WORLD COMET RESEARCH GROUP

International Journal of Artificial Intelligence & Digital Transformation
Vol. 1, No. 2, 2018
Doi: 10.67228/30713315/IJAIDT-2018PII4R3C
PP. 01-12

Original Article

Combining Augmented Reality (AR) with Software Engineering for Remote Collaboration

Chen Ming[1], Wang Li[2]

[1]*Software Architect, Huawei, China.*

[2]*Data Science Manager, Alibaba Group, China.*

Received: 06-03-2018 Revised: 06-24-2018 Accepted: 07-03-2018 Published: 07-05-2018

ABSTRACT

This paper explores the integration of Augmented Reality (AR) with software engineering practices to enhance remote collaboration among distributed teams. As remote work becomes increasingly common, software engineering teams face challenges in maintaining effective communication, real-time problem-solving, and collaboration. AR technologies offer innovative solutions to bridge these gaps by creating immersive, shared virtual environments that enable engineers to collaborate seamlessly, regardless of geographic location. This paper discusses the potential of AR to transform remote software development, focusing on its applications, benefits, challenges, and real-world examples. By investigating the intersection of AR and software engineering, the paper provides insights into how this emerging technology can improve efficiency and collaboration in the future of remote work.

KEYWORDS

Augmented Reality (AR), Remote Collaboration, Software Engineering, Distributed Teams, Immersive Collaboration, Virtual Environments, Real-Time Problem Solving, Remote Work Solutions.

ISSN: 3071-3315
PP: 01-12.

1. INTRODUCTION

The global shift toward remote work, accelerated by the COVID-19 pandemic, has made remote collaboration an essential part of the modern workforce, particularly in software engineering. As teams become geographically dispersed, traditional methods of communication and collaboration are being tested, often revealing gaps in efficiency, coordination, and effectiveness. Remote work introduces several challenges, including the difficulty of fostering real-time collaboration, maintaining team cohesion, and overcoming communication barriers. For software engineering teams, these challenges are compounded by the need for continuous problem-solving, debugging, and real-time coordination in complex projects. As a result, there is a growing need for innovative solutions that can bridge the physical divide between team members and improve the collaboration experience.

Among these solutions, Augmented Reality (AR) has emerged as a promising technology that can significantly enhance remote collaboration. AR enables users to overlay digital content onto the real world, creating immersive, interactive experiences that blend virtual and physical environments. In software engineering, this can translate into more engaging, intuitive, and effective ways for teams to work together remotely. AR has the potential to transform how engineers visualize code, debug systems, and collaborate in real-time, making remote work feel more like being in the same physical space. This paper explores the integration of AR with software engineering, examining how it can improve remote collaboration by offering interactive, real-time problem-solving tools, virtual environments, and collaborative spaces.

The purpose of this paper is to explore the intersection of AR and software engineering, focusing on how AR can be used to enhance remote collaboration within distributed teams. The paper aims to investigate the potential benefits, challenges, and real-world applications of AR in software engineering, providing insights into how this emerging technology can shape the future of remote work in the field.

2. BACKGROUND AND RELATED WORK

Remote collaboration in software engineering has been a topic of research for decades, particularly as teams have become more distributed due to advancements in communication technologies and globalization. Traditional collaboration tools such as video conferencing, instant messaging, and version control systems have been foundational in supporting remote teams. However, these tools often lack the immersive, interactive elements needed to truly replicate the experience of working together in a shared physical space. As a result, researchers and practitioners have been exploring more innovative solutions, such as virtual reality (VR) and AR, to enhance remote collaboration in software engineering.

Existing research has explored various aspects of remote collaboration, including the design of collaborative software development environments, real-time communication tools, and systems for collaborative debugging and code review. Many studies have highlighted the limitations of current

ISSN: 3071-3315
PP: 01-12.

tools, especially in terms of real-time feedback and visual interaction. While tools like screen sharing and code commenting facilitate communication, they often fail to provide a truly interactive or immersive experience. For software engineers, the lack of a shared visual space can hinder problem-solving and slow down the development process.

