Version Control of Files in Aerospace Activities: Solutions for Technical Files and Documents

Cristian OGNERU*

*Corresponding author
INCAS – National Institute for Aerospace Research “Elie Carafoli”,
Bd. Iuliu Maniu 220, Bucharest 061126, România,
ogneru.cristian@incas.ro

DOI: 10.13111/2066-8201.2025.17.3.11

Received: 08 August 2025/ Accepted: 25 August 2025/ Published: September 2025

Abstract: *Version control of files plays a critical role in ensuring quality, supporting the development and management of research-development-innovation projects, and overseeing product lifecycles within the aerospace sector. The rising complexity of CAD/CAE/CFD model details, data files, technical documentation, and related administrative records demands robust solutions that guarantee data integrity, enable multidisciplinary collaboration, and maintain full change traceability. This paper examines the architecture of versioning systems, compares leading VCS (Version Control Systems) and PDM/PLM (Product Data/Lifecycle Management) platforms, and outlines the legal considerations associated with deploying these solutions.*

Key Words: *version control, SVN, PDM/PLM, CAD, aerospace management*

1. INTRODUCTION

In the aerospace industry, each designed part, CAD/CAE/CFD model, source code, and specification document represents critical intellectual property.

Proper file management is an essential component [7] of research - development and innovation project management, particularly in the aerospace sector, where iterative changes to working files (CAD, CAE, CFD), source code (Fortran, MATLAB, Python, etc.), and documentation must be tracked with precision.

The absence of rigorous version control leads to loss of design-decision traceability, integration conflicts across disciplines, additional costs, extended delivery timelines, and risks of non-compliance with aerospace management system standards such as AS9100 /ISO 9001 [3][5].

The objective of this study is to provide an overview of versioning solutions tailored to the specific requirements of aerospace activities and to propose a framework for selecting and implementing these solutions in research - development and innovation projects, ensuring a professional working environment, efficient collaboration, and compliance with aerospace regulations.

2. PRINCIPLES AND VERSIONING MODELS

The concept of version control relies on storing only the differences between successive revisions and maintaining a history that can be navigated in both directions [2].

This approach applies to files containing ASCII/Unicode characters (CSV, TXT, domain-specific scripts such as CFD code, Fortran, MATLAB, Python, etc.) as well as to binary files. Automated conflict detection and partial resolution are limited for binaries, often requiring additional software tools or modules.

Technical files differ from plain text files in several key ways:

- Size and format: CAD files (SolidWorks, CATIA), 2D and 3D models (STL, OBJ, DWG), CAE and CFD datasets, and documents (PDF, DOCX, XLSX) can reach tens of gigabytes.
- Binary nature: impossibility of applying conventional text comparison software applications, requiring archiving at the binary block level [2].
- Multiplicative dependencies: for example, assemblies containing hundreds of subassemblies, each with its own version set.

There are three main approaches to versioning models:

a) Centralized systems (e.g., Apache Subversion SVN): store the entire revision history on a single server, provide backups and rapid rollback to any version and offer centralized access control.

b) Distributed systems (e.g., Git): allow full repository cloning, enable offline work and complex branching workflows.

c) Proprietary commercial solutions (specific to the type of application): optimized for large, binary files, provide native support and high-bandwidth transfer mechanisms.

3. REFERENCE PLATFORMS AND TECHNOLOGIES

3.1 Classic Version Control Systems (VCS):

a) Apache Subversion (e.g., VisualSVN): better support for binary files, but reduced flexibility in distributed teams [2].

Apache Subversion (SVN) was originally conceived as a command-line interface, which involves opening a terminal and typing commands directly.

To operate, SVN requires two essential components (hardware & software): a server that stores all file versions (a single database) and a client that can manipulate data either directly in the database (on the server) or via a local working copy of those files on the user's machine. Local copies, called working files, are where each user makes changes; then, using a dedicated command, those changes are sent to the server.

This creates a new version, allowing users to work with the latest revision by default. If ever the need arises to restore a previous state of a file or the entire project, users can revert to a previous version at any time.

Implementing this system therefore requires two distinct software tools: the server, as the single repository, and the client, distribution installed on each project participant's work station (PC).

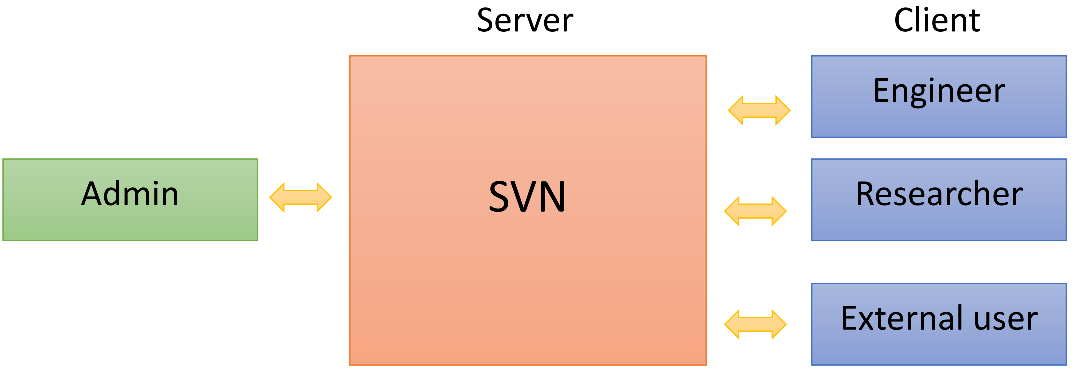

Figure 1. Simplified structure of the version control system

These software tools can be open-source (generally cost-free) or proprietary, and installable on various operating systems.

