2026 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
***MODELING, SIMULATION, PROTOTYPING, AND VALIDATION (MSPV)* TECHNICAL SESSION**
AUGUST 11-13, 2026 - NOVI, MICHIGAN

COMPARATIVE EVALUATION OF CAD SYSTEMS AS ENGINEERING DESIGN ARCHITECTURES

Katelynn Hughes[1], Emily Petty[1], Gregory M. Mocko[1]

[1]Clemson University, Clemson, SC

ABSTRACT

There is a need for advanced collaboration tools to meet the needs of increasingly complex engineering design challenges and of global design teams. Specifically, virtual design reviews are a means by which teams can obtain a better understanding of a product's design. Many CAD systems have integrated cloud features, which can be utilized to perform design reviews within these systems. The focus of this paper is to evaluate CAD systems in three CAD architectures: traditional, cloud-enabled, and cloud-native, to assess their capabilities and highlight areas of improvement for CAD developers. Using a list of previously generated requirements, CAD systems were evaluated on how well they met each requirement to determine how they can foster collaborative work. While cloud-native CAD systems performed the best against the requirements, no CAD system was able to fully meet all requirements, showcasing multiple areas for future development.

Citation: K. Hughes, E. Petty, G. Mocko, "Comparative Evaluation of Engineering Design Architectures," In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium* (GVSETS), NDIA, Novi, MI, Aug. 11-13, 2026.

1. INTRODUCTION

The growing complexity of engineered products and the increasing distribution of design teams have increased the need for advanced collaboration tools. As product development timelines compress and teams become more globally dispersed, traditional computer-aided design (CAD) systems face significant limitations in supporting collaboration. While early improvements introduced cloud-enabled platforms that integrated Product Data Management (PDM) systems, these solutions largely preserved sequential access models and often introduced additional process overhead. More recently, cloud-native CAD architectures have emerged, offering real-time, multi-user access to shared models within a centralized environment designed specifically for concurrent collaboration.

Understanding how these three distinct architectural approaches — traditional CAD, cloud-enabled CAD, and cloud-native CAD — support or inhibit collaboration is essential for organizations seeking to optimize design review processes and accelerate innovation. Each architecture presents trade-offs in areas such as model concurrency, version control, annotation capabilities, and system scalability. However, a systematic evaluation of these architectures with respect to collaboration requirements remains limited.

While many studies compare the collaborative abilities of cloud-native and traditional CAD architecture together, these efforts focus on how a team that is working together synchronously performs against a single-user in place of how the architectures change how design teams need to interact [1 – 3]. Additionally, another area that is overlooked is the comparison of design reviews in each different CAD architecture. Design reviews within CAD systems allow for the ability to perform view transformations and add adjustments to models and assemblies that allow for an improvement in decision making and understanding of design [4]. One path that some CAD systems have integrated is a virtual environment (VE) and virtual reality (VR) integration [5]. However, the same collaboration needs are required for design reviews held in either a CAD system or in a VE. Previously, 35 requirements were introduced that set how a multi-user CAD (MUCAD) system should meet to be able to be effective at hosting design reviews [6]. These requirements can be used as a baseline of comparison between different CAD systems and the three different architectural approaches to CAD.

This paper presents a comparison of CAD system architectures with respect to their collaboration functionalities. By evaluating the strengths and limitations of each approach, this work aims to guide future development of collaborative CAD platforms.

2. OVERVIEW OF CAD ARCHITECTURES

2.1. Traditional CAD

Traditional computer-aided design platforms were originally built for standalone use, with files saved locally on individual workstations or shared over network drives. In this architecture, only one user can effectively work on a model at a time, often requiring manual coordination to avoid overwriting changes. Design iterations are typically managed by saving file versions, with naming conventions used to distinguish revisions. Collaboration involved exchanging files through email or shared folders, introducing risks related to version confusion, lost feedback, and delays in integrating contributions from multiple team members.

While collaboration on traditional CAD system architecture is sufficient in a single location, it presents significant barrier to dispersed collaboration and more detailed design projects. A lack of real-time updates means that design feedback cycles take a long time and concurrent modeling is limited.

The traditional CAD systems that were evaluated were SolidWorks 3D CAD, Siemens NX, and Autodesk Inventor. While these programs have plugins or additional applications that allow them to become cloud-enabled CAD architectures, the base form of these systems will be used to get a general understanding of how traditional CAD platforms perform against the requirements.

2.2. Cloud-Enabled CAD

Cloud-enabled CAD architectures integrate centralized data management systems, such as Product Data Management (PDM) or Product Lifecycle Management (PLM) solutions. In this model, design files are stored on a central server, with users accessing models through a check-in/check-out mechanism. When a user checks out a file, they gain editing privileges, while others are typically restricted to read-only access until the file is checked back in.

This centralized control improves version tracking and security but maintains the sequential editing paradigm of traditional systems. Coordination remains dependent on file status, and collaboration continues to follow a linear path rather than allowing simultaneous contributions. While cloud-enabled platforms reduce the risk of accidental overwrites they do not fully address the need for real-time, multi-user interaction during design development and review.

The check-in/check-out workflow, although more organized than the traditional CAD architecture, still hinders the collaboration process when users contribute concurrently on a single model.

The CAD platforms that were compared against the requirements were Siemens NX with Teamcenter, SolidWorks PDM, and PTC Creo+. The performance of these programs can be used to determine how the average cloud-enabled CAD system would perform in a collaborative environment. Compared to cloud-native CAD architecture, these programs are not fully based in the cloud and can be accessed offline.

2.3. Cloud-Native CAD

Cloud-native CAD architecture is designed specifically for multi-user collaboration. These systems host the model in a centralized, cloud-based environment where users can access, edit, and review the same model simultaneously. Changes made by one user are immediately visible to others without the need for manual synchronization or file exchange.

In a cloud-native environment, version history, branching, and merging operations are integrated directly into the platform. Collaborative annotations, comments, and embedded feedback mechanisms are also typically supported, allowing design teams to document and discuss model changes within the system itself. Access control and permissions management are handled dynamically, enabling flexible collaboration across organizational boundaries.

By enabling real-time interaction and seamless feedback integration, cloud-native MUCAD architectures can improve efficiency and the quality of collaboration in design and design reviews.