AR technologies, which allow users to interact with digital content in real-time and in the context of their physical environment, have the potential to address many of these limitations. AR can provide an intuitive way to represent complex software systems, visualize code in 3D, and overlay digital annotations onto physical objects, such as hardware or development environments. This ability to create shared, interactive visual spaces could significantly enhance collaboration, as engineers could work together in virtual spaces that feel more tangible and engaging than traditional methods.

In other industries, AR has already shown its potential to improve collaboration and productivity. In healthcare, for example, AR is being used for remote surgeries, where a surgeon in one location can assist a surgeon in another by providing real-time visual feedback and guidance. In manufacturing, AR is enhancing remote troubleshooting by overlaying digital instructions onto physical equipment, allowing workers to interact with the system in real time. These applications demonstrate how AR can transform collaborative work by offering immersive, context-aware solutions that bridge the gap between virtual and physical environments.

While AR has yet to be widely adopted in the software engineering field, there have been promising early-stage implementations. Some projects have explored using AR for collaborative code review, where developers can visualize code in a shared virtual space, interact with it, and offer real-time feedback. Other studies have looked at how AR can assist in debugging, allowing engineers to visualize system components and interactions in a more intuitive, interactive manner. These examples provide a glimpse into the potential of AR for software engineering, suggesting that the technology could improve how developers collaborate, solve problems, and understand complex systems.

In conclusion, while remote collaboration tools have come a long way, there remains a significant opportunity to enhance these tools with AR. Drawing from research in other industries and early experiments in software engineering, this paper explores the potential of AR to transform the way software engineering teams collaborate remotely, offering new methods of visualization, interaction, and real-time problem-solving.

3. THE ROLE OF AUGMENTED REALITY IN SOFTWARE ENGINEERING

3.1. How AR can be integrated into the software development process

Augmented Reality (AR) offers a transformative potential for the software development process by enhancing how software engineers interact with code, collaborate on problem-solving, and visualize complex systems. AR can be integrated into the development cycle in several key areas:

ISSN: 3071-3315
PP: 01-12.

design, coding, testing, debugging, and collaboration. One way AR can be integrated is by providing software developers with a virtual environment that displays code structures, system flows, and dependencies in 3D or interactive formats, allowing engineers to visualize and interact with the software in a more intuitive way. AR can also enhance collaboration by enabling distributed teams to interact with the same virtual environment, helping them work together on the same project as if they were co-located. For instance, developers can use AR to place virtual code annotations, draw connections between code modules, or demonstrate system behavior, all of which can be more easily understood than traditional 2D representations. This integration allows for more immediate problem identification, faster iterations, and enhanced collaboration across team members working remotely.

Table 1: Benefits and Challenges of AR Adoption in Software Engineering

Aspect	Benefits	Challenges
Visualization	Better understanding of complex systems	High computational requirements
Collaboration	Enhanced remote teamwork	Network latency issues
Debugging	Faster fault identification	Tool integration complexity
Training	Improved learning experience	Initial learning curve
Productivity	Reduced development time	Hardware costs
User Experience	Immersive interaction with software artifacts	AR device limitations
Decision-Making	Improved situational awareness	Scalability concerns

3.2. Specific examples of AR applications in software engineering

In the context of software engineering, AR applications are already being explored in various ways. One notable example is collaborative coding environments, where AR technology allows multiple developers to interact with a shared, virtual code space. In this environment, developers can view and edit the same code simultaneously, while also seeing each other's actions represented in real-time as holographic annotations or visual overlays. This allows for a much more interactive experience compared to traditional video conferencing and screen-sharing tools. Another powerful application of AR in software engineering is virtual debugging. With AR, engineers can create virtual overlays that represent system components or processes and track their interactions in real-time. For example, when debugging a complex distributed system, developers can use AR to visualize the flow of data between components, which would be difficult to grasp in a traditional flat interface. By doing so, AR enables developers to pinpoint bugs more easily, understand system behaviors intuitively, and reduce the time spent identifying issues. These applications demonstrate how AR can enhance the day-to-day tasks of software engineers and improve productivity.

