International Journal of
Technology and
Systems
(IJTS)

An In-Depth Analysis of the Historical Progression and Future Trends in Source Control Management within Software Development

Journal of Technology and Systems
ISSN : 2788-6344 (Online)
Vol. 6, Issue No. 1, pp 13 – 24, 2024
www.carijournals.org

An In-Depth Analysis of the Historical Progression and Future Trends in Source Control Management within Software Development

Sridhar Mooghala

Fiserv

Senior Advisor

https://orcid.org/0009-0007-9959-1830

Accepted: 12th Dec 2023 Received in Revised Form: 26th Dec 2023 Published: 10th Jan 2024

Abstract

Purpose: This research looks closely at the history and future directions of Source Control Management (SCM) when making software. This research aims to explain how SCM has changed over time. It looks at its important moments and guesses where it might progress next.

Methodology: The study uses a solid method to review the history books and update them with current trends and new things that are changing the subject. Research shows how SCM has changed over time, from its first versions to the advanced distributed control systems used today. The research looks at important things that affect how supply chain management grows. These include the need for teamwork, wanting to handle big workloads, and more use of DevOps methods. The study shows new patterns like decentralized SCM models and joining with Continuous Integration/Continuous Deployment (CI/CD) pipes.

Findings: The exceptional help to the idea is in joining past learning with a future look. This gives a detailed view of how SCM changes over time. This study is suitable for software makers, builders and leaders. Giving important information helps they make intelligent choices when using SCM tools. The advice about rules shows that all companies should follow the same rules. This highlights how SCM can make computer programs better, work together more efficiently and be quicker.

Unique contributor to theory, policy and practice: In the end, this study helps us better understand SCM's history and how it might change in the future. It gives a complete view useful for work practices and education discussions.

Keywords: *GitOps, Source Control Management, Distributed Version Control Systems, Centralized Version Control System, GitHub, Continuous Integration/Continuous Deployment pipes.*

Introduction

Developing Source Control Management (SCM) in software creation is changing [1]. It has changed chiefly business coding methods. In the beginning, SCM had manual versions. Over time, it developed into more complex Distributed Version Control Systems (DVCS) like Git, changing how people work together on coding projects. Supply chain management has been very important. This review contains a theory study that fully understands SCM's history. It explains where it came from and points out significant changes that have helped it grow. In a time of increasing technology changes, the theory review looks ahead to new things and improvements that changed how software groups manage their work.

This is important because it shows how important it is for people, teams, and companies to understand supply chain management's changing nature and history to improve their methods.

Source Control Management Historical Progression

SCM is essential for making software. It helps manage teamwork and ensures that project plans stay the same [2]. This article, rich with theory, looks at the new things and changes that have shaped history in SCM. It looks at how code management went from hand to more advanced systems like Git. It shows that SCM is significant for making group coding practices better.

The theory viewpoint is essential for understanding the basic ideas behind SCM. It helps us know where these ideas came from and gives us a deeper look at new patterns in software creation linked to source control.

1. Early Concepts and Versioning

The research design incorporates a historical perspective, delving into the basic concepts and versioning practices of source control management (SCM) [3] starting in the 1970s and 1980s. The target population is driven by developers engaged in coding practices during this period. Data collection uses archival research, interviews with prominent professionals of the era, and purposive sampling to select individuals with significant contributions. Thematic analysis is applied to data about characteristics, reveals the complexities and innovations of basic translation processes, and contributes to a nuanced understanding of SCM's evolution.

