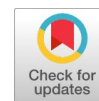


Integrating Agile and the Systems V-Model: Proposing a Hybrid Solution

Marc Berry, Venkata Allada



Abstract: This study introduces an Agile Hybrid model that combines the Agile and Systems V philosophies for product development initiatives. Agile frameworks thrive in environments that offer supportive iterative development, continuous integration, and incremental delivery. The current literature introduces existing Agile Hybrid models, but they are presented only at a conceptual level. This leaves practitioners without clear direction on how to implement an Agile Hybrid framework in a complex, real-world scenario. Based on research and practitioner experience, the findings show that the primary challenges are executive understanding and communication links, cultural resistance, management of hardware integration, integration methods, and backlog management. Executives have reported feeling that planning is incomplete when using Agile. This proposed Agile V model provides a practitioner-centric approach to improving these challenges. The models consist of two main components. First, an Agile V Overlay—an Agile/Systems V visual roadmap that aligns Sprints to Gates and ties executive leadership's interests to the development team's work evolution—can be used for both project planning and execution. Secondly, a Comprehensive Adaptive Backlog (CAB) is an artefact repository that contains all engineering- and programmatic-type artefacts typically required for new product development initiatives. The CAB ties work artefacts to Sprints and to the Systems V gate review. Combined, the Overlay and CAB provide tools to help Project Managers and Engineering Managers plan and execute complex product development projects in a highly regulated environment. The goal is to provide tools that enable design evolution while maintaining a systematic, gated approach that supports regulatory compliance. The authors have conducted initial interviews with industry practitioners, which have yielded positive results for the model's utility. Practitioner feedback has been incorporated to fine-tune the model and improve its alignment with real-world applications. Future research will adopt a mixed-methods approach, including interviews with industry practitioners to continue validating the Agile V model.

Keywords: Agile, Systems Engineering, Agile Hybrid, Systems Engineering V Model, Project Management, New Product Development.

Abbreviations

ADP: Acceptance Data Package

CAB: Comprehensive Adaptive Backlog
CAD: Computer-Aided Design
CDR: Critical Design Review
CONOPS: Concepts of Operations
EDU: Engineering Development Unit
LLPL: Long Lead Parts Procurement Start
MRR: Manufacturing Readiness Review
PDR: Preliminary Design Review
PIC: Program Integration Cycles
POD: Parts on Dock
PoP: Period of Performance
SRR: System Requirements Review
SVR: System Verification Review
TRR: Test Readiness Review
V&V: Verification and Validation
XP: Extreme Programming

I. INTRODUCTION

In today's product development environment, Project and Engineering managers face the challenge of shorter time-to-market and lower new-product costs. Basically, looking for products that are cheaper while maintaining the same level of quality excellence. Traditional project management methods, such as the Systems Engineering V-Model, provide a systematic structure, but often lack the flexibility to allow changing requirements [22]. This is particularly true when using an Agile framework with hardware-focused systems engineering projects, which can affect the traditional “pure Agile” approach. Systems engineering projects can benefit from Agile Hybrid methods, which combine Agile flexibility with structured approaches to achieve the same benefits as “pure Agile” methods in software design. The Agile V-Model helps address these concerns by integrating Agile and Systems Engineering V-Model principles to meet current research needs in new product development initiatives. The framework consists of two main elements: the Agile V Overlay, a product development lifecycle visual roadmap, and the Comprehensive Adaptive Backlog (CAB), an artefact repository that identifies and lists backlog requirements and aligns them with sprints and systems engineering V gate reviews. The Agile V Overlay in the model provides a high-level visual overview of the product development process, while the CAB serves as a planning and execution tool for sprints. The framework enables management of physical products, as its categorisation of hardware-centric deliverables—Digital Modelling, Additive Manufacturing, Engineering Development Units, Qualification Units, and Qualified Design Units—shows a natural progression of hardware development through the Agile sprinting process. The integration of sprints with

Manuscript received on 24 October 2025 | First Revised Manuscript received on 29 October 2025 | Second Revised Manuscript received on 05 November 2025 | Manuscript Accepted on 15 November 2025 | Manuscript published on 30 November 2025.

*Correspondence Author(s)

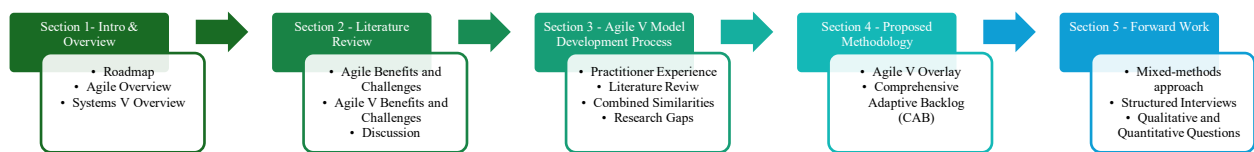
Marc Berry*, Department of Engineering Management and Systems Engineering, Missouri Science and Technology, Rolla, US Minor Outlying Islands. Email ID: mabhz5@umsystem.edu, ORCID ID: [0009-0002-8010-2471](https://orcid.org/0009-0002-8010-2471)

Dr. Venkata Allada, Professor, Department of Engineering Management and Systems Engineering, Missouri Science and Technology, Rolla (MO), United States of America (USA). Email ID: allada@mst.edu

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open-access article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Systems Engineering V gates and high-level engineering lifecycle phases enables executive leaders to track project performance as development teams execute. This solves the planning-incomplete issue that researchers and practitioners have identified and documented. Future research will validate the Agile V-Model through a mixed-methods approach that combines practitioner surveys with semi-structured interviews to assess quantitative effectiveness and qualitative insights into practitioner implementation. The dual evaluation method will provide an assessment of the model's usability and its potential for improvement across different industries and business sectors. This paper is structured as follows:

Section 1 provides an introduction and overview of Agile practices and the Systems Engineering V-Model to establish the study's context. Section 2 examines current research on the advantages and challenges of Agile and V-Model approaches. Section 3 describes the development process, while Section 4 outlines the proposed methodology for introducing the Agile V model. To conclude, section 5 presents proposed research paths for future work that the authors will conduct to evaluate model validation through practitioner interviews. Figure 1 below depicts the manuscript roadmap.



[Fig.1: Manuscript Roadmap]

II. AGILE OVERVIEW

The agile methodology is a project management approach that breaks a project into sprints and emphasizes continuous collaboration and improvement throughout the project lifecycle. Teams follow a cycle of planning, execution and evaluation [1]. Agility goes back to the manifesto of agile software development, developed by a consensus of 17 programmers on how to work more efficiently than traditional approaches under uncertain and dynamic conditions [3]. The manifesto consists of four values and 12 principles.

- The precedence of individuals and interactions over procedures and tools
- Functional software is preferred to lengthy documentation.
- Prioritizing client interaction over contract deinitialization
- Responding to change rather than adhering to a predetermined course of action

Agile approaches include Lean, Extreme Programming (XP), Scrum, and Kanban. Scrum is the preferred methodology because it works well with hardware and physical development [7]. Although agile comes in various forms, [7] suggests that Scrum is the most applicable to physical product development initiatives.

A. Agile-Scrum

[18] define Scrum as: Scrum is an easy method that helps people and teams generate value via changing solutions for complicated problems. In summary, the Scrum framework requires a Scrum Master to provide a working environment that:

- The work for a complex problem is ordered into a Product Backlog by a Product Owner.

- During a Sprint, the Scrum Team converts a portion of the labour into an increment of value.

- The Scrum Team and its stakeholders evaluate the outcomes and make necessary modifications for the subsequent Sprint.

- Reiterate

The Scrum framework is deliberately incomplete and straightforward, defining only the parts required to implement Scrum theory. Scrum is built on the collective intelligence of its users. Rather than providing detailed instructions, the Scrum rules guide people's relationships and interactions. Various processes, techniques, and methods can be employed within this framework. Scrum wraps around existing practices and renders them unnecessary, while making visible the relative efficacy of current management, environment, and work techniques so that improvements can be made [19]. The Scrum model, as described by [19], is iterative and includes three repetitive stages: product backlog development, main sprint, and daily sprints. Each iteration increases customer value [20].

B. Core Scrum Concepts

- Product Backlog*: a prioritized list of work for the project team to complete.
- Sprint Goal*: Leading toward project milestones and deliverables.
- Sprint Backlog*: A selected subset of the artefacts that need to be completed that align with the sprint goal.
- Sprint*: a short development period (usually one to four weeks) aiming to complete the (incomplete sentence/thought)
- Sprint*: Backlog includes agile events such as daily meetings, sprint planning, sprint demonstration, sprint review, and sprint retrospective.

- vi. *Daily Meeting*: Regular check-in to keep the team informed and on track to complete the sprint.
- vii. *Sprint Planning*: The first event of each sprint in which the team determines the sprint backlog, which is a subset of the product backlog.
- viii. *Sprint Demo and Review*: A sprint event in which the outcome of the sprint is first reviewed with stakeholders (demonstration) and against the sprint goal to help inform future work.
- ix. *Sprint Retrospective*: Final event of each sprint in which the team identifies ways to improve their agile practices.

