating a call site and testing a piece of code. Microtasks have been generated iteratively in order to nurture the quality of final product. Authors have conducted experiments on CrowdCode and their findings depicted that microtasking enables the crowd-workers to onboard onto a project easily and quickly. Study examined that crowd workers were able to complete 1008 microtasks, complete every type of microtask in less than 5 minutes on average, and submitted their first microtask in less than 15 minutes. Instead of spending several days, microtasks enabled the participants to contribute to the project easily and quickly [14].

Emad Aghayi et.al [11] described the novel workflow by using behavior-driven development (BDD) for the decomposition of programming related tasks into self-contained multiple microtasks. Each microtask involves the identification, testing, implementation and debugging of single function within a function. The authors conducted case-study to compare the BDD workflow with traditional workflow and concluded that 350 microtasks completed, 13 functions implemented and contributed 5 new behaviors in less than 24 minutes. Developed workflow is suitable for small size software development [11].

In 2019, Sebastian Heil and his companions [21] addressed the idea of reverse engineering and used its application i.e., concept assignment. The authors have proposed classification technique for automatic task extraction, source code anonymization, result aggregation and quality control. The authors conducted experiments on crowdsourcing platform microworker.com to verify their proposed technique and concluded the effectiveness of technique in terms of efficiency. As a result, experiments produced 187 results by 34 workers which classified 10 code fragments. According to the researcher, an evaluation with larger budget, more concrete and tailored measures between client and crowd workers will produce further insights in future [21].

Thomas et.al [55] described behavior-driven development (BDD) approach for the implementation of microservices through microtasks. Microservices decomposed the complex web back-end tasks into short and single purpose services. Instead of traditional workflow, behavior driven development approach allows only one participant to perform all tasks i.e., identify, test, implement and debug; which are related to a single behavior of a function. In traditional workflow, each participant performs separate task e.g., single crowd-worker identifies all behaviors of a function, second crowd-worker implements all the behaviors and so on. Through behavior driven development approach, communication time between the crowd-workers (to understand the system) can be saved [55].

In 2020, Huan et. al [43] investigated the sequential task crowdsourcing i.e., the tasks which are interdependent. According to their developed guidelines, tasks should be arranged in such a way that most difficult task should be placed first and the easiest task at the last, then execution of the task takes place according to the given hierarchy. Guidelines are applied on tasks related to the proof reading and conducted the experiments on Amazon Mechanical Turk (AMT), concluded that task decomposed by developed guidelines achieved quality results [43].

In 2020, Shinobu et.al [54] used the concept of insourcing of project. Instead of utilizing the efforts of crowd-workers from outside of the organization for the development of web-based application, the authors utilized the slack time of workers of the same organization. According to the expertise of software engineer of the organization, he decomposed the tasks manually; non-technical and independent tasks were given to the crowd-workers to be performed; however, technical and interdependent tasks were performed by himself. As the authors developed web-based application, so this study only highlighted the micro specifications related to the design and development (front-end as well as back-end) [54].

Several researchers studied the methods to decompose the task into microtasks, designing of complex crowdsourcing applications and implementation of micro-services through microtasks. Few studies developed the behavior-driven development approach for task assignment and presented the comparison between traditional workflow and behavior-driven development approach. However, the studies focus mainly on task assignment, sequential task decomposition and manual decomposition of programming related tasks.

Limited studies have been conducted for task decomposition frameworks in which a study presented by Muhammad et al. [63] is a notable exception, which decomposed the survey related tasks into microtasks. Other studies [11], [14], [54], [55] discussed the types of microtasks and manually decomposed the programming related tasks (which are technical in nature) into multiple microtasks. Less research has focused on dynamism of decomposition of complex tasks into short and self-contained microtasks.

Due to the absence of generic model which can decompose the technical and non-technical tasks into microtasks, industry is facing problems regarding to the decomposition of complex task and execution of microtasks in crowdsourced software development. Furthermore, due to the lack of execution mechanism of microtasks, industry is unable to fully leverage the nature of microtasking model of crowdsourcing. This study aims to identify the types of microtasks (microtasking activities) which exist in crowdsourcing environment, identification of task decomposition methods and to propose the task decomposition model (equally suitable for technical and non-technical tasks) in crowdsourced software development.

2.11 Summary

Chapter 2 covers comprehensive description of crowdsourcing, usage of crowdsourcing in software engineering, crowdsourcing platforms and models, detailed description of microtasking (one of the models of crowdsourcing), utilization of microtasking in crowdsourced software development, microtasking models, existing microtasking platforms and existing studies.

CHAPTER 3

METHODOLOGY

3.1 Overview

The objective of this chapter is to describe the methodologies which are used to pursue the research. Research design and procedure will be described in Section 3.2 in which methodologies adopted for each research question are reported. In Section 3.3, overall research process will be described which explicitly shows the research phases. After that, each research phase along with their research methodology will be explained in detail. At the end, chapter will be summarized in Section 3.4.

3.2 Research Design and Procedure

Research methodology is an important aspect of doing research, as it is an essential ingredient for research that tells how research is to be carried out [66]. It is the systematic way to investigate the issue from different perspectives [67]. In Software Engineering, it may consists of combination of different techniques in order to explore the issue in depth and more accurate [68].

The Systematic Literature Review (SLR) was conducted to identify the microtasking activities and task decomposition methods which exist in literature. This SLR comprised of comprehensive review of studies which were related to the types of microtasks i.e., microtasking activities and the methods which are used to decompose the complex tasks into the number of microtasks. The SLR gives the full leverage to conduct the fair and detailed review of literature, due to its built-in search strategy. The purpose to conduct this SLR was to

identify the microtasking activities, their categories and groups. Furthermore, SLR was conducted to come up with a list of task decomposition methods which industry is utilizing to divide the complex tasks into the number of short and simple tasks i.e., microtasks.

Besides SLR, comparative review was conducted to critically analyze the existing methods of task decomposition. Moreover, expert review was conducted to survey the industry's opinion in order to validate the findings of SLR and comparative review. Table 3.1 shows the summary of research methodologies which are used to accomplish the research objectives.

Table 3.1: Research summary

Research Questions	**Research Objectives**	**Methodology**	**Outcome**
What microtasking activities exist in crowdsourced software development?	To identify the microtasking activities which exist in crowdsourced software development	Systematic Literature Review (SLR) (B. Kitchenham, 2007; B. Kitchenham, et. al., 2009)	List of microtasking activities which exist in crowdsourced software development.
		Expert review (Boring, et. al., 2005)	List of validated microtasking activities along with their categories and groups.
What are the existing methods for task decomposition of microtasking in crowdsourced software	To identify the methods, approaches, frameworks and models which decompose the complex task into	Systematic Literature Review (SLR) (B. Kitchenham, 2007; B. Kitchenham, et.	Task decomposition methods, approaches, frameworks, models and prototypes.

development?	multiple microtasks	al., 2009)	
How task decomposition can be improved for effective microtasking in crowdsourced software development?	To compare and critically analyze the existing task decomposition methods	Comparative Review (P. Vartiainen, 2015)	- Classification of existing task decomposition methods for technical and non-technical tasks. - Classification of dependent, independent and hybrid task groups. - Check the suitability of existing task decomposition methods for dependent, independent and hybrid task groups.
	To develop the model for effective task decomposition	Expert Review (Boring, et. al., 2005)	- Validated task groups - Proposed Task Decomposition Model

3.3 Research Process

Overall research process is shown in Figure 3.1. The research was primarily consisted of three main phases. Phase 1 is related to the identification of microtasking activities and existing methods for task decomposition. In order to do so, Systematic Literature Review is conducted.

Phase 2 of the research is related to the comparison and critical analysis of existing methods which are used for task decomposition. Comparative review is conducted to differentiate the dependent, independent and hybrid task groups and to check the suitability of existing methods for those task groups.

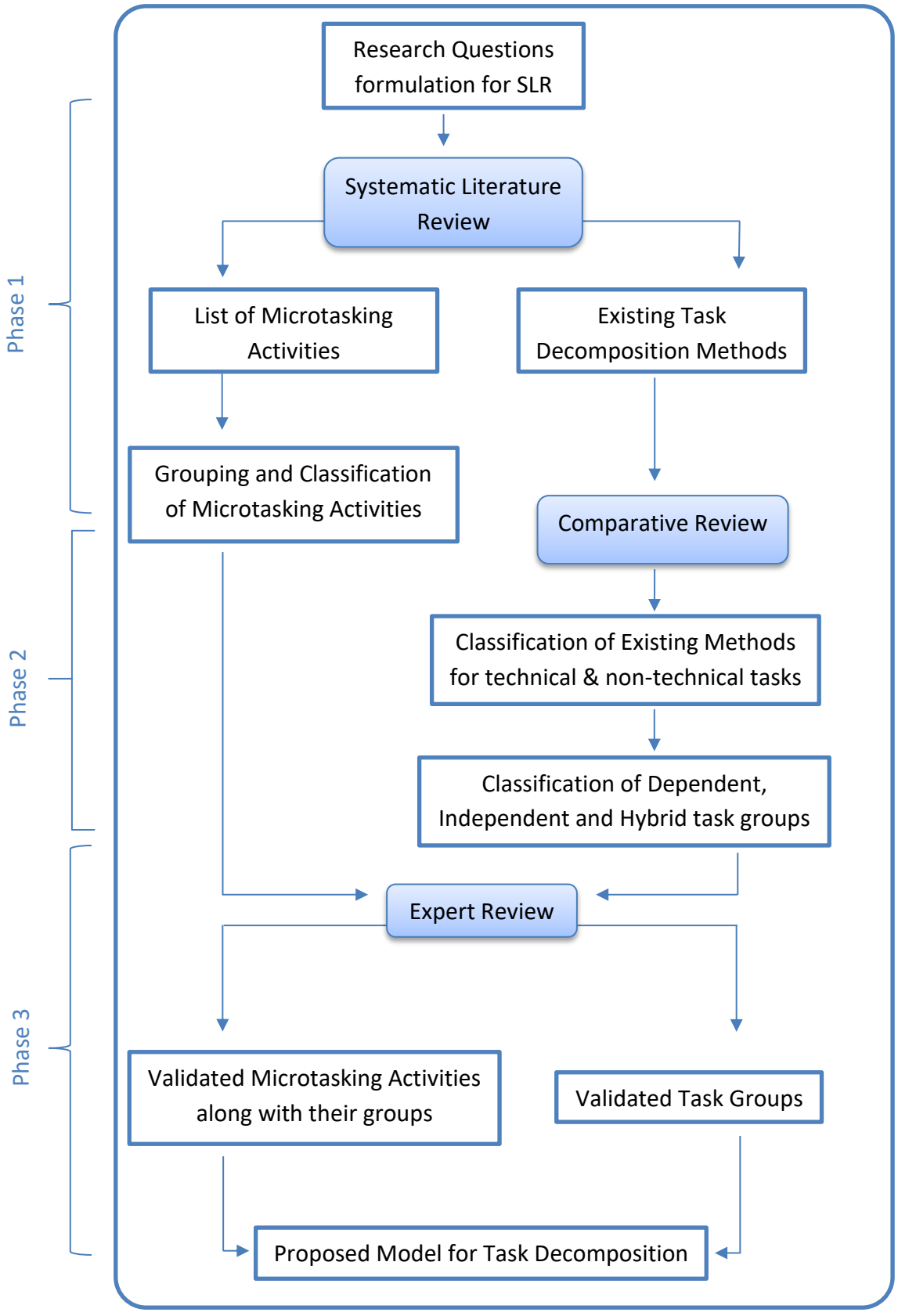

Figure 3.1: Overall Research Process

Phase 3 of the research is related to the validation of list of identified microtasking activities and the classification of task groups. Expert review is conducted to evaluate the naming conventions of identified microtasking activities, microtasking categories and positioning of microtasking activities in the relevant categories. Moreover, expert review is performed to validate the nature of dependent, independent and hybrid tasks and to classify them into their relevant groups.

3.3.1 Research Phases

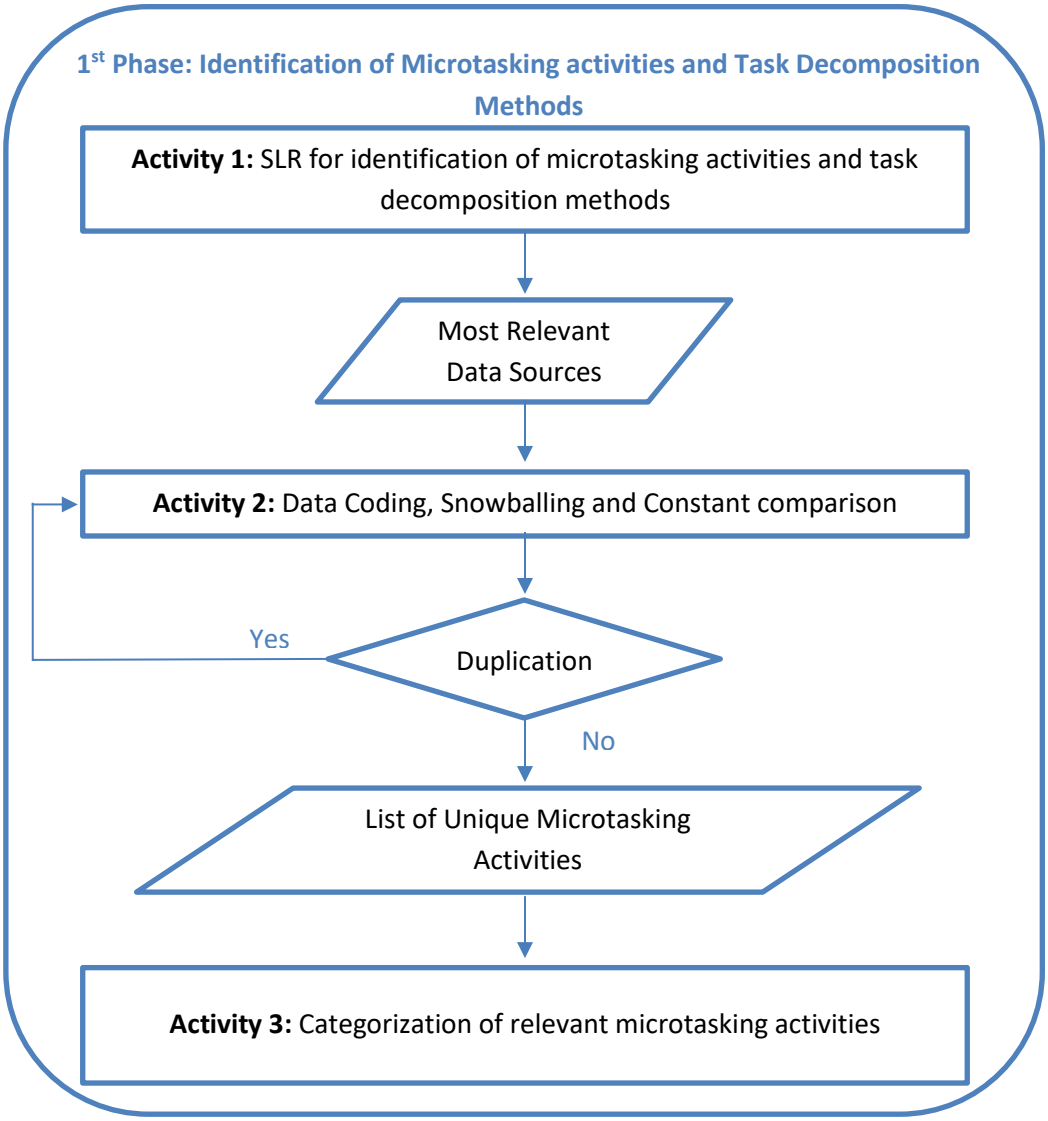

Figure 3.2: Phase I of research

Figure 3.2 shows the first phase of research which is related to the identification of microtasking activities and task decomposition methods which exist in literature. In order to do so, three activities are performed; i) SLR is conducted, ii) Constant comparison, snowballing and data coding techniques are applied to review the literature in-depth and to transform the identified microtasking activities into meaningful terminologies, iii) categorization of microtasking activities into their relevant groups.

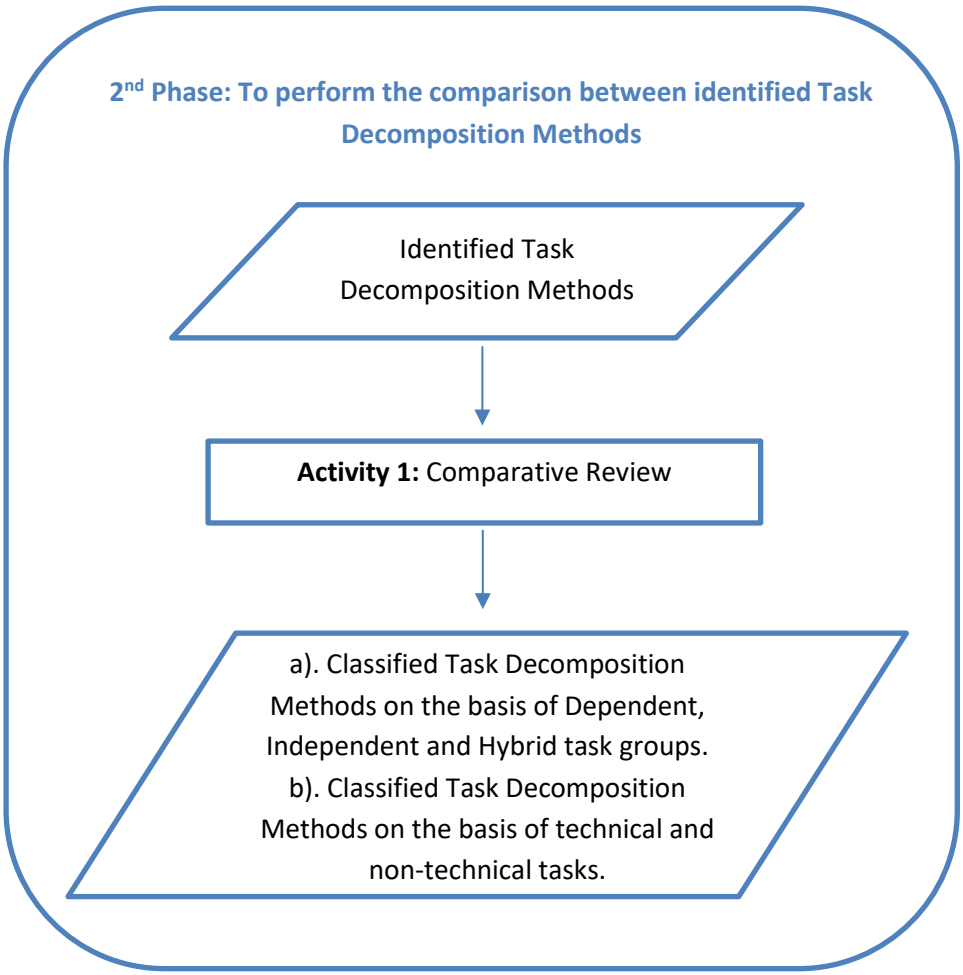

Figure 3.3: Phase II of research

Figure 3.3 shows the second phase of the research which was related to the comparison and critical analysis of identified methods which are used to decompose the complex task into the number of simple and shorts tasks i.e., microtasks. In this phase, identified task decomposition methods act as input. Comparative review was performed to classify the

identified task decomposition methods on the basis of dependent, independent and hybrid task groups. Furthermore, identified task decomposition methods are also classified on the basis of technical and non-technical tasks. Each identified task decomposition method was critically analyzed to check if it is equally suitable for the decomposition of dependent, independent and hybrid tasks.

