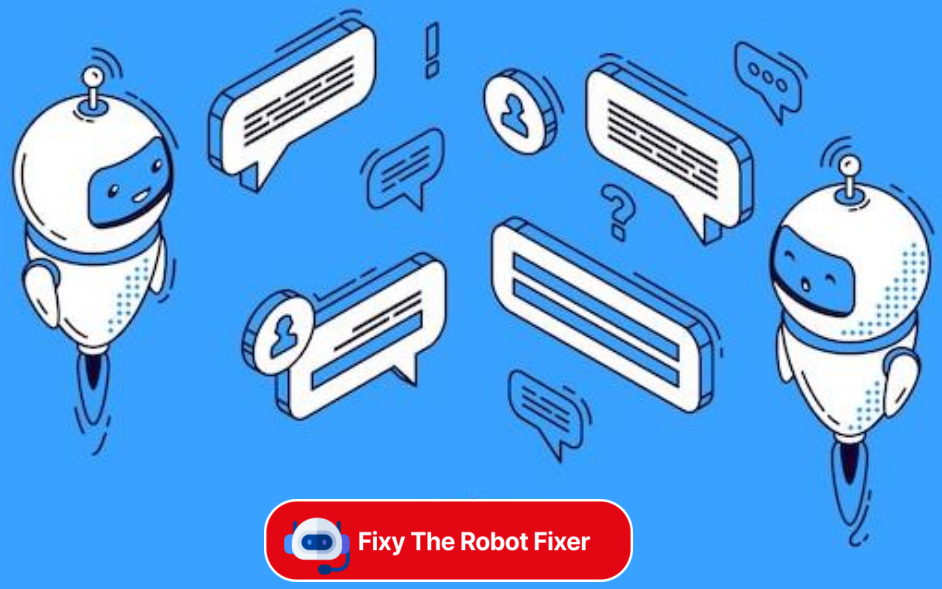
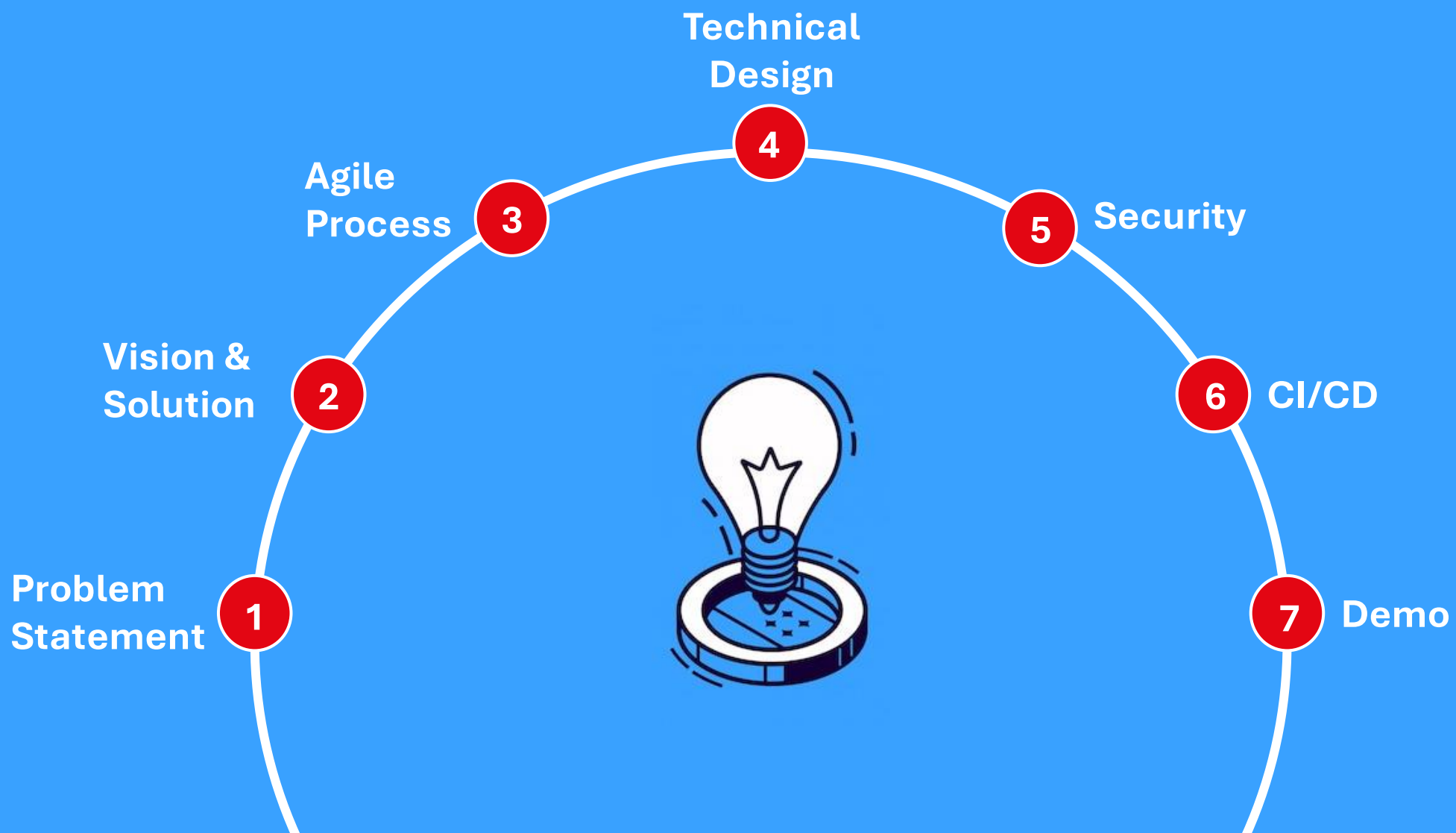
Centralized Ticketing System for OCBC



Team 2:
Chan Horng Chyi, Deepakrishna S/o Tangarajan, Elicia Loh Qi Yu,
Marion Tan Yan Tong, Tan Si Ning, Tan Poh Choo Christina, Tharik Nezar S/O AKBAR

Content.


