

# Software development methodologies

## Agile methodology

Imagine you are the project manager at a software company and you receive the following e-mail:

**Subject:** Project Proposal: Smart Inventory Tracking Dashboard

Dear Lead Project Engineer,

I am writing to propose a new development project that I believe will significantly reduce our current operational bottlenecks: a Smart Inventory Tracking Dashboard.

Currently, our team spends an excessive amount of time manually cross-referencing inventory logs, which leads to delays and occasional data discrepancies. I propose we develop a centralized, web-based dashboard that consolidates real-time inventory data from our existing databases into a single, highly visual interface.

The goal is to provide automated low-stock alerts, visual trend tracking, and seamless data exporting. This tool will directly empower the operations team to make faster, data-driven decisions without needing to wait for end-of-week manual reports.

To ensure this tool actually meets the day-to-day needs of the operations team and doesn't get bogged down in a rigid, lengthy development phase, the project must adhere to the following execution requirements:

- Iterative delivery: we will not wait for a single, final launch. Development will be broken down into two-week cycles. We must deliver a usable, bare-bones prototype (core tracking only) by the end of cycle one, adding more complex features (like predictive alerts) in subsequent cycles;
- Flexible prioritization: we will maintain a dynamic list of desired features. At the start of every two-week cycle, we will evaluate this list and select the most valuable tasks to work on, allowing us to pivot easily if business priorities shift;
- Continuous stakeholder feedback: at the end of every cycle, we will hold a mandatory review session with a core group of operations staff to demonstrate the latest working software, gather their immediate feedback, and course-correct our next steps based on their actual usage;
- Daily alignment: the development group will hold brief, 15-minute syncs every morning to highlight what was accomplished yesterday, what is planned for today, and immediately flag any blockers;
- Continuous Integration (CI): the architecture must support frequent, incremental updates to the live environment so that new features and fixes can be pushed to the users as soon as a cycle is completed and tested.

I am confident this approach will keep the project highly adaptable, keep the team motivated, and ensure we build exactly what the operations team needs.

Best regards,

Director of Marketing  
Elias-Clarke Publications

Having made the appropriate considerations and additional assumptions, create a rough structure of the life cycle phases of the assigned project by implementing the Agile methodology.

Grading rubric:

- Phase Identification and Logical Sequence (10 marks);
- Application of Project Context (10 marks);
- "Gate" and Documentation Focus (5 marks);
- Handling of Assumptions (5 Marks).

**Total marks: 30**