18th International Conference on Space Operations, Montreal, Canada, 26 - 30 May 2025.

SpaceOps-2025, ID # 00213

Automation and Advanced SSDLC Methodologies for Security and Resilience in Satellite Control Systems

Silvia Abarca[a]*, Aurora María Salvador[b]

[a] *GMV, sabarca@gmv.com*
[b] *GMV, amsalvador@gmv.com*
* Corresponding Author

Abstract

In the rapidly evolving "new space" industry, where innovation and accelerated timelines are paramount, ensuring the security and resilience of satellite control systems is more critical than ever. This abstract explores advanced strategies and methodologies for embedding security and resilience throughout the lifecycle of satellite control systems, from ground and onboard systems to communication networks from plan and design to operation. We emphasize the crucial role of the Secure Software Development Life Cycle (SSDLC) in this process, highlighting some of the best practices such as threat modelling, risk analysis, supply chain security, and vulnerability management.

Different organizations, including governmental bodies and private sector entities like telecommunications companies, have unique requirements, timelines, and budgets. This diversity necessitates tailored frameworks for planning, design, implementation, testing, operation and maintenance. We will analyze how these varying needs are addressed through flexible and adaptable methodologies, ensuring that all stakeholders can achieve robust and secure satellite control systems.

We will present real-world use cases, including governmental and private organizations successful use cases, demonstrating how these methodologies are applied in various contexts. These examples will showcase the tangible benefits of integrating SSDLC into testing processes, ensuring comprehensive coverage of potential security and resilience issues.

The new space industry demands more agile and rapid processes to keep up with the pace of innovation. Automation and advanced testing methodologies play a pivotal role in meeting these demands. We will argue how these approaches not only speed up development and testing cycles but also enhance the overall quality and security of the systems. Practical examples will be provided to illustrate how automated testing can be seamlessly integrated into a shift-left approach, enabling early detection and mitigation of issues. This proactive stance significantly reduces the risk of critical failures later in the development process.

The ultimate goal of this presentation is to provide participants with a comprehensive understanding of how advanced testing methodologies, robust standards, and automation can dramatically improve the reliability and security of critical aerospace systems. By bringing secure development practices closer to the new space paradigm, we can ensure that satellite control systems are well-equipped to handle the challenges of the future. Participants will leave with actionable insights and practical knowledge that can be applied to their own projects, contributing to the advancement of secure and resilient satellite control systems in the new space era.

Keywords: Cybersecurity, SSDLC (Secure Software Development Life Cycle), resilience, AI-Driven Security & Automation, Multi-Sectorial Security Standards Alignment.

Acronyms/Abbreviations

- Secure Software Development Life Cycle (SSDLC)
- Static Application Security Testing (SAST)
- Dynamic Application Security Testing (DAST)
- Software Composition Analysis (SCA)
- Software Billing of Materials (SBOM)
- National Institute of Standards and Technology (NIST)
- NIST Cybersecurity Framework (CSF)
- Network and Information Security Directive 2 (NIS2)
- Digital Operational Resilience Act (DORA)
- Sarbanes-Oxley Act (SOX)
- Infrastructure as a Service (IaaS)

- Internet of Things (IoT)
- Operational Technology (OT)
- Public-private partnerships (PPPs)
- Non-Disclosure Agreements (NDA)
- Service-level agreements (SLAs)
- Artificial Intelligence (AI)
- Generative AI (GenAI)
- Scaled Agile Framework (SAFe)
- Health Insurance Portability and Accountability Act (HIPAA)
- Electronic health records (EHRs)
- Payment Card Industry Data Security Standard (PCI DSS)
- General Data Protection Regulation (GDPR)
- International Organization for Standardization (ISO)
- International Electrotechnical Commission (IEC)
- Information security management system (ISO/IEC 27001)
- International Traffic in Arms Regulations (ITAR)
- National Aeronautics and Space Administration (NASA)
- European Space Agency (ESA)
- ESA's Space Advisory Board (SAB)
- European Union Agency for the Space Program (EUSPA)
- North Atlantic Treaty Organization (NATO)
- Indian Space Research Organization (ISRO)
- Japan Aerospace Exploration Agency (JAXA)
- European Space Policy Institute (ESPI)
- Proof of Concept (PoC)
- Infrastructure as Code (IaC)
- Federal Aviation Administration (FAA)

1. Introduction

The "new space" industry is undergoing a rapid transformation, driven by innovation, commercialization, and cost reductions that allow more players—both governmental and private—to participate in space exploration and satellite deployment. This accelerated pace brings unique security and resilience challenges, particularly for satellite control systems, which include ground stations, onboard software, and communication networks. Each of these components introduces distinct vulnerabilities that, if exploited, could lead to mission failures, data breaches, or geopolitical consequences.

Ensuring the security and resilience of space systems is no longer an option but a necessity. Traditional cybersecurity measures fall short in addressing the growing sophistication of cyber threats targeting satellite communication links, supply chain components, and mission-critical infrastructure. State-sponsored cyberattacks, satellite jamming, and software exploitation have demonstrated that aerospace systems require a proactive security approach.

To address these challenges, the SSDLC provides a structured methodology for embedding security at every stage of software development, from initial planning and design to deployment, operation, and maintenance. By integrating threat modeling, risk analysis, and automated testing, SSDLC helps identify vulnerabilities early, reduce attack surfaces, and enhance mission resilience. These principles are widely applied in critical sectors such as finance and healthcare, where regulatory frameworks like NIS2 and DORA enforce robust cybersecurity measures centered around risk management. While the aerospace sector has traditionally focused on functional safety and reliability, it is now embracing SSDLC methodologies to address emerging security challenges.

This paper explores SSDLC practices in satellite control systems, addressing threat landscapes, supply chain security risks, and the benefits of automation and advanced security approaches. Through real-world case studies, it provides actionable insights to enhance the security and resilience of satellite control systems in the New Space era.

- *Military Missions*. Led by government defense agencies (e.g., DoD, NATO, national space forces) in collaboration with defense contractors, these missions operate under strict NDAs, security clearances, and compliance requirements (e.g., NIST 800-171, ITAR, NATO frameworks). SSDLC integrates formal verification, zero-trust architectures, adversarial threat modeling, and continuous cyber threat monitoring.

- *Scientific Missions*. Managed by space agencies (NASA, ESA, JAXA, ISRO), often involving academic institutions and private aerospace firms. SSDLC focuses on risk-based access control, cryptographic data protection, and secure telemetry, ensuring data integrity and controlled IP sharing while balancing international collaboration and security constraints.
- *Commercial Missions*. Driven by private-sector entities (e.g., SpaceX, OneWeb, Blue Origin, Planet Labs), these missions prioritize market-driven efficiency, compliance (ISO 27001, NIST CSF, GDPR), and supply chain security. Increasing dual-use applications require a heightened focus on secure cloud architectures and automated compliance monitoring.

The convergence of military, scientific, and commercial missions has accelerated the evolution of SSDLC, requiring continuous adaptation to security, regulatory, and operational demands. This paper examines how SSDLC can be optimized to strengthen cybersecurity as a foundational principle of space operations.

This paper is structured as follows:

- Section 2 – State-of-the-art review of SSDLC adoption across industries, focusing on space applications.
- Section 3 – Advanced SSDLC practices, including automation, AI-driven security, simulation techniques, continuous monitoring, resilience, and self-healing systems.
- Section 4 – SSDLC framework tailored for New Space, outlining best practices and security integration strategies.
- Section 5 – Results, discussion, and outlook, offering recommendations for strengthening secure software development in space missions.

2. State-of-the-art in SSDLC

The SSDLC has become a fundamental approach to ensuring robust security across various industries, embedding security measures at every phase of development to proactively address vulnerabilities while enhancing system resilience and integrity. While finance and healthcare have long led in SSDLC adoption due to strict compliance requirements and the need to protect sensitive data, the aerospace industry has traditionally prioritized reliability and safety over cybersecurity. However, as aerospace systems become increasingly digitalized, interconnected, and vulnerable to cyber threats, integrating security and resilience into software development has become essential.

The rise of large-scale satellite constellations, such as Starlink and OneWeb, has further transformed aerospace software development, introducing shorter development cycles, increasing system complexity, and demanding automation. Traditional manual verification and long testing processes are no longer sustainable, making continuous security validation and automated testing crucial. Additionally, modern satellites now feature software-defined payloads, AI-driven decision-making, and real-time reconfiguration capabilities, exposing new attack surfaces. Ensuring the integrity, availability, and adaptability of these systems requires secure software development principles, real-time threat monitoring, and secure update mechanisms throughout the entire software lifecycle.

The advent of Generative AI (GenAI) in software development presents both opportunities and risks for SSDLC. While AI-powered tools enhance developer productivity by automating code generation, predicting security weaknesses, and strengthening automated security testing, they also introduce challenges such as unclear accountability for AI-generated vulnerabilities and the risk of replicating insecure coding patterns. To address these concerns, aerospace organizations are adopting agile frameworks like SAFe and DevSecOps, ensuring continuous risk assessment, proactive threat mitigation, and automated security validation. By integrating security and resilience as intrinsic components of SSDLC, the aerospace sector aligns with more mature industries, reinforcing that proactive security integration is essential for protecting complex, high-stakes space systems in an era of increasing cyber threats.

2.1 SSDLC Across Industries

The implementation of SSDLC varies across industries, driven by regulations, evolving threats, and operational needs. While finance and healthcare have mature cybersecurity frameworks, aerospace and IT/cloud services are rapidly adapting. Drawing from GMV's three decades of expertise across multiple sectors, this section presents sector-specific SSDLC solutions, supported by academic research, industry reports, and regulatory guidelines to address cyber risks effectively.

2.1.1 Finance

The financial industry has one of the most mature SSDLC implementations, driven by stringent regulations such as PCI DSS, SOX, and DORA. Given the high value targets that financial institutions represent for cybercriminals, SSDLC practices are deeply embedded into software development processes. These include:

- Mandatory threat modeling and risk assessments at the early stages of development including operational risks that are continuously revisited throughout the lifecycle.
- Continuous security testing integrated into CI/CD pipelines, including scenario-based testing, threat-led penetration testing (TLPT), and performance testing under stress conditions.
- Strict access control measures for software supply chain security and third-party risk management with strict contractual clauses and conducting due diligence on third-party security practices.
- Training programs on secure development, secure coding boot camps and certification programs that are often mandated to maintain high security standards across all development teams.
- Dedicated SSDLC teams and security specialists who oversee the entire software lifecycle, responsible for defining security policies, conducting threat modeling, risk assessments, and secure architecture reviews supported by external security experts for penetration testing, red teaming, and independent security audits to validate the robustness of financial applications.