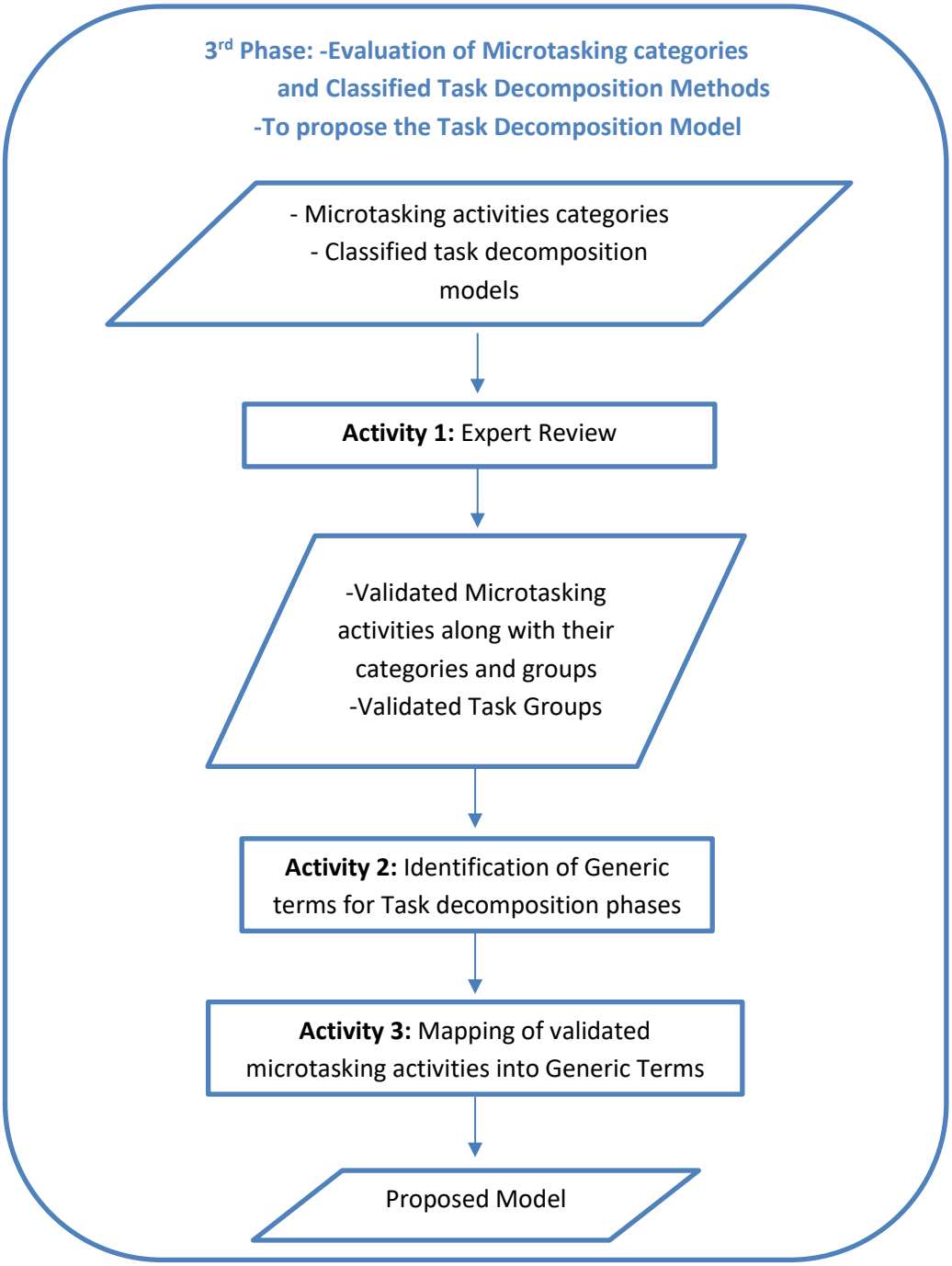

Figure 3.4: Phase III of research

Figure 3.4 describes the third phase of research which is related to the evaluation of microtasking categories and identified task groups. Expert review is conducted to validate the naming convention of microtasking activities as well as their categories and groups. In the previous phase, identified task decomposition methods were classified on the basis of dependent, independent and hybrid task groups. In this phase, task groups are validated by the experts of industry. Furthermore, task decomposition phases are identified and generic names are given to them. At the end, task decomposition model is proposed and shown in Chapter 5, Section 5.3.

3.3.2 Systematic Literature Review (SLR)

In order to identify the microtasking activities and task decomposition methods from the literature, SLR was conducted. In this research, work of Kitchenham is followed as it gives full leverage to conduct the comprehensive SLR in the field of software engineering [69]. SLR comprised of three steps; Review planning, Review conduction and Result Reporting. Figure 3.5 shows the overview of SLR steps.

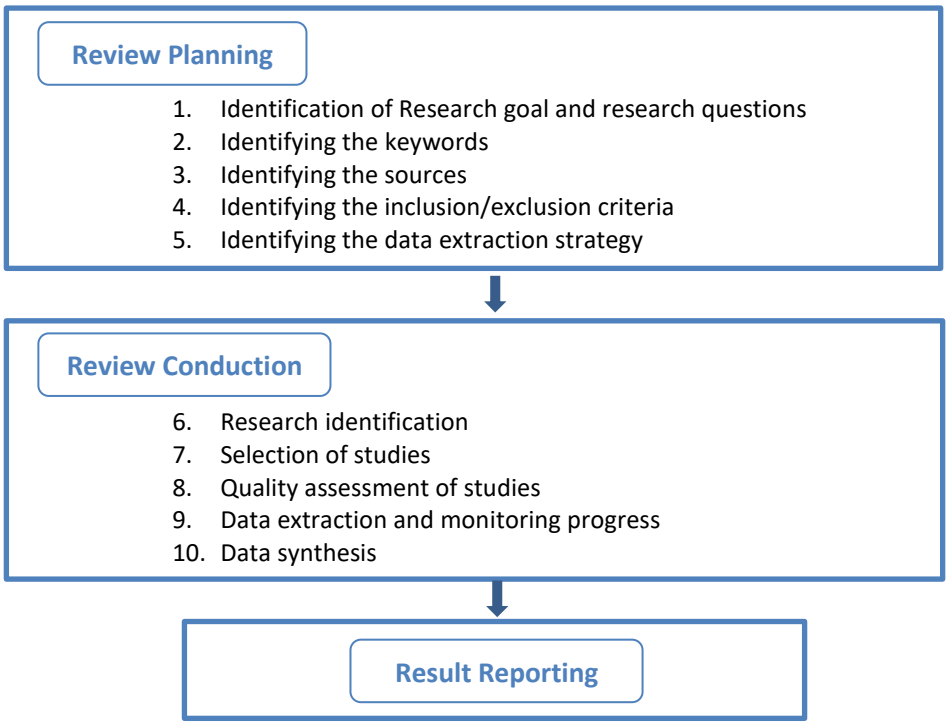

Figure 3.5: Overview of steps of systematic literature review [69]

Research Goal and Research Questions: Systematic Literature Review was aimed to come up with the list of microtasking activities and task decomposition methods in crowdsourced software development which exist in literature. The unique list of microtasking activities along with their categories and groups, and identified task decomposition methods answered the first and second research question (RQ: 1, RQ: 2) of the study.

Identification of keywords: Major terms (keywords) of Microtasking, Activities, Methods, Task Decomposition and Crowdsourced Software Development were highlighted. Alternate spellings and synonyms of keywords were identified and shown in Table 3.2.

Table 3.2: Synonyms of keywords used for systematic literature review

Keywords	Synonyms
Microtasking	Microtask, Small tasks, Simple tasks, short tasks, Decomposed tasks, Microtasks, Independent tasks, Micro-task.
Activities	Types, Kinds, Tasks, Actions.
Crowdsourced Software Development	Crowdsourcing, Software crowdsourcing, Software outsourcing, Crowdsourced development, Crowdsourced software, Crowdsourced computing.
Methods	Mechanisms, Models, Approaches, Techniques, Frameworks, Ways, Workflows.
Task Decomposition	Task distribution, Task breakdown, Task allocation, Task assignment, Task division, Task disintegration.

The search string for both research questions was formulated on the basis of major terms and their synonyms, shown below.

Search string for RQ: 1

(Microtasking OR All synonyms of Microtasking) AND (Activities OR All synonyms of Activities) AND (Crowdsourced Software Development OR All synonyms of Crowdsourced Software Development)

Search string for RQ: 2

(Methods OR All synonyms of Methods) AND (Task Decomposition OR All synonyms of Task Decomposition) AND (Microtasking OR All synonyms of Microtasking) AND (Crowdsourced Software Development OR All synonyms of Crowdsourced Software Development)

Search Process: In this step of SLR, search sources were highlighted. Four databases were selected names as; ACM, IEEE, Springer Link and Science Direct. These databases were selected due to the availability of numerous software engineering researches and literature in them. Each database was searched for journal articles, conference papers, workshop papers, and chapters of book and published thesis. The data sources i.e., research articles which were related to the microtasking activities and task decomposition methods in perspective of crowdsourced software development was considered as relevant. Researches from 2011 to 2021 were searched from the databases and relevant studies were included in this research. The distribution of data sources for each database is shown in Chapter 4; Section 4.2 and 4.3.

Inclusion/ Exclusion Criteria: For searching each database, comprehensive inclusion/ exclusion criteria were applied on searched data sources i.e., research articles. Applied inclusion/ exclusion criteria is described below:

- All those papers are included from the mentioned databases which either addressed the microtasking activities in general, microtasking activities in crowdsourced software development or discussed the microtasks which exist in software development.

- All those papers are included from the mentioned databases which either highlighted the task decomposition methods, used task decomposition prototypes or developed task decomposition workflows or approaches.

The studies which were excluded from SLR are explained below:

- Papers which were not having any of the keywords; Microtask, Task Decomposition, Methods and Crowdsourced Software Development.
- Table of contents, or giving information related to the proceedings of conference and workshops, or on the basis of their title.
- Papers whose title was in English language but their remaining content was in any other language.
- Papers which were repeated in data sources were excluded in the later stages.

Quality Assessment: Quality of selected papers was evaluated by following the checklist from the work of Kitchenham. Table 3.3 shows the checklist used in this research for determining the quality of selected studies.

Table 3.3: Quality assessment criteria for selection of papers for SLR

S. No	Question	Answer
1.	Are the aims clearly stated?	Yes/ No /Partially
2.	Are the findings credible and important?	Yes/ No /Partially
3.	Are the prediction techniques used clearly described and their selection are justified?	Yes/ No /Partially
4.	Is the knowledge or understanding been extended by the research?	Yes/ No /Partially
5.	Is the diversity of perspective and context been explored?	Yes/ No /Partially
6.	Are the links between data, interpretation and conclusions are clear?	Yes/ No /Partially
7.	Does the detail/ depth/ complexity of the data is conveyed?	Yes/ No /Partially

Questions mentioned in the checklist were answered by the researchers who were selected to read the papers for their quality assessment. It was a collaborative process in which selected research articles were randomly allocated to the postgraduate students. They were organized in two groups and each group consisted of 7 members. In total, 77 research papers

(after applying inclusion/ exclusion criteria) were randomly allocated among two groups. Each member was provided with 11 papers; hence each paper was reviewed by two respondents, in order to get fair assessment. Specifically, each respondent of each group was given 11 papers to assess their quality, based on the above-mentioned criteria shown in Table 3.3. The scoring scale was Yes = 1, Partially = 0.5, No = 0.

The feedback of respondents against each question mentioned in quality assessment criteria was recorded and given the values according to the scoring scale. For each paper, scores of their two respective respondents were accumulated. Those papers whose accumulated summed values were ranging from 0.5 to 1 were selected. From total of 77 papers, it was found that accumulated values of 17 papers were below 0.5. hence, remaining 60 papers were selected in which 26 papers were selected to identify the microtasking activities which exist in crowdsourced software development, and 34 papers were selected to identify the task decomposition methods. The quality scores of each paper are tabulated in quality assessment tables which are shown in Appendix A and B.

Data Extraction: The data extracted from each research article was: Data source (database), Title, Publication type (journal, conference, book chapter, thesis), Conference/ Journal/ Book/ Thesis name, Publication year, Author's name, Methodology applied in the paper, Microtasking activities (for RQ1) and Task decomposition methods (for RQ2). The example of selected research article's information form is shown in Table 3.4 which shows the Paper Id (unique Ids were allocated to each paper), Data source (database from where that paper was downloaded), Title of the research article, Author's name, Publication year, Publication type, Name of conference/ journal, Methodology and findings. The identified data extraction unit information for each research article was maintained in tabular form and shown in Appendix C.

Table 3.4: Research article's information form

Data extraction unit	**Data extraction unit information**
Paper Id	P02
Data Source	ACM
Title	Break It Down: A Comparison of Macro and Microtasks
Author	Justin Cheng, Jaime Teevan, Shamsi Iqbal

Year of Publication	2015
Publication Type	Conference
Conference/Journal Name	ACM Conference on Human Factors for Computing Systems (CHI)
Methodology	Experiment
Findings	Copyediting

Result Reporting: After the quality assessment of resarch articles, selected studies were reviewed and the findings of SLR are comprehensively described in Chapter 4 Section 4.2.2 and 4.3.2.

3.3.3 Comparative Review

To compare the identified task decomposition methods, the work of P.Vartiainen [70] is followed, which is comprehensive approach for comparative study. To conduct the comparative review of identified task decomposition methods, four steps has been followed which includes selection of evaluation object, level of comparative evaluation, conceptual comprehension and analysis of evaluation research results.

Selection of evaluation object: Comparative review was aimed to compare the features of identified task decomposition methods. Existing task decomposition methods were classified to check their suitability for the decomposition of technical and non-technical tasks. Task groups i.e., dependent, independent and hybrid are selected as an evaluation object. Identified methods are evaluated on the basis of selected evaluation objects.

Level of comparative evaluation: In the selection of level of comparison, one has to select if comparison will be conducted between national and international organizations, or government and private organizations. In this comparative review, similarities and dissimilarities of the selected evaluation objects have been identified which are reported in Chapter 4 Section 4.3.3.

Conceptual comprehension: In this step of comparative review, concepts and definitions of selected evaluation objects and their results are clearly defined.

Analysis of evaluation research results: After the identification of evaluation objects and building of comprehensive concepts about the dependent, independent and hybrid task groups, results of the comparison are analyzed on the basis of selected evaluation objects and reported in Chapter 4 section 4.3.3.

3.3.4 Expert Review

Once a list of microtasking activities was generated and task decompsotion methods were identified, the outcome was forwarded to industrial experts. The primary aim of expert review was not only the validation of naming conventions and categorization of identified microtasking activities, but also the validation of classification of task groups. In order to conduct the expert review, expert opinion elicitation guideline by Ayyub was followed, as it is mature and detailed guideline to gather expert opinions [71]. Figure 3.6 shows the steps which have been followed to conduct expert reviews.

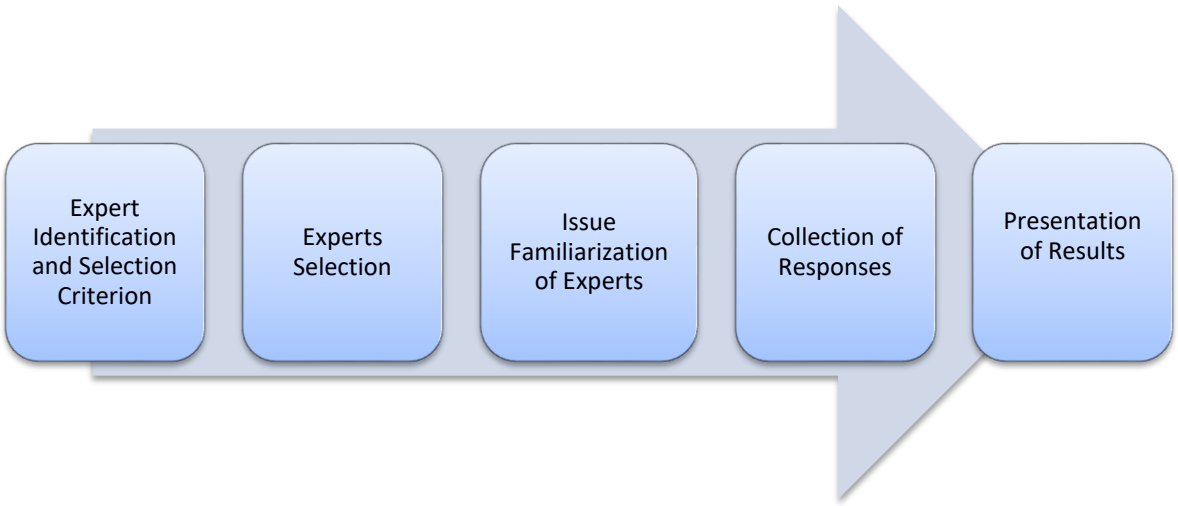

Figure 3.6: Steps to conduct Expert review [71]

Expert Identification and Selection Criterion: To conduct the expert review for this study, expert selection criterion is given below:

- Must be Software Engineering professionals either from academia or industry.
- Must be specialized in crowdsourced software development.
- Must have knowledge of microtasked crowdsourcing.
- Must have more than 5 years of experience.

According to the above-mentioned criterion, six experts were contacted and four experts showed their interest to give the time and valuable feedback. The details of selected experts are shown in appendix D and expert evaluation document is shown in appendix E. Initially, experts were familiarized with the steps they were supposed to perform in order to review the naming conventions, categorization and grouping of identified microtasking activities. They were asked to review the list of microtasking activities and task groups. The recommended changes were applied to the microtasking list which is described in Chapter 4, Section 4.2.3.