For example, on Windows OS the server (e.g., VisualSVN Server) carries a minimal license cost depending on the number of users, while the client application (e.g., TortoiseSVN) is free and integrates natively into „File Explorer".

These two software tools are successfully applied in aerospace activities, enabling proper management of project data at minimal costs.

The scalability of the version control system is not limited by the number of users, but depends on the capacity of the storage infrastructure.

b) Git: excellent for files containing ASCII/Unicode but with poor performance on large binary files.

Control is distributed, with each participant holding a complete copy of the project history. Git is heavily used in software development activities.

3.2 PDM/PLM platforms

These include a version control system and additionally offer data or product management and security services throughout the entire lifecycle [1].

3.3 Integrated platforms

These provide a unified environment where data, applications, and users converge on the same platform [3].

All roles, CAD/PLM features, collaboration tools, and development processes are accessible within a single ecosystem.

One such platform is 3DEXPERIENCE, featuring a web services and microservices architecture available on-premise in a client-server configuration.

It offers differential control (versioning and comparison) at the binary level, native multi-CAD integration, multiple configurable approval workflows, complete change tracking, and scalability to hundreds of thousands of users.

Additionally, it provides customizable dashboards on role and industry, and project management tools such as Gantt charts, advanced analytics, and real-time reporting.

Table 1: Main Features of Platforms Used in Aerospace Activities (Selection)

Feature / Brand	Architecture	CAD integration	Approval workflows, Change tracking	Scalability (Users)	Total cost
Windchill (PTC)	client-server	native / plug-in	configurable, detailed history	tens of thousands	high
Teamcenter (Siemens)	client-server	native / plug-in	configurable, detailed history	tens of thousands	high
Enovia (Dassault)	integrated module	native CATIA	configurable, detailed history	hundreds of thousands	high
Aras Innovator (ARAS)	client-server	native / plug-in	configurable, detailed history	thousands	medium
Vault (Autodesk)	client-server	native / plug-in	configurable, detailed history	hundreds	high
TortoiseSVN + VisualSVN server	client-server	plug-in	limited, detailed history	unlimited	low / medium
3DEXPERIENCE (Dassault)	client-server	native	configurable, detailed history	hundreds of thousands	high

Each system/platform facilitates collaboration between project entities, and they benefit from special permissions to work with files and documents. Documents uploaded to the workspace (server) are accessible only to a limited group of users designated by the process administrator.

4. LEGAL ASPECTS AND STANDARDIZATION

In international collaborations, ownership rights over 3D models and associated documents must be explicitly defined in contracts, covering distribution, modification, licensing and export.

Adopting globally recognized technical standards facilitates interoperability, guarantees product quality and reinforces the security of information exchanged between partners.

Key standards:

- ISO 10303 (STEP) [4] and ANSI/ASME Y14.26M (IGES), neutral formats that enable CAD data transfer and eliminate conversion errors between different design systems.
- AS9100, which extends ISO 9001 requirements by adding strict traceability, audit and continuous improvement, essential for the aerospace sector [5].
- ISO/IEC 27001, which specifies an information security management system [6].

5. CONCLUSIONS

Classic version control systems (VCS) allow significant budget savings for file management, but they come with compromises when handling binary file, where their capabilities are limited.

Commercial PDM/PLM platform solutions provide advanced performance features and a full audit trail of development processes, but the high total cost requires calibrating the system implementation (dedicated hardware allocation for servers) based on the size of the organization and the complexity of the projects.

Implementing an integrated version control strategy — combining classic VCS with PDM/PLM or integrated — yields substantial benefits in aerospace activities. It ensures data integrity, traceability, legal compliance and information security while optimizing costs and project timelines.

Defining clear selection and integration criteria for the versioning system will simplify the rollout of such a strategy. These criteria should address: total cost relative to number of users, user learning and assimilation time, accessibility and, last but not least, the hardware requirements.

Classic version control systems have total costs per user are on the order of tens of euros, with learning times measured in hours and minimal hardware needs. Integrated platforms and commercial PDM/PLM (excluding open-source options) incur total costs per user in the thousands of euros, require days to weeks of training for integrated platforms, and depend on dedicated hardware.

Moreover, by adopting such a system and revising project quality standards, an aerospace organization's reputation improves significantly, easing access to new contracts both as an independent entity and as a valued partner.

REFERENCES

[1] M. Meeshi & V. Kulkarni & V. Gaitonde & S. Jangali & B. B. Kotturshettar, A review on PLM platform deployment în aerospace industries, *AIP Conference Proceedings*, 2358. 100011. 10.1063/5.0057962, 2021.

[2] P. Khudyakov & A. Kisel'nikov & I. Startsev & A. Kovalev, Version control system of CAD documents and PLC projects, *Journal of Physics: Conference Series*, 1015. 042020. 10.1088/1742-6596/1015/4/042020, 2018.

[3] F. Mas, J. L. Menéndez, M. Oliva, A. Gómez, J. Ríos, 2013, Collaborative Engineering Paradigm Applied to the Aerospace Industry. În: Bernard A., Rivest L., Dutta D. (eds) Product Lifecycle Management for Society, PLM 2013, *IFIP Advances în Information and Communication Technology*, vol **409**, Springer, Berlin, Heidelberg.

[4] * * * *ISO 10303:2014*, Industrial automation systems and integration - Product data representation and exchange, International Organization for Standardization, 2014.

[5] * * * *AS9100:2016*, Quality Management Systems - Requirements for Aviation, Space and Defense Organizations , SAE International, 2016.

[6] * * * ISO/IEC 27001:2022 Information technology - Security techniques - Information security management systems – Requirements, International Organization for Standardization, 2022.

[7] * * * *NASA Systems Engineering Handbook*, NASA/SP-2016-6105 Rev2, 2017.