The cloud-native systems that were evaluated were PTC OnShape, Autodesk Fusion, and Siemens Solid Edge. These programs were used to develop a general metric of how cloud-native CAD architectures would be expected to perform against the requirements.

3. COMPARATIVE ANALYSIS ACROSS DIFFERENT ARCHITECTURES

3.1. Metrics for Comparison

The collaborative capabilities were compared through their ability to enable design reviews. All CAD programs were compared against 35 requirements to determine the degree in which they met each requirement [6] (see Table 1). The requirements were also organized into different categories based on similar characteristics among the requirements. These categories were concurrency, annotations, heterogeneity, IP protection, security, version control, and viewing [7]. Requirements in the annotations category deal with editing, commenting, or adding suggestions to a model [7]. Requirements in the concurrency category are related to aspects of the CAD system that keep the system the same for all users at any time [7].

Table 1: MUCAD system requirements list.

#	Requirement	Category(s)
1	Operations shall always result in the same model for everyone.	Concurrency
2	Operations shall always result in the same construction tree for everyone.	Concurrency
3	Multiple people shall be able to access a model simultaneously without locking others out or changing the system.	Concurrency
4	The system shall allow users to leave text-based annotations or comments within assembly and component models.	Annotations
5	The system shall enable reviewers to quickly provide feedback through sketches viewable by other users.	Annotations
6	Users shall be able to capture notes/feedback through video, audio recordings accessible to other users.	Annotations
7	Suggestions should be available for users to make and other users to accept.	Annotations
8	The system shall be software OS agnostic (iOS, Windows, Linux).	Heterogeneity
9	The system shall support the ability to import/export multiple formats.	Heterogeneity
10	The system shall support the interoperability across CAD tools.	Heterogeneity
11	The system shall provide overall security permissions to access the system.	IP Protection
12	The system shall provide the option for individual part/subsystem security permissions.	IP Protection
13	The owner of the model shall be able to control security access/permissions for annotations for individual users.	IP Protection
14	The system shall have backup log/operations and the ability to roll back.	Security
15	The system should be isolated from the operations of unauthorized users.	Security
16	The system shall have a means of releasing stable versioning as well as capturing rationale and changes across revisions.	Version Control
17	The system shall be easy to replace parts/subsystems to show design options.	Version Control
18	The system shall allow for branching like Git.	Version Control
19	The system shall enable all operations to be undoable.	Version Control
20	The system shall enable users to snap to other users' views.	Viewing
21	The system shall identify the users accessing the model through an avatar or similar denotation.	Viewing
22	The system shall have a mode where everyone can explore freely and a "presenter" mode where one person drives everyone's views.	Viewing
23	The system shall support synchronized representations including transparency, selective visualization, and section view tools.	Viewing
24	Users should be able to switch between component to assembly view.	Viewing
25	The system shall only allow for the creation of valid geometry.	Concurrency Version Control
26	The system shall enable to model and associated files to be simultaneously and independently viewed.	Concurrency Viewing
27	The system shall enable access to engineering simulation (i.e., FEA, CFD, motion studies) in real-time (live) as well as in their own time.	Concurrency Viewing
28	The system shall indicate what current feature(s) is/are being edited.	Concurrency Version Control
29	The system shall prevent conflict changes through revision control/lockout.	Concurrency Version Control

30	The system shall enable editing/sketching/feedback from multiple users on the same part/subassembly.	Concurrency Annotations
31	The system shall track each user's editing history.	Concurrency Version Control
32	The system shall provide warnings to users for suggested changes that invalidate geometry.	Concurrency Annotations Version Control
33	The history of annotations and edits should be available to all those with access to view the history.	Annotations IP Protection Version Control
34	The system shall provide multiple levels of permissions for annotations to the system as a whole (view, edit, save, comment, delete).	Annotations IP Protection
35	The system shall provide multiple levels of permissions for annotations to individual parts/subsystems (view, edit, save, comment, delete).	Annotations IP Protection

Heterogeneity requirements refer to data transfer and interoperability between different platforms [7]. The IP protection requirement category has requirements related to access clearance the CAD system and what tools can be limited to specific users [7]. The requirements within the security category deal with document security of models when the model is shared between internal and external users [7]. The version control category involves requirements related to keeping models consistent through versioning [7]. Lastly, the viewing requirement category involves the viewing and view-changing capabilities between different users in a CAD system [7].

For each CAD platform, the systems were evaluated through either the use of the program or through an analysis of their product information. The type of requirement evaluation for each of the observed programs is shown in Table 2.

For programs that were evaluated through user interaction, requirements were measured through manually going through the software. Platforms that were evaluated through advertised information had the requirements measured through the posted information by the platform's companies such as the product information page, release notes, demonstration videos, community forums and instructional videos and guides.

Table 2: Evaluation metrics for each CAD platform.

CAD Software	Evaluation Type
SolidWorks	User Interaction
Siemens NX	User Interaction
Autodesk Inventor	Advertised Information
Siemens NX + Teamcenter	User Interaction
SolidWorks PDM	Advertised Information
Creo+	Advertised Information
OnShape	User Interaction
Autodesk Fusion	Advertised Information + User Interaction
Siemens Solid Edge	Advertised Information

The three different CAD architectures are compared to the 35 requirements using a 4-level Likert scale. In this scale, the degree in which each requirement is met is identified by using a categorical name. Each category is associated with a corresponding score and key. The categories are not met, partially met, mostly met, and fully met. If a requirement is not met, that means that the platform does not carry out the requirement listed. A partially met requirement is one that is met marginally or only meets less than half of the requirement. A mostly met requirement is one that is able to accomplish most of a requirement but is missing portions of the requirement. A fully met requirement is one that meets all terms mentioned in the requirement.

Each CAD system is assigned a score based on a qualitative analysis using company information, direct user interaction, and published information in academic journals and conferences. The score is used to quantitatively assess qualitative data, while the key and items are used to visually represent the degree in which each CAD system meets a given requirement. The CAD systems within each architecture were compared via two metrics. The first is through comparing the number of requirements in each category. With the Likert scale, each CAD system can be given a score based on how it enables or inhibits collaboration. This gives information about which CAD systems, and therefore which CAD architectures, perform the best against the requirements. The second metric of comparison is through the analysis of total fulfillment in each requirement category.