3.4 Summary

Chapter 3 covers the overall research procedure, methodologies adopted for each research question, phases of the research and detailed description of each phase. Furthermore, steps followed to conduct Systematic Literature Review (SLR), expert review and comparative review are also described in this chapter.

CHAPTER 4

RESULTS

4.1 Overview

The objective of this chapter is to describe the findings of research questions. Section 4.2 will describe the overview of studies which are found to identify the microtasking activities, their quality assessment details, and extracted data from the filtered studies in the form of list of microtasking activities. Furthermore, a detail of expert review is explained which was conducted to validate the naming conventions of microtasking activities and to check the correct positioning of microtasking activities under microtasking categories. Section 4.3 will describe the overview of studies which were found during SLR, identified task decomposition methods and their comparison.

4.2 Overview of the studies related to RQ1

As discussed in Chapter 3, Systematic Literature Review was conducted to find the microtasking activities. Four databases were explored to come up with a set of papers from where microtasking activities are finally extracted. The overall schematic view to find the microtasking activities is illustrated in Figure 4.1 which shows the papers selection procedure including the first search results along with inclusion/exclusion criteria and quality assessment.

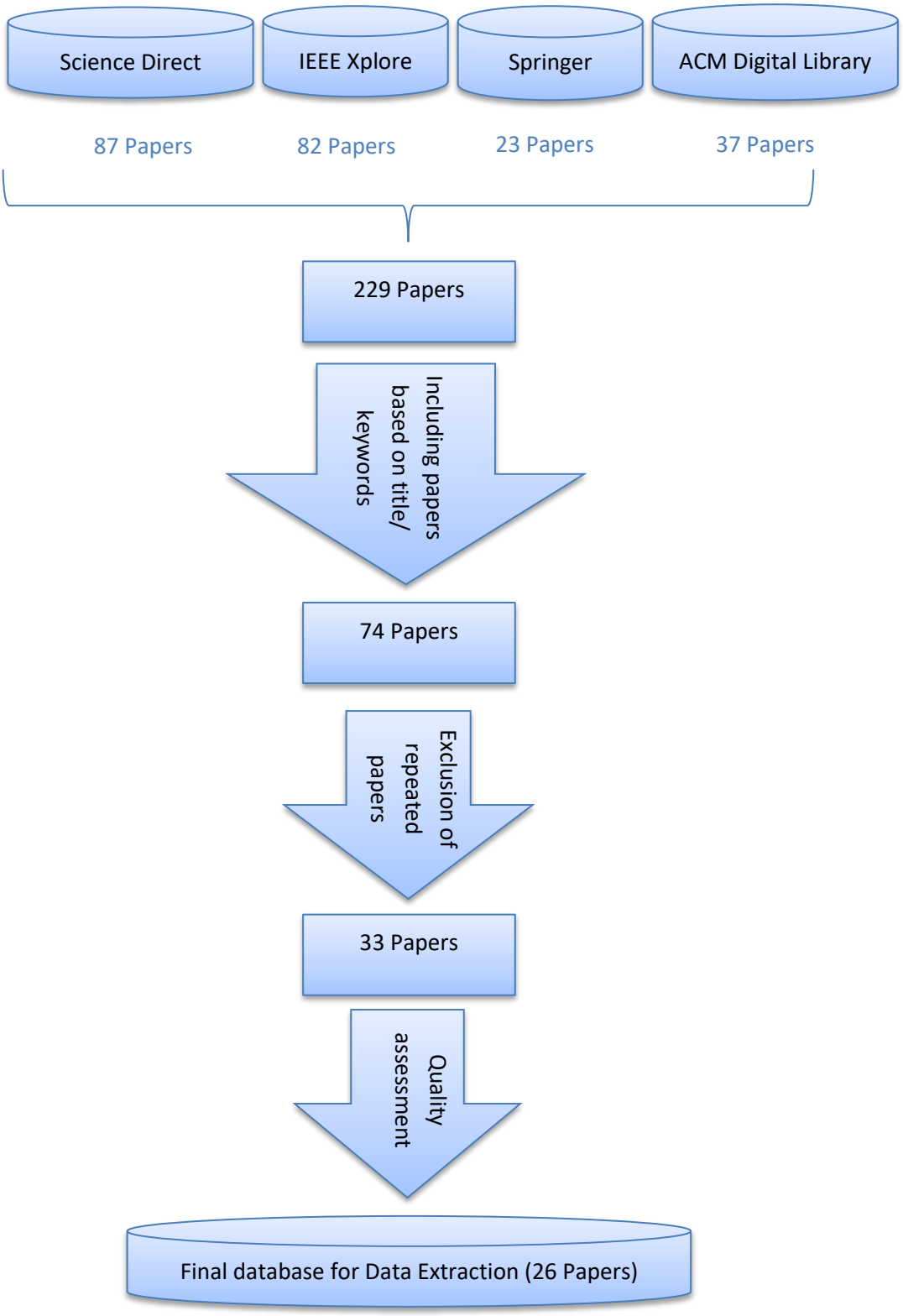

Figure 4.1: Studies selection procedure to find the microtasking activities

Figure 4.1 shows the overall procedure of selection of studies. Initially 229 papers were found when four (IEEE Xplore, ACM, Science Direct and Springer Link) databases were explored by the search string mentioned in Chapter 3 Section 3.3.2. After applying first inclusion/exclusion criteria, 74 papers were selected. The papers whose title and keywords were related to the microtasking activities in crowdsourced software development, were selected in the first step of inclusion/exclusion. In the second step, repeated papers and the papers whose content is not in English language were excluded and a total of 33 papers were selected in this step. After applying inclusion/exclusion criteria, quality of the selected papers has been assessed by followed the criteria as explained in Chapter 3 Section 3.3.2. Details of quality assessment is shown in Appendix A.

After quality assessment, a total of 26 papers were selected for the identification of microtasking activities. Number of retrieved papers from each database is shown in Table 4.1.

Table 4.1: Source databases for identification of microtasking activities

Database	**Retrieval date**	**Number of retrieved papers**
ACM	February 2021	7
IEEE Xplore	February 2021	7
Springer Link	February 2021	5
Science Direct	February 2021	7

4.2.1 Unique Paper's ID for RQ1

After quality assessment, a total of 23 papers were selected to identify the microtasking activities which exist in crowdsourced software development. Unique IDs were allocated to each paper which is shown in Table 4.2.

Table 4.2: Unique IDs for papers to find microtasking activities

S. No	**Paper ID**	**Paper Title**
01	P01	A Taxonomy of Microtasks on the Web [72].

02	P02	Break It Down: A Comparison of Macro- and Microtasks [53].
03	P03	The Effect of Peripheral Micro-tasks on Crowd Ideation [73].
04	P04	Estimating Conversational Styles in Conversational Microtask Crowdsourcing [74].
05	P05	ReLauncher: Crowdsourcing Micro-Tasks Runtime Controller [75].
06	P06	SimilarHITs: Revealing the Role of Task Similarity in Microtask Crowdsourcing [76].
07	P07	A Profile-Aware Microtasking Approach for Improving Task Assignment in Crowdsourcing Services [6].
08	P08	Exploring Microtask Crowdsourcing as a Means of Fault Localization [77].
09	P09	Microtask Crowdsourcing Marketplace for Social Network [78].
10	P10	Large-Scale Microtask Programming [79].
11	P11	A Crowd-in-the-Loop Approach for Generating Conference Programs with Microtasks [80].
12	P12	Human Beyond the Machine: Challenges and Opportunities of Microtask Crowdsourcing [81].
13	P13	Microtask Programming [14].
14	P14	Toward Microtask Crowdsourcing Software Design Work [26].
15	P15	Task assignment in microtask crowdsourcing platforms using learning automata [82].
16	P16	Microtasking: redefining crowdsourcing practices in emergency management [12].
17	P17	Implementing Microservices through Microtasks [11].
18	P18	Crowdsourced Reverse Engineering: Experiences in Applying Crowdsourcing to Concept Assignment [21].
19	P19	Quality Assurance Strategies in Microtask Crowdsourcing [32].
20	P20	Motivation of Workers on Microtask Crowdsourcing Platforms [83].
21	P21	Crowdsourcing Platforms: Objective, Activities and Motivation [25].
22	P22	Efficient Task Decomposition in Crowdsourcing [41].
23	P23	A Task Decomposition Framework for Surveying the Crowd Contextual Insights [63].

24	P24	Crowdsourced Microservices: Behavior-Driven development Applied to Microtask Programming [55].
25	P25	Efficient Task Decomposition for Sequential Crowdsourced Task Solving [43].
26	P26	Ask the Crowd: Scaffolding Coordination and Knowledge Sharing in Microtask Programming [16].

Microtasking activities which exist in crowdsourced software development are identified from each selected paper (shown in Table 4.2). Data extracted from each paper is maintained and extraction tables for each paper are shown in Appendix C.

4.2.2 Identified Microtasking Activities

The purpose of first research question was to come up with the list of microtasking activities which exist in crowdsourced software development. A total of 72 microtasking activities have been found through Systematic Literature Review. Depending on the nature and execution process of the identified microtasks, relevant microtasking activities are then grouped into different categories. Generic names are given to the categories e.g., all identified microtasking activities which are related to the translation and conversion of one language or file format to the other, are places under the category of 'Data transcription'. Table 4.3 shows the identified microtasking activities along with their categories. Furthermore, a short description against each microtasking category is explained in the table.

Table 4.3: Identified Microtasking Activities

S. No	**Categories**	**Microtasking activities**	**Description about microtasking categories**	**Papers**	**Frequency**
1.	Information finding	-Metadata finding -Organizing the data -Information finding -Data collection	Any type of microtask which requires the information finding e.g., author name of any research	[72], [53], [75], [76], [6], [78], [80],	9

		-Information gathering -Filtration and synthesize the data	article, its published date and exploring of email address of owners from company's website.	[81], [12],	
2.	Verification & validation	-Content verification -Spam detection -Data matching -Data tagging -Product comparison	Microtasks which are used to verify and validate the data e.g., verification of service providing by the freelancers against description written in their offer (or gig).	[72], [74], [76], [6], [78], [81], [12], [21], [41], [63], [43], [83].	12
3.	Content creation	-Data categorization -Data enhancement -Data classification -Data selection -Gathering of terms for taxonomy creation -Dataset's module creation -Label an image -Pasting the data -Data mapping -Addition of annotations -Restructure the data into standardized reports -Documentation -Listing of data -Organizing the data -Data collection	Any small task which can be used to create the content e.g., gathering of words/terms for the creation of taxonomy, conversion of random data into the useful information, and then into the standardize documents.	[72], [53], [74], [75], [76], [6], [78], [80], [81], [12], [21]. [25], [83].	13
4.	Data	-Media	Microtasks	[72],	8

	transcription	transcription -Data translation -Image transcription -Transcribing the speech's sentences -Digitizing local-language documents -Human Optical Recognition tasks -Audio translation -Language translation -Video translation	related to translation and transcription e.g., translation of a paragraph, audio and video from one language to another, or converting the image file into the editable text file etc.	[53], [74], [75], [6], [78], [12], [32].	
5.	Interpretation & Analysis	-Sentiment analysis -Content moderation -Data Analysis -Data interpretation -Interpretation of visual data -Checking and listing of websites	Microtasks related to the opinions and feelings of a specific entity e.g., "What do you think about new features launched in iPhone X?". These tasks depend on the intelligence and perception of the crowd.	[72], [74], [75], [76], [78], [81], [12], [21], [41], [55], [32].	11
6.	Surveys	-Content feedback -Conduct an interview	Microtasks which are related to the survey for taking feedback against the products and services.	[72], [76], [78], [81], [12], [63], [32], [83].	8
7.	Content access	-Promotion e.g., webpages -Copying of the data -Content access	These microtasks usually require the workers to consume time, visit the website	[72], [74], [76], [6], [81],	8

		-Capture the photos -Sharing of data with different sites -Logging of information onto a page -Watch an online video	or any specific location and access the data by clicking on the link provided e.g., "Click on the link below and watch the animated video for further understanding".	[12], [41], [83].	
8.	Quality assessment	-Debugging of program -Debugging of UI -Test a line of code -Algorithmic debugging -Delta debugging -Implement a unit test -Identify, test, implement and debug behaviors in code -Locate known faults in code fragments -Review of function behavior	Microtasks to ensure the quality of a code, design or interface e.g., identification and removal of errors from the code, algorithm and design.	[72], [77], [79], [14], [11], [41], [43], [16], [32], [83].	10
9.	Designing	-Designing a single component of logo -Selection of fonts -Sketching of small design related to interface -Identification of design problems	Microtasks related to the graphic designing.	[73], [26],	2
10.	Development	-Writing a piece of code	Microtasks related to the	[72], [77],	9

		-Writing test-cases -Edit a function -Adding pseudo-code -Implementing part of a function -Human computation	programming.	[79], [14], [26], [11], [55], [16], [32].	
11.	Identification	-Identification of main decision points from set of requirements -Identification of alternative solution -Identification of missing values in the dataset.	Microtasks related to the identification of any missing data or information. These microtasks are performed by the experts of their relevant fields.	[72], [77], [79], [26], [82],	5

It has been observed that number of microtasking activities related to ‘Content creation’ are higher in frequency i.e., 13 papers addressed the microtasks related to creation of content. On contrary, microtasks related to ’Design’ category is least in frequency. Only 2 papers have discussed the microtasks which are related to designing field.

4.2.3 Validation of Identified Microtasking Activities

In order to validate the identified microtasking activities, expert review has been conducted. As described in Chapter 3 Section 3.3.4, four experts were finally selected to collect their opinion regarding the identified microtasking activities. Selected experts were software engineers, specialized in crowdsourced software development and had knowledge of microtasked related crowdsourcing. 2 experts have been working with reputable software organizations and the 2 were from academia who had in-depth knowledge of microtasked related crowdsourcing. Demographic data of experts and expert evaluation form is shown in appendix D and E respectively.

Experts validated the naming conventions of microtasking activities and their position under the microtasking categories. According to the expert, "*There is duplication of few microtasking activities e.g., 'Data collection' and 'Gathering of terms' are conveying the same meanings".* According to another expert, "*Please write short description against each microtasking activity to let the reader know about the actual function of that microtask".* According to the feedback of another expert, "*The microtasks 'Data analysis' and 'Data interpretation' are same".*

According to the experts, "*The microtasks 'Image transcription' and 'Audio transcription' comes under the microtask 'Media transcription'".* According to the opinion of an expert, "*The microtasks 'Label an image' and 'Data tagging' are same in nature. Give them a generic name and merge them or write an argument to present a difference between them in their respective description column".* According to the feedback of another expert, *"Create a link between each microtasking activity and its respective category in the form of description".* According to another expert, *"'Data collection' lies in two categories, remove it from 'Information finding' category".*

The recommended changes are applied to the list of microtasking activities along with their categories. A total of 72 microtasking activities were found from SLR, after applying the recommended changes, researcher came up with 61 unique microtasking activities along with their description. Table 4.4 shows the validated list of unique microtasking activities.

Table 4.4: Validated Microtasking Activities

S. No	Categories	Microtasking activities	Description about microtasking activities
1.	Information finding	Metadata finding	Microtasks which require the author's name, published date, conference/journal's name etc. of a published article and email addresses of corresponding employees from the company's website.
		Information finding	Microtasks which require the general information e.g., information about the company in the USA, find the cheapest air fare in the selected dates and destination.
		Data Filtration	Microtasks which require to filter the data in order to find the particular information e.g., apply the filter in MS Excel to find how many students got A grade.
		Data synthesize	Microtasks which require the grouping of different modules of data in order to make it specific information.
2.	Verification & validation	Content verification	Microtask which requires the verification of content e.g., check if the particular company website describe the correct business.
		Spam detection	Microtask which requires to verify if spam filter is working correctly.

		Data matching	Microtask which requires to verify if data extracted by one worker is correct. Microtasks which are related to verification of service providing by the freelancers against description written in their offer list.
		Data tagging	Microtask which allows users to organize information more efficiently by associating pieces of information e.g., Give a suitable tag for a given product or service.
		Product comparison	Microtask which requires the comparison between given products e.g., their names, their brand names, their quantity etc.
3.	Content creation	Data categorization	Such microtask involves the organization of entities into groups and subgroups with the same features, in order to create a content e.g., Name the categories for same features.
		Data classification	Such microtask involves the assigning of entities to classes according to predetermined principles e.g., Choose the most suitable product for each category.
		Data enhancement	Such microtask involves the addition of data/content related to specific topic e.g., wiki loves to explore about your area, Write few lines about the specific area.
		Data selection	Microtask which requires to select the words/terms, images, audio or video for the specific topic in order to create the content.
		Gathering of terms for taxonomy creation	Microtask which requires to add the words/terms or gathering the selected words/terms in a document, for the creation of taxonomy.

		Dataset's module creation	Such microtask which is related to addition of entities in rows and columns of dataset, in case of missing data. Or according to one's expertise, addition of data to create a small module of dataset.
		Label an image	Such microtask which involves the detailed description of the given image e.g., Write description of the given image.
		Pasting the data	Microtask which involves the pasting of data at a suitable/given position of the document or site for the content creation.
		Data mapping	Such microtask involves the matching of data fields from one database/site to another in order to integrate and manage the data for content creation.
		Addition of annotations	Microtask which involves the addition of comments to check the user's or viewer's perspective.
		Listing of data	Microtask which involves the creation of list e.g., Make a list of Engineering Universities from a given content.
		Organizing the data	Such microtask involves the organization of data to check if the categories and content under them, hierarchy, link and flow of the content is correct.
		Restructure the data into standardized reports	Microtask which involves the documentation i.e., formatting of given content, in order to convert into a given standardized report.

4.	Data transcription	Media conversion	Microtask which is related to the conversion of image, audio or video into the other formats i.e., .jpeg, .png, .mp3, .mp4, .ai etc.
		Media transcription	Microtask which is related to the translation of image, audio or video into the other languages.
		Data translation	Microtask related to the translation of data from one language to another. For example, Type what you see in the following captchas.
		Human Optical Recognition tasks	Microtask which is related to the conversion of image file into the editable text file e.g., Convert the given image into the editable text file by using any OCR (Optical Character Recognition) software.
		Digitizing local-language documents	Microtask which involves the creation of text files from any image, audio or video file.
5.	Interpretation & Analysis	Sentiment analysis	Microtasks related to the opinions and feelings of a specific product e.g., “What do you think about new features launched in iPhone X?”
		Content moderation	Microtasks related to the moderation of content for guideline violations, spam or inappropriate content e.g., Moderate the images for inappropriate content.
		Data Analysis and interpretation	Such microtask depends on the intelligence and perception of the crowd. For examples, identify if the given tweets are positive, negative, or neutral.

