



Mindtree

A Larsen & Toubro Group Company



Building Offline Apps

Offline apps are a necessity in many scenarios. For instance, they provide seamless user experience during limited or zero Internet availability. Modern mobile apps built for navigation, gaming and reading use offline mobile apps.

In this whitepaper, we discuss the solution approach, key best practices and the DevOps for building an offline mobile app using Xamarin framework.

→ Offline Apps

In this whitepaper, we have considered a usecase for agent enrollment in the insurance domain. The application discussed in this whitepaper is built as a thick client for desktop, laptop and tablets. We will discuss the drivers, solution architecture and best practices of offline mobile apps.

Given below are the main drivers for the use case:

1. Seamless agent enrollment in offline mode across various user devices
2. Handling complex business logic offline without any dependency on enterprise systems
3. High mobile app experience
4. Quick download of huge reports
5. Seamless viewing of large PDFs in offline mode
6. Integration of the mobile app with the existing enterprise applications

Overall solution architecture

We have depicted the overall solution architecture in Figure 1

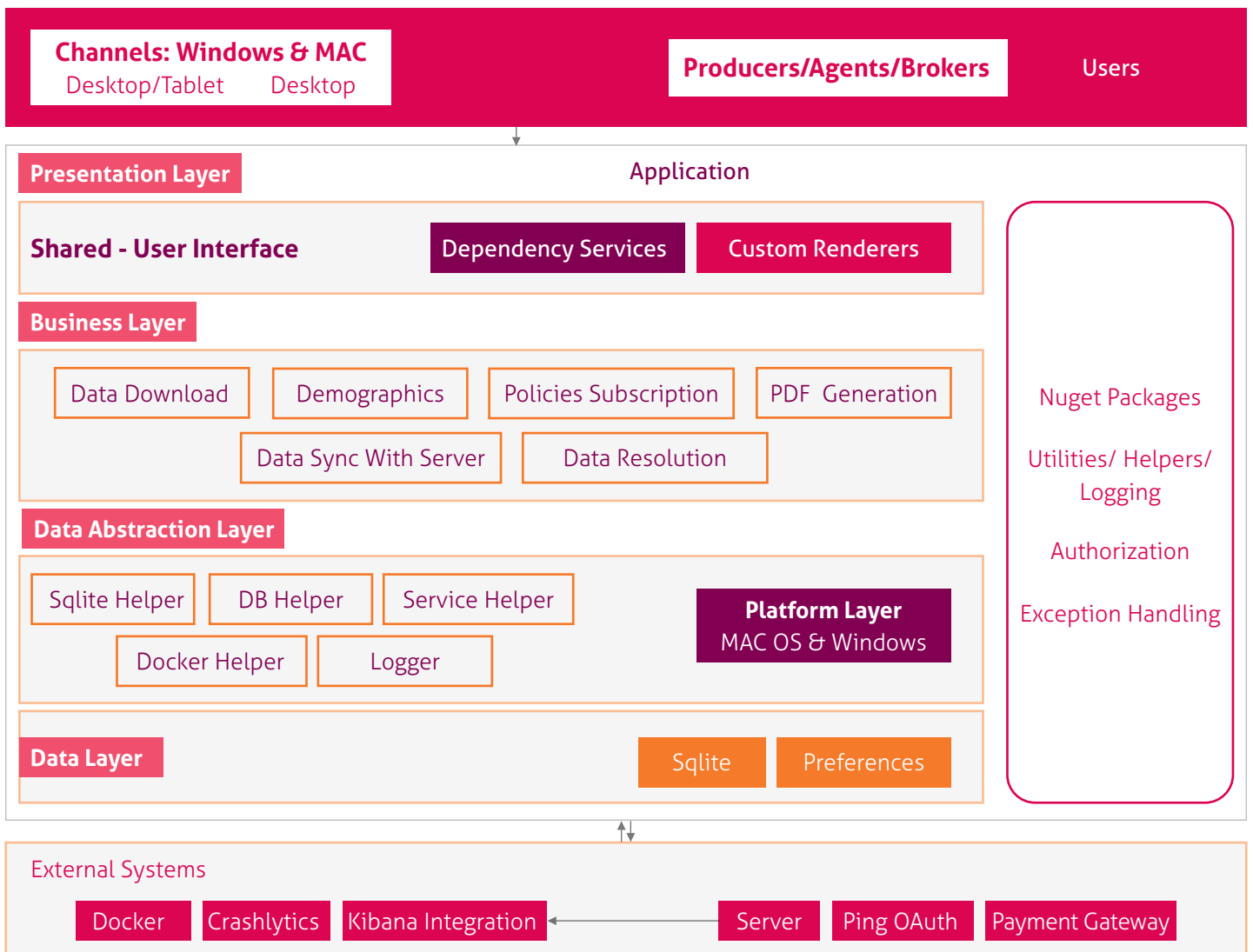


Figure 1 Overall solution architecture

This application is supported on Windows (tablet/desktop) and Mac OS, and users are mainly producers/agents/brokers. The application has been developed using Xamarin technology, where 80% of code sharing has been done between Mac and windows. There has been an explicit use of custom renderers, behaviors, converters and the dependency service to exploit the native functionality. The user interface (UI) of the application is responsive and can be used in different screen sizes and resolutions. Data download deals with fetching all the needed data to support the offline mode using static and dynamic global dump. The existing web APIs are being used to get the group specific data and once the entire data is available, the business logic and computations are handled to determine the eligible data, which the user can use for subscription of policy. The communication with the docker is established to generate a PDF once the policy is subscribed.