The total fulfillment ($TF\%$) is calculated by dividing a CAD system's score for the category (S_c) by the maximum possible score (S_m). The maximum possible score is a representation of a 'perfect' score in a category, which is dependent on the number of requirements in a category.

Total fulfillment is used to see if there are trends within the architectures. The performance within different categories also highlights the strengths and weaknesses of CAD systems regarding their collaborative capabilities and can be used to analyze where improvements are needed most. The equation for total fulfillment is represented by Equation 1.

$$TF\% = \frac{S_c}{S_m} \cdot 100\% \qquad (1)$$

3.2. Determination of Traditional CAD Architecture Performance

The evaluation of the three traditional CAD architectures is included in Appendix Table A.1. Overall, these three CAD programs performed similarly. For all three CAD systems, at least two-thirds of the requirements were not met to any degree. All CAD systems fully met seven requirements, and no system had any requirements that were mostly met. This indicates that the traditional CAD architecture does not support virtual collaboration well. Out of all three programs, Inventor had the highest score of the three CAD programs, while SolidWorks had the lowest score. In Figure 1, the requirements are totaled into groups based on the degree to which they are met. The average number of requirements in each group is represented with a horizontal line.

Due to the nature of the architecture, it is to be expected that the CAD systems evaluated would perform poorly. As mentioned previously, this architectural style lacks the ability to have real-time updates, automatically iterate on the design, or share parts internally. Because all three systems were evaluated through user interaction, people working on a project with these programs may need to take steps to prevent significant delays due to feedback cycles, the integration of different parts, and the time it takes to share models externally for feedback or modification.

Because the traditional CAD architecture is not built with collaborative features in mind and with zero cloud features, it could be argued that it is not comparable to the other two architectures. However, the traditional CAD architecture is an important reference for what many engineers are trained on and what many engineering firms use in their day-to-day activities. Many schools teach students in traditional CAD systems, engineering students are unfamiliar with how to best utilize cloud features when they encounter CAD systems that have them, often keeping habits formed around versioning or file sharing with traditional CAD systems. Additionally, CAD systems with could features are often built on traditional CAD systems or are traditional CAD systems with an additional component, with higher pricing that some engineering firms choose not to use in favor of a more economical option. Therefore, the inclusion of the traditional CAD architecture and highlighting its poor performance serves to showcase how support for cloud collaboration needs to improve in not only the features available in cloud tools, but also through the education and training that engineers are given with CAD tools.

Out of the seven requirement categories, systems within this architectural format performed best in requirements related to heterogeneity. This can be expected, as this category deals with data transfer between different platforms, which while important for MUCAD, is something that is not inherent to synchronous systems.

In comparison, all requirements within the security and IP protection categories were not met by any CAD platform within this architecture. This is also something that can be attributed to the single-user nature of these platforms. Because these platforms are only used by individual users, there are no built-in features to restrict operations as the person working on the file would always need administrative access (see Table 3).

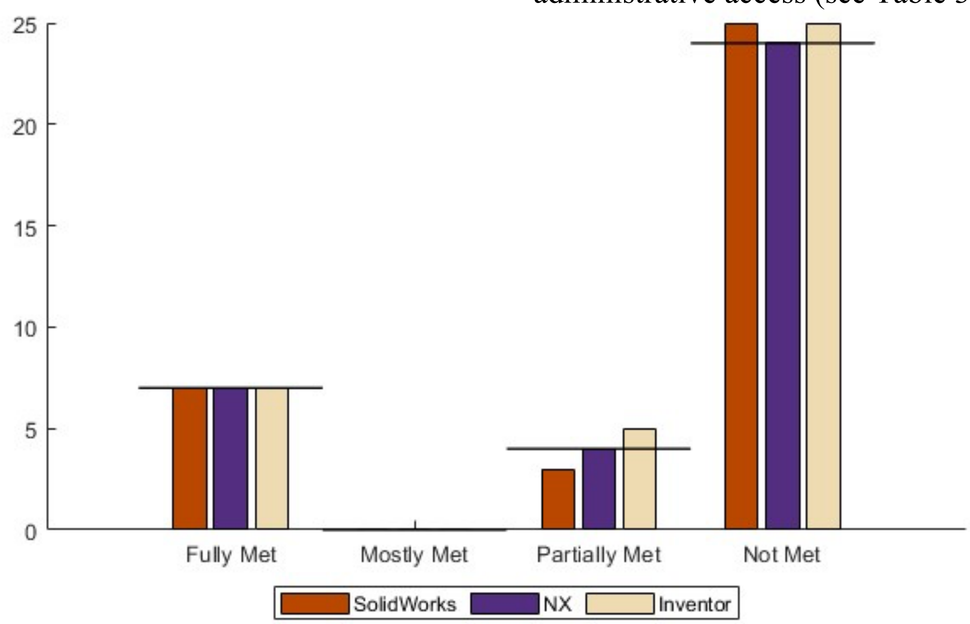

Figure 1: Comparison of requirement meeting of systems within the traditional CAD architecture.

Table 3. Total fulfillment by category of systems of traditional CAD architecture.

Category	SolidWorks TF%	NX TF%	Inventor TF%	Average TF%
Concurrency	21	21	21	21
Annotations	26	26	30	27
Heterogeneity	67	78	67	70
IP Protection	0	0	0	0
Security	0	0	0	0
Version Control	37	37	40	38
Viewing	14	14	14	14

3.3. Determination of Cloud-Enabled CAD Architecture Performance

The evaluations of the cloud-enabled three systems within the cloud-enabled CAD architecture are shown in Table A.2.

Within this architecture, two of the three systems fully met at least half of the requirements, with all CAD systems mostly meeting at least half of the requirements. NX + Teamcenter had the highest number of requirements that were met to some degree, while SolidWorks PDM had the least. Creo+ had the highest number of fully met requirements, while NX + Teamcenter had the least number. The number of requirements in each requirement meeting category for each of the CAD programs is shown in Figure 2.

Compared to the other CAD architectural styles, the cloud-enabled CAD architecture had a wide range of variety. In addition, it can be noted that the program that performed the 'worst' in regard to the number of fully met requirements, NX + Teamcenter, was the only program that was evaluated through user interaction. The other systems, SolidWorks PDM and Creo+, were evaluated through a review of information available online about the systems. In addition, NX + Teamcenter is also the only program that had no requirements that were not met in any capacity.

