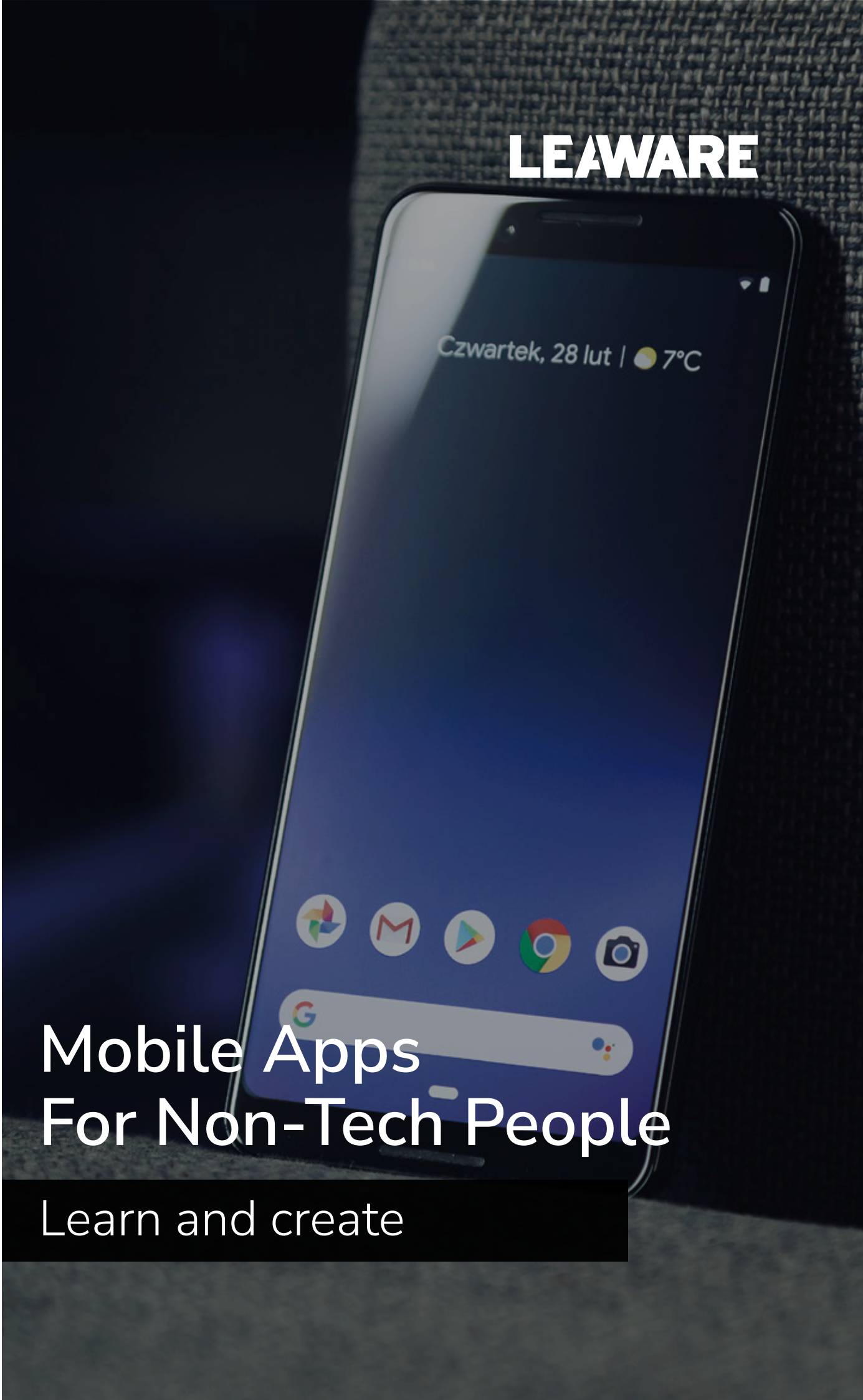
Imagine

LEAWARE



Mobile Apps For Non-Tech People

Learn and create

CONTENTS

CONTENTS

What will you find in this Content?

What you need to know when you want to develop a mobile app and you are not a technical person.

Who is the target of this content?