Another recent source [13], states Agile represents an iterative and incremental method which delivers value through flexible teamwork and continuous delivery instead of following strict long-term plans. The early 2000s brought forth Agile as a response to traditional project management philosophies that were inflexible when handling change. The Agile Manifesto, published in 2001, established formal Agile principles by presenting four core values and twelve guiding principles for project management. The Agile Manifesto establishes its core values through four fundamental ideals as follows:

The Agile Manifesto prioritises individual team members and their interactions over all other organisational systems and technological resources. The Agile approach allows 1) higher importance to delivering operational software versus developing extensive documentation, 2) allows for team collaboration with customers instead of reliance on contract negotiations and finally, 3) supports flexibility and adaptation instead of maintaining strict project plans.

C. Systems Engineering V Overview

The Systems Engineering V-Model, as described by [14] is a framework that visually and conceptually outlines the systems engineering process in a unique "V" shape. This model encompasses aspects of system design evolution, from initial conception to final disposal [14]. states that the model incorporates key technical reviews, System Requirements Review (SRR), Preliminary Design Review (PDR), Critical Design Review (CDR), Test Readiness Review (TRR), Manufacturing Readiness Review (MRR), and System Verification Review (SVR), which are major milestones used to ensure system design meets system requirements. The left side of the "V" identifies the requirements definition phase, laying the foundation for the system development phase. The systems engineering V model is used to manage systems engineering projects through the following Milestones, also referred to as gates. The systems V model, as described by [9], establishes it as a fundamental component of the systems engineering philosophy.

Although there are many different V models with different nomenclature, the most common ones are as follows:

SRR - A formal system-level review held to ensure system requirements have been identified and that a mutual understanding and agreement with all stakeholders exists. SRR establishes the functional baseline [9].

PDR - A formal system-level review that evaluates preliminary design fidelity by the results of requirements of trade studies, prototyping, and critical technology demonstrations. PDR will establish the allocated baseline and confirm that the system under review is ready to proceed into the next phase of system development with allowable risk agreed to by all stakeholders [9].

CDR - A formal system-level review that evaluates technical artefacts to ensure that a system can proceed into the next phase of system development, which consists of manufacturing and testing, and meeting stated performance requirements within cost, schedule, and risk [9].

MRR - A formal system-level review that evaluates if a product's design and manufacturing plan are ready for production, which identifies and mitigates any risks before manufacturing. Significant areas of an MRR review include checking the design, production processes, personnel, facilities, and quality management systems to ensure the product can be built on time, within budget, and to the required quality and quantity [9].

TRR - A formal system-level review that evaluates if a system is ok to start formal system testing by assessing test objectives, methods, procedures, and safety requirements. It ensures that all resources, such as personnel, equipment, and documentation, are in place. Additionally, it assures all stakeholders that the system has sufficient design and planning to proceed to the next phase. A successful TRR confirms that planned tests align with program requirements and user needs [9].

SVR - A formal system-level review that evaluates the system-level performance of the product by reviewing the verification reports to determine if the configuration end item meets its item performance specifications as documented in the allocated baseline post SRR review [9].

D. Benefits and challenges of Agile, Traditional and Hybrid methodologies

Agile, Systems Engineering V-Model, and hybrid agile project management processes each have their own benefits and challenges. Based on the literature for each of these methods, an overview of the approaches, with their key benefits and challenges, is shown in Figure 2 below.

Integrating Agile and the Systems V-Model: Proposing a Hybrid Solution

Approach	Benefits	Challenges	Key Sources
Agile	-Increased team productivity and employee satisfaction -Agile minimizes redundant meetings, repetitive planning, excessive documentation, quality defects, and low-value product features -Faster and more predictable delivery of valuable products and features	-Executive misunderstanding of agile -Tension between agile teams and traditional organizational units -Traditional waterfall processes limit agile benefits -Organizational barriers and unclear authority	[17] Rigby, Sutherland, & Takeuchi (2016)
	-Adoption of Agile methods aims to improve project performance by promoting agility -Ability to Respond to Change -Potential for Broader Application	-Lack of a unique and clear definition of the agility construct -challenging to identify appropriate metrics to measure performance -Agile concepts and principles are rooted in various disciplines, which sometimes lead to inconsistent understanding	[4] Conforto et al. (2016)
	-higher success rates in terms of meeting project constraints -Agile facilitates greater flexibility and adaptability to changing requirements.	-Agile requires significant changes in team dynamics, roles, and organizational culture -Projects with fixed deadlines, budgets, or regulatory requirements may struggle to	[20] Serrador & Pinto (2015)
Systems Engineering V-Model	-V-Model-based Systems Engineering offers structured process rigor and traceability beneficial for complex system validation	-faces challenges in dealing with agile software practices, diverse supplier abstraction levels, dynamic product updates, and the complexity of cross-company collaboration needed in modern systems engineering	[16] Prostep ivip Association (2024)
	-provides a structured and rigorous approach -V-Model views help present complex system development more effectively aiding stakeholders' understanding -model emphasizes verification and validation activities that should be integrated early and throughout the development cycle, reducing risk and late-stage problems	-failed to explicitly incorporate iterations; -No single model, including the V-Model, can explain and correct all aspects of a diverse system development project due to its complexity; alongside the V-Model, other factors and tools must be considered -the V-Model alone does not ensure success without process rigor, metrics, and culture	[21] Sutherland et al. (2015)
	-emphasizes decomposition of the system into system, subsystem, and component levels -V&V activities are enforced at system, subsystem, and component levels -multi-level V&V enhances quality assurance and helps identify potential issues early -align well with collaboration challenges such as unclear responsibilities, lack of system-wide view, and quality assurance	-Applying the V-Model demands more effort than ad-hoc or less formalized approaches. This includes rigorous documentation, strict V&V processes, and clear responsibility tracking -Poor documentation or lack of consistent documentation causes collaboration challenges -Ensuring data quality is difficult, and experience shows multiple iterations are often needed	[23] Wu (2024)
Hybrid	-Time-boxed sprints create urgency, which speeds projects up -Early, cheap, physical or visual prototypes allow customer feedback -handles situations where requirements can't be fully defined upfront -Gates remain in place to review deliverables and make go/kill decisions	-Teams sometimes become too focused on immediate sprint goals and lose sight of the ultimate longer-term product objectives -leaders are initially skeptical about Agile and may resist changing -requires senior management to engage more frequently to maintain strategic direction	[6] Cooper (2016)
	-Allows for much more adaptive and flexible responses to changing product or customer requirements -Enhanced communication within the development team and across the organization -improves productivity by combining Agile micro-planning and iterative work cycles with the Stage-Gate structure	- There is a lack of larger sample studies with quantitative and operational performance metrics to fully validate the hybrid model's effectiveness -senior management in some firms expressed concern that teams focused heavily on short sprints lost sight of long-term goals -Senior management in traditional industries is often skeptical of Agile	[7] Cooper & Sommer (2016)
	-hybrid approach allowed firms to speed up development -Increased team morale and ability to handle uncertainties -Regular scrums with the entire project team present enhanced knowledge sharing, task syncing, and problem discussion	-Embedding Agile teams into classic organizational structures and modularizing physical products into increments fitting sprints proved difficult -dealing with "other tasks" interrupting sprint focus, and balancing existing bureaucratic gating processes with Agile flexibility	[12] Edwards et al. (2020)
	-Hybrid projects serve as an "interim solution" that allows companies to gradually shift towards more agile methods -enable companies to balance the dynamic needs of agile with the constraints and rigor of traditional processes,	-The "paradigm perplexity"—the clash or incompatibility between agile and classical mindsets -can create complexity in coordination, communication, and mindset of reconciling agile with classical approaches in physical product development	[2] Atzberger et al. (2019)
	-Increased flexibility in managing product development projects -Ability to capture many benefits seen in software Agile implementations -Helps companies address the complexity of physical product development which cannot always conform to pure Agile practices	-Pure Agile methods as applied in software don't directly fit physical product development -Difficult to redefine "done sprint" because a working physical product cannot typically be delivered every sprint -Overcoming down-management skepticism within organizations	[8] De Borja, Trabasso, & Pessoa (2019)

[Fig. 2: Agile Hybrid Benefits and Challenges]

III. METHODOLOGY

The Agile V model emerged from a combination of practitioner experience in product development, project management, and systems engineering, coupled with a comprehensive literature review that identified research gaps aligned with the authors' knowledge. Based on the literature, Agile methods have proven effective for software development, yet the strict, software-oriented Agile approach has been challenging to apply to hardware-based projects. The implementation of Agile principles as a standalone approach revealed significant deficiencies during the execution of established system engineering milestones, including the System Requirements Review (SRR), Preliminary Design Review (PDR), and Critical Design Review (CDR). To reconcile Agile implementation with Systems V requirements, a hybrid strategy was required. The Agile Systems V Model development process used a systematic approach that united field-practitioner knowledge with extensive research into deficiencies in project management frameworks. The development process consisted of four stages: documenting practitioner experiences, conducting a systematic literature review of Agile-Stage-Gate and Agile-V-Model combinations, and combining these findings to create the new Agile Systems V Model.

