www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 16(4), Sep-Dec 2025, 303-318. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2025.v16i4.957

Exploring key factors for cohesion in software development teams

Adriana Peña Pérez Negrón[1], Tina Marie del Rosario[2], David Bonilla Carranza[1], José Esteban Hernández de León[1]
[1] Universidad de Guadalajara, Guadalajara, México
[2] Pepperdine University, CA, USA
adriana.ppnegron@academicos.udg.mx, tina.delrosario@pepperdine.edu, jose.bcarranza@academicos.udg.mx, jose.hernandez8793@academicos.udg.mx

Abstract. Team cohesion plays a significant role in software development, shaping performance and productivity. It fosters strong connections among members working towards a shared objective. Cohesion is a dynamic process influenced by personal, interpersonal and domain-related factors. To identify key drivers of cohesion in software teams, we posed four open-ended questions to 137 industry-experienced developers. Responses most frequently highlighted effective communication, shared goals and mutual support as enablers of better organisation, role clarity and synchronisation, thereby providing structural support for cohesion. Additionally, crisis management, handling external dependencies, conflict resolution and adaptability were cited as foundations for problem-solving. Leadership, recognition and synergy were reported to foster social affinity and receptiveness to peer feedback. Overall, respondents associated cohesion with outcomes such as working affinity, social affinity, dependability, receptiveness to feedback, team-based decision-making and efficiency.
Keywords: soft skill, teamwork, team cohesion, software engineering, team-based decisions

Article Info
Received February 25, 2025
Accepted July 6, 2025

1 Introduction

In an increasingly digital world, software development is growing in importance. Software impacts safety, security, and even human values, especially as automation plays a more significant role in decision-making processes (Chun et al., 2020). The advance of complexity in current software has prompted its development to demand teamwork (Peña Pérez Negrón et al., in press), causing its development to become a highly complex socio-technical activity, requiring both technical expertise and social collaboration, where the team structure and organization influence productivity, efficiency, and project outcomes. The complexity of software teams then lies not only in managing technical tasks, but in the alignment of individual contributions, the resolution of conflicts, and the fostering of collaboration (Sawyer, 2004). Moreover, teamwork improves quality and efficiency, a divided project is easier to manage leading to a faster delivery, associated with fewer errors, and teamwork allows members to bounce ideas, and reinforces novel solutions (Muñoz et al., 2017).

Although, most project teams share common principles of collaboration and goal-setting, software development teams differ significantly in their workflows, tools, flexibility, and approach to management. Developing software is inherently more iterative, automated, and dependent on remote collaboration tools, distinguishing it from other teams in traditional industries. In this sense, software development teams face unique characteristics and challenges for the team members and their management, they require specialized technical skills and knowledge that enable them to manage code complexity, and the integration of multiple systems. In addition, software teams are more likely to be remote or hybrid, with distributed workflows, frequently with a high degree of automation in testing, deployment, and monitoring. Moreover, they deal with changing requirements and continuous updates, handling change and uncertainty; this entails that the members of a software developer team have to stay up-to-date with rapidly evolving technologies (Jiménez, et al., 2009; Ryan & O'Connor, 2009; Hoda, et al., 2012).

In this context, team cohesion has been found essential for productivity, collaboration, and success (Sawyer, 2004; Barros de Morais, 2013; Dey & Ganesh, 2020); whether in software development, business, or sports, cohesive teams outperform those that lack unity. Cohesion has been defined as the degree of unity, trust, and commitment among team members as they work toward a common goal, it reflects how well individuals in a team bond, collaborate, and remain motivated to achieve shared objectives (Carron et al., 1985). As mentioned, software development teams are not an exception, research on cohesion in this domain highlights its importance for team effectiveness and performance (Sawyer, 2004; Barros de Morais, 2013; Dey & Ganesh, 2020).

1.1 Related work

Cohesion has been found to be influenced by various factors, including social gatherings and technical meetings (Carron et al., 1985; Barros de Morais, 2013). Barros de Morais (2013) found that the relationship between cohesion and effectiveness is not always proportional, and that factors that may weaken cohesion include low turnover, isolated work environments, and activities that do not require interaction.

Nootjarat et al., (2015) explored the leader role in fostering collaboration to strengthen cohesion in central task-advice networks. They conducted a survey on software development teams and found that teams with highly central leaders had a stronger team cohesion-performance effect. Based on their results, the authors claim that the leader does not lead in a social vacuum, and those with high centrality play an important role in connecting team members.

Kakar & Kakar (2018) investigated the relationship between team cohesion and innovation. According to the authors, while cohesion is often linked to positive outcomes they explored whether high levels of cohesion might hinder innovation by promoting conformity and suppressing dissent. The authors found that cohesion can have both positive and negative effects on innovation, depending on how it interacts with other factors like team climate, leadership, and communication openness. Results showed that cohesion enhances innovation when paired with psychological safety and constructive conflict, but may limit it when groupthink dominates. The study highlighted the importance of balancing cohesion with mechanisms that support diverse perspectives and open dialogue, offering insights for managing team dynamics in innovation environments.

One of the aforementioned authors also explored in Kakar (2018) the role of autonomy and independence in team cohesion. Through empirical results in software development projects, it was found that high levels of interdependence, autonomy, and interdependent tasks, represent a synergistic impact on cohesion, while low levels have an antagonistic impact on team cohesion.

Dey & Ganesh (2020) conducted a study of social cohesion in software development teams. They analyzed the members' interaction on the effects of team boundedness, formal coordination, and organization tenure diversity. Team boundedness is understood as teamwork, formal coordination as the team members' alignment to procedures in their daily task activities, and organizational tenure diversity as the team composition that conveys the availability of team knowledge. Dey & Ganesh (2020) results showed that task and social cohesion are positively affected by team boundedness and formal coordination; formal coordination has a stronger impact on task cohesion; and organization tenure diversity negatively affects both types of cohesion, with a stronger impact on social cohesion.

Apparently, communication represents a significant factor in cohesion. Castro-Hernández et al., (2022) found in their study of global software development teams that temporal patterns of communication affect cohesion. Frequent and rhythmic interactions positively correlated with task cohesion. According to them, providing cohesion-based feedback to teams increases the message count, response rate, and individual cohesion scores. Also, frequent communications measured by pacing rate, show a negative correlation with task cohesion, suggesting that more frequent interactions increase cohesion. Additionally, synchrony in communication rhythms positively correlates with task cohesion, emphasizing the importance of establishing consistent communication patterns. The authors claim that these temporal models, at both individual and group levels, are good predictors of task cohesion, denoting the significance of communication frequency and rhythm in fostering team cohesion.

Correspondingly, the different factors that support cohesion are team boundedness (i.e., a sense of teamwork), social gatherings, shared goals or task alignment, formal coordination, recognition, support among members, leadership with high centrality (e.g., well-connected, collaborative leaders), mutual trust, understanding of roles and responsibilities, frequent and rhythmic communication patterns, open and honest feedback culture, collaborative work environments, opportunities for knowledge sharing, and inclusion in decision-making processes.

