A black and white logo

AI-generated content may be incorrect.

**Welcome to Armada**

This document is a comprehensive guide to help you understand our workflows, methodologies, tools, and best practices. Our goal is to ensure a smooth transition into your role so that you can contribute effectively from day one.

**1. Company Overview**

Armada is committed to cutting-edge AI-driven solutions, focusing on industrial safety, real-time video analytics, and edge computing. Our projects span multiple domains, including AI model development, cloud computing, and software engineering.

**2. Engineering Team Structure**

At Armada, the engineering team is structured into cross-functional teams that collaborate on various projects. Key roles include:

* **Software Engineers** – Develop and maintain backend/frontend systems.
* **AI Engineers** – Build AI models and data pipelines.
* **Engineering Managers -** Lead and mentor engineering teams, oversee technical execution, drive innovation, and ensure delivery of high-quality software solutions while aligning with business objectives.
* **Product Managers** – Drive product strategy, define requirements and align teams to business goals.
* **Technical Program Managers (TPMs)** – Lead project execution, coordinate cross-functional teams, and ensure alignment with strategic objectives.

**3. Understanding Our Development Framework**

**Agile and Scrum Methodologies**

At Armada, we follow Agile and Scrum methodologies to ensure an iterative and efficient development process.

* **Sprint Cycles**: Each sprint lasts two weeks and focuses on completing defined features, improvements, and fixes.
* **Epics and Stories**: Work is broken down into Epics, Features, and Stories to ensure systematic tracking and execution.
* **Daily Stand-ups**: Quick daily meetings to track progress, address blockers, and realign team priorities.
* **Bi-weekly Sprint Reviews**: Evaluate achievements, challenges, and refine the roadmap.

**Sprint Workflow**

[https://dev.azure.com/Armadasystems/Commander/\_sprints/directory](https://dev.azure.com/armadasystems/Commander/_sprints/directory)

1. **Planning**: Identifying sprint objectives and assigning tasks.
2. **Development**: Coding, testing, and integrating features.
3. **Testing**: Unit, integration, and system testing.
4. **Deployment**: CI/CD pipeline deployment to Dev, Staging, and Production.
5. **Retrospective**: Review of what went well and areas for improvement.

A screenshot of a computer

AI-generated content may be incorrect.

**An Epic at Armada includes:**

[https://dev.azure.com/Armadasystems/Commander/\_backlogs/itrellis.portfolio-plus-plus.portfolio-pp-workitem-feature-timeline/Commander%20Team/Features](https://dev.azure.com/armadasystems/Commander/_backlogs/itrellis.portfolio-plus-plus.portfolio-pp-workitem-feature-timeline/Commander%20Team/Features)

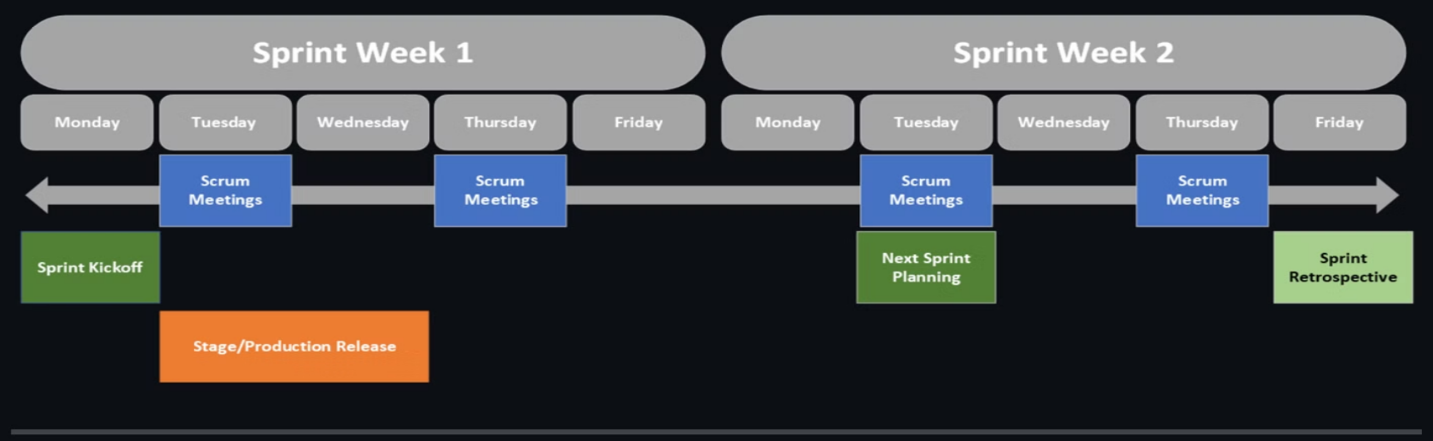
1. An Epic is a larger body of work that is broken down into features and tasks and often spans multiple sprints. It represents a high-level business or technical goal.
2. For instance, an Epic at Armada could be:  
    "Enhancing OpsSafety AI for Industrial Safety Monitoring"  
   This Epic could include several sprints, covering:
   1. Developing advanced video action detection models (Sprint 1)
   2. Integrating real-time alerts with the OpsSafety dashboard (Sprint 2)
   3. Improving UI/UX for reporting safety violations (Sprint 3)
3. By breaking large initiatives into Epics and Features, Armada ensures systematic and efficient development across its product lines.

