

# A Mapping Study About Training in Industry Context in Software Engineering

Breno Alves de Andrade  
Cesar School  
Recife, Brazil  
baa3@cesar.school

Rodrigo Siqueira  
cesar.school, Mesa Inc.  
Recife, Brasil  
rms9@cesar.school

Lidiane C S Gomes  
cesar.school  
Recife, Brazil  
lcsg@cesar.school

Antonio Oliveira  
cesar.school  
Recife, Brazil  
aaspo@cesar.school

Danilo Monteiro Ribeiro  
cesar.school, Zup Innovation  
Recife, Brazil  
dmr@cesar.school

## ABSTRACT

**Context:** Corporate training plays a strategic role in the continuous development of professionals in the software engineering industry. However, there is a lack of systematized understanding of how training initiatives are designed, implemented, and evaluated within this domain. **Objective:** This study aims to map the current state of research on corporate training in software engineering in industry settings, using Eduardo Salas' training framework as an analytical lens. **Method:** A systematic mapping study was conducted involving the selection and analysis of 26 primary studies published in the field. Each study was categorized according to Salas' four key areas: Training Needs Analysis, Antecedent Training Conditions, Training Methods and Instructional Strategies, and Post-Training Conditions. **Results:** The findings show a predominance of studies focusing on Training Methods and Instructional Strategies. Significant gaps were identified in other areas, particularly regarding Job/Task Analysis and Simulation-based Training and Games. Most studies were experience reports, lacking methodological rigor and longitudinal assessment. **Conclusions:** The study offers a structured overview of how corporate training is approached in software engineering, revealing underexplored areas and proposing directions for future research. It contributes to both academic and practical communities by highlighting challenges, methodological trends, and opportunities for designing more effective training programs in industry.

## KEYWORDS

Corporate training, Software engineering education, Systematic mapping, Training methods, Instructional strategies, Salas' framework, Training evaluation, Learning technologies, Human capital development

## 1 Introduction

The acquisition of knowledge, skills, and abilities (KSAs) for present tasks is a key element in enabling individuals to contribute to their organizations and succeed in their current roles [14].

Research on training consistently highlights two key findings: (a) training is effective, and (b) the design, delivery, and implementation of training programs significantly influence their outcomes [33]. In contemporary organizations, where rapid innovation is crucial, continuous learning and skill development have become important. Companies invest in training because they recognize

that a qualified and constantly evolving workforce is a strategic advantage [33].

This need is particularly pressing in the Information Technology (IT) sector. IT firms face growing pressure to build and sustain high-quality human capital, which is essential for maintaining innovation and competitiveness [11]. However, a persistent mismatch between the offerings of educational institutions and the demands of the software industry has led to a global shortage of adequately trained professionals [11]. This gap forces companies to invest heavily in internal training programs to close the skill deficit [11].

Human capital is one of the most strategic assets for any software organization. Decisions about how to develop this capital—whether by upskilling existing staff, hiring new talent, outsourcing tasks, or adopting a hybrid strategy—are central to sustaining performance [6, 45]. Nonetheless, despite the strategic importance of training, education and workforce development are often not core concerns in software engineering companies [45]. The industry is not enthusiastic about reforming educational processes, since training new hires is often seen as less costly than investing in long-term improvements to education [26].

Although numerous studies discuss corporate training initiatives in the software engineering industry, the current body of knowledge remains fragmented. There is no consolidated effort to systematize the area as a whole, nor to provide a broad overview that allows researchers and practitioners to understand its structure, trends, and knowledge gaps. This fragmentation can lead to several consequences: redundant or isolated efforts, low transferability of findings, limited strategic alignment with organizational goals, and missed opportunities to inform policy and theory-building.

To address this gap, this study presents a systematic mapping study that investigates how training has been conducted, evaluated, and reported within the software engineering industry. Using Salas' framework—which organizes training activities into four categories: *Training Needs Analysis*, *Antecedent Training Conditions*, *Training Methods and Instructional Strategies*, and *Post-Training Conditions*—we analyzed 26 primary studies to identify trends, gaps, and opportunities.

This study contributes by offering an organized perspective on industrial training in software engineering and by highlighting underexplored areas that may benefit from future investigation. It provides researchers and practitioners with actionable insights to

design training programs that are better aligned with organizational strategy and professional development goals.

The remainder of this paper is structured as follows. Section 2 provides the background and related work. Section 3 describes the methodology and procedures used in this study. Section 4 presents the results, including answers to the research questions. Section 5 discusses the findings, their implications, and Section 6 the limitations. Finally, Section 7 concludes the paper and suggests areas for future research.

## 2 Background

### 2.1 Definitions

The terms **training**, **education**, **development**, and **learning** are frequently used interchangeably, yet each has a distinct meaning within the human resources domain. Clarifying these concepts is essential for understanding their role in corporate training and workforce development.

**Training** refers to practical, task-oriented instruction aimed at improving performance in current roles. It is typically hands-on and focuses on the transmission of knowledge, skills, and abilities (KSAs) required for job-related tasks [22]. According to Fitzgerald, training is “(a) the acquisition of knowledge and skill for present tasks, (b) a tool to help individuals contribute to the organization and be successful in their current positions, and (c) a means to an end” [14]. Effective training must be linked to performance outcomes and result in behavioral changes observable in the workplace [14].

**Education**, in contrast, is conceptually broader and more reflective. It emphasizes understanding, critical thinking, and theoretical knowledge rather than direct application. Education often takes place in formal academic settings and is influenced by lifelong social interactions that help individuals adapt to dominant norms and values [12, 22].

**Development** encompasses both training and education and adopts a long-term perspective. It aims to prepare individuals for future roles by enhancing their KSAs through a continuous process involving learning, reflection, and practical experience [14]. Development initiatives often align with career progression strategies and organizational sustainability goals.

Despite their differences, training, education, and development share a central concern with learning [15]. According to the Cambridge Dictionary, **learning** is the acquisition of knowledge through study or experience and is characterized by behavioral change resulting from external stimuli [15]. Clear distinctions among these terms support more effective human resource strategies and organizational planning [22].

In this context, **competencies** refer to the integrated set of KSAs required for effective job performance. Parry defines competencies using three foundational components [27]: **Knowledge**: Conceptual and procedural understanding, the *know-how*. **Skills**: Practical capabilities developed through experience, the *can-do*. **Abilities**: Innate or developed traits enabling performance, the *capable of doing*.

Understanding education and its underlying purpose requires analyzing the broader sociocultural and organizational context in which educational practices and discourses are situated [13].

**Table 1: Salas' Framework**

Category	Description
Training Needs Analysis	This first step in training development focuses on the process of deciding who and what should be trained.
Antecedent Training Conditions	Individual factors before training, such as cognitive ability, self-efficacy, and goal orientation, have been particularly influential in shaping behavior and outcomes.
Training Methods and Instructional Strategies	It incorporate a combination of tools, methods, and content that together form a structured instructional approach.
Post-Training Conditions	Events that occur to evaluate training, and on examining the events that ensure transfer and application of newly acquired knowledge, skills, and abilities (KSAs).

**Corporate education** has emerged as a strategic response to the growing mismatch between higher education outcomes and industry demands. It addresses the rapid obsolescence of knowledge and the need for continuous workforce qualification [23]. It is defined as an internal training system aligned with organizational goals, structured to support managers and specialists at all levels [23].

