

Non-Functional Requirements

1. Performance Requirements

1.1. Dashboard Load Time: All dashboard pages must load within 2 seconds under normal network conditions.

1.2. Data Processing: DORA metrics calculations must complete within 5 seconds of data ingestion from integrated tools.

1.3. API Response: API endpoints must respond within 500ms for 95% of requests under typical load.

2. Scalability Requirements

2.1. User Capacity: Support up to 10,000 concurrent users with no degradation in performance.

2.2. Repository Handling: Process data from 500 concurrent repositories simultaneously.

3. Reliability & Availability

3.1. Uptime: Achieve 99.9% annual uptime, excluding scheduled maintenance.

3.2. Real-Time Updates: Ensure metric updates reflect in the dashboard with a maximum latency of 30 seconds after data ingestion.

4. Security Requirements

4.1. Access Control: Implement role-based access control (RBAC) for Managers and Team Members.

4.2. Audits: Conduct security audits and vulnerability assessments.

5. Usability & Accessibility

5.2. Accessibility Compliance: Ensure the dashboard meets WCAG 2.1 AA accessibility standards.

5.3. Responsive Design: Optimize the interface for seamless use on desktop and mobile devices.

6. Compatibility & Integration

6.1. Tool Integration: Support integration with GitHub, GitLab, and Jira via their latest public APIs (2025 versions).

6.2. Browser Support: Ensure compatibility with Chrome, Firefox, Safari, and Edge (latest two versions).

7. Maintainability & Support

7.1. Microservices Updates: Allow independent deployment of microservices with ≤ 1 hour downtime per component.

7.2. Documentation: Provide detailed API documentation and architectural guidelines.

8. Legal & Compliance

8.1. Data Privacy: Comply with GDPR and POPIA regulations for user data protection.

8.2. User Consent: Obtain explicit user consent for data collection and processing during sign-up.

9. Architectural Patterns

9.1. Microservices Architecture: Design the system using loosely coupled services for independent scaling and deployment.

9.2. API-First Design: Ensure all integrations and internal components expose well-documented APIs for extensibility.

9.3. Event-Driven Processing: Use asynchronous event handling for real-time metric updates and alerts.

9.4. Scalable Data Storage: Employ cloud-native databases (e.g., Firebase, Supabase) to handle growing data volumes.

10. Constraints

10.1. GitHub Dependency: The system must prioritize GitHub for sourcing metrics (e.g., commits, pull requests, CI/CD data).

10.2. Dashboard Functionality: Dashboards must display at least 4 DORA metrics and support role-based privacy controls.

10.3. Hosting Limitations: Deploy only on approved platforms (Vercel, Supabase, Firebase) to ensure cost efficiency.

10.4. Real-Time Data: Metrics must reflect updates within 30 seconds of data ingestion.

10.5. Compliance Deadlines: Adhere to GDPR and POPIA requirements from initial deployment.