Technical Design
4
Agile Process
3
5
Security
Vision & Solution
2
6
CI/CD
Problem Statement
1
7
Demo

Problem Statement

Bank expansion and hybrid work adoption create **increased IT support demands**

Fragmented IT support system causes **delays in issue resolution**

Long wait times (48-72 hours) due to multiple disconnected support channels.

Repetitive IT requests (e.g., password resets, software installations) overwhelm support staff

Employee satisfaction with IT support dropped by 25%, leading to low morale

Vision

Enhance employee productivity and IT efficiency with an AI-powered ticketing system that **automates issue resolution**, **improves response times**, and **ensures security**

Solution /Outcome

1. **Centralized ticketing system**
   - With AI-powered chatbot to streamline IT support and improve user experience

2. **Automate simple issue resolution**
   - By enabling the AI chatbot to handle common IT requests, reducing dependency on human support staff

3. **Automate ticket categorisation and prioritisation**
   - To ensure critical issues receive faster attention and reducing resolution time

4. **Integrate with Microsoft Teams and collaboration tools**
   - For seamless real-time support and efficient communication

5. **Adopt microservices architecture**
   - To enhance system scalability, security, and performance, ensuring seamless integration with banking infrastructure

6. **Enable continuous monitoring and analytics**
   - To track IT support trends, optimize chatbot performance, and enhance overall IT efficiency

Our Agile Journey

Meet the Scrum Team

PRODUCT OWNER

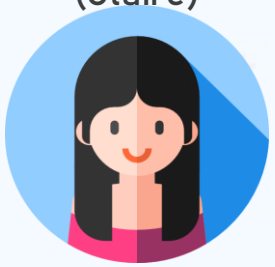
Christina Tan

SCRUM MASTER

Deepakrishna

UI/UX LEAD

Tan Si Ning (Claire)



TEST LEAD

Elicia Loh

DEVOPS LEAD

Tharik

DEVOPS LEAD

Marion Tan

SECURITY LEAD

Chan Horng Chyi (Francis)

The Scrum Team in Action

8 Epics:

AI Chatbot	Ticket Creation	Ticket Management	Ticket Tracking
Dashboard Analytics	Enhance Search Function	Multi-Channel Integration	Ticket Progress Notifications

Total User Stories:

49

Sprint 0

Key Objective:

Set up required IT infrastructure

User Stories:

4

Sprint 1

Key Objective:

To integrate the chatbot's core functionality into the staff portal

User Stories:

13

Sprint 2

Key Objective:

To develop a ticketing system dashboard for IT to view and update assigned tickets

User Stories:

10

Sprint 3

Key Objective:

To allow users to escalate tickets and track ticket statuses

User Stories:

9

Sprint 4

Key Objective:

To implement search function for IT support staff and artifact upload during ticket creation for users

User Stories:

9

Sprint 5

Key Objective:

To implement notification system for users and IT support staff, and multi-channel integration

User Stories:

8

Fixy The Robot Fixer

Total Story Points:

153

Average Sprint Velocity:

30

Summary of Sprints Burndown

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
Storypoints at the start of iteration	153	120	90	60	30
Completed during iteration	33	30	30	30	30
Changed estimates	0	0	0	0	0
Storypoints from new stories	0	0	0	0	0
Storypoints at end	**120**	**90**	**60**	**30**	**0**

Steady Progress:

The team's commitment and collaboration are evident in the burndown chart, which steadily declines over five sprints, **closely mirroring the ideal burn rate**.

Predictable Burn Rate:

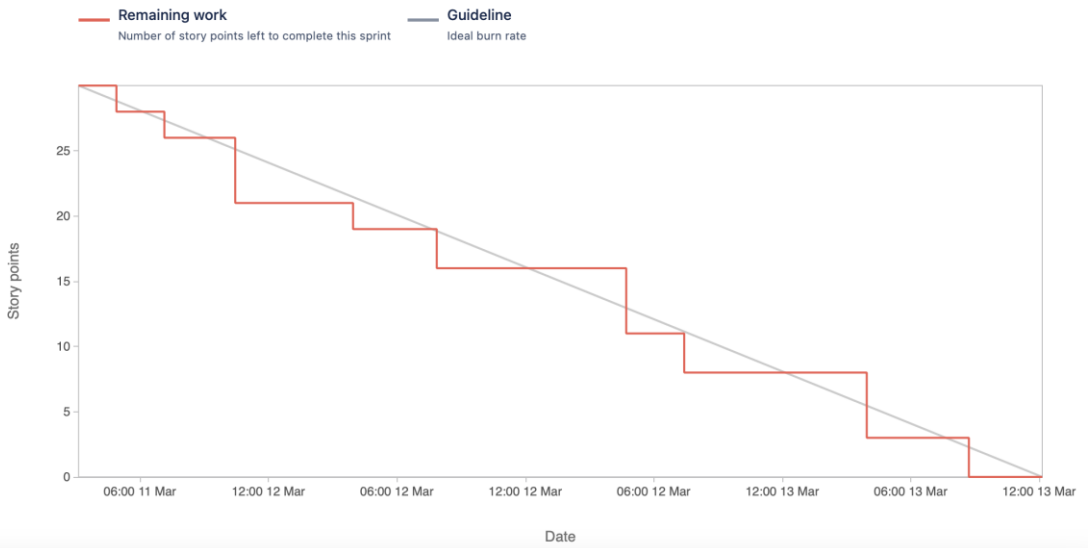
The team consistently delivers on their commitments, **maintaining a reliable pace**. Their steady performance showcases not just technical efficiency but also strong teamwork and clear communication.

Completion of All Stories:

At the end of the 5th sprint, the team celebrates a significant achievement—**every story has been completed!** This milestone reflects their dedication and ability to stay on track despite the tight deadlines.