2. Centralized Version Control Systems

This study uses history. It looks at how Centralized Version Control Systems (CVCS) changed and had problems during the 80s and 2000s. The plan wants to give a complete study of the reasons, steps, and then problems that led CVCS towards more adaptable control systems for changes. The study is about people who make software, those in charge of projects, and groups using CVCS during the given time. [4] The focus is on workers dealing with complicated storage, connection and teamwork issues. This mainly applies to big companies. They use both research from old records and talking to people. Research in archives looks at old records, books and instructions linked to the growth and use of CVCS. Talking to important people who use CVCS gives us a close look at the problems they face and why they choose to use it. Picking people is used to choose

those with important experience and contributions to CVCS planning. Important people like builders and project leaders facing problems with big storage and mixing provide deep, useful information. A thematic study is used on qualitative data from research records and interviews. It focuses on finding repeated patterns and problems and how the growth of CVCS changes over time. The study helps us better understand how version control systems developed over time. It shows the limits that changed where they were going toward models that could be more spread and flexible.

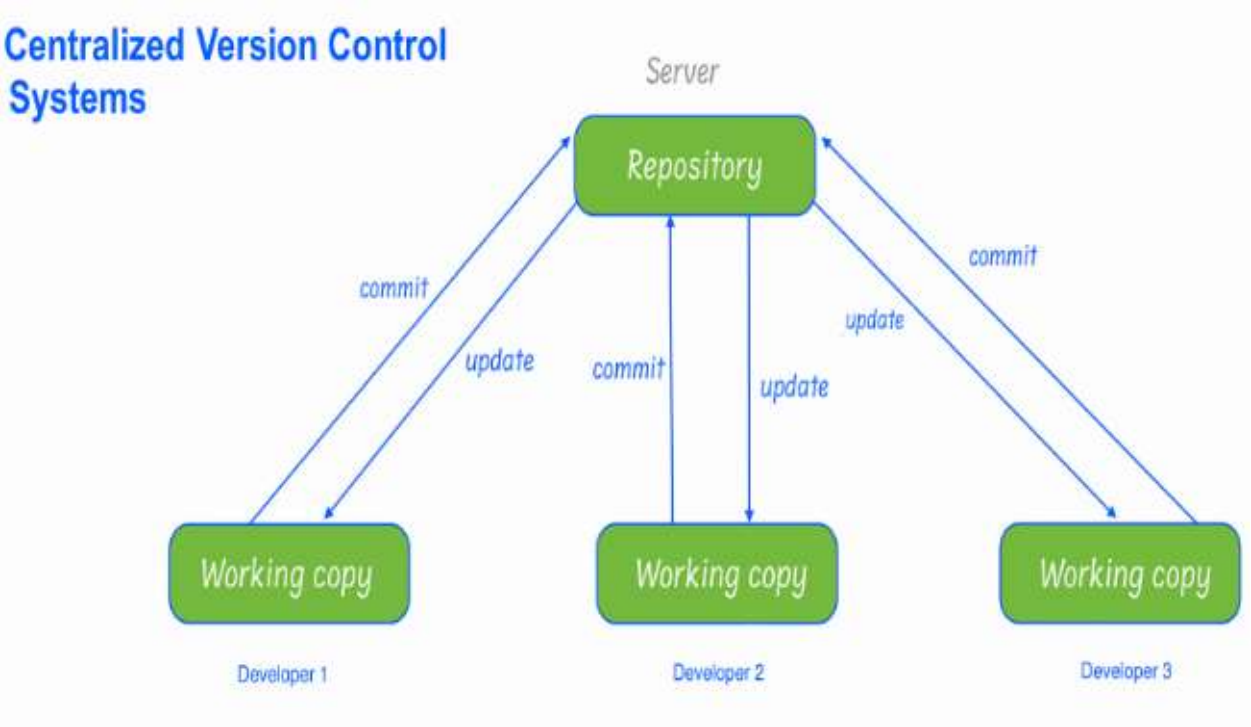

Figure 1: Centralized Version Control Systems

3. Distributed Version Control Systems

Changes in Source Control Management and the rise of distributed version control systems (DVCS), especially Git and Mercurial, marked a major change in the 2000s. It used a historical research framework to examine the transformational changes that Git's decentralized model brought in the [5]. The target population includes software developers, project managers, and teams that are currently determining DVCS adoption. The study captures prior findings on motivations and challenges associated with DVCS adoption using record analysis, technical document analysis, and interviews. Purposive sampling ensures the inclusion of and provides key indicators that different perspectives enhance the analysis. Thematic analysis is then used to reveal the profound impact DVCS, particularly Git, has had on enterprise practices and software development processes during this period.