		Interpretation of visual data	Microtask related to the interpretation of given image or video. For example, identify the gesture conveying by the man in the given image.
		Checking and listing of websites	Microtask which involves the listing of the given websites according to the given criteria.
6.	Surveys	Content feedback	Such microtask required the crowd workers to give feedback against the service or product they have used. For example, help us improve our website.
		Conduct an interview	Such microtask required the crowd workers to give an interview against specific product, service or a day e.g., Mother's Day and Father's Day interview.
7.	Content access	Promotion e.g., webpages	In such microtasks, workers are asked to access and consume content e.g., "Visit the webpage by clicking on the provided link." or Visit the website or any specific location and access the data by clicking on the link provided.
		Copying of the data	Such microtasks involves the access of the content and copy it for future tasks.
		Content access	Such microtasks require the crowd workers to simply access the content e.g., "Read the information by following the website link".

		Capture the photos	Microtask which involves the access of content, product, or any landscape by simply capturing it.
		Sharing of data with different sites	Such microtask involves the sharing of information (can be in any format e.g., text, image or video) or your website's, YouTube channel's link to the other sites to allow the viewers to access the content which are intended to provide them.
		Watch an online video	These microtasks usually require the workers to consume time e.g., "Click on the link below and watch the animated video for further understanding".
8.	Quality assessment & Testing	Debugging of program	Microtask to ensure the quality of a program e.g., "Identify and remove the errors from a given program".
		Test a line of code	Microtask to ensure the quality of a line of code e.g., "Identify if any error exist in the given line of code, if so, correct them".
		Debugging of UI	Microtask to ensure the quality of User Interface e.g., "Check if the color scheme, font face, font size, positioning of images with respect to text, white spacing and alignment are according to the design brief".
		Implement a unit test	These microtasks usually require the workers to implement a unit test in order to ensure the quality of a single unit of code.

		Algorithmic debugging	Microtask to ensure the quality of an algorithm e.g., “Identify if any error exist in the given algorithm, if so, correct them”.
		Delta debugging	These microtasks usually require the workers to ensure the quality of program or piece of code by using given automated debugging tool.
		Identify, test, implement and debug the behaviors in code	Such microtask involves the identification and removal of errors according to given programming behavior, and then implement the code with new changes and debug again.
		Locate known faults in code fragments	In these microtasks, pieces of code with errors are provided to the crowd workers and ask them to remove the errors to ensure the quality of the given code fragments.
		Review of function behavior	These microtasks usually require the workers to ensure the behavior of the function in the code for which it was intended to develop.
		Implementing part of a function	These microtasks usually involve the implementation of a function. For example, “Run the given piece of code into your system. (This implementation is for quality assurance purpose that if the given function runs smoothly at all operating system’s versions, browsers, plug-ins and machines)”.

9.	Designing	Designing a single component of logo	Microtasks related to the designing of components of logo. For example, “Draw the crescent shape of a given color in vector form”.
		Sketching of small design related to interface	These microtasks usually require the workers to design the small components of the interface e.g., “Draw the home shaped colorful vector by using basic shapes”.
		Selection of fonts	These microtasks usually require the workers to select the appropriate fonts according to design brief e.g., “Select three appropriate fonts for logo design of a given brand design brief”.
10.	Development	Writing a piece of code	Microtask which requires the workers to write a piece of code according to given requirement, in order to develop the system.
		Writing test-cases	According to the give system’s perspective, these microtasks usually requires the workers to write the test-cases, in order to develop the test suites.
		Edit a function	These microtasks usually involve the modification of a given function(s) of a program e.g., “Change the name and parameters of the function according to the given details”.

		Adding pseudo-code	These microtasks involve the addition of pseudo-code and comments in the code, to make the reviewers clear about the functionality performed by the code.
		Human computation	These microtasks usually involve the human computation in order to develop any (can be public) system. For example, “Circle all the cats in a potato” or “Identify the pictures in white horse is seen”.
11.	Identification	Identification of main decision points	These microtasks usually involve the identification of main decision points from the set of requirements. For example, “SRS document has been attached, you are required to identify the main decision points from it”.
		Identification of alternative solution	These microtasks usually involve the identification of alternative solutions of a given problem. For example, “Design brief document has been attached, you are required to provide the alternatives of the problems with brief description”.
		Identification of missing values in the dataset	Microtasks related to the identification of any missing data or information from the given dataset or the dataset’s brief. These microtasks are performed by the experts of their relevant fields.

4.2.4 Validation of Categorization of Microtasking Activities

In order to validate the microtasking categories, expert review has been conducted. The experts who validated the microtasking activities, also validated the microtasking categories. According to the opinion of an expert, "*The name of a microtasking category 'Quality assessment' should be 'Quality assessment and Testing'".* Experts suggested to combine the relevant microtasking categories and group the categories into the major groups. According to the expert, 'Content Creation' and 'Content Access' are the relevant categories, so they should place into the 'Content' group. By following the suggestions of the experts, researcher came up with 5 microtasking groups which are unique in their functionality. Table 4.5 shows the grouping of microtasking categories.

Table 4.5: Grouping of microtasking categories

S. No	Groups	Microtasking Categories
1	Content	Content creation
		Content access
2	Survey	Surveys
		Identification
3	Information	Information finding
		Interpretation and Analysis
4	Data transcription	Data transcription and Translation
5	Code implementation	Designing
		Development
		Verification and Validation
		Quality assessment and Testing

4.3 Overview of the studies related to RQ2

As discussed in Chapter 3, Systematic Literature Review was conducted to find the task decomposition methods. Four databases were explored to come up with a set of papers from

where task decomposition methods were explored. The overall schematic view to find the task decomposition methods is illustrated in Figure 4.2 which shows the papers selection procedure including the first search results along with inclusion/exclusion criteria.

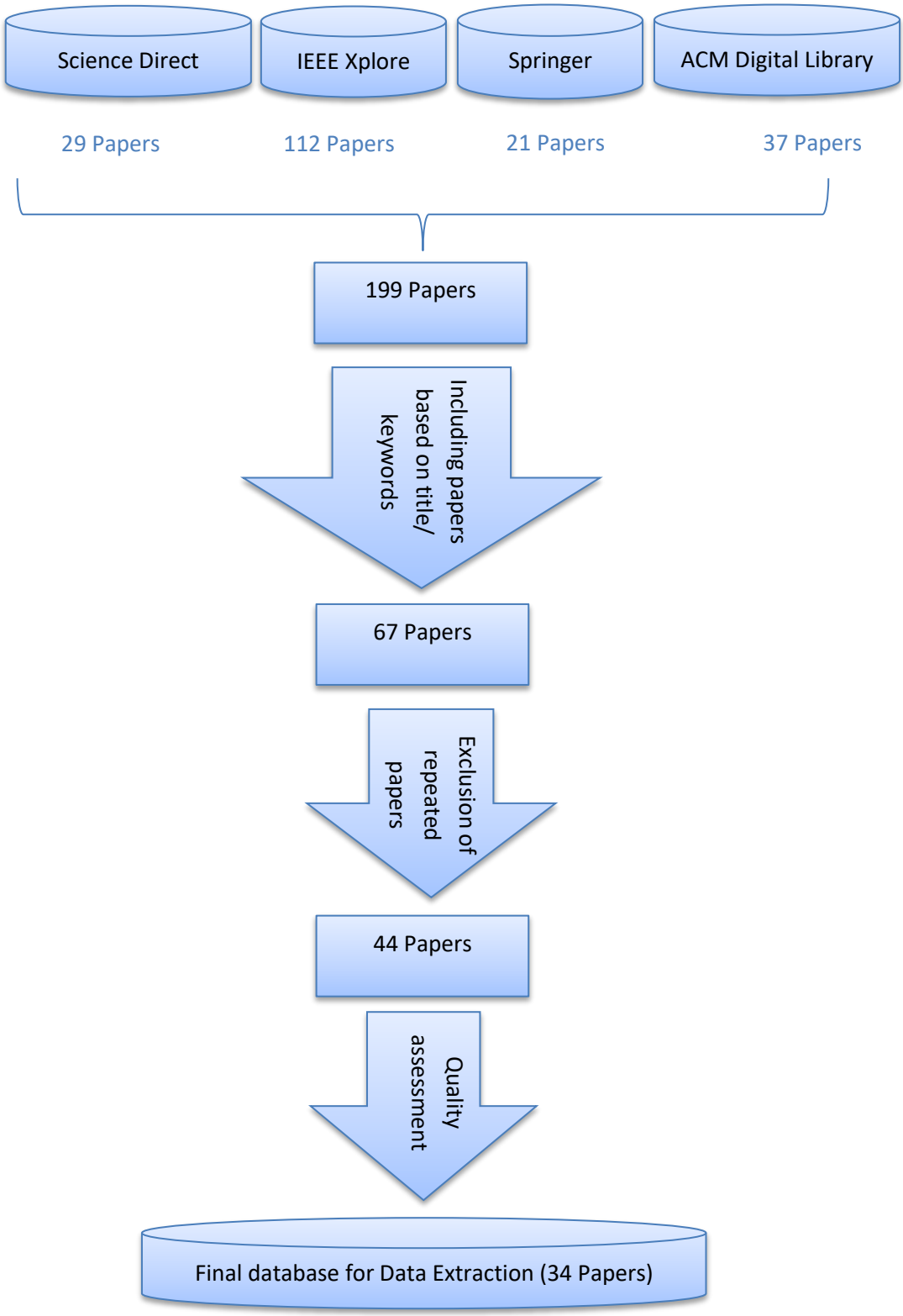

Figure 4.2: Studies selection procedure to find the task decomposition methods

Figure 4.2 shows the overall procedure of selection of studies to identify the task decomposition methods. Initially 119 papers were found when four (IEEE Xplore, ACM, Science Direct and Springer Link) databases were explored by the search string mentioned in Chapter 3 Section 3.3.2. After applying first inclusion/exclusion criteria, 67 papers were selected. The papers whose title and keywords were related to the microtasking activities in crowdsourced software development, were selected in the first step of inclusion/exclusion. In the second step, repeated papers and the papers whose content was not in English language were excluded and a total of 44 papers were selected in this step. After applying inclusion/exclusion criteria, quality of the selected papers has been assessed by followed the criteria as explained in Chapter 3 Section 3.3.2. Details of quality assessment procedure is shown in Appendix B.

After quality assessment, a total of 34 papers were selected for the identification of task decomposition methods. Number of retrieved papers from each database is shown in Table 4.6.

Table 4.6: Source databases for identification of task decomposition methods

Database	**Retrieval date**	**Number of retrieved papers**
ACM	February 2021	11
IEEE Xplore	February 2021	12
Springer Link	February 2021	6
Science Direct	February 2021	5

4.3.1 Unique Paper's ID for RQ2

After assessing the quality of papers, unique IDs were allocated to each paper which were selected to find the task decomposition methods Table 4.7 shows the unique paper IDs for second research question of the research.

Table 4.7: Unique IDs for papers to find task decomposition methods

S. No	Paper ID	Paper Title
01	P27	Adaptive Task Assignment for Crowdsourced Classification [84].
02	P28	Cognition-based Task Routing: Towards Highly-Effective Task-Assignments in Crowdsourcing Settings [85].
03	P29	Batch allocation for decomposition-based complex task crowdsourcing e-markets in social networks [86].
04	P30	Is Dutch Auction Suitable for Decomposable Tasks in Competitive Crowdsourcing Markets? [87].
05	P25	Efficient Task Decomposition for Sequential Crowdsourced Task Solving [43].
06	P31	SLADE: A Smart Large-Scale Task Decomposer in Crowdsourcing [65].
07	P32	Task Allocation in Spatial Crowdsourcing: Current State and Future Directions [88].
08	P33	Two-sided Online Micro-Task Assignment in Spatial Crowdsourcing [89].
09	P23	A Task Decomposition Framework for Surveying the Crowd Contextual Insights [63].
10	P34	Batch Allocation for Tasks with Overlapping Skill Requirements in Crowdsourcing [90].
11	P35	Crowdsourcing Software Task Assignment Method for Collaborative Development [91].
12	P36	Feedback Based High-Quality Task Assignment in Collaborative Crowdsourcing [92].
13	P37	Dynamic Worker-and-Task Assignment on Uncertain Spatial Crowdsourcing [93].
14	P38	Task Assignments in Complex Collaborative Crowdsourcing [94].
15	P39	A workload-dependent task assignment policy for crowdsourcing [95].
16	P40	Task Assignment in Spatial Crowdsourcing: Challenges and Approaches [96].
17	P41	Predictive Task Assignment in Spatial Crowdsourcing: A Data-driven Approach [97].
18	P42	A network-based mechanism for managing decomposable tasks via crowdsourcing [98].

19	P43	Personalized and Diverse Task Composition in Crowdsourcing [99].
20	P44	Prediction-Based Task Assignment in Spatial Crowdsourcing (Technical Report) [100].
21	P07	A Profile-Aware Microtasking Approach for Improving Task Assignment in Crowdsourcing Services [6].
22	P45	Age-Based Task Specialization for Crowdsourced Proofreading [42].
23	P22	Efficient Task Decomposition in Crowdsourcing [41].
24	P46	Designing Complex Crowdsourcing Applications Covering Multiple Platforms and Tasks [64].
25	P47	Can Microtask Programming Work in Industry? [54].
26	P48	A Brief Perspective on Microtask Crowdsourcing Workflows for Interface Design [31].
27	P49	TurKit: Tools for Iterative Tasks on Mechanical Turk [101].
28	P50	TurKit: Human Computation Algorithms on Mechanical Turk [102].
29	P51	CrowdWeaver: Visually Managing Complex Crowd Work [103].
30	P52	PlateMate: Crowdsourcing Nutrition Analysis from Food Photographs [104].
31	P53	CrowdForge: Crowdsourcing Complex Work [105].
32	P54	Collaboratively Crowdsourcing Workflows with Turkomatic [106].
33	P55	Microtask Programming: Building Software with a Crowd [107].
34	P56	Soylent: A Word Processor with a Crowd Inside [108].

The methods and approaches which are used to decompose the task in the context of crowdsourced software development are identified from the selected papers (shown in Table 4.7). Data extracted from each paper is maintained and explained in the following section.

4.3.2 Identified Task Decomposition Methods

The purpose of second research question of this study was to identify existing methods, approaches, workflows, models or mechanisms which are used to decompose the complex task

into the number of simple and short tasks i.e., microtasks, in the context of crowdsourced software development.

S. Saito et.al [54] developed the web application system which follows the manual decomposition of complex tasks into the number of microtasks. They utilized the slack time of the employees of the company to develop that project. Depending on the expertise, a software engineer manually decomposed the tasks into microtasks. Technical tasks are performed by software engineer and non-technical (simple) tasks are performed by the crowd-workers of the company. This decomposition method addressed the tasks which are related to design and development.

Aniket Kittur et.al [105] developed the web-based prototype named CrowdForge which decomposes the article writing and decision making related tasks into the microtasks. For article writing, the approach used in the prototype decomposes the task into the number of microtasks i.e., make outline of the given topic, collect the information for each section, write content for each section, proofreading the content, combine the content and restructure the content into the required format.

M. S. Bernstein et.al [108] developed the web-based word processing interface named Soylent which decomposes the complex tasks related to proof reading, shortening of text and natural language crowd scripting. It introduces the Find, Fix, Verify crowd programming pattern to proofread the documents. For the accomplishment of each microtask i.e., finding the errors, fixing those errors and verification; multiple crowd workers can be hired to get the authentic results. Soylent also applies multiple iterations for task decomposition as well as their execution, in order to get reasonable outcome.

Greg Little et.al [101], [102] developed TurKit, a toolkit to deploy human computational iterative tasks to Mechanical Turk. It provides the iterative text improvement facilities including brainstorming, image tagging, and image description, feedback of any content and addition of annotations by decomposition of the specific task into the number of microtasks. As recognizing bad handwriting is difficult task for computers, Turkit provides facility to decipher bad handwriting by distributing scanned image of the piece of text to the multiple turkers (crowd workers of MTurk) and iterate the process until required results achieved.

Jon Noronha et.al [104] developed a web-based prototype named PlateMate; that allows users to upload photos of their meals and receives estimate of food intake (in calories) and food composition. It crowdsources nutritional analysis from uploaded meal images using Amazon Mechanical Turk. As estimation of food intake and composition of food is complex task, PlateMate decomposes the complex task into the microtasks i.e., identify (identification of food elements in each image), estimate (portion of food) and measure (calculate nutritional data).

Huan Jiang et.al [41], [43] presented vertical task decomposition method which is related to dependent subtasks. It is used to proofread the documents and utilizes the Find, Fix, Verify crowd programming pattern to proofread the document. Suppose a document has three paragraphs, each paragraph has to be assigned to a crowd worker and he then applies Find Fix, Verify strategy to proofread the paragraph. In vertical task decomposition, the microtasks ('fix' and 'verify') are dependent on the previous microtask i.e., 'find'. Crowd worker must find the errors from the paragraph before fixing them and then verification process performs.

Huan Jiang et. al [43], [41] presented horizontal task decomposition method which is related to independent subtasks. It is used to proofread the documents and utilizes the Find, Fix, Verify crowd programming pattern to proofread the document. Suppose a document has to be proofread, each microtask (find, fix and verify) is assigned to a crowd worker. First crowd worker finds the errors (spelling, grammatical, style) from the document, second crowd worker fix that identified errors and third worker verifies the changes made to the document.

Aniket Kittur et. al [103] presented a web-based system named CrowdWeaver that visually creates and manages crowd workflows. The system basically allows the users to manually split the tasks, system monitors their workflow and displays error messages to manage them. Anand Kulkarni [106] developed web-based tool named Turkomatic which helps the requesters in solving complex tasks by recruiting the crowd workers. It allows the registered crowd workers to manually decompose the complex tasks into the shorter and simpler ones. During the decomposition of complex tasks, Turkomatic monitors the status of worker's designed workflows in real time.

Thomas D. LaToza et. al [107] developed an approach which decomposes programming related complex tasks. The approach uses the database in which possible programming related

microtasks saved. Furthermore, when one microtask has performed its functionality, a list of possible subsequent microtasks is shown to select the next task to be performed. Mengyao et. al [31] presented the approach which aids to decompose the interface design related complex tasks into microtasks. The approach compares the microtasks decomposed by task owners and task workers; and concluded that the task decomposed by task workers are more successful and gives quality outcome in less resources.