3.3. Interaction models of AR for real-time collaboration

In AR, interaction models play a crucial role in enabling real-time collaboration between remote software engineers. One of the key models is the shared virtual space, where multiple engineers can work together in the same augmented environment, even if they are physically located in different places. Through this model, participants can share a 3D representation of the software system, where code, system behaviors, and project elements are presented as interactive, tangible objects within the virtual space. Engineers can move around these objects, interact with them, and manipulate them as though they were real-world objects, providing a more intuitive and immersive

ISSN: 3071-3315
PP: 01-12.

way to collaborate. Another compelling interaction model is the use of holographic representations of code. This model allows developers to see code structures, dependencies, and data flows as 3D holograms projected into their real-world environment, making it easier to visualize relationships and potential issues. By using hand gestures or voice commands, developers can manipulate these holograms in real time, streamlining collaboration and decision-making. These interaction models allow engineers to break free from traditional 2D interfaces, fostering more engaging and effective real-time collaboration.

4. BENEFITS OF AR FOR REMOTE SOFTWARE ENGINEERING COLLABORATION

4.1. Enhanced communication through shared visual representations

One of the most significant benefits of AR in remote collaboration is the enhancement of communication through shared visual representations. Traditional remote collaboration tools, such as video conferencing or screen-sharing, often rely on text and voice to convey ideas, which can lead to misinterpretations or confusion, especially in complex software engineering tasks. AR overcomes these limitations by enabling engineers to interact with digital content in a way that's visually intuitive and immersive. With AR, team members can create visual markers, annotations, and interactive diagrams that represent complex systems, making it easier for all participants to understand the current state of the project. For example, engineers working on a piece of code can use AR to highlight sections that need attention, demonstrate how different modules are connected, or visually represent system errors, all of which foster clearer and more effective communication between remote team members.

