

The Hyper-Accelerated Daily Delivery Guide

The Definitive Guide to HADD:
Your Guide to Hyper-Accelerated Daily Delivery

2024 Edition

Table of Contents

[Purpose of the HADD Guide](#)

[HADD Definition](#)

[HADD Theory](#)

[Transparency](#)

[Inspection](#)

[Adaptation](#)

[HADD Values](#)

[HADD Team](#)

[Product Owner](#)

[Dev / AI Engineer](#)

[Software Architect / Cloud Architect](#)

[Quality Assurance \(QA\)](#)

[HADD Events](#)

[The Micro-Sprint](#)

[Morning Stand-up & Planning](#)

[Development Phase](#)

[Pre-Delivery Meeting](#)

[HADD Artifacts](#)

[Product Backlog](#)

[Daily Backlog](#)

[Increment](#)

[Definition of Done](#)

[HADD Success Metrics](#)

[End Note](#)

Purpose of the HADD Guide

We developed HADD in response to the accelerating pace of software development in the Age of AI. As organizations seek to leverage artificial intelligence and cloud-native architectures to deliver value faster, traditional two-week sprint cycles have become bottlenecks. The Hyper-Accelerated Daily Delivery (HADD) Guide contains the definition of HADD, which consists of HADD's roles, events, artifacts, and the rules that bind them together.

Ken Schwaber and Jeff Sutherland developed HADD; the HADD Guide is written and provided by them. Together, they stand behind the HADD Guide.

The HADD Guide is purposefully incomplete, only providing the minimal required elements to implement HADD. Various processes, techniques, and methods can be employed within the HADD framework. HADD makes visible the relative efficacy of current management, environment, and work techniques, so that improvements can be made.

HADD Definition

HADD is a lightweight framework that helps people, teams, and organizations generate value through adaptive solutions for complex problems.

In a nutshell, HADD requires a Product Owner to order the work for a complex problem into a Product Backlog. The HADD team turns a selection of the work into an Increment of value during a Micro-Sprint. The HADD team and its stakeholders inspect the results and adjust for the next Micro-Sprint. Repeat.

HADD is simple. Try it as is and determine if its philosophy, theory, and structure help to achieve goals and create value. The HADD framework is purposefully incomplete, only defining the parts required to implement HADD theory. HADD is built upon by the collective intelligence of the people using it. Rather than provide people with detailed instructions, the rules of HADD guide their relationships and interactions.

Various processes, techniques, and methods can be employed within the framework. HADD wraps around existing practices or renders them unnecessary. HADD makes visible the relative efficacy of current management, environment, and work techniques so that improvements can be made.

HADD Theory

HADD is founded on empiricism and lean thinking. Empiricism asserts that knowledge comes from experience and making decisions based on what is observed. Lean thinking reduces waste and focuses on the essentials.

HADD employs an iterative, incremental approach to optimize predictability and control risk. HADD engages groups of people who collectively have all the skills and expertise to do the work and share or acquire such skills as needed.

HADD combines four formal events for inspection and adaptation within a containing event, the Micro-Sprint. These events work because they implement the empirical HADD pillars of transparency, inspection, and adaptation.

Transparency

The emergent process and work must be visible to those performing the work as well as those receiving the work. With HADD, important decisions are based on the perceived state of its three formal artifacts. Artifacts that have low transparency can lead to decisions that diminish value and increase risk.

Transparency enables inspection. Inspection without transparency is misleading and wasteful.

Inspection

The HADD artifacts and the progress toward agreed goals must be inspected frequently and diligently to detect potentially undesirable variances or problems. To help with inspection, HADD provides cadence in the form of its five events.

Inspection enables adaptation. Inspection without adaptation is considered pointless. HADD events are designed to provoke change.

Adaptation

If any aspects of a process deviate outside acceptable limits or if the resulting product is unacceptable, the process being applied or the materials being produced must be adjusted. The adjustment must be made as soon as possible to minimize further deviation.

Adaptation becomes more difficult when the people involved are not empowered or self-managing. A HADD team is expected to adapt the moment it learns anything new through inspection.

HADD Values

Successful use of HADD depends on people becoming more proficient in living five values:

The Hyper-Accelerated Delivery Manifesto (HADM)

We are uncovering better ways of developing software in the Age of AI by doing it and helping others do it.

