

development standards and protocols: initial stage guide

Proposed by,

rithishsamm

**SUMMARY:**

This document is about developers helping the administrators to help them manage the application after delivery by having a streamlined workflow with the ongoing projects that are getting developed under our organization. This document follows **application development standards and protocols that help the administrators to micromanage operationally intensive mundane tasks** that are getting delivered to the operations from the developers.

Here, we are getting started with our motion going up towards developing projects where the organization is in need of a workflow that has to be and get streamlined to follow such practices and progress on the same following such standards as we grow as an organization shifting from doing all these manual to automation.

This document will guide you towards such a path and give you a better understanding about managing projects, fostering collaboration, and embracing continuous improvement by feedback loops for better enhancement of the projects that we all do.

Sooner or later, we will be progressing towards a systematic workflow on a larger scale that includes all the defined stages of the application that have to get passed, such as development, build and test, staging, and production, which aligns closely with the DevOps with Agile methodologies.

**SYNOPSIS**:

1. Introduction
2. Goals, Specifications and Milestones
3. Roadmap
4. Problem Statements
5. Solutions
6. Pros and Cons
7. Pay off
8. Conclusion

**GOALS:**

**TLDR:**

To help developers stay on course.

### **Project Management Goals in Agile Software Development**

### **1. Enhance Collaboration Across Teams**

* **Goal**: Foster a collaborative environment where cross-functional teams (development, QA, design, etc.) work closely together.
* **Action Items**:
  + Implement regular cross-team stand-ups and sync meetings.
  + Use collaborative tools (e.g., Jira, Confluence, Slack) for transparent communication and documentation.
  + Encourage knowledge-sharing sessions to enhance team skills and understanding.

#### **2. Optimize Sprint Planning and Execution**

* **Goal**: Improve the efficiency of sprint planning and execution to ensure timely delivery of features.
* **Action Items**:
  + Define clear acceptance criteria for all user stories.
  + Prioritize the backlog based on business value and team capacity.
  + Review and adjust sprint goals based on team performance and stakeholder feedback.

#### **3. Implement Continuous Integration and Continuous Deployment (CI/CD)**

* **Goal**: Establish a robust CI/CD pipeline to automate testing and deployment, ensuring high-quality releases.
* **Action Items**:
  + Integrate automated testing tools in the development pipeline.
  + Set up deployment automation to reduce manual errors and speed up release cycles.
  + Monitor the CI/CD process continuously for bottlenecks and areas of improvement.

#### **4. Ensure High-Quality Deliverables**

* **Goal**: Maintain high standards of quality throughout the development process.
* **Action Items**:
  + Conduct regular code reviews and pair programming sessions.
  + Establish a definition of "done" that includes testing, documentation, and code quality.
  + Encourage teams to adopt best practices and coding standards (e.g., clean code, SOLID principles).

#### **5. Foster a Culture of Continuous Improvement**

* **Goal**: Create an environment where teams regularly reflect on their processes and outcomes.
* **Action Items**:
  + Conduct sprint retrospectives to gather feedback and identify improvement areas.
  + Implement action items from retrospectives in subsequent sprints.
  + Encourage teams to experiment with new practices and tools to enhance productivity.

#### **6. Enhance Stakeholder Engagement and Communication**

* **Goal**: Keep stakeholders informed and engaged throughout the project lifecycle.
* **Action Items**:
  + Schedule regular demo sessions at the end of each sprint to showcase progress.
  + Maintain a transparent backlog accessible to stakeholders for tracking priorities and progress.
  + Create feedback loops to incorporate stakeholder input into planning and execution.

#### **7. Manage Risks and Dependencies Effectively**

* **Goal**: Proactively identify and manage risks and dependencies across teams.
* **Action Items**:
  + Establish a risk management framework to identify, assess, and mitigate risks.
  + Use dependency management tools to track and address inter-team dependencies.
  + Conduct regular risk assessments as part of sprint planning.

#### **8. Measure Performance and Outcomes**

* **Goal**: Use metrics to assess team performance and project health.
* **Action Items**:
  + Define key performance indicators (KPIs) such as velocity, burn-down charts, and lead time.
  + Regularly review and analyze metrics to inform decision-making.
  + Create dashboards for real-time visibility into project status and team performance.

#### **9. Embrace Agile Scaling Practices**

* **Goal**: Scale Agile practices effectively across multiple teams.
* **Action Items**:
  + Implement frameworks like SAFe (scaled Agile Framework) or LeSS (large scale scrum) to coordinate efforts across teams.
  + Foster a community of practice groups for shared learning and best practices.
  + Ensure alignment of team goals with organizational objectives.

#### **10. Deliver Value to Customers Continuously**

* **Goal**: Focus on delivering incremental value to customers throughout the development process.
* **Action Items**:
  1. Prioritize features based on customer feedback and market needs.
  2. Use iterative development to release MVPs (minimum viable products) for quicker validation.
  3. Regularly gather and analyze customer feedback to inform future iterations.

# **SPECIFICATIONS:**

. **1. Collaboration Framework**

* **Tools**: Implement collaborative tools (e.g., Jira, Confluence, Slack).
* **Meetings**: Schedule regular cross-team stand-ups and sync meetings.
* **Knowledge Sharing**: Facilitate knowledge-sharing sessions to enhance skills.

#### **2. Sprint Planning and Execution**

* **Backlog management**: prioritize user stories based on business value and team capacity.
* **Acceptance Criteria**: Define clear acceptance criteria for all user stories.
* **Goal Review**: Regularly review and adjust sprint goals based on performance.

#### **3. CI/CD Implementation**

* **Automation**: Integrate automated testing and deployment in the CI/CD pipeline.
* **Monitoring**: Continuously monitor the CI/CD process for bottlenecks.

#### **4. Quality Assurance Standards**

* **Code Reviews**: Conduct regular code reviews and encourage pair programming.
* **Definition of Done**: Establish a comprehensive definition of "done."
* **Best Practices**: Promote adherence to coding standards (e.g., clean code).

#### **5. Continuous Improvement Culture**

* **Retrospectives**: Conduct sprint retrospectives to identify improvement areas.
* **Action Implementation**: Integrate retrospective action items into future sprints.
* **Experimentation**: Encourage teams to test new practices and tools.

#### **6. Stakeholder Engagement**

* **Regular Demos**: Schedule demos at the end of each sprint to showcase progress.
* **Transparent Backlog**: Maintain an accessible backlog for stakeholder visibility.
* **Feedback Loops**: Establish mechanisms for incorporating stakeholder feedback.

#### **7. Risk and Dependency Management**

* **Risk Framework**: Implement a risk management framework for proactive assessment.
* **Dependency tracking**: utilize tools to manage inter-team dependencies.
* **Regular Assessments**: Conduct regular risk assessments during planning.

#### **8. Performance Metrics**

* **KPIs**: Define KPIs such as velocity, burn-down charts, and lead time.
* **Regular reviews**: metrics to inform decision-making.
* **Dashboards**: Create real-time dashboards for project visibility.

#### **9. Agile Scaling Practices**

* **Frameworks**: Implement scaling frameworks like SAFe or LeSS.
* **Communities of Practice**: Foster groups for shared learning and best practices.
* **Goal Alignment**: Ensure team goals align with organizational objectives.

#### **10. Customer Value Delivery**

* **Feature Prioritization**: Focus on features based on customer feedback and needs.
* **Iterative Development**: Release MVPs for quicker validation and feedback.
* **Feedback Analysis**: Regularly analyze customer feedback for future iterations.

# **MILESTONES:**