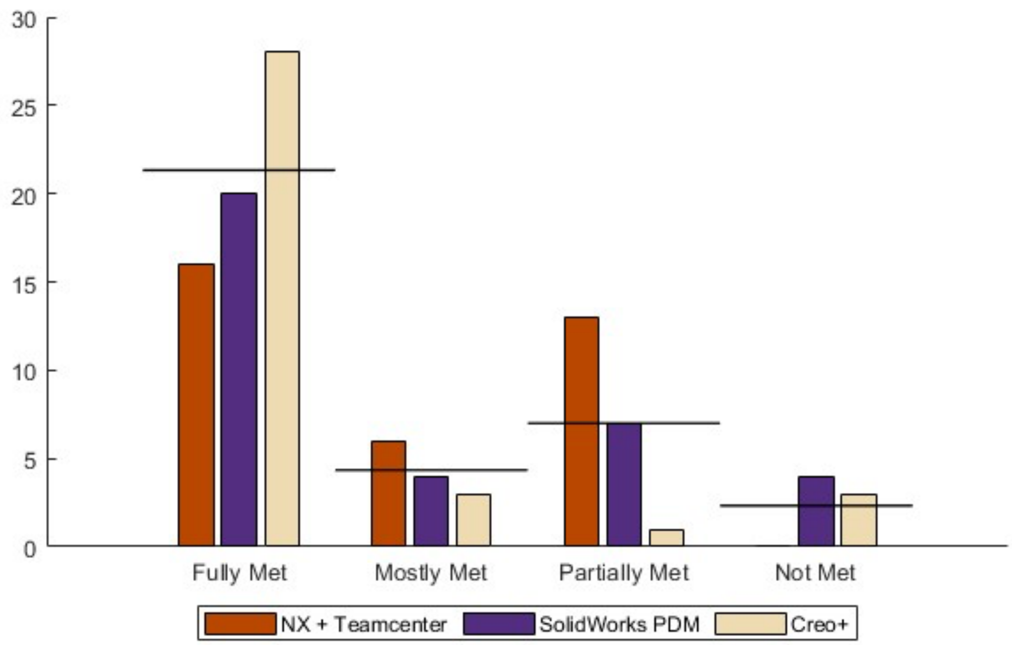

Figure 2: Comparison of requirement meeting of systems within the cloud-enabled CAD architecture.

Table 4. Total fulfillment by category of systems within cloud-enabled CAD architecture.

Category	NX + Teamcenter TF%	SolidWorks PDM TF%	Creo+ TF%	Average TF%
Concurrency	67	61	97	75
Annotations	70	74	85	77
Heterogeneity	67	78	67	70
IP Protection	94	94	100	96
Security	100	100	100	100
Version Control	83	87	93	88
Viewing	48	38	76	54

Both things could be impacted through the evaluation process. Because the information advertised about programs is much more limited than using the program, it is possible that the programs could be more limited in their performance than promotional wording would suggest. Additionally, it is also possible that the corporations that developed these systems do not mention every capability of the program, which could also mean that some categories that are not met could be classified within a different requirement meeting category.

When comparing the performance of the three CAD systems by which category they performed best with, it was noted that all three had fully met all requirements within the security category. Creo+ had also fully met all requirements within the IP protection category, while the other two CAD systems had high total fulfillment for this category but had mostly met some requirements which prevented it from being 100% fulfilled. This is something that can be expected of programs in this category. Security and privacy vulnerabilities are one of the biggest concerns that cloud systems can have, so it is important for the developers to implement the ability to protect work from being stolen, modified, or destroyed [8]. The total fulfillment by category for cloud-enabled architectures is shown in Table 4.

The three evaluated systems also perform similarly well in the version control and annotations categories, with most of the requirements in these categories being fully met. With access to the cloud, cloud-enabled systems allow for individuals to share updated models easily and to review and annotate on these models [8]. SolidWorks PDM and Creo+ had also had most of the heterogeneity category fully met while NX + Teamcenter did not, but the three performed similarly when considering the total fulfillment. This can be explained by NX + Teamcenter being more limited with its interoperability and integrating with CAD programs with PLM connectors well and having limited interoperability with CAD programs that do not.

Regarding the other two categories, both had more variation than the rest. With the concurrency category, Creo+ had a total fulfillment of 97%, which was noticeably higher than the performance of NX + Teamcenter and SolidWorks PDM. This can be explained by the nature of the CAD programs. NX + Teamcenter and SolidWorks PDM both operate on a check-in/check-out model, while Creo+ doesn't. Therefore, two people cannot work concurrently when using these platforms. With Creo+, however, users can work in collaboration sessions. These allow for more collaboration to occur, making the CAD systems meet requirements in this category to a higher degree than the other two.

The viewing category had the lowest total fulfillment among the categories for both SolidWorks PDM and NX + Teamcenter, with many of the requirements in these categories being partially met or not met by these programs. Creo+, however, had fully met most of the requirements in this category. This is another outcome that can be explained by the check-in/check-out workflow in SolidWorks PDM and NX + Teamcenter, as it hinders the ability to view a model at the same time.

For the variation between these three categories, this could be explained by the nature of the architectural style. Cloud-enabled CAD programs can be created by adding cloud capabilities to existing legacy CAD programs [8]. For example, Siemens NX is changed from being within the traditional CAD architecture to the cloud-enabled CAD architecture with the addition of Teamcenter. While cloud capabilities are available within this infrastructure, collaborative tools are an additional feature that may not be fully realized in these systems [9].

3.4. Determination of Cloud-Native CAD Architecture Performance

The evaluations of the three systems within the cloud-native CAD architecture are shown in Appendix Table A.3.

Between the three CAD programs within this architectural style, there was an average number of 28.33 fully met requirements, 3.67 mostly met requirements, 2.33 partially met requirements, and 0.67 requirements that were not met. Out of the three systems, OnShape had fully met the most requirements and Solid Edge had the least number of fully met requirements. Both OnShape and Solid Edge had some portions of every requirement met, while Fusion was the only system that had some requirements that were not met to any degree. The number of requirements in each requirement meeting category for the three CAD programs are shown in Figure 3. The average value is represented with a horizontal line.

