The Impact of COVID-19 in Software Design Activities in Global Software Engineering – A Study from Security Perspective

Faisal Nabi (FAISAL.NABI@YAHOO.COM)  
Muhammad Ali Jinnah University

Hina Fatima  
Maju https://orcid.org/0000-0002-1757-1200

Systematic Review

Keywords: Global Software Engineering (GSE), Human Aspects of Distributed Development, GSE Education and Training, social interaction

Posted Date: September 22nd, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3355942/v2

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations:  
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
The Impact of COVID-19 in Software Design Activities in Global Software Engineering – A Study from Security Perspective

Faisal Nabi, Hina Fatima
Internet Security Research Group, Faculty of Computing
Muhammad Ali Jinnah University, Karachi, Pakistan

Department of Management Science, Muhammad Ali Jinnah University, Karachi, Pakistan

Background: The greatest significant global health catastrophe of the century is thought to be the COVID-19 epidemic. Software development is not an exception to how it has affected other corporate sectors around the world. The effects of COVID-19 on software projects and software development professionals are seriously threatened. The social effects on Global Software Engineering (GSE) extends geographical, temporal, and cultural boundaries in distributed environments. Over the past two decades, GSE research has evolved to manage software development for distributed teams. The COVID-19 pandemic highlights the need for comprehensive research, particularly during the software design phase, to support team collaboration in distributed development. Aim: This study systematically analyzes the evolution of research emphasis in the GSE field, specifically exploring whether the research focuses increasing on software design due to the global pandemic, also discuss the social interaction gap in design phase of software development. Method: We systematically analyzed the existing literature in two phases. In the first phase of our study, we mapped GSE research over the two decades leading to the pandemic (2000-2020). In the second phase, we used the forward snowballing approach to examine the literature on the software design phase published between 2020 and 2022. Results: The analysis of 592 research studies in the two phases reveals various trends in GSE research. Evaluation research is the most explored research type in methods and processes, and human aspects of development. Despite the paradigm shift caused by the COVID-19 pandemic that increased reliance on distributed teams, results show that while software organizations are extensively studied across all software engineering phases, the software design phase remains one of the least explored areas. Conclusion: This work highlights the evolving GSE research trends, emphasizing the rising significance of collaborative software design in distributed settings. Our findings address current research gaps and underscore (The technical and non-technical challenges involve three dimensions of distance-temporal, geographical, and socio-cultural) the need for further research on software design activities along with social interaction gap in software team-based design. This contribution envisions a more collaborative, adaptable GSE field, guiding future research to support distributed team.

Keywords: Global Software Engineering (GSE); Human Aspects of Distributed Development; GSE Education and Training, social interaction.

1. Introduction

Software development teams have evolved from small co-located to large geographically distributed locations in the past two decades. In parallel, Global Software Engineering (GSE) has emerged to manage geographical, temporal, and cultural boundaries in distributed environments. The adoption of GSE has been widespread
in software organizations due to the COVID-19 pandemic [1], as organizing software engineering activities in geographically distributed development teams became a common practice [2]. The increasing globalization of the software industry with different stakeholders and infrastructures has drawn attention to GSE. Researchers and practitioners [3] presented methods and processes [4; 5], focusing on coordination [6; 7], tools to enhance collaboration [8; 9], strategies to support a GSE-based business model to hire distributed teams [10; 11], and pedagogies for GSE curricula [12; 13] to improve project management in a distributed environment. While the research in GSE is progressively growing [14] and provides support for diverse challenges [15], researchers highlighted the limited literature available to support and manage software design activities in distributed teams [16].

This study aims to understand whether the pandemic increased the focus on software design activities to avoid architecture erosion in distributed teams. To this end, we systematically analyzed the existing literature by completing two phases. First, we conducted a systematic mapping study [17] to obtain an overview of the GSE research area in the two decades preceding the pandemic; then, we used snowballing to retrieve and analyze only the papers focusing on the design phase published after 2020. Results show that the software design phase is the least discussed by the GSE community to manage software design and architecture activities in a distributed environment. After the global pandemic, the related research for software design in GSE still needs to be improved to support collaboration in distributed teams.

The paper is organized as follows. Section 2 provides the background of GSE. Section 3 presents the existing systematic mapping studies in GSE. Section 4 explains the research method used in the study. Section 5 presents the results, and Section 6 further discusses the research gaps and future implications. Section 7 elaborates the threats to validity. Finally, Section 8 concludes the paper with suggestions for future work.

2. GSE Summary

This section provides the conceptual foundation of GSE and an overview of the existing literature.

2.1. GSE as a Concept

Global Software Engineering (GSE) has become prominent in the software industry, growing from a phenomenon to a paradigm shift for software development processes in distributed environments [18]. While GSE is often used interchangeably with Global Software Development (GSD), it specifically refers to the software development process in a distributed environment. In contrast, GSD reflects the code-centric distributed development to achieve optimal software prototypes [19]. To establish the foundational framework of GSE, Gumm [20] presented four distribution dimensions: geographical, temporal, organizational, and stakeholder group
distribution. These dimensions serve as a categorization mechanism for individuals and artifacts in GSE projects. The author used a 3-point ordinal scale (low, medium, high) to understand their correlation.

