**1. Introduction**

This system is intended to streamline **project delivery, release management, and long-term maintenance** by integrating **Agile project management, DevOps automation, and cloud-agnostic deployment**. It must support **collaborative teams, scalable workflows, and compliance requirements**.

**2. Objectives**

* Provide a **single platform** for project planning, delivery, and support.
* Enable **Agile ways of working** with Scrum/Kanban boards, sprint tracking, velocity insights.
* Support **DevOps practices**: CI/CD, test automation, monitoring, infra-as-code.
* Ensure **vendor-neutral cloud support** (deployable to AWS, Azure, GCP, on-prem).
* Integrate seamlessly with **JIRA, GitHub/GitLab/Bitbucket, Jenkins/GitHub Actions/Azure DevOps**, etc.
* Improve **traceability** from requirements → development → deployment → maintenance.

**3. Core Functional Requirements**

**3.1 Project & Agile Management**

* Backlog management (epics, user stories, tasks, bugs).
* Sprint planning & tracking (capacity planning, burndown).
* Kanban workflows for continuous delivery teams.
* Integration with JIRA for **real-time synchronization**.
* Role-based dashboards (Project Manager, Product Owner, Developer, QA, Ops).
* Reporting: Velocity charts, cycle time, lead time, throughput.

**3.2 DevOps & CI/CD**

* Integration with CI/CD pipelines (Jenkins, GitHub Actions, GitLab CI, Azure DevOps).
* Automated build, test, deploy workflows.
* Environment-aware deployments (dev, test, staging, prod).
* Container orchestration (Docker, Kubernetes).
* Artifact management (Nexus, JFrog Artifactory).
* Monitoring integration (Prometheus, Grafana, ELK).

**3.3 Cloud Agnostic Deployment**

* Support for AWS, Azure, GCP, On-premises via **Terraform/Helm/Ansible**.
* Pluggable deployment templates.
* Abstracted cloud service definitions (storage, compute, networking).
* Automated scaling and cost-optimization policies.

**3.4 Maintenance & Operations**

* Incident management integration (JIRA Service Management, ServiceNow, PagerDuty).
* Automated SLA tracking.
* Knowledge base & documentation repository.
* Change management workflows.
* Post-release health checks & automated rollbacks.

**4. Non-Functional Requirements**

* **Scalability**: Handle 1000+ concurrent users.
* **Performance**: <2s latency for dashboard updates.
* **Security**: RBAC, SSO (SAML/OAuth2), audit logging, data encryption (at rest & transit).
* **Compliance**: GDPR, SOC2, ISO27001 alignment.
* **Reliability**: 99.9% uptime SLA.
* **Extensibility**: API-first, plugin support.

**5. Integrations**

* **JIRA**: Two-way sync for issues, sprints, epics.
* **Git Repos**: GitHub, GitLab, Bitbucket.
* **CI/CD**: Jenkins, GitHub Actions, GitLab CI, Azure DevOps.
* **Monitoring/Logging**: Prometheus, Grafana, ELK.
* **Communication**: Slack, MS Teams.
* **Ticketing**: JIRA Service Management, ServiceNow.

**6. High-Level Architecture**

* **Frontend**: Web app (React/Angular/Vue).
* **Backend**: Microservices (Node.js/Java/Spring Boot/FastAPI).
* **APIs**: REST/GraphQL for integrations.
* **Data Layer**: PostgreSQL/NoSQL (MongoDB).
* **Deployment**: Kubernetes clusters (Helm/Terraform).
* **Security Layer**: OAuth2, SSO, RBAC.

**7. Roadmap (Phased Delivery)**

* **Phase 1**: Project & Agile management core + JIRA sync.
* **Phase 2**: CI/CD integration + cloud-agnostic deployments.
* **Phase 3**: Maintenance/Operations + monitoring integration.
* **Phase 4**: Advanced analytics, AI-based forecasting, cost optimization.