2.1.2 Healthcare

The healthcare industry follows similar SSDLC principles as the financial sector but is primarily driven by compliance with HIPAA, GDPR, and ISO 27799. With the rise of IoT-enabled medical devices and EHRs, and AI-driven diagnostics has made security a critical concern, as cyberattacks targeting healthcare infrastructure can lead to data breaches, operational disruptions, and patient safety risks. Key SSDLC measures include:

- Federated healthcare networks: As healthcare systems move toward interoperability, the challenge of securely sharing medical records between institutions has grown. GMV leads projects like TARTAGLIA [1] project that consist of federated healthcare network initiative, which enables secure and privacy-compliant data exchange across hospitals, research institutions, and regulatory bodies. This initiative enhances collaborative medical research while ensuring compliance with data protection laws.
- Strict security requirements for secure data encryption and strict patient data access controls: Protecting patient confidentiality and data integrity requires end-to-end encryption, strict access control policies, and role-based authentication mechanisms. Solutions such as uTile PET (GMV solution) [2] enable secure data exchange across healthcare systems, ensuring that sensitive patient data is encrypted, anonymized, and protected against unauthorized access.
- Medical device security and SBOM [50]: to address risks such as ransomware, supply chain attacks, and unauthorized remote access. SSDLC best practices include SBOM generation, device authentication, firmware integrity checks, and real-time anomaly detection.
- Security awareness for developers and IT staff, to proactively build security into code and systems.
- Automated vulnerability scanning in software updates to identify and remediate vulnerabilities. This is particularly relevant for implantable medical devices, smart hospital systems, and cloud-based EHR platforms, where cyber threats can directly impact patient care.

2.1.3 Information Technology (IT) and Cloud Computing

IT and cloud service companies have pioneered DevSecOps, integrating security into agile development (SSDLC). While more flexible than highly regulated industries, they face increasing compliance needs and adopt flexible and scalable security measures, including:

- Dedicated security teams handle secure architecture, compliance, and testing, often collaborating with external specialists for penetration testing and forensics to accelerate secure development.
- IT companies regularly conduct PoC to select security solutions that balance automation, vulnerability management, and development efficiency. These PoCs focus on CI/CD pipeline integration, accurate vulnerability detection, reduced false positives, and workflow automation.
- IaC frameworks (e.g., Terraform, AWS CloudFormation, Ansible) allow the automation of infrastructure provisioning. This embeds security policies into infrastructure definitions, minimizing human error, enabling scalable security configurations, rapid rollback and disaster recovery.
- Security checks are integrated directly into software pipelines to ensure continuous adherence to security standards for code and infrastructure, including automated SAST and DAST, SCA, container security scanning, and runtime monitoring, for early vulnerability detection and reducing risks.
- IT companies use cloud-native security solutions such as zero-trust architectures replace traditional perimeter security, enforcing continuous authentication, least privilege access, and micro-segmentation. Additionally,

mechanisms like IAM, behavior-based anomaly detection, and serverless security monitoring protect multi-cloud and hybrid-cloud environments.

- IT firms utilize continuous compliance automation solutions with frameworks like ISO 27001 and SOC 2 to track security controls, audit logs, and access policies against evolving regulatory standards. This reduces manual effort, enables immediate detection of violations, and ensures alignment with industry best practices.
- Continuous security improvement by recognizing that security is an ongoing process, IT companies regularly assess their security posture (AS-IS) to define roadmaps (TO-BE) for enhancing automation, compliance, and threat detection, supported by periodic strategy reviews aligned with business goals and the threat landscape.

2.1.4 Automobile

The automobile sector is undergoing a profound transformation, evolving from traditional mechanical engineering to increasingly software-defined vehicles. This shift has placed cybersecurity and robust Secure Software Development Lifecycle (SSDLC) practices at the forefront. Software vulnerabilities in vehicles can have direct, life-threatening consequences, making robust security an absolute imperative, now increasingly apparent by emerging regulations like UN R155 and ISO/SAE 21434. Key SSDLC measures within the automotive industry:

- Continuous Threat Modelling and Risk Assessment that considers attack vectors within the vehicle, its communication channels and the backend infrastructure.
- Rigorous security testing including in-vehicle testing, functional testing, fuzzing techniques and penetration testing.
- Supply chain security management and traceability.
- Best practices and SSDLC guidelines that require considering aspects such secure coding guidelines, apply trusted design and implementation principles for architectural cybersecurity and utilizing different layer of cybersecurity controls to improve vehicle cybersecurity.
- Vulnerability analysis and management throughout the vehicle lifecycle.

2.2 SSDLC in the Space Industry

2.2.1 Unique Challenges in Aerospace & Space Systems

The aerospace sector presents distinct challenges when integrating security practices into the SSDLC. Unlike other industries, space missions demand high reliability, long operational lifetimes, and stringent certification processes, making security integration a complex endeavor. Furthermore, the multi-stakeholder nature of space programs, involving government agencies, defense organizations, commercial entities, and research institutions, adds additional layers of complexity to security governance and implementation.

2.2.1.1 Key Challenges common to all type of missions

- **Long Development Cycles & High Certification Costs**. Aerospace systems, particularly satellites and spacecraft, have decades-long lifecycles, making security updates and patching difficult once deployed. The accreditation and certification processes required for space software add further complexity, as compliance with strict regulatory frameworks and multi-stakeholder governance often slows the adoption of modern security measures. Organizations such as EUSPA and ESA's SAB [4] oversee security and operational requirements, influencing mission approval and funding processes. Additionally, integrating agile development methodologies into highly regulated environments remains a challenge, as military, scientific, and commercial space programs operate under varying levels of oversight and compliance constraints.

- **Legacy Systems, Technological Debt & Closed Ecosystems.** Many space systems rely on aging software and legacy architectures, often developed decades ago with security not being a primary concern. Upgrading or replacing these systems requires substantial investment and often involves re-certification, extensive regression testing, and risk assessment. Bridging the gap between old and modern security architectures requires strategies such as virtualization, containerization, and secure software abstraction layers to retrofit security measures into existing infrastructure. The lack of standardized security frameworks across agencies and companies leads to fragmented approaches.

- **Cybersecurity Risks for Satellites, Ground Stations, and Communication Links.** Space systems are increasingly vulnerable to hijacking, spoofing, jamming, and cyberattacks targeting telemetry, tracking, and control (TT&C) links, satellite communication networks, and mission-critical software. Supply chain attacks pose a growing risk, as third-party components and software dependencies may introduce vulnerabilities that can be exploited during manufacturing, deployment, or operation. The adoption of Zero Trust Architecture (ZTA) and secure-by-design principles remains a work in progress, as space agencies and private companies struggle to balance security, cost, and operational constraints.

- **Commercial solutions cannot be used as they are.** While commercial and cross-sector solutions can be leveraged for space cybersecurity, they cannot be directly applied without significant modifications and customization. The unique operational constraints of aerospace systems—such as real-time constraints, harsh environmental conditions, and limited update capabilities—require tailored security adaptations. As noted by the ESPI, integrating commercial solutions into military and space applications remains a challenge that demands careful adaptation and validation [6]. Cybersecurity solutions originally designed for other critical sectors—such as healthcare, banking, and IT infrastructure—are increasingly being adapted for space applications needing significant effort. For instance:
  - GMV's Utile cybersecurity platform, initially developed for federated healthcare networks, has been repurposed to enhance secure data sharing and infrastructure protection in space systems [2][3]
  - GMV's Checker solution, originally designed to secure banking ATMs, has served as the foundation for developing cybersecurity solutions tailored for space assets [8][9].

- **Evolving Threat Landscape:** The rise of state-sponsored cyberattacks on space assets highlights the urgent need for comprehensive cybersecurity measures. Regulatory frameworks such as the EU's Decision 2013/488 [10] on Security Rules stress the importance of classified information protection, encryption, and personnel security clearance procedures to safeguard sensitive space systems. However, policy measures alone are insufficient—technical solutions must be continuously adapted and integrated into SSDLC processes to ensure space systems remain resilient against evolving threats.

- **Specific requirements, standards and procedures:** The aerospace sector operates under strict regulatory frameworks and security standards each tailored to the specific needs, governance structures, and operational constraints of different entities. While all aim to ensure reliability, safety, and security, they are not interchangeable, as each organization establishes its own best practices, validation processes, and compliance obligations. These standards directly impact the SSDLC for all type of space missions, with variations in implementation approaches, security rigor, and certification processes.

Some examples are:

- NASA's Secure Software Development Self-Attestation Resources and Knowledge [41]:
  - Establishes requirements for contractors and third-party developers to ensure secure-by-design principles in all NASA-related software.
  - Provides self-attestation checklists for organizations developing software intended for NASA missions, covering security validation, supply chain security, and risk management.

- NASA's Software Engineering Procedural Requirements (SEPR) that establish best practices for software security, mission-critical software validation, and lifecycle management [5]. These requirements emphasize:
  - Formal software validation and verification (V&V) to ensure mission reliability.
  - Security and risk assessment at every phase of software development.
  - Strict coding standards and review processes to minimize software-related vulnerabilities.
  - Continuous monitoring and updates for critical software components, particularly in long-duration missions.

- ESA follows a comprehensive set of standards known as the European Cooperation for Space Standardization (ECSS) [11], which defines requirements for software engineering, cybersecurity, and mission assurance. These include:
  - ECSS-E-ST-40C Rev.1 [12]: Defines general software engineering requirements for space systems, covering the full software lifecycle, development methodologies, and validation processes.
  - ECSS-Q-ST-80C Rev.1 [13]: Establishes software product assurance guidelines, ensuring software reliability, quality control, and compliance with safety-critical requirements.
  - ECSS-E-ST-80C [14]: Focuses on security in space systems lifecycles, addressing threat modeling, risk assessment, cybersecurity measures, and security testing throughout the software development process.

2.2.1.2 Military missions' challenges

Military space missions prioritize national security and require stringent security protocols to prevent espionage, sabotage, and cyber warfare attacks. However, the integration of commercial space assets into military operations introduces challenges in aligning defense-grade security standards with commercial practices.

- The U.S. Department of Defense's Commercial Space Integration Strategy (CSIS) and the U.S. Space Force's Commercial Space Strategy (CSS) (both published in April 2024) provide guidelines for collaboration between military and commercial space operators.
- These strategies aim to bridge security gaps by outlining principles for cybersecurity integration, risk management, and interoperability between classified defense networks and commercial satellite systems. [7]
- Challenges remain in ensuring that commercial vendors can meet military-grade security expectations, especially in supply chain security and secure software development.

2.2.1.3 Scientific Missions Challenges

Scientific missions, focus on data integrity and collaborative research. The challenge is to balance security with accessibility, ensuring that research data remains secure yet available to the global scientific community.

- Open science initiatives encourage data sharing but also introduce risks of data tampering and cyber threats.
- Secure software development practices must integrate encryption, digital signatures, and controlled access mechanisms to protect sensitive scientific data while maintaining collaboration.

2.2.1.4 Commercial Missions

Commercial space missions, emphasize innovation, cost-efficiency, and rapid development cycles. However, fast-paced commercial operations often lead to security trade-offs, where speed-to-market takes precedence over security.

- PPPs have emerged as a solution to balance security and commercial innovation, allowing government agencies to leverage commercial expertise while ensuring compliance with cybersecurity frameworks.
- The European Space Policy Institute (ESPI) highlights the importance of integrating commercial space assets into defense and governmental operations, emphasizing the need for robust security policies and regulatory alignment [6].
- Cybersecurity in commercial missions must scale with industry growth, requiring automated security enforcement, real-time risk assessment, and integration of cybersecurity controls into development pipelines.

2.2.2 Current SSDLC Practices in Space Systems

The adoption of SSDLC in space programs is increasing, but it remains highly adapted to mission-critical constraints. Space systems must balance security, performance, reliability, and long mission lifetimes, making traditional agile security integration difficult. As a result, SSDLC practices in the space industry incorporate formal software assurance processes, cybersecurity risk management, and secure software engineering frameworks tailored to space-specific challenges.