People who want to outsource mobile app development but are not technical. This can be a startup founder or someone responsible for developing a mobile app in a company

What is the reader challenge?

Those without experience in developing apps may lack knowledge of available options, not be aware of risks, not know good practice, and may select unsuitable partners. If a person is not equipped with essential knowledge, it could result in:

- An application (app) with a lot of bugs.
- Technology limits on future app development.
- Expensive app development.
- Rewriting the app due to technical debt.

So, this is our solution

- Explain the risks that a person should consider.
- Learn about good practice.
- Demonstrate how a mobile app should be developed.
- Show how mobile app development differs from the development of other solutions.

INTRODUCTION

The world is mobile; there's no doubt about it. Technologies are evolving and are present in many aspects of our lives. Entering this digital world with your company requires some knowledge to prevent you from becoming lost. When you are planning to build a mobile app for your business, you need to make plenty of decisions, be aware of the risks, and be aware of best practices.

Let us guide and help you to understand the process of mobile app development. It is crucial to make the right decisions not only about the shape and performance of your app, but also the tech partner you need to choose.

MOBILE DEVELOPMENT TRENDS

These are the current trends reggarding to mobile development in 2022.

- Wearable technology has a bright future as the number of users worldwide grows each year. Ensuring that your mobile app is compatible with wearable devices such as fitness trackers or smartwatches means that you can boost user retention with ease.

- The simpler your development, the easier it will be to hook your audience. Don't forget to continually innovate.

- Make sure that you know the needs and challenges of your audience.

- Understand who the user is. Set your customer at the center.

- Don't design the solution or choose crucial functionalities for the first version of your product until you know who your user is.

- The main problem that you have chosen to solve should be significant.

- Focus on quality.

- Setting the direction and creating the shape of the app should be based on understanding the main problems and needs of potential users.

- Validate your idea and solution with early adopters to confirm the right direction and gain trust.

- Plan the process of your development to monitor growth.

- Add the functionalities necessary to achieve your app's first business goal.

- User feedback confirms the right direction and suitable functionalities for the app.

- Listen to your user and implement their feedback to improve your product.

- Users won't use the product if the app partially solves their problems.

- If you want to boost your sales, increase your reach and get more satisfied customers.

- AI-based solutions can be successfully applied in various industries for a variety of purposes.

- From a business perspective, cross-platform development is a perfect solution as it reduces development costs, save time, and be easier to maintain and scale.

WHAT THE WRONG CHOICE OF DEVELOPMENT TEAM CAN DO

There are some examples of mobile apps that have failed. You may have heard of them. Lack of experience, inadequate market research, no market fit, or no ideas for further development are potential reasons.

- **Building an app that has bugs reduces the quality of the product.**
- **Choosing inappropriate technology limits future development.**
- **It is likely that the development of the mobile app will be costly.**
- **Poor quality apps need to be rewritten down the line.**

Vine - An app for short-looped videos. It became popular at a point, which is when Twitter purchased it. However, its popularity didn't last long. Vine had its potential and was created before Instagram, but when Instagram succeeded, it took Vine's users. Vine didn't care enough about their user's needs, changes in user behavior, and adapting to changes. There was no clear plan for Vine's development.

Rdio - Was the first music streaming app launched in 2010. It was created by Skype's founders who were focused on developing the perfect music app. They lost the balance between development, research, and scaling. They didn't have a plan on when to expand and how to do it.

The marketing efforts were also poor. Rdio was a good product, but the creator's business sense and approach weren't good enough. There can be many other reasons as to why mobile app's fail. Let's investigate the main areas to consider when developing your mobile app.

Hailo - Was a mobile taxi app founded in the UK, matching taxi drivers with passengers. It was designed to be a great success and gained $100 million in funding, but the business never became successful. Crucially, poor market research and poor business decisions led to the failure of the business.

Hailo didn't predict that London and New York are completely different and different cities need different approaches. Hailo was functioning well in the UK but launching it in New York was a mistake. Finally, competitors like Uber overtook Hailo.

BE WELL PREPARED

When it comes to creating a digital product, the factors that you need to consider are always the same. The sense of building a solution stays in the center. The process of mobile app development should be well planned, designed, and managed.