These findings underscore the complex nature of cohesion in software development teams and its impact on team dynamics and performance. Social and task-related factors have been studied in the context of software development teams to describe the factors

that support cohesion to get insights into this important dynamic process that affects team performance. Regardless of the importance of team cohesion in improving team performance, there is no complete agreement on the factors that contribute to its development, particularly in software development teams given their special features. In this study, a structured interview with four open questions was applied to 137 software developers in the industry, to establish the key factors that benefit the process of cohesion.

2 Background

Back in the mid-80s Carron et al., (1985) conducted a study widely recognized for its strong theoretical and psychometric foundation that constituted the basis for the Group Environment Questionnaire (GEQ). The study was applied to sports teams, nonetheless, it has been adapted to several languages and contexts (e.g., Hanrahan & Gallois, 1993; Iturbide, et al., 2010; Prokešová et al., 2015; Do Nacimiento Junior, et al., 2012; Bosselut et al., 2018). Carron et all. (1985) model differentiates two categories: group members' perception of the team and their attraction to the team. Within this framework, group integration (GI) represents the degree to which the group remains unified based on social interactions and relationships. And the individual attraction to the group (ATG) reflects the engagement and commitment of its members. Both categories are shaped by task and social-related factors, giving as a result four dimensions: social aspects of group integration (GIS); task aspects of group integration (GIT); social aspects of individual attraction to the group (ATGS); and task aspects of individual attraction to the group (ATGT).

In a previous study (Peña Pérez Negrón et al., in press), because of the particular characteristics of software development teams, instead of adapting the GEQ, it was decided to follow Carron et al. (1985) methodology allowing the team members to be involved as active agents in expressing the meaning of cohesion. To identify cohesion concepts and the items that reflect the expression of those concepts, Carron et al. (1985) designed four projects with different methodologies: 1) interview questions to sports team members; 2) interview questions to undergraduate sports team members; 3) interview questions to undergraduate sports team members in an ongoing competition; and 4) a literature search of studies on the topic of cohesion.

The conducted study in (Peña Pérez Negrón et al., in press) with undergraduate students used the Carron et al. (1985) design for the questionnaires. Each participant answered one type of focus in the questions, adapted for software development teams as shown in Table 1.

Table 1. Questions for software developers, undergraduate students (adapted from Carron et al., 1985)

Reasons to	A) Self-focused	B) Team-focused	C) Focused on software development engineering
Join a team	A.1) Why would you join a team to develop software?	B.1) Why do you think your peers would join a software development team?	C.1) Why do you think a software developer would join a team?
Leave a team	A.2) Why would you leave a software development team?	B.2) Why do you think your peers would leave a software development team?	C.2) Why do you think a software developer would leave a team?
Stay in a team	A.3) Why would you stay with a software development team?	B.3) Why do you think your peers would stay on a software development team?	C.3) Why do you think a software developer would stay on a team?

Three software project leaders categorized the responses in 11 key factors for cohesion according to their dimension as shown in Table 2.

Results showed that the main reason to join a software development team among undergraduates (42% of the mentioned factors) corresponded to "division of labor", very probably to split the assignment workload. Also, the idea that working on a team indicates organization and efficiency was mentioned several times, along with the appreciation that working in a team makes the assignment easier to accomplish. On the contrary, the main reasons for leaving a team in undergraduate students were the lack of affinity, bad communication, and disagreements on the course of the project. And, the main reasons to stay in a team were affinity, enjoying working on that team, communication, and organization (Peña Pérez Negrón et all., in press).

As was pointed out in the study (Peña Pérez Negrón et all., in press), in contrast with sports teams, undergraduate students have a different perception of the reasons that contribute to team cohesion. And, because the students have different interests than

software developers in the industry, this study targets the software developer teams in the industry, following the approach of the first project in Carron et al., (1985).

Table 2. Factors for cohesion in undergraduate software development teams (taken from Peña Pérez Negrón et al., in press)

Dimension	Factor	Description
ATGS	Social affinity	Getting along with the others, liking them at a social level. This includes the counterpart for leaving a team, and a lack of affinity.
ATGT	Project attractiveness	The project is interesting enough to motivate their participation. This includes the causes project, shared goals, different goals, and contributing to a project.
	Working affinity	The developer enjoys the workflow of the team. This factor comprises the topics of facilitating the job easily, enjoying working on that team, and not working as a team.
	Self-improvement	The opportunity to develop hard and soft skills, from the topic with the same name, includes the counterpart, not self-improvement.
GIS	Organization	Proper project management aimed at efficiency. The causes considered were organization, disorganization, good practices, no proper methodology, unequal workload, equitable job, and bad leadership.
	Communication	The communication among all members is likely and proper.
	Good performance	Software quality improvement. This includes the issues of software quality, lo w-quality software, getting results and not getting results, and a lack of knowledge
	Sharing ideas	The importance of having others' points of view.
GIT	Division of labor	Having the advantages of breaking out the task to make it more manageable, for less work and effort. This included the topics of division of labor, complexity; and also, complementing each other because skills complementation allows specialization.
	Sharing knowledge	The importance of learning within the project from their peers includes the topics of support and camaraderie.
	Dependability	This refers to the responsibility of the members to do their part, the topic includes a lack of engagement.

3 Experimental procedures

A structured interview with open answers was conducted using the following four questions (adapted from Carron et al., 1985): 1) For you, what is the meaning of cohesion in teamwork?; 2) Mention a situation in which you see reflected cohesion; 3) Do you recall an incident that denotes low or the absence of cohesion?; and 4) Describe the factors that you consider contribute to the development of team cohesion.

Participants. One hundred thirty-seven software developers in the industry were interviewed. Twelve females, and 125 males. Ten participants work in an administrative position with staff under supervision. The average time in the industry was 50 months, with a high dispersion of data, and a 2 to 408 months range.

Data. The responses were transcribed and classified by three experts in the field, project managers in the software development industry. Tables 3, 4, 5, and 6 present the categories of issues found in the responses for each question. The Table structure in each line is the name of the factor followed in parentheses by the number of times it was mentioned, then a list with description based on the answers, and an example of one answer mentioning such factor.

The resulted categorization of the responses to the first question "*a)* For you, what is the meaning of cohesion in teamwork?" is next listed with the number of times they were mentioned in parenthesis, followed by its descriptions, and an example answer:

Communication (32 times)

- The ability to exchange information clearly and effectively.
- Open discussions, timely feedback, and understanding of what others are doing.
- Assertive communication and periodic check-ins or meetings to ensure clarity.

Example answer: *"The key is having clear objectives so that everyone knows what needs to be done. Regular updates and feedback help keep the team aligned."*

Shared goal (24 times)

- A shared vision, all working toward the same objective.

Example of answer: *"Cohesion means that everyone is aligned with the same goal and follows a structured path toward achieving it."*

Support (22 times)

- Recognition of a colleague in need of help and stepping in.
- Openness to ask for assistance and provide support when needed.
- Comprehension of the fact that no one is perfect and that teams function better when members help each other.