The variety between the performance between the three programs could be caused by two reasons. The first is different priorities from the CAD developers. Each CAD system is created to solve a different niche, so it is to be expected that the systems are not identical.

The second is that Fusion and Solid Edge, which both have less fully met requirements, were both evaluated through advertised information while OnShape was evaluated through user interaction.

It is possible that the product information provided could underrepresent the true capabilities of the program, which could lead to the actual requirement meeting of the programs being higher. The opposite is also true, where the product information could overrepresent the system's capabilities, and the CAD programs could also have worse performances. There is also information that can be obtained from the performance of the CAD systems within the different categories for the requirements. This can be used to understand trends between the performance of systems within the architectural style by looking at the total fulfillment for each category (see Table 5).

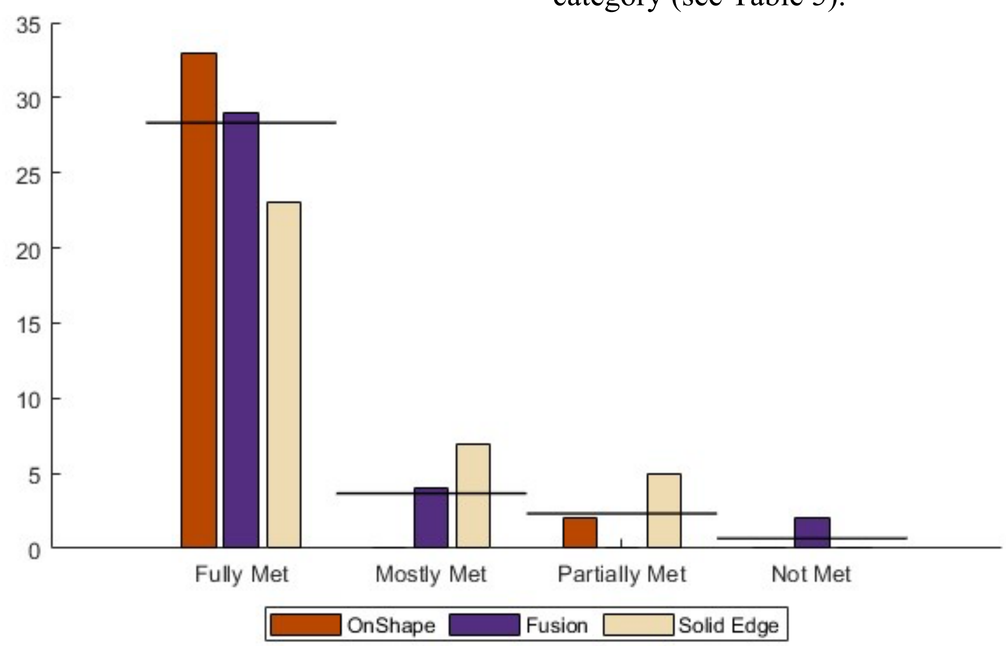

Figure 3: Comparison of requirement meeting of systems within the cloud-native CAD architecture.

Table 5. Total fulfillment by category of systems within cloud-native CAD architecture.

Category	OnShape TF%	Fusion TF%	Solid Edge TF%	Average TF%
Concurrency	94	94	79	89
Annotations	93	81	93	89
Heterogeneity	100	100	100	100
IP Protection	100	94	100	98
Security	100	100	67	89
Version Control	100	97	87	94
Viewing	90	81	71	81

Overall, the best performance among the categories was heterogeneity. All three CAD systems evaluated had fully met all requirements in this category. IP protection was also another category where the CAD systems performed well. OnShape and Solid Edge had met all the categories, and Fusion had a high total fulfillment. This is something that can be expected from a cloud system, especially the SaaS platforms that OnShape and Solid Edge system, as these programs have access permissions built in that the user is responsible for managing [9].

Each CAD system had a total fulfillment of at least 67% for all categories. The worst performance was by Solid Edge in the security category. The other two CAD systems had fully met all requirements in this category while Solid Edge had one fully met requirement, and one partially met requirement. This performance is unexpected due to the cloud-based nature, but it could be explained by Solid Edge not allowing rollback in its synchronous mode. Because this program was evaluated through advertised information, this could also explain the discrepancy compared with user-evaluated programs, and the true performance of Solid Edge matches that of the other two systems.

Overall, it was expected that the cloud-native CAD systems have high total fulfillments. The requirements themselves were written from the perspective of using a MUCAD software for design reviews. The online nature of this architectural style and inherent multi-user abilities that are enabled in the cloud mean that the average CAD system should be able to fully meet many of the requirements while having limited, if not zero, requirements that are not met.

OnShape had its worst performance in the viewing category. This is due to the limited simulation capabilities of OnShape, which require integration with external partners [6]. Lastly, Fusion had its lowest total fulfillment percentages in both the annotations and viewing categories. The low performance in the viewing category can be attributed to the lack of a clear presenter mode, while the low performance in the annotation category is due to the lack of internal audio and visual recording. It can be noted, however, that both CAD programs performed better in the viewing category than Solid Edge had. This is due to the limited ability within Solid Edge to share views within the system. However, because Solid Edge was evaluated through advertised information, it is possible for Solid Edge to have better viewing capabilities than what is featured, as most advertisements are shown in a single-user state.

Between the three CAD programs evaluated, there were two that no system had fully met. These were requirements 6 and 27. Requirement 6 regards the ability to capture audio and video feedback within the system and is in the annotations category. On the other hand, requirement 27 is regarding the simulation capabilities of the system and co-simulation and is in the concurrency and viewing categories. This aligns with the results by category, as these three categories are the only ones without any program to have a total fulfillment of 100%.

3.5. Inter-Architecture Comparison

As mentioned previously, the different CAD architectures were evaluated through comparing the average performance among the three CAD systems evaluated for each architecture. Some deviation is expected between each CAD software, so the average performance is used to represent an expected performance of any CAD system within the architecture.

The number of requirements can be compared to evaluating how well an architecture is able to meet collaborative requirements. If a system were only able to partially or mostly meet requirements, then that presents a gap in use that the CAD developers could investigate. Additionally, if one architecture consistently performs worse against the collaborative requirements, then that would represent an inherent aspect of the architecture. The total fulfillment of the different requirements can be compared to determine how the different architectures perform within the different categories. If a CAD architecture has met requirements to a high degree in all but one category, then that can be identified as an area for CAD developers to evaluate potential change in the system. The average number of requirements in each requirement meeting category for the three CAD architectures is shown in Figure 4.