GSE has evolved across different forms over time, initially encompassing standard software production methods for large [21] and small [22] companies. It then transitioned to outsourcing or offshore tasks [23] to address concerns related to project development costs, securing high-skilled resources, and expanding global production to counter international market demands [24]. In parallel, educational institutes started designing courses and pedagogies to equip software engineering students with practical GSE experience within distributed teams [25; 26].

2.2. Historical Review of GSE

This section explores the evolution of GSE research (shown in Figure 1), discussing five major research areas to gain insights into the shifts in areas within the field.

2.2.1. Human aspect of development

In GSE, researchers and practitioners have addressed diverse trends to enhance team collaboration, knowledge management, and software quality management [2]. The technical and non-technical challenges involve three dimensions of distance-temporal, geographical, and socio-cultural [27] for supporting distributed teams. The recommended solutions and practices for these challenges are cataloged in various ways [28; 29]. In order to manage the coordination activities, researchers suggested task allocation models that consider critical factors before assigning tasks within distributed teams [30; 31]. Such models are beneficial in addressing GSE project coordination and management challenges [32].

From a socio-cultural perspective, diversity and inclusion have emerged as significant factors influencing GSE practices [33]. The related research highlights various benefits of diversity; for example, it increases team creativity, innovation,
and problem-solving skills [34]. Since software organizations working in distributed teams recruit highly skilled professionals from the international market, researchers and practitioners have highlighted the complexities of onboarding newcomers in ongoing distributed projects [35; 36]. The related research in this aspect continues to evolve, aiming to refine the onboarding experience and furnish guidelines for mitigating challenges faced by software organizations.

2.2.2. Methods and processes

While the human aspects of GSE have become a fundamental area of research, various methods and processes have been explored to facilitate global collaboration. Agile methodology is considered the best practice for projects with high uncertainty [37]. This approach has been adapted to GSE projects, which heavily rely on face-to-face communication [38]. Examples include Scrum [39; 40], eXtreme Programming (XP) [41], Scale Agile Framework (SAFe) [42], and global DevOps [43], all effectively tailored to the GSE environment [44]. As the GSE field evolved, the research studies shared the growing complexity of executing software development activities such as planning, requirement engineering, design, development, and testing in a distributed environment [45; 46]. In order to mitigate potential business risks, methods have been proposed to integrate knowledge management for distributed teams [47]. The related literature also underlined the importance of communication as an essential and continuous process at every phase of software development [48].

With recent GSE advancements, cloud computing has emerged as a promising method for assisting software organizations in outsourcing development activities through advanced technologies [49]. However, various challenges incorporate cloud computing in GSE [50]. Amidst the global pandemic, emerging process support for remote work has gained prominence, with hybrid teams forming a vital part of the ongoing GSE transition [51].

2.2.3. GSE education and training

GSE has transformed software project management and development processes in both academic and industrial settings. With the growing complexity of handling software project management and development activities in a distributed environment, universities have emphasized the significance of GSE education. This emphasis ensured that software engineering students acquire the requisite knowledge and skills to guide them accordingly [52]. Consequently, numerous universities have introduced GSE-based courses designed to equip students with the required skill set to coordinate and manage projects within distributed teams [53; 54]. In this context, the literature outlines three primary approaches: project-based collaborations among multiple universities [55], integration of open-source projects with industrial clients [56], and serious-game-based simulations [57], all of which provide students with practical exposure to GSE scenarios.
The global pandemic has inevitably impacted various social and economic activities worldwide, including those within the education sector, forced to shift from co-located to distributed environments. As a result, researchers have shared experience reports on conducting online courses during the pandemic [58; 59], offering insights into the implications for future research in the field.

2.2.4. Emerging technologies

In GSE, software technologies are essential to enhance collaboration and knowledge management among distributed teams. In order to prevent knowledge vaporization, various collaboration tools have been proposed for knowledge management, incorporating features such as wiki, document management, or blogs [60]. Similarly, to improve the quality of the GSE products, software process management tools have been introduced to enhance interaction among distributed teams throughout various phases of software development [61]. The biggest group of tools comprises research-based solution proposals for managing software development activities [60; 61]. However, this tool increase is not uniform across all studied software engineering phases.

The COVID-19 outbreak disrupted face-to-face communication within software organizations, prompting exploration into effective media to keep collaboration and knowledge-sharing among distributed teams. This led to the integration of external software development tools like GitHub [62]. As a result, many software organizations started using enterprise social network (ESN) [63] applications such as Slack, Yammer, Microsoft Teams, Jira [8; 64]. This trend indicates a shift in distributed teams, not only in software development but also in team organization dynamics.

2.2.5. Global business strategy

Software organizations currently use various methods for software product development, including ad-hoc software, outsourcing projects, offshore and onshore initiatives, expansion through acquisitions, and joint ventures. Among these methods, offshore and onshore projects, a form of GSE practice, have gained preference. This involves software organizations leveraging resources from a separate entity within the same or different country [65]. However, industrial surveys indicate that such projects bring forth distinct business risks. To mitigate them, practitioners have shared various strategies for refining organizations’ business models [66; 67]. As strategy evolves to transform software organizational frameworks, the literature outlines a multi-site governance model to structure distributed teams [68] effectively. This model is based on organizing and administering distributed teams and managing collaboration, communication, and knowledge management processes across various operational tiers of the organization [68].