The first thing to consider is your business goal. Start with solid market research, gathering knowledge about potential users, and verifying the problem you want to solve. Know the why, who, and how. If you fail to properly address the needs of users, you may create a solution that no one will use.

That's why the following steps are critical:

1. Understanding your users challenge.

"Let's make a digital product, it's easy, isn't it?" (check our white paper).

2. Product discovery process of MLP Version in a budget.

"Avoid overinvestment in the 1.0 version of your application" (check our white paper).

CHOOSE THE RIGHT MOBILE TECHNOLOGY

The choice of the right mobile technology is a crucial step. There are different technologies that you can choose from, but you should always have your business goal in mind. Be sure of what you want to achieve, thinking about the time and cost of the development, performance of the app, and UX/UI design. The choice should be based on:

-The availability of experienced developers who will be able to develop your mobile app in the future.

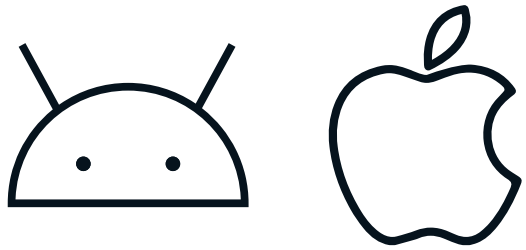
-Other technologies you currently use for solutions in your business.

-Limits of a specific technology.

To gain the highest quality and performance in the app, choose well-supported technologies. There are two main categories of mobile technologies:

- **Native.**
- **Cross-Platform**

Native



Cross - Platform

Both categories of technology evolve. When deciding to build a native or a cross-platform technology to build an app, remember that every technology has its advantages and disadvantages. Technology should be adequate for your project and your business. The wrong choice can cost you time, money, resources, and can cause problems with the app in the future.

Native Technologies

Are used for building apps that are dedicated to work on mobile operating systems. The two main operating systems are Android and iOS. Some popular apps made in native technologies are Netflix, Evernote, Uber, LinkedIn, and WhatsApp. Native technologies are a good choice for building long-term digital products as they are more stable.

Advantages:

- **Native apps are secure, intuitive, and responsive.**
- **They perform better and work faster compared to cross-platform apps.**
- **They provide direct access to all operating system capabilities.**
- **New updates of the system are more available.**

Disadvantages:

- **Native development can generate higher costs both initially and post-launch.**
- **Some iOS app-building functionalities are locked in this technology.**
- **There is no code reuse.**
- **Native development can be more time-consuming as separate teams are needed to develop app versions for different platforms.**
- **Native development projects are more difficult to manage.**

Cross-Platform Technologies

Are for building mobile apps that can run on many platforms. Examples of mobile cross-platform technologies are Xamarin, React Native, and Flutter. Using cross-platform technologies can be a good choice for prototyping, for startups, and for products that need to go to market faster.

Advantages:

- **Development is shorter and more cost-effective as the code can be duplicated for many platforms.**

- **Faster bug fixing.**
- **Easy maintenance.**
- **UX is close to native.**
- **Apps are created with great performance and experience, close to native apps.**

Disadvantages:

- **They work slower compared to native apps.**
- **Some delays in updating may occur, especially operating system updates.**
- **The technology takes up more storage space.**

Native

- Excellent Perfomance
- System Updates Avaible Quicly
- Intuitive, Responsive, Fast-Working

- Higher Cost of Development
- No Code Reuse
- More Time Consuming Development

Cross - Platform

- Fast & Cost-Effective Development
- Quick Bug Fixing
- Easy Maintenance

- Slower Perfomance
- Possible Delays in Updating
- More Storage Space Needed

THE ARCHITECTURE OF A MOBILE APP

App architecture describes how the elements of the solution work together and what structural elements it is composed of. Even in small solutions, the architecture can be complicated, containing many components. All elements need to be well-built to ensure the quality of the product.

MOBILE DEVICE CAPABILITIES

Online/offline support

-Offline mode is not always possible. Some apps are solely online.

-The server will always recalculate data faster than a smartphone. For example, if you have a large database with many relationships, it is sometimes better to use the server than the database. Sending a query to the API and then retrieving the data is faster than the mobile app.