It has been observed that 11 task decomposition methods and workflows were found from the selected studies. As this SLR was conducted by individual researcher, there were more chances of validity threats in the context of finding the task decomposition methods. Due to the consequences of individual's judgement and bias, researcher planned to distribute the remaining papers to different researchers. In order to do so, initially issue was familiarized to 5 researchers and 23 papers were distributed among them. They were asked to find the task decomposition methods from the given papers.

It was an iterative process and took almost one month to review the papers. Researchers discarded 19 papers and claimed that generally these papers are related to the task assignment and task allocation in crowdsourced software development. Few papers discussed the optimal solutions and developed the model for better completion of decomposable tasks in crowdsourcing. According to the reviewers, 4 papers were related to the task decomposition methods which are added in the study and compared with other existing decomposition methods.

4.3.3 Comparative Review of Identified Task Decomposition Methods

In order to check the suitability of identified task decomposition methods on the basis of technical and non-technical tasks, a comparative review of existing decomposition methods is conducted. Table 4.8 shows the details of comparative review.

Table 4.8: Comparison of identified decomposition methods for technical tasks

S. No	Decomposition Method	Technical tasks	Non-technical tasks
1	S. Saito et.al [54]	Supported	Partially supported
2	Aniket et.al [105]	Not supported	Partially supported
3	Jon Noronha et.al [104]	Not supported	Partially supported
4	M. S. Bernstein et.al [108]	Not supported	Supported
5	Greg Little et.al [101], [102]	Not supported	Partially supported
6	Huan Jiang et.al [41], [43]	Not supported	Partially supported
7	Yongxin et. al [65]	Partially supported	Not supported
8	Huan Jiang et. al [43], [41]	Not supported	Partially supported
9	Kittur et. al [103]	Not supported	Supported
10	Marco et. al [64]	Partially supported	Partially supported
11	Anand Kulkarni [106]	Not supported	Partially Supported
12	D. LaToza et. al [107]	Supported	Not supported
13	Mengyao et. al [31]	Partially supported	Not supported
14	Saeed et. al [63]	Not supported	Partially supported

It can be interpreted from Table 4.8 that only 2 existing task decomposition methods can decompose technical tasks. Amongst them, 1 method supports the manual decomposition of tasks. 3 methods can decompose few of the tasks which are technical in nature. On contrary, 2 methods support the decomposition of non-technical tasks and few of the methods i.e., 9 approaches can decompose few of the tasks which are non-technical in nature.

In order to check the suitability of existing task decomposition methods for every microtasking category in the following section, unique IDs are given to the microtasking categories which are shown in Table 4.9.

Table 4.9: Unique IDs for microtasking categories

S. No	Microtasking Category	Microtasking category ID
1.	Information finding	C1
2.	Interpretation & Analysis	C2
3.	Data transcription & translation	C3
4.	Surveys	C4
5.	Identification	C5
6.	Content creation	C6
7.	Content Access	C7
8.	Designing	C8
9.	Development	C9
10.	Verification & validation	C10
11.	Quality assessment & Testing	C11

As described in Section 4.3.2, existing task decomposition approaches are suitable for specific niches e.g., vertical task decomposition method provides the solution for proofreading, utilizes the find-fix-verify task decomposition pattern. Table 4.10 shows the comparison which depicts if each existing task decomposition method supports the decomposition of 11 identified microtasking categories.

Table 4.10: Comparison of decomposition methods for microtasking categories

S. No	Existing Task Decomposition Methods	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
1.	S. Saito [54]	×	×	×	×	×	×	×	✓	✓	×	✓
2.	Kittur [105]	✓	×	×	×	×	✓	×	×	×	×	×
3.	Bernstein [108]	✓	×	×	×	×	✓	×	×	×	×	×

4.	Greg [101], [102]	×	×	×	✓	×	✓	×	×	×	×	×
5.	Noronha [104]	✓	✓	×	×	✓	×	×	×	×	✓	×
6.	Huan [41], [43]	×	×	×	×	✓	×	×	×	×	✓	✓
7.	Huan [41], [43]	×	×	×	×	✓	×	×	×	×	✓	✓
8.	Yongxin [65]	×	×	×	×	✓	×	×	×	✓	×	✓
9.	Kittur [103]	×	×	×	×	✓	×	×	✓	×	×	×
10.	Marco [64]	×	×	×	×	×	×	×	×	✓	✓	×
11.	Kulkarni [106]	✓	×	×	×	✓	✓	×	×	×	×	×
12.	D. LaToza [107]	×	×	×	×	✓	×	×	×	✓	×	✓
13.	Mengyao [31]	×	×	×	×	×	×	×	✓	×	×	×
14.	Saeed [63]	×	×	×	✓	×	×	×	×	×	×	×

It can be interpreted from the Table 4.10 that most of the existing task decomposition approaches supports the decomposition of ‘Verification’ and ‘Content creation’ related complex tasks. It can also be interpreted that no existing task decomposition approach facilitates the decomposition of complex tasks related to ‘Data transcription and translation’ and ‘Content access’. Furthermore, no existing task decomposition approach can decompose all types of tasks which are shown in Table 4.9.

4.4 Summary

Chapter 4 covers the overview of the studies which were found to identify the microtasking activities and task decomposition methods, their quality assessment details, details of unique IDs given to the studies, data extracted from the studies and validation of naming conventions of identified microtasking activities and their positioning through experts’ opinion. In the last section, comparison of identified task decomposition methods is presented.

CHAPTER 5

PROPOSED MODEL

5.1 Overview

In Chapter 4, existing task decomposition methods are identified and comparative review was conducted to check if existing methods can decompose technical, non-technical and hybrid tasks. In this chapter, Hybrid model is proposed which can decompose technical, non-technical, dependent, and independent and hybrid tasks. Furthermore, four phases of the model are explained in detail with examples.

5.2 Discussion

Microtasking activities are identified through Systematic Literature Review in this study. Relevant identified microtasking activities are then combined into different categories e.g., information finding, interpretation and analysis etc. which are comprehensively explained for all the activities in Chapter 4, Section 4.2.3. Table 5.1 shows the grouping of relevant microtasking categories suggested by experts and unique IDs of task groups.

Table 5.1: Grouping of validated microtasking categories

Group ID	**Task Groups**	**Categories**
G-1	Information	Information finding
		Interpretation & Analysis
G-2	Data transcription	Data transcription & translation
G-3	Survey	Surveys
		Identification
G-4	Content	Content creation
		Content Access

G-5	Code implementation	Designing
		Development
		Verification & validation
		Quality assessment & Testing

Table 5.1 shows the detail of microtasking categories along with their task groups. It consists of three columns; first column shows the unique IDs which are given to every task group. Second column of the table shows the names of task groups which are suggested by experts as explained in Chapter 4 Section 4.2.4. Third column of the table shows the microtasking categories which are explained in Chapter 4 Section 4.2.3 and validated by the experts.

On the basis of tasks performed by each category, nature of the groups is identified. According to the task execution pattern and task prerequisites, nature of the task groups are identified which are shown in Table 5.2.

Table 5.2: Nature of the 'task groups' on the basis of dependent, independent and hybrid tasks

Task Groups	**Nature of the Task Group**
Information	Dependent task group
Data transcription	Independent task group
Survey	
Content	Hybrid task group
Code implementation	

Table 5.2 shows the nature of the tasks of the relevant groups. Literature has revealed that information related tasks (G-1) depends on the other tasks, hence they have to be performed sequentially [43]. Identification of information of the particular entity depends on the other tasks e.g., 'Find the reviews of client against the particular product'. Accomplishment of this task depends on the fact that if the particular client has used that product or not. As the

identification of information depends on other tasks, hence it lies under the category of dependent task group.

On the other hand, ‘Data transcription’ (G-2) and ‘Survey’ (G-3) are categorized as independent task groups. Data transcription can be decomposed into numerous tasks e.g., ‘Identify the file format’, ‘Select the language’ and ‘Translate the given paragraph in Chinese’ and ‘Convert the given image file into (.png) format’ [72]. These tasks are independent in nature, hence ‘Data transcription’ lies under the category of independent task group.

Surveys are used to get the feedback of users against particular entity, product or service. They are also used for sentiment analysis and to get opinion of users for any product, day or service e.g., ‘What do you think about interface of Mac OS X?’ and ‘Choose a gift which you want to give you mother on this Mother’s Day’. As these microtasks do not require any pre-requisites, hence ‘Surveys’ (G-3) are also categorized as independent task group.

‘Content’ group (G-4) is the combination of two categories i.e., content access and content creation. For *Content Access,* users directly access the required content without depending on any other task e.g., ‘Click on the given link to watch the video’. Hence, accessing the content does not require any other task to be performed first. So, *Content Access* is independent type of tasks. On the contrary, creation of content requires the preliminary steps for its completion.

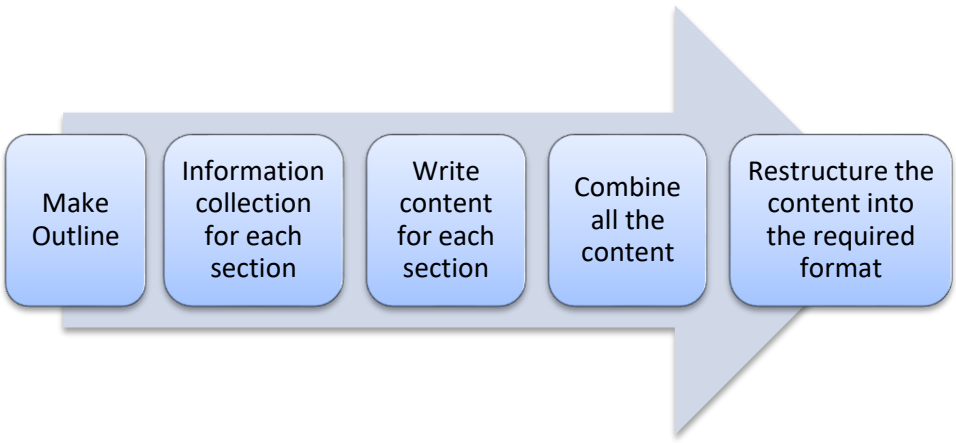

Figure 5.1: Overview of steps to create content [108]

Figure 5.1 shows the steps for the creation of content. Initially, given topic is divided into different sections (headings) to make the outline, then information is gathered for each section. After the collection of relevant information, it is then organized in each section, then content is restructured into the given format. For the creation of content, each step requires the completion of previous step. Hence, content creation lies under the category of dependent task.

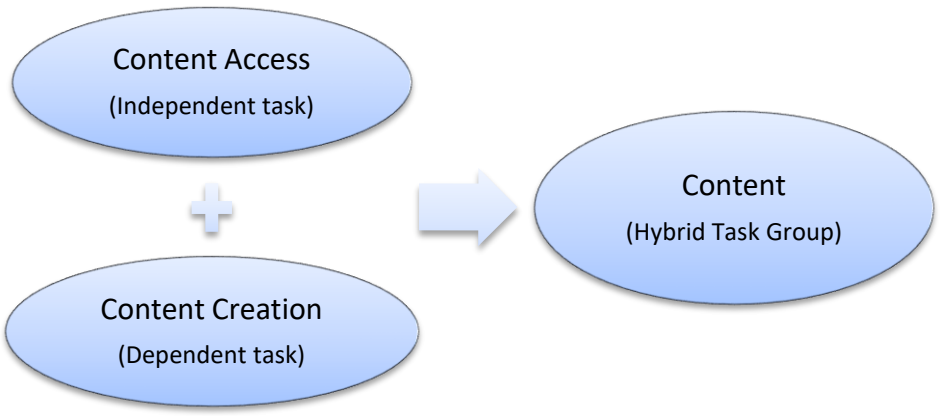

Figure 5.2: Formation of Hybrid task group

Figure 5.2 shows that the 'Content' group (G-4) is the combination of two categories i.e., *Content Access* and *Content creation*. Tasks related to Content access are independent in nature and those tasks which are related to content creation, are dependent in nature. Hence, 'Content' group is named as Hybrid task group.

'Code implementation' group (G-5) is the combination of various categories i.e., design and development, verification and validation, Quality assurance and testing. Tasks related to few categories are dependent in nature i.e., design and development related task needs the pre-requisites for their completion e.g., 'Write the piece of code which calls the function given in the mentioned file'. This task requires the worker to identify the function from the file then write the piece of code according to the given instruction and output. Tasks related to other categories i.e., verification and validation, quality assurance and testing may or may not depends on any other task(s). Therefore, 'Code implementation' group (G-5) is also considered as Hybrid task group.

5.3 Proposed Model

As described in Chapter 4, Section 4.3.3, no existing task decomposition model supports the decomposition of technical and non-technical tasks at the same time. Furthermore, no existing methods can decompose all types' pf tasks which are shown in Chapter 4, Table 4.9. In order to decompose technical, non-technical, hybrid, dependent and independent tasks, Hybrid Model is proposed which can be used in crowdsourced software development. Figure 5.3 shows the Hybrid Model for task decomposition.

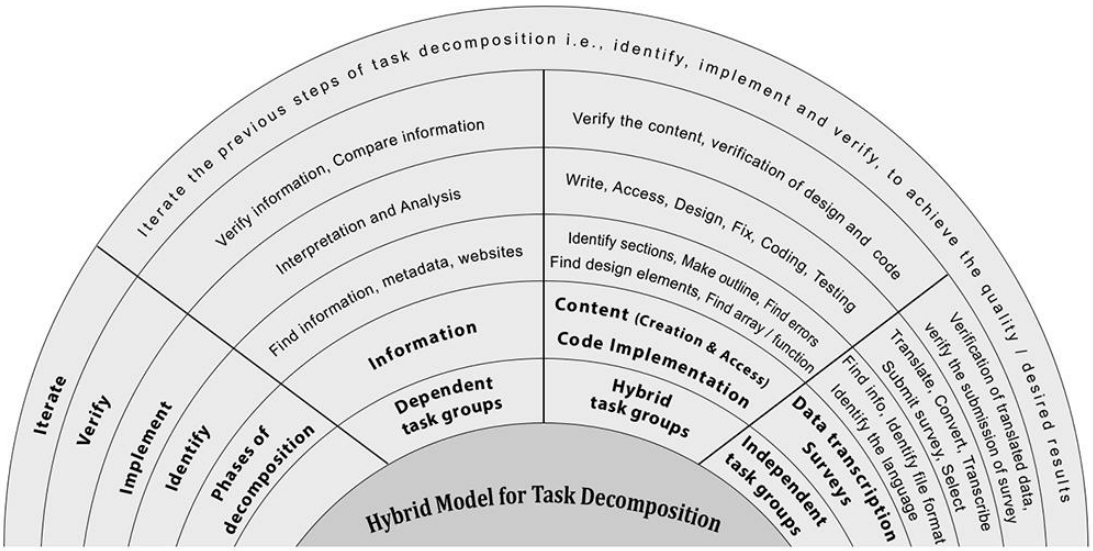

Figure 5.3: Proposed Hybrid model for task decomposition

Figure 5.3 shows the Hybrid model for the decomposition of complex task into the microtasks. First layer of the model elaborates that this model is suitable for the decomposition of all types of complex tasks i.e., dependent, independent and hybrid tasks. In the second layer of the model, different task groups (G-1, G-2, G-3, G-4, G-5) are placed against the nature of the task group. Left side of the model highlights the phases of the task decomposition. From the nature of microtasking categories, it has been noticed that decomposition of any task involves few major steps which are the phases of proposed task decomposition model. Generic names are given to phases, shown in Table 5.3.

Table 5.3: Phases of the Hybrid model

Phases of Task Decomposition	**Phase ID**
Identify	PH-1
Implement	PH-2
Verify	PH-3
Iterate	PH-4

Table 5.3 shows the task decomposition phases with their unique IDs. Generic names are given to all phases of the decomposition so that every type of the complex task will fit into the model for their decomposition.

5.3.1 First phase (PH-1) of the proposed model

This phase of the task decomposition is related to the identification of the microtasks against each group. For the 'Information' group, crowd workers have to identify the information about any product, incident, and meta-data of the article, any service, interpreted data, entity, analysis or results of the specific data. For the 'Content' group, crowd workers have to identify the sections of the document, headings and subheadings under each section. For the 'Code implementation' group, microtasks related to *Identification* (Phase) can be identify design elements, find error, find test cases, identify test suite, find array or function and line of the code in which particular function is called.

As 'Data transcription' group is related to the translation and transcription of the data from one language or one format to the other. Hence, microtasks under this phase (PH-1) can be identification of language for translation of the given data, identify the file format to which client wants to convert the file. Similarly, for the 'Survey' group, microtasks can be 'find information' and 'identification of user's perspective and feedback'.

Let’s take an example that user needs to translate English sentence into different languages. As the first phase of Hybrid model for task decomposition is ‘Identify’. For the particular example, user needs to identify the language in which he/she requires the translation of English sentence.

5.3.2 Second phase (PH-2) of the proposed model

This phase plays the vital role in the accomplishment of the task, as it is primary part of the decomposition. It is related to the *implementation*, which purely varies for each group. For the ‘Information’ group, crowd workers have to interpret and analyze the given information against the given criteria. For the ‘Content’ group, this phase elaborates the implementation for the creation of content. Microtasks for this (G-4) group can be writing of content under each section, access the content, click on the given link, organize the content and restructure the content according to the given format.

‘Code implementation’ group is related to the technical tasks i.e., designing and development of any software, quality assurance and testing of software project and their verification and validation. For this phase (PH-2), *implementation* related microtasks can be writing the piece of code, implement the function, designing the layout of software, designing the User Interface (UI) of website, writing the test cases and testing of given code.

As ‘Data transcription’ group is related to the translation and conversion of file formats. For this phase (PH-2), *implementation* related tasks can be translate the given paragraph into Urdu, transcribe the given video file into audio file and convert the given image (.ai) file into (.jpeg) file. For ‘Survey’ group, *implementation* related microtasks can be selection of the options and submission of the survey.

For the particular example explained in the first phase of the model i.e., translation of English sentence into different languages. As the second phase of the model is ‘Implementation’, hence it decomposes the task into ‘translate the English sentence into the language which user selected in first phase’.

5.3.3 Third phase (PH-3) of the proposed model

This phase is related to the verification of the microtasks performed in the previous phase (PH-2). For the 'Information' group, it compares and verifies the identified information, interpreted and analyzed data. For hybrid task groups, it verifies if developed content is according to the client's requirement and format. Moreover, microtasks related to the verification of design, code and developed test cases are also performed in this phase. For independent task groups, it verifies if the survey is submitted successfully, if the file is converted into the correct file format and whether the translated content is conveying the correct and sense.