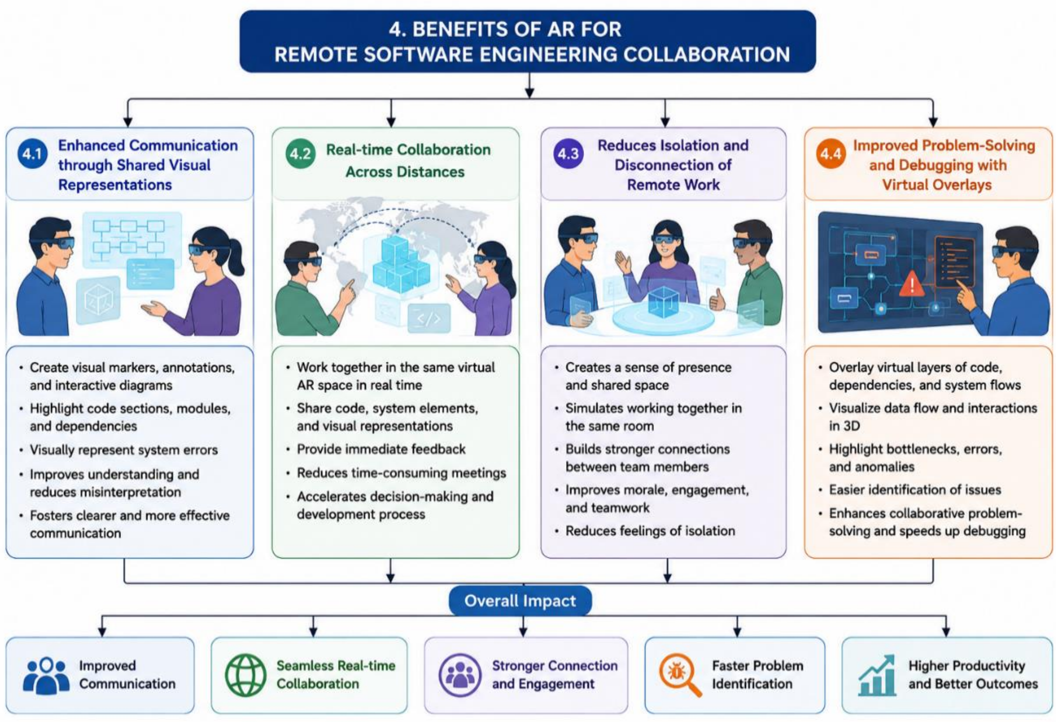

Fig 1 – Benefits of AR for Remote Software Engineering Collaboration

4.2. Real-time collaboration across distances

AR also facilitates real-time collaboration across distances, a critical need in the modern software engineering landscape. As teams become more distributed globally, maintaining seamless communication and coordination becomes increasingly difficult. AR allows engineers in different locations to work together in real-time within the same virtual space, enabling them to share visual representations, interact with code or system elements, and provide immediate feedback. For example, a developer in one country might be able to demonstrate a coding issue in a virtual environment while a colleague in another location offers solutions, all within the same immersive AR space. This capability of real-time interaction across distances reduces the need for time-consuming meetings, ensures quicker decision-making, and accelerates the overall development process.

4.3. The potential to reduce the isolation and disconnection of remote work

Remote work, while offering flexibility and cost savings, can often lead to feelings of isolation and disconnection among team members, which can hinder collaboration and productivity. AR addresses this by fostering a sense of presence and shared space. Unlike video calls that can feel disconnected or flat, AR creates an environment where team members can virtually "stand" next to one another and engage in collaborative tasks as though they were physically together. This presence reduces the emotional and social isolation that often accompanies remote work, fostering stronger connections between team members. By simulating the experience of working in the same room, AR creates an engaging and supportive environment that helps engineers feel more connected to each other, which can lead to improved morale, better teamwork, and more effective collaboration.

4.4. Benefits of AR in problem-solving and debugging through virtual overlays

Another key benefit of AR in remote software engineering collaboration is its ability to assist in problem-solving and debugging through virtual overlays. Debugging complex software systems can often be a challenging task, requiring engineers to trace through layers of code, dependencies, and system interactions. With AR, engineers can overlay virtual representations of these layers directly onto the physical world or into a virtual space, offering a more intuitive understanding of system behavior. For instance, when debugging a multi-layered application, engineers can use AR to see the data flow between components as animated 3D paths, highlighting potential bottlenecks or errors. By interacting with these visual overlays, engineers can spot problems more easily and collaboratively work with teammates to resolve them. This not only accelerates the debugging process but also enhances problem-solving by making abstract concepts tangible and accessible.

In summary, AR offers numerous advantages in the context of remote software engineering collaboration. By providing shared visual representations, facilitating real-time interactions, reducing isolation, and enhancing debugging capabilities, AR can help transform the way remote software teams collaborate, solve problems, and build software more effectively. These benefits represent just the beginning of how AR can reshape the landscape of software engineering, offering new possibilities for improved productivity and collaboration.

ISSN: 3071-3315
PP: 01-12.

5. CHALLENGES IN IMPLEMENTING AR FOR REMOTE COLLABORATION

5.1. Technical challenges (e.g., hardware requirements, network latency)

The integration of Augmented Reality (AR) into remote software engineering collaboration comes with a variety of technical challenges. One of the most prominent issues is hardware requirements. AR typically requires specialized hardware, such as AR glasses, headsets, or high-performance computing devices, which can limit its accessibility. These devices are not yet universally available or affordable, especially for smaller teams or companies, making it difficult for many to adopt AR-based tools. Additionally, the development and implementation of AR applications often require significant computing power to process real-time visuals and complex interactions. As a result, team members working on lower-end devices might face performance issues, such as lag or graphical glitches, which would disrupt the collaboration process.

Another significant technical challenge is network latency. AR applications that rely on real-time collaboration require a stable and high-speed internet connection. Remote collaboration using AR demands that the data and visual representations be transmitted without noticeable delay, ensuring that users' interactions with the virtual environment remain seamless. However, network latency can hinder this experience, causing delays in updating visual elements or displaying changes made by other collaborators. In areas with unreliable internet infrastructure, such as remote or rural locations, network latency may significantly degrade the quality of the AR experience, leading to frustration and decreased productivity.