Amidst the global pandemic, many software organizations have transitioned their organizational structure to align with GSE practices. However, due to limited experience and preparedness for working in a distributed environment, many
software organizations encountered challenges adapting to this change [69]. Despite these specific challenges of working in a distributed environment, numerous software organizations acknowledge the benefits of operating with diverse and distributed teams [70]. This underscores the need for further research to explore effective business strategies for software organizations.

3. Related Work: Existing Systematic Mapping Studies in GSE

To classify the terminologies used for GSE, Šmite et al. [23] presented a comprehensive mapping study followed by a survey involving GSE experts to build a glossary sharing the key concepts in the GSE field. The taxonomy lacked the language and cultural dimensions as part of the GSE taxonomy, which was later covered by Britto et al. [19] as an extended GSE taxonomy. Jalali and Wohlin [71] presented a systematic mapping to explore the agile practices used in the field. The results show that agile practices allow to adapt accordingly to support specific project-based characteristics in the distributed environment. Since GSE projects require continuous communication (synchronous and asynchronous) and collaboration among geographically distributed teams, Portillo et al. [72] conducted a systematic mapping to classify what type of software tools (commercial, free, or research-based) are available to assist GSE software development activities. Similarly, Chadli et al. [73] reviewed software project management tools used in GSE projects to analyze the support for communication, coordination, and cooperation. The authors found limited tools to assist software project management areas, especially for decision management, risk management, and measurement. This prevents distributed teams from objectively providing quality products. Bajta et al., [74] reviewed software project management approaches used in GSE and highlighted the limited research available to extend the contribution in the field. Moreover, it emphasizes the need for more effective collaboration between the software industry and software engineering education and training that can help enhance formal development through practical projects.

With respect to the related systematic mapping studies, our study provides a holistic view of different areas of research in GSE. Further, it investigates whether the focus towards software design activities increases in distributed software organizations post-pandemic.

4. Research Method

As detailed in Figure 2, we systematically analyzed the existing literature by completing two phases. First, we conducted a systematic mapping study [17] to obtain an overview of the GSE research area in the two decades preceding the pandemic; then, we used forward snowballing [75] to retrieve and analyze only the papers focusing on the design phase published after 2020. Forward snowballing involves identifying and including new citations from the research studies that reference the
initial set of papers. This method helps ensure the inclusion of the most recent and relevant studies, thereby enhancing the comprehensiveness of our analysis.

Fig. 2. The research method of this study.

4.1. Research Questions
To obtain a comprehensive view of GSE, in both phases of this study, we considered the following research questions to build a solid foundation for the evolution of the existing literature in the field:

RQ1. What is the temporal distribution of the studies?
RQ2. What are the types of research contribution?
RQ3. Who are the main stakeholders of the studies?
RQ4. What are the main areas of focus of GSE research?

In the first phase, we considered an additional research question that is not relevant to the second phase as it only focused on a specific software engineering phase (i.e., software design):

RQ5. What are the software engineering phases considered in the studies?

Table 1 compares the existing systematic studies (Section 3) to highlight the specific focus and research gap aligned with the research questions.

4.2. Search for Primary Studies
For the first phase, we used the following comprehensive query string to analyze the evolution of research within the area before the pandemic (i.e., 2000-2020):

“Global Software Engineering” OR “Global Software Development” OR “Distributed Software Development” OR “Large Scale Agile Development"
Table 1. Focus of the existing systematic studies on GSE and of this study.

<table>
<thead>
<tr>
<th>Ref</th>
<th>Year</th>
<th>RQ1</th>
<th>RQ2</th>
<th>RQ3</th>
<th>RQ4</th>
<th>RQ5</th>
<th>Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>[71]</td>
<td>2010</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>Agile practices in GSE</td>
</tr>
<tr>
<td>[72]</td>
<td>2012</td>
<td>P</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Communication and coordination tools to support distributed teams</td>
</tr>
<tr>
<td>[73]</td>
<td>2016</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Software project management tools in GSE</td>
</tr>
<tr>
<td>[74]</td>
<td>2018</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Software project management approaches in GSE</td>
</tr>
</tbody>
</table>

Legend: P = Partial, Y = Yes, N = No

We defined the search string iteratively by combining synonyms, as GSE practitioners and researchers use different alternative terms [23; 19]. For example, practitioners commonly use “distributed software development” to share the idea of teams working in a distributed environment. Similarly, researchers and practitioners use “large scale development” to discuss software organizations adopting agile development practices for working in distributed environments [44].