A. Practitioner Experience

The Agile V models' foundation was built on practitioner knowledge from projects that used both the Systems Engineering V-Model and Agile frameworks. The project management team gained specific expertise through their work with gate-controlled processes in controlled settings and their implementation of Agile techniques in software development projects. The team members showed intense interest in combining Agile's iterative methods with V-Model-structured verification steps to improve efficiency and reduce overall time to market.

B. Literature Overview on Hybrid Models

The literature review identified four Agile hybrid models that integrate Agile with traditional methods for product development management as summarized in the following table. The findings show that Agile Hybrid models provide benefits in terms of improved time-to-market metrics, team communication and cohesion, and flexibility. Conversely, it shows challenges with management support, integration guidance, and model validation. These concepts are excellent in theory, but the proposed Agile V model delves deeper and provides tools for project planning and execution in complex product development projects.

Hybrid Model	Key Features	Strengths / Benefits	Limitations / Gaps	Source
Agile–Stage–Gate (A-SG) by Cooper & Fürst (2023)	Hybrid Agile–Stage–Gate model; embedded Agile in gating stages; customized approaches; focus on customer feedback and iterative sprints; self-managed teams.	More product–customer fit; higher team morale; flexibility; improved productivity; better innovation on physical products.	Not a cure-all; risk of speed over radical innovation; requires culture change; complexity in customization; dropped Agile practices; needs strong front-end homework; no guaranteed time savings; integration challenges with gating.	[5] Cooper & Fürst (2023)
Agile–Stage–Gate (A-SG) by Cooper & Sommer (2016)	Agile tools embedded in Stage–Gate, sprints, backlogs, hybrid roles	Faster market response, better communication, higher productivity, reduced work/rework, better customer alignment	Lack of quantitative validation, planning integration issues, leadership concerns, need adaptations for physical goods, role conflicts, management skepticism	[7] Cooper & Sommer (2016)
Agile–Stage–Gate Integration by Salvato & Laplume (2020)	ASGM blends traditional Stage–Gate Management (SGM), focused on structure and control, with Agile principles emphasizing flexibility and rapid feedback cycles (sprints)	-ASGM accelerates time-to-market by enabling iterative learning and fast feedback loops without waiting for all requirements to be fixed upfront -The model offers a "healthy tension" balancing Agile's flexibility for learning and iteration with SGM's structured governance	-Higher Resource Costs and Inefficiencies; Managing both traditional gate deliverables and Agile sprint outputs leads to duplicated documentation and reporting, creating complexity and resistance	[18] Salvato & Laplume (2020)

[Fig.3: Agile Hybrid Model Comparison]

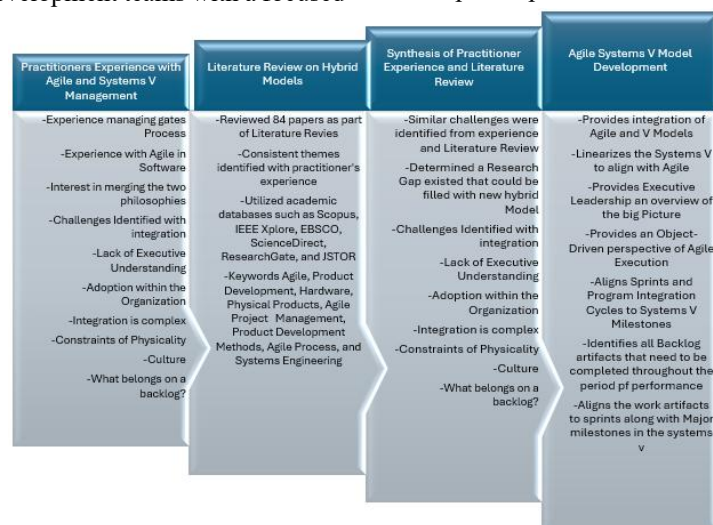
C. Experience and Literature Review Synthesis

The next phase of the Methodology combined knowledge from practitioner experience with the literature review. Findings show that both practitioners and peer-reviewed literature face similar challenges. Challenges include a lack of executive understanding of Agile, communication complications between executives and development teams, cultural resistance, and difficulties with hardware integration and Agile philosophies. The matching themes between these two domains strengthened the validity of the identified problems and identified the need for a hybrid solution to help close these gaps. This synthesis process used a systematic comparison of practitioner experience and academic findings to develop a model that combines theoretical foundations with real-world applications.

D. Development of the Agile Systems V Model

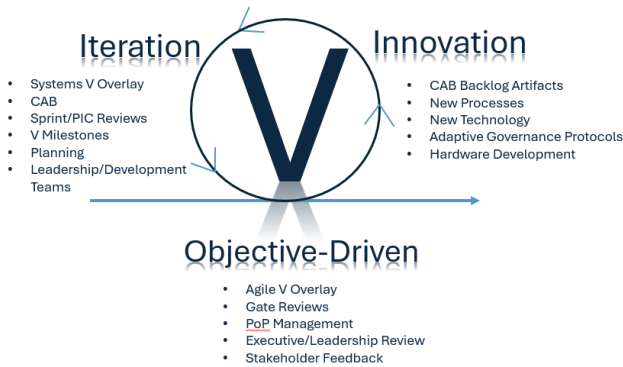
The proposed Agile Systems V Model was derived as a hybrid solution that combines Agile with the Systems Engineering V-Model. The model merges the traditional V-Model into an Agile sprint-based structure while still maintaining requirement-to-verification traceability throughout the verification stages as defined by [14]. The model enables executive leaders to monitor program execution while providing development teams with a focused

view of Agile work execution. This is needed because the executive team typically monitors project performance at a high level and is interested in the status of significant milestones. Secondly, the executive team usually consists of people who are not well-versed in Agile principles and understand the traditional, milestone-driven framework better. The model connects sprints with Systems V milestones through a central alignment representation visual system. The model provides a system-level milestone visibility tool for leadership oversight and a backlog management tool to identify and manage work artefacts for product development and regulatory conformance. The model integrates backlog management into the V-Model visual aid to define backlog content and maintain complete artefact tracking between sprints and systems engineering V milestones. The Agile Systems V Model integrates these features to solve the identified problems by (1) enhancing executive oversight, (2) handling system integration challenges and oversight, (3) considering constraints of physicality, and (4) establishing a method for backlog planning and management. The resulting methodology combines Agile flexibility with V-Model structure to address the research need across both practical applications and the literature. Figure 4 provides an overview of the Methodology development process.



[Fig.4: Methodology]

IV. AGILE V MODEL INTRODUCTION



[Fig 5: Agile V Core Concepts]

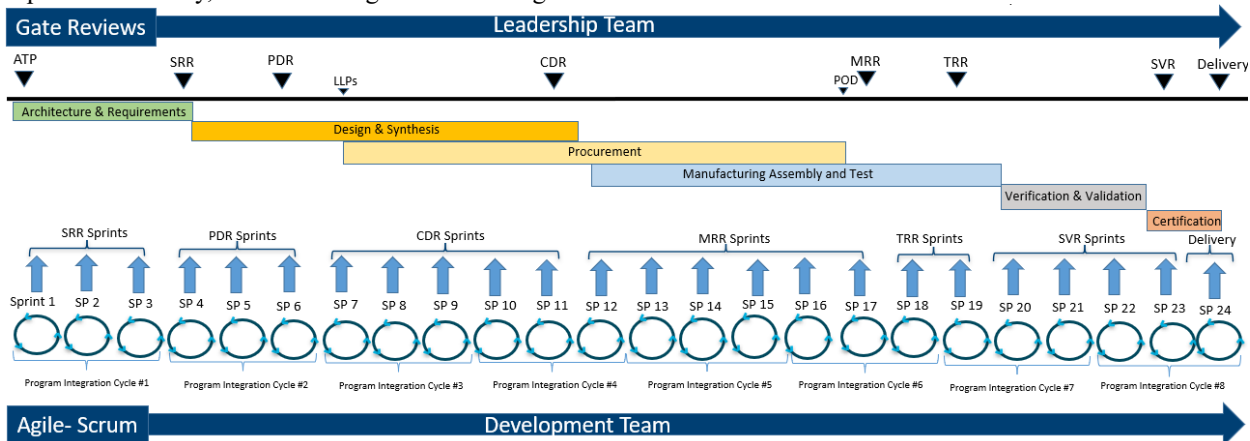
The Agile System V-Model combines the Agile V Overlay with the CAB to create a practical framework for product development. The V Overlay shows how Agile sprints integrate within the Systems-V timeline structure. The CAB serves as the core organizational unit which enables project and sprint planning activities that executives often feel are lacking in Agile hybrid models.