In summary, training and development activities benefit both employees, by improving career trajectories, and organizations, by increasing operational efficiency and resilience [37].

Because training is a systemic process involving pre-training, delivery, and post-training phases, this study adopts Salas' framework to organize the literature review. This model offers a structured approach for categorizing training practices and outcomes, helping to align analysis with the realities of the software engineering industry [28].

### 2.2 Salas' Framework

The framework proposed by Salas offers a comprehensive perspective on organizational training, drawing from extensive research conducted from the early 1990s to the early 2000s. It emphasizes advances in training theory, instructional methods, and the critical role of post-training activities. The model highlights the influence of cognitive, motivational, and organizational factors, as well as the growing integration of technological tools in training delivery [32].

Table 1 provides a brief overview of the four main components of Salas' framework. Each of these dimensions will be detailed in the following subsections.

**2.2.1 Training Needs Analysis.** The first phase of training development involves identifying the essential elements required for successful intervention. This includes determining who needs training, what should be taught, and how the training aligns with organizational goals [33]. Training Needs Analysis (TNA) plays a crucial role in optimizing resource allocation and maximizing training effectiveness.

**Organizational Analysis.** This dimension focuses on the alignment between training goals and broader organizational factors such as strategic objectives, available resources, constraints, and support for learning transfer [32].

*Job/Task Analysis.* This analysis identifies the specific KSAs required to perform particular tasks effectively, considering the expected outcomes and role-specific demands [32].

**2.2.2 Antecedent Training Conditions.** Antecedent conditions refer to the factors present before training that influence its success. These include individual characteristics and contextual variables that shape learners' readiness and engagement. Such conditions are as critical as the training itself in determining outcomes [32].

*Individual Characteristics.* Trainees bring to the learning environment distinct traits such as cognitive ability, self-efficacy, and goal orientation—all of which significantly influence training engagement and learning outcomes [32].

*Training Motivation.* Motivation to learn is shaped by both individual variables (e.g., age, personality) and organizational factors (e.g., climate, support). Higher motivation leads to greater knowledge retention and behavioral change [33].

*Training Induction and Pretraining Environment.* Pre-training strategies, such as advanced organizers, structured preparation, and cognitive scaffolding, are essential to optimize learning conditions and ensure that trainees are equipped to absorb and apply new information effectively [32].

**2.2.3 Training Methods and Instructional Strategies.** Instructional strategies encompass the design and application of tools, content, and delivery techniques to improve learning retention and job performance[32].

*Specific Learning Approaches.* These include pedagogical techniques such as feedback loops, distributed practice, and reinforcement, which are used to increase the retention and transfer of learning[32].

*Learning Technologies and Distance Training.* Training approaches increasingly leverage digital technologies—such as video conferencing, e-learning platforms, and online courseware, to enable flexible and scalable delivery [32].

*Simulation-Based Training and Games.* Simulations replicate real-world environments and tasks, offering immersive learning experiences that foster deep engagement, reduce error rates, and improve performance [32].

*Team Training.* Team-based training focuses on building collaborative skills, fostering communication, and enhancing group performance. It prepares individuals to apply learned skills in various team contexts, improving both individual and collective effectiveness.

**2.2.4 Post-Training.** Post-training conditions are vital to ensuring that acquired KSAs are retained and applied in the workplace. The long-term impact of training is strongly linked to the follow-up and evaluation processes [32].

*Training Evaluation.* Evaluation involves measuring the effectiveness of training through behavioral, cognitive, and affective indicators. Well-defined objectives and assessment criteria ensure alignment with performance goals [33].

*Transfer of Training.* Training transfer refers to the application, generalization, and maintenance of KSAs in the work context. It is a core objective of any training initiative and directly impacts organizational effectiveness [32].

## 2.3 Training in the Software Engineering Industry

Technology companies face increasing pressure to develop and sustain high-quality human capital in order to maintain innovation and competitiveness [11]. Factors such as software quality, delivery speed, and cost efficiency have become central to market differentiation [31]. In this context, continuous training is a strategic necessity, requiring dedicated structures aligned with ongoing organizational learning efforts.

**Software engineering** as a discipline exhibits characteristics that distinguish it from other engineering fields, particularly in the context of workforce development:

- (1) **Software engineering is an experimental discipline:** Knowledge must be continuously refined through real-world development experiences, which are then translated into educational and training programs [31].
- (2) **Software is developed, not manufactured:** Success depends heavily on human expertise and creativity, making education and training essential for adopting new technologies [31].
- (3) **Each software product is unique:** Training programs must be contextualized to reflect the specific challenges, goals, and technological environments of the organization [31].

The integration of training, education, development, and learning is therefore particularly relevant in software engineering. The human-centered, evolving, and context-specific nature of software development demands tailored strategies for capacity building. As organizations increasingly recognize the link between human capital strategies [11] and product outcomes [31], aligning training approaches with strategic goals becomes critical for fostering innovation and sustainable performance [14].

For the purposes of this study, we adopt the definition of **corporate education** as a long-term organizational strategy focused on the continuous development of KSAs through a structured set of training initiatives aligned with business goals. In this perspective, training addresses operational competencies (*what to do*), training programs define tactical approaches (*how to do it*), and corporate education sets strategic direction (*where to go*). Together, they form an integrated framework essential for building and sustaining a competitive workforce. demonstrate

## 2.4 Related Work

This underscores the ongoing challenge of translating training into workplace performance an issue similarly relevant in software engineering, where collaboration and coordination are critical, yet the impact of training remains insufficiently studied. Some research highlights the importance of training in the software industry. The relationship between professional training, human capital development, and the challenges of the software industry has been addressed from different perspectives over the past years. Arora and

Athreye[2] highlight the importance of the software industry in fostering human capital and increasing productivity within the Indian economy. According to the authors, the sector not only contributes to job creation and exports but also drives innovative organizational models and private investments in training, directly influencing corporate governance and professional qualification.

Kulkarni et al. [21] provide an empirical perspective on the initial training offered by Indian technology companies to recent graduates. The study reveals a misalignment between traditional academic education and industry expectations, which leads companies to invest heavily in their own training programs. In addition to technical skills, corporate training programs also address behavioral competencies and cultural aspects relevant to working in distributed teams.

More recently, Chethana and Noronha [9] analyzed training structures using diverse sources of gray literature—including Google Scholar publications, symposium materials, financial reports, official websites, news portals, and social media testimonials—focusing on training programs in Indian companies from 2012 to 2021. Although methodologically distinct from our systematic approach, their proposed framework aligns closely with Salas' foundational training model [32], comprising:

- (1) Needs assessment for training and development,
- (2) Clear objective setting,
- (3) Methodology selection and detailed planning,
- (4) Program implementation,
- (5) Results analysis, and
- (6) Ongoing monitoring.

Salas et al. (2006) [34] reviewed 28 studies on Crew Resource Management (CRM), a structured training model originally developed in aviation and later adopted in other high-risk domains such as medicine, offshore oil, and nuclear power. CRM focuses on enhancing non-technical skills—including communication, teamwork, and decision-making—to improve safety and performance. Using Kirkpatrick's evaluation framework Kirkpatrick et al. [20] found that while trainees generally responded positively to CRM training, the results regarding learning outcomes, behavioral changes, and organizational impact were mixed.