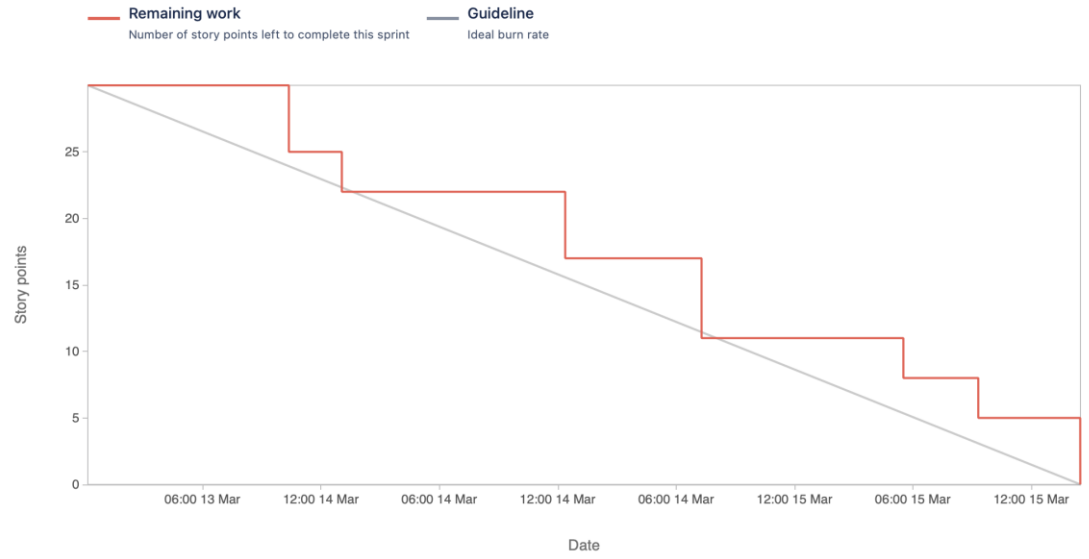
Sprint 3 Burndown chart

(March 11th, 2025 - March 13th, 2025)



Sprint 4 Burndown chart

(March 13th, 2025 - March 15th, 2025)



Sprint Retrospective

What went well?

"The sprint planning was clear and efficient"
- It helps clarify priorities, responsibilities, and expectations, streamlining sprint execution.

"We effectively collaborated as team"
- The team appreciated smooth communication, strong engagement, and effective knowledge sharing.

"We met our sprint goals"
- The team remained focused and productive, successfully achieving the goals set at the start of the sprint.

What can be better?

"Address blockers or dependencies earlier"
- Sooner the identification and resolution of blockers and dependencies would maintain the momentum of the team and avoidance of delay.

"Improve test practices"
- The need to refine testing practices such as increasing test coverage, automating tests, and conducting earlier validations to catch issues sooner.

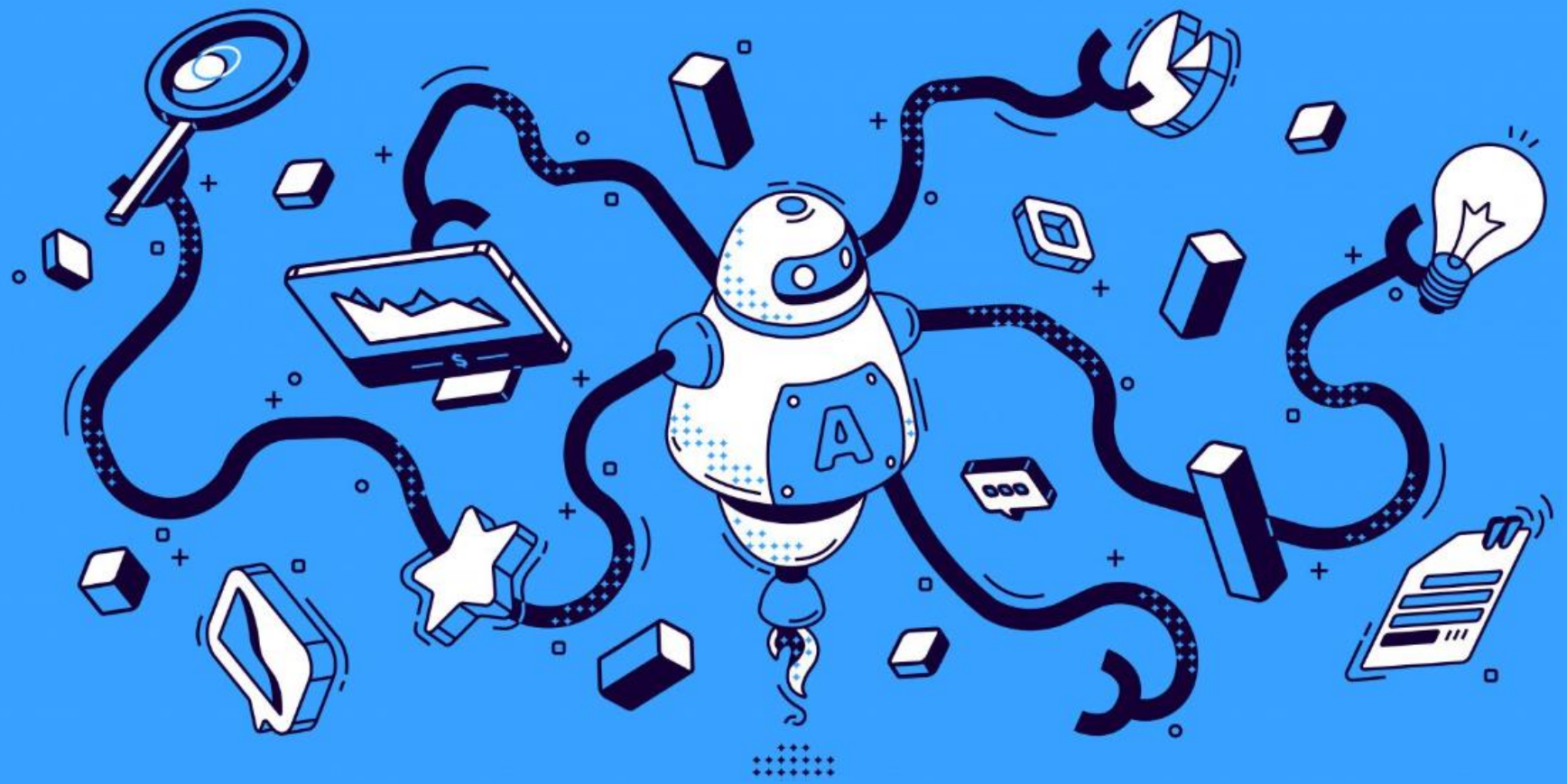
What should we try doing next?