Iteration, Innovation, and Objective-Driven execution are the three main pillars of the Agile V Model. The integration of these components provides a structured method for product development. Iteration is achieved through the CAB management, Sprint Reviews, and Systems V integration. Secondly, Innovation is achieved by identifying new strategies for developing backlog artefacts, improving development efficiency, and advancing new technologies to

enhance system development. Lastly, provide an Objective-Driven environment stemming from the systems V framework, which utilises Gate Reviews to drive execution rigour, improve leadership communication, and enhance overall PoP Management within a Sprint-to-Sprint Agile framework. Figure 3 highlights the model's key components, which were derived from practitioners' experience and the literature review.

V. AGILE V OVERLAY

The Agile V Overlay Model presents a visual representation that merges the systems V Model with the Sprint cycles throughout the product development lifecycle. This tool enables improved strategic planning, resource management, and risk and opportunity management. The extended planning cycle enables agile planning for work execution to align with traditional systems engineering milestones downstream of the product development lifecycle. The hybrid approach demonstrates potential to improve systems engineering concepts by combining adaptable methods with structured, milestone-based approaches. In summary, the overlay model is used to lay out the engineering lifecycle and tie Systems V gates to Sprints for planning. It also provides a mechanism to track the project's progress. Sometimes, project progress gets lost in sprints, leading to a loss of sight of the big picture. This overlay picture is helpful to the Executive and Development teams. The Agile Overlay is depicted in Figure 4. Gate Review Milestone & Engineering Lifecycle Phases definitions can be found in Appendix B.



[Fig.6: Agile V Overlay]

A. Sprints & Program Integration Cycles Integration with Systems V Milestones

In typical Agile practice, sprints are fixed-length, time-boxed iterations in which cross-functional teams deliver incremental progress on system development. After sprint planning is complete, sprint execution begins. The sprint cycle concludes with a sprint retrospective that reflects on the sprint just completed. Its primary purpose is to enable the team to evaluate their performance, processes, and collaboration, identifying what went well, what could be improved, and how to implement changes for future sprints. The length of a sprint depends on organisational needs and the project's complexity. Typically, shorter sprints enable rapid adaptation in rapidly changing environments, while

longer sprints allow for more thorough progress in more challenging environments. The model uses sprint duration to align with V-Model gates, enabling teams to deliver incremental work to support milestone goals while maintaining adaptability to requirement changes. PICs oversee team output coordination, manage larger work packages, and ensure program-wide goal alignment. PICs operate with a different approach than sprints, focusing on strategic alignment through larger work packages that help achieve program-level goals. Basically, the PIC philosophy provides a longer planning cycle than a typical 30-day sprint to enable longer-term planning.

B. Sprint-to-Milestone Alignment

Sprints 1 through 3 produce artefacts which serve as review packages for the System Requirements Review (SRR). The three sprints included all backlog items from the CAB required to fulfil SRR requirements. Sprints 4–6 focus on producing the artefacts required to support the Preliminary Design Review (PDR). These sprints address all backlog items identified in the CAB that are critical to achieving PDR objectives. In addition to monitoring and updating the artefacts created during Sprints 1–3, new efforts focus on advancing specifications, refining requirements, and maturing programmatic plans, such as make/buy strategies, supplier management approaches, and risk mitigation planning. The development of Critical Design Review (CDR) supporting artefacts continues through Sprints 7–11. The development work from Sprints 4–6 serves as the base for these sprints, which complete all backlog items from the CAB to achieve PDR objectives. The current phase requires teams to enhance specifications and requirements, advance make/buy plans, and refine supplier management approaches. The team develops essential plans and procedures to help the project meet downstream review requirements and system delivery needs. The design engineering work during these sprints' advances from 10–20% drawing development to more mature design stages, reaching a 90% drawing completion readiness level to support CDR. The expanded engineering output serves two essential purposes: delivering vital information for the PDR assessment and laying the foundation for the Critical Design Review (CDR) phase. The development of Manufacturing Readiness Review (MRR) supporting artefacts continues through Sprints 12–17. The systems engineering process builds on the design maturity from Sprints 7–11, progressing toward manufacturing and production readiness. The remaining backlog items from the CAB which match MRR requirements are focused on fulfilling both technical and programmatic requirements to complete MRR. The teams enhance system requirements to production requirements while making final purchasing decisions for tooling and manufacturing readiness. The development of detailed manufacturing procedures, assembly plans, quality assurance protocols, and configuration management systems continues to support production capability verification efforts as part of the MRR. The design engineering process moves forward with drawing development that reaches complete definition for production, fabrication, and integration purposes. Sprints 18 through 19 consist of all the necessary work required to conduct a TRR successfully. All required artefacts under the input criteria are identified in the CAB. This is the point in the project where a transition occurs from manufacturing to systems testing readiness review. The primary intent of these sprints is to develop the artefacts for the required testing documentation, testing facility readiness, testing personnel readiness, and testing equipment needed to conduct the system test. The development of System Verification Review (SVR) supporting artefacts occurs during Sprints 20–23. The System Verification Review (SVR) requires this phase to demonstrate that the manufactured system fulfils all requirements. The system design and manufacturing outputs are validated through extensive testing during Sprints 20–23, demonstrating that the product meets all contractual and

customer requirements. The program achieves readiness for the System Verification Review through these deliverables, which also create a base for acceptance and operational readiness transition. The main objective of Sprint 24 is to complete and review all necessary artefacts that will enable system delivery to the customer. The system uses verified outputs from Sprints 20–23 and SVR success to create delivery-ready products and documentation. The sprint focuses on completing CAB backlog items that fulfil delivery needs and meet contractual, technical, and logistical requirements. The sprint produces complete configuration management records, test reports, and quality artefacts necessary for customer buy-off. The sprint also unifies supplier close-out packages with compliance statements to prove complete requirement fulfilment throughout the supply chain and product development lifecycle. The artefacts produced during Sprint 24 serve as complete proof of program completion, enabling a system that fulfils requirements and meets stakeholder expectations while being operationally ready. A comprehensive listing of all artefacts and their definitions is in Appendix A. A comprehensive listing and description of lifecycle phases is available in Appendix C.

C. Strategic Milestone Alignment Bars

The Agile V-Model operates through two distinct communication pathways that target different stakeholder groups via the Executive/Leadership Bar and the Development Bar. The Agile V Overlay includes tools that address the ongoing challenge of connecting executive oversight to Agile development. Typically, executives are more familiar with traditional project management frameworks and can't see the "big picture" within Agile cycles. The Executive/Leadership Bar shows the overall development status of major V-Model milestones, including System Requirements Review (SRR), Preliminary Design Review (PDR), Critical Design Review (CDR), Manufacturing Readiness Review (MRR), System Verification Review (SVR), and Delivery.

VI. CAB OVERVIEW

The CAB is a tool that will maintain and monitor all backlog artefacts which fulfil compliance and regulatory needs throughout the PoP. The categorisation method creates a unified system that links work artefacts to sprint cycles and major Systems V milestones for monitoring across agile development cycles and gate reviews. The CAB establishes a logical sequence for physical deliverables, consisting of the Digital Modelling, Additive Manufacturing, Engineering Development Unit, Qualification Unit, and Final Qualified Design Unit stages. This framework identifies typical hardware evolution that "pure Agile" systems struggle with. The CAB identifies backlog orders, which guide teams on work sequencing by working all artefacts in parallel rather than following a sequential finish-to-start method. The parallel processing method speeds up development schedules while maintaining flexibility, which supports the model's goal of merging fast development with controlled systems engineering practices. The CAB serves as an adaptable starting point for Agile V-Model artefact

management, which organisations can customise to meet their specific organisational and regulatory requirements. The CAB aligns with various compliance requirements and project needs through its adaptable design, which supports

traceability and scalability. An overview of the CAB is depicted in Figure 5 below, and the complete CAB is attached in Attachment 1.

LEGEND											
X: Specifies when the artifact must be completed to align with the Milestone/Sprint.											
---: Represents the timeframe during which the artifact should be initiated and progressively developed until completion.											
Gate Review	Priority	ATP	SRR	PDR	CDR	TRR	SVR	Cert			
Program Integration Cycle		PIC 1	PIC 2	PIC 3	PIC 4	PIC 5	PIC 6	PIC 7	PIC 8		
Physical Deliveries	Sprint	Sprint 1	Sprint 4	Sprint 5	Sprint 6	Sprint 11	Sprint 12	Sprint 13	Sprint 14	Sprint 19	Sprint 23
Digital Modeling		X	X	X	X	X	X	X	X	X	X
Digital Modeling + Additive Manufacturing			X	X	X	X	X	X	X	X	X
Digital Modeling + Additive Manufacturing + Engineering Development Unit						X	X	X	X	X	X
Digital Modeling + Additive Manufacturing + Engineering Development Unit + Qualification Unit								X	X	X	X
Qualified Design Unit										X	X
Risk & Issue & Opportunity Management	1	→	→	→	→	→	→	→	→	→	X
Staffing Plans	1	→	→	→	→	→	→	→	→	→	X
Performance Metrics/KPIs	1	→	→	→	→	→	→	→	→	→	X
Change Management Records	1	→	→	→	→	→	→	→	→	→	X
Integrated Schedule (IMS)	1	→	→	→	→	→	→	→	→	→	X
Cost Management	1	→	→	→	→	→	→	→	→	→	X
FMECA	1	→	→	→	→	→	→	→	→	→	X
Documentation/Soft Deliveries (EPICs/Tasks)											
Stakeholder Needs Assessment	1	→	X								
CONOPS (Concept of Operations)	1	→	X								
System Trade Studies	1	→	X								
System Architecture	1	→	X								
Initial Top-Level (System Level/A-Specs) Requirements	1	→	X								
Production Strategy	1	→	X								
Make/Buy Plans	1	→	X								
Supply Chain Architecture	1	→	X								
Draft Verification Matrix	1	→	X								
Engineering Modeling Strategy	1	→	X								
Critical Suppliers Identified	1	→	X								
Develop Supplier SOWs and Procurement Specs	1	→	X								
Draft ICDs (Interface Control Documents)	2	→	X								
Develop Functional Flow Diagrams	2	→	X								
Develop Control Drawings	2	→	X								
Update System Safety Engineering Plan	2	→	→	→	X						
Develop Interface Definition	2	→	→	→	X						
Update Program Management Plan	2	→	→	→	X						
Update Systems Engineering Management Plan	2	→	→	→	X						
Update Risk Management Plan	2	→	→	→	X						
Update Make/Buy Plans	2	→	→	→	X						
Update Prototype Plan	2	→	→	→	X						
Update Verification Plans	2	→	→	→	X						
Update Configuration Management Plan	2	→	→	→	X						