Complementing this study, systematic reviews by Garousi et al. [16] and Cico et al. [10] establish the context for corporate training by respectively identifying critical skill gaps and the adoption of modern industry trends in academia. While these studies focus on what is needed or taught, our work differs by investigating how the industry itself structures and implements its own training programs.

These related works form our "golden set", serving both as foundational references for constructing our search strings and as primary sources for snowballing procedures.

### 3 Method

In this section, we present an adapted and applied **systematic mapping study** in software engineering, based on the guideline [28]. The study was carried out in five main steps: (1) Definition of Research Questions, (2) conducting the Search, (3) Screening of Papers, (4) Keywording using Abstracts, (5) Data Extraction and Mapping Process.

### 3.1 Research Questions

The primary goal of this systematic mapping study is to provide an overview of what the scientific literature are talking about the Training in Industry Context in Software Engineering and identify research and results available of the approaches and impacts of training of competency in the professional for software engineering research and results available. Additionally, the study aims to identify with areas of Salas' framework the article are located, the main challenges of the area map publication frequencies over time to observe trends and identify the forums where research in the area has been published. These objectives are reflected in the following research questions, which guide this study. The rationale for each question is provided after they are listed below.

- **RQ1** - What evidence does the scientific literature provide about of training for software engineering professionals in industry contexts?
  - **RQ1.1**- How the articles were categorized according to the research areas defined in Salas' framework? (4.1)
  - **RQ1.2** - Which variables are the authors analyzing for the development of the training? (4.2)
  - **RQ1.3** - What are the objectives of the trainings? (4.3)

**RQ1** is the primary goal of this systematic mapping study is to identify evidence-based practices and critical gaps in corporate training for software engineering professionals. This overarching question guides our investigation of training approaches, their impacts, and implementation challenges.

**RQ1.1** aims to provide an overview by categorizing the articles according to their research areas. This helps to clarify the distribution of studies, highlighting both well-explored and under-represented areas within the literature.

**RQ1.2** focuses on identifying the variables analyzed by the authors during the development of training. Understanding these variables is essential to grasp how training programs are structured and what factors are considered critical for their success.

**RQ1.3** investigates the main objectives of the training initiatives described in the literature. This allows us to understand the intended outcomes of such programs and how they align with industry needs.

### 3.2 Conduct Search for Primary Studies

Our primary studies were identified using search strings in automatic databases or by manually searching [28] and also snowballing our references. Our research was conducted using two authoritative software engineering digital libraries: ACM and IEEE. We selected them because ACM is widely used in the field of computer science, IEEE is a leading applied engineering repository. Although Scopus and Springer were initially considered, they were excluded from the automated search due to institutional access constraints.

We acknowledge that this exclusion may have impacted our coverage. To mitigate this potential limitation, we supplemented our search with manual methods, including forward snowballing, as described by Petersen et al. [28]. While this process did identify articles in Scopus and Springer, they were ultimately rejected because they did not meet our inclusion criteria.

We used a "golden set" of articles as a reference to build our search string. These articles, which we considered representative of our topic, address corporate training in software engineering

and helped us identify relevant keywords and synonyms. For instance, terms such as “corporate training,” “in-house training,” and “software engineering education” were extracted from this set. Additionally, the golden set supported our snowballing phase, helping to ensure that relevant studies not captured by the initial search were also retrieved.

Our automated search strategy targets three core domains: training, software engineering, and industry applications. To ensure comprehensive coverage, we applied the OR operator to include synonyms of the key terms and the AND operator to ensure that the concepts and context were combined effectively.

The search string was designed to query only the **title** and **abstract** fields in two digital library services: IEEE Xplore Digital Library and ACM Digital Library.

We used the following synonyms to build the query for **training**: corporate training, corporate education, on-the-job training, in-house training, job training, professional training, training in companies, employee education, professional development, training program, educational program, IT training, personnel development, institutional education, bootcamp, corporate university, and development of human capital.

For **software engineering**, we included: software development, IT, and software engineering.

For **industry**, we considered: industry, IT firms, IT industry, corporate, organizations, and workplace.

This query is designed to include topics related to corporate education, software engineering, and industry. These terms provide a comprehensive coverage of key areas such as workforce training, software development, and industrial applications, making sure that the search results are relevant and aligned with these important fields.

The process was conducted in four main stages: **Automated Search Strategy**, **Primary Study Selection**, **Full Read** and **Snowballing**. The automated search yielded 574 articles from ACM and 1,571 from IEEE, totaling 2,145 articles. In all selection stages, two researchers analyzed the articles individually and compared the results at the end. A third researcher, with more experience in systematic reviews and education in Software Engineering, resolved conflicts regarding the selected articles, ensuring a general agreement between both researchers. In the first stage, the selection was based on the title and abstract. In the next stage, the analysis included the previous fields complemented by the conclusion, reducing the number to 126 articles selected.

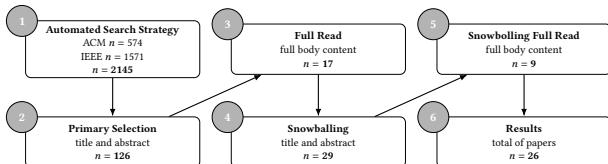


Figure 1: Phases and study counts

In all selection stages, two researchers independently analyzed the articles and compared results, with a third experienced researcher resolving conflicts to ensure consensus. The first stage filtered articles based on titles and abstracts. The second stage

added conclusions to the evaluation criteria, reducing the set to 126 articles.

In the third phase, we performed a full-text reading and selected 12 articles that met all the criteria. In the fourth phase, the snowballing technique was applied, resulting in the selection of 5 additional relevant articles from the 12 analyzed. Additionally, 4 more articles were included from our references, totaling 9 articles obtained through snowballing.

After completing all stages, the systematic mapping resulted in **26 studies** (12 from full-text analysis + 9 from snowballing), as illustrated in Figure 1 and Figure 2.

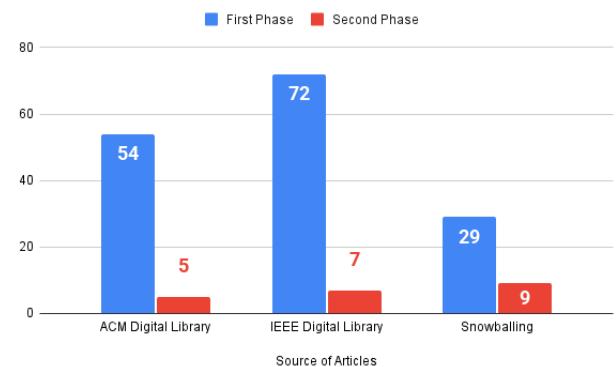


Figure 2: Conduct Search

### 3.3 Screening of Papers for Inclusion and Exclusion

Inclusion and exclusion criteria were used to exclude studies that were not relevant to answering the research questions [28].

#### Inclusion Criteria:

- Primary studies from the software engineering industry focusing on corporate education, which propose analyses, frameworks, methods, training approaches, or evaluations.

#### Exclusion Criteria:

- Articles not published in peer-reviewed venues.
- Articles in languages other than English or Portuguese.<sup>1</sup>
- Articles related to IT support.
- Articles not containing relevant terms in the title and abstract.
- Articles not directly related to corporate education applied in the software engineering industry.
- Articles that were not implemented or applied in practice.
- Articles conducted exclusively with academic students.
- Articles involving both students and professionals in which it is not possible to isolate the results related to professionals
- Articles that address the IT field in a generic way, alongside other areas.