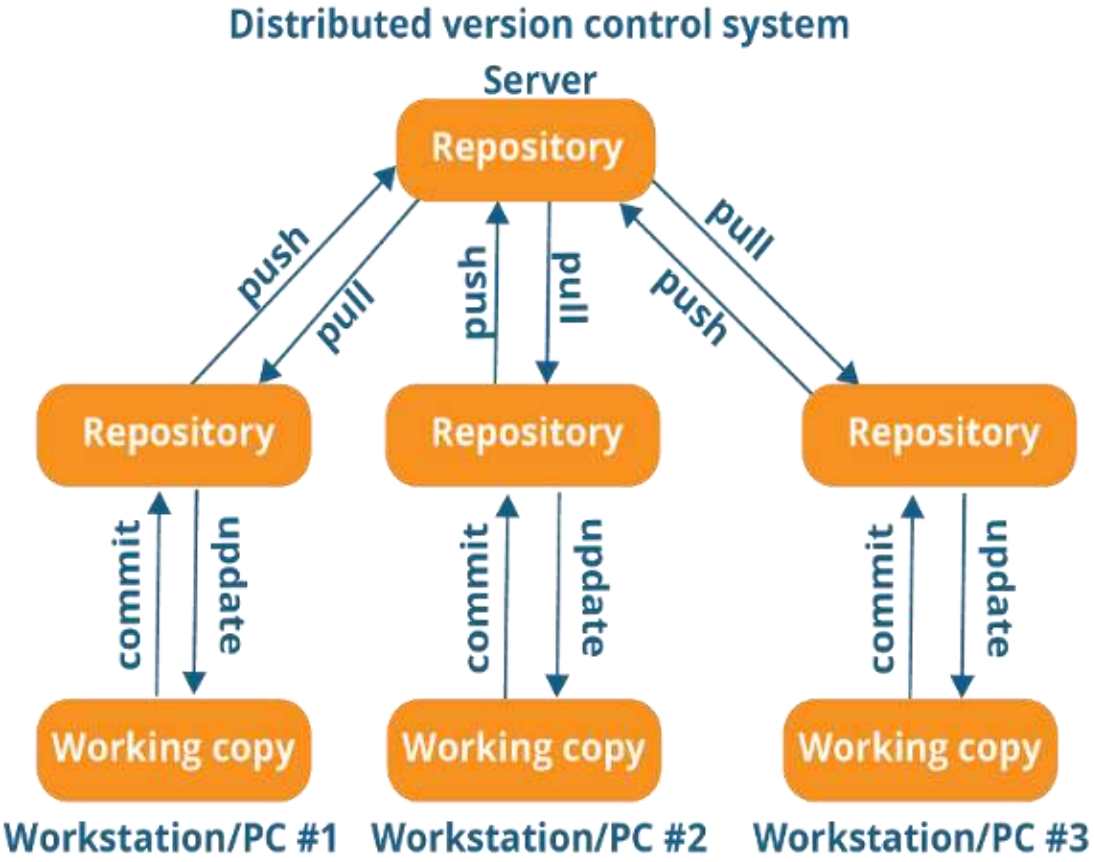

Figure 2: Distributed version control system

4. Git and GitHub

The ascendance of Git as a paramount version control system was significantly propelled by platforms like GitHub in 2008. GitHub, serving as a centralized hosting service for Git repositories, played a pivotal role in transforming collaboration dynamics through features like pull requests, issue tracking, and wikis. It adopts a research design focused on Git and GitHub's influence, targeting software developers, project managers, and teams engaged in utilizing these platforms. [6] The study captures firsthand insights into the transformative impact of GitHub's 'fork and pull request' workflow by employing archival research, technical documentation analysis, and interviews. Purposive sampling ensures the inclusion of key informants, enriching the analysis with diverse perspectives. Thematic analysis unravels the profound influence of Git and GitHub on collaboration practices, fostering global professional communities during this transformative period.