5.2. Human factors (e.g., learning curve, user adoption)

In addition to technical challenges, human factors play a critical role in the implementation of AR for remote collaboration. One of the primary human challenges is the learning curve associated with AR tools. While these tools offer innovative and immersive features, they often come with complex interfaces and new methods of interaction that users may find difficult to master. For software engineers, accustomed to traditional coding environments and collaboration tools, adapting to AR's new interaction models—such as manipulating 3D objects, navigating virtual spaces, or using hand gestures can be a time-consuming and frustrating experience. Overcoming this learning curve requires training, support, and a period of adjustment, which may initially slow down productivity.

Another human factor is user adoption. Even if the technical aspects of AR are implemented successfully, getting team members to adopt these tools is not guaranteed. Employees may resist changing from their familiar, traditional methods of collaboration, especially if they feel that AR is unnecessary or difficult to use. Additionally, some engineers may be concerned about the potential distractions or fatigue associated with using AR, particularly when it involves wearing headsets or glasses for long periods. To drive successful adoption, it is essential that AR tools are intuitive, user-friendly, and demonstrably beneficial to the engineers' workflow. If the tools fail to meet these criteria, they may not be widely embraced, and their potential for improving collaboration could go unrealized.

ISSN: 3071-3315
PP: 01-12.

5.3. Cost and scalability concerns

The cost of implementing AR in a remote collaboration environment is another barrier to widespread adoption. The specialized hardware required for AR, such as AR glasses, motion sensors, and high-performance computers, can be expensive, especially when purchasing them for an entire team or organization. These costs are further exacerbated by the need for specialized software and applications designed for AR collaboration. For smaller companies or startups, the financial investment needed for AR implementation may be prohibitive, particularly if the technology's return on investment is not immediately clear. Additionally, the cost of maintaining AR tools, including updating hardware and software, may place a long-term financial burden on the organization.

Scalability is another concern when it comes to implementing AR for remote collaboration. As teams grow in size, the complexity of managing AR tools and ensuring smooth interactions among multiple users increases. For example, AR collaboration tools need to support a large number of users interacting in the same virtual environment, each with different devices and connection speeds. Ensuring that the system remains responsive and functional for a large, diverse team requires significant infrastructure and careful design. The challenge is compounded when teams are geographically dispersed, as network issues and hardware inconsistencies become more pronounced. Scaling up AR collaboration tools while maintaining a high-quality user experience is a complex task that requires ongoing investment in infrastructure and technological development.

6. CASE STUDY: IMPLEMENTATION OF AR IN A REMOTE SOFTWARE ENGINEERING TEAM

6.1. A detailed example of a team implementing AR-based collaboration tools

One company that successfully implemented AR-based collaboration tools was a remote software engineering team working on a complex, multi-tiered application for the healthcare industry. The team was geographically distributed, with members spread across three continents. They faced challenges in maintaining clear communication and collaboration, especially when reviewing complex code or discussing system design. In an effort to improve the remote collaboration process, the team adopted AR tools that allowed them to visualize code in 3D and interact with virtual system components. The tools allowed developers to see virtual overlays of the system architecture, with each component displayed as an interactive, 3D model. This helped the team understand the overall structure and functionality of the application in a more intuitive manner, which was particularly useful for debugging and identifying potential issues in the system.

6.2. Challenges faced and solutions adopted

The implementation of AR in this team was not without its challenges. One major hurdle was ensuring that all team members had access to compatible hardware. Some developers were using older computers that lacked the processing power to handle the AR tools effectively, leading to performance issues. To address this, the team invested in upgrading hardware for key members, while others used simplified versions of the AR tools, designed for lower-end devices. Another challenge was the learning curve associated with the AR software. Initially, engineers were hesitant to adopt the technology due to unfamiliar interfaces and difficulty in using the 3D visualization tools. The team responded by providing training sessions and user guides to help engineers become familiar with the new tools. These efforts paid off as developers began to see the benefits of the AR tools in terms of increased collaboration efficiency and improved understanding of complex system components.

6.3. Results and feedback from team members

The implementation of AR tools resulted in noticeable improvements in the team's collaboration process. The 3D visualization of code and system architecture helped reduce misunderstandings and fostered more effective discussions during code reviews. Developers were able to point out issues in real-time by placing virtual markers on specific code sections or components, making debugging much more efficient. One engineer noted that the ability to visualize complex interactions between system components in an AR environment significantly improved their ability to troubleshoot and fix bugs quickly. Additionally, the immersive nature of the AR tools helped mitigate the isolation that some team members felt working remotely, as they could virtually "stand" next to each other and work on the same tasks in a shared space. Despite initial challenges, feedback from team members was overwhelmingly positive, with many expressing excitement about the future potential of AR in improving remote collaboration. The team reported increased productivity and a greater sense of connection between team members, despite the geographic distance.