The Core Values

- **AI-Augmented Velocity** over Manual Process
- **Continuous, Daily Delivery** over Fixed Sprints
- **Tightly-Coupled Dev/QA Teams** over Handoffs
- **End-of-Day Working Code** over Daily Progress Reports

The Principles

1. Satisfy the customer through continuous and daily delivery to Staging.
2. Welcome change, responding to it within the same workday.
3. Deliver verified working software to the Staging environment at the end of every workday.
4. Business (PO) and Dev/QA teams work together constantly and tightly.
5. Leverage AI agents to achieve sustainable development velocity.
6. The QA-validated code on Staging is the only measure of progress.
7. The best architectures emerge from self-organizing, cross-functional daily teams.
8. The week culminates in a final Production Release (MVP) every 5 business days.

When these values are embodied by the HADD team and the people they work with, the empirical HADD pillars of transparency, inspection, and adaptation come to life, building trust.

HADD Team

The fundamental unit of HADD is a small team of people, a HADD Team. The HADD team consists of one Product Owner, one or more Dev/AI Engineers, an embedded Software/Cloud Architect, and one or more Quality Assurance specialists. HADD teams are cross-functional, meaning the members have all the skills necessary to create value each day. They are also self-managing, meaning they internally decide who does what, when, and how.

The HADD team is small enough to remain nimble and large enough to complete significant work within a single workday. The HADD team model is designed to optimize flexibility, creativity, and productivity.

The entire HADD team is accountable for creating a valuable, useful Increment every day. HADD defines three specific accountabilities within the HADD team: the Developers, the Product Owner, and the Architect.

Product Owner

The Product Owner is accountable for maximizing the value of the product resulting from the work of the HADD team. How this is done may vary widely across organizations, HADD teams, and individuals.

The Product Owner is also accountable for effective Product Backlog management, which includes:

- Developing and explicitly communicating the Product Goal
- Creating and clearly communicating Product Backlog items
- Ordering Product Backlog items
- Ensuring that the Product Backlog is transparent, visible, and understood

The Product Owner may do the above work or may delegate the responsibility to others. Regardless, the Product Owner remains accountable.

For Product Owners to succeed, the entire organization must respect their decisions. These decisions are visible in the content and ordering of the Product Backlog, and through the Product Owner's comments and direction during the Morning Stand-up and Pre-Delivery Meeting.

Daily Gatekeeper: Clarifies Product Backlog Items (PBIs) in the morning, accepts delivery in the evening.

Dev / AI Engineer

Dev/AI Engineers are the people in the HADD team that are committed to creating any aspect of a usable Increment each day.

The specific skills needed by the Dev/AI Engineers are often broad and will vary with the domain of work. However, Dev/AI Engineers are always accountable for:

- Creating a plan for the day's work, the Daily Backlog
- Instilling quality by adhering to a Definition of Done
- Adapting their plan each day toward the Daily Goal
- Holding each other accountable as a professional team
- Leveraging AI agents for code generation, testing, and validation
- Performing same-day peer and architectural review

AI Velocity: Guides and validates AI agent output, performs same-day review and coding.

Software Architect / Cloud Architect

The Software Architect or Cloud Architect is embedded within the Dev team and is accountable for ensuring that all work adheres to architectural standards, security requirements, and scalability considerations. They work closely with Dev/AI Engineers to validate that AI-generated code meets non-functional requirements.

The Architect is accountable for:

- Ensuring architectural alignment and guardrails
- Validating cloud-native patterns and practices
- Reviewing code for security, scalability, and maintainability
- Ensuring CI/CD pipeline integrity
- Managing technical debt and complexity metrics

Architectural Guardrail: Ensures AI-generated code meets non-functional requirements and pipeline integrity.

Quality Assurance (QA)

Quality Assurance specialists are accountable for ensuring that every Increment meets quality standards and acceptance criteria. QA works concurrently with Dev, immediately validating code as it becomes available.

QA is accountable for:

- Defining test cases during Morning Stand-up
- Executing functional and exploratory tests immediately upon code availability

- Confirming all Acceptance Criteria are met
- Validating zero High or Critical bugs remain open
- Managing the secure push to Staging/QA environment
- Maintaining test automation and CI/CD test management

Same-Day Reliability: Works concurrently with Dev, immediately validates code, and manages the Staging push.