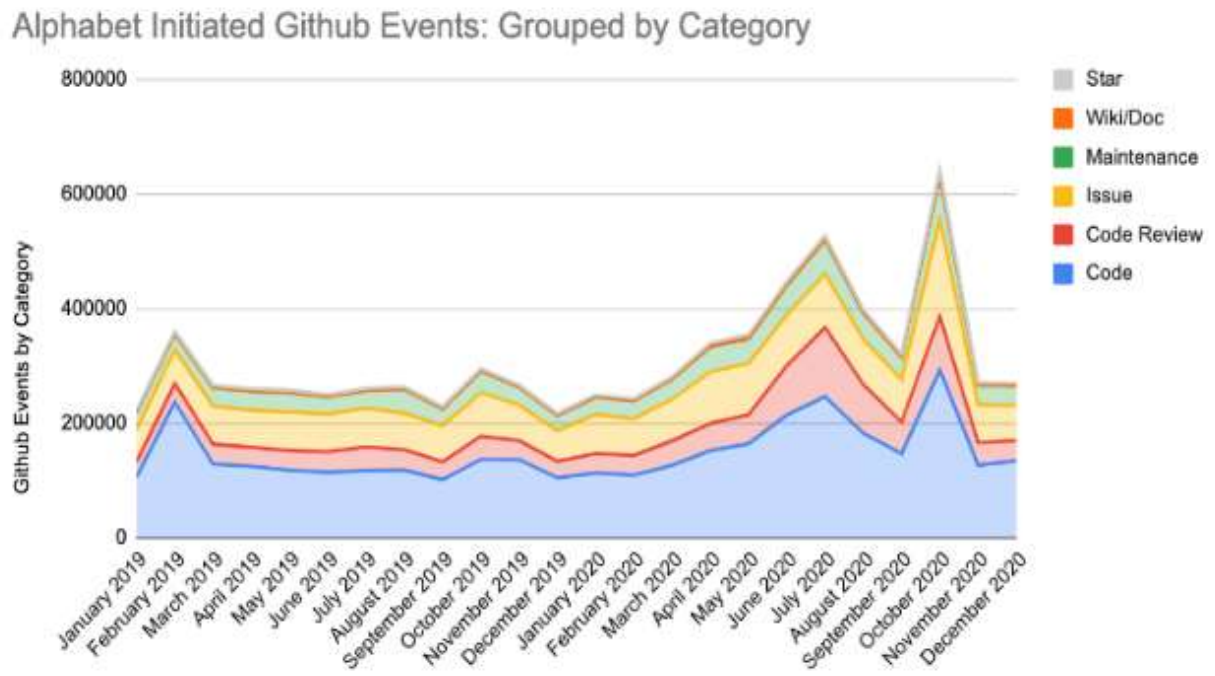

Figure 3: GitHub community collaboration

5. Branching and Merging Strategies

As Git and other distributed version control systems (DVCS) pioneered in 2000, some of the target populations and software developers, managers, and teams actively use these techniques. Using record analysis, technical documentation analysis, and interviews, the survey captures insights into Git's branching model, [7] GitFlow's transformative impact and alternatives such as GitHub Flow and GitLab Flow Objective examples ensure key indicators are included, provides a multi-perspective analysis. Thematic analysis reveals these strategies' profound impact, addresses collaborative development challenges, and contributes to a more efficient development environment.