Best Practices in SSDLC for Space Systems

- **Risk-Based Security & Threat Modeling**
  - SSDLC practices incorporate risk-based security strategies, ensuring that threat assessments and risk modeling are integrated at each development phase.
  - NASA's Space Security Best Practices Guide (2023) outlines structured risk assessments and mission continuity planning [15] [16].
  - The Common Vulnerability Scoring System (CVSS) is widely applied in ESA and EU space programs to classify security risks and define mitigation priorities.

- **Formal Code Review & Security Testing**
  - Space agencies and organizations enforce static and dynamic security testing to identify vulnerabilities before deployment.
  - Standards such as ECSS-E-ST-40C and NASA's Software Engineering Procedural Requirements mandate formal verification, code audits, and penetration testing to ensure software integrity.
  - Security measures include vulnerability scanning, penetration testing, and continuous monitoring to detect exploitable weaknesses in space-based systems.

- **Adoption of Secure Coding Standards**
  - Secure coding guidelines are critical to prevent code injection, buffer overflows, and unauthorized access in space software.
  - ECSS-E-ST-40C establishes secure coding practices that align with space system engineering requirements.

- NASA's Software Engineering Handbook emphasizes coding best practices, secure software architecture, and validation to reduce cybersecurity risks.

➢ **Security Compliance Reporting & Continuous Monitoring**
- Space organizations enforce regular security audits and continuous vulnerability assessment and management to adapt to evolving cyber threats.
- ECSS-Q-ST-80C mandates quarterly security vulnerability reporting, detailing CVSS scores, mitigation strategies, and patching requirements.
- ESA's Cyber Resilience Strategy integrates threat intelligence, continuous security monitoring, and real-time anomaly detection to enhance space mission security posture.

➢ **Resilience and Business Continuity Considerations**
- In recent years, Business Continuity and Resilience have surged in importance within Space Organization cybersecurity.
- While the maturity status of these Business Continuity Plans (BCPs) is often less formally defined than in sectors like IT, which have established frameworks for organizational continuity, general training, and awareness, a significant evolution is underway.
- Space organizations are moving beyond simple Disaster Recovery Plans (DRPs). They are progressively developing comprehensive BCPs that encompass broader organizational resilience, acknowledging the unique challenges and long-term implications of disruptions in the space domain.

2.2.2.1 Real-World SSDLC Implementation in Space Projects

NASA Artemis Program (Lunar Exploration Missions)

The NASA Artemis Program, designed to return humans to the Moon and establish a sustainable lunar presence, has incorporated robust SSDLC practices (SSDLC Adoption through aligning with NIST 800-218 Secure Software Development Framework), including Independent Verification & Validation (IV&V), risk-based testing, and automated security validation to enhance mission security and reliability [38][39][40]. However, recent evaluations by the U.S. Government Accountability Office (GAO) and the NASA Office of Inspector General (OIG) have identified areas for improvement, highlighting the need for stronger integration of cybersecurity into spacecraft acquisition policies and more proactive risk management approaches to address evolving cyber threats.

In May 2024, the U.S. Government Accountability Office (GAO) reported that while NASA had established cybersecurity requirements for spacecraft in 2019, these had not yet been fully incorporated into acquisition policies and standards. The report emphasized the need for NASA to develop a plan with time frames to update its spacecraft acquisition policies to include essential cybersecurity controls, ensuring a comprehensive defense against potential cyber threats. [35]

In November 2021, the NASA Office of Inspector General (OIG) highlighted that the Artemis missions faced technical difficulties and delays, which could impact the timely integration of cybersecurity measures. The report underscored the importance of addressing these challenges to maintain mission schedules and ensure that cybersecurity protocols are effectively implemented throughout the software development lifecycle. [36]

Analyses from external sources have raised concerns about the cyber threats facing the Artemis program. These assessments suggest that nation-states and organized crime groups could target NASA and its commercial partners, seeking to exploit vulnerabilities for financial gain or geopolitical advantage. The interconnected nature of IT systems in space missions necessitates robust SSDLC practices to mitigate such risks. [37]

To bolster SSDLC practices in aerospace projects like the Artemis program, it is recommended to:

- **Update Acquisition Policies.** Develop and implement a structured plan to incorporate comprehensive cybersecurity requirements into spacecraft acquisition policies and standards, as advised by the GAO.
- **Strengthen Program Management.** Address technical and scheduling challenges proactively to ensure that cybersecurity measures are integrated effectively throughout the software development lifecycle, aligning with OIG recommendations.
- **Conduct Continuous Threat Assessments.** Regularly evaluate emerging cyber threats from various actors, including nation-states and cybercriminals, to adapt and enhance SSDLC practices accordingly.

By implementing these recommendations, aerospace programs can enhance their cybersecurity posture, ensuring the resilience and success of missions like Artemis.

Starlink (Commercial Broadband Constellation by SpaceX)

Starlink, SpaceX's low-Earth orbit (LEO) satellite broadband network, seems to incorporate SSDLC methodologies to ensure system security, resilience, and continuous service availability. The company follows ISO/IEC 27001:2013 and PCI DSS standards, embedding security best practices at every stage of software development. [29][30]

As part of its SSDLC framework, Starlink enforces secure coding policies (A.14.2.1), requiring developers to follow well-defined security guidelines throughout the software development process. Additionally, system security testing (A.14.2.8 and A.14.2.9) is a mandatory pre-deployment step, ensuring that every software release undergoes rigorous validation before being deployed across the satellite fleet. The company also maintains a technical vulnerability management process (A.12.6.1 and A.18.2.3) to identify, track, and remediate software vulnerabilities, enforcing continuous security monitoring and automated patch management. Even though the current certification is based on the 2013 version (which already provides a robust risk management framework and effective security controls), the updated ISO/IEC 27001:2022 version introduces enhanced controls specifically focused on secure coding practices.

Some of the SSDLC key practices in Starlink are:

- **Secure Coding & Software Development Standards**
  - Enforces secure coding policies (ISO/IEC 27001 A.14.2.1), requiring developers to follow standardized security guidelines throughout the software lifecycle.
  - Implements automated security testing before deployment to detect vulnerabilities, misconfigurations, and compliance deviations [30].

- **Security Testing & Continuous Validation [29]**
  - System security testing (ISO/IEC 27001 A.14.2.8, A.14.2.9) is a mandatory pre-deployment step, ensuring that each software release undergoes comprehensive validation before deployment across the satellite fleet.
  - Technical vulnerability management process (ISO/IEC 27001 A.12.6.1, A.18.2.3) ensures continuous vulnerability tracking, risk assessment, and automated patch management [34].

- **Continuous Integration & Deployment (CI/CD) with Large-Scale Simulation**
  - Starlink utilizes high-performance computing clusters to run full-scale network simulations, validating software updates before production rollout [33].
  - This proactive testing approach enables the early detection of software anomalies, mitigating potential performance or security issues before deployment.

- **Bug Bounty [31] & External Security Validation [32]**
  - Starlink actively engages security researchers through its bug bounty program, incentivizing ethical hacking efforts to uncover vulnerabilities.
  - The company collaborates with third-party cybersecurity experts to continuously strengthen its software security posture.

- **By Design – Shift-Left Approach for Critical Aspects, including Security, due to Starlink Software-Specific Challenges [33]**
  - Frequent software updates: Unlike traditional aerospace systems, Starlink deploys software updates weekly or biweekly, requiring agile SSDLC methodologies.
  - Real-time satellite handover: Since satellites move at 27,000 km/h, software must seamlessly transition network connections between ground stations and satellites without service disruption.
  - In-orbit software reconfiguration: Starlink dynamically adjusts its software and network parameters to optimize connectivity based on real-time environmental conditions.
  - Low-latency networking challenges: Software is highly optimized to minimize propagation delays between satellites and ground stations.

- **Resilience by design - Resilient & Fault-Tolerant Architecture [33]**

- Starlink employs a self-redundant architecture, leveraging triplicated computing nodes that continuously cross-validate calculations and detect discrepancies in real time.
- This built-in fault tolerance enhances network resilience, allowing the system to autonomously recover from failures or cyber threats without disrupting user service availability

2.2.3 Gaps & Considerations in Adopting SSDLC for Space

Despite advancements, several challenges remain in fully integrating SSDLC into space software development.

➢ **Balancing Security with Functional Safety**
- Safety-critical systems prioritize reliability over security, leading to vulnerabilities in military and mission-critical software.
- Security validation must be integrated to ensure comprehensive protection without compromising operational reliability.
- For example, DO-178C [17], a widely used aviation software safety standard, ensures functional reliability but lacks detailed cybersecurity controls, creating potential security gaps.

➢ **Lack of Standardized Space Cybersecurity Regulations**
- Unlike other critical sectors, no globally unified cybersecurity standard exists for space systems, leading to inconsistent security practices across agencies and private operators.
- ESA, NASA, and commercial space operators develop their own security frameworks, increasing complexity and hindering interoperability.
- The absence of a universal regulatory framework makes it difficult to enforce consistent security measures across international space missions.

➢ **Challenges in Agile and DevSecOps Adoption**
- Traditional SSDLC models struggle in aerospace, where waterfall methodologies dominate due to strict certification and regulatory constraints.
- Agile and DevSecOps approaches face adoption challenges because security updates require costly recertification, limiting rapid iteration and security patching.
- Space agencies and contractors must adapt software development models to balance agility with mission-critical certification requirements.

➢ **Security Gaps in Rapid Commercial Space Innovation**
- In the commercial space sector, the fast pace of innovation often outpaces security protocol development, leading to potential vulnerabilities.
- Integrating robust security without hindering innovation is a major challenge, as commercial operators prioritize cost-efficiency and time-to-market.
- Public-private collaboration is essential to establish security standards that evolve alongside technological advancements in the New Space industry.

2.2.4 Diverse Standards and Regulatory Frameworks for Secure Space Software Development

The secure development of space systems is not governed by a single universal standard, but rather by a diverse set of international regulations, industry best practices, and space agency-specific guidelines. Organizations such as ESA, NASA, FAA, NIST, and ISO have each established cybersecurity frameworks and engineering standards to address the unique security challenges associated with developing, deploying, and maintaining software for satellite control systems, space exploration missions, and commercial space programs.

Due to the complex and evolving threat landscape, security in space software development requires a multi-framework approach, combining elements from software engineering, cybersecurity, supply chain security, and operational resilience. The following key security standards and regulatory frameworks provide a foundation for secure space system development:

➢ **ECSS standards such ECSS-E-ST-40C, ECSS-Q-ST-80C and ECSS-E-ST-80C**
- ESA's software engineering standards define development, verification, and validation requirements for space systems.
- Incorporate security controls to ensure software integrity, reliability, and resilience, especially in long-duration and autonomous missions.
- Establish guidelines for software product assurance and lifecycle security in space environments.

- **NASA Space Security Best Practices Guide**
  - Defines mission-specific cybersecurity requirements for NASA space systems.
  - Includes threat modeling, software assurance, and continuous monitoring to protect against cyberattacks targeting spacecraft and ground systems.

- **NASA's Secure Software Development Self-Attestation Resources and Knowledge**
  - Integrates NIST 800-218 Secure Software Development Framework (SSDF) to enforce secure software supply chain practices.
  - References CISA's Secure by Design initiative to promote security-first software engineering.
  - Recommends automated security testing, vulnerability management, and supply chain risk mitigation for aerospace software ecosystems.

- **NIST IR 8401 [18]**
  - Applies the NIST Cybersecurity Framework (CSF) to satellite ground segments, providing security guidelines for satellite command and control systems.
  - Focuses on authentication, encryption, anomaly detection, and secure satellite communication protocols.