We have exposed multiple interfaces to avail different services that are needed to process business logic. Some of these are SQLite service, service helper, DB service, and logger. Data layer is the main repository where there are huge records of data stored using SQLite segregated across multiple database, with data being secured using database encryption. We have also used preferences to persist primitive data types. All the CRUD (Create, Read, Update and Delete) operations are implemented to deal with data transaction.

NuGet packages provided by .net framework has been used to incorporate some of the services. There is also dependency on external systems such as ping authentication for secure login and payment gateway for payment-related services. Kibana integration is done to trace and debug API failures. We have also used a login tool called Sentry, which provides both offline and online crash/exception reports for Windows and Mac platforms.

→ Key best practices

- Prism framework is used for making the application loosely-coupled, maintainable and testable
- Segregation of presentation and business logic is achieved through the Model-View-View-Model (MVVM) design pattern
- Text localization implemented using Resx files
- Implementation of unit test cases using Xunit and Moq
- SonarQube is used to continuously scan codebase for quality and security
- Modularity - Better maintainability through clearly defined and independent modules
- Compatibility - Better design for integration with other products
- Extensibility – Flexibility to add new capabilities without affecting the existing architecture and design
- Fault-tolerance - Recover from failure. Handling unexpected crashes / errors



→ Code Reusability

- Custom renderers, reusable UI controls created with custom bindable properties
- Multiple visual element instances are created and applied with application level XAML styles
- Interfaces/services written for common business logic used in different modules
- Common code coverage across platforms ~ 85%
- Dependency services to exploit native features.

→ Challenges & Solutions

We have depicted the main challenges in Figure 2, Figure 3 and Figure 4.