The inclusion and exclusion criteria were defined to ensure the relevance and the quality of the selected sample. Only studies from

<sup>1</sup>Portuguese is the mother language of the researchers.

the software engineering industry with an explicit focus on training were included, as the main goal of this research is to understand how training is designed, implemented, and evaluated within the specific context of the software engineering industry. These methodological decisions aimed to ensure that the final set of studies was representative, coherent with the research objectives.

### 3.4 Data Extraction and Mapping of Studies

For the data extraction, we used a Google Sheet to document the process. This spreadsheet contained two tabs with identical columns, allowing each researcher to enter data from the selected papers independently. A third tab, named 'Compilation,' was then created to merge the data from both researchers, incorporating resolved conflicts and the consensus reached with an advisor acting as a mediator. Our analysis followed a structured process combining deductive coding (based on thematic analysis) and narrative synthesis. When completing the extraction form for each paper, researchers provided a brief rationale and evidence explaining its classification under one of Salas' areas. This process was guided by a data extraction form designed specifically to gather the necessary information to answer our research questions..

## 4 Results

This section presents the findings of the systematic mapping study. **A total of 26 primary studies** focused on corporate education in software engineering, specifically those conducted and implemented by industry, were selected and analyzed. The complete list of selected articles and their identifiers is presented in Table 2.

To identify temporal trends in this research domain, the selected articles were also organized by year of publication, as illustrated in Figure 3. The earliest study dates back to **1984**. A modest rise in publication volume were observed in the years **2008, 2011, 2018, 2023 and 2024**, while the other years showed low or no activity in this topic area. Despite its relevance to industry, the topic has received little systematic investigation in the academic literature.

To structure the analysis, each study was classified according to the categories defined in Salas' framework for corporate training: **Training Needs Analysis, Antecedent Training Conditions, Training Methods and Instructional Strategies, and Post-Training Conditions**. The distribution of studies these categories is shown in Figure 4.

### 4.1 RQ1.1 - How the articles were categorized according to the research areas defined in Salas' framework?

To classify the selected studies within Salas' framework, each training initiative was analyzed according to the specific subareas that comprise the four main categories. This classification was based on the evidence reported in each article—such as objectives, methods, and implementation strategies—indicating alignment with one or more subareas. Table 3 presents the mapping between the reviewed studies and the corresponding areas of Salas' framework.

**4.1.1 Training Needs Analysis.** Three studies were classified under the subarea **Organizational Analysis**, as they emphasize

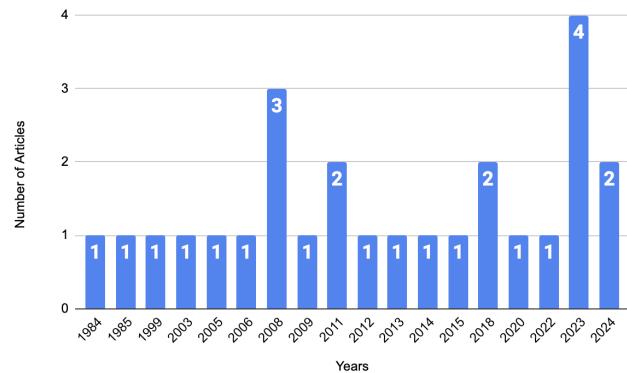


Figure 3: Distribution of articles by year of publication

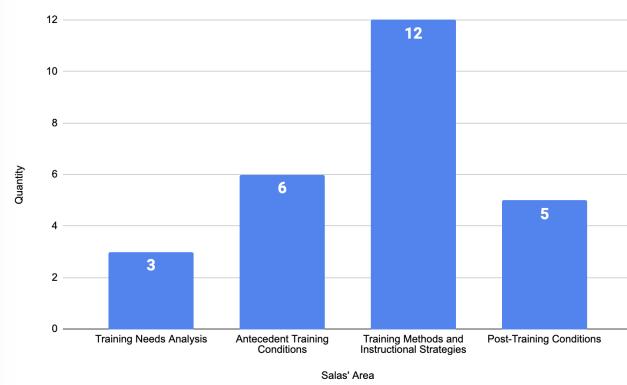


Figure 4: Distribution of Training Research Across Salas' Framework

strategic planning to either optimize existing training practices or design new programs aligned with organizational needs:

Ruhe [31] applies an organizational analysis approach using the Goal-Question-Metrics (GQM) method to assess the effectiveness of training programs.

Prikladnicki and Pilatti [29] proposes a strategic plan for implementing training initiatives that address the global needs of a software company.

Ulema and Larsen [42] presents a corporate education plan focused on software engineering, highlighting the need to adapt training content to organizational contexts.

**4.1.2 Antecedent Training Conditions.** We identified works in three subareas: **Individual Characteristics, Training Motivation and Training Inductions and Pretraining Environment**. In the **Individual Characteristics**.

Wingreen and Blanton [44] investigates behavioral and perceptual differences among IT professionals, particularly focusing on the concept of Person–Organization (P–O) fit. The study argues that even when professionals possess similar technical skills, experience, and compensation, their perception of alignment with organizational values may vary significantly. These perceptions play a central role in shaping training motivation and engagement,

**Table 2: Selected articles and their identifiers**

Ref.	Title
[7]	A blended approach to course design and pedagogy to impart soft skills: An IT company's experiences from software engineering course
[43]	A Q-Methodological Study of IT Professionals' Person-Organization Fit for Training and Development
[24]	Aligning workforce development software process improvement strategy for accelerated adoption of software engineering capability
[41]	Analysis of the Motivation of Learners in the In-House Training of Programming in Japanese ICT Industries
[45]	Assessing the IT Training and Development Climate: An Application of the Q-methodology
[38]	Corporative Education Used to Attract and Qualify Workforce for the Software Industry
[18]	Deploying an online software engineering education program in a globally distributed organization
[47]	Design and Evaluation of an Agile Framework for Continuous Education in Software Engineering
[36]	Design using software engineering principles: overview of an educational program
[1]	Employees Perception towards Training in IT Sector
[19]	Employees' training and development in IT sector: An essential instrument for effectiveness of organizational productivity
[31]	Experience Factory-based professional education and training
[5]	Experience with Training a Remotely Located Performance Test Team in a Quasi-agile Global Environment
[3]	Human Capital Investments and Employee Performance: An Analysis of IT Services Industry
[29]	Improving Contextual Skills in Global Software Engineering: A Corporate Training Experience
[48]	Integrating Cloud-Based AI in Software Engineers' Professional Training and Development
[44]	IT professionals' person–organization fit with IT training and development priorities
[17]	Nurturing Talent: The IT Academy Journey into Quality Development
[40]	Personalizes training for success implementation of advanced technologies in the IT industry
[42]	Planning for in-house software engineering education
[50]	Professional Development of IT Industry Specialists at the Workplace: Trends, Focus and Prospects
[25]	Requirements Analysis Skills: How to Train Practitioners?
[39]	Role of leadership in enhancing the effectiveness of training practices: Case of Indian information technology sector organizations
[35]	Synchronous training in distributed software development team
[4]	The Evaluation of a Requirements Engineering Training Program at Siemens
[30]	Understanding Self-Efficacy in Software Engineering Industry: An Interview Study

**Table 3: Mapping between Salas' Areas and Articles**

Area	Sub-Area	Articles
Training Needs Analysis	Organizational Analysis	[29, 31, 42]
	Job/Task Analysis	-
Antecedent Training Conditions	Individual Characteristics	[43, 44]
	Training Motivation	[41]
	Training Induction and Pretraining Environment	[39, 40, 45]
Training Methods and Instructional Strategies	Specific Learning Approaches	[17, 24, 25, 36, 47, 48]
	Learning Technologies and Distance Training	[5, 18, 35, 50]
	Simulation-Based Training and Games	-
	Team Training	[7, 38]
	Training Evaluation	[1, 3, 19, 30]
Post-Training Conditions	Transfer of Training	[4]

underscoring the importance of individual characteristics in training outcomes.