6. Containerization and Infrastructure as Code

The rise of containerization technology led by Docker has led to a major shift in source control, expanding its domain from code to include defining and maintaining infrastructure [8]. Target populations include software developers, DevOps engineers, and teams that actively participate in containerized IaC practices. Using record reviews, technical document analysis, and interviews, the research itself monitors the impact of change in tools like Terraform, Ansible, and more. Thematic analysis reveals the profound impact of containerization and IaC, fostering collaboration and ensuring a consistent, repeatable process for modern software development and deployment.

7. Connection to CI/CD pipeline

Fusing continuous integration of source control with continuous deployment (CI/CD) pipelines is a common practice in modern software development [9]. The target population includes software developers, DevOps engineers, and teams actively involved in CI/CD practices. Using a combination of record analysis, technical document analysis, and interviews, the study captures direct perceptions of the transformational impact of integrating source control into CI/CD operations. The purposive model ensures that key indicators will be included, enhancing research with multiple perspectives. Subject analysis highlights the significant impact of these integrations, improves collaboration, accelerates the development cycle, and reduces the risk of product defects.

8. GitOps and Infrastructure as Code (IaC)

This concept, called GitOps, became popular in 2020 and is derived from the concept of DevOps. Today, the GitOps Git repository uses the source code and the source of truth where infrastructure and alert settings are defined and managed. [10] It is a declarative infrastructure control technology where you specify coded configurations in code, and the underlying source software and infrastructure in the repository are modified in various ways. The figure below shows a representation of GitOps.