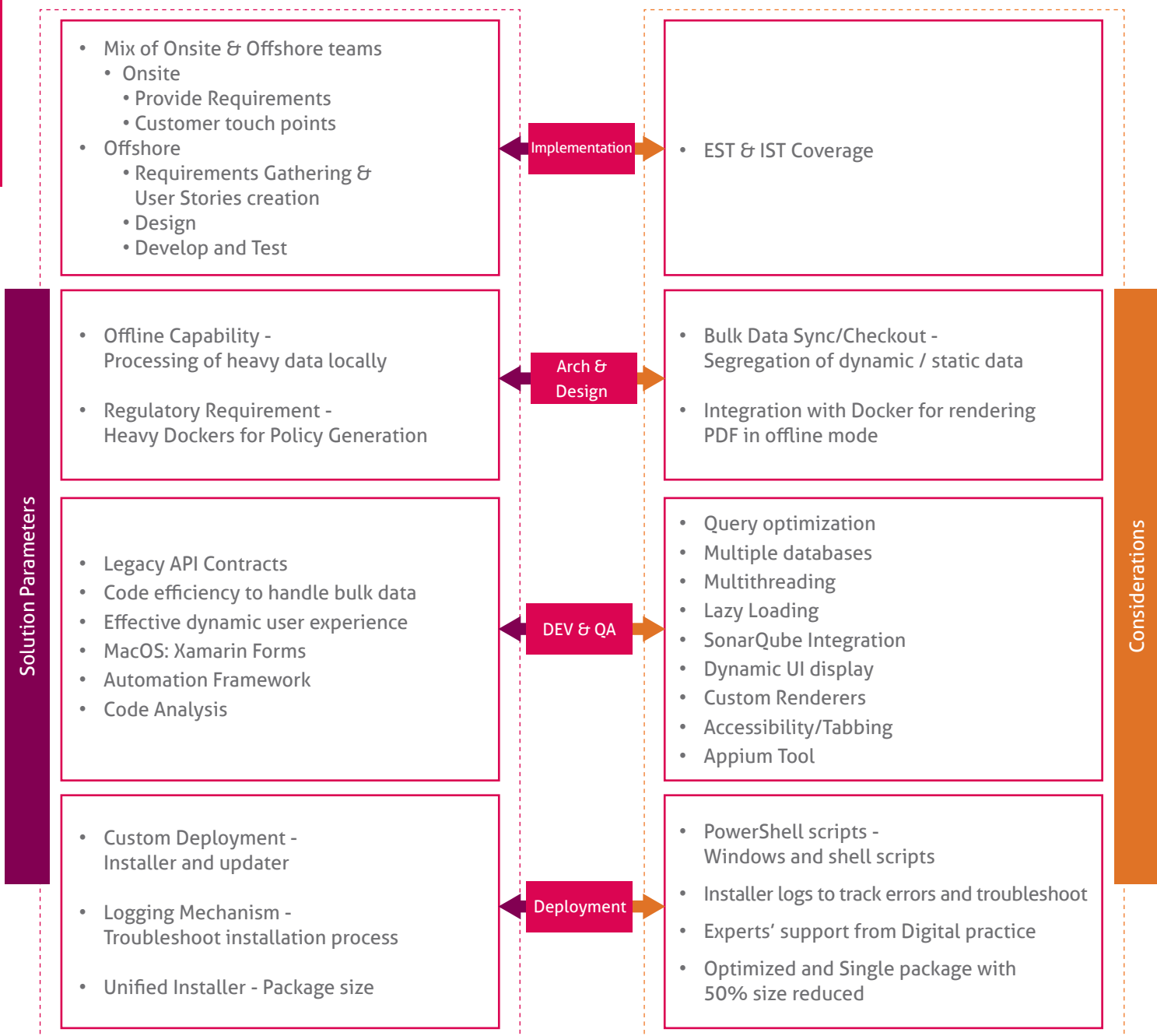


Figure 2 SDLC related challenges

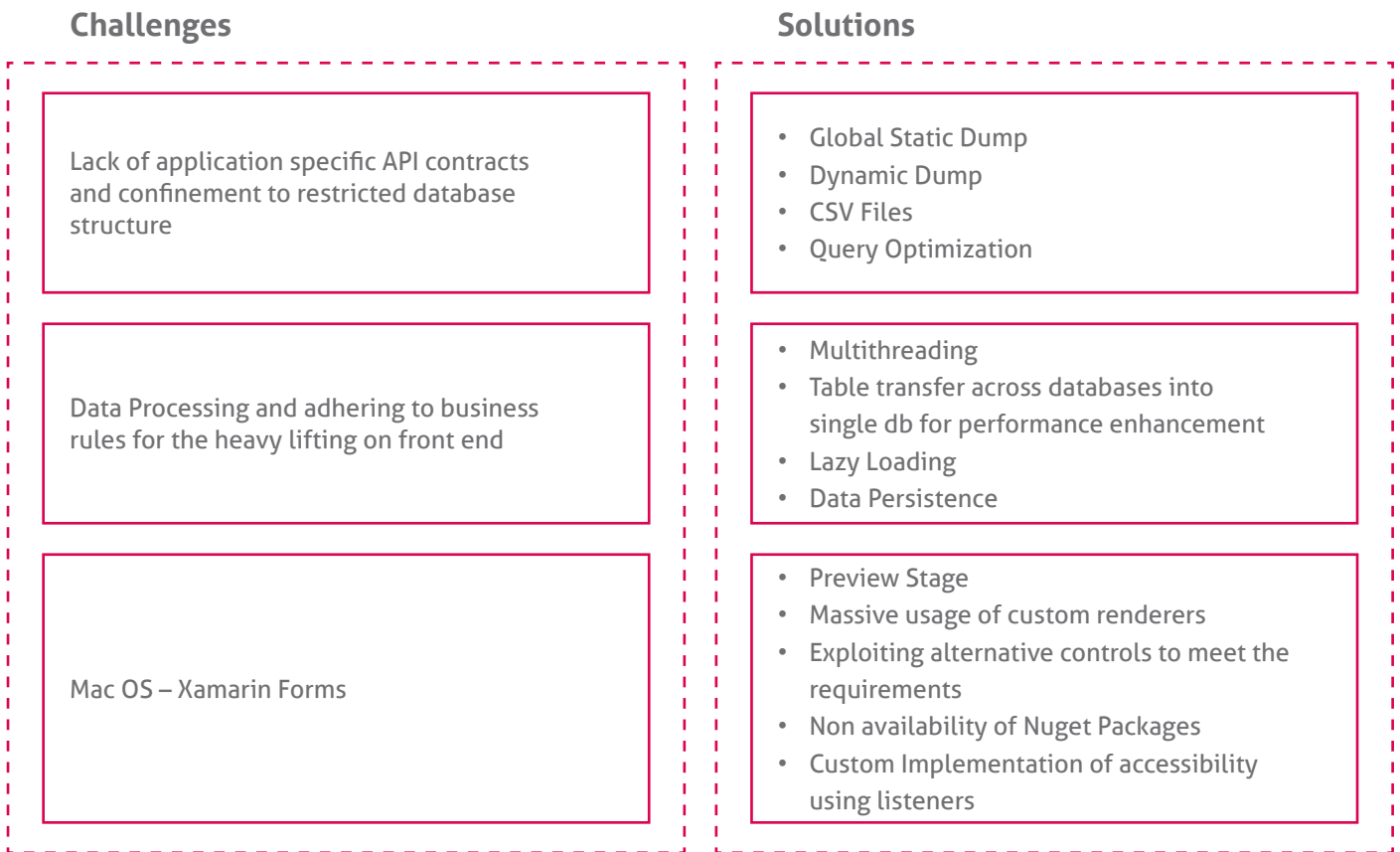


Figure 3 Key challenges

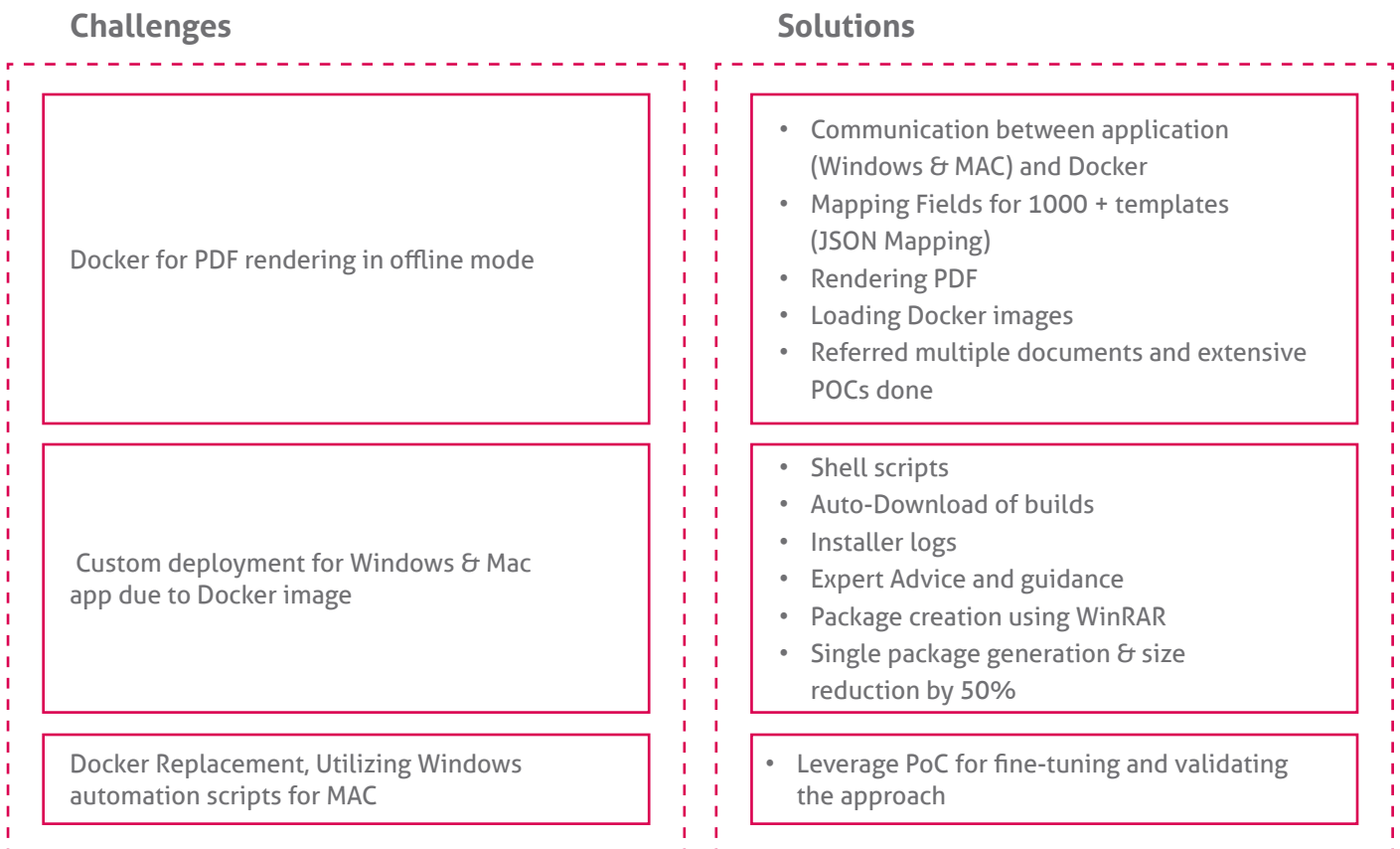


Figure 4 Deployment Challenges

→ Deployment

We have discussed the deployment details in this section.

Single Click Installation

- Unified installer is a single-click executable file that can install all the packaged software required to be installed and effortlessly managed.
- The unified installer is an executable file which has been created by using WinRAR (a third party tool).
- Users will be provided with the setup wizard guide, where they are guided about the impact of using this installer.
- Based on the user input, we start the installation by calling the PowerShell script automatically.

Simplified Script with Powershell

- PowerShell scripting simplifies the configuration and helps automate tasks on the Windows operating system.
- Our first challenge was to install the application using PowerShell, for which, we used the Windows PowerShell Integrated Scripting Environment (ISE).
- ISE is used as the host application for Windows PowerShell scripting. The ISE's command with common parameters helped us with writing commands for our installer.

Runtime Business Intelligence

- The logic of the overall installation procedure is driven by the unified installer's main installation script. The product feature include scripts that are necessary. For example, the prerequisites of a software application which has been validated before installation.

We have depicted the installation workflow in Figure 5 and update workflow in Figure 6