Example of answer: *"Cohesion is knowing when to ask for help and when to offer support to keep the team moving forward."*

Working affinity (22 times)

- The ability to work well together, sharing responsibilities, and avoiding conflicts.

Example of answer: *"A strong team has good relationships and trust, which makes collaboration much easier and more enjoyable."*

Synchronization (18 times)

- Aligned workflows, where tasks are completed in harmony.
- Avoid workflow bottlenecks and ensure that no team member is left waiting on another.

Example of answer: *"Cohesion means that even if we work on different things, we do it in a way that keeps everything moving forward without delays."*

Synergy (12 times)

- Connection between team members, working as a single unit.
- Each member complements others and uses diverse skills to create better results.

Example of answer: *"Cohesion is like synergy—many parts working together as one, with a shared mindset and mutual collaboration."*

Understanding roles (10 times)

- Everyone knows their role and understands how their work impacts the whole project.
- Meaningful contributions of everyone.

Example of answer: *"Cohesion means knowing your role, understanding what your teammates do, and working together efficiently."*

Efficiency (7 times)

- Tasks efficiently, meet deadlines, and organize workflows.
- Reduction of unnecessary steps and avoidance of dependencies.
- Comprehension of priorities and methodical work.

Example answer: *"Cohesion means meeting deadlines, ensuring a steady workflow, and keeping work structured to prevent delays."*

Adaptability (6 times)

- Adjustment to new challenges without disruption of the workflow.
- Handle unexpected problems, roles, or responsibilities adjustments when necessary.
- Open to new ideas and willing to shift approaches if needed.

Example of answer: *"A cohesive team adapts to changes quickly and finds solutions without unnecessary delays."*

As can be observed the most frequently mentioned elements of team cohesion were communication, shared goals, support, work affinity, and synchronization. It can be said that these factors create an environment where people collaborate effectively with a proper assignment of tasks, trust in each other, and adaptability to challenges together.

The resulted categorization of the responses to the second request for the participants "*b)* Mention a situation in which you see reflected cohesion" is next listed with the number of times they were mentioned in parenthesis, followed by its descriptions, and an example answer:

Communication (32 times)

- Keep all team members informed to ensure smooth collaboration.
- Regular meetings, daily stand-ups, and discussions for identification and solving problems early.

Example of answer: *"In my company, we hold daily meetings of 5-10 minutes to discuss our progress, issues, and questions. This ensures we all stay aligned."*

Understanding roles (28 times)

- When each team member knows their responsibilities and can execute tasks without confusion.
- Assignment of tasks based on expertise, equal workload distribution, and efficient workflows.
- Avoiding overlapping responsibilities and planning tends to lead to better performance.

Example of answer: *"For a web development project, the backend team worked on APIs while the frontend team focused on UI, ensuring a seamless integration."*

Support (34 times)

- Help others during challenges, no one is left struggling alone.
- Senior developers who assist juniors, or step in to solve a difficulty together.
- Mentorship among them, and help newcomers to integrate quickly.
- Sessions of training, code reviews, and collaborative learning

Example of answer: *"A developer got stuck with a bug, so the team gathered, analyzed the code together, and quickly identified the issue."*

Crisis management (20 times)

- React well under pressure, being able to organize quickly to find and implement solutions.
- Handle critical difficulties, fix production issues, or recover from failed deployments.

Example of answer: *"When a malware attack hit our system, the team immediately coordinated remotely and resolved the issue efficiently."*

Synchronization (18 times)

- Well-integrated workflows, smoothly transition in phases without roadblocks. Handoffs between design, development, and test, ensure seamless project progression.

Example of answer: *"We followed a structured process: first, designers provided mock-ups, then developers implemented them, and finally, QA tested before deployment."*

Organization (15 times)

- An adequate methodology. Clear process planning and regular progress review.

Example of answer: *"During our sprint planning, we discuss each task, ensure no one is overloaded, and adjust priorities together as a team."*

Handling external dependencies (12 times)

- Coordination with external stakeholders.
- Keep clients informed, manage expectations, and ensure compatibility between systems.
- Integration of client feedback smoothly

Example: *"When working on an API integration, our team stayed in constant contact with the external provider to ensure a smooth deployment."*

Team-based decisions (8 times)

- Collective decisions; practices like brainstorm sessions, team-wide discussions, and voting on solutions as signs of strong cohesion.

Example of answer: *"Before making major design changes, we always hold a meeting to discuss potential options and agree on the best approach together."*

Cultural and remote work adaptability (6 times)

- Effective work across different locations and time zones.
- Adaptation to working with international teams.

Example of answer: *"Since we work with a remote team in another country, we use asynchronous communication to keep everyone informed despite time zone differences."*

In accordance with the responses, the cohesion in software development teams seems to be related to strong communication, clear roles and task assignments, mutual support, and efficient crisis management or problem-solving. Synchronization across roles and tasks, organization mainly using methodologies, handling external dependencies like stakeholders or clients, and a culture of collaborative learning were other factors mentioned that strengthened cohesion.

For the question "*c)* Do you recall an incident that denotes low or absence of cohesion?" the answers are the counterpart, which means, those factors that provoke low or absence of cohesion. The resulted categorization is next listed with the number of times each factor was mentioned in parenthesis, followed by its descriptions, and an example answer:

Poor Communication (36 times)

- Unclear instructions, lack of updates, missing documentation, and misinterpreted requirements.

Example of answer: *"Due to a lack of communication, we ended up deleting an entire database by accident."*

Lack of organization (28 times)

- Poorly assigned responsibilities, confusion about roles, unclear instructions, or failing to properly distribute tasks.

Example of answer: *"Our manager assigned a task, but nobody really understood what needed to be done, leading to delays and confusion."*

Individualism (24 times)

- Refusal to collaborate, ignoring struggling teammates, or focusing only on their own work. Also, refuse to share information or documentation.

Example of answer: *"I had a teammate who refused to explain the full context of a project, only answering the minimum required, which slowed me down."*

Lack of working affinity (26 times)

- Power struggles, egos, and personality clashes, senior members treating juniors badly, team members openly criticizing each other's work, and cliques isolating members.

Example of answer: *"A teammate constantly criticized others' work instead of offering constructive feedback, creating an uncomfortable environment."*

Misalignment in goals (18 times)

- Different interpretations of project goals, working on features that were not needed, or applying the wrong approach due to a lack of alignment.

Example of answers: *"Each person interpreted the requirements differently, so when we put everything together, nothing worked as intended."*

Resistance to feedback (15 times)

- When team members reject feedback, refuse to adjust their work, or ignore process improvements, and also when developers do not accept that they are mistaken.

Example of answer: *"One teammate refused to accept feedback, insisting his way was right, even though it caused errors in production."*

Failures in distributed work (12 times)

- Remote teams faced scheduling conflicts, missing information, and a lack of inclusion in key meetings.

Example of answer: *"Since I work remotely, I was left out of important meetings, which made it difficult to keep up with project changes."*

Failures in requirements (10 times)

- Teams struggled with last-minute requirement changes or unclear specifications.