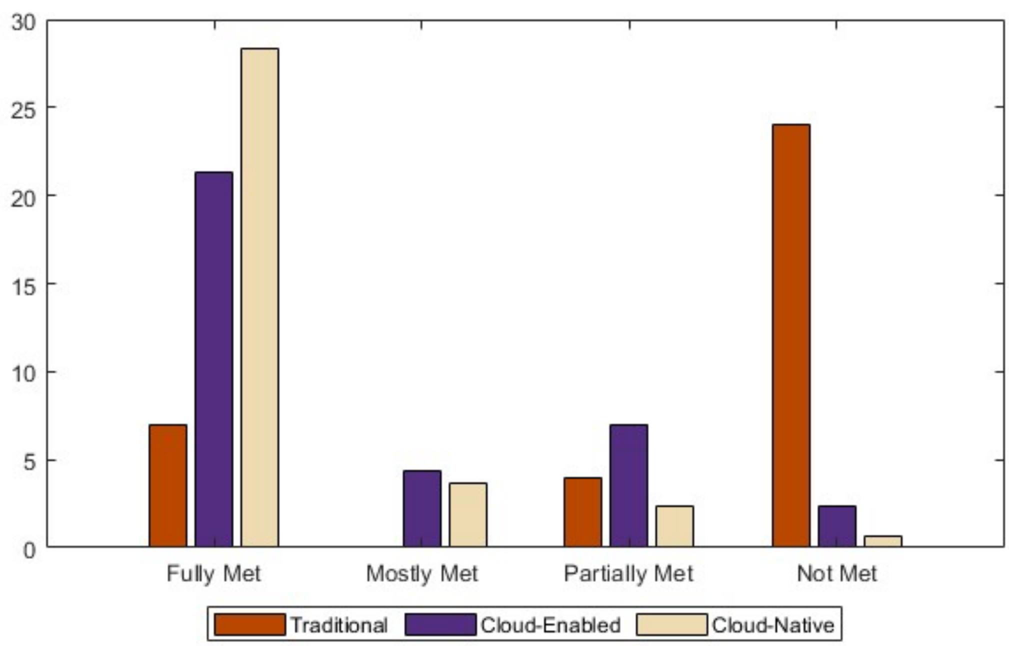

Figure 4: Comparison of requirement meeting of systems within each architectural.

Table 6: Comparison of total fulfillment of requirements by category between each CAD architecture.

Category	Traditional TF%	Cloud-Enabled TF%]	Cloud-Native TF%
Concurrency	21	76	89
Annotations	26	81	89
Heterogeneity	67	78	100
IP Protection	0	100	98
Security	0	100	89
Version Control	37	90	94
Viewing	14	57	89

There is an upward trend in the number of requirements that are fully met when more cloud features are introduced. However, this trend is not consistent when comparing partially or mostly met requirements, indicating that some of the increase in collaborative features is not dependent on the architectural style but rather on what collaborative tools and features CAD developers introduce and how they are implemented.

From the comparison of requirement meeting, it can be noted that the cloud-native CAD architectural style performed the best against the collaborative requirements, then followed by the cloud-enabled CAD architecture, with the traditional CAD architecture performing the worst. The poor performance of the traditional CAD architecture can be explained due to the systems being evaluated as single-user systems. This leads to the expectation that the CAD systems within this architecture would perform worse when fulfilling requirements that were designed for CAD systems with multiple users [7]. The performance of the cloud-enabled CAD architecture was also expected to be between that of the other two architectures, as many systems within this category are based on expansions from existing CAD systems that used traditional CAD architecture [8]. Systems within the cloud-enabled CAD architecture often use check-in and check-out models, which reduce the real-time collaboration capabilities of a system [7]. Additionally, other collaboration features are typically more limited compared to cloud-native CAD features. The cloud-native CAD architecture's performance is also expected, as systems within the architecture are purely online. Cloud platforms such as the internet allow for concurrent, digital collaboration [10]. This nature can be extended to cloud-native CAD systems, which met the requirements to the higher degrees compared to the other architectural styles.

As mentioned previously, the average total fulfillment within the CAD architectures in different requirement categories was used to compare the three architectural styles. The average total fulfillment for the three architectures is shown in Table 6.

From the total fulfillment by category, two categories stand out due to the drastic rise from the traditional architecture to the cloud-enabled architecture and the cloud-native architecture. These are the IP protection and security categories. Software within the traditional CAD architecture are single-user applications, where the CAD software is downloaded onto a local drive to be used by one individual at a time. For the end users of these softwares, the authentication and security protections are typically run through the end users' credential and network directory service [11]. These protections are put in place by the company or school that has licensed the service, leading to them being responsible for securing the files from the CAD software. This results in the traditional CAD architecture having a total fulfillment of 0% within the IP protection and security categories because this type of service is not needed for software that does not have any cloud capabilities. The privacy and security requirements of software change when cloud capabilities are added, as is the case within the cloud-enabled and cloud-native CAD architectures. The cloud nature of software within this architecture allows users to connect through networks that are in different geographic areas, which can lead to security threats such as server identity theft and unauthorized user operation [11].

It can also be noted that the traditional and cloud-enabled architecture performed similarly within the heterogeneity category. From Table 10, the average total fulfillments of the two systems were 67% and 78%, respectively. This is the only category in which the cloud-enabled architecture's total performance was closer to the traditional architecture's than the cloud-native architecture. This is due to the traditional and cloud-enabled architectures' performance on requirement 8, as shown in Table 4 and Table 6. While the browser-based nature of cloud-native CAD software allows the software to function on any operating system, traditional and cloud-enabled CAD software requires a local installation that is often not OS agnostic. This makes these programs more difficult to access for users or teams with different operating systems, as it requires their OS to be uniform and compatible with the CAD software.

It may seem intuitive that the cloud-native CAD architecture would perform better than the cloud-enabled CAD architecture and the traditional CAD architecture. However, there are several cases to where systems within these categories would be used in place of cloud-native ones.