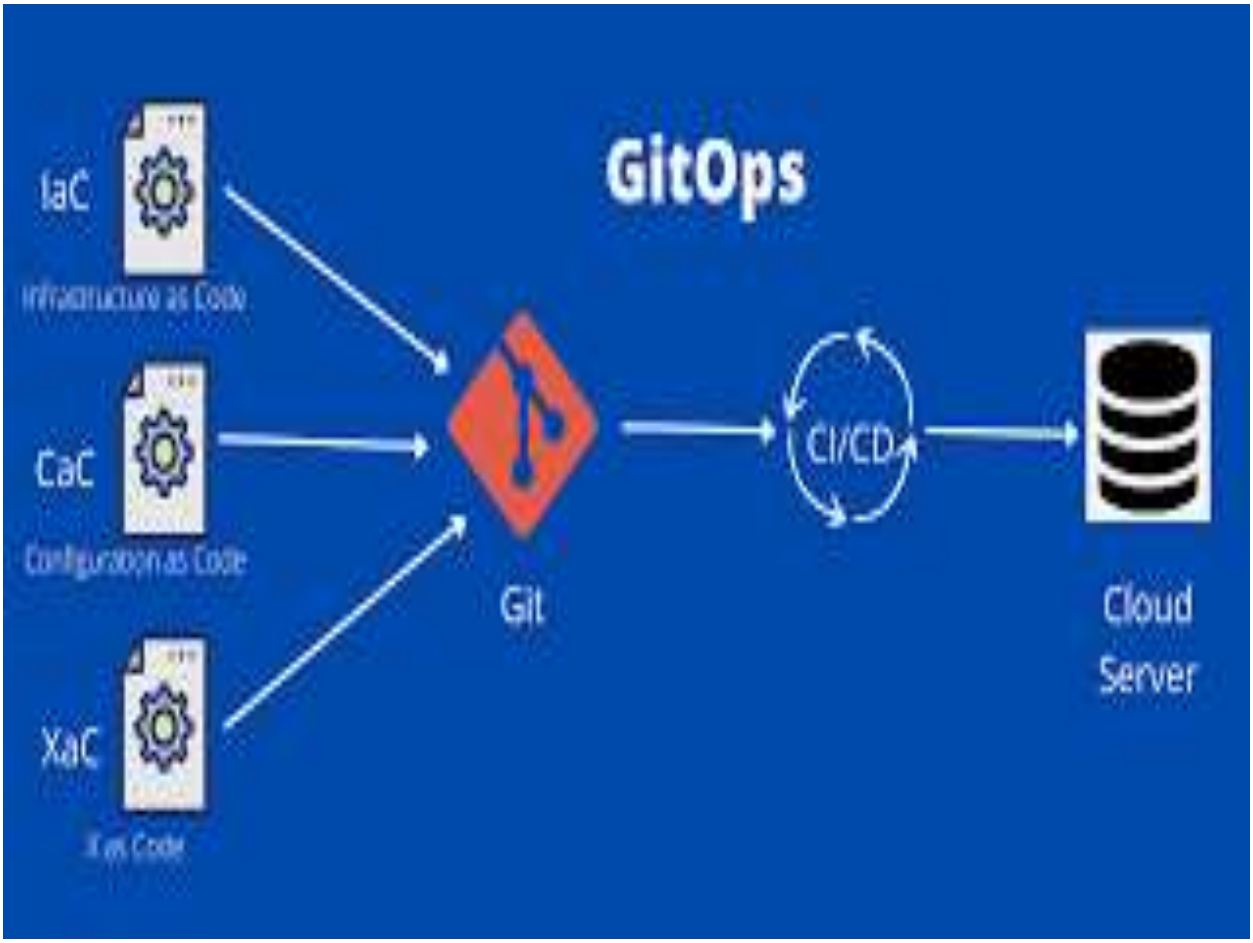

Figure 4: GitOps

9. Hybrid and Multi-Cloud Environments

In today's business world, where many use a mix of cloud systems, it is important to manage source control change. This will help handle issues from creating software in different ways on various platforms. It uses a study design that focuses on how source control, specifically tools like Git, can change things in both mixed and multi-cloud settings. [11] The people we want to help are software makers, DevOps workers and teams that handle source control in these big areas. The study uses historical records, technical writing examination, and personal talks to get a real-life understanding of the issues and benefits of working with many cloud providers. Using on-purpose sampling helps to include important people. This adds different views and makes the study better. After looking at Git and how it helps with group work and makes things faster and simpler in a mix of cloud systems, this is called theme study or analysis.

Future Trends and Challenges

Looking ahead, there are several trends and challenges to determine the future of resource management. Integrating machine learning and artificial intelligence into SCM tools promises to offer services such as code analysis, identification of potential problems, and optimal branching strategies, and the integration of tools and branches will continue to evolve and focus on facilitating development. - Temporary operation. Security is an important concern, and future developments in SCM will likely include enhanced methods for protecting code stores and identifying vulnerabilities. Integrated blockchain technology can provide additional mechanisms to ensure the authenticity and authenticity of regulatory changes, providing an immutable ledger for tracking changes

As a result, [12], the historical evolution of source control management has been characterized by continuous innovation, from manual version control to distributed version control systems and beyond. As software development practices evolve, source control management remains fundamental and will adapt to new features and technologies to facilitate efficient, secure, and collaborative software development processes.

The Future Trends of Source Control Management

The way SCM develops software is changing rapidly and continuously due to adaptive technologies and processes. This review will explore various proposed product improvements, redefine professional testing guidelines, and create new collaborations to enhance product quality standards. [13] As cyberspace continues to evolve and development methods continue to improve, detailed knowledge of the appropriate SCM strategies for every developer, team, and company is valuable and essential. This insight is applied to various aspects of SCM characterized by the design of a joint learning collaborative innovation project with artificial intelligence. Considering the challenges and opportunities that will shape the future and enable the advanced development of business models based on intelligent automation, this discussion takes the reader on a journey from SCM facilities to other companies.

1. Integrating artificial intelligence and machine learning

The interface between source control management and artificial intelligence (AI) is an evolving frontier that promises to change how developers interact with code repositories. AI algorithms are used to automate routine tasks, such as code reviews, identify possible issues, and predict the best branching strategies. You are experts, which significantly increases the efficiency and accuracy of the development process. [14] This section analyses the applications, benefits, and challenges associated with integrating AI and machine learning into SCM and focuses on how these technologies are poised to redefine the developer experience.