[Fig.7: Comprehensive Adaptive Backlog (CAB)]

A. Backlog Artefacts

The CAB serves as the core element of the Agile V-Model by creating a flexible database which expands the standard Agile backlog to support hardware-based and regulatory requirements. The system captures all engineering artefact deliverables, which include digital models, engineering drawings, design information, test plans, test reports, compliance documentation, and certification evidence. All this accumulated evidence is typically referred to as the ADP (Acceptance Data Package), which contains all the necessary information customers require to accept the deliverable hardware. There are two things to deliver in large-scale systems engineering development products: 1) the physical hardware itself and 2) the documentation that shows compliance with regulatory requirements. The items follow a structured arrangement that links to essential V-Model stages, including SRR, PDR, CDR, MRR, SVR, and Delivery, to meet program targets and comply with regulatory standards. The CAB enables flexible Agile iteration and milestone-driven oversight through its product structure, which supports teams in managing complex changes and maintaining compliance throughout the entire process.

B. Digital Modelling

The process of digital modelling starts with hardware development, including the creation of virtual system representations such as sketches, 3D models, and simulations. The Agile V-Model requires this work to be completed during the first sprints to achieve the SRR and PDR milestones. In some cases, there could be physical models and prototypes by the SRR/PDR timeframe. The team conducts multiple rounds of model enhancement based on stakeholder feedback, producing CAD drawings and simulation artefacts. The CAB

tracks output origins through its management system, and the Agile V Overlay allows teams to connect their work progress to milestone assessments for performance tracking by both team members and executives. The iterative approach allows designers to make quick adjustments that would be challenging to achieve in the standard V-Model process.

C. Additive Manufacturing

The process of additive manufacturing creates physical prototypes from complete digital models through 3D printing to verify fit, form, and, if applicable, functional performance. The stage is prepared for the Critical Design Review. The iterative prototyping process in sprints enables teams to evaluate hardware concepts by testing prototypes, which produce experimental results and material analysis findings. The CAB contains items that meet the review requirements, and the Agile V Overlay monitors progress toward the milestone. The process reduces development time by enabling quick feedback cycles that outperform conventional sequential prototyping.

D. Engineering Development Unit (EDU)

The Engineering Development Unit integrates mechanical and electrical components into a working prototype, enabling complete system testing. The EDU needed multiple development sprints leading up to the Manufacturing Readiness Review to reach mechanical integrity and basic functional performance validation readiness for teams. The EDU units typically start after the PDR, as the design begins. EDU units are used for preliminary testing at early stages of the design. The Agile V Overlay system provides an effective way to identify where in the process the system design is. From experience, executives

sometimes get lost in sprints and struggle to see the big picture, especially when it comes to physical product development cycles.

E. Qualification Unit

The EDU validation process results in the creation of qualification units, which must undergo formal testing and verification in operational settings. Qualification unit development typically begins after CDR approval, as the milestone establishes the product baseline. The CAB tracks environmental test data, compliance data, and verification/test reports generated during this stage to verify compliance with requirements.

F. Fully Qualified Design Unit

The last production phase produces operational-ready hardware products that have completed qualification. The last stage of the sprint includes completing integration work, user acceptance testing, and preparing qualification documentation. The CAB verifies that all artefacts, such as compliance evidence and final design packages, are finished and have proper traceability. The Agile V Overlay represents the final stage of performance, which unites flexible iterative work with strict milestone-based monitoring.

G. Development Artefact Allocation to Sprints and Milestones

The Agile Systems V-Model uses the CAB as its central repository for all artefacts needed to complete the project scope within organisational and regulatory requirements. It uses a systematic approach to identify and distribute artefacts between sprints and milestones by placing each item before assigning it to a sprint and Program Integration Cycle (PIC) while connecting it to its supporting Systems V milestone. The Agile V Overlay contains completion points that indicate when artefacts must be completed, and start points that define when work activities must begin. The philosophy is that all artefacts should start at the same time (Authorisation to Proceed) and progress through the product development process. The goal is to execute tasks in parallel (start-to-start) rather than serially (finish-to-start).

VII. DISCUSSION

The Agile Systems V Model combines the philosophies of Agile and the Systems Engineering V Model to address four main concerns identified in the literature and practitioners' experience. The four concerns are 1) Integration complexity, 2) backlog management, 3) executive oversight, and 4) constraints of physicality. The previous frameworks showed Agile and Stage-Gate principles could work together, but they did not provide specific operational methods for their integration. The Agile Systems V Model resolves this gap by integrating the Agile V Overlay & CAB. The Agile Systems V Model is a theoretical framework that requires experimental testing to demonstrate its effectiveness. The model needs further research to determine how backlog prioritisation, cost management, and resource distribution should operate within large, function-based teams. A question that arises is: "What type of organisation structure supports an Agile V model?" Additionally, understanding how travel time would impact the workflow and backlog is essential.

Would Earned Value Management, which is typically used in traditional methods, be used in the Systems V model? In real-world scenarios, all work isn't always completed on time, so if the job is not completed within planned sprints, does that uncompleted work initiate a new sprint? Or is the current sprint prolonged? The model requires additional research to develop operational elements and test its ability to scale up for practical implementation. To complete model validation, future work will use a mixed-methods research design to assess the Agile Systems V Model through surveys and semi-structured interviews to capture practitioners' experience.

VIII. CONCLUSION & FORWARD WORK

The Agile V Hybrid Model serves as a practitioner-centric framework that combines Agile adaptability with the Systems Engineering V-Model's systematic structure. The model enables sprint alignment through milestone reviews via the Agile V Overlay and CAB, which support planning and execution while also inhibiting executive oversight. The Agile V-Model provides a visual framework that enables organisations to implement theoretical models for managing highly regulated software and hardware projects. The manuscript offers both theoretical and practical aspects of the framework. Yet, future studies need to demonstrate their operational value through mixed-methods research that combines practitioner interviews with industry case studies to evaluate their practicality, effectiveness, and broader applicability. The Agile V Hybrid Model enables researchers and practitioners to tackle current product development challenges through its adaptable framework, which incorporates oversight elements for innovative work. The proposed Agile V-Model validation will use a mixed-methods research design, which combines survey data with semi-structured interview responses to collect feedback from practitioners who use Agile in their actual work environments. The surveys will use Likert-scale questions to measure practitioner satisfaction and system development process improvements, and to collect quantitative data about model effectiveness, usability, and project outcome impact. The research will use semi-structured interviews to gather qualitative data that will help practitioners explain their experiences, present their challenges, and propose improvements. The mixed-methods approach will validate the Agile V-Model, demonstrate its real-world usability, and reveal potential enhancements. Initials have been completed, and the initial findings do show positive signs. Some findings provided valuable suggestions for model improvement, which have been incorporated.

DECLARATION STATEMENT

After aggregating input from all authors, I must verify the accuracy of the following information as the article's author.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been funded by any organizations or agencies. This independence ensures that the research is conducted with objectivity



and without any external influence.

- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.
- **Data Access Statement and Material Availability:** The adequate resources of this article are publicly accessible.
- **Author's Contributions:** The authorship of this article is contributed equally to all participating individuals.