HADD Team Structure

Role	Recommended Team Size	Key Skills Required	HADD Contribution
Product Owner (PO)	1	Requirement Definition, Stakeholder Management, Acceptance Testing	Daily Gatekeeper: Clarifies PBI in the morning, accepts delivery in the evening
Dev / AI Engineer	Min. 2 (Max. 4)	LLM/AI prompt engineering, Cloud-native code, Code Review, Unit Testing	AI Velocity: Guides and validates AI agent output, performs same-day review and coding
Software Architect / Cloud Architect	Embedded within Dev Team	Deep knowledge of Cloud Platform (AWS/Azure/GCP), CI/CD, Security, Scalability	Architectural Guardrail: Ensures AI-generated code meets non-functional requirements and pipeline integrity
Quality Assurance (QA)	Min. 1	Test automation, Exploratory testing, CI/CD management	Same-Day Reliability: Works concurrently with Dev, immediately validates code, and manages the Staging push

Recommended Total Team Size: 3 to 5 members (Minimum 3, Maximum 5)

HADD Events

The Micro-Sprint is a container for all other HADD events. Each event in HADD is a formal opportunity to inspect and adapt HADD artifacts. These events are specifically designed to enable the transparency required. Failure to operate any events as prescribed results in lost opportunities to inspect and adapt. Events are used in HADD to create regularity and to minimize the need for meetings not defined in HADD.

Optimally, all events are held at the same time and place to reduce complexity.

The Micro-Sprint

The heart of HADD is a Micro-Sprint, a time box of one workday (8 hours) during which a "Done", useable, and potentially releasable product Increment is created.

Micro-Sprints have consistent durations throughout a development effort. A new Micro-Sprint starts immediately after the conclusion of the previous Micro-Sprint.

Micro-Sprints enable predictability by ensuring inspection and adaptation of progress toward a Product Goal at least every day.

Each Micro-Sprint may be considered a short project.

Various practices exist to forecast progress, like burn-downs, burn-ups, or cumulative flows. While proven useful, these do not replace the importance of empiricism. In complex environments, what will happen is unknown. Only what has already happened may be used for forward-looking decision-making.

Micro-Sprint Planning

Micro-Sprint Planning initiates the Micro-Sprint by laying out the work to be performed for the day. The resulting plan is created by the collaborative work of the entire HADD team.

The Product Owner ensures that attendees are prepared to discuss the most important Product Backlog items and how they map to the Product Goal. The HADD team may also invite other people to attend Micro-Sprint Planning to provide advice.

Micro-Sprint Planning addresses the following topics:

- **Topic One: Why is this Micro-Sprint valuable?** The Product Owner proposes how the product could increase its value and utility in the current Micro-Sprint. The whole HADD team collaborates to define a Daily Goal that communicates why the Micro-Sprint is valuable to stakeholders. The Daily Goal must be finalized prior to the end of Micro-Sprint Planning.

- **Topic Two: What can be Done this Micro-Sprint?** Through discussion with the Product Owner, the Dev/AI Engineers select items from the Product Backlog to include in the current Micro-Sprint. The HADD team may refine these items during this process, which increases understanding and confidence.
- **Topic Three: How will the chosen work get done?** For each selected Product Backlog item, the Dev/AI Engineers plan the work necessary to create an Increment that meets the Definition of Done. This is often done by decomposing Product Backlog items into smaller work items of one day or less. How this is done is at the sole discretion of the Dev/AI Engineers. No one else tells them how to turn Product Backlog items into Increments.

By the end of the Morning Stand-up & Planning, the Dev/AI Engineers should be able to explain to the Product Owner and QA how they intend to work as a self-managing team to accomplish the Daily Goal and create the anticipated Increment.

Daily Goal

The Daily Goal is the single objective for the Micro-Sprint. Although the Daily Goal is a commitment by the Dev/AI Engineers, it provides flexibility in terms of the exact work needed to achieve it. The Daily Goal also creates coherence and focus, encouraging the HADD team to work together rather than on separate initiatives.

The Daily Goal is created during the Morning Stand-up & Planning event and added to the Daily Backlog. As the Dev/AI Engineers work during the Micro-Sprint, they keep the Daily Goal in mind. If the work turns out to be different than they expected, they collaborate with the Product Owner to negotiate the scope of the Daily Backlog within the Micro-Sprint without affecting the Daily Goal.

Morning Stand-up & Planning

The Morning Stand-up & Planning is a 15-minute event for the HADD team to plan the day's work. Morning Stand-up & Planning answers the following:

- What PBI/Bug will we complete today?
- What are the Acceptance Criteria?
- How will we work together to accomplish this?
- What obstacles might we encounter?

The Morning Stand-up & Planning is held at the same time and place each day to reduce complexity. During the event, the HADD team plans work for the day, creating a Daily Backlog. The Daily Goal, selected Product Backlog items, and the plan for delivering them are together referred to as the Daily Backlog.