2. Enhanced Security Mechanisms

Protecting code repositories has become a key priority in the rapidly evolving digital landscape. Future developments in Source Control Management (SCM) highlight the need for advanced security mechanisms to combat vulnerabilities and unauthorized access. This includes changes in security practices, noted by [15] increasing cryptographic techniques and blockchain integration for tamper-resistant version history. Cryptographic techniques use encryption and secure key management to protect important code and provide a strong shield. Because blockchain integration offers an unchangeable version history, the development process is more transparent and dependable. Access is further strengthened by advanced authorizations like adaptive authorization and multi-factor authentication. Software development teams defend themselves against threats and enable their systems to function, lay a strong foundation, and remain safe in the face of an increasingly complex digital landscape by foreseeing possible security challenges and implementing new measures.

3. GitOps and Declaration Resources

The GitOps model is gaining popularity as a revolutionary way of looking at source code and infrastructures. This section describes how GitOps uses Git repositories as a source of truth to [16] define and manage infrastructure changes. Focusing on declarative infrastructure management, GitOps aligns infrastructure configuration with desired states and provides a consistent and analyzable approach to infrastructure changes.

4. Evolution of Collaborative Workflows

Collaboration is at the core of successful software development, and future trends in SCM are reshaping enterprise workflows to foster more efficient and inclusive collaboration. [17] The integration of collaboration tools within SCM platforms and as stand-alone solutions is explored, focusing on their role in facilitating improved communication, feedback mechanism times, and meeting the diverse needs of development teams.

5. Blockchain Technology in SCM

Originally synonymous with cryptocurrencies, blockchain seeks applications beyond finance, and its integration into SCM has the potential to address challenges of transparency, tamper resistance, and traceability. This section explores how blockchain technology will be used to create an

immutable ledger and decentralized source code. [18] Exploring the potential benefits and drawbacks of integrating blockchain into SCM, this article provides insight into how this technology can reshape version control practices, especially in trusted, accountable, and decentralized environments. Collaboration is paramount. The diagram below shows blockchain representation.

Figure 5: Blockchain

6. Hybrid and multi-cloud SCM environments

As organizations increasingly adopt hybrid and multi-cloud systems, the future of SCM is changing with the dynamic development of distributed development across platforms. [19] Emphasis is placed on ensuring seamless collaboration, version compatibility, and development efficiencies across environments with different infrastructure configurations. Given the need for hybrid and

multi-cloud development, it can play a critical role in helping SCM organizations with their cloud adoption journey.

An in-depth exploration of the fate of resource management in software program development is a popular panorama characterized by innovation, observation, and collaboration, as artificial intelligence and gadget studies redefine how developers interact with code, more security strategies improve virtual solidity, GitOps changes are driven by infrastructure, collaboration. Business processes are evolving, new block systems are being moved, and there is a trend towards SCM-hybrid multi-cloud environments where this scenario is strategically agile and informed is essential. With emerging trends, it makes better use of manufacturers and companies to make better use of these developments. SCM that lives as a parallel can benefit from the wealth of software development teams by offering possibilities to ensure their collaboration builds modern applications in a robust, highly responsive, flexible virtual environment. Below is a representation of hybrid/multi-cloud.

Figure 6: Hybrid/multi-cloud

Conclusion

An in-depth analysis of the historical trends and future trends of source control management (SCM) in software development reveals a dynamic and evolving landscape affecting the software industry's productivity and collaboration and exhibits more excellent stability. The historical journey from basic version control systems to the widespread adoption of distributed version control systems has shown a consistent effort to meet the evolving needs of development teams

References

[1] E. Matenga and K. Mpofu, "Blockchain-Based Cloud Manufacturing SCM System for Collaborative Enterprise Manufacturing: A Case Study of Transport Manufacturing," *Applied Sciences*, vol. 12, no. 17, p. 8664, Aug. 2022.