In conclusion, while there are several challenges to implementing AR for remote software engineering collaboration—such as technical requirements, human factors, and cost—real-world examples show that these challenges can be overcome through thoughtful planning and support. With the right tools, training, and infrastructure, AR can significantly enhance collaboration, reduce isolation, and improve problem-solving in distributed teams.

7. FUTURE DIRECTIONS

7.1. Emerging trends in AR and software engineering collaboration

The field of Augmented Reality (AR) continues to evolve, and new trends are emerging that promise to further enhance remote collaboration in software engineering. One key trend is the shift toward more immersive AR experiences, using sophisticated devices like mixed reality (MR) headsets and spatial computing environments. These tools allow for greater interactivity, enabling software engineers to manipulate virtual objects and code structures in 3D space as if they were real. As AR technology becomes more refined, we can expect to see the integration of AR with more advanced collaborative platforms that support not only real-time visualizations but also real-time coding environments. Another emerging trend is the increased focus on mobile AR, where developers use smartphones and tablets as AR tools for remote collaboration. As AR becomes more portable and accessible, software engineering teams can collaborate without requiring expensive, specialized equipment, which could democratize its use in the industry. Additionally, AR tools are likely to integrate more closely with existing software development environments, such as Integrated

Development Environments (IDEs) and version control systems, allowing for a seamless, AR-augmented workflow.

7.2. Possible developments in AR hardware and software that could enhance collaboration

The development of AR hardware is likely to continue advancing, with more lightweight, comfortable, and affordable devices expected to enter the market. Future AR headsets and glasses may become more ergonomic, allowing developers to wear them for longer periods without discomfort. This will address one of the current barriers to widespread adoption: the physical strain of wearing AR devices during extended work sessions. In addition, improvements in battery life and wireless connectivity will allow for more efficient and immersive experiences, reducing the risk of lag and providing a more consistent and reliable AR environment. On the software side, more advanced AR development platforms will emerge, offering increased functionality, ease of use, and customization. These platforms could include improved natural language processing (NLP) and gesture recognition, allowing for more intuitive interactions with virtual elements. Furthermore, software tools designed for collaborative development will likely begin to incorporate AR features more seamlessly, with better support for real-time communication, shared workspaces, and virtual whiteboards. The combination of enhanced hardware and software will undoubtedly improve how remote teams collaborate using AR in software engineering, making it a more scalable and accessible solution.

7.3. The role of AI and machine learning in augmenting AR-based remote collaboration

Artificial Intelligence (AI) and machine learning (ML) will play a crucial role in the future of AR for remote collaboration in software engineering. One major way in which AI can enhance AR is by providing real-time, context-sensitive assistance to developers. For example, AI could analyze a developer's interactions within an AR environment and suggest optimizations or highlight areas of code that might benefit from further investigation, all in real-time. Machine learning algorithms could also be used to personalize the AR experience, adapting the interface and interactions based on the user's preferences or workflow habits. Additionally, AI can improve collaboration by enabling automatic translation of communications, facilitating cross-linguistic collaboration among global teams. Another important area is AI-powered debugging, where machine learning algorithms could help analyze code in an AR environment, providing intelligent suggestions for potential bugs or areas that need improvement. As AI and AR converge, we can expect more sophisticated, intelligent systems that will make remote software engineering collaboration more efficient, intuitive, and effective.

8. CONCLUSION

8.1. Recap of the benefits of AR in remote software engineering collaboration

Augmented Reality (AR) has the potential to revolutionize remote software engineering collaboration by providing immersive, interactive experiences that improve communication, problem-solving, and teamwork. AR enables software engineers to visualize complex code structures, system architectures, and debugging processes in 3D, offering an intuitive and engaging way to understand and manipulate software. The ability to work in shared virtual spaces, interact with holographic representations of code, and collaborate in real-time across distances allows engineers to overcome the limitations of traditional remote collaboration tools. As a result, AR fosters more effective communication, reduces the isolation of remote work, and accelerates development cycles by enhancing teamwork and decision-making.