To ensure a comprehensive extraction of relevant studies, in the first phase, we applied the search string to meta-data in four digital libraries (IEEE, ACM, ScienceDirect, and Springer) to extract research studies from 2000 to 2020. The selection of this specific time frame for the systematic mapping depended on the dynamic evolution of software development practices that significantly impacted the landscape of software engineering, particularly in the context of globalization and distributed collaboration [4]. With the rise of globalization, rapid technological advancements, and widespread internet connectivity, the early 21st century marked a turning point for the way software development was approached across geographical boundaries [76]. The increasing emphasis on remote work, the growth of software organizations, and the recognition of the benefits of cultural diversity in teams advanced the practice of GSE forward [2]. Moreover, 2000 to 2020 correspond to the emergence and development of GSE concepts, enabling a comprehensive analysis of the field's evolution, challenges, and solutions developed over two decades.

In the second phase, we used forward snowballing [75] to retrieve all the work on software design in GSE post-pandemic. Starting from the reference list of the primary studies (identified in the first phase) published in 2020, we used Google Scholar to find the citations of the related papers from 2020 to 2022. The reason to start from the reference list from 2020 onwards arises from the fact that the central impact of the pandemic began in 2020. As software organizations worldwide had to adopt distributed work models, including remote software development activities, post-2020 studies can offer insights into the potential shifts for software design in GSE. It aims to examine whether the changing dynamic prompted by the pandemic has influenced different practices and perspectives in the field.
4.3. Screening of Papers

Table 2 shows the inclusion and exclusion criteria used in both the study phases.

<table>
<thead>
<tr>
<th>Inclusion criteria</th>
<th>First Phase</th>
<th>Second Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>Only English</td>
<td>Only English</td>
</tr>
<tr>
<td>Publication date</td>
<td>2000-2020</td>
<td>2020-2022</td>
</tr>
<tr>
<td>Publication type</td>
<td>Conference and journal</td>
<td>Conference and journal</td>
</tr>
<tr>
<td>Focus</td>
<td>Studies that have primary emphasis on GSE and discuss methods, processes, tools, or evaluations</td>
<td>Studies that concentrate on software design and architecture within GSE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exclusion criteria</th>
<th>First Phase</th>
<th>Second Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publication type</td>
<td>Panel discussions, conference summaries, position papers, posters, systematic mapping studies, systematic literature reviews</td>
<td>Panel discussions, conference summaries, position papers, posters, systematic mapping studies, systematic literature reviews</td>
</tr>
<tr>
<td>Focus</td>
<td>Studies that have a secondary focus and do not cover GSE methods, processes, tools, or evaluations</td>
<td>Studies that lack emphasis on software design and architecture in the context of GSE</td>
</tr>
</tbody>
</table>

To mitigate potential bias, the second author independently and randomly selected papers; all authors conducted subsequent reviews and uncertain judgments were resolved through consensus. Figure 3 shows the selection process to finalize the primary studies of the study.

![Fig. 3. Primary studies selection process.](image-url)
4.4. Keywording using abstract

We created the classification scheme shown in Table 3 to define the research context, nature, and contribution.

Table 3. Classification scheme.

<table>
<thead>
<tr>
<th>Facet</th>
<th>RQ</th>
<th>Ref.</th>
<th>Categories</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic-independent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pub.year</td>
<td>RQ1</td>
<td></td>
<td>Validation research</td>
<td>Prototype, simulation, tool analysis, experiment</td>
</tr>
<tr>
<td></td>
<td>RQ2</td>
<td>[17]</td>
<td>Evaluation research</td>
<td>Case study, implementation, result, evaluate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Solution proposal</td>
<td>Framework, approach/methodology/tool proposal</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Experience paper</td>
<td>Experience based, experimental study</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Opinion paper</td>
<td>Suggestion, survey</td>
</tr>
<tr>
<td>Stakeholder</td>
<td>RQ3</td>
<td></td>
<td>Software organizations</td>
<td>Project managers, requirement engineers, software developers, software testers, product owners</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Education</td>
<td>Lecturer, instructor, student, undergraduate, graduate, researcher</td>
</tr>
<tr>
<td>Contribution</td>
<td>Map</td>
<td>[77]</td>
<td>Model</td>
<td>Architecture, conceptual model, development process</td>
</tr>
<tr>
<td>type</td>
<td></td>
<td></td>
<td>Method/Framework</td>
<td>Prototype, tool development, implementation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tool</td>
<td>Interview, questionnaire, survey, ethnographic study</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Empirical</td>
<td>Research outcomes</td>
</tr>
<tr>
<td>Contribution</td>
<td></td>
<td></td>
<td>Lesson learned</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Topic-specific</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GSE areas of research</td>
<td>RQ4</td>
<td>[78]</td>
<td>Methods and processes</td>
<td>Open source development, GitHub, DevOps, remote programming, large-scale agile, Scrum, hybrid processes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Emerging technologies</td>
<td>Collaboration tools, simulations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Global business strategy</td>
<td>Business model, employee hiring strategy, transforming organization structure, economic issue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Human aspect of development</td>
<td>Trust and team building, social relationship, language barrier, time-zone difference, cultural difference</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GSE education and training</td>
<td>SE education, student teams, distributed software development, GSE education</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Planning</td>
<td>Project plan, task distribution, task allocation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Requirement Engineering</td>
<td>Requirement elicitation, requirement analysis, user requirement, functional/non-functional requirement</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Design</td>
<td>UML, class/use case/sequence diagram, software architecture</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Development</td>
<td>Project prototype, system development, tool development</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Testing</td>
<td>Unit testing, system maintenance</td>
</tr>
</tbody>
</table>
The selection of facets for the scheme was based on the study’s objective and mapping with the research questions. The classification scheme consists of two categories: topic-independent and topic-specific. In topic-independent classification, we presented four categories: publication year, stakeholder, research type, and contribution type. In topic-specific classification, we presented two categories focusing on GSE areas of research and the software engineering phase. To define areas of focus, we used the Call for Papers of the International Conference on Global Software Engineering (ICGSE 2021) [78], one of the leading conferences in the field.