Example of answer: *"We built a feature based on unclear requirements, only to be told later it was not what the client wanted."*

Not team-based decision-making (8 times)

- Leaders fail to consider team input, making decisions without consulting developers or imposing unrealistic deadlines.

Example of answer: *"Our manager ignored our concerns about a new technology, forcing us to use it even though half the team was not trained for it."*

According to the responses, teams with low cohesion often suffer from poor communication, unclear roles, individualistic attitudes, and misalignment of goals. These lead to delays, rework, frustration, and inefficient workflows.

The last request for the participants "*d)* Describe the factors that you consider contributing to the development of team cohesion" was categorized. The list with the number of times each factor was mentioned in parenthesis, followed by its descriptions, and an example answer is next presented:

Communication (36 times)

- Active listening, clarity in instructions, and timely updates.

Example of answer: *"Good communication helps everyone understand what they are working on and how their work fits into the bigger picture."*

Shared goals (38 times)

- Clarity on long-term project goals and how individual tasks contribute.

Example of answer: *"When all team members understand the project's purpose, they collaborate more effectively to achieve it."*

Dependability (28 times)

- When team members trust each other's work and judgment, there is respect for different opinions, roles, and contributions.

Example of answer: *"A team that trusts each other can work more efficiently, as no one is second-guessing others' decisions."*

Organization (24 times)

- Understand the responsibilities and each member's contribution to the project.

- Leadership that helps to maintain structure and resolve conflicts.

Example of answer: *"A well-structured team with clear roles reduces confusion and keeps workflows smooth."*

Support (18 times)

- Willingness to help each other.
- Collective problem-solving, pair programming, code reviews, and shared decision-making.

Example of answer: *"If a team member struggles, offering support instead of letting them fail alone helps the whole team succeed."*

Conflict resolution (15 times)

- Strategies for resolving disputes constructively.

Example of answer: *"When conflicts arise, discussing and solving them professionally strengthens team unity."*

Social affinity (12 times)

- A positive work culture, good morale, and strong relationships.
- Social interactions, such as team-building activities and casual meetups.
- A friendly work environment makes people more willing to collaborate and communicate.

Example of answer: *"When teams socialize outside of work build stronger relationships and work better together."*

Efficiency (10 times)

- Understanding of tools and technologies, experienced members mentoring newer team members.
- Continuous learning helps teams stay adaptable.

Example of answer: *"Knowing how to use our tools properly prevents unnecessary mistakes and improves efficiency."*

Leadership (8 times)

- Leaders set the tone for organization, motivation, and decision-making.
- Proactive leaders resolve conflicts quickly and fairly

Example of answer: *"A good leader ensures everyone stays aligned and motivated."*

Recognition (6 times)

- Team members feel more engaged when their contributions are recognized.
- Recognition of both successes and learning opportunities maintains motivation.

Example of answer: *"When team members receive feedback, they know what they are doing well and where to improve, making the team stronger."*

According to the number of times the factors were mentioned in the answers, the most important contributors to cohesion are communication, trust, clear roles, teamwork, and shared goals. Also, teams that focus on strong leadership, continuous learning, and a positive work culture present better cohesion.

4 Results

The answers were merged in Table 3. In the first column is the named factor; next to it in parentheses, the number of times it was mentioned. The second and third columns were taken from the descriptions in the responses. They contain the description itself, and the features of these factors that contribute to team cohesion. When the answers were a counterpart factor, it was included in this same column with a hyphen preceding its description (as in our previous study, Peña Pérez Negrón et al., in press). “Resistance to feedback” was individually changed to “Receptivity to feedback” as its counterpart. The factors were ordered by the number of times they were mentioned. Also, “Failures in distributed work” and “Failures in requirement” were included in the “Organization” factor because they were related to the fact that there is no proper methodology that supports the organization. Finally, in “Adaptability” was included “Cultural and remote work adaptability” as a special way of adaptability. And, although some of the mentioned detrimental issues for cohesion in this factor have to do with “Organization” too, it was decided to keep them here because adaptability was mentioned.

In Fig. 1 are presented the all the factors founded in the responses that contribute to team cohesion and the number of times they were mentioned in the four questions of the interviews. Finally, because cohesion might be perceived differently on the career stage, the factors were clustered based on the years of professional experience. Three groups were formed 0 to 3 years of experience considered as early career developers, 4 to 8 years of experience as mid-level developers, and with more than 9 years of experience as senior developers or leaders. The most mentioned factors for each of these groups were, by career developers, support, communication, shared goals, understanding roles, leadership, and recognition; by mid-level developers, communication, team-based decisions, organization, working affinity, conflict resolution, and adaptability; and by senior developers or leaders, leadership, dependability, crisis management, handling external dependencies, recognition, synchronization, and efficiency.

Table 3. Merge of categorization of the responses to the four questions.

Factor	Description	Features that contribute to team cohesion
Communication (136) -Poor communication	The ability to exchange information clearly and effectively	Open discussions, timely feedback, and understanding of what others are doing. Assertive communication. All team members were informed to ensure smooth collaboration. Regular meetings, and daily stand-ups to identify and solve problems early. Clear instructions, updates, no missing documentation, and well-interpreted requirements
Shared goals (78) ***-Misalignment in goals***	A shared vision, all working toward the same objective.	Clarity on long-term project goals and how individual tasks contribute. Same interpretations of project goals, working only on pm features that are needed, and using the right approach.
Support (78) ***-Individualism***	Openness to ask for assistance and provide support when needed	Step in when a colleague needs help. When team members help each other during challenges, no one is left struggling alone. Senior developers assisting juniors, or team members stepping in to solve a difficulty together. Understanding that no one is perfect and that teams function better when they help each other. Not refusing to share information or documentation Collective problem-solving, pair programming, code reviews, and shared decision-making.
Working affinity (48) ***-Lack of working affinity***	*The ability to work well together, sharing responsibilities, and avoiding conflicts.*	Avoid power struggles, egos, and personality clashes, senior members treating juniors badly, or team members openly criticizing each other's work, cliques isolating members.
Organization (39) ***-Failures in distributed work*** ***-Failures in the requirement***	An adequate methodology	Clear planning processes and progress are regularly reviewed. Comprehension of responsibilities and each member's contribution to the project. Leadership that helps to maintain structure and resolve conflicts. Remote teams without scheduling conflicts, missing information, or a lack of inclusion in key meetings. Requirement changes and unclear specifications.
Understanding roles (38)	Everyone knows their role and understands how their work impacts the whole project.	Meaningful contributions. Team members know their responsibilities and can execute tasks without confusion. Task assignments based on expertise, equal workload distribution, and efficient workflows. Not overlapping responsibilities and planning tend to lead to better performance.
Synchronization (36)	Having aligned workflows, where tasks are completed in harmony**.**	When workflows are well-integrated, each phase smoothly transitions into the next without roadblocks. Handoffs between design, development, and testing, ensure seamless project progression.
Dependability (28)	Trusting each other's work and judgment	There is respect for different opinions, roles, and contributions.
Crisis management (20)	Reacting well under pressure, being able to organize quickly to find and implement solutions	Handling critical difficulties, fixing production issues, or recovering from failed deployments.
Efficiency (17) ***-Poor feedback***	Meeting deadlines, and keeping workflows organized.	Reduction of unnecessary steps and avoidance of dependencies. Everyone understands priorities and works methodically. Understanding of tools and technologies, experienced members mentoring newer team members.