"Prioritise resolving technical debt"
- Allocate time to the next sprint to address technical debt such as outdated code or shortcuts, improve code quality and maintainability.

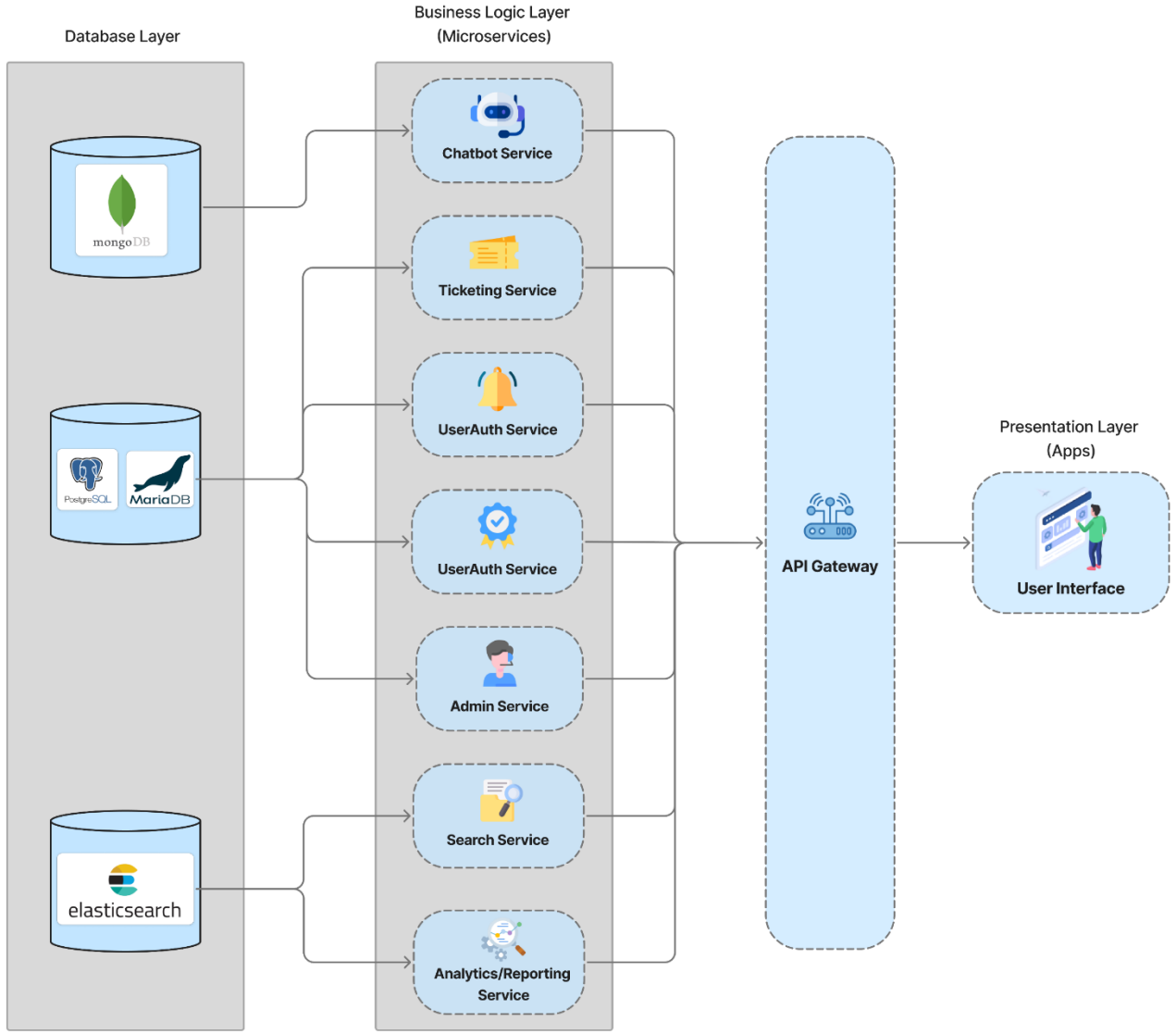
"Experiment with new tools for improved version control practices"
- Explore better branching strategies, stricter commit conventions and tools like Git hooks to streamline collaboration and reduce merge conflicts.

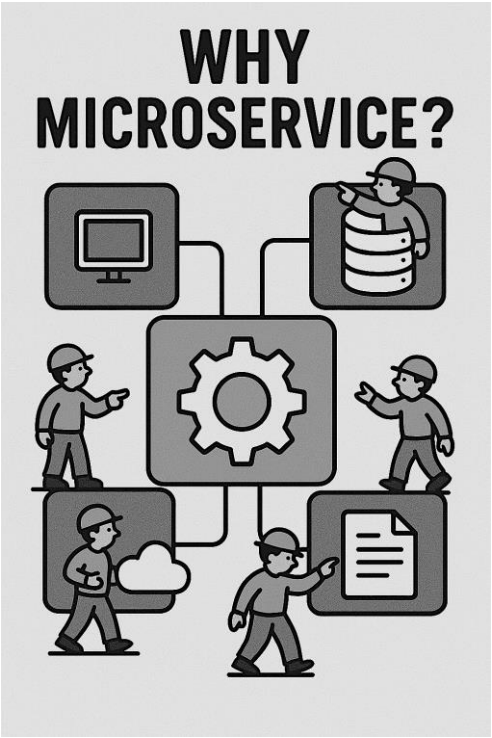
"Refine our Definition of Done"
- Expand the definition to ensure consistency in task completion and quality standard across the sprint.