REFERENCES

1. Atlassian. (2024). What is agile? <https://www.atlassian.com/agile>
2. Atzberger, A., & Paetzold, K. (2019b). Current challenges of agile hardware development: What are the pain points nowadays? Proceedings of the International Conference on Engineering Design, ICED, 2019-August, 2209–2218. DOI: <https://doi.org/10.1017/dsi.2019.227>
3. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). Manifesto for agile software development. <https://agilemanifesto.org/>
4. Conforto, Edvandro & Amaral, Daniel & Luis, da Silva, & Di Felippo, Ariani & Kamikawachi, Dayse. (2016). The agility construct in project management theory. International Journal of Project Management. 34. 660-674. DOI: <https://doi.org/10.1016/j.ijproman.2016.01.007>
5. Cooper, R. G., & Furst, P. (2023). Deploying agile for physical-product development: Big challenges and practical solutions. IEEE Engineering Management Review, 51(3), 44–52. DOI: <https://doi.org/10.1109/EMR.2023.3304356>
6. Cooper, R. G. (2016). Agile-stage-gate hybrids: The next stage for product development. Research-Technology Management, 59(1), 21–29. DOI: <https://doi.org/10.1080/08956308.2016.1117317>
7. Cooper, R. G., & Sommer, A. F. (2016). The agile-stage-gate hybrid model: A promising new approach and a new research opportunity. Journal of Product Innovation Management, 33(5), 513–526. DOI: <https://doi.org/10.1111/jpim.12314>
8. de Borja, J. C. R., Trabasso, L. G., & Pessôa, M. V. P. (2019). Agile management in product development. Research-Technology Management, 62(5), 63–67. DOI: <https://doi.org/10.1080/08956308.2019.1638488>
9. Defence Acquisition University. (2023). Defence acquisition guidebook: Chapter 3—Systems engineering. <https://www.dau.edu/sites/default/files/2023-09/DAG-CH-3-Systems-Engineering.pdf>
10. Defence Acquisition University. (2023). Engineering & Manufacturing Development (EMD) Phase. Defence Acquisition Guidebook. <https://aaf.dau.edu/aaf/mca/emd-phase/>
11. Devopedia. (2023). Requirements development. <https://devopedia.org/requirements-development>
12. Edwards, K., Cooper, R. G., & Sommer, A. F. (2020). Agile-stage-gate in SMEs: Evaluating implementation in manufacturing. International Journal of Innovation Management, 24(5), Article 2050045. DOI: <https://doi.org/10.1142/S0219877019500482>
13. European Scrum.org. (2025). 98 Agile methodologies guide (Version 1.0). European Scrum.org. <https://www.europeanscrum.org>
14. International Council on Systems Engineering. (2015). INCOSE systems engineering handbook: A guide for system life cycle processes and activities (4th ed.). Wiley. <https://www.amazon.in/INCOSE-Systems-Engineering-Handbook-Activities/dp/1118999401>
15. Jocham, R. (2024, May 13). *Doing the right thing right – Validation and verification*. Scrum.org. <https://www.scrum.org/resources/blog/doing-right-thing-right-validation-and-verification>
16. Prostep ivip Association. (2024). Smart Systems Engineering – SmartSE: prostep ivip Recommendation PSI 11 Version 4.0 (pp. 117–132). Prostep ivip Association. https://www.prostep.org/fileadmin/prod-pay-download-8c1d/PSI11_RecV4_Final_aw.pdf
17. Rigby, Darrell K., Jeff Sutherland, and Hirotaka Takeuchi. "Embracing Agile: How to Master the Process That's Transforming Management." Harvard Business Review 94, no. 5 (May 2016): 40–50. <https://www.hbs.edu/faculty/Pages/item.aspx?num=51051>
18. Salvato, J. J., & Laplume, A. O. (2020). Agile stage-gate management (ASGM) for physical products. R and D Management, 50(5), 631–647. DOI: <https://doi.org/10.1111/radm.12426>
19. Schwaber, K., & Sutherland, J. (2020). The Scrum guide: The definitive guide to Scrum: The rules of the game. <https://www.scrumguides.org>
20. Serrador, P., & Pinto, J. K. (2015). Does agile work? —A quantitative analysis of agile project success. International Journal of Project Management, 33(5), 1040–1051. DOI: <http://dx.doi.org/10.1016/j.ijproman.2015.01.006>
21. Sutherland, J., Kamiyama, H., Aoyama, K., & Oizumi, K. (2015). Systems Engineering and the V-Model: Lessons from an autonomous solar-powered hydrofoil. Department of Systems Innovation, Graduate School of Engineering, The University of Tokyo. https://www.researchgate.net/publication/304022539_Systems_Engineering_and_the_VModel_Lessons_from_an_Autonomous_Solar_Powered_Hydrofoil
22. Systems Engineering Body of Knowledge (SEBoK). Pyster, A., Olwell, D., Hutchison, N., Enck, S., Anthony, J., Henry, D., Squires, A., & the BKCASE Editorial Board. (2025, May 27). *Vee life cycle model*. In A. Pyster, D. Olwell, N. Hutchison, S. Enck, J. Anthony, D. Henry, & A. Squires (Eds.), *Guide to the Systems Engineering Body of Knowledge (SEBoK)* (v. 2.12). The Trustees of the Stevens Institute of Technology. https://sebokwiki.org/wiki/Vee_Life_Cycle_Model
23. Wu, J. J. W. (2024). An exploratory study of V-Model in building ML-enabled software: A systems engineering perspective. In Conference on AI Engineering Software Engineering for AI (CAIN 2024) (pp. 1–11). ACM. DOI: <https://doi.org/10.1145/3644815.3644951>

AUTHOR'S PROFILE



Marc Berry is a highly skilled technical leader with substantial experience in product development, large-scale system integration, and cross-functional team management. Marc, who has a solid foundation in engineering management, supplier management, procurement, and proposal development, has been a key figure in advancing automotive & aerospace and defence (i.e., Lear Corporation and The Boeing Company). Marc holds a Bachelor of Science in Electrical and Computer Engineering from the University of Michigan and a Master of Science in Systems Engineering from Missouri S&T. He is currently pursuing a Doctor of Philosophy degree in Engineering Management at the Missouri University of Science and Technology (Missouri S&T).



Dr. Venkat Allada, is an Engineering Management & Systems Engineering Professor at the University of Missouri, Rolla, USA. He served as Missouri S&T's inaugural vice provost of graduate studies from 2007 to 2017. He is the recipient of the 1998 ASEE Dow Outstanding New Faculty Award, the 1998 Outstanding Young Manufacturing Engineer Award by the Society of Manufacturing Engineers, and the 2005 Outstanding Contributions Award by MIT's Lean Aerospace Initiative. He is the recipient of several Missouri S&T Faculty Excellence Awards, the Outstanding Teaching Award of Excellence, and the Innovative Teaching Award. He has over 100 publications in refereed journals and conference proceedings.

SUPPLEMENTAL DATA

Appendix A – Agile V Model Definitions

Artifact	Definition
Digital Modeling	The creation of virtual representations of systems, components, or processes to support design, analysis, and lifecycle decision-making.
Digital Modeling + Additive Manufacturing	Integration of digital models with additive manufacturing (3D printing) to produce physical prototypes directly from virtual designs.
Digital Modeling + Additive Manufacturing + Engineering Development Unit	Combines digital modeling and additive manufacturing with the fabrication of Engineering Development Units (EDUs), which are physical prototypes used to validate design concepts.
Digital Modeling + Additive Manufacturing + Engineering Development Unit + Qualification Unit	Extends the use of EDUs with Qualification Units (QUs), which are production-representative builds used to demonstrate compliance with design, performance, and certification requirements.
Qualified Design Unit	A fully validated and certified system unit that has passed qualification testing and meets all regulatory and customer requirements.
Risk & Issue & Opportunity Management	A structured approach to identifying, analyzing, mitigating, and tracking program risks, current issues, and potential opportunities.
Staffing Plans	Forecasts of personnel requirements aligned to program phases, skill needs, and resource allocation.
Performance Metrics/KPIs	Quantifiable measures (Key Performance Indicators) used to assess program, technical, and organizational performance.
Change Management Records	Formal documentation and tracking of approved modifications to requirements, design, or processes.
Integrated Schedule (IMS)	A master schedule that integrates technical, cost, and program tasks into a time-phased network to manage dependencies and milestones.
Cost Management	Processes for estimating, budgeting, monitoring, and controlling program expenditures.
FMECA	Failure Modes, Effects, and Criticality Analysis—a systematic method for identifying potential failure modes, their consequences, and mitigation strategies.
Documentation/Soft Deliveries (EPICs/Tasks)	Recorded artifacts such as engineering reports, models, or interim digital deliverables that support program progress.
Stakeholder Needs Assessment	Analysis of stakeholder expectations, priorities, and constraints to inform system requirements and design.
CONOPS (Concept of Operations)	A high-level description of how a system will be used to achieve stakeholder objectives.
System Trade Studies	Analytical comparisons of design alternatives to balance performance, cost, risk, and schedule.
System Architecture	The structured framework defining system components, interfaces, and relationships.
Initial Top-Level (System Level/A-Specs) Requirements	The earliest set of formalized system-level requirements defining performance and functional needs.
Production Strategy	A plan that defines how products will be manufactured, assembled, and scaled for production.
Make/Buy Plans	Decisions specifying which system elements will be produced in-house versus procured externally.
Supply Chain Architecture	The structure of suppliers, logistics, and partnerships supporting procurement and delivery.
Draft Verification Matrix	An initial mapping of system requirements to planned verification methods (test, analysis, inspection, demonstration).
Engineering Modeling Strategy	The defined approach for applying models and simulations throughout system development.
Critical Suppliers Identified	The process of selecting suppliers whose products or services are vital to program success.
Develop Supplier SOWs and Procurement Specs	Creation of contractual documents specifying technical and performance requirements for suppliers.
Technology Readiness Assessment	Evaluation of the maturity of technologies required for system development.
Technical Performance Measures Identified	Selection of critical performance parameters tracked to assess system progress toward requirements.
Place Major Suppliers under Purchase Contract	Formal contractual agreements securing delivery of critical parts, systems, or services.
Preliminary Design Review (PDR) Checklist	A structured evaluation checklist confirming readiness for preliminary design review.
Initial System Safety Assessment	Early analysis identifying hazards, risks, and mitigation strategies in system operation.
Initial Program Management Plan	The baseline plan for managing program execution, governance, and resources.
Initial Systems Engineering Management Plan	Plan defining systems engineering processes, roles, and responsibilities.
Initial Risk Management Plan	The initial framework for identifying, mitigating, and tracking risks.
Initial Prototype Plan	Early planning for creating and testing physical prototypes.
Initial Verification Plan	Initial definition of verification approaches and activities to demonstrate requirement compliance.
Initial Configuration Management Plan Identified	The first documented plan for managing design baselines, revisions, and traceability.
Initial Human Factors Engineering Plan	Plan for integrating human usability and ergonomics into system design.
Initial Supplier Management Plan	Initial plan for evaluating and overseeing suppliers throughout development.
Initial Regulatory Compliance Plans	Plans ensuring system development aligns with applicable regulations and standards.
Initial Test Plan	Baseline plan describing test objectives, methods, and schedules.
Initial Test Procedure	Preliminary step-by-step instructions for executing specific tests.
Initial Certification Plan	Outline of activities required to achieve regulatory or customer certification.
Initial Safety Engineering Plan	Early safety-focused plan identifying hazards and compliance with safety standards.
Initial Logistics and Support Plan	Plan for sustaining the system, including training, spares, and maintenance.
Initial Obsolescence Management Plan	Framework for mitigating risks associated with technology or component obsolescence.