8.2. A summary of the key challenges and how they might be overcome

While the benefits of AR for remote collaboration are clear, several challenges remain. Technical issues such as hardware requirements, network latency, and compatibility with different devices can hinder the widespread adoption of AR tools. These challenges can be addressed by continuing to develop more affordable, lightweight, and compatible AR hardware, along with improving network infrastructure to support real-time collaboration. On the human side, the learning curve associated with new AR tools and resistance to adopting unfamiliar technology can be overcome through proper training, intuitive design, and demonstrating the clear value AR brings to daily workflows. Lastly, cost and scalability concerns can be alleviated by reducing the cost of AR hardware, increasing the accessibility of AR tools, and improving the efficiency of AR software to accommodate larger, distributed teams. As AR technology matures, these obstacles are likely to diminish, paving the way for more widespread and effective use of AR in software engineering collaboration.

8.3. Final thoughts on the future of remote collaboration in the software engineering field

The future of remote collaboration in software engineering is undoubtedly exciting, and AR is poised to play a pivotal role in shaping this future. As AR technology continues to advance, it will offer new opportunities for software engineering teams to collaborate more effectively, creatively, and efficiently, regardless of their physical location. While challenges remain, the continued development of AR hardware and software, along with the integration of AI and machine learning, will likely address many of the current obstacles. The growing trend of remote work, combined with advancements in AR, suggests that in the near future, teams will be able to work together in more immersive, intuitive, and productive ways than ever before. AR has the potential to redefine the boundaries of remote collaboration, creating a new paradigm for software engineering that could make distributed teams feel as connected and effective as co-located ones. As we look ahead, AR's role in software engineering collaboration seems not only promising but essential for the continued evolution of the industry.

REFERENCES

[1] Lukosch, S., Billinghurst, M., Alem, L., & others. (2015). *Collaboration in Augmented Reality*. Computer Supported Cooperative Work (CSCW), 24, 515-525. SpringerLink+1

[2] Barakonyi, I., Fahmy, T., & Schmalstieg, D. (2004). *Remote collaboration using Augmented Reality Videoconferencing*. Proceedings of Graphics Interface 2004, 89-96. graphicsinterface.org

[3] Lukosch, S., Billinghurst, M., Alem, L., & Kiyokawa, K. (2015). Collaboration in augmented reality. *Computer Supported Cooperative Work (CSCW), 24*(6), 515–525.

[4] Barakonyi, I., Fahmy, T., & Schmalstieg, D. (2014). Remote collaboration using augmented reality videoconferencing. *Proceedings of IEEE International Symposium on Mixed and Augmented Reality*.

[5] Peake, I. D., Blech, J. O., Nath, S., Aharon, J. J., & McGhie, A. (2017). Towards a cloud-based architecture for visualization and augmented reality to support collaboration in manufacturing automation. *arXiv preprint arXiv:1711.05997*.
[6] Billinghurst, M., Clark, A., & Lee, G. (2015). A survey of augmented reality. *Foundations and Trends in Human–Computer Interaction, 8*(2–3), 73–272.
[7] Alem, L., & Huang, W. (2011). Recent trends of mobile collaborative augmented reality systems. *Springer Handbook of Augmented Reality*, 657–688.
[8] Green, S. A., Billinghurst, M., Chen, X., & Chase, J. G. (2008). Human-robot collaboration: A literature review and augmented reality approach. *International Journal of Advanced Robotic Systems*.
[9] Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments, 6*(4), 355–385.
[10] Kiyokawa, K., Billinghurst, M., Hayes, S. E., & Gupta, A. (2002). Communication behaviors of co-located users in collaborative AR interfaces. *Proceedings of IEEE ISAR*.
[11] Billinghurst, M., & Kato, H. (2002). Collaborative mixed reality. *Proceedings of International Symposium on Mixed and Augmented Reality (ISMAR)*.