Technical Design



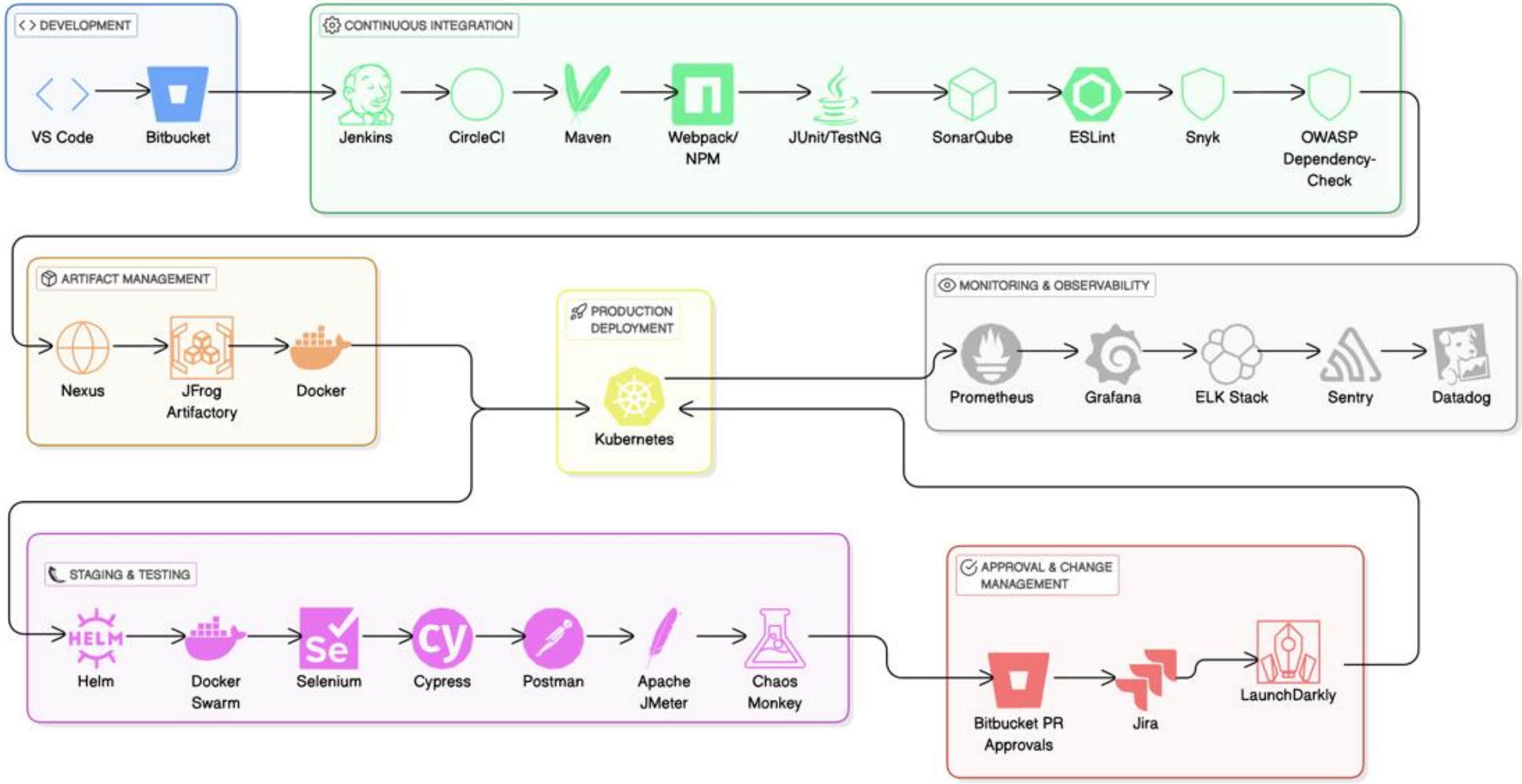
Technical Architecture:





- Modularity & Maintainability
- Scalability
- Technology Flexibility
- Fault Isolation
- Independent Deployment & CI/CD