And for Regulatory Landscape:

- **ESA Cybersecurity Framework [19]**
  - Mandates secure software architecture, encryption, and incident response plans for space programs.
  - Ensures compliance with ECSS security standards and EU space cybersecurity policies.

- **FAA & NIST Space Security Recommendations**
  - Provide guidance on secure software development for commercial spaceflight operators, emphasizing secure software lifecycles, supply chain security, and in-orbit software resilience.
  - Align with FAA [20] safety regulations for space launch systems and NIST cybersecurity frameworks, ensuring end-to-end mission security.

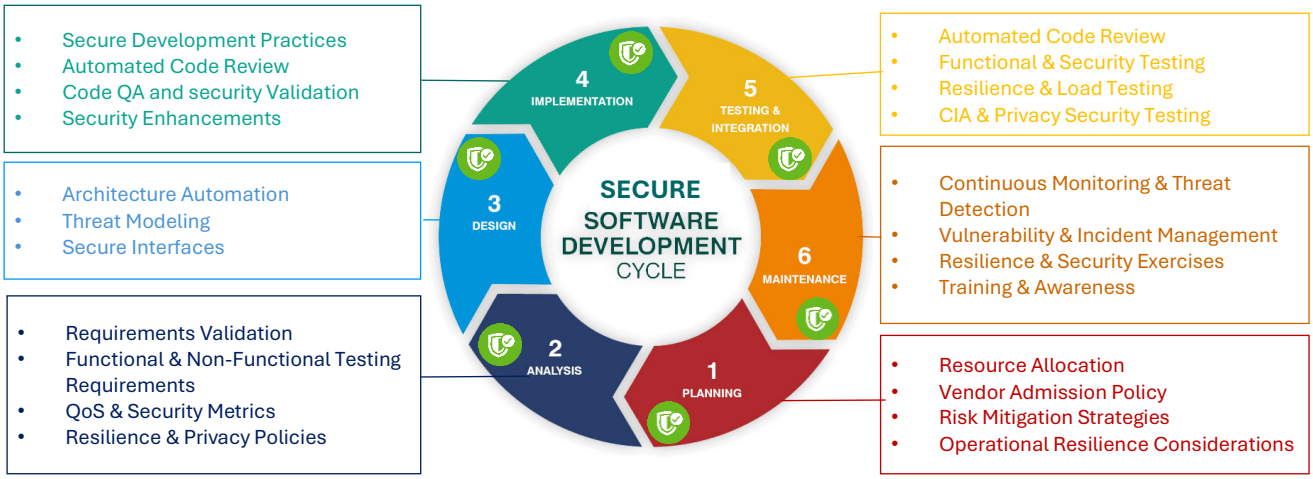
3. Advanced SSDLC Practices

The growing complexity of space software, the shortening time-to-market, and the increasing reliance on software-driven operations demand the adoption of advanced SSDLC practices. As AI, automation, and advanced simulation technologies become integral to aerospace software development, traditional testing and validation methods are no longer sufficient. These innovations require sophisticated methodologies for security testing, continuous validation, and compliance enforcement to ensure that mission-critical aerospace systems remain efficient, secure, and resilient against evolving cyber threats. Implementing AI-driven security validation, automated compliance checks, and digital simulation models enhances software reliability, adaptability, and long-term sustainability in space operations.

3.1 Enhancing SSDLC Through Automation

Automation is a fundamental enabler for SSDLC implementation in aerospace projects, ensuring continuous testing, validation, and compliance monitoring while minimizing operational disruptions. Automated integration and validation facilitate verification, rapid testing, and deployment, of software updates, reducing the risk of human error and enhancing mission readiness.

Automation can be applied at various stages of the SDLC, optimizing security, performance, and reliability and always focusing on shifting the security to the left to early stages of the SDLC:



In the Planning phase, automation may manage resource allocation, supplier validation, and AI-driven risk mitigation, ensuring disaster recovery and resilience planning. At analysis automation can be applied for security requirement validation, compliance enforcement, anomaly detection, and resilience impact assessments to maintain system integrity.

The Design phase leverages Infrastructure as Code (IaC) for security-driven architecture, AI-powered threat modeling, and Zero Trust frameworks, ensuring resilient and secure system interactions. During Implementation, automation embeds code analysis with SAST and SW dependencies security scans in CI/CD pipelines, enabling developers to identify, mitigate, and learn about insecure coding practices.

Testing & Integration may include automatic static and dynamic security scans, fuzz testing, penetration testing, and CIA validation, while resilience and load testing ensure disaster recovery readiness. However, while automation significantly improves efficiency and detection rates, there is always a manual component where expert security analysts review, validate, and reassess automated results to eliminate false positives and ensure critical vulnerabilities are correctly prioritized and addressed.

Operations & Maintenance leverage AI-driven SOC monitoring, automated patch assessments, and continuous security tracking to provide real-time insights into system vulnerabilities and emerging threats. Proactive Red/Blue/Purple team exercises simulate real-world cyberattacks, generating key performance indicators (KPIs) that help identify security weaknesses and areas requiring improvement. By analyzing security metrics, organizations can assess the effectiveness of implemented measures, ensuring that resources are allocated efficiently. This data-driven decision-making approach allows for targeted security enhancements, ensuring that training efforts are redirected to address specific skill gaps or recurring vulnerabilities.

To address these challenges, GMV has developed advanced cybersecurity solutions, integrating automated security validation, continuous monitoring, and risk-based vulnerability prioritization, ensuring mission-critical software remains resilient and secure.

To address some of these automated activities, GMV has developed cutting-edge solutions that ensure efficient software validation, continuous security monitoring, and vulnerability prioritization, providing a comprehensive approach to cybersecurity.

GMV has developed an Automated Integration and Validation Platform that leverages orchestration, test automation solutions, and Infrastructure-as-Code (IaC) to:

- Automate the generation of test environments and simulation platforms for software validation.
- Schedule software deployments and upgrades, ensuring consistent software lifecycle management.
- Execute predefined test batteries in simulators to validate software behavior under various conditions, including:
  - Functional Testing: Ensuring compliance with mission-critical and operational requirements.
  - Performance Testing: Assessing response times, resource utilization, and stability under normal and extreme conditions.

- Stress and Resilience Testing: Evaluating system robustness, failure recovery, and resistance to high-load scenarios.
- Regression Testing: Confirming that software updates do not introduce unintended defects or break existing functionality.
- Security Testing: Verifying input/output correctness, software logic, and expected performance. Uncover vulnerabilities through unexpected inputs. And detect misconfigurations, improper secrets management, and exposure of sensitive data.

Beyond automated software validation, GMV has integrated advanced security testing and vulnerability management solutions in multiple missions to enhance cyber resilience throughout the SSDLC. These solutions ensure continuous security enforcement, automated threat detection, and proactive risk mitigation across software development and deployment.

Software quality analysis is strengthened through custom quality gates and security profiles in SonarQube, enforcing coding best practices. Automated code reviews, embedded within CI/CD pipelines, detect and block non-compliant code before it reaches production, reducing security risks early in development. Automated security scans play a critical role in identifying vulnerabilities across multiple layers. SAST detects security flaws in source code, automates triage, and integrates security reports into CI/CD pipelines. DAST simulates real-world cyberattacks to uncover runtime vulnerabilities, ensuring applications are protected against evolving threats. Additionally, to mitigate supply chain risks, automated third-party dependency analysis is implemented, detecting untrusted open-source components before they enter the repository or build process. However, automated scans are performed, as complete security assessments require expert review, automation complements, rather than replaces, manual security evaluations, allowing for less frequent but more in-depth manual analysis while maintaining continuous security oversight.

AI-powered penetration testing is enabled through GMV Penbot [21], which automates penetration testing using Reinforcement Learning, continuously refining attack strategies. It identifies OWASP Top 10 vulnerabilities such as SQL injection, XSS, authentication flaws, and misconfigurations. By automating routine security checks, GMV Penbot reduces manual effort, allowing security teams to focus on high-priority vulnerabilities and streamline security validation processes.

To streamline vulnerability detection, tracking, and prioritization, GMV has developed Gestvul [22], an advanced cybersecurity solution for automated vulnerability management.

- Aggregates vulnerabilities from multiple sources such manual security analysis, penetration testing (pentest), SAST (source code analysis) and Infrastructure and application security assessments
- Able to identify and categorize automatically vulnerabilities as new, reopened, or closed, allowing for effective risk prioritization.
- Automates the generation of vulnerability reports, ensuring security teams receive structured, actionable insights.
- Provides real-time notifications for newly discovered vulnerabilities requiring immediate action, reopened vulnerabilities that persist after remediation attempts, pending security issues that need follow-up, ensuring that vulnerabilities are properly addressed.
- Ensures efficient remediation tracking by enabling continuous security validation, ensuring that unresolved vulnerabilities are escalated and resolved in a timely manner. And facilitating automated follow-ups and audit tracking, guaranteeing compliance with internal and external security standards.

Complementary to Gestvul, GMV has also developed a tool specific for Galileo program named Cyber Monitoring and Asset Management (CMAM) that enhances security oversight and asset tracking:

- Extracts real-time asset information, providing a detailed inventory through an agentless approach.
- Automatically detects deviations from baseline security configurations, identifying security misconfigurations and non-recommended practices.
- Manages system changes through an integrated workflow, helping to analyze and track authorized and unauthorized modifications.
- Enables compliance monitoring, ensuring adherence to industry hardening standards and security policies.

3.2 AI-Driven Software Development and Security Validation in SSDLC

The integration of Generative AI into software development has transformed code generation, security validation, and compliance enforcement. AI enables faster software development, but it also introduces new security risks such as insecure coding patterns, unverified dependencies, and supply chain vulnerabilities. To address these challenges, organizations must implement structured validation processes, automated compliance checks, and third-party software admission controls to ensure that AI-generated and third-party code meets the highest security and quality standards before integration into mission-critical systems.

GMV's R&D program has developed several AI-powered solutions to enhance software quality, security, and compliance while leveraging AI-driven development. *GenAICode* ensures AI-generated code adheres to GMV-defined secure coding standards, producing compliant software. *GenAICodeVal*, integrated with SonarQube, automatically scans AI-generated code for vulnerabilities in OWASP Top 10, SANS Top 25, enforces predefined security rules, and validates against industry-specific standards like MISRA C. To strengthen software reliability, *GenAIUnitTest* automates unit test creation for both AI-generated and manually written code, improving test coverage with minimal human intervention.

3.3 Digital Twins for Enhanced Mission Safety and Sustainability

Advanced SSDLC practices for satellite control systems are increasingly leveraging Digital Twins (DT) and Artificial Intelligence/Machine Learning (AI/ML) technologies to enhance security, resilience, and development efficiency [24].

Particularly the space sector (satellites and ground segment), can significantly benefit from using advanced testing with technologies such as Digital Twins (DT) to enhance mission safety, success rates, and promote sustainable space exploration through efficient resource management and environmental monitoring.

SpaceX uses digital twins for design and testing of spacecraft, allowing prediction and resolution of potential issues in a virtual environment [26].