Factor	Description	Features that contribute to team cohesion
		Continuous learn to help teams stay adaptable.
Receptivity to feedback (15) ***-Resistance to feedback***	Team members are receptive to getting feedback	Work adjustment, not ignoring process improvements, and acceptance of mistakes.
Conflict resolution (15)	Strategies for resolving disputes constructively	Disagreements happen, but how they are handled affects cohesion.
Synergy (12)	Having a connection between team members, working as a single unit.	Each member complements others and uses diverse skills to create better results.
Handling external dependencies (12)	Coordination with external stakeholders and clients	Informed clients and stakeholders, management of expectations, and compatibility between systems. Integrating client feedback smoothly
Social affinity (12)	A positive work culture, good morale, and strong relationships.	Social interactions, team-building activities, and casual meetups. A friendly work environment that makes people more willing to collaborate and communicate.
Team-based decisions (8) ***-Not team-based decision-making***	Collectively make decisions	Practices like brainstorming sessions, team-wide discussions, and voting on solutions. Leaders consider team input, make decisions by consulting developers, and do not impose unrealistic deadlines.
Leadership (8)	Leaders set the tone for organization, motivation, and decision-making.	Proactive leaderships resolve conflicts quickly and fairly.
Adaptability (6) ***Cultural and remote work adaptability (6)***	Adjustment to new challenges without disrupting the team's workflow.	Handling unexpected problems and adjusting roles or responsibilities when necessary. Open to new ideas and willing to shift approaches if needed. Adapting to international teams. Effective work across different locations and time zones.
Recognition (6)	Recognition of both successes and learning opportunities maintains motivation.	Team members feel more engaged when their contributions are recognized.

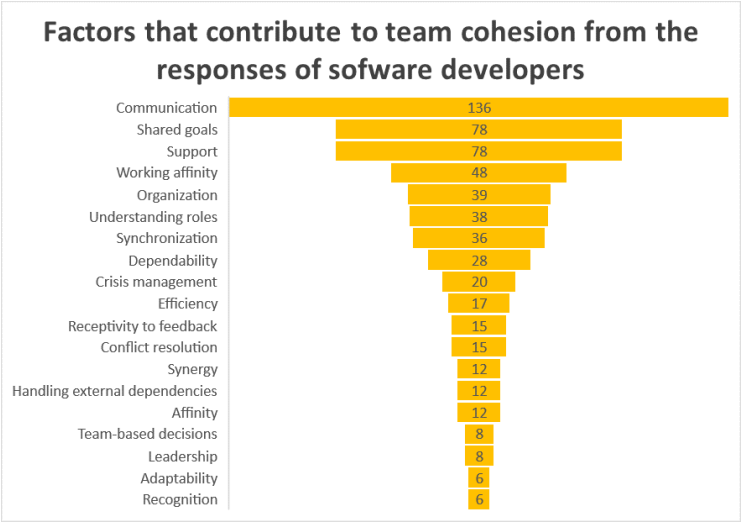

Fig. 1. Factors that contribute to team cohesion and the number of times they were mentioned in the four questions of the interviews.

4.1 Discussion

It is important to highlight that the descriptions of the factors and their features are derived directly from the responses obtained in the interviews. The insights provided reflect the perspectives and experiences shared by the participants, offering a qualitative understanding of the factors influencing cohesion in software development teams. These answers serve as the foundation for identifying key themes, trends, and patterns that emerge from real-world experiences, ensuring that the conclusions drawn are grounded in authentic team dynamics.

We consider that the number of times a factor was mentioned might not reflect the importance of the factor, although it is probably the most common factor or one that affects other factors as well. Then to underscore hidden relations among the factors conceptual maps were developed for better comprehension.

It seems that the three more mentioned factors, that is, *communication*, *shared goals*, and *support* are the best connected to other factors, Fig.2 depicts these relations. With effective *communication* as a central enabler, *communication* ensures *synchronization,* that team members are aligned in *understanding roles*, and that supports them to be able to make *team-based decision*s. Communication also enhances project *organization*, *support*, and collaboration, reducing misunderstandings that lead to inefficiency. And, strong communication facilitates *adaptability*, allowing a quick response to changes.

Shared goals provide a common vision, then they reinforce *the organization* that requires each member to understand their contribution to the project, *synergy* by having a connection between team members, and *work affinity* that support *working well together* and *sharing responsibilities*. *Shared goals* are also enhanced by *team-based decisions.*

*Support*ing, and assisting each other, build *social affinity* and *dependability* seems to ensure effective collaboration complementing each other's strengths, producing a supportive work environment, where *recognition* by the team members is also encouraged. *Recognition* reinforces positive contributions, creating a more engaged and committed team, and therefore, *dependability*.

The strong interconnection among communication, shared goals, and support suggests several practical implications for managing software development teams. First, fostering effective communication should be a strategic priority, as it enables synchronization, role clarity, collaborative decision-making, and adaptability, key components for maintaining team alignment and reducing inefficiencies. By establishing consistent communication channels, such as regular stand-ups, retrospectives, and documentation practices, can be enhanced this dynamic. Second, defining and reinforcing shared goals can strengthen organizational structure, promote synergy, and increase work affinity. This can be achieved through collaborative planning sessions and ensuring all members understand how their contributions align with the project vision. That means that by cultivating a culture of mutual support and recognition not only builds social cohesion and dependability but also enhances engagement and commitment. And, practices such as peer mentoring, acknowledgment of contributions, and structured feedback mechanisms can reinforce the social dynamics, ultimately leading to more cohesive and high-performing teams.

Four factors can be associated with the structure and operation of the team: *organization, understanding roles, synchronization,* and *leadership,* see Fig. 3. *The organization* provides a framework for the work structure, ensuring that the roles are clearly defined and the workloads are evenly distributed leading to *understanding roles* and team *efficiency*, to a smooth *handling of external dependencies*, and to be prepared for proper *crisis management*.

By *understanding roles*, each team member will know their responsibilities and how their work affects others, supporting *dependability* contributing to the respect for different roles and contributions, facilitating *leadership*, and *team-based decision-making*. Also, *understanding roles* enhances *synchronization*.

Synchronization aligns workflows across different roles, reducing bottlenecks, and ensuring *efficiency* and *adaptability* to changing requirements.

Teamwork also benefits from effective *leadership* that balances task management and social cohesion, driving t*eam-based decisions*. Strong *leadership* is key to *conflict resolution,* and to triggering *support* between the team members.