As mentioned previously, most engineers are trained on traditional CAD systems. Without familiarity with cloud-based CAD systems, some engineers may elect to choose to use a traditional CAD and collaborate through external means. Additionally, many cloud-enabled CAD systems are merely an additional component that is added to a traditional CAD system, which requires additional licensure costs that some engineering teams may decide is not worth the collaborative cloud features. There are also arguments that with cloud-native CAD systems, the CAD engine is not as powerful for modeling and simulation due to the browser-based nature of these systems.

4. IMPLICATIONS FOR ENGINEERING DESIGN REVIEWS

The performance of each CAD architecture allows for an understanding of how each different CAD architecture could be used when performing a design review. With all three types of CAD architecture, the systems evaluated failed to enable reviewers to capture audio or video feedback within the system. As mentioned previously, the traditional CAD architecture systems did not meet more than half the requirements. As these are single-user systems inherently, they were expected to perform worse for this reason. Systems within the traditional CAD architecture lack the functionality to perform engineering design reviews. In order for this architectural approach to have this capability, it requires either external screen-sharing software, which would have limited to no functionality for commenting within the system or suggesting design changes, or through including add-ins within the CAD system to enable multi-user capabilities through the cloud, which would change the architectural format to a cloud-enabled system. One area where multi-user capabilities have been used to improve collaboration and data management is with software development. After the adoption of GitHub repositories, software development teams were able to collaborate more often, more effectively, and be able to work with more spread-out teams. This idea can be extended to CAD modeling, where developers of traditional CAD systems can use the growth of GitHub as a justification for introducing MUCAD features into existing systems.

Systems within the cloud-enabled architecture, while having the capability for synchronous work, also has limitations with this functionality. While these systems have at least some capabilities for synchronous viewing, these programs have either no "presenter" view or limited capabilities for this. The same is true for snapping to other users' views. This can lead to difficulty within virtual engineering design reviews, as the presenter(s) leading the review may have difficulty keeping the other members of the review on the correct model. Additionally, this can also be difficult on management or customers involved in the design reviews, as they would also require knowledge of how the software functions to follow the presenter(s). Overall, cloud-enabled architecture performs slightly worse or equal with cloud-native architecture in all requirements except for requirement 27. This represents how improvements can be made in this architectural framework to improve their simulation capabilities.

Among all three architectural frameworks, there was a broad lack of audio and video recording within their system. This is an important requirement in order to be able to perform engineering design reviews within a CAD system. Without the ability to record within a CAD system, this hinders the ability to present design reviews virtually and removes the opportunity to record design meetings without relying on external software. Additionally, the lack of a recording can cause knowledge loss from a design review, even if meeting minutes are taken [13].

Another impact on design reviews is the CAD software's capability of hosting multiple users. With traditional CAD systems, this cannot be done, meaning that design reviews that involve this type of CAD architecture would require design reviews to be run using external software. For these types of design review, the external software would be the main platform of the design review, with the CAD system being a supplementary tool. Both cloud-enabled and cloud-native CAD systems have MUCAD capabilities, which could allow design reviews to be run using external software to support the CAD system instead. However, these systems require the consumer to have in-house support for the CAD system to be able to function smoothly, which can be difficult for smaller companies or universities to use MUCAD systems. The difficulty with managing cloud-enabled and cloud-native CAD systems means that universities primarily offer and teach software within the traditional CAD architecture. This would prevent students from gaining experience with using CAD systems as a means of real-time collaboration and hosting design reviews.

5. CONCLUSION AND FUTURE WORK

In their current state, no type of architecture would be able to support a design review fully without using additional external software. This is an area of interest for the developers behind CAD software, particularly developers of CAD software within the cloud-enabled and cloud-native architectures. These evaluations are used to determine the strengths and setbacks of each CAD architecture as it relates to collaborative work and hosting design reviews. While CAD systems within the cloud-native architecture performed the best against the requirements, no system was able to fully meet all requirements, meaning that there are areas of improvement for all three architectures.

In comparison, CAD systems within the traditional CAD architecture performed the worst against the requirements. As systems within this architecture are frequently used in industry or in academic settings, this leads to many CAD users being trained on single-user CAD systems. This provides an area for improvement for CAD developers, as improving access or training for cloud-enabled or cloud-native CAD systems would improve MUCAD capabilities amongst users, which could increase the use and functionality of a design review held in a MUCAD system.

This also presents an area of improvement for systems within the cloud-enabled CAD architecture, where many of the systems are based on traditional CAD systems. The check-in/check-out process common in many systems in this architecture is a result of this, which reduces the multi-user capabilities of the CAD systems due to not allowing for synchronous work. This is another area of improvement for CAD system developers, as improving synchronous capabilities would improve the ability for users to work concurrently on the same design.

Because some CAD systems were evaluated through different means, this provides an area of future work. While the qualitative analysis of systems through published papers, user guides, and marketing information can provide insight into a system's capabilities, these pages often do not focus on collaboration with the system. The approach to use this method for some CAD systems in place of a full user evaluation was made due to licensing and training limitations, and reevaluations of all the systems with user evaluation can be made to determine if there is an improvement in the apparent collaborative capabilities.

An additional area of future work is through a quantitative case study. In this study, design reviews can be performed remotely in three contexts: in a video call with single-user CAD systems and shared screens, in a MUCAD system with audio support, and in a MUCAD system with no external support. The design reviews can be measured in the number of actions and time taken to perform certain tasks, such as changing between design versions, and in the length of the design review. The results of this study can be used to evaluate the effectiveness of remote design reviews held in MUCAD systems and where steps can be taken to improve MUCAD systems for this purpose.

6. REFERENCES

[1] V. S. Phadnis, K. A. Leonardo, D. R. Wallace, and A. L. Olechowski, "An Exploratory Study Comparing CAD Tools and Working Styles for Implementing Design Changes," *Proceedings of the Design Society: International Conference on Engineering Design*, vol. 1, no. 1, pp. 1383–1392, Jul. 2019, doi: 10.1017/dsi.2019.144.

[2] K. Cheng and A. Olechowski, "Analysis of Collaborative Assembly in Multi-User Computer-Aided Design," *Journal of Mechanical Design*, vol. 146, no. 031701, Nov. 2023, doi: 10.1115/1.4063759.

[3] V. Phadnis, D. Wallace, and A. Olechowski, "A Multimodal Experimental Approach to Study CAD Collaboration," *CADandA*, vol. 18, no. 2, pp. 328–342, Jul. 2020, doi: 10.14733/cadaps.2021.328-342.