Some of the key benefits and applications of Digital Twins in conjunction with SSDLC in the space sector are:

- **Design and Testing.** Digital twins offer several advantages in the realm of testing, particularly through virtual testing methodologies. These advantages span various aspects of product development, manufacturing, and cybersecurity, and can lead to more efficient, cost-effective, and reliable outcomes [28]. Furthermore, they enable comprehensive testing and validation, which quickly identifies malfunctions and inefficiencies, rules out design mistakes, and tests the practicality of physical solutions. This allows for early verification of component-level requirements and specifications [25]. In addition to that, digital twins have also been proposed in the automotive industry to analyze the components and generate and execute tests to verify its security [27].
- **Real-time Monitoring and Fault Prediction.** Digital twins provide real-time monitoring and fault prediction, which is crucial in extreme space conditions. They reflect the real-time status, dynamic processes, and behaviors of corresponding entities, offering support for design optimization, status monitoring, fault prediction, and health management. By analyzing differences between the model and actual behavior, potential failures can be predicted, helping decision-makers quickly pinpoint problems and devise solutions [26].
- **Resource and Cost Efficiency.** Digital twin models allow extensive experiments and tests on the ground, reducing the need for actual space environment trials, thus saving substantial resources. Maintenance plans and schedules can be optimized by simulating different maintenance strategies and pathways, reducing maintenance time and enhancing operational efficiency [26].
- **Anomaly Detection:** Through pattern recognition and machine learning, digital twins can identify abnormal behavior and detect potential faults in a timely manner [25][26].

Challenges and Considerations:

- **Model Accuracy and Reliability.** The accuracy and reliability of digital twin models are crucial for their technical applications, which require large amounts of data that might not be readily available [26].
- **Technical Difficulties**: There can be difficulties in constructing a space digital twin system, especially with entity mapping, data management, and space operation and maintenance.
- **Cost and resources.** Implementing digital twin technology requires significant initial investment in research and development, hardware deployment, and maintenance. Continuous data processing and analysis also consume many resources [26].

Uses cases:

- **Reusable Rockets**. To improve the design and reusability of rockets, organizations like the German Aerospace Center (GAC) and SpaceX are utilizing digital twin technology. This approach involves creating virtual models of rockets and their launch procedures, allowing for the simulation of multiple launches. This enables engineers to experiment with various flight paths and design iterations in a safe, digital setting, ultimately leading to optimized rocket performance and increased reuse efficiency [23][26].
- **Satellite Navigation and Orbit Optimization**: Digital twins precisely simulate the complex orbital dynamics of satellites. These models provide help optimize satellite orbits, enhancing the overall performance of satellite navigation systems [26].
- The European Space Agency (ESA) has made significant advancements in the field of on-orbit services by incorporating digital twin technology into its spacecraft maintenance robots [26].

3.4 Continuous Monitoring, Resilience, and Self-Healing Systems

Satellite Control Systems demand a robust, resilient architecture beyond code security capable of withstanding a broad spectrum of threats that require systems that can not only detect faults as they occur but also recover and adapt autonomously [42][43][44].

This is achieved through automatic fault detection and self-healing systems, designed to ensure reliability, scalability, and continuous operation in mission-critical applications [43]. These systems leverage a combination of advanced techniques such as continuous monitoring, anomaly detection, predictive analytics, machine learning, and autonomous recovery mechanisms to minimize operational costs, boost performance, and reduce downtime [43] [46].

Continuous Monitoring and Fault Detection

Continuous monitoring is based on gathering real-time data to track the health of every component [47]. This real-time monitoring forms the basis of fault detection, where sophisticated anomaly detection techniques that use statistical models and machine learning identify deviations from expected system behavior [47][48].

Adaptive mechanisms further refine these techniques by dynamically adjusting detection strategies to distinguish between temporary fluctuations and permanent faults, increasing detection accuracy [47][48]. In parallel, predictive analytics based on trends and historical data enable systems to forecast potential issues, allowing for proactive interventions before minor issues escalate into major failures [46].

Fault Diagnosis and Autonomous Recovery

Once a fault is detected a rapid and accurate diagnosis is critical. By integrating data from hardware metrics, software logs, and network activity, systems can pinpoint the root causes of anomalies [47][48]. AI-driven analytics play a crucial role in this phase by reducing false positives and accelerating the response process [46][47]. After diagnosis, the system's autonomous recovery capabilities kick in.

Automated responses such as rerouting network traffic, restarting components or reallocating computing resources, enable fast restoration of system functionality [43][46][47]. Additionally, built-in redundancy and failover strategies ensure that even if a primary component fails, backup resources immediately take over, maintaining continuous service availability [43][47].

Self-Healing Systems

Self-healing systems go beyond immediate fault handling and make dynamic adjustments by monitoring performance criteria and reconfiguring themselves to optimize resource usage [45]. Scalability is a central feature; these systems are designed to operate efficiently across large, distributed networks without sacrificing performance [49].

Learning and adaptation are further enhanced through machine learning techniques, which allow the system to refine fault detection and recovery strategies over time. By analyzing past failure records, systems not only improve their diagnostic accuracy but also implement predictive maintenance measures such as replacing hardware components before they reach a failure threshold to extend operational lifespans [47][48].

4. Proposed SSDLC Framework for the New Space Industry

As the New Space industry evolves, with the expansion of commercial satellite constellations, scientific exploration, and military space operations, security must be a foundational principle in software development. A modern SSDLC framework tailored to space applications must align with global security frameworks such as

OWASP SAMM, SLSA, NIST CSF, security standards such as ISO/IEC 27001 and established space standards from organizations like NASA and ESA.

This framework proposes a holistic, proactive approach that ensures all mission stakeholders—developers, engineers, security teams, and operators—are actively involved in security processes. The core philosophy is that security is everyone's responsibility, requiring a cross-team security culture, continuous training, and up-to-date threat awareness.

4.1 Key Principles of the Proposed SSDLC Framework for Space Systems

- **Security Culture & Proactive Security Mindset**
  - Security must be proactive, not reactive—preventing risks before they materialize rather than responding to incidents after they occur.
  - Specialized security roles (e.g., Cyber Security Manager, Security Auditors, Risk Analysts) should be established to provide continuous guidance, enforce best practices, and ensure secure software development across all teams.
  - A mission-wide security culture ensures that all stakeholders—from developers to operators—are engaged, trained, and responsible for maintaining a strong security posture.
  - Developers should be trained in secure coding practices and guided by security specialists to apply automated security testing tools effectively.

- **Shift-Left Security: Embedding Security from the Start**
  - Security must be integrated from the earliest phases of development, rather than being treated as an afterthought.
  - Threat modeling and risk assessments should be conducted at the design stage, ensuring vulnerabilities are identified before code is written.
  - Security & Resiliency by Design:
    - Incorporate security analysis and requirement from the beginning and keep trace of compliancy through the project development.
    - Design built-in redundancies and failover mechanisms to ensure operational continuity in case of cyberattacks.
    - Require hardened firmware and boot security to prevent unauthorized modifications.
    - Define strategies for automated backups of scientific data with tamper-proof logging.
    - Promote long-term software sustainability practices, ensuring compatibility with future mission upgrades.

- **Secure Development Environments.** In aerospace and mission-critical software development, development environments are rarely updated, especially in operational settings where stability is prioritized over frequent updates. However, these environments must be maintained as vulnerability-free as possible to prevent security risks while ensuring compatibility with legacy systems.
  - Development environments should be hardened, following least privilege principles and enforcing secure configurations. Development environments must be protected against unauthorized access and supply chain threats.
  - Operating system versions used in development should be monitored for newly discovered vulnerabilities, and mitigation strategies should be applied, even if full updates are not feasible.
  - Developers must follow secure procedures when committing, retrieving, and downloading code from repositories, ensuring that: Version control systems enforce multi-factor authentication (MFA) and role-based access control (RBAC). All commits are signed with cryptographic signatures to prevent unauthorized modifications. Pull requests and merges undergo security scanning before approval. All external downloads (dependencies, frameworks) are verified for integrity using cryptographic hashes.
  - Monitoring within the organization should be established to detect and alert on anomalous developer behaviors, such as: Unusual access to repositories or code changes from untrusted locations. Unexpected administrative actions on version control or CI/CD systems. Attempts to introduce unapproved third-party components into the codebase.
  - Continuous security monitoring of development environments should detect and mitigate supply chain attacks before they propagate into production. This may include AI-assisted anomaly detection to flag unauthorized or suspicious activities.
  - Regular security audits of development environments should be conducted, focusing on:
    - Network segmentation and access controls for developer workstations.
    - Security patch management tailored for long-term stable environments.

  - Endpoint security monitoring to detect potential malware or unauthorized modifications.
  - Strict logging and traceability of all development-related actions.
- Secrets management best practices should be implemented to prevent unauthorized access to API keys, cryptographic credentials, and sensitive configurations
- Cryptographic protection for software updates to ensure authenticity and integrity.
- Versioning and integrity verification for datasets and software updates.
- CI/CD pipelines with security validation before production deployment.
- Resilient cloud deployments with automated failover and recovery.

➢ **Security evaluation and status tracking**

- Security evaluation is a fundamental for ensuring that software, infrastructure, and configurations remain resilient against cyber threats throughout their lifecycle. By performing comprehensive security assessments, organizations can obtain a security status snapshot at a given moment, allowing them to assess risk, prioritize remediation efforts, and maintain continuous security assurance across development and operational environments.
- Regular security evaluations should be conducted. Security is a process not a one-shot activity.
- Code-Level Security Assessments *(Shift-Left Security): SAST and SCA analysis*
  - Static Application Security Testing (SAST): Detect vulnerabilities in source code.
  - Software Composition Analysis (SCA): Identify security flaws and compliance risks in third-party dependencies before integration, enforcing supply chain security.
- Deployment & Runtime Security Evaluations
  - Dynamic Application Security Testing (DAST): Simulates real-world attacks against running applications to identify vulnerabilities that only manifest during execution.
  - Penetration Testing (Pentest): Conducts controlled cyberattacks to evaluate the effectiveness of security measures and uncover deep-seated vulnerabilities that automated tools might miss.
  - Configuration & Infrastructure Security Audits: Validate access controls, authentication mechanisms, encryption policies, and infrastructure configurations.
- Software Supply Chain Security - Enforcing SBOM (Software Bill of Materials) ensures transparency and accountability in software dependencies.
- Regular comparison of security assessments helps organizations track security evolution, detect deviations, and implement continuous improvements.
- Security Status Evolution Analysis: Organizations should compare historical security snapshots to identify trends, recurring vulnerabilities, and long-term improvements in security posture.
- Dedicated security roles are highly recommended for enforcing security controls, managing third-party risks, and maintaining mission-critical software integrity. Some of these roles are:
  - Risk Manager that conducts risk analysis to identify and assess security threats, maintains a risk registry, tracking potential vulnerabilities and their mitigation strategies, translates identified risks into security requirements, ensuring alignment with regulatory standards, develops and maintains the Security Concept, defining key security controls and objectives and evaluates the effectiveness of implemented security controls to ensure compliance.
  - Cyber Security Manager that implements cybersecurity requirements and strategic security plans for the organization, monitors the performance and effectiveness of security measures across all environments, tracks vulnerabilities and coordinates mitigation strategies, ensuring timely responses, reports security progress, identified risks, and implemented measures to key stakeholders and provides guidance and support in the implementation of security measures within development teams.
  - Cyber Internal Auditor that conducts security audits focusing on Business Continuity & Disaster Recovery Plans, System & Infrastructure Configurations, Vulnerability Assessments, Information Security Policies & Compliance. And performs acceptance audits to verify the effectiveness of mitigation strategies and compliance of security documentation and records
  - Third-Party Risk Analysts & Secure Software Supply Chain Managers that oversee third-party risk management, ensuring only rigorously vetted, secure, and compliant software is integrated into mission-critical applications, validate software provenance, enforce Software Bill of Materials (SBOM) tracking, and monitor third-party security compliance and implement automated security validation tools, such as Sonatype Nexus Firewall, to prevent insecure dependencies from entering the development pipeline.
- NOTE: The NIST Recommended Minimum Standard for Vendor or Developer Verification of Software under Executive Order 14028 provides essential security measures to ensure secure software development and supply chain integrity [52][53][54]. This can be taken as baseline to start with SSDLC practices.