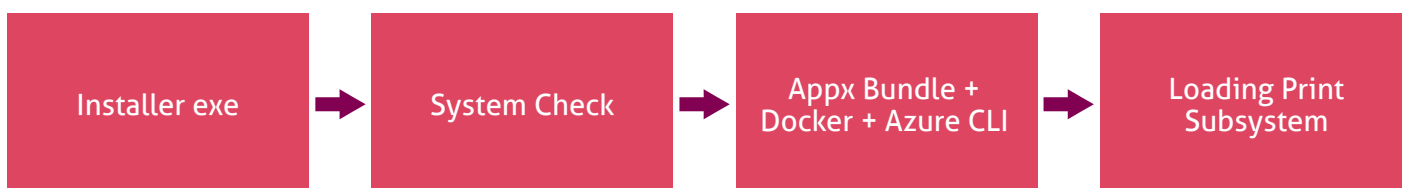
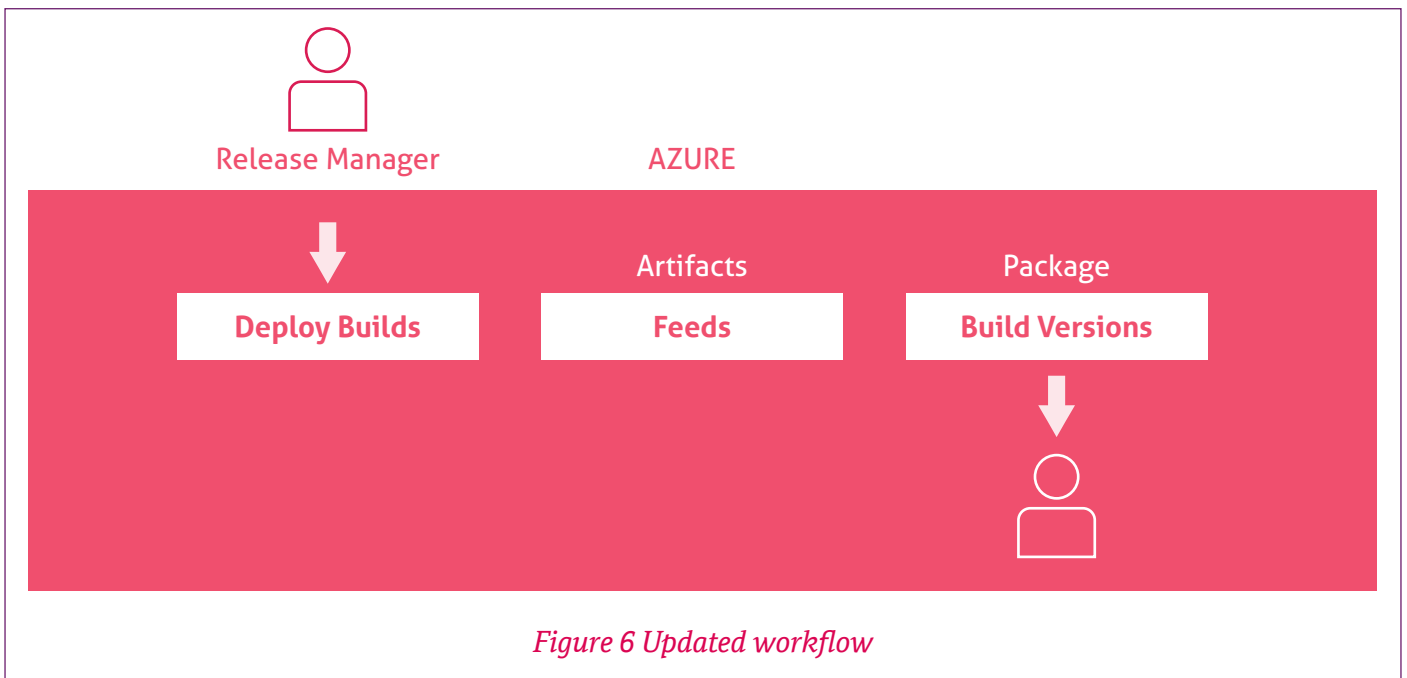