Wingreen and Blanton [43] The seven person–organization (P–O) fit perspectives revealed by this study demonstrate that IT professionals hold personal beliefs about their alignment with IT training and development programs that span multiple dimensions – across areas such as content, venue, and resources – rather than representing a unitary and one-dimensional concept.

In the **Training Motivation** Tsukamoto et al. [41] investigates with the motivational attitude was compared across different groups based on criteria such as gender, academic background, and prior programming experience.

In the **Training Inductions and Pretraining Environment** Sinha and Sengupta [39] investigates how leadership styles, transformational and transactional, influence the effectiveness of training practices in the IT sector in India.

Stanojeska [40] shows that personalized training programs significantly increase employee engagement and the effectiveness of new technology implementation, strengthening organizational performance and competitiveness.

Wingreen et al. [45] investigates with the Q-methodology to develop a tool for the assessment of the IT training and development climate, identify and interpret natural groups of IT professionals according to their own professional development priorities to be used as a baseline for managerial decision making, and make recommendations for its use as a managerial decision tool.

**4.1.3 Training Methods and Instructional Strategies.** The selected articles were classified into three subareas: **Specific Learning Approaches**, **Learning Technologies and Distance Training**, and **Team Training**. No articles were associated with the **Simulation-Based Training and Games** subarea.

For **Specific Learning Approaches**, a variety of strategies were identified. Wolfschwenger et al. [47] employed Problem-Based Learning (PBL) to promote learning through real-world problem-solving activities related to DevOps practices. Schaul [36] described a workshop model focused on applying mathematical models for functions and data abstractions in software programs. Wolfschwenger et al. [48] adopted a hands-on approach using Artificial Intelligence to deliver personalized feedback, interactivity, and adaptability to individual learning styles. Morales-Ramirez and Alva-Martinez [25] implemented a mentoring-based training plan in Requirements Engineering to help professionals acquire practical knowledge in the field. Goelzer et al. [17] emphasized hands-on learning within software quality training, integrating practices like BDD and agile methodologies. Mason [24] utilized a class combination to training professionals in software engineering offering flexibility distance training.

In the **Learning Technologies and Distance Training** sub-area, the following approaches were observed Hudepohl et al. [18] utilized video content, e-learning platforms, and webinars to train globally distributed software teams. Bondi and Ros [5] conducted remote training sessions for a performance testing team located across different regions. Samuel et al. [35] explored both synchronous and asynchronous training formats for distributed teams, highlighting how communication strategies supported effective knowledge acquisition. Zubareva et al. [50] explored Smart educational environment involves the integration of innovative concepts, intelligent hardware and software, smart classrooms equipped with

the latest technologies, and educational processes based on modern and intelligent teaching and learning strategies.

In the **Team Training** subarea, Boyeena and Goteti [7] developed group-based training programs in which participants were evaluated on both technical and soft skills. Silva et al. [38] provided training as part of a recruitment and retention process for new employees in software engineering roles.

**4.1.4 Post-Training Conditions.** This area includes two subareas: **Training Evaluation** and **Transfer of Training**.

In The **Training Evaluation**, focusing on individual development and organizational outcomes. Bapna et al. [3] demonstrated that corporate training programs can improve individual performance. Agarwal et al. [1] The study shows that there are seven major factors which affect the perception of employees towards training in the IT sector. These factors include Training Duration, Training Method, Training Impact, Training Outcome, Job Performance, Training benefit and Training Effectiveness. Jan and Muthuvelayutham [19] The research through Structural Equation Modeling (SEM) using AMOS infers that the good training programmes will increase the productivity of the organization. Ribeiro et al. [30] found that bootcamp-style training improved team self-efficacy, enhanced deliverable quality, reduced lead times, and encouraged knowledge sharing.

The **Transfer of Training** subarea was addressed by Berenbach and Rayment [4], which analyzed the practical application of training content in the participants' daily work activities.

## 4.2 RQ1.2 - Which variables are the authors analyzing for the development of the training?

Three main variables were identified across the studies: training duration, subjects covered, and methods or tools used to support the training.

**Training Duration:** The length of the training programs varied significantly across studies. For example, Berenbach and Rayment [4] reported a short-term course of three days, Boyeena and Goteti [7] presented a five-day program, Schaul [36] described a two-week training, and Silva et al. [38] reported a six-month program. This variation may reflect differences in pedagogical goals, participant profiles, and organizational contexts.

**Subjects Covered:** The content of the training programs often focused on core software engineering areas. Requirements Engineering appeared in Boyeena and Goteti [7], Hudepohl et al. [18], and Berenbach and Rayment [4], while Software Testing was featured in Boyeena and Goteti [7], Hudepohl et al. [18], and Bondi and Ros [5].

**Methods and Tools:** Various tools and approaches were reported to support training. In the **Training Needs Analysis** category, Ruhe [31] used the Goal-Question-Metrics (GQM) method, and Prikladnicki and Pilatti [29], Ulema and Larsen [42] employed structured interviews to capture stakeholder expectations. In the **Antecedent Training Conditions** category, Wingreen and Blanton [43, 44] used Q Methodology to explore person-organization fit. In the **Training Methods** category, workshops Schaul [36], PBL Wolfschwenger et al. [47], hands-on activities Bondi and Ros

[5], Goelzer et al. [17], Wolfschwenger et al. [48], and e-learning platforms Boyeena and Goteti [7], Hudepohl et al. [18], Morales-Ramirez and Alva-Martinez [25], Samuel et al. [35], Silva et al. [38] were frequently used. For **Post-Training Evaluation**, Berenbach and Rayment [4] employed the Kirkpatrick Model, Bapna et al. [3] used the PCMM method, and Ribeiro et al. [30] conducted interviews.

## 4.3 RQ1.3 - What are the objectives of the trainings?

The analyzed papers reveal study groups with common themes, although directed toward different goals faced by the industry. Below are the objectives of the studies reviewed.

**Software Engineering Practices:** The paper Boyeena and Goteti [7] aims to align corporate training with organizational demands, focusing on the technical and behavioral development of new talents, based on the 5 main areas of the SWEBOK.

**Distributed Training:** The article Hudepohl et al. [18] address to implement an effective online education program for software engineers in a globally distributed organization. The article Bondi and Ros [5] focuses on conducting a testing training program with two distributed teams—one located in India and the other in New Jersey, United States. The main challenges of this training were cultural differences and time zone discrepancies. The article Prikladnicki and Pilatti [29] reports on a training experience carried out in a multinational company to develop the technical team's skills and highlights the same challenges mentioned in Bondi and Ros [5]. Finally, the article Samuel et al. [35] presents the use of synchronous and asynchronous training formats to support geographically distributed teams.