In the third phase of the model, for the particular example explained in above sections, this phase will verify the translated sentence. It will check the user's input i.e., whether the translated sentence is in the same language which was required by the user. It will also verify if the translated sentence is conveying the same meanings and sense of the original sentence of English language.

5.3.4 Fourth phase (PH-4) of the proposed model

In order to achieve the quality results against each group, iterations of the previous phases (PH-1, PH-2, PH-3) are to be performed. If the output of third phase (PH-3) is not satisfactory, then fourth phase (PH-4) i.e., 'Iterate' should be performed to achieve the fruitful results. It is possible that multiple iterations are performed for a specific task. For iterations, tasks can be assigned to different crowd workers to achieve the authentic and quality results.

For the particular example explained in the above sections of the model i.e., translation of English sentence into different languages. This phase will iterate the previous steps to get the quality results. It will analyze if the output of verification phase is not satisfactory, then iterations of the previous steps will be performed until the stopping condition met. If the translated sentence is not accurate, then iterations will be performed and previous microtask

i.e., implement (translate the sentence into mentioned language) will again assigned to the crowd worker(s).

The proposed model is different from the existing task decomposition methods as it is an iterative method to decompose the complex tasks in crowdsourced software development. It gives leverage to gain the satisfactory output i.e., if the implementation of the task is not giving the required results, 'iteration' phase of the model helps to get the required results. Furthermore, it supports the decomposition of technical, non-technical, dependent, and independent and hybrid tasks.

5.4 Comparison of proposed model with existing task decomposition methods

As described in Chapter 4 Section 4.3.3, most of the existing task decomposition methods support the decomposition of tasks related to 'content creation' and 'verification'. Comparison of proposed hybrid task decomposition model and existing methods is tabulated in Table 5.4.

Table 5.4: Comparison between proposed model and existing task decomposition methods

S. No	Existing Task Decomposition Methods/Authors	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
1.	S. Saito [54]	No	No	No	No	No	No	No	Yes	Yes	No	Yes
2.	Kittur [105]	Yes	No	No	No	No	Yes	No	No	No	No	No
3.	Bernstein [108]	Yes	No	No	No	No	Yes	No	No	No	No	No
4.	Greg [101], [102]	No	No	No	Yes	No	Yes	No	No	No	No	No
5.	Noronha [104]	Yes	Yes	No	No	Yes	No	No	No	No	Yes	No
6.	Huan [41], [43]	No	No	No	No	Yes	No	No	No	No	Yes	Yes

7.	Huan [41], [43]	No	No	No	No	Yes	No	No	No	No	Yes	Yes
8.	Yongxin [65]	No	No	No	No	Yes	No	No	No	Yes	No	Yes
9.	Kittur [103]	No	No	No	No	Yes	No	No	Yes	No	No	No
10.	Marco [64]	No	No	No	No	No	No	No	No	Yes	Yes	No
11.	Kulkarni [106]	Yes	No	No	No	Yes	Yes	No	No	No	No	No
12.	D. LaToza [107]	No	No	No	No	Yes	No	No	Yes	Yes	No	Yes
13.	Mengyao [31]	No	No	No	No	No	No	No	Yes	No	No	No
14.	Saeed [63]	No	No	No	Yes	No	No	No	No	No	No	No
15.	Hybrid Model for task decomposition	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

It can be interpreted from the Table 5.4 that no existing task decomposition methods can decompose all types of tasks which are mentioned in table 5.1. The proposed model is generic in nature and can decompose all types of tasks i.e., technical, non-technical, dependent, independent and hybrid tasks. It can decompose such tasks which are related to information finding, interpretation and analysis, data transcription and translation, survey, identification, content creation, designing and development, verification and validation, quality assessment and testing.

5.5 Summary

In chapter 5, Hybrid Model is proposed which gives leverage to decompose all types of tasks i.e., technical, non-technical, dependent, independent and hybrid tasks. Different layers and phases of the model are explained in detail with the help of examples.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Overview

This chapter presents the conclusion and contributions of the study. Some potential areas are suggested where future studies can be conducted for the development of automated task decomposition system.

6.2 Conclusion

Software crowdsourcing has gained more value and place in software development industry, in recent years. It is a viable distributed problem-solving approach that utilizes the mutual efforts of online crowd workers. It facilitates the software engineering process by integrating the various tasks (requirement elicitation, prototyping, designing, coding, testing etc.) performed by globally hired crowd workers. Four types of crowdsourcing models exist which includes peer production, investment model, competitions and microtasking model.

Microtasking is the model of software crowdsourcing in which micro-workers from different economies participate in the completion of digital tasks. It has beenobserved that it is difficult for crowdsourcing platform to define the criteria for decomposition of tasks into microtasks. At this moment in time, the challenge is to develop a mechanism for task decomposition for crowdsourced software development.

In this study, researcher presented the first step in the form of identification of microtasking activities which exist in crowdsourced software development. Systematic Literature Review has been conducted to identify the microtasks. 26 research papers are

reviewed in SLR and a total of 72 microtasking activities were identified from SLR. The relevant microtasks are then combined into different categories and meaningful names are given to microtasking categories. Expert review has been conducted to validate the naming conventions and duplication of identified microtasking activities. Furthermore, experts reviewed the microtasking categories and the positioning of microtasking activities in them. Experts suggested to further group the relevant microtasking categories; generic and meaningful names are given to the major groups. Changes recommended by experts, are applied and researcher came up with a list of 61 validated microtasking activities along with 11 microtasking categories and 5 major groups.

In the second step of the study, Systematic Literature Review has been conducted to identify the task decomposition methods which are used in crowdsourced software development. 34 research papers are reviewed to find the existing task decomposition methods and 14 methods were found. Comparative review was conducted to compare the features of existing task decomposition methods. Identified methods were compared to check the suitability that if they support the decomposition of technical, non-technical and hybrid tasks. Findings of the comparative review revealed that most of the existing task decomposition approaches supports the decomposition of few of the tasks which are either technical or non-technical in nature. Moreover, no existing task decomposition method can decompose all types of tasks i.e., dependent, independent and hybrid tasks.

In the third step of the study, Hybrid model is proposed for the decomposition of dependent, independent and hybrid tasks which exist in crowdsourced software development. There are four phases of the proposed model which decomposes the complex task. The proposed model supports the decomposition of technical, non-technical, dependent, independent and hybrid tasks. It is an iterative approach to decompose the complex tasks in crowdsourced software development. It gives leverage to gain the satisfactory output i.e., if the implementation of the task is not giving the required results, ‘Iteration’ phase of the model helps to get the required results.

6.3 Limitations and Future recommendations

The study has few limitations which are considered as areas for future recommendations. The limitations and future work are discussed below.

- Four databases (IEEE Xplore, ACM, Springer Link and Science Direct) are explored to identify the task decomposition methods and microtasking activities which exist in crowdsourced software development. Still there is a variety of other databases available, which can be explored in future studies to find microtasking activities and task decomposition approaches.
- Microtasking activities and their categories have been evaluated by four industrial and academic experts. In future, identified microtasking activities and categories can be evaluated by more experts in order to get more validated results.
- The proposed Hybrid model can be validated from software industry in future. Although, experts validated the task groups which are used in proposed model, still there is need to evaluate it by software industrialists to use it in more generic way.
- On the basis of proposed model, an algorithm and then web-based system can be developed for automatic task decomposition. In order to do so, for every task, a list of possible microtasks and alternative ways of decomposition can be placed in the database and system generates the microtasks according to the nature of the given task.
- Experiments can be conducted on different crowdsourcing and microtasking platforms by using the proposed model as task decomposition mechanism.
- Experiments can be conducted to check the efficiency and effectiveness of the proposed task decomposition model.

6.4 Summary

This chapter briefly concluded the research and presented the contributions of the study. Limitations and future directions are drawn which will help the future studies for possible improvements.

REFERENCES

[1] T. Hasanov, “Microcredit Risk Assessment using Crowdsourcing and Social Networks,” *15th IEEE/ACIS Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distributed Comput.*, pp. 313–322, 2014.

[2] M. N. Malik and H. H. Khan, “Investigating Software Standards: A Lens of Sustainability for Software Crowdsourcing,” *IEEE Access*, vol. 6, pp. 5139–5150, 2018.

[3] S. Board, C. Science, and S. Lagerl, “Crowdsourcing software development : problems experienced from the developers ’ perspective,” *Master Thesis*, p. 30, 2017.

[4] A. Sari and G. I. AlptekİN, “An Overview of Crowdsourcing Concepts in Software Engineering,” *Int. Jorunal Comput.*, vol. 2, no. 1, pp. 106–114, 2017.

[5] T. D. Latoza and A. Van Der Hoek, “Crowdsourcing in Software Engineering : Models , Opportunities , and Challenges,” *IEEE Softw.*, vol. 33, no. 1, pp. 1–13, 2016.

[6] S. Africa and S. Africa, “A Profile-Aware Microtasking Approach for Improving Task Assignment in Crowdsourcing Services,” *IST-Africa Week Conf.*, pp. 1–10, 2016.

[7] T. D. Latoza, “Crowdsourcing in Software Engineering: Models, Motivations and Challenges,” *IEEE Softw.*, vol. 3, pp. 71–77, 2016.

[8] A. Kittur, E. H. Chi, and B. Suh, “Crowdsourcing user studies with Mechanical Turk,” *Conf. Hum. Factors Comput. Syst. - Proc.*, pp. 453–456, 2008.

[9] A. Tomczak and A. Brem, “A conceptualized investment model of crowdfunding,” *Ventur. Cap.*, vol. 15, no. 4, pp. 335–359, 2013.

[10] D. H. Kim, A. Lavoie, G. Ratté, C. Beaumont, L. Germain, and D. Larouche, “The Future of Crowd Work,” *IEEE Internet Comput.*, vol. 41, no. 4, pp. 466–472, 2015.

[11] E. Aghayi, T. D. Latoza, P. Surendra, and S. Abolghasemi, “Implementing Microservices through Microtasks,” *Int. Jorunal Comput.*, vol. 32, pp. 1–12, 2019.

[12] A. M. Poblet and M. Fitzpatrick, “Microtasking : redefining crowdsourcing practices in emergency management,” *Aust. J. Emerg. Manag.*, vol. 32, no. 2, pp. 47–53, 2017.

[13] M. Storey, A. Zagalsky, F. F. Filho, L. Singer, and D. M. German, “How Social and Communication Channels Shape and Challenge a Participatory Culture in Software Development,” *IEEE Xplore*, vol. 43, no. 2, pp. 185–204, 2017.

[14] T. D. Latoza, A. Di Lecce, F. Ricci, W. Ben Towne, and A. Van, “Microtask Programming,” *IEEE Trans. Softw. Eng.*, vol. 45, pp. 1106–1124, 2019.

[15] Y. Chen, S. W. Lee, Y. Xie, Y. W. Yang, W. S. Lasecki, and S. Oney, “Codeon: On-

demand software development assistance," *Conf. Hum. Factors Comput. Syst. - Proc.*, vol. 2017-May, pp. 6220–6231, 2017.

[16] T. D. Latoza, A. Di Lecce, F. Ricci, W. Ben Towne, and A. Van Der Hoek, "Ask the crowd: Scaffolding coordination and knowledge sharing in microtask programming," *Proc. IEEE Symp. Vis. Lang. Human-Centric Comput. VL/HCC*, vol. 2015-Decem, pp. 23–27, 2015.

[17] Y. Gao, S. Member, Y. Chen, S. Member, and K. J. R. Liu, "On Cost-Effective Incentive Mechanisms in Microtask Crowdsourcing," *IEEE Trans. Comput. Intell. AI GAMES*, vol. 7, no. 1, pp. 3–15, 2015.

[18] P. Hirth, Matthias & Hossfeld, Tobias & Tran-Gia, "Human Cloud as Emerging Internet Application - Anatomy of the Microworkers Crowdsourcing Platform," *IEEE Internet Comput.*, vol. 5, pp. 1–22, 2011.

[19] Z. Wang, H. Sun, Y. Fu, and L. Ye, "Recommending Crowdsourced Software Developers in Consideration of Skill Improvement," *2017 32nd IEEE/ACM Int. Conf. Autom. Softw. Eng.*, pp. 717–722, 2017.

[20] C. López, "An Experiment in Microtask Crowdsourcing Software Design," *Master Thesis*, p. 120, 2015.

[21] S. H. B, V. Siegert, and M. Gaedke, "Crowdsourced Reverse Engineering : Experiences in Applying Crowdsourcing to Concept Assignment," *Eval. Nov. Approaches to Softw. Eng.*, pp. 215–239, 2019.

[22] J. D. Herbsleb, A. Mockus, T. A. Finholt, A. Arbor, and R. E. Grinter, "An Empirical Study of Global Software Development : Distance and Speed," *Proc. 23rd Int. Conf. Softw. Eng.*, 2001.

[23] B. J. Howe, "The Rise of Crowdsourcing," *Wired*, vol. 14, pp. 1–5, 2006.

[24] R. R. Morris, M. Dontcheva, and E. M. Gerber, "Priming for better performance in microtask crowdsourcing environments," *IEEE Internet Comput.*, vol. 16, no. 5, pp. 13–19, 2012.

[25] F. R. Vianna, J. Peinado, and A. R. Graeml, "Crowdsourcing platforms: Objective, activities and motivation," *25th Am. Conf. Inf. Syst. AMCIS 2019*, pp. 1–10, 2019.

[26] E. R. Q. Weidema, C. López, S. Nayebaziz, F. Spanghero, and A. Van Der Hoek, "Toward Microtask crowdsourcing software design work," *Proc. - 3rd Int. Work. CrowdSourcing Softw. Eng. CSI-SE 2016*, pp. 41–44, 2016.

[27] "BestGlobalBrands." [Online]. Available: https://www.interbrand.com/best-brands/. [Accessed: 11-Jul-2021].

[28] “Planbox.” [Online]. Available: https://www.planbox.com/4-companies-that-are-killing-it-with-crowdsourcing/. [Accessed: 25-Apr-2021].

[29] “cadcrowd,” 2021. [Online]. Available: https://www.interbrand.com/best-brands/. [Accessed: 29-Apr-2021].

[30] K. Mao, L. Capra, M. Harman, and Y. Jia, “A survey of the use of crowdsourcing in software engineering,” *J. Syst. Softw.*, vol. 126, pp. 57–84, 2017.

[31] M. Zhao and A. Van Der Hoek, “A Brief Perspective on Microtask Crowdsourcing Workflows for Interface Design,” *IEEE Xplore*, pp. 45–46, 2015.

[32] P. Kucherbaev, “Quality Assurance Strategies in Microtask Crowdsourcing,” *PhD Thesis*, 2016.

[33] F. Developers, “fiverr tasks,” 2020. [Online]. Available: https://www.fiverr.com/gigs/tasks. [Accessed: 05-Oct-2020].

[34] “freelancer,” 2020. [Online]. Available: https://www.freelancer.com/about. [Accessed: 15-Oct-2020].

[35] D. A. G. and E. L. Karim R. Lakhani, “TopCoder (A): Developing Software through Crowdsourcing.,” *Harvard Bus. Sch. Case 610-032*, p. 20.

[36] R. L. Saremi, Y. Yang, G. Ruhe, and D. Messinger, “Leveraging crowdsourcing for team elasticity: An empirical evaluation at topcoder,” *Proc. - 2017 IEEE/ACM 39th Int. Conf. Softw. Eng. Softw. Eng. Pract. Track, ICSE-SEIP 2017*, pp. 103–112, 2017.

[37] A. L. Zanatta, L. S. Machado, G. B. Pereira, R. Prikladnicki, and E. Carmel, “Software Crowdsourcing Platforms,” *IEEE Softw.*, vol. 33, pp. 112–116, 2016.

[38] G. Kuti, Mónika ; Madarász, “Crowdfunding,” *Public Finan. Q.*, pp. 1–7, 2014.

[39] P. Belleflamme, T. Lambert, and A. Schwienbacher, “Individual crowdfunding practices,” *Ventur. Cap.*, vol. 15, no. 4, pp. 313–333, 2013.

[40] P. Belleflamme, N. Omrani, and M. Peitz, “The economics of crowdfunding platforms,” *Inf. Econ. Policy*, vol. 33, pp. 11–28, 2015.

[41] H. Jiang and S. Matsubara, “Efficient task decomposition in crowdsourcing,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8861, pp. 65–73, 2014.

[42] M. Kobayashi, T. Ishihara, and T. Itoko, “Age-Based Task Specialization for Crowdsourced,” *Proc. 7th Int. Conf. Univers. Access Human-Computer Interact. user Context Divers.*, vol. 2, pp. 104–112, 2013.

[43] J. Huan, Z. U. O. Min, and M. Shigeo, “Efficient Task Decomposition for Sequential Crowdsourced Task Solving,” *Chinese J. Electron.*, vol. 29, no. 3, pp. 468–475, 2020.

[44] K. Stol and T. D. Latoza, “Crowdsourcing for Software Engineering,” *IEEE Softw.*, vol. 34, no. 2, pp. 30–36, 2017.

[45] Y. Benkler, “Peer production , the commons , and the future of the firm,” *Harvard Libr. Sch. Commun.*, vol. 15, no. 2, pp. 264–274, 2016.

[46] D. Kreiss, M. Finn, F. Turner, and D. Kreiss, “The limits of Peer production,” *Sage*, pp. 243–259, 2011.

[47] K. Stol, B. Caglayan, and B. Fitzgerald, “Competition-Based Crowdsourcing Software Development : A Multi-Method Study from a Customer Perspective,” *IEEE Trans. Softw. Eng.*, vol. 5589, pp. 1–25, 2017.

[48] L. Vaz, I. Steinmacher, and S. Marczak, “An Empirical Study on Task Documentation in Software Crowdsourcing on TopCoder,” *Proc. - 2019 ACM/IEEE 14th Int. Conf. Glob. Softw. Eng. ICGSE 2019*, pp. 48–57, 2019.

[49] E. R. Q. Weidema, C. López, S. Nayebaziz, F. Spanghero, and A. Van Der Hoek, “Toward Microtask Crowdsourcing Software Design Work,” *2016 3rd Int. Work. Crowdsourcing Softw. Eng.*, pp. 41–44, 2016.

[50] D. Renard and J. G. Davis, “Social interdependence on crowdsourcing platforms,” *J. Bus. Res.*, vol. 103, no. August 2018, pp. 186–194, 2019.