Figure 5 Installation Workflow



→ DevOps Strategy

We have discussed the overall DevOps strategy in Figure 7

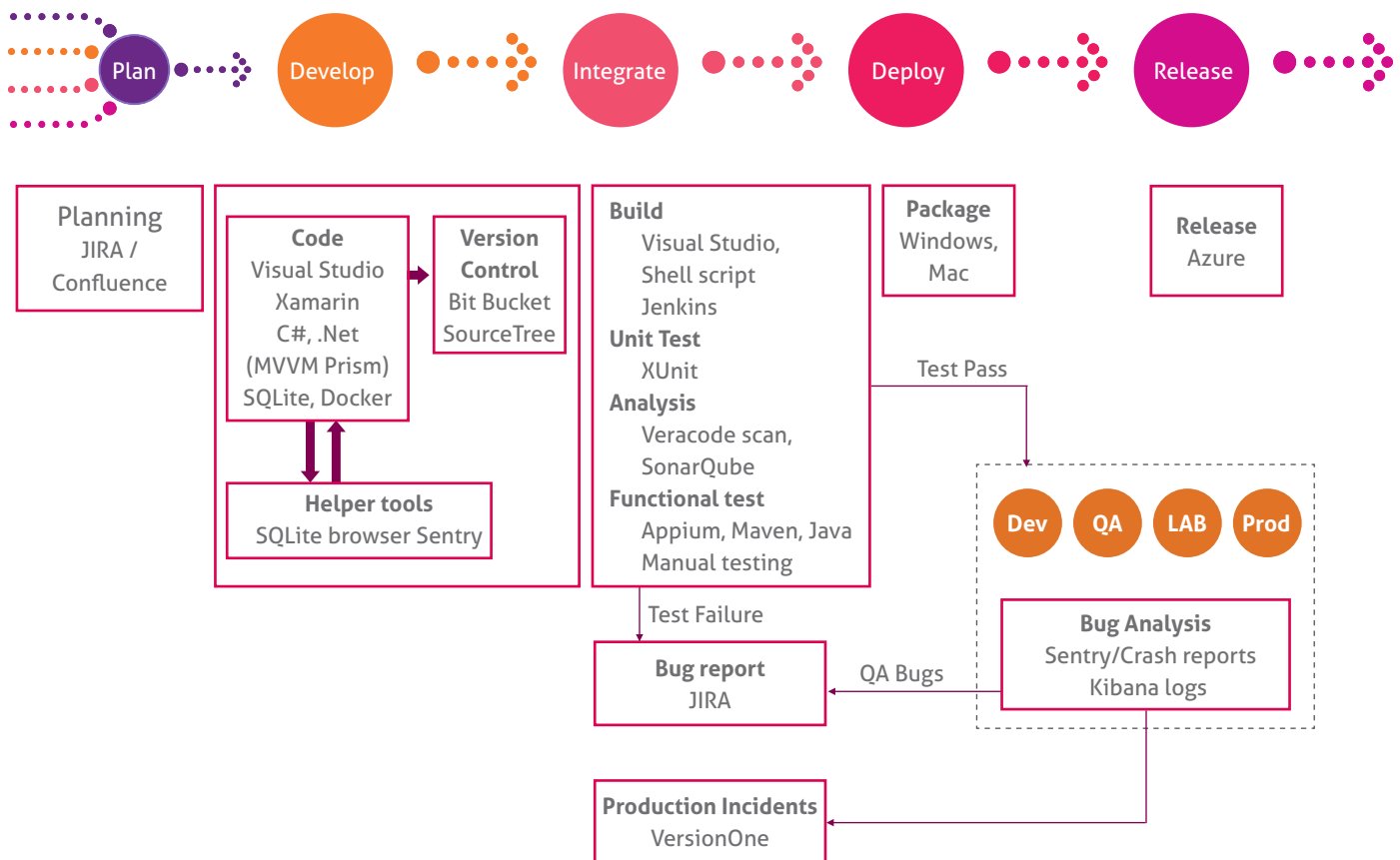


Figure 7 DevOps Flow

While requirements are gathered from clients, JIRA is used to create epics and user stories. After user stories are groomed and effort is estimated, it becomes ready for development. For development and unit testing, Visual Studio IDE is used. Our app is a cross platform mobile application, which uses Xamarin Forms by using XAML for the frontend and C# for the backend. Segregation of presentation and business logic is achieved through the Model-View-View-Model (MVVM) design pattern. This clean segregation between application logic and the UI helps make it easier to test, maintain, and evolve, and also helps improve code reusability. SQLite is used as the database engine and we use the DB browser for SQLite to create and edit the database files. Bit Bucket is used for version control and its integration with JIRA helps with traceability management. As part of the user story, we can track the commits, the branch it was worked upon and the review status.

Every week, the SonarQube Jenkins job is run to identify bugs and code smells. These bugs are logged in JIRA and fixed as part of development. XUnit is used to test the business logic inside the view model. Kibana integration is done to trace and debug API failure. We have also used a login tool called Sentry, a crash reporting platform, which is used to get all the necessary data for analyzing and replicating app crash issues and provides both offline and online crash/ exception reports for Windows and Mac platforms.

Post development, the build is triggered pointing to the QA environment. QA performs manual testing and also runs the automation scripts that are developed using tools such as Appium and Maven. If any bugs are identified as part of this process, they get logged in JIRA under the related epic. QA provides a sign off based on the sign off criteria. Post QA sign off, pre-prod and prod builds are created. Once the high level prod sign off is provided, build is released into production. Any production incidents are tracked in the VersionOne tool.