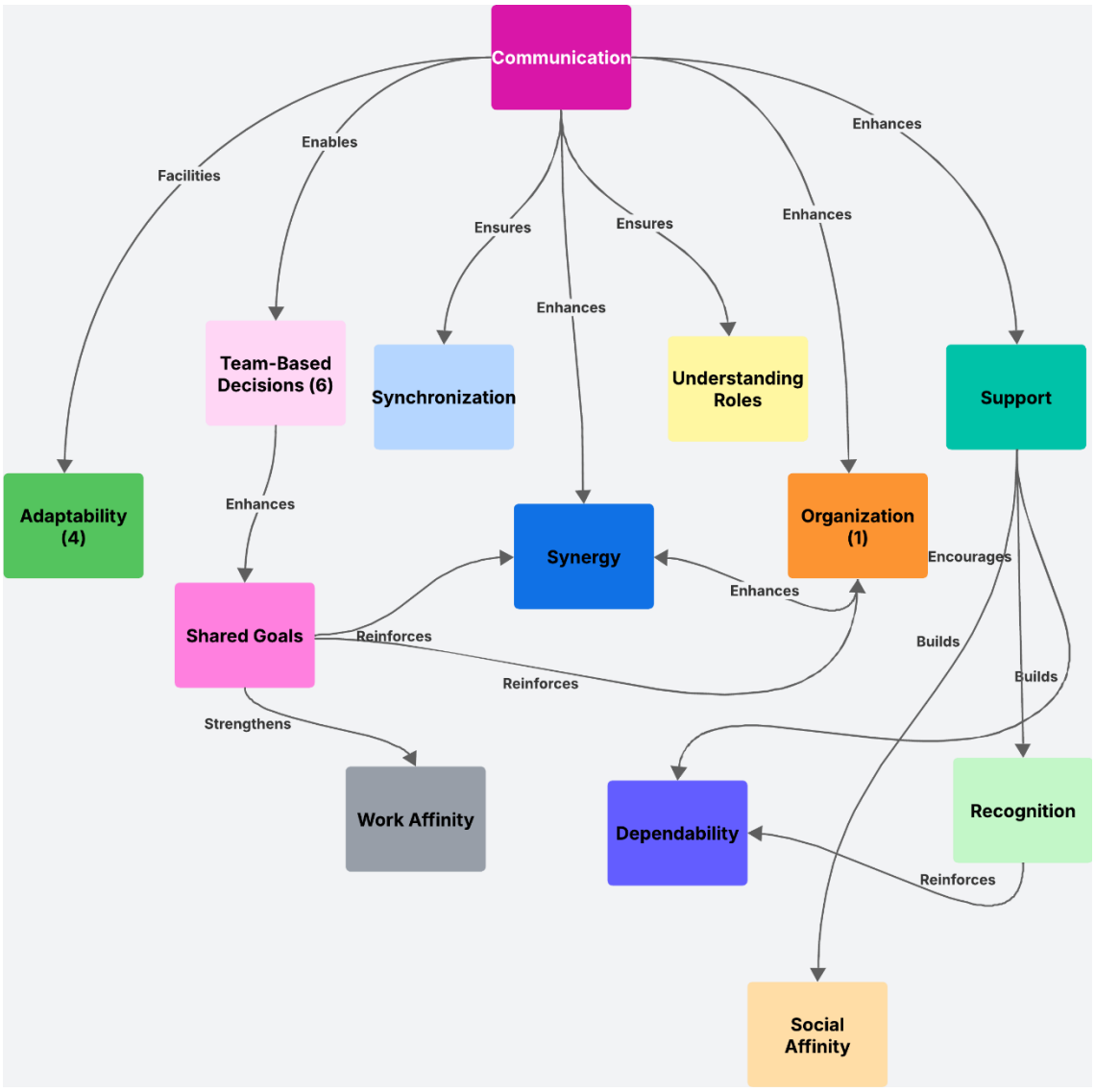

Fig. 2. Communication, shared goals, and support are the best related to other factors.

The identification of organization, understanding roles, synchronization, and leadership as structural and operational pillars of team cohesion yields important effects on software development management. Also, the establishment of clear organizational frameworks facilitates role definition, workload balance, and efficient coordination, all of which support external dependency management and effective crisis response. The promotion of role clarity among team members should foster mutual respect, dependability, and facilitates distributed leadership and collaborative decision-making. While, placing emphasis on workflow synchronization should help to reduce operational bottlenecks, enhance efficiency, and strength the team's capacity to adapt to evolving project requirements. Additionally, effective leadership has to be seen as essential not only for guiding task execution but also for nurturing a cohesive social environment. Leaders who balance strategic oversight with interpersonal engagement can better manage conflict resolution, encourage collaboration, and cultivate inclusive decision-making processes, all of which contribute to a more resilient and cohesive team structure.

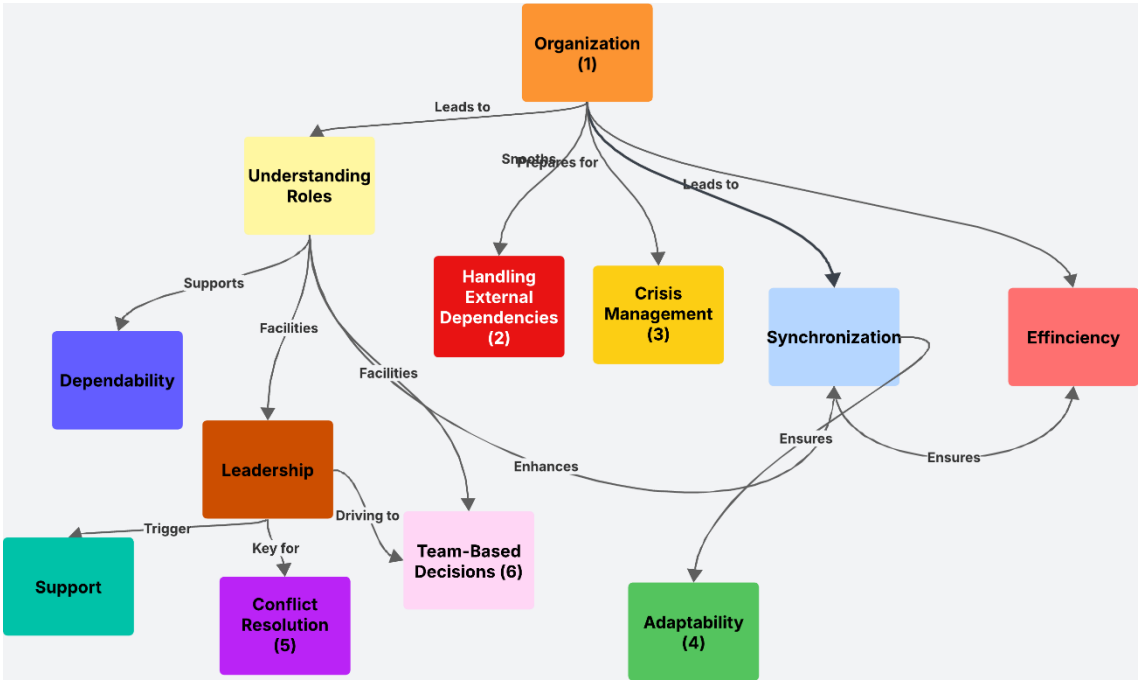

Fig. 3. Main factors associated with the structure and operation of the team: organization, understanding roles, synchronization, and leadership.