4.5. Data extraction and mapping process

Based on the classification scheme, we extracted data from the primary studies to understand, analyze, and classify current research in GSE. To this end, we created a template (Table 4) to organize the main features specific to the classification scheme. Then, we highlighted the keywords in the studies to keep track of all the terms and stored them for peer review with the other co-authors. Based on the extracted data, we further analyzed the primary studies to explore possible mappings across different dimensions of our research.

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Unique identifier for the research study</td>
</tr>
<tr>
<td>Title</td>
<td>Title name of the research study</td>
</tr>
<tr>
<td>Digital library</td>
<td>Name of the conference, journal, book, or workshop where the research study was published</td>
</tr>
<tr>
<td>Publication type</td>
<td>Categorization of the research study: conference, journal, position paper, systematic mapping, systematic literature review, panel discussion or conference summary</td>
</tr>
<tr>
<td>Year</td>
<td>Publication year of the research study</td>
</tr>
<tr>
<td>Research type</td>
<td>Categorization of research study w.r.t. type: validation, evaluation, solution proposal, experience report, opinion paper</td>
</tr>
<tr>
<td>Research type (Notes)</td>
<td>Detail of the techniques or methods used to identify research type</td>
</tr>
<tr>
<td>Stakeholder</td>
<td>Identification of the target audience</td>
</tr>
<tr>
<td>Area of focus</td>
<td>Categorization the research study w.r.t. areas of research in GSE: methods and processes, emerging technologies, global business strategy, human aspect, and GSE education and training</td>
</tr>
<tr>
<td>Area of focus (Notes)</td>
<td>Description of the main contribution of the research study</td>
</tr>
<tr>
<td>SE phase focus</td>
<td>Categorization of the research study w.r.t. software engineering phase: planning, requirement, design, development, and testing</td>
</tr>
</tbody>
</table>

5. Results

In the first phase, we retrieved 3317 studies (Figure 4), which were reduced to 580 by inclusion/exclusion criteria. In the second phase, we selected 12 additional primary studies. The resulting 592 papers are the final set analyzed in this study.
5.1. RQ1: What is the temporal distribution of the studies?

The distribution of 592 primary studies extracted in both phases (Figure 5) highlights the increasing interest in GSE-based research studies.

Notably, a substantial surge in research occurred in 2010, making it a trending year for GSE research [80]: new research trends and areas emerged, leading to a shift in publications from agile practices to cloud computing and enterprise social networks. Amidst the pandemic, researchers and practitioners refocused on GSE, driven by the need to adapt to the transitional changes of working in a distributed environment. This shift was crucial for understanding GSE technologies, application tools, infrastructure support, and the human aspect of distributed software development. The results from the second phase complement and corroborate the findings from the first phase, highlighting the limited research available to support
and emphasize the importance of the software design process in GSE.

5.2. RQ2: What are the types of research contribution?

Figure 6 shows the primary studies classified into research types based on the schema proposed by Petersen et al. [17].

![Figure 6. Answer to RQ2: distribution of studies for research type.](image)

In the first phase (i.e., 2000-2020), evaluation research studies emerged as the most prevalent category, constituting 53.8% of GSE’s primary studies. These evaluation studies are mostly based on empirical investigation performed using interviews, questionnaires, and case studies. The second most explored research type is solution proposal, comprising 22.7% of the primary studies. These studies mainly focus on proposing frameworks or methods for different settings. Validation research contributes 12% of the primary studies, comprising the analysis of application prototypes or frameworks to assess the impact on various stakeholders. Furthermore, 10.5% of the primary studies are experience papers describing previous experiences and research evaluations, lessons learned, and guidelines for future research. The remaining 1% consists of opinion papers, which share suggestions and collective opinions of researchers and practitioners on various aspects of GSE.

In the second phase (i.e., 2020-2022), 58.3% of the studies are evaluation research studies, focusing on empirical methods such as case studies, interviews, and expert surveys. Additionally, 33.3% of the primary studies are solution proposals, while only 8.3% are experience papers. No validation research and opinion papers were identified in this study phase. Overall, several changes are evident in the distribution of research types in the two phases of this study. The most prominent shift is the increased emphasis on evaluation research studies in the second phase and solution proposal studies.
5.3. RQ3: Who are the main stakeholders of the studies?