[4] I. J. Freeman, J. L. Salmon, and J. Q. Coburn, "CAD Integration in Virtual Reality Design Reviews for Improved Engineering Model Interaction," in *Proceedings of the ASME 2016 International Mechanical Engineering Congress and Exposition*, 2016. doi: 10.1115/IMECE2016-66948.

[5] W. Hawthorne, M. Sutton, V. Ransing, G. Mocko, C. Turner, and J. Walton, "the use of virtual reality to support engineering design reviews," presented at the 2022 NDIA Michigan Chapter Ground Vehicle Systems Engineering and Technology Symposium, Novi, Michigan, United States, Aug. 2022, pp. 2024-01–3965. doi: 10.4271/2024-01-3965.

[6] E. Petty, "Cloud-Native MUCAD for Design Reviews," MS Thesis, Clemson University, All Thesis, 2025. [Online]. Available: https://open.clemson.edu/all_theses/4536

[7] E. Petty, N. Edmonds, G. Mocko, and E. Patterson, "Collaborative CAD in Support of Engineering Design Reviews," 2024. doi: 10.1115/DETC2024-142597.

[8] L. A. Odell, R. R. Wagner, and T. J. Weir, "Department of Defense Use of Commercial Cloud Computing Capabilities and Services," *Institute for Defense Analyses*, 2015.

[9] J. Robertson, J. M. Fossaceca, and K. W. Bennett, "A Cloud-Based Computing Framework for Artificial Intelligence Innovation in Support of Multidomain Operations," *IEEE*, vol. 69, no. 6, pp. 3913–3922, Dec. 2022, doi: 10.1109/TEM.2021.3088382.

[10] S. Xia, D. Sun, C. Sun, D. Chen, and H. Shen, "Leveraging single-user applications for multi-user collaboration: the coword approach," in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, in CSCW '04. New York, NY, USA: Association for Computing Machinery, Nov. 2004, pp. 162–171. doi: 10.1145/1031607.1031635.

[11] C.-C. Teng, F. N. Mensah, J. Ekstrom, R. Helps, and C. G. Jensen, "Hierarchical Role-Based Access Control for Multi-User Collaborative CAD Environment," *Computer-Aided Design and Applications*, vol. 13, no. 3, pp. 397–408, May 2016, doi: 10.1080/16864360.2015.1114398.

[12] Y. Hu, J. Zhang, X. Bai, S. Yu, and Z. Yang, "Influence analysis of Github repositories," *SpringerPlus*, vol. 5, no. 1, p. 1268, Aug. 2016, doi: 10.1186/s40064-016-2897-7.

[13] G. Huet, S. J. Culley, C. A. McMahon, and C. Fortin, "Making sense of engineering design review activities," *AIEDAM*, vol. 21, no. 3, pp. 243–266, Aug. 2007, doi: 10.1017/S0890060407000261.

7. CONTACT INFORMATION

Gregory Mocko
Professor
Clemson University
216 Fluor Daniel EIB, Clemson, SC
gmocko@clemson.edu

8. ACKNOWLEDGMENT

This work was supported by Clemson University's Virtual Prototyping of Autonomy Enabled Ground Systems (VIPR-GS) under Cooperative Agreement W56HZV-21-2-0001 with the US Army DEVCOM Ground Vehicle Systems Center (GVSC).

Appendix

Table A.1: Requirement fulfillment evaluations of traditional CAD architectures.

#	SolidWorks	Siemens NX	Autodesk Inventor
1	0	0	0
2	0	0	0
3	0	0	0
4	3	3	3
5	1	1	1
6	0	0	1
7	0	0	0
8	0	1	0
9	3	3	3
10	3	3	3
11	0	0	0
12	0	0	0
13	0	0	0
14	0	0	0
15	0	0	0
16	0	0	1
17	3	3	3
18	0	0	0
19	1	1	1
20	0	0	0
21	0	0	0
22	0	0	0
23	0	0	0
24	3	3	3
25	3	3	3
26	0	0	0
27	0	0	0
28	0	0	0
29	1	1	1
30	0	0	0
31	0	0	0
32	3	3	3
33	0	0	0
34	0	0	0
35	0	0	0
Key: 0 = Not Met, 1 = Partially Met, 2 = Mostly Met, 3 = Fully Met			

Table A.2: Requirement fulfillment evaluations of cloud-enabled CAD architectures.

#	Siemens NX + Teamcenter	Solidworks PDM	Creo+
1	2	2	3
2	2	2	3
3	1	0	3
4	3	3	3
5	1	1	2
6	1	1	0
7	1	3	3
8	1	1	0
9	3	3	3
10	2	3	3
11	3	3	3
12	3	3	3
13	2	3	3
14	3	3	3
15	3	3	3
16	3	3	3
17	3	3	3
18	1	3	3
19	2	2	2
20	1	0	0
21	1	3	3
22	1	0	1
23	1	0	3
24	3	3	3
25	3	3	3
26	2	1	3
27	1	1	3
28	1	1	3
29	3	3	3
30	1	1	3
31	3	3	2
32	3	3	3
33	3	2	3
34	3	3	3
35	3	3	3
Key: 0 = Not Met, 1 = Partially Met, 2 = Mostly Met, 3 = Fully Met			

Table A.3. Requirement fulfillment evaluations of cloud-native CAD architectures.

#	OnShape	Fusion	Solid Edge
1	3	3	3
2	3	3	2
3	3	3	2
4	3	3	3
5	3	3	3
6	1	0	2
7	3	3	3
8	3	3	3
9	3	3	3
10	3	3	3
11	3	3	3
12	3	3	3
13	3	3	3
14	3	3	1
15	3	3	3
16	3	3	3
17	3	3	3
18	3	3	3
19	3	2	1
20	3	3	1
21	3	3	3
22	3	0	1
23	3	3	2
24	3	3	3
25	3	3	3
26	3	3	3
27	1	2	2
28	3	3	1
29	3	3	3
30	3	2	2
31	3	3	3
32	3	3	3
33	3	3	3
34	3	3	3
35	3	2	3
Key: 0 = Not Met, 1 = Partially Met, 2 = Mostly Met, 3 = Fully Met			