→ DevOps Model

We have depicted the overall DevOps model in Figure 8

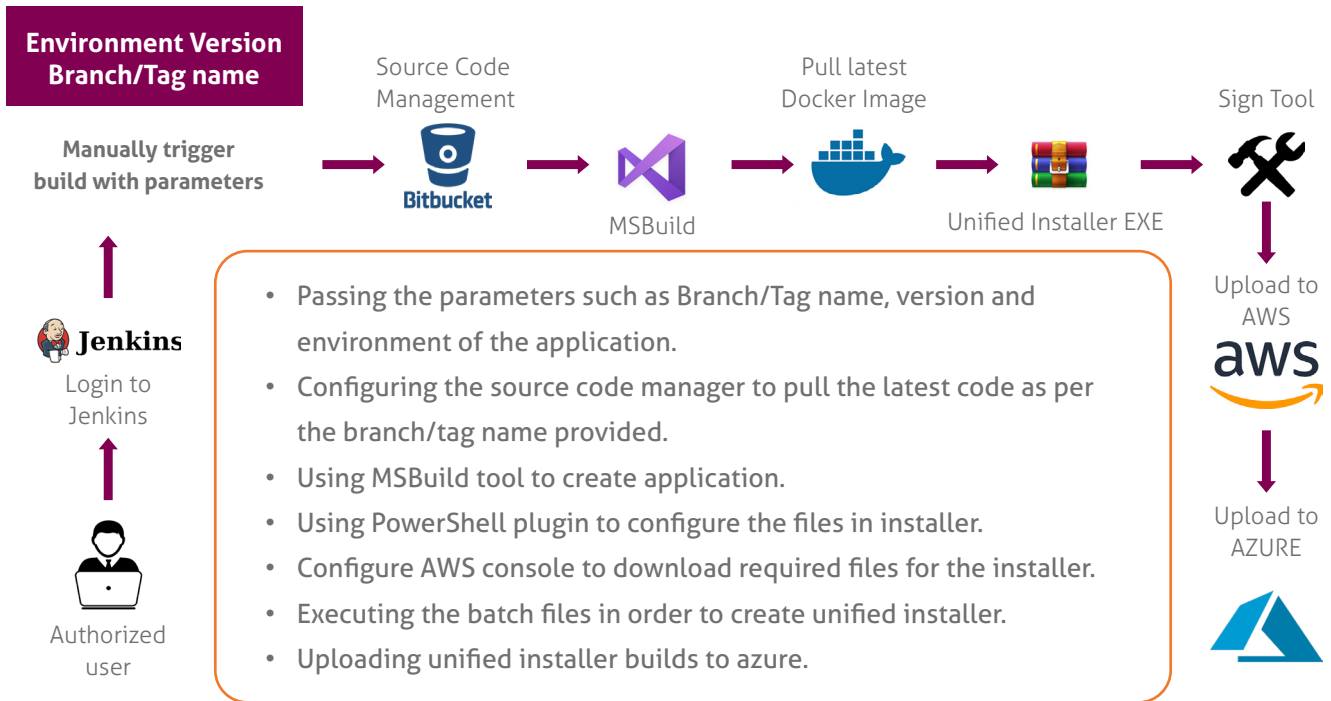


Figure 8 DevOps Model



→ Automations

Various test-related automations are depicted in Figure 9 and Figure 10

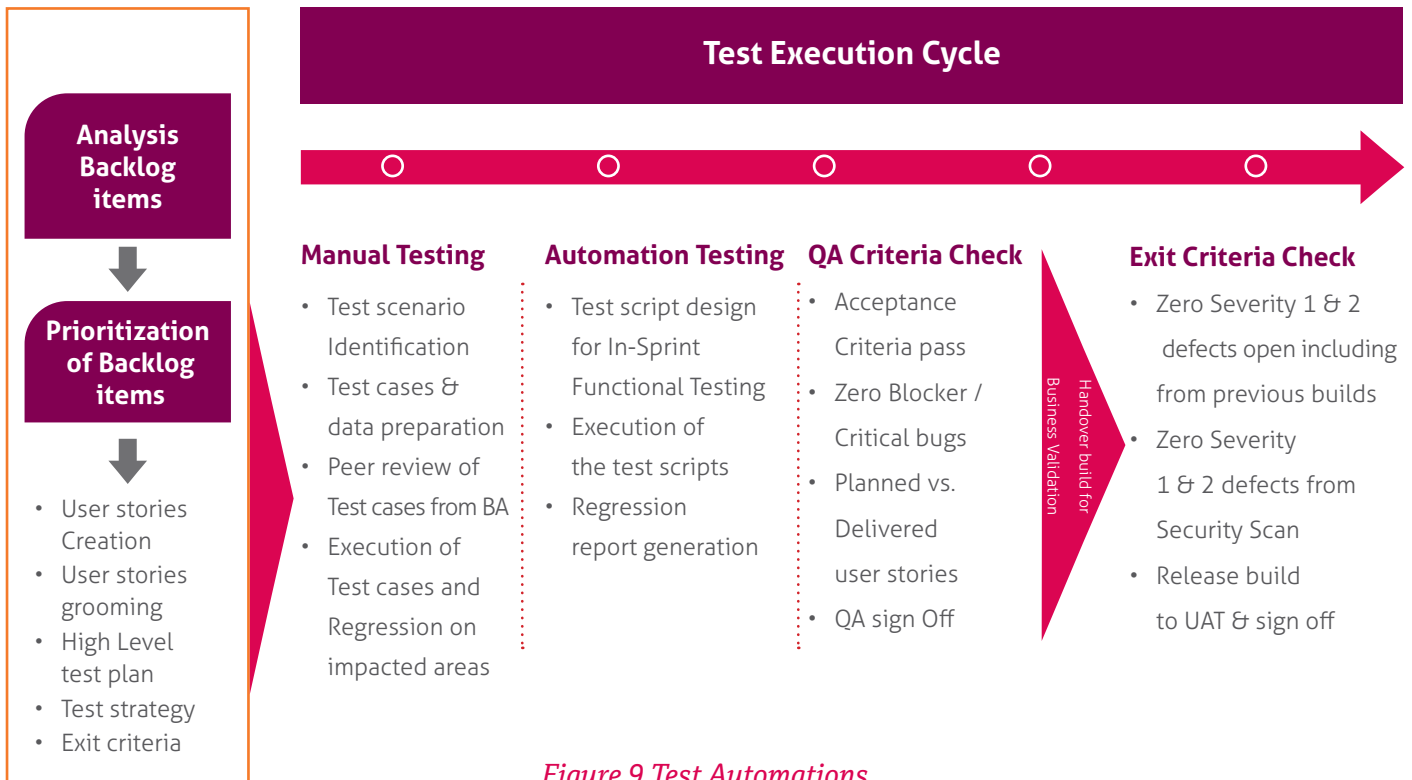


Figure 9 Test Automations

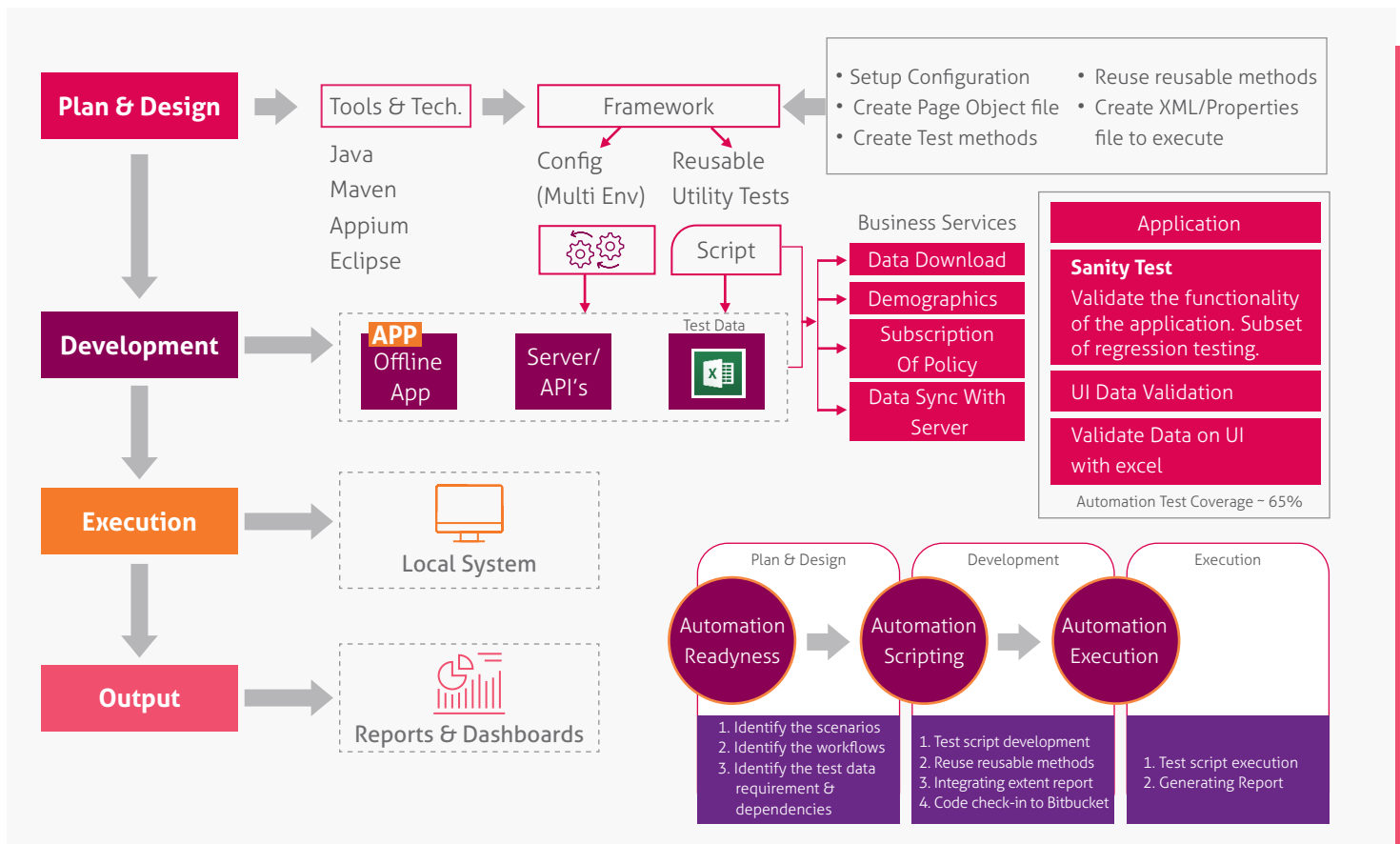


Figure 10 Unplugged Framework

→ Conclusion

One can use Xamarin for building offline apps, which can be shared across Microsoft Windows and Mac platforms. Also, business services and data abstraction layer components in the overall solution can be segregated to implement separation of concerns. One can also use other best practices such as MVVM pattern and SonarQube for code analysis and code reusability using modular components. A Jenkins-based DevOps model can be used to automate the release management activities.

About the authors



Susmitha G K
Module Lead

Susmitha is a Module Lead working on mobile projects and has experience in design, development and implementation of cross-platform mobile applications. Her interests include solving complex issues and studying project management methodologies.



Akshaya Narayan Naik
Module Lead

Akshaya is a Module Lead working on Mobile Application projects and has experience in understanding and analyzing client needs/requirements. She is experienced in the design, development and implementation of mobile applications. Her interests include developing complex UI and solving difficult problems.



Narendra Sharma
Module Lead

Narendra Sharma is a Module Lead working on mobile projects and has end-to-end experience in developing cross-platform applications, installer scripts for Windows and Mac OS and devops setup and configuration. His interests include working on new technologies and solving customer problems.



Md. Sahin Ahmed
Module Lead

Md. Sahin Ahmed is a Module Lead working on mobile projects and has end-to-end experience in developing cross-platform applications and REST APIs. His interest includes working on new technologies and solving complex problems.



Daniel Kiran J
Module Lead