As shown in Figure 7, in 2000-2020, 73.7% of the primary studies focused on software organization-based stakeholders. The focus on software organizations can be explained by the growing trend of remote working, which is widely encouraged in the software industry to facilitate low-cost global development [70]. Furthermore, 16.5% of the studies concentrate on education, exploring project-based training adopted by various software engineering departments in universities. This approach is widely recognized for its emphasis on collaborative environments [81]. However, 9.2% of the primary studies address both stakeholders, suggesting an opportunity for researchers and practitioners to develop collaboration in the field further.

In the second phase (i.e., 2020-2022), 92.1% of the primary studies focus on GSE-based software organizations, reflecting the prominence of this group in the field. These studies explore different aspects of collaboration and development practices for distributed teams. The remaining 8.3% of the studies explore distributed collaboration in education. These studies highlight the adaptation of software engineering education to distributed teams, sharing insights for researchers and practitioners. While the focus on GSE-based software organizations remains consistent, the outbreak of COVID-19 and its impact on global working dynamics emphasized exploring various aspects of collaboration and development practices for distributed teams.
5.4. RQ4: What are the main areas of focus of GSE research?

As shown in Figure 8, in the first phase, 40.3% of the primary studies focus on managing methods and processes in GSE software process management.

Most of these studies primarily explored agile software methodology practices, while other processes include open-source development, DevOps, and lean development. The second most contributed research area, comprising 24.3% of the primary studies, discusses the social and human aspects of working in GSE environments. The research studies address various challenges and provide communication, collaboration, and coordination guidelines within distributed teams. Although the collaboration within distributed teams is facilitated through emerging technologies such as development tools and simulation models, 19.4% of the primary studies are related to emerging technologies that provide the automation and control of the development process for GSE. Moreover, 12% of the primary studies describe various approaches for teaching GSE. The approaches ranged from project-based collaboration between multiple universities to incorporating open-source projects with software organizations. Some studies also explore serious game-based simulation to give students hands-on practice of working in GSE-based software organizations. The remaining 2.9% of the primary studies cover global business strategies. In contrast, a mere 1% of the primary studies combine multiple focus areas, such as introducing distributed scrum teams and analyzing communication challenges in software development activities.

Among the additional papers extracted in the second phase, the 33.3% focus on different approaches to enhance the software design process within distributed teams. To understand the human aspect of distributed development during the soft-
ware design process, 41.6% of the primary studies discuss persistent collaboration challenges distributed teams working in different locations face. Following, 8.3% of the primary study focus on technological support to enhance collaboration for distributed teams during software design. Furthermore, 16.6% of the primary studies combine multiple areas of research. However, no studies address GSE education and training and global business strategies.

5.5. RQ5: What are the software engineering phases considered in the studies?

The primary studies within the mapping study have been classified according to different software engineering phases as defined by Sommerville [79]. The results is shown in Figure 9, in the first phase, 60% of the primary studies discuss the overall development process in distributed environments.

![Fig. 9. Answer to RQ5: distribution of studies for software engineering phase in GSE.](image)

These studies present various methodologies to analyze the experience of distributed teams during software management and development process. 11.5% of the primary studies presents software planning process, discussing various task distribution methods to enhance team collaboration in distributed environments. For software requirement activities, 9.6% of the primary studies highlights the existing challenges in managing requirement discussions within distributed teams. 10.3% of the primary studies focus on the software development describing strategies that enhance participation levels in distributed teams. 2.9% of the primary studies focus on the software design process, particularly on facilitating collaborative design process in distributed teams. The remaining 4.8% of the studies contributes to software testing, specifically software quality monitoring in distributed projects.
5.6. Mapping
This section further examines the key findings to provide additional insights into the relevance of the results concerning the current state of knowledge.

5.6.1. GSE areas of research vs. Research types
Figure 10 shows the map of research areas vs. research type by combining the findings from both phases of this study.

At each intersection, the size of the bubble depends on the total number of papers in the two phases of the study (i.e., 2000-2022). The darker grey shade highlights those bubbles where studies are present in 2020-2022; the light grey shade is used when studies only belong to the first phase. Figure 10 highlights the extensive exploration of evaluation research across all GSE research domains, particularly in methods and processes as well as the human aspect of development. The research studies within this type share various types of empirical analyses to evaluate how different methods affect project management and its impact on the performance of distributed teams. Solution proposal is the second most popular research type for methods and processes and emerging technologies, aiming to enhance the overall experience of working in distributed environments. These research studies emphasize exploring different frameworks to effectively support knowledge management within distributed teams and enhance technological support for distributed software development. These results highlight gaps in GSE education/training and global business strategy. Most studies in GSE education and training focus on theoretical guidelines and practical evaluations of teaching GSE courses. Since training students with the
right soft and technical skills for the GSE environment is crucial, more studies are necessary to provide realistic outcomes. Similarly, the limited research in global business strategy underlines the gap in exploring business strategies for structuring software organizations in distributed environments.

5.6.2. GSE areas of research vs. Software engineering phase focus in GSE

Figure 11 analyzes GSE research areas for software engineering phases to identify the development process focus in a distributed environment.