Software Development & Deployment Workflow

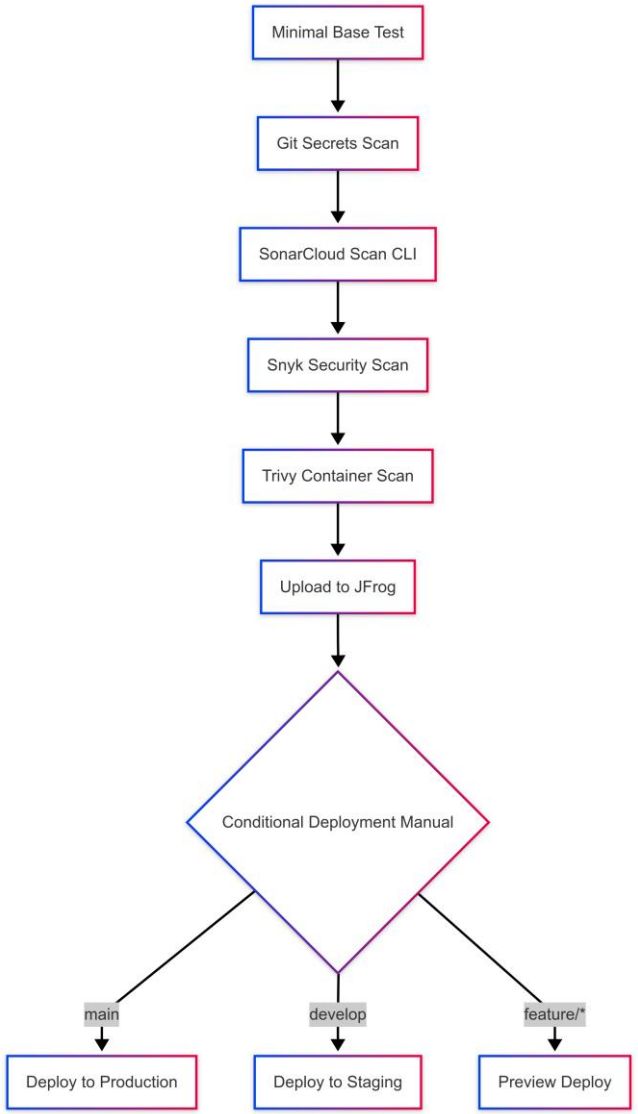


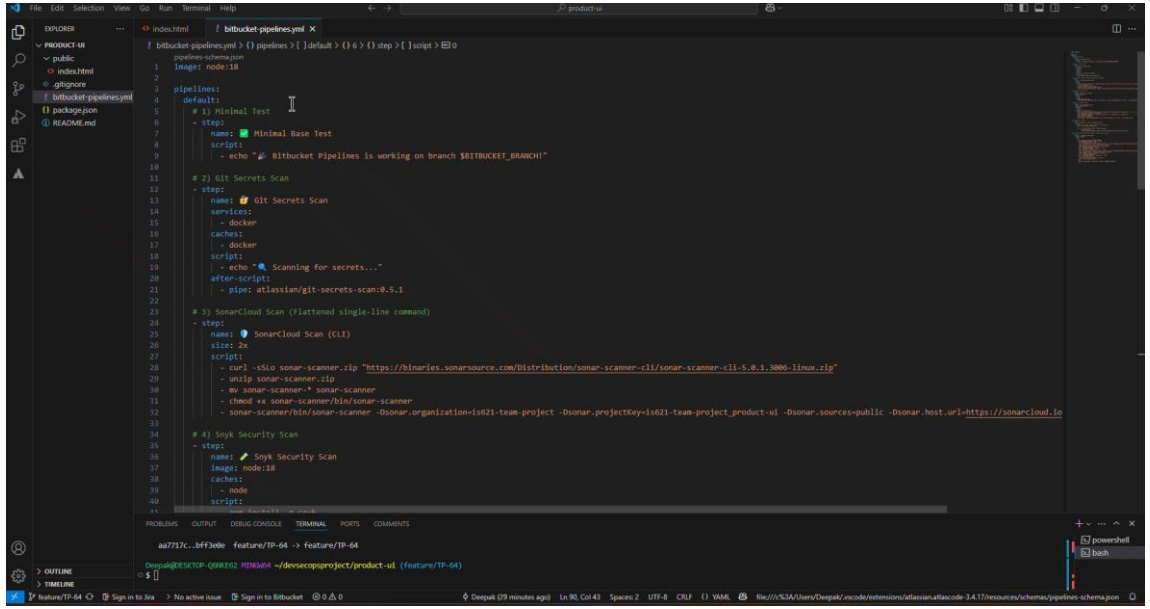
Security

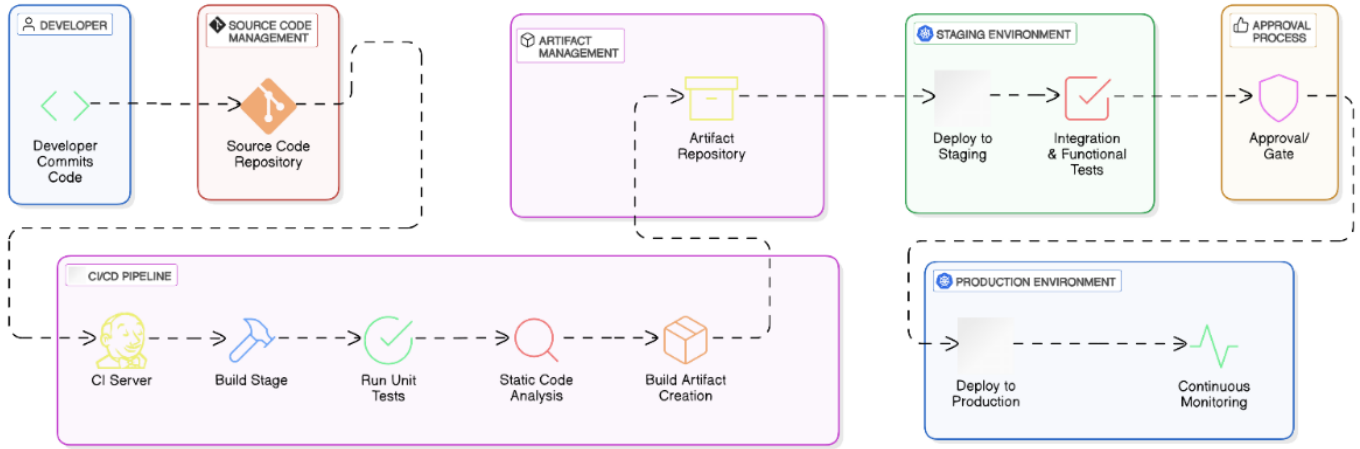
Centralized Ticketing System Security Considerations