- **Security Automation**
  - Automated security testing should be embedded within CI/CD pipelines, enforcing for example SAST and SCA to detect code vulnerabilities and insecure dependencies before deployment. In cases where automation is not possible, manual enforcement of security validation must be mandatory prior to release.
  - Comprehensive Testing Strategies: Security cannot be guaranteed without extensive testing across multiple dimensions:
    - Security Testing: Validate the effectiveness of security controls through penetration testing, fuzz testing, API security testing, and infrastructure security assessments.
    - Performance Testing: Ensure software can handle expected loads and stress conditions without degrading security mechanisms.
    - Resiliency Testing: Simulate failures, attacks, and unexpected conditions to verify system recovery capabilities.
    - Continuity Testing: Test disaster recovery, failover strategies, and backup mechanisms to confirm that systems remain secure and functional under adverse conditions.
    - Software Supply Chain Testing: Validate software provenance, verify SBOM (Software Bill of Materials), and enforce cryptographic signing policies to prevent unauthorized modifications.
  - Strict third-party evaluation and approval processes must be enforced before integrating external software components. Only pre-vetted, secure, and compliant dependencies should be included in development projects.
  - Admission policies for third-party libraries, open-source software, and COTS solutions must be automated and continuously enforced, ensuring that external components meet security compliance standards before integration.
  - Continuous monitoring with proactive analysis of dependencies and external components should be established, as software dependencies may become vulnerable over time due to newly discovered threats.
  - Automated security scanning tools must assess third-party dependencies in real-time, identifying and mitigating vulnerabilities before components enter the software ecosystem.
  - Additionally, automated reviews of secret storage policies should be conducted to ensure compliance and eliminate misconfigurations.
  - Automated vulnerability detection and notification mechanisms should be in place to promptly identify risks and alert security teams as soon as vulnerabilities emerge, enabling timely assessment and mitigation actions.
  - Generative AI & Security Automation: With the rise of AI-generated code, automated security validation must extend to AI-assisted development, ensuring that AI-generated software is reviewed, scanned, and approved using the same security rigor as human-written code:
    - AI-Generated Code Security: With the increasing use of Generative AI for code development, accountability becomes a diffuse responsibility, requiring strict validation and review processes for AI-generated code.
    - Automated Threat Modeling: AI-driven threat modeling improves security risk assessments by dynamically identifying potential vulnerabilities in design and architecture.
    - Adaptive Security Governance: AI-based security tools can continuously update security policies and compliance frameworks, ensuring that evolving threats are addressed dynamically.
    - AI-Assisted Admission Policies: AI-powered admission control mechanisms must evaluate third-party software, libraries, and commercial off-the-shelf (COTS) solutions before integration into mission-critical systems.

- **Continuous Security Risk Assessment and vulnerability management**
  - Risk assessments should be iterative, spanning design, implementation, and operational phases to ensure evolving threats are addressed continuously. Given the extended lifecycles of these systems, it is crucial to adapt to changing threats, re-evaluate the risk management approach, and proactively adjust it when substantial changes arise in system weaknesses, the likelihood of threats materializing, or the severity of potential consequences.
  - Vulnerability management must be ongoing, with dedicated security personnel ensuring that risk assessments, patches, and mitigations are consistently applied throughout the mission lifecycle.
  - AI-driven security analytics should be leveraged to assess third-party dependencies, predict vulnerabilities, and enforce adaptive risk mitigation strategies.
  - Automated monitoring of AI-generated code and security logs should be used to detect security drift, anomalies, and potential vulnerabilities introduced by automated processes.

- **Secure code development.** Apply best secure development principles such as:
  1. Separation of Responsibilities
     - Strict compartmentalization of subsystems to minimize attack vectors.
     - Role-based access control (RBAC) to limit access to classified systems.
     - Independent execution environments to isolate sensitive data.
     - Separation between core mission-critical functions and research tools to prevent unintended modifications.
     - Fine-grained access control for shared scientific data to ensure collaboration while maintaining data integrity.
     - Clearly defined ownership and accountability for software and data security in long-term projects.
     - Microservices-based architectures to separate business logic from operational controls.
     - Automated security enforcement in CI/CD pipelines for third-party integrations.
     - API security enforcement using and RBAC policies to limit excessive permissions.

  2. Keep It Simple, Stupid & Secure (KISS Principle)
     - Minimalist software design to reduce attack surfaces in embedded satellite systems.
     - No unnecessary software libraries to mitigate risks from unknown dependencies.
     - Limited communication interfaces to prevent unauthorized access.
     - Avoid overengineering while ensuring data validation and protection.
     - Use well-documented open-source libraries that follow secure development practices.
     - Ensure data integrity through cryptographic hashing to prevent accidental or malicious alterations.
     - Reduce system complexity to accelerate time-to-market, balancing security and cost efficiency.
     - Leverage cloud-native security features to simplify secure deployments.
     - Automate security compliance checks to avoid manual errors while maintaining agility.

  3. Minimize Attack Surface
     - Air-gapped systems when feasible to isolate sensitive mission controls.
     - Strict network segmentation to prevent lateral movement in case of a breach.
     - Redundant security mechanisms to handle electronic warfare threats.
     - Controlled access to mission control systems to prevent unintentional changes.
     - Log all access and modifications to scientific data for traceability.
     - Data encryption at rest and in transit using industry standards.
     - Data encryption using AES-256 for storage and TLS 1.3 for transmission.
     - Cloud-based micro segmentation to protect APIs and services.
     - Continuous security scanning of all internet-facing services.
     - Endpoint detection and response (EDR) tools for early threat detection.

  4. Zero Trust Architecture (ZTA)
     - All requests must be authenticated and authorized—no implicit trust.
     - Multi-factor authentication (MFA) for all access points.
     - Hardware security modules (HSMs) for cryptographic operations.
     - Strict user authentication for critical systems.
     - Role-based permissions with multi-level security clearance.
     - Audit logs for all system interactions.
     - Continuous authentication and identity verification for services.
     - Cloud-based zero-trust access policies.
     - Token-based access control.

  5. Use of Open Standards & Secure APIs
     - Strictly controlled API access, with military-grade encryption.
     - Custom, hardened security implementations following classified security protocols.
     - Minimal reliance on commercial APIs to reduce supply chain risks.
     - Use open-source APIs where security can be independently verified.
     - Standardized security mechanisms for data exchange between research teams.
     - APIs must enforce access controls and validate data integrity.
     - Use authentication and authorization technologies for secure authentication such OAuth 2.0, OpenID Connect, and JWT.
     - API security gateways to restrict unauthorized access.
     - Real-time monitoring of API interactions for anomaly detection.

- **Resilience Engineering**
  - Enable failover mechanisms for ground stations and satellite communications.
  - Ensure redundant security measures protect against technical failures and malicious disruptions.
  - Define and document a Continuity and Resilience Plan that includes Disaster Recovery Procedures (DRP) and Incident Response Procedures (IRP):
    - The Continuity Plan should define and categorize the critical systems that need resilience strategies based on mission impact, along with the continuity requirements that these systems need.
    - Disaster Recovery Procedures should be defined for each of the critical system identified and they should include the specific technical recovery and/or restoration strategy (i.e.: failover, redundancy, etc.) along with detailed steps that are needed in order to restore the system's functionality.
    - Incident Response Procedures must detail specific actions beyond technical procedures such as DRPs, providing a structured framework for effectively managing all incident types.
  - Continually test and validate both the technical failover mechanisms implemented as well as the documented procedures (DRPs and IRPs) to ensure relevance.

- **Tracking Known Vulnerabilities & Lessons Learned**
  - Maintain a centralized vulnerability database to track known security flaws across all mission software.
  - Conduct regular vulnerability scans and penetration testing to identify new risks.
  - Lessons learned from past security incidents should be documented and shared across projects.
  - Use standards such CVSS and CWE classifications to prioritize vulnerability mitigation.
  - Automate security rule updates in SAST, DAST, and SCA tools to ensure that newly discovered vulnerabilities and evolving threat patterns are continuously integrated into security testing.
  - Implement AI-driven security monitoring to dynamically update vulnerability detection models and improve predictive security analysis.
  - Establish security governance policies to ensure that all teams apply updated security rules and compliance requirements consistently across all development environments.

- **Managing Legacy Code & Staying Up to Date**
  - Regularly refactor and modernize critical software components to eliminate legacy security risks.
  - Apply security patches in a controlled manner, following strict validation processes.
  - Adopt infrastructure as code (IaC) practices to ensure configurations remain secure and reproducible.
  - Leverage machine learning-based security monitoring to detect vulnerabilities in legacy software.

4.2 Security Improvement vs. Implementation Complexity Quadrant

To effectively evaluate the impact of applying SSDLC security measures, it is crucial to analyze the balance between security improvement and implementation complexity. Some security principles offer high-impact improvements but require significant effort to implement, while others provide incremental security enhancements with lower integration complexity.

The following quadrants categorize recommended security measures based on:

- Security Impact: the degree to which the measure improves overall system security, resilience, and risk mitigation.
- Implementation Difficulty: the level of effort, resources, and complexity required to integrate the measure into the software lifecycle.

We have also considered the level of effort required to implement these measures in legacy systems versus new systems. Security practices are typically easier to integrate into newer systems where they can be built in from the start. In contrast, implementing security improvements in legacy environments often involves significant refactoring, compatibility challenges, or workarounds, which can increase complexity.

