

CI/CD Pipeline Adoption using Azure DevOps

Implementation of a CI/CD strategy using Azure DevOps and Fastlane to automate the build and release process for a flagship property management application, resolving integration conflicts and accelerating release velocity.

Overview

QBurst implemented a modern CI/CD strategy for the client's mobile development, moving them from slow manual processes to an automated pipeline.

- **Automated Pipeline:** Designed and configured a continuous integration, delivery, and deployment pipeline using Azure DevOps.
- **Build & Deploy Automation:** Integrated Fastlane to automate the build, signing, and deployment process for all environments (Dev, QE, Staging, Prod).
- **Reduced Errors & Effort:** Eliminated error-prone manual steps, resulting in faster, reliable software delivery and reduced developer effort in generating builds.



Client Profile

Our client is a leading property development company based in the Middle East. The company provides property development and management services across the Middle East, Asia, Europe, and North America.

Challenges

- **Delayed Code Integration:** Conflicts arising from delayed code integration cost the development team a significant amount of time, slowing down progress.
- **Manual Build Complexity:** Creating builds for different environments (Dev, QE, Staging, Production) was a challenging, multi-step manual process.
- **High Error Rate:** The error-prone manual build process frequently led to mismatches in build numbers, environments, and certificates, causing deployment failures.

- **Slow Release Velocity:** The combination of these issues resulted in extremely long release cycles, taking several months to get new features to production.

QBurst Solution: A Modern CI/CD Strategy

We proposed and implemented a comprehensive continuous integration, delivery, and deployment (CI/CD) strategy using Azure DevOps. This approach was designed to reduce integration errors, automate the build pipeline, and significantly increase project velocity.

Azure DevOps was selected as the CI/CD platform because the client already used it for project management and it offered a highly configurable pipeline. We integrated Sonar Cloud for automated Static Code Analysis on every Pull Request (PR) and Fastlane to automate the build, sign, and deployment process for all environments. The pipeline was configured with specific triggers for each branch merge, ensuring consistency and quality at every stage.

Technical Highlights

- **CI/CD Platform:** Leveraged Azure DevOps for its highly configurable pipeline and native integration with the client's existing project management.
- **Build Automation:** Utilized Fastlane to build, sign, and deploy the application, updating configurations automatically for QE, staging, and production branches.
- **Static Code Analysis:** Integrated Sonar Cloud to run static analysis on all incoming Pull Requests and full analysis on the develop branch.
- **Automated Triggers:** Configured automated pipeline triggers for each merge to the develop, QE, staging, and production branches, ensuring hands-free progression.
- **QE Team Empowerment:** Enabled the QE team to generate their own builds, reducing dependency on the development team and minimizing delays.

Impact: Increased Project Velocity and Productivity

- **Faster, Reliable Delivery:** The automation and configuration of resources in a reliable and repeatable manner led to a 70% reduction in the overall release cycle time.
- **Reduced Developer Effort:** Significantly reduced the manual effort required from the development team in generating builds, resulting in a 60% increase in developer productivity.
- **Reduced Errors:** The automated process nearly eliminated manual build errors, leading to a 50% reduction in deployment-related bugs.
- **Streamlined QE Process:** Reduced dependency between the QE and development teams by allowing QE to generate builds on demand, reducing testing delays.