	Confidentiality	Integrity	Availability
Objectives/ Requirements	Protect sensitive financial data, customer information and internal banking operations from unauthorized access	Ensure trustworthiness and accuracy of ticketing records, financial transactions and notification logs	Ticketing System and Dashboard must remain accessible and functional at all times
Risks	Unauthorized access to sensitive bank information via API interceptions, phishing attacks and insider threats targeting our services such as Admin and UserAuth	Data tampering of ticketing records, admin settings, and notification logs. Threats may include SQL injection and unauthorized modifications to APIs	DDoS attacks, Ransomware, and hardware failures disrupting critical banking services and our centralized ticketing system
Mitigations	• Implement robust end-to-end encryption protocols (i.e. TLS 1.3) for secure communications • Strict enforcement of RBAC and MFA for all users to counter phishing & social engineering defense • Use RBAC to restrict admin and IT staff access based on job roles or PAM (implement JIT access for IT staff to reduce exposure	• Deploy Web Application Firewall (WAF) to inspect and block malicious SQL injection attempts in real time • Database Hardening by following MAS TRM guidelines to enforce least privilege access, strong authentication and encryption for stored data • Use OAuth 2.0 for strong authentication of API requests and enforce RBAC	• Use load balancers to distribute traffic across DC with anti-DDoS protections to prevent service disruptions • Implement 3-2-1 backup strategy (3 copies of data, 2 diff. storage media, one at offsite) • Implement HA infrastructure and failover strategies to maintain network and system uptime

CI/CD Demo

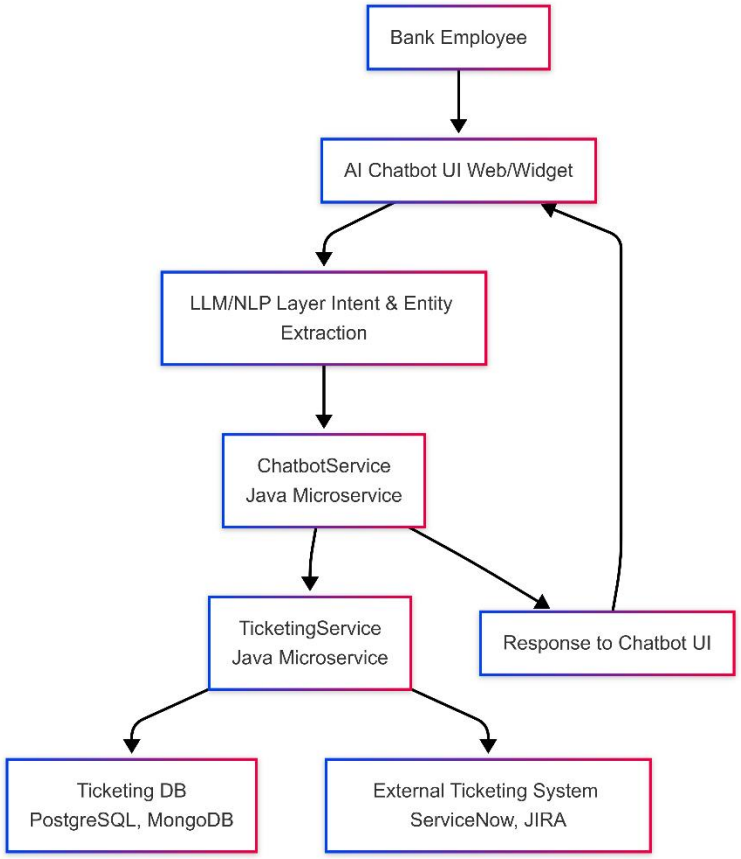

Minimal Base Test
Git Secrets Scan
SonarCloud Scan CLI
Snyk Security Scan
Trivy Container Scan
Upload to JFrog
Conditional Deployment Manual
main
develop
feature/*
Deploy to Production
Deploy to Staging
Preview Deploy