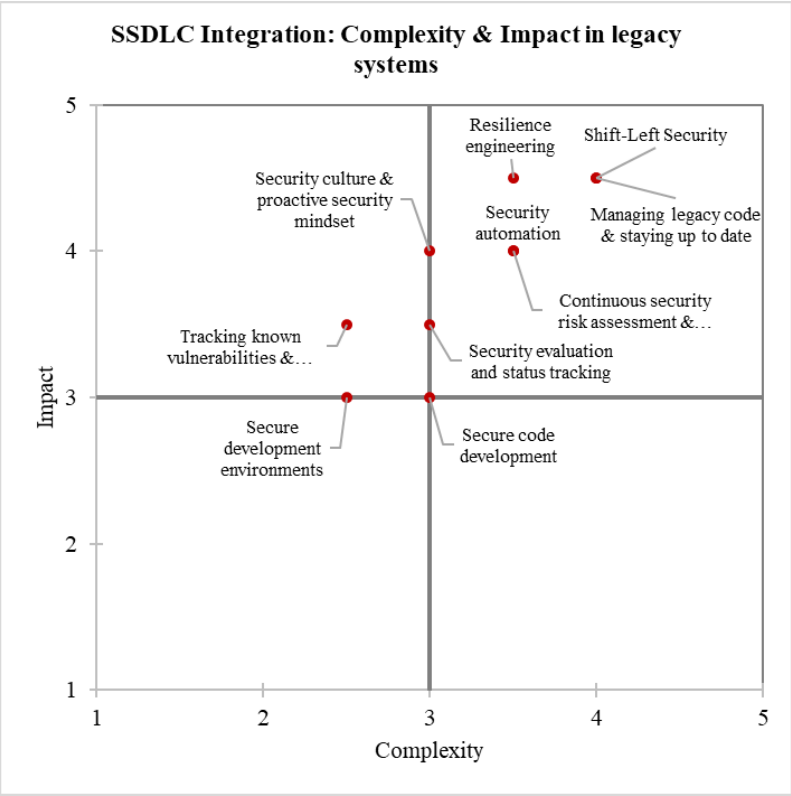


Fig. 1 SSDLC Integration: Complexity & Impact in legacy systems

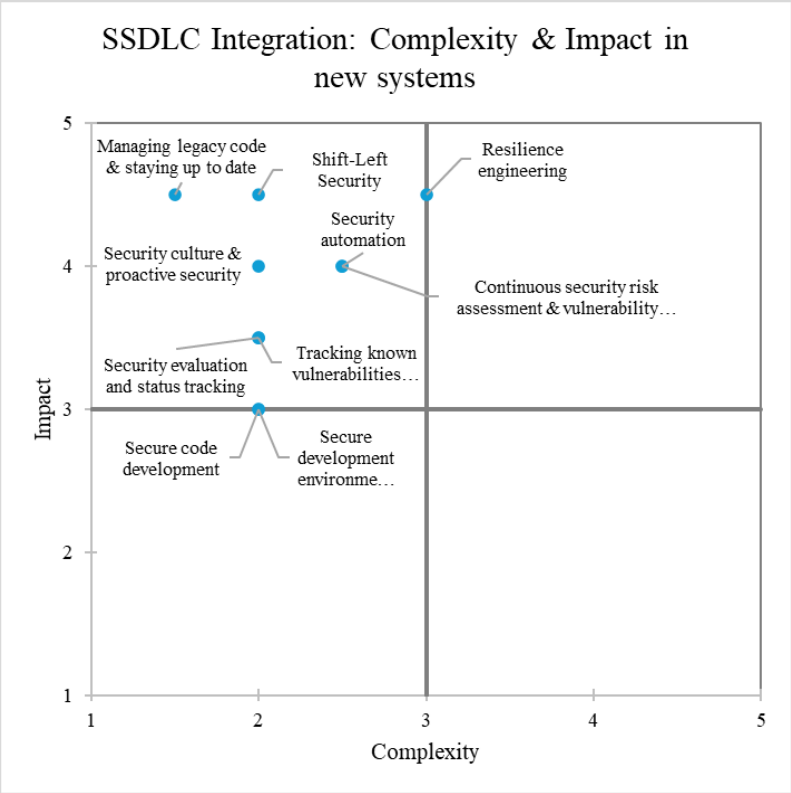


Fig. 2 SSDLC Integration: Complexity & Impact in new systems

5. Results and Discussion

The implementation of modern SSDLC methodologies in aerospace and space systems has led to significant improvements in software security, resilience, and efficiency. This section presents key findings from real-world

case studies, the quantifiable benefits of automation, and a comparative analysis of traditional vs. modern SSDLC approaches in the aerospace sector.

5.1 Key Findings from Case Studies

Several aerospace organizations and commercial satellite providers have begun integrating automated SSDLC frameworks and advanced practices, yielding measurable improvements in security, operational efficiency, and risk mitigation:

Case Study: Boeing's Implementation of Digital Twin Technology

Boeing's adoption of digital twin technology has resulted in significant improvements in manufacturing processes:

- Enhanced First-Time Quality: The use of digital twins led to a 40% improvement in the first-time quality of parts and systems used in both commercial and military aircraft manufacturing. [56]
- Increased Efficiency: The digital twin approach contributed to an 80% reduction in assembly hours and a 50% decrease in software development time for the T-7A aircraft development, enabling the aircraft to progress from design to first flight within 36 months. [51]

Case Study: NIST application security and software vulnerabilities reduction

The National Institute of Standards and Technology (NIST) has produced multiple studies and reports emphasizing the importance of early security integration, cost reduction, and effective vulnerability mitigation in software development. These insights are directly applicable to aerospace projects, where mission-critical software requires rigorous security controls, proactive risk management, and compliance with cybersecurity regulations.

Two key NIST resources provide valuable guidance on secure software development and cost-effective security implementation [55]. Key Highlights from these studies are:

- Definition of Application Security: Measures taken throughout the application's lifecycle to prevent exceptions in the security policy of an application or the underlying system through flaws in design, development, deployment, upgrade, or maintenance.
- Cost Implications: The presentation emphasizes that incorporating security during the initial stages of software development ("baking it in") is more cost-effective than addressing security issues after deployment ("bolting it on").

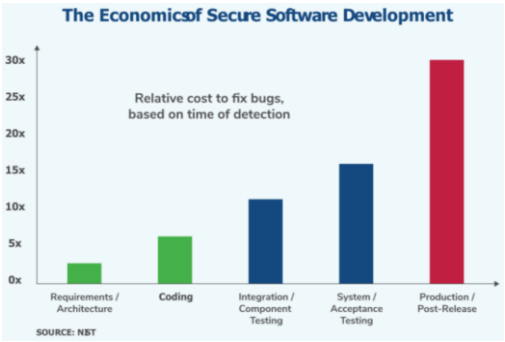


Fig. 3 NIST: Economics of Secure Software Development

- A defect identified during the requirements phase might have a relative cost of 1X to fix, whereas if the same defect is discovered during the integration and component testing phase, the cost could escalate to 10 times more. The effort required to identify and fix defects increases exponentially as software progresses through the five broad phases of development.
- Roles and Responsibilities: It highlights the importance of the development team in ensuring application security, suggesting that primary responsibility lies with them during the early phases.
- Training and Awareness: Providing application security training to development and testing teams is recommended to reduce security defects and enhance overall system security.

Summary of Benefits of Automation and Advanced SSDLC Methodologies

The transition from traditional to modern, automated SSDLC practices has yielded measurable benefits:

Table 1. Benefits of automation and advanced SSDLC methodologies

SSDLC Enhancement	Traditional Approach	Modern Approach (Automated SSDLC)	Observed Benefit
Security Testing	Manual penetration testing, late-stage validation	Automated SAST/DAST in CI/CD, AI-assisted threat modeling	Faster vulnerability detection and mitigation
Code Review	Developer-driven, occasional audits	AI-driven code review, automated security scanning	Reduced human errors and increased security compliance
Third-Party Dependency Management	Limited vetting, static vendor evaluation	Continuous monitoring, SBOM analysis, AI-powered risk assessment	Lower risk of supply chain attacks
Incident Response & Recovery	Reactive response to breaches	AI-based anomaly detection, automated remediation	Faster recovery, reduced system downtime
Secure Code Generation	Developer-driven, inconsistent practices	AI-assisted secure coding with policy enforcement	Improved consistency in secure coding practices

5.2 Comparison of Traditional vs. Modern SSDLC Approaches in Aerospace

Traditional SSDLC models in aerospace relied on manual validation processes, long development cycles, and limited real-time security monitoring. While these methodologies provided stability, they lacked adaptability to evolving cyber threats. The shift from traditional to modern SSDLC methodologies in the aerospace sector has led to:

- Reduction in Security Incidents: Projects incorporating automated security validation observed a significant decrease in critical vulnerabilities in production software.
- Accelerated Deployment Cycles: Automated SSDLC processes have expedited time-to-market, facilitating faster feature rollouts without compromising security.
- Cost Savings: Early-stage security integration has reduced the need for post-deployment security fixes, leading to substantial decreases in security-related maintenance costs.

Table 2. Comparison of Traditional vs. Modern SSDLC Approaches in Aerospace

Aspect	Traditional SSDLC	Modern SSDLC (Automated & AI-Driven)
Security Integration	End-of-cycle testing	Continuous security validation
Development Cycle Length	Long due to manual security audits	Shortened with automated compliance checks
Threat Detection	Static analysis, periodic audits	AI-driven behavioral analysis, real-time anomaly detection
Supply Chain Security	Vendor-based trust	Continuous dependency analysis, SBOM tracking
Incident Response	Reactive security fixes	Proactive threat mitigation, self-healing architectures

By transitioning to automated, AI-assisted SSDLC models, the aerospace industry can reduce attack surfaces, enhance resilience, and maintain secure, mission-critical software at scale.

6. Conclusions

The Secure Software Development Life Cycle (SSDLC) is an essential framework for ensuring the security, reliability, and resilience of space industry software. As space systems become increasingly software-dependent, the integration of security at every stage of development is not optional but imperative. This paper has highlighted key best practices, real-world implementations, and emerging trends that reinforce the need for structured SSDLC methodologies in aerospace applications.

6.1 Key Takeaways

➢ Cybersecurity in Space is a Mission-Critical Requirement
- The aerospace industry faces unique security challenges, including state-sponsored cyber threats, supply chain vulnerabilities, and real-time operational constraints.
- SSDLC practices ensure secure-by-design principles, reducing risks before software is deployed into mission-critical environments.

- Regulatory & Compliance Standards Shape SSDLC
  - Space agencies like NASA, ESA, and FAA, along with cybersecurity bodies such as NIST and CISA, have established comprehensive frameworks to secure aerospace software and infrastructure.
  - Implementing SSDLC aligned with NIST 800-218, ECSS-Q-ST-80C, and NASA SEPR ensures compliance with global security regulations.

- Automation & AI-Driven Security Are Transforming SSDLC
  - The shift from manual security validation to automated SSDLC processes has improved vulnerability detection, response times, and compliance tracking.
  - AI-assisted security testing, SBOM enforcement, and automated penetration testing have become essential tools in securing aerospace software.

6.2 The Need for Continuous Evolution in Aerospace SSDLC

As the aerospace sector advances towards autonomous systems, commercial space travel, and deep-space missions, traditional SSDLC approaches must evolve to address new cybersecurity challenges:

- AI-Driven Security Enhancements
  - AI-powered threat modeling & anomaly detection will enable proactive cyber defense strategies.
  - Machine learning-based security validation will improve code development, analysis, automated penetration testing, and risk assessment.

- Blockchain for Software Integrity & Supply Chain Security
  - Decentralized identity verification can enhance software authentication & update integrity.
  - Smart contracts can be used to enforce security policies in real-time across distributed space networks.

- Automation & Self-Healing Systems
  - Automated remediation mechanisms will enable self-repairing software architectures, ensuring mission continuity in the event of cyberattacks.
  - Zero-trust security models & adaptive security frameworks will be crucial for securing interconnected space assets.