[51] A. Muralidhar, “InvestTech and Crowdsourcing Best Ideas: Investment strategies for a low-yield environment,” *SSRN*, pp. 1–12, 2017.

[52] T. D. Latoza and A. Van Der Hoek, “A Vision of Crowd Development,” *IEEE?ACM 37th Int. Conf. Softw. Eng.*, pp. 563–566, 2015.

[53] J. Cheng, J. Teevan, S. T. Iqbal, and M. S. Bernstein, “Break It Down : A Comparison of Macro- and Microtasks,” *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst.*, pp. 4061–4064, 2015.

[54] S. Saito, Y. Iimura, E. Aghayi, and T. D. Latoza, “Can Microtask Programming Work in Industry ?,” *Proc. 28th ACM Jt. Meet. Eur. Softw. Eng. Conf. ans Symp. Found. Softw. Eng.*, pp. 1263–1273, 2020.

[55] T. D. L. Emad Aghayi, “Crowdsourced Microservices: Behavior-Driven development Applied to Microtask Programming,” *IEEE Trans. Softw. Eng.*, pp. 1–25, 2019.

[56] M. Poblet and M. Fitzpatrick, “Microtasking models and managerial challenges,” *Proc. CSCW 2016-Workshop Towar. a Typology Particip. Crowdwork*, pp. 65–72, 2016.

[57] M. Blumberg, “Patterns of Connection,” *Handb. Hum. Comput.*, pp. 5–12, 2013.

[58] D. Bollier, “Coming to Terms with Innovative Payment Systems, Digital Currencies and Online Labor Markets,” *Weight. Marketpl.*, pp. 1–68, 2014.

[59] B. Naderi *et al.*, “Development and Validation of Extrinsic Motivation Scale for Crowdsourcing Micro-task Platforms,” *Proc. 2014 Int. ACM Work. crowdsourcing Multimed.*, pp. 31–36, 2014.

[60] “Micro-jobs site.” [Online]. Available: https://moneymint.com/best-micro-jobs-sites/. [Accessed: 11-Jul-2021].

[61] B. Naderi, I. Wechsung, and S. Möller, “Effect of Being Observed on the Reliability of Responses in Crowdsourcing Micro-task Platforms,” *2015 Seventh Int. Work. Qual. Multimed. Exp.*, pp. 4–5, 2015.

[62] “Microtasking platforms.” [Online]. Available: https://vitaldollar.com/microtask-websites/. [Accessed: 21-Mar-2021].

[63] M. Allahbakhsh, S. Arbabi, and M. Shirazi, “A Task Decomposition Framework for Surveying the Crowd Contextual Insights,” *2015 IEEE 8th Int. Conf. Serv. Comput. Appl.*, pp. 155–162, 2015.

[64] S. Network, “Designing Complex Crowdsourcing Applications,” *J. Web Eng.*, vol. 14, no. 5, pp. 443–473, 2015.

[65] Y. Tong, L. Chen, Z. Zhou, and S. Member, “SLADE : A Smart Large-Scale Task Decomposer in Crowdsourcing,” *Trans. Knowl. Data Eng.*, vol. 4347, no. c, pp. 1–14, 2018.

[66] M. Patel and N. Patel, “Exploring Research Methodology,” *Int. J. Res. Rev.*, vol. 6, no. 3, pp. 48–55, 2019.

[67] T. Sastry, “The Significance of Research Methodology in Human Rights : A Bird’s Eye View,” *Delhi Law Rev.*, vol. 32, pp. 105–114, 2013.

[68] S. E. Group, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” 2007.

[69] B. Kitchenham and C. Ebse, “Guidelines for performing Systematic Literature Reviews in Software Engineering Executive summary,” 2007.

[70] P. Vartiainen, “Evaluation methods and comparative study,” *Univ. Vaarsa*, pp. 1–8, 2015.

[71] B. M. Ayyub, “A Practical Guide on Conducting Expert-Opinion Elicitation of Probabilities and Consequences for Corps Facilities,” p. 83, 2001.

[72] U. Gadiraju, R. Kawase, and S. Dietze, “A taxonomy of microtasks on the web,” *HT 2014 - Proc. 25th ACM Conf. Hypertext Soc. Media*, no. January 2016, pp. 218–223, 2014.

[73] V. Girotto, E. Walker, and W. Burleson, “The effect of peripheral micro-tasks on crowd

ideation," *Conf. Hum. Factors Comput. Syst. - Proc.*, vol. 2017-May, pp. 1843–1854, 2017.

[74] D. Version, "Estimating Conversational Styles in Conversational Microtask Crowdsourcing Estimating Conversational Styles in Conversational Microtask Crowdsourcing," *Proc. ACM Human-Computer Interact.*, vol. 4, pp. 32–57, 2020.

[75] P. Kucherbaev, F. Daniel, S. Tranquillini, and M. Marchese, "ReLauncher : Crowdsourcing Micro-Tasks Runtime Controller," *Proc. 19th ACM Conf. Comput. - Support. Coop. Work Soc. Comput.*, pp. 1609–1614, 2016.

[76] A. Aipe, "SimilarHITs : Revealing the Role of Task Similarity in Microtask Crowdsourcing," *Proc. 29th Hypertext socia Media*, pp. 115–122, 2018.

[77] C. M. Adriano and A. Van Der Hoek, "Exploring Microtask Crowdsourcing as a Means of Fault Localization," *Proc. - 3rd Int. Work. CrowdSourcing Softw. Eng. CSI-SE 2016*, pp. 1–28, 2016.

[78] D. Prasetya, M. Zuhri, and C. Candra, "Microtask Crowdsourcing Marketplace for Social Network," *2018 5th Int. Conf. Data Softw. Eng.*, pp. 1–6, 2018.

[79] E. Aghayi, "Large-Scale Microtask Programming," *2017 4th Int. Work. Crowdsourcing Softw. Eng.*, pp. 27–28.

[80] N. Kobayashi, M. Matsubara, K. Tajima, and A. Morishima, "A Crowd-in-the-Loop Approach for Generating Conference Programs with Microtasks," *2017 IEEE Int. Conf. Big Data*, pp. 4394–4396, 2017.

[81] U. Gadiraju, "Human Beyond the Machine : Challenges and Opportunities of Microtask Crowdsourcing," *AI Web Sci.*, pp. 81–85, 2015.

[82] A. Moayedikia, K. Ong, Y. Ling, and W. G. S. Yeoh, "Engineering Applications of Artificial Intelligence Task assignment in microtask crowdsourcing platforms using learning automata," *Eng. Appl. Artif. Intell.*, vol. 74, no. April, pp. 212–225, 2018.

[83] B. Naderi, "Motivation of Workers on Microtask Crowdsourcing Platforms," *T-Lab Ser. Telecommun. Serv.*, p. 125, 2018.

[84] C. Ho, S. Jabbari, and J. W. Vaughan, "Adaptive Task Assignment for Crowdsourced Classification," *Proc. 30th Int. Conf. Mach. Learn.*, vol. 28, pp. 534–542, 2013.

[85] M. Feldman and A. Bernstein, "Cognition-based Task Routing : Towards Highly-Effective Task-Assignments in Crowdsourcing Settings," *Proc. Int. Conf. Inf. Syst. - Build. a Better World through Inf. Syst. ICIS*, pp. 1–10, 2014.

[86] J. Jiang, Y. Zhou, Y. Jiang, Z. Bu, and J. Cao, "Knowledge-Based Systems Batch allocation for decomposition-based complex task crowdsourcing e-markets in social

networks ☆," *Knowledge-Based Syst.*, pp. 1–14, 2020.

[87] S. K. Mridha and M. Bhattacharyya, "Is Dutch Auction Suitable for Decomposable Tasks in Competitive Crowdsourcing Markets ?," *Hum. Comput.*, vol. 4, no. 1, 2017.

[88] B. Guo *et al.*, "Task Allocation in Spatial Crowdsourcing : Current State and Future Directions," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1–16, 2018.

[89] Y. Tong, Y. Zeng, B. Ding, L. Wang, and L. Chen, "Two-sided Online Micro-Task Assignment in Spatial Crowdsourcing," *Trans. Knowl. Data Eng.*, vol. 3, no. 5, pp. 1–14, 2019.

[90] J. Jiang, B. An, Y. Jiang, P. Shi, Z. Bu, and J. Cao, "Batch Allocation for Tasks with Overlapping Skill Requirements in Crowdsourcing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 8, pp. 1–14, 2019.

[91] D. Yu, "Crowdsourcing Software Task Assignment Method for Collaborative Development," *IEEE Access*, vol. 7, pp. 35743–35754, 2019.

[92] L. Qiao, F. Tang, and J. Liu, "Feedback Based High-Quality Task Assignment in Collaborative Crowdsourcing," *2018 IEEE 32nd Int. Conf. Adv. Inf. Netw. Appl.*, pp. 1139–1146, 2018.

[93] Y. Sun and J. Wang, "Dynamic Worker-and-Task Assignment on Uncertain Spatial Crowdsourcing," *IEEE 22nd Int. Conf. Comput. Support. Coop. Work Des.*, pp. 755–760, 2018.

[94] W. He, L. C. B, and C. Huang, "Task Assignments in Complex Collaborative Crowdsourcing," *CCF Conf. Comput. Support. Coop. Work Soc. Comput.*, pp. 574–580, 2019.

[95] I. Catallo, S. Coniglio, P. Fraternali, and D. Martinenghi, "A workload-dependent task assignment policy for crowdsourcing," *World Wide Web*, 2017.

[96] H. To, "Task Assignment in Spatial Crowdsourcing : Challenges and Approaches," *3rd ACM Sigspatial PhD Work.*, pp. 95–99, 2016.

[97] Y. Zhao, K. Zheng, Y. Cui, H. Su, F. Zhu, and X. Zhou, "Predictive Task Assignment in Spatial Crowdsourcing : A Data-driven Approach," *IEEE 36th Int. Conf. Data Eng.*, pp. 13–24, 2020.

[98] S. Kumar and M. Malay, "A network based mechanism for managing decomposable tasks via crowdsourcing," *Electron. Commer. Res.*, pp. 11–19, 2018.

[99] M. Alsayasneh *et al.*, "Personalized and Diverse Task Composition in Crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 1, pp. 1–14, 2017.

[100] P. Cheng, X. Lian, L. Chen, and C. Shahabi, “Prediction-Based Task Assignment in Spatial Crowdsourcing (Technical Report),” *IEEE 33rd Int. Conf. Data Eng.*, pp. 1–15, 2017.

[101] G. Little, L. B. Chilton, R. C. Miller, and M. Goldman, “TurKit : Tools for Iterative Tasks on Mechanical Turk,” *IEEE Symp. Vis. Lang. Human-Centric Comput.*, pp. 1–3, 2009.

[102] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller, “TurKit : Human Computation Algorithms on Mechanical Turk,” *Proc. 23rd Annu. ACM Symp. User interface Softw. Technol.*, pp. 57–66, 2010.

[103] A. Kittur, S. Khamkar, P. André, and R. E. Kraut, “CrowdWeaver : Visually Managing Complex Crowd Work,” *Proc. ACM 2012 Conf. Comput. Support. Coop. Work*, pp. 1033–1036, 2012.

[104] J. Noronha, E. Hysen, H. Zhang, and K. Z. Gajos, “PlateMate : Crowdsourcing Nutrition Analysis from Food Photographs,” *Proc. 24th Annu. ACM Symp. User interface Softw. Technol.*, pp. 1–12, 2011.

[105] R. E. Kraut, “CrowdForge : Crowdsourcing Complex Work,” *Proc. 24th Annu. ACM Symp. User interface Softw. Technol.*, pp. 43–52, 2011.

[106] A. Kulkarni, M. Can, and B. Hartmann, “Collaboratively Crowdsourcing Workflows with Turkomatic,” *Proc. ACM 2012 Conf. Comput. Support. Coop. Work*, pp. 1003–1012, 2012.

[107] T. D. Latoza, W. Ben Towne, C. M. Adriano, A. Van Der Hoek, and F. Ave, “Microtask Programming : Building Software with a Crowd,” *Proc. 27th Annu. ACM Symp. User interface Softw. Technol.*, pp. 43–54, 2014.

[108] M. S. Bernstein *et al.*, “Soylent : A Word Processor with a Crowd Inside,” *Proc. 23rd Annu. ACM Symp. User interface Softw. Technol.*, pp. 313–322, 2010.

APPENDIX A

QUALITY ASSESSMENT OF STUDIES FOR IDENTIFICATION OF MICROTASKING ACTIVITIES

Papers	**Respondents**														**Avg**
	Respondent 1							**Respondent 2**							
	Q1	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	
A Taxonomy of Microtasks on the Web	1	1	1	1	0.5	1	0.5	1	0.5	1	1	0.5	1	0.5	0.82
Cyber Cafes as Microtasking Hubs	0	0	0.5	1	0.5	0	0.5	0.5	0	0	0	0.5	0.5	0	0.28
Break It Down: A Comparison of Macro- and Microtasks	1	1	1	1	1	0.5	0.5	1	1	0.5	1	1	1	1	0.89
Efficient Task Decomposition in Crowdsourcing	1	1	1	0	0.5	0.5	0.5	0.5	0	0.5	0.5	1	0.5	0.5	0.57
The Effect of Peripheral Micro-tasks on Crowd Ideation	1	1	1	0.5	1	1	1	1	1	0.5	1	1	1	1	0.92
Estimating Conversational Styles in Conversational Microtask Crowdsourcing	1	1	1	0.5	1	1	1	1	1	1	1	1	0.5	0.5	0.89
ReLauncher: Crowdsourcing Micro-Tasks Runtime Controller	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
SimilarHITs: Revealing the Role of Task Similarity in Microtask Crowdsourcing	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

A Brief Perspective on Microtask Crowdsourcing Workflows for Interface Design	1	1	0.5	0.5	0.5	0	0	0	0	0.5	1	1	0.5	0	0.46
A Task Decomposition Framework for Surveying the Crowd Contextual Insights	1	0.5	0.5	1	0.5	1	1	1	1	1	0.5	0.5	0	0.5	0.71
A Profile-Aware Microtasking Approach for Improving Task Assignment in Crowdsourcing Services	1	1	1	0.5	0.5	0.5	0	0	0.5	0.5	0.5	0.5	1	0.5	0.57
Crowdsourced Microservices	1	1	0.5	0.5	0	1	0.5	0.5	0	0	1	1	0	0.5	0.53
Exploring Microtask Crowdsourcing as a Means of Fault Localization	1	1	1	0	0.5	0.5	0.5	0.5	0	0.5	0.5	1	0.5	0.5	0.57
Microtask Crowdsourcing Marketplace for Social Network	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Large-Scale Microtask Programming	1	1	1	0.5	1	1	1	1	1	1	1	1	0.5	0.5	0.89
A Crowd-in-the-Loop Approach for Generating Conference Programs with Microtasks	1	0.5	0.5	1	0.5	1	1	1	1	1	0.5	0.5	0	0.5	0.71
Ask the Crowd: Scaffolding Coordination and Knowledge Sharing in Microtask Programming	1	1	1	0.5	0.5	0.5	0	0	0.5	0.5	0.5	0.5	1	0.5	0.57

A Dynamic Microtask Scheduling Approach for SLO based Human-augmented Computing	0.5	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0
Two-sided Online Micro-Task Assignment in Spatial Crowdsourcing	1	1	0.5	0.5	0	0	0.5	0.5	0	0	1	1	0	0.5	0.46
Human Beyond the Machine: Challenges and Opportunities of Microtask Crowdsourcing	1	0.5	0.5	1	0.5	1	1	1	1	1	0.5	0.5	0	0.5	0.71
Microtask Programming	1	1	1	0.5	0.5	0.5	0	0	0.5	0.5	0.5	0.5	1	0.5	0.57
Crowdsourced Platforms: Objective, Activities and Motivation	1	0.5	0.5	1	0.5	1	1	1	1	1	0.5	0.5	0	0.5	0.71
Toward Microtask Crowdsourcing Software Design Work	1	1	1	0.5	1	1	1	1	1	1	1	1	0.5	0.5	0.89
Task assignment in microtask crowdsourcing platforms using learning automata	1	0.5	1	1	1	1	1	1	1	1	1	1	0.5	0.5	0.89
Microtasking: redefining crowdsourcing practices in emergency management	1	0.5	0.5	1	0.5	1	1	1	1	1	0.5	0.5	0	0.5	0.71
Implementing Microservices through Microtasks	1	1	1	0.5	1	1	1	1	1	0.5	1	1	1	1	0.92
Digital labour platforms and the future of work	0	0.5	0	1	0	0.5	1	0	0	0	0	0	0	0.5	0.25

Age-Based Task Specialization for Crowdsourced Proofreading	0.5	0.5	0	1	0	0.5	1	0	0	0	0.5	0	0	0.5	0.32
Crowdsourced Reverse Engineering: Experiences in Applying Crowdsourcing to Concept Assignment	1	0.5	0.5	1	0.5	1	1	1	1	1	0.5	0.5	0	0.5	0.71
Crowd Anatomy Beyond the Good and Bad: Behavioral Traces for Crowd Worker Modeling and Pre-selection	0	0.5	0	1	0	0.5	1	0	0	0	0	0	0	0.5	0.25
Quality Assurance Strategies in Microtask Crowdsourcing	1	1	1	0.5	0.5	0.5	0	0	0.5	0.5	0.5	0.5	1	0.5	0.57
Motivation of Workers on Microtask Crowdsourcing Platforms	1	1	1	0.5	1	1	1	1	1	1	1	1	0.5	0.5	0.89
Efficient Task Decomposition for Sequential Crowdsourced Task Solving	1	1	0.5	0	1	1	0.5	0.5	1	0	1	1	0.5	0.5	0.67

APPENDIX B

QUALITY ASSESSMENT OF STUDIES FOR IDENTIFICATION OF TASK DECOMPOSITION METHODS

Papers	Respondents														Avg
	Respondent 1							Respondent 2							
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
Adaptive Task Assignment for Crowdsourced Classification	1	1	1	1	0.5	1	0.5	1	0.5	1	1	0.5	1	0.5	0.82
The effects of task instructions in crowdsourcing innovative ideas	0	0	0.5	1	0.5	0	0.5	0.5	0	0	0	0.5	0.5	0	0.28
Cognition-based Task Routing: Towards Highly-Effective Task-Assignments in Crowdsourcing Settings	1	1	1	1	1	0.5	0.5	1	1	0.5	1	1	1	1	0.89
Batch allocation for decomposition -based complex task crowdsourcing e-markets in social networks	1	1	1	0.5	1	1	1	1	1	0.5	1	1	1	1	0.92
Is Dutch Auction Suitable for Decomposable Tasks in Competitive Crowdsourcing Markets?	1	1	1	0.5	1	1	1	1	1	1	1	1	0.5	0.5	0.89
Why Individuals Participate in Micro-task Crowdsourcing Work Environment: Revealing Crowdworkers ' Perceptions	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0