**DevOps and Cloud:** The article Wolfschwenger et al. [47] focuses to apply an agile, hands-on framework to promote the continuous education of software engineers, with a focus on DevOps and cloud computing.

**Use of Artificial Intelligence:** The article Wolfschwenger et al. [48] aims to apply generative AI to enhance productivity and quality during training for software developers.

**Requirements Engineering:** The article Morales-Ramirez and Alva-Martinez [25] goals to develop a training plan for educating professionals in the discipline of Requirements Engineering.

**Software Quality:** The article Goelzer et al. [17] objectives to develop a quality mindset in professionals participating in the training, emphasizing the implementation of quality best practices from the early stages of the software development process. In turn, the article Schaul [36] focuses on the need to introduce software engineering principles and the application of mathematical models in the design of training programs to reduce defects and costs, while improving software quality.

## 5 Discussion

### 5.1 General

The analysis reveals that studies in the software engineering industry address all four categories proposed by Eduardo Salas as essential for corporate training. However, not all subareas within each category are equally represented. This uneven distribution highlights both the maturity of certain research directions—such

as **Training Methods and Instructional Strategies**—and the absence of empirical work in other foundational subareas.

Specifically, in the **Training Needs Analysis** category, only the subarea *Organizational Analysis* was addressed [29, 31, 42], while *Job/Task Analysis* was not explored in any of the selected studies. The lack of focus on job/task-level needs may compromise the relevance and precision of training initiatives, since identifying specific competencies required for job performance is a key starting point for effective program design.

In the **Antecedent Training Conditions**, it was observed that some subareas were addressed by only a few studies. Individual Characteristics were explored in two articles [43, 44], while Training Motivation was examined in only two studies [41, 45]. The subarea Pretraining Environment and Induction was covered in two other studies [39, 40]. Despite the occasional presence of these investigations, the scarcity of research in these subareas is concerning, as these factors directly influence learner readiness, engagement, and learning effectiveness. Overlooking such conditions may result in training programs that are misaligned with the participants' actual needs, thereby compromising knowledge transfer and the intended organizational impact.

For **Training Methods and Instructional Strategies**, all subareas except *Simulation-Based Training and Games* were addressed. The use of simulations can provide realistic and controlled learning environments that are especially useful for complex, high-stakes contexts—yet no study in the review employed or evaluated this method. This omission suggests an opportunity for future work to investigate the applicability and impact of simulation-based training in software engineering contexts.

In the **Post-Training Conditions** category, both subareas were covered, but to varying extents. *Training Evaluation* was addressed by three studies [1, 3, 19, 30], while *Transfer of Training* appeared in only one [4]. This limited attention to transfer raises concerns, as the real value of training lies in its application to daily work. Without examining how new knowledge and skills are transferred to the job context, it is difficult to assess whether training achieves its intended impact.

Compared to other domains—such as healthcare, manufacturing, and Industry 4.0—where corporate training has been more extensively studied and systematized [8], the field of software engineering still lacks a cohesive body of literature on this topic. This highlights the novelty and importance of the present study in offering a first structured overview based on an established training framework.

Moreover, the scarcity of longitudinal assessments is notable. Most reviewed studies emphasize immediate outcomes rather than long-term effects on performance, behavior, or retention. This absence reinforces the need for future research to adopt longitudinal designs that can better capture training impact over time.

The lack of studies exploring training aligned with individual learning needs and the reliance on anecdotal evidence also suggest that the academic field is still in an early stage of development. It may be premature to speak of a consolidated research tradition on corporate training in software engineering. Therefore, future studies might benefit from adapting or extending existing frameworks—such as Salas’—to build context-specific models that reflect the nuances of software development environments.

A possible explanation for the limited number of studies found in this mapping is the lack of publication of empirical research by industry. Companies often develop and implement internal training programs but do not share these initiatives with the academic community. This behavior may be driven by factors such as strategic information protection, market competitiveness, or even the absence of incentives or partnerships with research institutions. As a result of this lack of publication, there is little contribution to the consolidation of scientific knowledge on training in Software Engineering. This limitation directly impacts the ability to generalize findings and hinders the advancement of the field, which lacks more robust empirical studies connected to real-world industry practices. Therefore, it is essential to strengthen the connection between academia and industry, fostering collaborations that enable the documentation and critical analysis of training practices that are already in use but remain invisible in the scientific literature.

Furthermore, since companies rarely disclose training outcomes publicly, an alternative approach could involve extracting relevant data from annual and financial reports to obtain more primary-level insights - mirroring Chethana's successful approach of analyzing alternative sources [9] through diverse gray literature. This suggests that exploring supplementary data channels may offer valuable pathways to bridge the current research gap in corporate training studies.

Another relevant point observed in the selected studies is the absence of reports on negative experiences or failed training initiatives. This lack of critical reflection limits the field's maturity, as it prevents researchers and practitioners from learning from challenges, design flaws, or contextual mismatches that might undermine training effectiveness. Including such cases in future studies would contribute to a more comprehensive understanding of what works, what doesn't, and under which conditions.

This mapping offers researchers an updated view of the state of training initiatives in the software engineering industry. By identifying underexplored areas and patterns of instructional design, this study also serves as a reference for professionals responsible for designing training programs. It encourages evidence-based decision-making and highlights theoretical foundations that can inform and enhance future corporate training efforts.

## 5.2 Implications for Industry, Society, and Academia

**For Industry:** This Study presents a series of inputs for a practical framework that organizations can use to evaluate and improve their corporate training strategies. By highlighting underexplored areas—such as job/task analysis and Simulation-Based Training and Games can better align training programs with organizational goals and employee needs. Additionally, the findings point to the importance of structured evaluation methods and long-term follow-up to ensure return on investment.

**For Society:** The quality of software development directly affects the reliability and safety of digital systems used across all sectors of society. Improving training for software engineers ultimately supports the creation of better digital services, enhances cybersecurity practices, and contributes to economic development. Furthermore, by encouraging inclusive and context-aware training

practices, this research promotes the development of more equitable and accessible digital solutions.

**For Academia:** This work identifies critical research gaps and proposes directions for future studies. It lays the groundwork for a research agenda in corporate training for software engineering, advocating for more rigorous methodologies, longitudinal designs, and interdisciplinary approaches. Scholars can build on this mapping to explore the interaction between human factors, organizational behavior, and training outcomes, as well as to develop and validate training frameworks tailored to the software industry context.

## 6 Threats to Validity and Limitations

As with any systematic mapping study, this research is subject to a number of limitations and potential threats to validity. To address these, we followed guidance from relevant methodological literature [46, 49].

**Search and Selection Bias:** There is always a risk that relevant studies were missed due to the use of specific keywords or databases. To mitigate this, we built the search string based on Salas' framework and existing reviews, and conducted both automatic and manual searches.

**Data Extraction Bias:** The process of classifying studies and extracting evidence may be influenced by the researchers' interpretations. To reduce this threat, data extraction was performed in pairs, with a third researcher consulted in cases of disagreement.

**Publication Bias:** Our study only included published research, which may favor positive findings and omit null or negative results. This is a known issue in systematic reviews [49], and should be considered when interpreting the frequency and distribution of findings.

**Generalizability:** This mapping focused on the field of software engineering. Although the use of Salas' framework provides a generalizable lens, findings may not reflect practices in other industries. Moreover, the use of Salas' framework outside its original organizational psychology context introduces potential limitations in fit or interpretation.