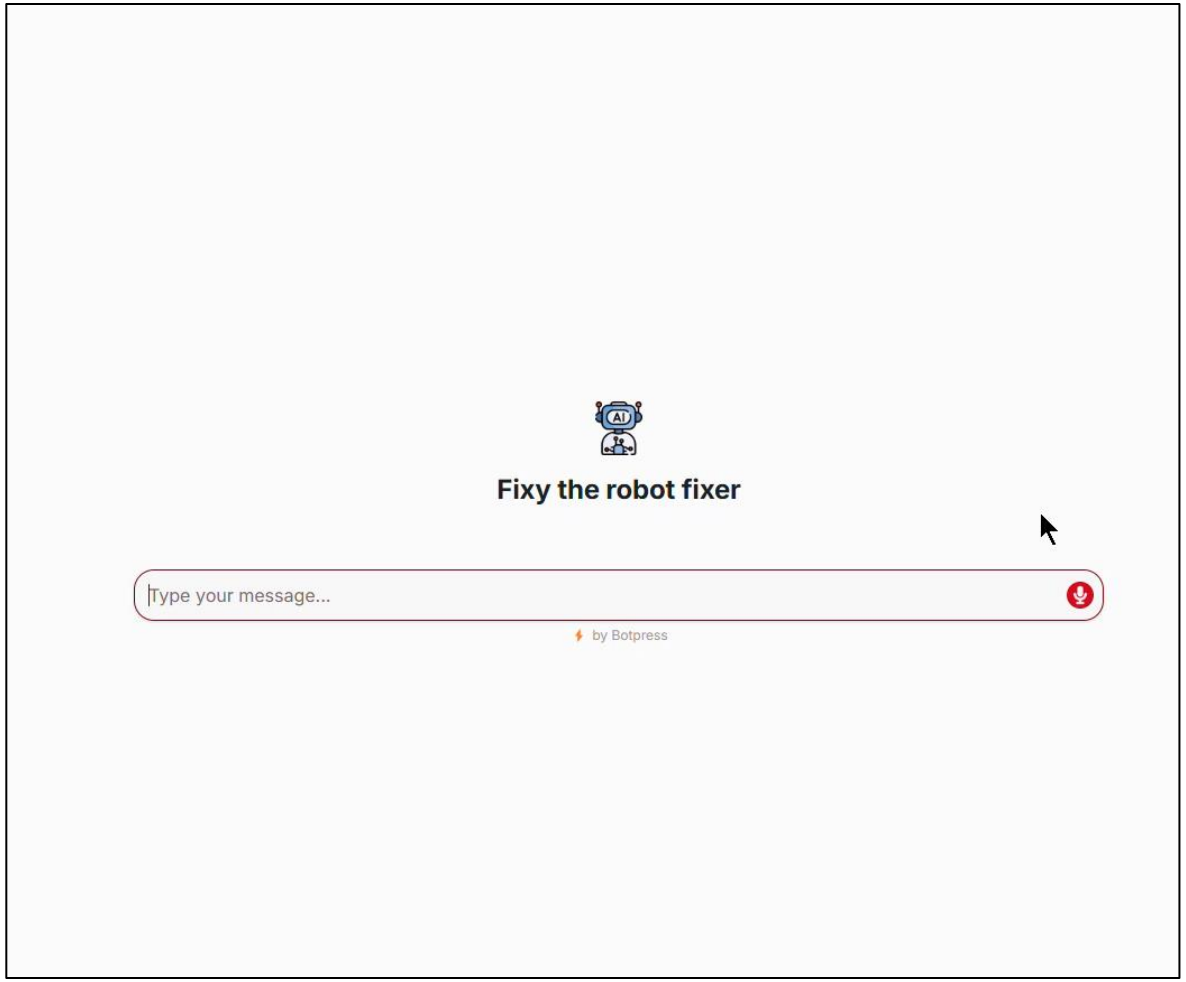



DEVELOPER
Developer Commits Code
SOURCE CODE MANAGEMENT
Source Code Repository
ARTIFACT MANAGEMENT
Artifact Repository
STAGING ENVIRONMENT
Deploy to Staging
Integration & Functional Tests
APPROVAL PROCESS
Approval/ Gate
CI/CD PIPELINE
CI Server
Build Stage
Run Unit Tests
Static Code Analysis
Build Artifact Creation
PRODUCTION ENVIRONMENT
Deploy to Production
Continuous Monitoring

Prototype

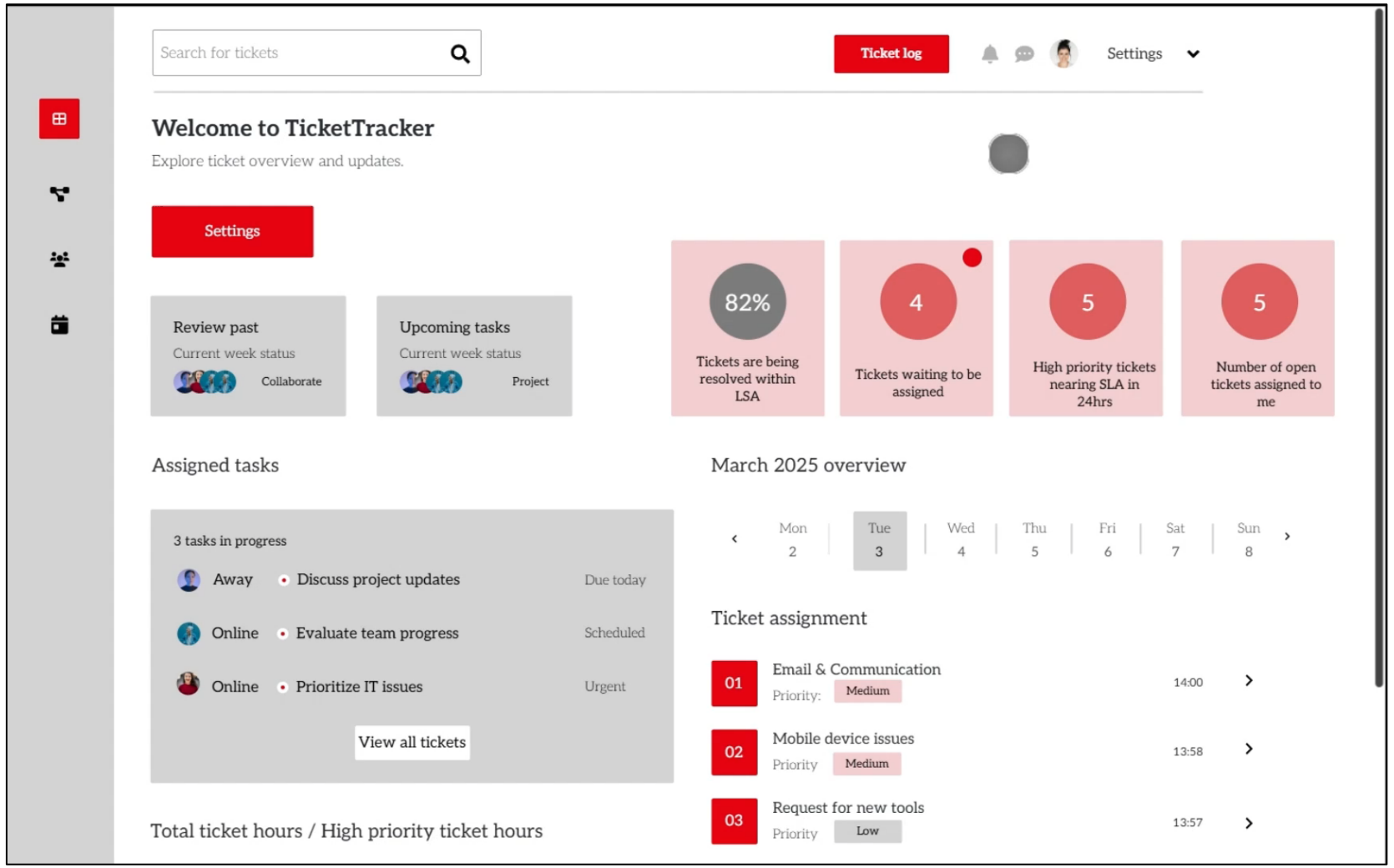
Chatbot





Prototype

Dashboard



Thank you!