Efficient Task Decomposition for Sequential Crowdsourced Task Solving	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Automated crowdsourcing task generation	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A Novel Approach to Publishing Tasks for Collaborative Crowdsourcing Workflows	0	0.5	0	1	0	0.5	1	0	0	0	0	0	0	0.5	0.25
SLADE: A Smart Large-Scale Task Decomposer in Crowdsourcing	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Task Assignment on Multi-Skill Oriented Spatial Crowdsourcing (Tech Report)	1	1	0.5	0.5	0.5	0	0	0	0	0.5	1	1	0.5	0	0.46
Task Allocation in Spatial Crowdsourcing : Current State and Future Directions	1	1	1	0.5	0.5	0.5	0	0	0.5	0.5	0.5	0.5	1	0.5	0.57
Task Composition in Crowdsourcing	1	1	0.5	0.5	0	0	0.5	0.5	0	0	1	1	0	0.5	0.46
Two-sided Online Micro-Task Assignment in Spatial Crowdsourcing	1	1	1	0	0.5	0.5	0.5	0.5	0	0.5	0.5	1	0.5	0.5	0.57
A Task Decomposition Framework for Surveying the Crowd Contextual Insights	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Batch Allocation for Tasks with Overlapping Skill Requirements in Crowdsourcing	1	1	1	0.5	1	1	1	1	1	1	1	1	0.5	0.5	0.89
Crowdsourcing Software Task Assignment Method for Collaborative Development	1	0.5	0.5	1	0.5	1	1	1	1	1	0.5	0.5	0	0.5	0.71

Crowdsourced Microservices	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Destination-aware Task Assignment in Spatial Crowdsourcing : A Worker Decomposition Approach	0.5	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0
On task assignment for real-time reliable crowdsourcing	1	1	0.5	0.5	0	0	0.5	0.5	0	0	1	1	0	0.5	0.46
Feedback Based High-Quality Task Assignment in Collaborative Crowdsourcing	1	0.5	0.5	1	0.5	1	1	1	1	1	0.5	0.5	0	0.5	0.71
Dynamic Worker-and-Task Assignment on Uncertain Spatial Crowdsourcing	1	1	1	0.5	0.5	0.5	0	0	0.5	0.5	0.5	0.5	1	0.5	0.57
Online Mobile Micro-Task Allocation in Spatial Crowdsourcing	1	1	0.5	0.5	0	0	0.5	0.5	0	0	1	1	0	0.5	0.46
Task Assignments in Complex Collaborative Crowdsourcing	1	1	1	0.5	1	1	1	1	1	1	1	1	0.5	0.5	0.89
Preference-Aware Task Assignment in Spatial Crowdsourcing	1	1	0.5	0.5	0	0	0.5	0.5	0	0	1	1	0	0.5	0.46
A workload-dependent task assignment policy for crowdsourcing	1	1	1	0.5	1	1	1	1	1	1	1	1	0.5	0.5	0.89
Task Assignment in Spatial Crowdsourcing : Challenges & Approaches	1	0.5	0.5	1	0.5	1	1	1	1	1	0.5	0.5	0	0.5	0.71
Predictive Task Assignment in Spatial Crowdsourcing : A Data-driven Approach	1	1	1	0.5	1	1	1	1	1	0.5	1	1	1	1	0.92
A network based mechanism for managing	1	0.5	0.5	1	0.5	1	1	1	1	1	0.5	0.5	0	0.5	0.71

decomposable tasks via crowdsourcing															
Age-Based Task Specialization for Crowdsourced Proofreading	0.5	0.5	0	1	0	0.5	1	0	0	0	0.5	0	0	0.5	0.32
Personalized and Diverse Task Composition in Crowdsourcing	1	0.5	0.5	1	0.5	1	1	1	1	1	0.5	0.5	0	0.5	0.71
An Explorative Approach for Crowdsourcing Tasks Design	0	0.5	0	1	0	0.5	1	0	0	0	0	0	0	0.5	0.25
Prediction-Based Task Assignment in Spatial Crowdsourcing (Tech Report)	1	1	1	0.5	0.5	0.5	0	0	0.5	0.5	0.5	0.5	1	0.5	0.57
Profit-driven Task Assignment in Spatial Crowdsourcing	1	1	0.5	0.5	0	0	0.5	0.5	0	0	1	1	0	0.5	0.46
A Profile-Aware Microtasking Approach for Improving Task Assignment	1	1	1	0.5	1	1	1	1	1	1	1	1	0.5	0.5	0.89

APPENDIX C

DATA EXTRACTION TABLES

Table1 Data Extraction P01

Paper Id	**P01**
Data Source	ACM
Title	A Taxonomy of Microtasks on the Web
Author	Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze
Year of Publication	2014
Publication Type	Conference
Conference/Journal Name	Proceedings of the 25th ACM conference on Hypertext and social media
Methodology	Survey (Questionnaire)
Findings	Metadata finding Content verification, Spam detection, Data matching Data categorization, Data classification Media transcription Ranking Content moderation Sentiment analysis Data collection and enhancement (Content creation) Content feedback (Survey) Promotion e.g., webpages Quality assessment and testing Data selection, translation and tagging

Table2 Data Extraction P02

Paper Id	**P02**
Data Source	ACM
Title	Break It Down: A Comparison of Macro- and Microtasks
Author	Justin Cheng, Jaime Teevan, Shamsi Iqbal, Michael S. Bernstein
Year of Publication	2015
Publication Type	Conference
Conference/Journal Name	*ACM Conference on Human Factors for Computing Systems (CHI)*
Methodology	Experiment
Findings	Gathering of words/terms for taxonomy creation Copyediting Transcribing the speech's sentences

Table3 Data Extraction P03

Paper Id	**P03**
Data Source	ACM
Title	The Effect of Peripheral Micro-tasks on Crowd Ideation
Author	Victor Girotto, Erin Walker, Winslow Burleson
Year of Publication	2017
Publication Type	Conference
Conference/Journal Name	Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems
Methodology	Experiment
Findings	Designing a single component of logo in T-shirt designing.

Table4 Data Extraction P04

Paper Id	**P04**
Data Source	ACM
Title	Estimating Conversational Styles in Conversational Microtask Crowdsourcing
Author	Sihang Qiu, Ujwal Gadiraju, Alessandro Bozzon
Year of Publication	2020
Publication Type	Conference
Conference/Journal Name	Proceedings of ACM Human-Computer Interaction (CSCW)
Methodology	Experiments on AMT
Findings	Dataset's module creation Information finding Image transcription

Table5 Data Extraction P05

Paper Id	**P05**
Data Source	ACM
Title	ReLauncher: Crowdsourcing Micro-Tasks Runtime Controller
Author	Pavel Kucherbaev, Florian Daniel, Stefano Tranquillini, Maurizio Marchese
Year of Publication	2016
Publication Type	Conference
Conference/Journal Name	The 19th ACM conference
Methodology	Experiment on CrowdFlower
Findings	Data Collection Data Analysis Label an image Translate a sentence Human computation

Table6 Data Extraction P06

Paper Id	**P06**
Data Source	ACM
Title	SimilarHITs: Revealing the Role of Task Similarity in Microtask Crowdsourcing
Author	Alan Aipe, Ujwal Gadiraju
Year of Publication	2018
Publication Type	Conference
Conference/Journal Name	29th ACM Conference on Hypertext and Social Media
Methodology	Experiment
Findings	Survey Content creation Content access Interpretation and analysis Verification and validation Information finding

Table7 Data Extraction P07

Paper Id	**P07**
Data Source	IEEE
Title	A Profile-Aware Microtasking Approach for Improving Task Assignment in Crowdsourcing Services
Author	Jabu Mtsweni, Ernest Ketcha Ngassam, Legand Burge III
Year of Publication	2016
Publication Type	Conference
Conference/Journal Name	IIMC International Information Management Corporation
Methodology	Design Science Research (DSR) methodology
Findings	Information gathering Tagging Classification Taking a photo of products at specific shops Digitizing local-language documents Human Optical Recognition tasks

Table8 Data Extraction P08

Paper Id	**P08**
Data Source	IEEE
Title	Exploring Microtask Crowdsourcing as a Means of Fault Localization
Author	Christian M. Adriano, Andre van der Hoek

Year of Publication	2016
Publication Type	Conference
Conference/Journal Name	Proceedings of ACM Human-Computer Interaction (CSCW)
Methodology	Experiment through AMT
Findings	Writing a piece of code Debugging Writing test-cases Adding pseudo-code Testing line of code Algorithmic debugging Delta debugging Locate known faults in code fragments

Table9 Data Extraction P09

Paper Id	**P09**
Data Source	IEEE
Title	Microtask Crowdsourcing Marketplace for Social Network
Author	Davin Prasetya, Muhammad Zuhri Catur Candra
Year of Publication	2018
Publication Type	Conference
Conference/Journal Name	5th International Conference on Data and Software Engineering (ICoDSE)
Methodology	Experiment through LINE
Findings	Image classification Product comparison Image moderation Collecting information Survey Sentiment analysis

Table10 Data Extraction P10

Paper Id	**P10**
Data Source	IEEE
Title	Large-Scale Microtask Programming
Author	Emad Aghayi
Year of Publication	2020
Publication Type	Conference
Conference/Journal Name	5th International Conference on Data and Software Engineering (ICoDSE)
Methodology	Controlled experiment
Findings	Implement a unit test

	Debugging Identify, test, implement and debug behaviors

Table11 Data Extraction P11

Paper Id	**P11**
Data Source	IEEE
Title	A Crowd-in-the-Loop Approach for Generating Conference Programs with Microtasks
Author	Naoki Kobayashi, Masaki Matsubara, Keishi Tajima, Atsuyuki Morishima
Year of Publication	2017
Publication Type	Conference
Conference/Journal Name	IEEE International Conference on Big Data
Methodology	Experiment
Findings	Metadata elicitation Data classification Data clustering

Table12 Data Extraction P12

Paper Id	**P12**
Data Source	IEEE
Title	Human Beyond the Machine: Challenges and Opportunities of Microtask Crowdsourcing
Author	Ujwal Gadiraju, Gianluca Demartini, Ricardo Kawase, Stefan Dietze
Year of Publication	2015
Publication Type	Journal
Conference/Journal Name	IEEE Intelligent Systems (Volume: 30, Issue: 4)
Methodology	Experiment on AMT
Findings	Information finding Data verification and validation Interpretation and analysis Content creation Content access Audio translation Language translation Annotation

Table13 Data Extraction P13

Paper Id	**P13**
Data Source	IEEE
Title	Microtask Programming

Author	Thomas D. LaToza, Arturo Di Lecce, Fabio Ricci, W. Ben Towne, Andre van der Hoek
Year of Publication	2018
Publication Type	Journal
Conference/Journal Name	IEEE Transactions on Software Engineering (Volume: 45, Issue: 11)
Methodology	Empirical study
Findings	Implementing part of a function Unit test Edit a function Writing test cases Debugging

Table14 Data Extraction P14

Paper Id	**P14**
Data Source	ACM
Title	Toward Microtask Crowdsourcing Software Design Work
Author	Edgar R.Q. Weidema, Consuelo Lopez, Sahand Nayebaziz, Fernando Spanghero, Andre van der Hoek
Year of Publication	2016
Publication Type	Workshop
Conference/Journal Name	3rd International Workshop on Crowdsourcing in Software Engineering
Methodology	Experiment on AMT
Findings	Identification of main decision points from set of requirements Identification of alternative solution Identification of design problems Sketching of small design related to interface

Table15 Data Extraction P15

Paper Id	**P15**
Data Source	Science Direct
Title	Task assignment in microtask crowdsourcing platforms using learning automata
Author	Alireza Moayedikia, Kok-Leong Ong, Yee Ling Boo, William G.S. Yeoh
Year of Publication	2018
Publication Type	Journal
Conference/Journal Name	Engineering Applications of Artificial Intelligence (Volume: 74)
Methodology	Literature Review
Findings	Identification of missing values in the dataset

Table16 Data Extraction P16

Paper Id	**P16**
Data Source	Science Direct
Title	Microtasking: redefining crowdsourcing practices in emergency management
Author	Marta Poblet, Dr Mari Fitzpatrick and Professor Prem Chhetri
Year of Publication	2017
Publication Type	Journal
Conference/Journal Name	Australian Journal of Emergency Management
Methodology	Literature Review & 3 Case studies
Findings	Translation Documentation Categorization Mapping Data collection, verification, filtration and synthesize the data Restructure the data into standardized reports Sharing of data with different sites Data searching Logging of information onto a page Checking and listing of websites Interpretation of visual data

Table9 Data Extraction P17

Paper Id	**P17**
Data Source	Science Direct
Title	Implementing Microservices through Microtasks
Author	Emad Aghayi, Thomas D. LaToza, Paurav Surendra, Seyedmeysam Abolghasemi
Year of Publication	2019
Publication Type	Journal
Conference/Journal Name	IEEE Transactions on Software Engineering (Volume: 46, Issue: 9)
Methodology	Case study
Findings	Writing a test case Identify, test, implement and debug individual behavior Implementation of function behavior Review of function behavior

Table18 Data Extraction P18

Paper Id	**P18**
Data Source	Springer
Title	Crowdsourced Reverse Engineering: Experiences in Applying Crowdsourcing to Concept Assignment
Author	Sebastian Heil, Valentin Siegert, Martin Geadki

Year of Publication	2019
Publication Type	Book
Conference/Journal Name	Evaluation of Novel Approaches to Software Engineering (pp.215-239)
Methodology	Experiment on microworkers.com
Findings	Data classification Checking and listing of data Data analysis

Table19 Data Extraction P19

Paper Id	**P19**
Data Source	Springer
Title	Quality Assurance Strategies in Microtask Crowdsourcing
Author	Pavel Kucherbaev
Year of Publication	2016
Publication Type	PhD Thesis
Conference/Journal Name	The University of Trento, Italy
Methodology	Survey
Findings	Audio transcription Surveys Design a small module of interface Interpretation of visual data

Table20 Data Extraction P20

Paper Id	**P20**
Data Source	Springer
Title	Motivation of Workers on Microtask Crowdsourcing Platforms
Author	Babak Naderi
Year of Publication	2017
Publication Type	Series
Conference/Journal Name	T-Labs Series in Telecommunication Services
Methodology	Industrial review & Experiment on MTurk, Microworkers and Crowdee (Crowdsourcing and microtasking platforms)
Findings	Image tagging Audio/video transcription Quality assessment e.g., testing Information finding Verification and validation Interpretation and analysis Content creation Surveys Content access e.g., watch an online video

APPENDIX D

DEMOGRAPHIC DATA OF THE PARTICIPANTS OF EXPERT REVIEW

Job Title/Position	Experience in (y)	Country
Senior Software Engineer	6+	Abu Dhabi
Sr PHP developer	5+	UK
Associate Professor	14+	Pakistan
Assistant Professor	7+	Pakistan

APPENDIX E

EXPERT EVALUATION DOCUMENT

My name is Maira Zulfiqar, MS Software Engineering student at NUML. The topic of my thesis is “A Mechanism for task decomposition of Microtasking in Crowdsourced Software Development”. For my thesis, I have identified the microtasking activities in crowdsourced software development. Identified microtasking activities have been classified into various categories. The aim to conduct expert evaluation is to validate the naming conventions of the microtasking activities. Moreover, whether microtasking activities correctly lie under mentioned categories.

As you have expertise in crowdsourced software development, you are requested to validate the naming conventions and categorization of microtasking activities. Please write your comments against each category (if any). You can also suggest any other category or microtasking activities which are not mentioned here.

Thank you!

S. No	Categories	Microtasking activities	Description about micro tasking activities	Comments
1.	Information finding	Metadata finding Organizing the data Information finding Data collection Information gathering Filtration and synthesize the data	Any type of microtask which requires the information finding e.g., author name of any research article, its published date and exploring of email address of owners from company's website.	
2.	Verification & validation	Content verification Spam detection Data matching Data tagging Product comparison	Microtasks which are used to verify and validate the data e.g., verification of service providing by the freelancers against description written in their offer (or gig).	
3.	Content creation	Data categorization Data enhancement Data classification Data selection Gathering of terms for taxonomy creation Dataset's module creation Label an image Pasting the data Data mapping Addition of annotations Restructure the data into standardized reports	Any small task which can be used to create the content e.g., gathering of words/terms for the creation of taxonomy, conversion of random data into the useful information, and then into the standardize documents.	

		Documentation Listing of data Organizing the data Data collection		
4.	Data transcription	Media transcription Data translation Image transcription Transcribing the speech's sentences Digitizing local-language documents Human Optical Recognition tasks Audio translation Language translation Video translation	Microtasks related to translation and transcription e.g., translation of a paragraph, audio and video from one language to another, or converting the image file into the editable text file etc.	
5.	Interpretation & Analysis	Sentiment analysis Content moderation Data Analysis Data interpretation Interpretation of visual data Checking and listing of websites	Microtasks related to the opinions and feelings of a specific entity e.g., "What do you think about new features launched in iPhone X?". These tasks depend on the intelligence and perception of the crowd.	
6.	Surveys	Content feedback Conduct an interview	Microtasks which are related to the survey for taking feedback against the products and services.	
7.	Content access	Promotion e.g., webpages Copying of the data Content access Capture the photos Sharing of data with different sites Logging of information onto a page Watch an online video	These microtasks usually require the workers to consume time, visit the website or any specific location and access the data by clicking on the link provided e.g., "Click on the link below and watch the animated video for further understanding".	

8.	Quality assessment	Debugging of program Debugging of UI Test a line of code Algorithmic debugging Delta debugging Implement a unit test Identify, test, implement and debug behaviors in code Locate known faults in code fragments Review of function behavior	Microtasks to ensure the quality of a code, design or interface e.g., identification and removal of errors from the code, algorithm and design.	
9.	Designing	Designing a single component of logo Selection of fonts Sketching of small design related to interface Identification of design problems	Microtasks related to the graphic designing.	
10.	Development	Writing a piece of code Writing test-cases Edit a function Adding pseudo-code Implementing part of a function Human computation	Microtasks related to the programming.	
11.	Identification	Identification of main decision points from set of requirements Identification of alternative solution Identification of missing values in the dataset	Microtasks related to the identification of any missing data or information. These microtasks are performed by the experts of their relevant fields.	