**Lack of Methodological Rigor in Primary Studies:** Many of the included papers are experience reports rather than empirical evaluations. As such, the quality and reliability of the evidence varies, which limits the strength of inferences about effectiveness or impact.

Future studies should seek to replicate this mapping in other domains, employ quality assessment checklists during selection, and explore cross-sector comparisons to strengthen the field's theoretical and practical base.

## 7 Conclusion

This study mapped the existing literature on corporate training in software engineering applied to industry, organizing the selected works according to Eduardo Salas' training framework. Through the analysis of 26 articles, the study identified prevailing themes, emerging patterns, and critical gaps in the development and evaluation of training programs targeting software engineering professionals.

The findings reveal a predominant focus on the category of **Training Methods and Instructional Strategies**, with a notable

diversity in pedagogical approaches such as e-learning, hands-on activities, workshops, and problem-based learning. However, significant gaps remain in other categories—particularly **Training Needs Analysis**, **Antecedent Training Conditions**, and **Post-Training Conditions**—highlighting the lack of studies on areas such as *Job/Tasks Analysis* and *Simulation-Based Training and Games*.

The study also observed that although some methods were employed to assess training needs, individual profiles, instructional delivery, and outcomes, there is insufficient evidence on the effectiveness and consistency of these methods across different organizational contexts. Most contributions take the form of experience reports, limiting generalizability and empirical robustness.

Additionally, authors considered practical aspects of training design—such as duration, structure, and curriculum content—but there is little standardization among the programs. This heterogeneity reflects the contextual nature of corporate education in the field but also underscores the need for more systematic and comparative research.

In doing so, this mapping contributes to the systematization of knowledge in the field by offering a structured overview of current practices and uncovering opportunities for future investigation. It serves as a resource for both researchers and practitioners aiming to design, implement, and evaluate more effective training initiatives within software engineering environments.

As a next step, we recommend expanding the scope of the review to include additional databases, gray literature, and potentially non-English publications. Future studies may also benefit from incorporating quality assessment criteria, conducting meta-analyses where applicable, and examining training initiatives through longitudinal or experimental designs to better understand their long-term impact on professional performance and organizational outcomes.

This research was originally developed in Portuguese. The authors used OpenAI's ChatGPT and DeepSeek Chat exclusively as support tools for translating the content into English and improving the cohesion and clarity of the text.

## ARTIFACT AVAILABILITY

Sheet with the extracted data a search string: link.

## REFERENCES

- [1] M. Agarwal, T. K. Nayak, and V. P. Gupta. 2008. Employees Perception towards Training in IT Sector. *Asia Pacific Business Review* 4, 2 (2008), 34–41. doi:10.1177/097324700800400205 Original work published 2008.
- [2] Ashish Arora and Suma Athreya. 2002. The software industry and India's economic development. *Information Economics and Policy* 14, 2 (2002), 253–273. doi:10.1016/S0167-6245(01)00069-5 The New Economy.
- [3] Ravi Bapna, Nishtha Langer, Amit Mehra, Ram Gopal, and Alok Gupta. 2013. Human capital investments and employee performance: An analysis of IT services industry. *Management Science* 59, 3 (2013), 641–658.
- [4] Brian Berenbach and Taryn Rayment. 2008. The evaluation of a requirements engineering training program at Siemens. In *2008 16th IEEE International Requirements Engineering Conference*. IEEE, IEEE, Barcelona, Spain, 205–210.
- [5] André B Bondi and Johannes P Ros. 2009. Experience with training a remotely located performance test team in a quasi-agile global environment. In *2009 Fourth IEEE International Conference on Global Software Engineering*. IEEE, IEEE, USA, 254–261.
- [6] John W Boudreau and Peter M Ramstad. 2005. Talentship, talent segmentation, and sustainability: A new HR decision science paradigm for a new strategy definition. *Human Resource Management: Published in Cooperation with the School of Business Administration, The University of Michigan and in alliance with the Society of Human Resources Management* 44, 2 (2005), 129–136.

[7] Manoj Boyeena and Prakash Goteti. 2011. A blended approach to course design and pedagogy to impart soft skills: An IT company's experiences from software engineering course. In *IEEE Technology Students' Symposium*. IEEE, IEEE, USA, 6–10.

[8] Guilherme Tavares Cazeri, Mariana Garcia, and Sergio Eduardo Gama Costa. 2022. Training for Industry 4.0: A systematic literature review and directions for future research. *The International Journal of Advanced Manufacturing Technology* 118 (2022), 4761–4786. doi:10.1007/s00170-021-08090-0

[9] KM Chethana and SD Noronha. 2023. A Review of the Strategies Used in the IT sector for Employee Training and Development. *International Journal of Case Studies in Business IT and Education* 7, 2 (2023), 141–155.

[10] Orges Cico, Letizia Jaccheri, Anh Nguyen-Duc, and He Zhang. 2021. Exploring the intersection between software industry and Software Engineering education: A systematic mapping of Software Engineering Trends. *Journal of Systems and Software* 172 (2021), 110736.

[11] Wellynton Diniz, Marcela Valen  a, C  sar Fran  a, Alessandro Santos, and Mariana Pincovsky. 2024. The Skill Gap in Software Industry: A Mapping Study. In *Simp  sio Brasileiro de Engenharia de Software (SBES)*. SBC, SBES, BRA, 192–200.

[12] Renata Ferreira Castaldelli dos Santos, Silvia Lucia Rocha, and Adalberto Zorzo. 2014. Educa  o corporativa. *Revista Cientifica da FHO/ Uniararas* 2, 1 (2014), 50–57.

[13] Marisa Pereira Eboli. 2004. *Educa  o corporativa no Brasil: mitos e verdades*. Editora Gente, BRA.

[14] William Fitzgerald. 1992. Training versus development. *Training & development* 46, 5 (1992), 81–84.

[15] Thomas N Garavan. 1997. Training, development, education and learning: different or the same? *Journal of European industrial training* 21, 2 (1997), 39–50.

[16] Vahid Garousi, G  rkem Giray, Eray T  z  n, Cagatay Catal, and Michael Felderer. 2019. Aligning software engineering education with industrial needs: A meta-analysis. *Journal of Systems and Software* 156 (2019), 65–83.

[17] Natalya Marjana Goelzer, Pedro Portella Possamai, Sabrina Marczak, Michael da Costa M  ra, and Daniel Antonio Callegari. 2024. Nurturing Talent: The IT Academy Journey into Quality Development. In *Proceedings of the XXIII Brazilian Symposium on Software Quality*. IEEE, USA, 508–518.

[18] John Hudepohl, Alpana Dubey, Sylvie Moisy, Jessica Thompson, and Hans-Martin Niederer. 2014. Deploying an online software engineering education program in a globally distributed organization. In *Companion Proceedings of the 36th International Conference on Software Engineering*. IEEE, USA, 301–310.

[19] N Akbar Jan and C Muthuvelayutham. 2012. Employees' training and development in IT sector: An essential instrument for effectiveness of organisational productivity. *European Journal of Scientific Research* 81, 3 (2012), 397–407.

[20] Donald L Kirkpatrick, RL Craig, and LR Bittel. 1970. *Evaluation of training*. Technical Report ED040336. ERIC. 35 pages. Retrieved from <https://eric.ed.gov/?id=ED040336>.