**4. Engineering Workflow & Best Practices**

### **Engineering Epic Flow:**

* **Initiative (Top Level)**
* A large strategic goal or business objective that the organization aims to achieve.
* Initiatives span across multiple releases and require coordinated efforts from multiple teams.
* **Epics (Large Features)**
* Each initiative is broken down into Epics, which are high-level features or large bodies of work that contribute to the initiative.
* At Armada, Epics would align with key AI functionalities, product features, or integrations.
* requirement, often written from a user perspective.
* **Sub features (Detailed Implementation)**
* Sub features are further split into Subtasks, which are specific development, testing, or operational tasks needed to complete a Story.
* At Armada AI, this could include coding AI models, creating APIs, testing integrations, or updating dashboards.

To ensure timely delivery and high-quality output, Armada employs a structured engineering schedule:



**Sprint Planning**

* This document is a bi-weekly update from Armada’s Product and Engineering teams, summarizing key achievements, challenges, and ongoing initiatives.
* The update provides insights into Armada’s technological progress and operational hurdles.

**Team Meetings**

* Quick meetings to assess progress and remove roadblocks.

At Armada, each team should determine the frequency of their meetings based on their needs. It is recommended to hold scrum or stand-up sessions approximately 2-3 times a week, but teams have the flexibility to choose a schedule that works best for them.

**Continuous Integration/Deployment (CI/CD)**

* Ensures automated testing and smooth deployment processes across Dev, Staging, and Production environments.

**5. Tools & Systems for Effective**

Armada leverages modern technology stacks to streamline project execution.

**Azure DevOps**

[https://dev.azure.com/Armadasystems/Commander/\_wiki/wikis/Commander.wiki/27/Getting-Started](https://dev.azure.com/armadasystems/Commander/_wiki/wikis/Commander.wiki/27/Getting-Started)

**Product Guide – Connect Overview Training Guide** [https://Armadaai.sharepoint.com/:w:/s/ArmadaOneDrive/Ef6alfVp4XlOgBne6d8nHp0BPXFnrKnB0iGkOiCQ99DnLQ?e=Qmw4Ex](https://armadaai.sharepoint.com/:w:/s/ArmadaOneDrive/Ef6alfVp4XlOgBne6d8nHp0BPXFnrKnB0iGkOiCQ99DnLQ?e=Qmw4Ex)

**End-to-End Release Checklist** [https://dev.azure.com/Armadasystems/Commander/\_wiki/wikis/Commander.wiki/239/End-to-End-Release-Checklist-(Dev-Stage-Prod](https://dev.azure.com/armadasystems/Commander/_wiki/wikis/Commander.wiki/239/End-to-End-Release-Checklist-(Dev-Stage-Prod))

* **Project Management**: Azure DevOps for backlog and sprint management.
* **Collaboration**: Microsoft M365 suite (Teams, Loop, and Planner).
* **Version Control**: GitHub for source code management and reviews.
* **Knowledge Management**: SharePoint and Azure DevOps Wiki for centralized documentation.

**6. Documentation & Knowledge Management**

Documentation is critical for product consistency and operational efficiency:

**Azure DevOps Wiki:** acentral knowledge hub for onboarding, project management, and technical documentation.

[https://dev.azure.com/Armadasystems/Commander/\_wiki/wikis/Commander.wiki/](https://dev.azure.com/armadasystems/Commander/_wiki/wikis/Commander.wiki/)

**New Product (Template)**

[https://Armadaai.sharepoint.com/:w:/r/sites/ArmadaOneDrive/\_layouts/15/Doc.aspx?sourcedoc=%7BE508085D-9F06-4EA3-B219-86B6FCC8B056%7D&file=NEW%20PRODUCT%20TEMPLATE.docx&action=default&mobileredirect=true](https://armadaai.sharepoint.com/:w:/r/sites/ArmadaOneDrive/_layouts/15/Doc.aspx?sourcedoc=%7BE508085D-9F06-4EA3-B219-86B6FCC8B056%7D&file=NEW%20PRODUCT%20TEMPLATE.docx&action=default&mobileredirect=true)

**Product Documentation - PRFAQs**

https://Armadaai.sharepoint.com/sites/ArmadaOneDrive/Shared%20Documents/Forms/AllItems.aspx? id=%2Fsites%2FArmadaOneDrive%2FShared%20Documents%2FTechnical%2FOperations%2FCustomer%20Experience%2F03%20%2D%20Product%20Documentation&p=true&ga=1

Other Documentation

* **Product Requirements Documents (PRD)**: Detailing features, specifications, and scope.
* **PRFAQs**: Frequently asked questions and answers for internal and external clarity.
* **AI Knowledgebase**: Centralized training material for AI Assistant
* **Connect Overview Document**