Integrating Agile and the Systems V-Model: Proposing a Hybrid Solution

Initial Production Tooling Plan	Planning for the design and acquisition of production tools and fixtures.
Initial Manufacturing Plan	High-level plan for manufacturing processes, resources, and facilities.
Initial Component Qualification Plan	Strategy for validating individual components through testing and analysis.
Initial System Test Plan	Preliminary definition of system-level testing objectives and methods.
Initial Cyber Security Plan	Early plan for addressing cybersecurity risks and compliance.
Initial Quality Management Plan	Plan defining how quality assurance and control will be achieved.
Baseline Engineering Models and Architecture	The initial, validated engineering models and system architectural baselines.
Develop Long Lead Part Procurement Plan	Plan to acquire parts requiring extended lead times.
Develop Engineering Drawing Release Plan	Schedule and process for issuing official engineering drawings.
Develop Verification/Validation Plan	Plan outlining verification and validation methods across the system lifecycle.
Draft Requirements Traceability Matrix	Early mapping of requirements to design, test, and verification elements.
Draft ICDs (Interface Control Documents)	Preliminary documents defining system and subsystem interface requirements.
Develop Functional Flow Diagrams	Creation of diagrams depicting operational and functional sequences.
Develop Control Drawings	Drawings specifying critical system interfaces and constraints.
Update System Safety Engineering Plan	Revised safety engineering plan incorporating new data and design evolution.
Develop Interface Definition	Detailed specifications of system interfaces.
Update Program Management Plan	Revised program management baseline.
Update Systems Engineering Management Plan	Updated systems engineering framework based on program progress.
Update Risk Management Plan	Refined risk strategies and mitigation tracking.
Update Make/Buy Plans	Updated sourcing strategy decisions.
Update Prototype Plan	Refined prototyping approach.
Update Verification Plans	Updated strategies for verifying requirements.
Update Configuration Management Plan	Revised configuration management strategy.
Update Human Factors Engineering Plan	Refined human-system integration considerations.
Update Supplier Management Plan	Refined supplier oversight strategy.
Update Regulatory Compliance Plans	Updated compliance approach reflecting design maturity.
Update Test Plan	Revised testing scope and objectives.
Update Test Procedure	Refined test execution steps.
Update Certification Plan	Updated strategy for achieving certification.
Update Verification/Validation Plan	Updated lifecycle verification and validation strategy.
Update Safety Engineering Plan	Refined safety engineering processes.
Update Logistics and Support Plan	Revised logistics and sustainment framework.
Update Obsolescence Management Plan	Refined strategy for managing obsolescence risks.
Update Production Tooling Plan	Updated tool acquisition and management strategy.
Update Manufacturing Plan	Refined manufacturing execution framework.
Update Component Qualification Plan	Updated validation plan for critical components.
Update System Test Plan	Revised system-level testing approach.
Update Cyber Security Plan	Refined cybersecurity controls and strategies.
Update Quality Management Plan	Revised quality assurance and control framework.
Update Engineering Drawing Plan	Updated engineering drawing release and revision control plan.
Develop Initial Analysis Reports	Early analytical studies to support design validation.
Develop Electrical Design Drawings	Creation of electrical schematics and layout drawings.
Develop Electrical Installation Drawings	Drawings specifying electrical installation details.
Develop Structural Design Drawings	Design drawings defining structural components.
Develop Structural Installation Drawings	Drawings specifying installation details for structural components.
Procure Long Lead Items	Acquisition of components requiring long manufacturing times.
Finalize Program Management Plan	Approved and baselined program management plan.
Finalize Systems Engineering Management Plan	Approved final systems engineering plan.
Finalize Risk Management Plan	Approved final risk management strategy.
Finalize Make/Buy Plans	Approved final sourcing strategy.
Finalize Prototype Plan	Approved final prototyping approach.
Finalize Verification Plans	Approved verification strategy.
Finalize Configuration Management Plan	Approved final configuration management framework.
Identified	

Finalize Human Factors Engineering Plan	Approved human-system integration strategy.
Finalize Supplier Management Plan	Approved supplier oversight plan.
Finalize Regulatory Compliance Plans	Approved compliance strategy.
Finalize Test Plan	Approved testing framework.
Finalize Test Procedure	Approved testing steps.
Finalize Certification Plan	Approved certification strategy.
Finalize Verification/Validation Plan	Approved verification and validation plan.
Finalize Safety Engineering Plan	Approved safety engineering framework.
Finalize Logistics and Support Plan	Approved sustainment and logistics framework.
Finalize Obsolescence Management Plan	Approved obsolescence management strategy.
Finalize Production Tooling Plan	Approved production tooling acquisition plan.
Finalize Manufacturing Plan	Approved manufacturing framework.
Finalize Component Qualification Plan	Approved component validation strategy.
Finalize System Test Plan	Approved system-level test plan.
Finalize Cyber Security Plan	Approved cybersecurity framework.
Finalize Quality Management Plan	Approved quality assurance and control framework.
Update Digital Models (DMs)	Revised and validated digital system models.
Fabricate Additive Manufacturing Models	Production of physical parts via additive manufacturing.
Fabricate Engineering Development Units (EDUs)	Construction of prototype units for design validation.
Release Final Engineering Drawings	Issuance of production-ready engineering drawings.
Develop Final Bill of Material	Completed list of all system components and materials.
Review Production Planning	Assessment of production readiness and planning adequacy.
Review Work Instructions	Evaluation of manufacturing and assembly procedures.
Develop Production Tooling Engineering	Engineering of specialized tooling required for production.
Procure Fixture & Tooling Material	Acquisition of materials for tooling and fixtures.
Parts Receivable	Receipt and inspection of procured components.
Certify Technicians	Formal qualification of production and test personnel.
Production Readiness Checklist	Checklist verifying readiness for manufacturing start.
Review Manufacturing Plan	Final review of manufacturing execution framework.
Develop Training Materials for Production	Creation of instructional content for production staff.
Conduct Manufacturing Readiness Assessment (MRA)	Formal assessment of manufacturing readiness.
Conduct MRR	Manufacturing Readiness Review—formal review confirming manufacturing maturity.
Finalize Test Plan	Locked and approved final test plan.
Finalize Test Procedures	Approved final test execution steps.
Finalize Requirements Traceability Matrix	Completed mapping of requirements to verification methods.
Finalize Test Environment Setup	Completion of test facilities and resources.
Verify Testing Resources	Validation of test equipment availability and adequacy.
Finalize Pre-test Calibration and Validation Reports	Completion of calibration and validation reports before testing.
Prepare Test Items and Acceptance Documentation	Readiness of test articles and acceptance records.
Review of Preliminary Test Results (Quick Look Reports)	Initial evaluation of test outcomes.
Finalize Site Safety Review	Completion of site-level safety review.
Verification of Test Parts and Material Acquisition	Confirmation that required test materials are available.
Conduct TRR (Test Readiness Review)	Formal review confirming readiness to conduct testing.
Write Quick Look Test Reports	Preparation of early test result summaries.
Complete Post-TRR Open Actions	Resolution of outstanding TRR-related issues.
Finalize Post-TRR Open Actions	Closure of all TRR follow-up items.
Finalize Test Reports	Approved final test reports.
Finalize Analysis Reports	Approved final analysis documents.
Finalize Inspection Reports	Approved final inspection records.
Finalize Demonstration Reports	Approved final demonstration reports.
Develop Verification Change Notices	Formal notices documenting verification changes.
Finalize Requirements Traceability Matrix	Approved final mapping of requirements to verification methods.
Conduct FCA	Functional Configuration Audit—formal verification that functional requirements are met.
Conduct PCA	Physical Configuration Audit—formal verification that physical design matches specifications.
Conduct System Verification Review (SVR)	Formal review confirming that the system meets all requirements.
Develop Transition to Production Plan	Plan guiding the shift from development to full production.
Certification Review	Final assessment confirming compliance with regulatory and certification standards.