Duration: 15 minutes

Participants: PO, Dev, QA

Goal/Outcome: Clarify PBI/Bug and Acceptance Criteria (AC). Team commits to daily delivery. QA begins defining test cases.

Development Phase

The Development Phase is the core working period of the Micro-Sprint, spanning approximately 7 hours. During this phase, the HADD team works concurrently to develop, test, and validate the day's increment.

Duration: ~7 hours

Participants: Dev (AI), Architect, QA

Activities:

- AI-Augmented Development: Dev/AI Engineers leverage AI agents for code generation, following architectural guidelines
- Concurrent Coding: Multiple team members work in parallel on different aspects of the increment
- Unit Testing: Automated tests are written and executed continuously
- Same-Day Peer Review: Code is reviewed by another Dev/Architect as it's developed
- Architectural Review: Architect validates code against non-functional requirements
- Immediate QA Testing: QA tests code as soon as it becomes available
- Frequent Commits: Code is pushed frequently to Dev branch pipeline

Goal/Outcome: Concurrent coding, unit testing, and same-day peer/architectural review. Code pushed frequently to Dev branch pipeline. QA tests immediately upon availability.

Pre-Delivery Meeting

The Pre-Delivery Meeting is the final event of the Micro-Sprint. It is a 30-minute event for the HADD team to present the results of their work to key stakeholders and progress toward the Product Goal.

During the event, the HADD team and stakeholders review what was accomplished in the Micro-Sprint and what has changed in their environment. Based on this information, attendees collaborate on what to do next. The Product Backlog may also be adjusted to meet new opportunities.

The Pre-Delivery Meeting is a working session and the HADD team should avoid limiting it to demonstrations only.

Duration: 30 minutes

Participants: PO, Dev, QA

Activities:

- Final review of the day's increment
- PO acceptance of validated PBI
- QA manages secure push to Staging/QA environment
- Team showcases working functionality
- Stakeholder feedback collection

Goal/Outcome: Final review. PO accepts validated PBI. QA manages the secure push to the Staging/QA environment.

HADD Artifacts

HADD's artifacts represent work or value. They are designed to maximize transparency of key information. Thus, everyone inspecting them has the same basis for adaptation.

Product Backlog

The Product Backlog is an emergent, ordered list of what is needed to improve the product. It is the single source of work undertaken by the HADD team.

Product Backlog items that can be done by the HADD team within one Micro-Sprint are deemed ready for selection in a Morning Stand-up & Planning event. They usually acquire this degree of transparency after refining activities. Product Backlog refinement is the act of breaking down and further defining Product Backlog items into smaller, more precise items. This is an ongoing activity to add details, such as a description, order, and size. Attributes often vary with the domain of work.

The Dev/AI Engineers who will be doing the work are responsible for the sizing. The Product Owner may influence the Dev/AI Engineers by helping them understand and select trade-offs.

Commitment: Product Goal

The Product Goal describes a future state of the product which can serve as a target for the HADD team to plan against. The Product Goal is in the Product Backlog. The rest of the Product Backlog emerges to define "what" will fulfill the Product Goal.

A product is a vehicle to deliver value. It has a clear boundary, known stakeholders, well-defined users or customers. A product could be a service, a physical product, or something more abstract.

The Product Goal is the long-term objective for the HADD team. They must fulfill (or abandon) one objective before taking on the next.

Daily Backlog

The Daily Backlog is composed of the Daily Goal (why), the Product Backlog items selected for the Micro-Sprint (what), as well as an actionable plan for delivering the Increment (how).

The Daily Backlog is a plan by and for the Dev/AI Engineers. It is a highly visible, real-time picture of the work that the Dev/AI Engineers plan to accomplish during the Micro-Sprint in order to achieve the Daily Goal.

The Daily Backlog is updated throughout the Micro-Sprint as more is learned. It should have enough detail that they can inspect their progress in the Pre-Delivery Meeting.

Commitment: Daily Goal

The Daily Goal is the single objective for the Micro-Sprint. While the Product Backlog items selected for the Micro-Sprint will likely deliver increments of functionality, the Daily Goal gives coherence to the work, focusing the HADD team on a larger purpose. The Daily Goal is created during the Morning Stand-up & Planning event and added to the Daily Backlog.

Increment

An Increment is a concrete stepping stone toward the Product Goal. Each Increment is additive to all prior Increments and thoroughly verified, ensuring that all Increments work together.

In order to provide value, the Increment must be usable. Each Increment is delivered to the Staging environment at the end of each workday, where it can be inspected by stakeholders and the Product Owner.