![Mapping between areas of research and software engineering phases. Color shades: Darker grey for studies in both phases, light grey for first-phase exclusives studies.](image)

The map shows that methods, processes, and human aspects of development are the extensive research areas for the overall software engineering process. The research conducted within these areas provides a broader perspective, addressing advantages and challenges associated with applying different methodologies in the distributed software development process. Research gaps are present in GSE education and training and global business strategy. Furthermore, while focusing on a specific software engineering phase, limited literature is available on the design phase. Unfortunately, due to the infrequent collaboration between academia and industry, there are fewer publications to provide strategies for the distributed collaborative design process among distributed teams. Conversely, studies related to global business strategy predominantly revolve around discussions of organizational change or hiring strategies, which apply to the broader development process. These findings underscore the importance of addressing research gaps and imbalances to promote a more comprehensive and well-rounded exploration of software engineering research across various phases and GSE research areas.
5.6.3. Software engineering phase focus in GSE vs. Research type

To explore the software engineering phase focus in GSE for various research type, we examined the findings of both phases, as shown in Figure 12.

![Fig. 12. Mapping between software engineering phases and research types. Color shades: Darker grey for studies in both phases, light grey for first-phase exclusives studies.](image)

Among the software engineering phases, evaluation research is the most extensively studied, followed by solution proposal research across all phases. This highlights the need for research studies to validate proposed solutions and share field study experiences to benefit researchers and practitioners. Following the results concerning software engineering phases, the design phase emerges as the least explored within various research types. This suggests a potential gap in exploring the design phase for the broader perspective of software engineering research. These observations strongly emphasize the need to obtain a more comprehensive understanding and deeper insight into the details of the design processes within GSE environments.

5.6.4. Software engineering phase focus in GSE vs. Stakeholders

Figure 13 further explores the software engineering phase in GSE for different stakeholders. Software organizations emerge as the most extensively studied among the various stakeholders across all phases. This reflects the importance of understanding software organizational dynamics in shaping software development processes, methodologies, and related outcomes for working in distributed environments. In contrast, education stakeholders are less explored among all phases. This disparity highlights the potential research gap in comprehensively addressing the educational aspect of the GSE environment. Training students to work in a distributed environment is essential as we transition into the post-pandemic phase. Since GSE-based
organizations are increasing, training students with skills tailored for these work environments becomes essential. Furthermore, focusing on software engineering phases studied for all stakeholders, the software design phase is the least explored for all stakeholders. This observation indicates a potential research gap regarding models and practices to manage software design in distributed environments.

Fig. 13. Mapping between software engineering phases and stakeholders. Color shades: Darker grey for studies in both phases, light grey for first-phase exclusives studies.

6. Discussion

This section discusses the results of the study, which show that, following the global pandemic, the software design phase remains the least explored aspect in terms of supporting distributed collaborative design activities within distributed teams. Additionally, the section summarizes potential research gaps based on the responses provided for RQ1-RQ5 and discusses future implications for researchers and practitioners.

6.1. Current Research Gaps in GSE

6.1.1. Paper evolution and stakeholder focus

The reported research studies show a steady increase until 2010; after that, we noticed an unstable rate of publication, which can be attributed to the diverse range of interest areas within the field, resulting in subsequent shifts in research focus - transitioning from agile practices to cloud computing and enterprise social networks. Nevertheless, the collective research studies reported in this period demonstrated emerging trends that have increased interest within the GSE field. Additionally,
the global pandemic helped researchers and practitioners to refocus on GSE, driven by the need to adapt to the demands of a distributed work environment. This adaptation was crucial for understanding various aspects of GSE, including technologies, tools, infrastructure, and the human aspects. The follow-up analysis in the second phase of this study, covering studies published between 2020 and 2022, complemented and validated the findings from the first phase. The results show a consistent increase in research studies with software organization stakeholders. The limited number of publications addressing education stakeholders shows the need to promote GSE-based projects to prepare students for the changing work environment and to bridge the gap between education and the software industry. This becomes particularly relevant after the COVID-19 pandemic, as many software organizations have transitioned towards distributed settings, which has now become the conventional practice in many sectors [82].

6.1.2. Common GSE areas of research trends in software engineering phases

The comprehensive analysis of the five GSE research areas shows that methods and processes are the most explored and prevalent research focus in the GSE field. The human aspects are also extensively studied across the software engineering phases. These areas provide a broader perspective by addressing the challenges and benefits of different methodologies in distributed software development. Our subsequent analysis in the second phase supports this trend, consistently revealing a significant concentration of studies within these research areas. Research gaps exist in GSE education and training and global business strategy. Regarding GSE education and training, studies primarily focus on theoretical guidelines and evaluations, highlighting the need to provide case studies that are applicable in real-world settings. Similarly, limited research in global business strategy underscores the need to explore business strategies for structuring software organizations in distributed environments. Moreover, the design phase needs more substantial literature, indicating a potential gap in understanding this crucial aspect within the broader context of software engineering research.

These findings collectively emphasize the need for balanced research exploration and addressing gaps in GSE education, global business strategy, and the design phase.