Appendix B – Systems Engineering V Milestones

A. Authorisation to Proceed (ATP)

Official authority for the contractor to begin work. The procuring contracting officer usually issues it. This is a common term used in government contracting and large-scale systems engineering product development initiatives.

B. System Requirements Review (SRR)

A System Requirements Review (SRR) is a project management step that verifies system requirements are complete, consistent, and achievable before proceeding to the next step of system design. It ensures requirements are unambiguous, verifiable, and traceable from system-level requirements down to component-level requirements. Additionally, confirming that all

parties have a shared understanding of the current state of the requirements and agree that the acceptable risk exists to move forward. This is a crucial technical review in the early phases of a project lifecycle. SRR establishes the functional baseline, which defines the system's performance, functional, and interface requirements in a formal, documented manner [10].

C. Preliminary Design Review (PDR)

A Preliminary Design Review (PDR) is a formal, critical checkpoint in the development of a complex system to determine whether the preliminary design is complete and technically ready to proceed to the next phase, detailed design. It confirms that the design meets requirements, that risks have been identified, and that mitigation plans are in place to ensure the project's successful execution. The Preliminary Design Review (PDR) is a formal event that establishes the allocated baseline. The allocated baseline defines the preliminary design of all hardware, software, and support systems for a project [10].

D. Long Lead Parts Procurement Start (LLPL)

Long-lead parts procurement refers to ordering and acquiring parts and/or materials with extended delivery times—typically months or even years—to ensure they are available without causing schedule delays. This involves early identification of these items, effective supplier management, proactive planning, and strategic risk mitigation to minimise their impact on the project's schedule and budget. The approach reduces long-lead schedule risks through its combination of planning to place necessary purchase contracts with suppliers early enough to allow for part and material development. The goal is to have the parts and materials at the manufacturing facility on or before the needed dates for the start of manufacturing. The parts procurement cycle in a hybrid model will need to span multiple sprints, as parts typically have lead times that do not fall within a 30-day sprint window.

E. Critical Design Review (CDR)

A Critical Design Review (CDR) is a formal, multi-disciplinary technical assessment to ensure a system design is complete and ready for manufacturing, assembly, and testing. A Critical Design Review (CDR) establishes the Product Baseline, which is the detailed design documentation that represents the system's physical characteristics intended for production, deployment, and operations [10]. The Critical Design Review (CDR) confirms that the entire system design fulfils all requirements and is prepared for production or implementation.

F. Parts on Dock (POD)

The Parts on Dock (POD) status indicates that all necessary components, including long-lead items, are present for assembly, integration, and test work. POD is the final step in the Long Lead Procurement task duration, as this is where all needed parts are available for the start of manufacturing. Again, the parts procurement cycle in a hybrid model will need to span multiple sprints, as parts lead times typically do not fall within a 30-day sprint window.

G. Manufacturing Readiness Review (MRR)

A Manufacturing Readiness Review (MRR) is a milestone that determines if a product's design and manufacturing processes are ready for production. This milestone typically follows CDR (Critical Design Review). Its purpose is to assess whether the design is stable, whether risks have been identified and addressed, and whether the organisation has adequate plans and resources for the manufacturing phase. This milestone helps to ensure the project can start production without unacceptable risks to cost, schedule, and performance [10].

H. Test Readiness Review (TRR)

A Test Readiness Review (TRR) is a milestone that determines that a product or system is ready to begin formal testing. It assesses if the test plan, procedures, and objectives are complete, traceable to program requirements, and that risks are manageable to proceed. This review helps prevent rework by confirming the system's developmental maturity and readiness to be tested within the defined scope and safety criteria [10].

I. System Verification Review (SVR)

System Verification Review (SVR) is a technical assessment, typically in defence systems, that verifies system performance against its functional baseline requirements identified at SRR to ensure compliance [10]. This review shows that the system is ready to move into the production phase, checks for discrepancies, and verifies that the final product meets the design requirements. This milestone typically verifies that the system was built and manufactured according to the intended design. SVRs are crucial in government and defence contracting to prevent costly rework and ensure the successful development of systems and projects.

J. Delivery

The delivery phase is the final stage of the development lifecycle, delivering the finished product or system to its intended end users. All previous gates and milestones have been completed, and the stakeholders have agreed to the terms and conditions for product development. At this point, the customer has officially purchased the product or system and has verified and validated that all requirements of the product development life cycle have been fulfilled.

Appendix C – Engineering Lifecycle Phases

The requirements development phase, known as requirements engineering, is a core process in systems engineering. The process defines system requirements that fulfil stakeholder needs and operational constraints, thereby achieving mission and system objectives. [11] explains that this stage includes requirement collection and analysis, specification and requirements development, and continuous management of stakeholder needs to achieve alignment between system design and stakeholder expectations. A robust requirements development phase helps prevent costly design rework and maintains system lifecycle traceability, which auditory personnel typically rely on later in the product development process



[11]. The systems engineering process includes the design phase, which connects architectural development to system implementation. The design phase converts system architecture and requirements into design requirements, specifications, and engineering drawings, which describe how each subsystem and component should perform and interact with other design artefacts. System design according to [9] requires developing detailed models of system element interactions, which verify that each component fulfils its unique requirements. The procurement process in complex systems acquisition maintains a close relationship with systems engineering activities. The acquisition strategy needs to align with system requirements, the existing architecture, and project limitations, including cost constraints, risk factors, technology readiness, and time deadlines [20]. The Agile V Overlay integrates procurement lifecycle phases, starting with Long Lead Parts Procurement Start (LLPL) and ending with Parts on Dock (POD), which are the primary milestones for large-scale system development procurement. The Long Lead Part process reduces supply chain risk by procuring parts and materials with the longest lead times under the Purchase Contract before official engineering is released. The parts procurement duration often exceeds 30 days, so long-lead items span multiple sprints. Procurement progress should be evaluated at each sprint to ensure schedule alignment. The Manufacturing stage transforms design artefacts into physical products through controlled processes, which produce components and subsystems [10]. The production of prototypes that represent actual products helps verify system fit, form, and functionality before starting full-scale production or deployment [20]. The Test phase runs concurrently with assembly in some cases to verify system functionality and operational requirements through multiple testing methods, including functional and environmental tests. The testing process verifies system performance under operational conditions and generates feedback that helps guide design modifications and process improvements, if needed [10]. At this point, if testing identifies a need for redesign, then schedule and cost impacts will most likely be realised. There are two ways to interpret this data. First, the purpose of testing is to identify flaws in the design, but the identified issues cause delays to the schedule and increased rework costs. The proper execution of manufacturing, assembly, and testing operations results in systems which are ready for production and fully verified and validated for operational use. System designs become reality through the Manufacturing, Assembly, and Test phases. The Agile V-Model includes sprints, which enable continuous improvement of production processes, product validation, and problem resolution during each development cycle. The Agile V Overlay defines a sequence which links these activities to Manufacturing Readiness Review (MRR) checkpoints, and the CAB confirms that manufacturing deliverables fulfil verification requirements during gate reviews. Verification and validation (V&V) steps are quality assurance provisions within the systems engineering framework that ensure a system is both properly built and meets its intended purpose. Verification focuses on confirming that the system, component, or product satisfies all specified requirements, addressing the question: “Did we build the product right?” [15]. Verification activities include

reviews, inspections, tests, and analyses conducted at components, subsystems, and system levels, comparing outcomes against requirements. These processes provide evidence that the system meets design requirements and maintains traceability across the development lifecycle [20]. In contrast, validation determines whether the final system fulfils stakeholder needs and operational expectations, answering the question: “Did we build the right product?” [15]. Validation typically occurs following system integration and involves stakeholder participation and concurrence. Certification is the final step in the systems engineering lifecycle, as it is the official step that “sells off” the product to the end user. [20] states that certification activities consist of three main steps: checking compliance documentation, verifying safety and reliability performance, and conducting audits and independent evaluations or inspections.

Attachment 1 – Full CAB Repository



CAB - Attachment
1.xlsx

See the attachment