Multiple Increments may be created within a Micro-Sprint. The sum of the Increments is presented at the Pre-Delivery Meeting, thus supporting empiricism. However, an Increment may be delivered to stakeholders prior to the end of the Micro-Sprint. The Micro-Sprint should end whenever the Daily Goal is met.

Work cannot be considered part of an Increment unless it meets the Definition of Done.

Commitment: Definition of Done

The Definition of Done is a formal description of the state of the Increment when it meets the quality measures required for the product.

The moment a Product Backlog item meets the Definition of Done, an Increment is born.

The Definition of Done creates transparency by providing everyone a shared understanding of what work was completed as part of the Increment. If a Product Backlog item does not meet the Definition of Done, it cannot be released or even presented at the Pre-Delivery Meeting. Instead, it returns to the Product Backlog for future consideration.

If the Definition of Done for an increment is part of the standards of the organization, all HADD teams must follow it as a minimum. If it is not an organizational standard, the HADD team must create a Definition of Done appropriate for the product.

The HADD team members must have a shared understanding of what it means for work to be complete, to ensure transparency. This is the definition of "Done" for the HADD team and is used to assess when work is complete on the product Increment.

Definition of Done (DoD) - The Daily Gate

This checklist must be met at the Pre-Delivery Meeting for the PBI/Bug to be deployed to Staging.

1. Code Quality & AI Validation

- **Code Complete:** Code is merged to the Dev Branch.
- **AI Code Analysis Passed:** Code passes automated/AI checks for Style, Linting, and minimum Unit Test Coverage ($\geq 85\%$).
- **Complexity & Debt Index Check:** AI-driven static analysis confirms no significant increase in technical debt or complexity.
- **Same-Day Peer & Architectural Review:** Code is formally reviewed and approved by another Dev/Architect, ensuring security and architectural alignment.

2. Testing & Functional Validation

- **Automated Tests Passed:** All existing and new Unit/Integration tests pass in the CI/CD pipeline.
- **QA Validation:** QA has successfully executed functional and exploratory tests, confirming all Acceptance Criteria are met.
- **No High/Critical Bugs:** The QA team confirms zero High or Critical bugs remain open for the item.

3. Deployment & Product Acceptance

- **Product Owner Acceptance:** PO visually verifies and accepts the working increment.
- **Deployment Ready:** Code is successfully deployed to the Staging/QA Environment.
- **Observability Instrumented:** Logging, monitoring, and tracing are confirmed in place.

HADD Success Metrics

Metrics focus on balancing Daily Flow (Speed) with Quality (Stability).

Speed Metrics (Flow)

- **Daily Delivery Frequency:** Number of unique PBIs/Bugs deployed to Staging per day.
- **Daily Cycle Time (Lead Time for Change):** Average time from Morning Stand-up (work start) to code being live on Staging (DoD met).

Quality Metrics (Stability)

- **Change Failure Rate:** Percentage of deployments to Staging/Production causing a failure or requiring rollback.
- **AI-Introduced Complexity/Debt Index:** Trend analysis of code complexity and security warnings in AI-assisted code blocks (used for tuning AI agents).
- **Time to Recover Service (MTTR):** Time to restore service after a Production failure (must be low due to small daily batch size).

These metrics should be tracked daily and reviewed during the Pre-Delivery Meeting to identify trends and opportunities for improvement.

End Note

HADD is free and offered in this Guide. The HADD roles, artifacts, events, and rules are immutable. Although implementing only parts of HADD is possible, the result is not HADD. HADD exists only in its entirety and functions well as a container for other techniques, methodologies, and practices.

Acknowledgements

HADD is built upon the foundational principles of Agile and Scrum, adapted for the Age of AI. We acknowledge the contributions of the Agile community and the pioneers of iterative and incremental development.

People

HADD was developed to address the accelerating pace of software development in the Age of AI. It represents a modernization of Agile principles, leveraging AI Agents and Cloud Architecture Expertise to compress traditional development cycles into daily micro-sprints.

HADD Guide History

The Hyper-Accelerated Daily Delivery (HADD) Guide was first published in 2024 as a response to the need for faster, more responsive software development methodologies that leverage artificial intelligence and cloud-native architectures.

HADD compresses the traditional two-week sprint into a single 8-hour daily micro-sprint, enabling organizations to deliver Minimum Viable Products (MVPs) or critical bug fixes within a 5-business-day weekly cycle.

© 2024 The Hyper-Accelerated Daily Delivery Guide

This guide may be freely copied and distributed in any form, provided that it is not sold or used for commercial purposes.