[21] Vidya Kulkarni, Christelle Scharff, and Olly Gotel. 2010. From Student to Software Engineer in the Indian IT Industry: A Survey of Training. In *2010 23rd IEEE Conference on Software Engineering Education and Training*. IEEE, USA, 57–64. doi:10.1109/CSEET.2010.25

[22] Mousa Masadeh. 2012. Training, education, development and learning: what is the difference? *European Scientific Journal* 8, 10 (2012), 62–68.

[23] Alfiya Rafisovna Masalimova and Lidiya Leonidovna Sabirova. 2014. Multi-dimensional classification of types and forms of corporate education. *American Journal of Applied Sciences* 11, 7 (2014), 1054–1058.

[24] James Mason. 2003. Aligning workforce development & software process improvement strategy for accelerated adoption of software engineering capability. In *Proceedings 16th Conference on Software Engineering Education and Training, 2003 (CSEE&T 2003)*. IEEE, IEEE, USA, 70–77.

[25] Itzel Morales-Ramirez and Luis H Alva-Martinez. 2018. Requirements analysis skills: how to train practitioners?. In *2018 IEEE 8th international workshop on requirements engineering education and training (REET)*. IEEE, IEEE, Banff, AB, Canada, 24–29.

[26] Damla Oguz and Kaya Oguz. 2019. Perspectives on the gap between the software industry and the software engineering education. *Ieee Access* 7 (2019), 117527–117543.

[27] Scott B Parry. 1996. The quest for competencies. *Training* 33, 7 (1996), 48.

[28] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. 2008. Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE)*. BCS Learning & Development, BCS Learning & Development, Bari, Italy, 68–77.

[29] Rafael Prikladnicki and Leonardo Pilatti. 2008. Improving contextual skills in global software engineering: A corporate training experience. In *2008 IEEE International Conference on Global Software Engineering*. IEEE, IEEE, USA, 239–243.

[30] Danilo Ribeiro, Rayfran Lima, C  sar Fran  a, Alberto Souza, Isadora Silva, and Gustavo Pinto. 2023. Understanding self-efficacy in software engineering industry: an interview study. In *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*. ACM, Oulu, Finland, 101–110.

[31] G  nther Ruhe. 1999. Experience Factory-based professional education and training. In *Proceedings 12th Conference on Software Engineering Education and Training (Cat. No. PR00131)*. IEEE, IEEE, New Orleans, LA, USA, 62–72.

[32] Eduardo Salas and Janis A Cannon-Bowers. 2001. The science of training: A decade of progress. *Annual review of psychology* 52, 1 (2001), 471–499.

[33] Eduardo Salas, Scott I Tannenbaum, Kurt Kraiger, and Kimberly A Smith-Jentsch. 2012. The science of training and development in organizations: What matters in practice. *Psychological science in the public interest* 13, 2 (2012), 74–101.

[34] Eduardo Salas, Katherine A Wilson, C Shawn Burke, and Dennis C Wightman. 2006. Does crew resource management training work? An update, an extension, and some critical needs. *Human factors* 48, 2 (2006), 392–412.

[35] Philip Samuel et al. 2015. Synchronous training in distributed software development team. In *2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE)*. IEEE, IEEE, Amritsar, India, 315–319.

[36] Michael Schaul. 1985. Design using software engineering principles: overview of an educational program. In *Proceedings of the 8th international conference on Software engineering*. IEEE Computer Society, London, England, 201–208.

[37] Sumaiya Shafiq and Sahibzada Muhammad Hamza. 2017. The effect of training and development on employee performance in private company, Malaysia. *International Journal of Education, Learning and Training* 2, 2 (2017), 42–56.

[38] Jos   Antonio Almeida Silva, Tamara Larissa Duarte De Albuquerque, Denis De Gois Marques, Felipe Tabosa Da Silva, Williams Barbosa Santos, and Bruno Bueno De Lira Nepomuceno. 2023. Corporative Education Used to Attract and Qualify Workforce for the Software Industry. In *2023 18th Iberian Conference on Information Systems and Technologies (CIST)*. IEEE, IEEE, Aveiro, Portugal, 1–6.

[39] Surabhi Sinha and Kalyan Sengupta. 2020. Role of leadership in enhancing the effectiveness of training practices: Case of Indian information technology sector organizations. *Paradigm* 24, 2 (2020), 208–225.

[40] Marija Stanojecka. 2024. Personalized training for successful implementation of advanced technologies in the IT industry. In *Proceedings of the International Conference on Engineering Management and Competitiveness (EMC 2024)*. University of Novi Sad, Faculty of Technical Sciences, Serbia, 84–88.

[41] Hidekuni Tsukamoto, Yasuhiro Takemura, Hideo Nagumo, and Kenichi Matsumoto. 2011. Analysis of the motivation of learners in the in-house training of programming in Japanese ICT industries. In *2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, IEEE, USA, 121–128.

[42] Mehmet Ulema and Jack Larsen. 1984. Planning for in-house software engineering education. In *Proceedings of the fifteenth SIGCSE technical symposium on Computer science education*. ACM, Philadelphia, PA, USA, 130–136.

[43] Stephen C Wingreen and J Ellis Blanton. 2006. A q-methodological study of IT professionals' person-organization fit for training and development: research-in-progress. In *Proceedings of the 2006 ACM SIGMIS CPR conference on computer personnel research: Forty four years of computer personnel research: achievements, challenges & the future*. ACM, New York, NY, USA, 158–163.

[44] Stephen C Wingreen and J Ellis Blanton. 2018. IT professionals' person-organization fit with IT training and development priorities. *Information Systems Journal* 28, 2 (2018), 294–317.

[45] Stephen C Wingreen, J Ellis Blanton, Sandra K Newton, and Madeline Domino. 2005. Assessing the IT training and development climate: an application of the Q-methodology. In *Proceedings of the 2005 ACM SIGMIS CPR conference on Computer personnel research*. ACM, Atlanta, GA, USA, 12–23.

[46] Claes Wohlin. 2014. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. ACM, London, England, UK, 1–10. doi:10.1145/2601248.2601268

[47] Patrick Wolfschwenger, Barbara Sabitzer, and Zsolt Lavicza. 2022. Design and evaluation of an agile framework for continuous education in software engineering. In *2022 IEEE Frontiers in Education Conference (FIE)*. IEEE, IEEE, Uppsala, Sweden, 1–9.

[48] Patrick Wolfschwenger, Barbara Sabitzer, and Zsolt Lavicza. 2023. Integrating Cloud-Based AI in Software Engineers' Professional Training and Development. In *2023 IEEE Frontiers in Education Conference (FIE)*. IEEE, IEEE, College Station, TX, USA, 1–5.

[49] Bing Zhou, Yan Yang, and He Zhang. 2016. Validity Threats in Empirical Software Engineering Research—A Review. In *Proceedings of the 18th International Conference on Software Engineering and Knowledge Engineering (SEKE)*. KSI Research Inc., USA, 257–262.

[50] Svetlana Zubareva, Elena Zubareva, and Galina Lemeshko. 2023. Professional Development of IT Industry Specialists at the Workplace: Trends, Focus and Prospects. In *Proceedings of the 2023 3rd International Conference on Technology Enhanced Learning in Higher Education (TELE)*. IEEE, USA, 41–44. doi:10.1109/TELE58910.2023.10184379