Closer associated with the operation are *crisis management* and *handling external dependencies,* see Fig. 4. As mentioned, *crisis management* requires *leadership*, and also *conflict resolution* to address unexpected issues, where *adaptability* is needed to generate different or new strategies. *Crisis management* also relies on *dependability* by trusting each other's work and judgment, and strong *communication* to prevent disruptions in software development.

Handling external dependencies relies on *communication* and *team-based decisions. Handling external dependencies* ensures that the team's work integrates smoothly with other systems, stakeholders, or regulatory requirements, and therefore, it can improve *efficiency*.

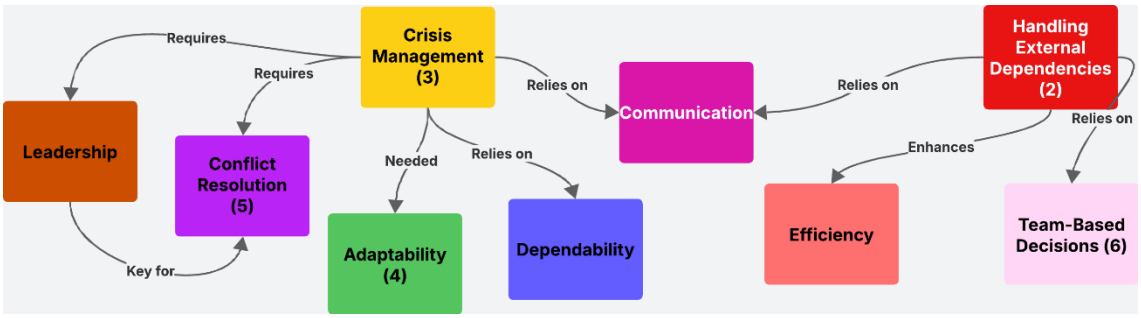

Fig. 4. Main factors associated with the operation of the team: crisis management and handling external dependencies.

Adaptability and *conflict resolution* can be seen as stability factors. *Adaptability* allows teams to respond to technical and organizational shifts, to ensure resilience in complex projects, while *conflict resolution*, the ability to resolve disputes constructively, supports *receptivity to feedback* and strengths *working affinity,* and *synergy*.

When the team members work as a team, *team-based decisions* encourage participation, also *working affinity* and *synergy* can be increased, leading to informed and accepted solution. But *leadership* has to be adaptive to ensure that decisions remain relevant in dynamic environments.

The operational dimensions of crisis management and handling external dependencies seem to have direct implications for maintaining cohesion and resilience in software development teams. Effective crisis management depends on strong *leadership*, *conflict resolution* skills, and *adaptability* to in time formulation of alternative strategies in response to unexpected challenges. It also requires a high level of *dependability*, where team members trust one another's judgment, and robust *communication* practices to minimize disruptions. Similarly, the management of external dependencies (e.g. integration with other systems, coordination with stakeholders, or compliance with external standards) heavily relies on clear *communication* and inclusive, *team-based decision-making*. These practices should ensure smoother collaboration across boundaries and improve overall efficiency. And, prioritizing operational capabilities should enhance the team's ability to respond cohesively to dynamic environments and maintain consistent delivery under complex conditions.

Other relations found were that *synergy, social affinity*, and *recognition* enhance collaboration. *Synergy* and *working affinity* drive motivation, increasing productivity and reducing interpersonal conflicts, that is, they support *conflict resolution*. *Synergy,* as mentioned, is enhanced when *team-based decisions* are made, and also when there is a proper *organization*, and *communication* alignment, allowing members to work seamlessly. While *social affinity* and *support* build trust, improving morale and *receptivity to feedback,* see Fig. 5.

From the factors that affect cohesion mentioned in the interviews, it can be observed that they lead to *work affinity, dependability, efficiency, receptivity to feedback, social affinity, and team-based decisions* as final goals to improve software development outcomes. It is important to stand out that this is not an exhaustive list of relations, but it highlights the complexity of the different factors that are somehow involved in cohesion. As mentioned, these connections or relations were found based on the obtained responses.

Finally, according to the developers' level of experience, the early career developers seem to understand cohesion as focused on interpersonal relationships, feeling supported, and with clear guidance; emotional safety and team inclusion were more visible concerns. The mid-level developers considered cohesion more related to both task and social aspects, valuing coordination and decision-making dynamics; they seem to look for a balance between team process and personal performance contribution. And cohesion for senior developers emphasized on systemic thinking, highlighting how cohesion improves workflow, crisis handling, and external integration; then their focus shifts toward maintaining long-term team performance and alignment, cohesion as a strategic asset for resilience and integration.

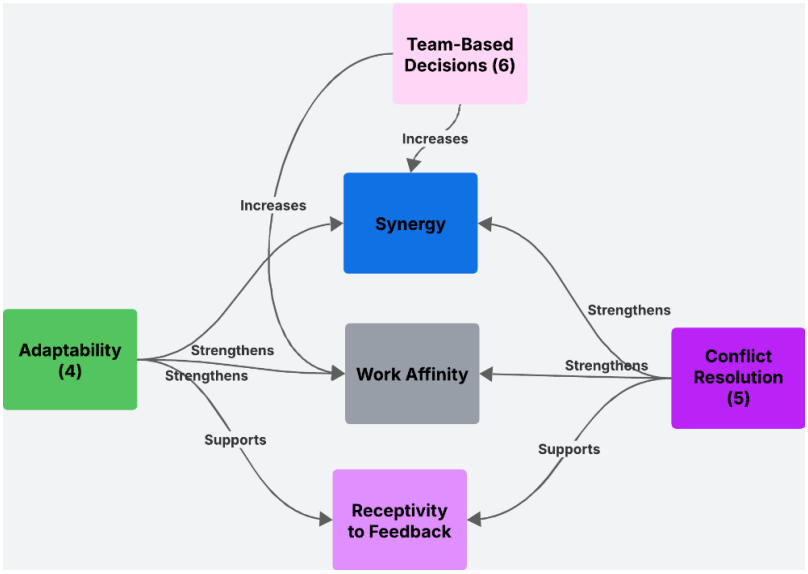

Fig. 5. Factors that improve synergy, work affinity, and receptivity to feedback.

5 Conclusions

While team cohesion is recognized as fundamental for productivity and success, the particularities of software development teams lead to get insights regarding the factors that determine it. In order to get an experience perspective, 137 software developers were consulted by a structured interview following the four proposed questions by Carron et al., (1985) about 1) the meaning of cohesion, 2) a situation reflecting cohesion, 3) incidents that denote low or lack of cohesion, and 4) what they consider factors that contribute for team cohesion, with open answers. The responses were analyzed by three expert project managers and categorized by question. Then the facts were merged into 19 key factors involved in team cohesion. In order to understand the relations among the 19 key factors, conceptual maps were constructed and the relations were rationalized.