6.2. Implication for Researchers and Practitioners

The results presented in this study provide an overview of the prevalent research trends in education and software organizations. In order to address the evolving trends within GSE-based projects, research collaboration is needed between these stakeholders to advance future research in the field. Subsequent potential research directions include:

1. Design tools to facilitate GSE adaptation in educational and software organiza-
tional environments. The results (Figure 11 and Figure 12) highlight a scarcity of tools to support the software development process in GSE environments. For example, few tools support the software design in distributed environments [83]. In order to enhance technological support within distributed teams, it is essential to develop software design tools to support collaboration practices in such settings. As highlighted by Tofan et al. [84], professionals emphasized that team decisions shape over 80% of software design, resulting in an improved software system quality. This underscores the current research gap in effectively managing software design in distributed environments.

2. Publish experience-based studies for prospective research. The results (Figure 10 and Figure 12) emphasize the scarcity of validation-based studies and experience papers across all areas of research within software engineering phases. Given this research gap, publishing more experience-based studies alongside the proposed models, frameworks, or tools is essential. These studies will provide hands-on experiences, enabling a comprehensive understanding of the complexities of the GSE field in a real-time environment.

3. Promote software industry collaboration with the education sector for GSE-based projects. The results illustrated in Figure 13 highlight the limited research available for both stakeholders. In order to address this gap, practitioners are encouraged to promote integrating GSE-based project development activities into educational settings. Activities such as boot camps, hackathons, and joint application development can be introduced in educational curricula. These initiatives are beneficial to train students’ socio-technical skills in real-time settings that extend beyond theoretical learning.

These research implications emphasize the importance of collaboration between the education and industry sectors. By following these recommendations, researchers and practitioners can formulate more structured studies that overcome the current research gaps, ultimately enhancing the related research within the field.

7. Threats to Validity

In this study, we followed Petersen et al.’s [17] evaluation rubric, which includes 26 relevant actions for conducting an SMS. The quality indicator we calculated is the ratio of the number of actions taken compared to the total number of actions. For this mapping study, the ratio stands at 38% (i.e., ten actions), surpassing the median quality (33%) of the studies analyzed in [17]. Additionally, we outlined the guidelines by Wohlin et al. [85] to address potential biases and threats to the validity of the research.

To address the potential threats to internal validity, we followed the strategy outlined in Section 4.2 to characterize the conclusions and validate the data extraction of the study. As discussed in Section 4.2, we defined search terms, libraries, and inclusion-exclusion criteria along with corresponding explanations. We incorporated alternate terms frequently used for the field to address potential search limitations.
and threats. Moreover, we used various digital libraries to ensure a comprehensive and inclusive collection of research studies for the study. Additionally, to minimize personal bias, all authors collectively selected the inclusion-exclusion criteria and only added research studies if they reached the predefined criteria.

We used the classification scheme outlined in Section 4.4 to strengthen construct validity. Each facet within the scheme aligns with the study’s research questions and stated goal. We formulated each research question to achieve the study objective while comprehensively addressing various facets of the study. As a result, we systematically addressed all research questions and the key findings following the established classification schema.

For external validity, we exclusively included research studies published in English (as outlined in Section 4.3). The findings presented in this study concentrate on the five areas of research which hold relevance across GSE research for various stakeholders. This broader scope enhances our findings’ generalization within these specific research areas. However, the omission of potential alternative areas of research introduces a potential challenge to the external validity of the study.

All authors conducted a comprehensive review of the final set of primary studies to avoid any research bias concerning conclusion validity, mitigating the likelihood of bias during the data extraction process. In case of disagreement, the decisions were reached through consensus among all authors. This approach helps protect against biases that could influence the validity of our study conclusions.

8. Conclusion

This paper provides a comprehensive overview of the Global Software Engineering (GSE) research field. The results emphasize that, even in the post-pandemic era, more literature needs to be dedicated to software design activities that effectively support distributed teams. Through the research questions, we have identified the evolution of predominant research areas within education and software organizations. The results highlight the current research focus and address the existing gaps that require attention from practitioners and researchers, shaping the direction of future research. A central theme is software design’s heightened importance in distributed environments. As software teams collaborate across geographical and cultural boundaries, the significance of collaborative software design becomes evident. The results of this study highlight the research gap despite its crucial role in enhancing software quality.

This research holds implications for researchers and practitioners by emphasizing the need for increased collaboration between education and software organizations. Based on this finding, as a way forward, future research can include a) developing tools that address the intricacies of GSE in academic and professional environments, b) giving more space to experience-based studies alongside theoretical models to enrich the understanding of GSE in real-time settings, and c) promoting collaboration between software industry and educational institutions to develop skills that are ap-
applicable beyond theoretical learning.

Finally, this study goes beyond its primary goal of a summarizing effort by pointing to the necessity for innovative methodologies and technologies to enhance the GSE field with greater adaptability and collaboration. By highlighting the crucial role of software design in GSE, we aim to contribute to the growth, improvement, and innovation so that future research can support distributed teams during software engineering phases.

References

[14] Neumann, M., Bogdanov, Y., Lier, M., Baumann, L. ‘The sars-cov-2 pandemic and


[54] Bosnić, I., Čavrak, I., Žagar, M.: ‘Assessing the impact of the distributed software development course on the careers of young software engineers’, ACM Trans Comput Educ, 2019, 18, (2)


2021. pp. 165–170