[2] N.Deepa, B.Prabadevi, K. LB, and B.Deepa, "An analysis on Version Control Systems," *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, Feb. 2020.

[3] E. Welty et al., "Worldwide version-controlled database of glacier thickness observations," Earth System Science Data, vol. 12, no. 4, pp. 3039–3055, Nov. 2020.

[4] P. Löwe et al., "Open Source – GIS," Springer eBooks, pp. 807–843, Jan. 2022.

[5] D. Ashenden and G. Ollis, "Putting the Sec in DevSecOps: Using Social Practice Theory to Improve Secure Software Development," New Security Paradigms Workshop 2020, Oct. 2020.

[6] M. Tushev, G. Williams, and A. Mahmoud, "Using GitHub in large software engineering classes. An exploratory case study," Computer Science Education, vol. 30, no. 2, pp. 155–186, Dec. 2019.

[7] L. E. Lwakatare et al., "DevOps in practice: A multiple case study of five companies," Information and Software Technology, vol. 114, pp. 217–230, Oct. 2019.

[8] R. Opdebeeck, A. Zerouali, and Coen De Roover, "Infrastructure-as-Code Ecosystems," Springer eBooks, pp. 215–245, Jan. 2023.

[9] F. Zampetti, S. Geremia, G. Bavota, and M. Di Penta, "CI/CD Pipelines Evolution and Restructuring: A Qualitative and Quantitative Study," IEEE Xplore, Sep. 01, 2021.

[10] Afzaal Ahmad Zeeshan, "Automating Everything as Code," *Apress eBooks*, pp. 109–162, Jan. 2020.

[11] S. R. Gundu, C. A. Panem, and A. Thimmapuram, "Hybrid IT and Multi-Cloud an Emerging Trend and Improved Performance in Cloud Computing," SN Computer Science, vol. 1, no. 5, Aug. 2020.

[12] Y. Ma et al., "World of code: enabling a research workflow for mining and analyzing the universe of open source VCS data," Empirical Software Engineering, vol. 26, no. 2, Feb. 2021.

[13] R. G. G. Caiado, L. F. Scavarda, L. O. Gavião, P. Ivson, D. L. de M. Nascimento, and J. A. Garza-Reyes, "A fuzzy rule-based industry 4.0 maturity model for operations and supply

chain management," International Journal of Production Economics, vol. 231, p. 107883, Jan. 2021.

[14] R. Cioffi, M. Travaglioni, G. Piscitelli, A. Petrillo, and F. D. Felice, "Artificial Intelligence and Machine Learning Applications in Smart Production: Progress, Trends, and Directions," Sustainability, vol. 12, no. 2, p. 492, Jan. 2020.

[15] M. Deshmukh and Arjun Singh Rawat, "Secure key sharing scheme using Hamiltonian path," International Journal of Information Technology, vol. 15, no. 8, pp. 4141–4147, Sep. 2023.

[16] I. Author, M. Fragner, T. Di, and L. Makor, "Applying GitOps principles to a cloud-native application Bachelor of Science in the Bachelor's Program," 2023. Available: https://www.ssw.uni-linz.ac.at/Teaching/BachelorTheses/2023/Fragner_Manuel.pdf

[17] P. McGrath, L. McCarthy, D. Marshall, and J. Rehme, "Tools and Technologies of Transparency in Sustainable Global Supply Chains," California Management Review, vol. 64, no. 1, p. 000812562110459, Sep. 2021.

[18] [1]S. V. Akram, P. K. Malik, R. Singh, G. Anita, and S. Tanwar, "Adoption of blockchain technology in various realms: Opportunities and challenges," Security and Privacy, vol. 3, no. 5, p. e109, Apr. 2020.

[19] S. R. Gundu, C. A. Panem, and A. Thimmapuram, "Hybrid IT and Multi-Cloud an Emerging Trend and Improved Performance in Cloud Computing," SN Computer Science, vol. 1, no. 5, Aug. 2020.