The responses about the key factors that contribute to cohesion in software development teams emphasized their importance for efficiency, collaboration, and project success. Based on the responses of the software developers in the industry, communication, shared goals, and support were identified as the most recognized elements that enhance team cohesion**.** Also, the findings support that effective communication has a strong relation with cohesion, as it enables synchronization, ensures clarity in roles, and enhances collaboration. Organization and clear role definitions contribute to dependability and efficiency, ensuring that work is distributed fairly and executed effectively. Leadership and team-based decision-making promote engagement and help in managing crises, while synergy and social affinity seem to improve motivation and teamwork. Conversely, the study identifies barriers to cohesion, including poor communication, resistance to feedback, and lack of collaboration, which negatively impact team performance.

Based on these results, some points that team leaders might consider to enhance cohesion include fostering open and frequent communication; the implementations of programs with regular team meetings to promote transparency and synchronization; encourage feedback loops where team members feel included; present a clear definition for each team member's responsibilities and how they contribute to shared goals, here the use of visual tools can reinforce task clarity; provide feedback in a timely and actionable manner; openly receive feedback and adjust behavior accordingly; support emotional safety and recognition by acknowledging contributions publicly and individually: be attentive to interpersonal dynamics and offer support when tensions arise; act as a connector and not just as a decision-maker, bridge communication across team members, especially those working remotely or asynchronously; and, empower others to lead initiatives or decisions when appropriate.

Regarding onboarding practices, introduce new members to the team's communication rhythm by sharing preferred channels, and pair new hires with a mentor for integration; emphasize shared goals and team values early, explaining the current priorities along with the long-term vision; facilitate early participation in team-based decisions, allowing for building confidence and connection; and encourage social connection, and organize informal interactions.

For the work process design, it is important to consider build flexibility into workflows; include mechanisms for rotating roles or responsibilities to foster shared understanding; support coordination and synchronization through shared tools, and align task dependencies to minimize bottlenecks; encourage regular retrospectives to assess progress, and also to reflect on cohesion, collaboration, and morale; for crisis management use clear protocols and define roles during high-pressure situations; train team members in conflict resolution and decision-making under stress; and create mechanisms for recognition taking into consideration that junior members may need more feedback and encouragement, while senior members may value autonomy, trust, and acknowledgment of mentorship roles.

The findings suggest that fostering an open, structured, and supportive work culture can significantly improve team cohesion, leading to higher productivity, better software quality, and enhanced problem-solving capabilities. In general, the research reinforces cohesion as a dynamic and multifaceted process that requires a balance between technical efficiency and social interaction. Future work should explore how these factors evolve in different team structures and software development methodologies to further optimize team performance.

References

Bosselut, G., Heuzé, J. P., Castro, O., Fouquereau, E., & Chevalier, S. (2018). Using Exploratory Structure Equation Modeling to validate a new measure of cohesion in the university classroom setting: The University Group Environment Questionnaire (UGEQ). *International Journal of Educational Research, 89, 1-9.* https://doi.org/10.1016/j.ijer.2018.03.003

Castro-Hernández, A., Swigger, K., & Ponce-Flores, M. P. (2014, October). Effects of cohesion-based feedback on the collaborations in global software development teams. In *10th IEEE international conference on collaborative computing: Networking, applications and Worksharing* (pp. 74-83). IEEE.

Carron, A. V., Widmeyer, W. N., & Brawley, L. R. (1985). The development of an instrument to assess cohesion in sport teams: The Group Environment Questionnaire. *Journal of Sport and Exercise psychology*, *7*(3), 244-266.

Chun, W. H. K., Soon, W., Wardrip-Fruin, N., & Zhu, J. (2022). Software Studies, Revisited. A Roundtable on the Software Studies Series at MIT Press. *Computational Culture*, *9*.

Dey, C., & MP, G. (2020). Impact of team design and technical factors on team cohesion. *Team Performance Management: An International Journal*, *26*(7/8), 357-374.

Do Nascimento Junior, J. R. A., Vieira, L. F., Rosado, A. F. B., & Serpa, S. (2012). Validation of the Group Environment Questionnaire (GEQ) for Portuguese language. *Motriz: Revista de Educação Física*, *18*, 770-782.

Hanrahan, S., & Gallois, C. (1993). Social interactions. In Singer, R.N., Murphy M. & Tennant L. (eds.) Handbook of research on sport psychology, 623-646. New York: MacMillan.

Hoda, R., Noble, J., & Marshall, S. (2012). Developing a grounded theory to explain the practices of self-organizing Agile teams. *Empirical Software Engineering*, *17*(6), 609-639.

Iturbide, L. M., Elosua, P., & Yanes, F. (2010). A measure of team cohesion in sport. Spanish adaptation of Group Environment Questionnaire (GEQ). *Psicothema*, *22*(3), 482-488.

Jiménez, M., Piattini, M., & Vizcaíno, A. (2009). Challenges and improvements in distributed software development: A systematic review. *Advances in Software Engineering*, *2009*(1), 710971.

Kakar, A., & Kakar, A. K. (2018). Is team cohesion a double edged sword for promoting innovation in software development projects?. *Pacific Asia Journal of the Association for Information Systems*, *10*(4), 5.

Kakar, A. K. S. (2018). Engendering cohesive software development teams: Should we focus on interdependence or autonomy?. *International Journal of Human-Computer Studies*, *111*, 1-11. https://doi.org/10.1016/j.ijhcs.2017.11.001

Muñoz, M., Hernández, L., Mejia, J., Peña, A., Rangel, N., Torres, C., & Sauberer, G. (2017). A model to integrate highly effective teams for software development. In *Systems, Software and Services Process Improvement: 24th European*

Nootjarat, R., Chantatub, W., & Chongstitvatana, P. (2015). The moderating effect of leader centrality on team cohesion and performance in software development projects. *International Journal of Business and Information*, *10*(3), 295-322.

Peña Pérez Negrón, A., Del Rosario T.M., Bonilla Carranza, D., & Hérnandez de León, J.E. (in press). Exploring cohesion factors in undergraduate software development teams. In Mejía, J., Muñoz, M., Rocha, A., Espinosa-Faller, F.J. & Trejo-Sanchez, J.A. (eds.) *New Challenges in Software Engineering.* Springer International Publishing.

Prokešová, E., Musálek, M., & Chalupová, E. (2012). A comparıson of perceıved team cohesıon ın athletes of ınteractıve and coactıve sports based on the group envıronment questıonnaıre. *Acta Universitatis Carolinae: Kinanthropologica*, *48*(2), 138-145.

Ryan, S., & O'Connor, R. V. (2009). Development of a team measure for tacit knowledge in software development teams. *Journal of Systems and Software*, *82*(2), 229-240.

Sawyer, S. (2004). Software development teams. *Communications of the ACM*, *47*(12), 95-99.