Daniel is an Engineer working on Mobile Application projects and has experience in understanding and analyzing the client needs/requirements. He is experienced in automation scripting and testing of mobile applications.



Dr. Shailesh Kumar Shivakumar
Solution Architect

Dr. Shailesh Kumar Shivakumar is a Solution Architect and has 19+ years of experience in a wide spectrum of digital technologies including, enterprise portals, content management systems, lean portals, and microservices. Dr. Shailesh holds a PhD degree in computer science and has authored eight technical books published by the world's top academic publishers such as Elsevier Science, Taylor and Francis, Wiley/IEEE Press, and Apress. Dr. Shailesh has authored more than 14 technical white papers, five blogs, twelve textbook chapters for various under-graduate and post graduate programs and has contributed multiple articles. He has published 20+ research papers in reputed international journals. Dr. Shailesh holds two granted US patents, apart from ten patent applications. Dr. Shailesh has presented multiple research papers at international conferences. Dr. Shailesh's Google Knowledge Graph can be accessed at <https://g.co/kgs/4YoaiN>. He has successfully led several large scale digital engagements for Fortune 500 clients. Shailesh can be reached at Shaileshkumar.Shivakumarasetty@mindtree.com

About Mindtree

Mindtree [NSE: MINDTREE] is a global technology consulting and services company, helping enterprises marry scale with agility to achieve competitive advantage. "Born digital," in 1999 and now a Larsen & Toubro Group Company, Mindtree applies its deep domain knowledge to 260 enterprise client engagements to break down silos, make sense of digital complexity and bring new initiatives to market faster. We enable IT to move at the speed of business, leveraging emerging technologies and the efficiencies of Continuous Delivery to spur business innovation. Operating in 24 countries across the world, we're consistently regarded as one of the best places to work, embodied every day by our winning culture made up of over 27,000 entrepreneurial, collaborative and dedicated "Mindtree Minds."