References

[1] GMV. (2023). GMV leads the consortium that will create a federated network to speed up the use of AI in healthcare. Retrieved from https://www.gmv.com/en-es/communication/news/gmv-leads-consortium-will-create-federated-network-speed-use-ai-healthcare
[2] GMV. (2023). Utile: Cybersecurity platform for the healthcare sector. Retrieved from https://www.gmv.com/en-es/products/cybersecurity/utile
[3] GMV. (2023). GMV presents the success story of Utile at the 1st Andalusian Artificial Intelligence Congress. Retrieved from https://www.gmv.com/en-es/communication/news/gmv-presents-success-story-utile-1st-andalusian-artificial-intelligence-congress
[4] European Union Agency for the Space Programme (EUSPA). (n.d.). About the EU Space Security Accreditation Board (SAB). Retrieved from https://www.euspa.europa.eu/about/about-eu-space-sab
[5] NASA. (n.d.). *Software Engineering Procedural Requirements, Standards, and Related Resources*. Retrieved from https://www.nasa.gov/intelligent-systems-division/software-management-office/nasa-software-engineering-procedural-requirements-standards-and-related-resources/
[6] European Space Policy Institute (ESPI). (n.d.). *Integrating Commercial Space for Military Applications in Europe: A Challenge and Opportunity*. Retrieved from https://www.espi.or.at/briefs/integrating-commercial-space-for-military-applications-in-europe-a-challenge-and-opportunity
[7] The Aerospace Corporation. (2018). *Partnerships: A Guide to Public-Private Collaboration in Space*. Retrieved from https://aerospace.org/sites/default/files/2018-06/Partnerships_Rev_5-4-18.pdf
[8] GMV. (2023). *GMV attends CYSAT 2023 to present cybersecurity solutions for space systems*. Retrieved from https://www.gmv.com/en-es/communication/news/gmv-attends-cysat-2023
[9] GMV. (n.d.). *Checker: ATM security solution*. Retrieved from https://www.gmv.com/en-es/products/cybersecurity/checker-atm-securityr
[10] European Union. (2013). *Decision No. 2013/488/EU on the security rules for protecting EU classified information*. Retrieved from https://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX:32013D0488

[11] European Cooperation for Space Standardization (ECSS). (n.d.). *ECSS: Standards for Space Engineering & Product Assurance*. Retrieved from https://ecss.nl/
[12] European Cooperation for Space Standardization (ECSS). (2022). *ECSS-E-ST-40C Rev.1: Space Engineering – Software (Public Review)*. Retrieved from https://ecss.nl/standard/ecss-e-st-40c-rev-1-dir1-space-engineering-software-public-review-23-august-18-october-2022/
[13] European Cooperation for Space Standardization (ECSS). (2017). *ECSS-Q-ST-80C Rev.1: Software Product Assurance*. Retrieved from https://ecss.nl/standard/ecss-q-st-80c-rev-1-software-product-assurance-15-february-2017/
[14] European Cooperation for Space Standardization (ECSS). (n.d.). *ECSS-E-ST-80C: Space Engineering – Security in Space Systems Lifecycles*. Retrieved from https://ecss.nl/standard/ecss-e-st-80c-space-engineering-security-in-space-systems-lifecycles/
[15] BankInfoSecurity. (2023). *NASA Releases First Space Cybersecurity Best Practices Guide*. Retrieved from https://www.bankinfosecurity.com/nasa-releases-first-space-cybersecurity-best-practices-guide-a-23967
[16] Surapaneni, M. (2023). *Studying NASA's SDLC & Software Engineering Handbook*. Retrieved from https://www.linkedin.com/pulse/studying-nasas-sdlc-software-engineering-handbook-manish-surapaneni-a1dmc/
[17] RTCA, Inc. (2011). *DO-178C: Software Considerations in Airborne Systems and Equipment Certification*. Retrieved from https://www.do178.org/
[18] National Institute of Standards and Technology (NIST). (2021). *NIST IR 8401: Satellite Ground Segment, Applying the Cybersecurity Framework to Satellite Command and Control*. Retrieved from https://csrc.nist.gov/publications/detail/nistir/8401/final
[19] European Space Agency (ESA). (n.d.). *ESA Cybersecurity Framework*. Retrieved from https://www.esa.int/About_Us/Security/ESA_Cybersecurity
[20] Federal Aviation Administration (FAA). (2023). *FAA Space Cybersecurity Guidelines & Best Practices*. Retrieved from https://www.faa.gov/space
[21] GMV. (n.d.). *GMV Penbot: AI-driven automated pentesting application for web vulnerability detection*. Retrieved from https://www.gmv.com/es-es/productos/ciberseguridad/gmv-penbot
[22] GMV. (n.d.). *Gestvul: Advanced Vulnerability Management Platform*. Retrieved from https://www.gmv.com/es-es/productos/ciberseguridad/gestvul
[23] Panarotto, M., Isaksson, O., & Vial, V. (2023). Cost-efficient digital twins for design space exploration: A modular platform approach. _Computers in industry_, _145_, 103813.
[24] Satyarthi, S., Pandey, D., & Khan, M. W. (2021). Adaptation of digital twins as a methodology for management and development of secure software systems. NeuroQuantology, 19(7), 300-309.
[25] Smith, D., Stevens, R., & Becklund, D. (2023). *Digital twins lifecycle applications*. The Aerospace Corporation. Retrieved March 3, 2025, from http://sef.aerospace.org/files/2023/06/OTR-2023-00822-Digital-Twins-Lifecycle-Applications.pdf
[26] Liu, W., Wu, M., Wan, G., & Xu, M. (2024). Digital Twin of Space Environment: Development, Challenges, Applications, and Future Outlook. *Remote Sensing*, *16*(16), 3023. https://doi.org/10.3390/rs16163023
[27] Marksteiner, S., Bronfman, S., Wolf, M., & Lazebnik, E. (2021, September). Using cyber digital twins for automated automotive cybersecurity testing. In *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* (pp. 123-128). IEEE.
[28] Grieves, M. (2023). Digital Twin Certified: Employing Virtual Testing of Digital Twins in Manufacturing to Ensure Quality Products. *Machines*, *11*(8), 808. https://doi.org/10.3390/machines11080808
[29] Starlink. (n.d.). *Starlink Security and Compliance Standards*. Retrieved from https://www.starlink.com/support/article/984d0c92-9fac-3036-1138-be0f390829dc
[30] Starlink**.** (n.d.). *Starlink Security Framework*. Retrieved from https://www.starlinkme.net/security-framework
[31] Starlink**.** (2023). *Starlink Welcomes Security Researchers: Bring on the Bugs*. Retrieved from https://www.starlink.com/public-files/StarlinkWelcomesSecurityResearchersBringOnTheBugs.pdf
[32] Starlink**.** (n.d.). *Bug Bounty & External Security Testing Program*. Retrieved from https://www.starlink.com/security-research
[33] Stack Overflow. (2021). *Building a Space-Based ISP: The Software Behind Starlink*. Retrieved from https://stackoverflow.blog/2021/05/11/building-a-space-based-isp/
[34] Kareem, K. M. (2024). *Cyber Threat Landscape Analysis for Starlink: Assessing Risks and Mitigation Strategies in the Global Satellite Internet Infrastructure*. https://arxiv.org/pdf/2406.07562
[35] U.S. Government Accountability Office (GAO). (2024). *NASA Needs to Fully Integrate Cybersecurity into Spacecraft Acquisition Policies and Standards*. Retrieved from https://www.gao.gov/assets/gao-24-106624.pdf

[36] NASA Office of Inspector General (OIG). (2024). *NASA's Artemis Program Management Challenges and Cybersecurity Risks*. Retrieved from https://oig.nasa.gov/wp-content/uploads/2024/02/IG-22-003.pdf
[37] OODA Loop. (2024). *The Cyber Threat to NASA's Artemis Program*. Retrieved from https://oodaloop.com/analysis/ooda-original/the-cyber-threat-to-nasas-artemis-program/
[38] NASA. (2023). *NASA-STD-8739.8: Software Assurance and Software Safety Standards*. Retrieved from https://standards.nasa.gov/sites/default/files/standards/NASA/B/0/NASA-STD-87398-RevisionB.pdf
[39] [4] NASA. (2023). *Adaptive IV&V Approaches for Large-Scale Space Missions*. Retrieved from https://ntrs.nasa.gov/api/citations/20230005123/downloads/12th%20IAASS%20Conference%20Paper%20IVV%20final3.pdf
[40] [5] NASA. (2023). *Risk-Based Verification & Validation Strategies in Artemis Missions*. Retrieved from https://ntrs.nasa.gov/api/citations/20230007162/downloads/IAASSS%20Conf%20Adaptive%20IV%26V.pdf
[41] NASA. (n.d.). *Secure Software Development Self-Attestation Resources and Knowledge*. Retrieved from https://www.nasa.gov/secure-software-development-self-attestation-resources-and-knowledge/
[42] Cybersecurity and Infrastructure Security Agency (CISA). (n.d.). *Secure by Design*. Retrieved from https://www.cisa.gov/securebydesign
[43] Burch, R. (2019). *Resilient space systems design: an introduction*. CRC Press.
[44] Sawik, B. (2023). Space mission risk, sustainability and supply chain: review, multi-objective optimization model and practical approach. Sustainability, 15(14), 11002.
[45] Kambala, G. (2024). Intelligent Fault Detection and Self-Healing Architectures in Distributed Software Systems for Mission-Critical Applications. *International Journal of Scientific Research and Management (IJSRM)*, *12*(10), 1647-1657.
[46] Ibrahim, S. K., Ahmed, A., Zeidan, M. A. E., & Ziedan, I. E. (2020). Machine learning techniques for satellite fault diagnosis. *Ain Shams Engineering Journal*, *11*(1), 45-56.
[47] Hedayati, M., Barzegar, A., & Rahimi, A. (2024). Fault diagnosis and prognosis of satellites and unmanned aerial vehicles: A review. *Applied Sciences*, *14*(20), 9487.
[48] Zhou, M., Wang, Z., Theilliol, D., Shen, Y., & Rodrigues, M. (2016, September). A self-healing control method for satellite attitude tracking based on simultaneous fault estimation and control design. In *2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol)* (pp. 349-354). IEEE.
[49] Mercorelli, P. (2024). Recent advances in intelligent algorithms for fault detection and diagnosis. *Sensors*, *24*(8), 2656.
[50] Cybersecurity and Infrastructure Security Agency (CISA). (n.d.). *Software Bill of Materials (SBOM)*. Retrieved from https://www.cisa.gov/sbom
[51] Boreham, J. (2024, June 20). *The Performance of Digital Twins Across Industry*. Digital Twin Insider. Retrieved from https://digitaltwininsider.com/2024/06/20/the-performance-of-digital-twins-across-industry/
[52] National Institute of Standards and Technology (NIST). (n.d.). *Software Security in Supply Chains: Executive Order 14028 Implementation Overview*. Retrieved from https://www.nist.gov/itl/executive-order-14028-improving-nations-cybersecurity/software-security-supply-chains-0
[53] National Institute of Standards and Technology (NIST). (n.d.). *Enhanced Software Security in Supply Chains: Implementing Executive Order 14028 Requirements*. Retrieved from https://www.nist.gov/itl/executive-order-14028-improving-nations-cybersecurity/software-security-supply-chains-enhanced
[54] National Institute of Standards and Technology (NIST). (n.d.). *Recommended Minimum Standard for Vendor or Developer Verification of Software Under Executive Order 14028*. Retrieved from https://www.nist.gov/itl/executive-order-improving-nations-cybersecurity/recommended-minimum-standard-vendor-or-developer
[55] National Institute of Standards and Technology (NIST). (2011). *Application Security Costs Discussion*. Retrieved from https://csrc.nist.gov/CSRC/media//Projects/Forum/documents/2011/FCSM-041211-Application-Security-Costs-Discussion.pdf
[56] Hannover Messe. (n.d.). Boeing aims to virtualize the development of aircraft. Retrieved from https://www.hannovermesse.de/en/news/news-articles/boeing-aims-to-virtualize-the-development-of-aircraft