-Data caching.

The app should be online with the ability to work offline, if possible.

Push notifications

- Notifications sent from the server, like a notification about a new promotion, news, etc.

- Local notifications, like a reminder about a meeting, watering plants, etc.

Sensors

- Accellometer.

- Camera.

-Not every sensor is available on both platforms, or is available with limitations.

Minimum version of operating system

- It is not worth using old versions of iOS or Android because their market share may be too low and this may limit the possibilities of the app. You do not want to limit functionality just because a user has an old phone.

ANALYSIS AND DESIGN

Mobile app development has dramatically changed the way we use our smartphones. People expect their phones to perform everyday tasks quickly. As useful mobile apps become more popular among users, businesses must pay close attention to the user experience and user expectations. Businesses can't get enough traffic or clicks on their mobile app with a plain design. More attention should be paid to the analysis and design of the app, which significantly improves the user experience.

MOCKUPS

A mockup is a relatively detailed outline of the appearance of an app. The difference between an app mockup and a wireframe is that the mockup includes colors, layout, images, and typography, while wireframes only feature sketches and text. A mockup is usually static. It gives you a glimpse into how the completed app will look and operate. Mockups are very important in the UX design process.

There are many ways of creating a mockup. If you are a developer, you can code a mockup of your mobile app. While this may be time-consuming, it is an excellent way to get started on the project. You can start figuring out what works and what kind of technical expertise will be needed to complete development.

Another way to create a mockup is to use graphic design software. Since the mockup design is static, this should be quite easy. However, you need to be an expert graphic designer, or you may struggle. The fastest and easiest way to create a mockup design is to use a mockup app.

There are many online tools that you can use to create a mockup design of your mobile app. Some of the top mockup apps are UXPin, MOQUPS, MockPlus, Balsamiq, Sketch, MockingBird, Invision, and Proto.io.

REASONS WHY MOCKUPS ARE USED

The purpose of developing mockups is to show, incorporate, and experiment with all the design elements that come together to offer a user experience. The aspects involved are:

- **Color schemes.**
- **Layout.**
- **Fonts and typography.**
- **Images and other visuals.**
- **Navigation.**
- **Icons.**
- **Spacing.**

When you are developing a mockup design for a mobile app, you are working on the look and feel of the app once it's ready to be launched.

ARCHITECTURE

Good mobile app architecture is the foundation of all well-designed software and can provide an excellent user experience. However, when it comes to mobile apps or platforms, there's always been a problem with good architecture.

WHAT MAKES GOOD MOBILE APP ARCHITECTURE?

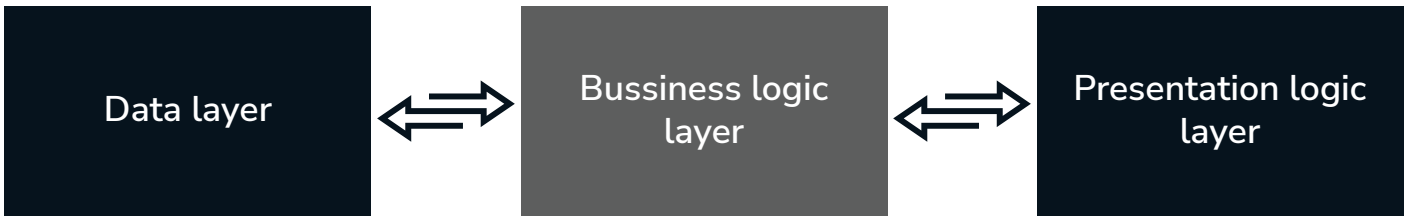
Good mobile app architecture enforces assumptions and good programming patterns like KISS, SOLID, and ensures that components have multiple responsibility layers.Meeting these conditions allows you to accelerate development and make future maintenance much easier, saving time and money.

A wisely selected architecture together with platform-specific technology (like Swift for iOS or Kotlin for Android) will be best for resolving complex business problems.

This will allow you to avoid problems caused by the quirks of hybrid technologies. It will result in saving time and money in the long term.

Good architecture should be abstract so that it can be applied on various platforms, like iOS or Android. **One of the most crucial features of a good architecture is responsibility layer separation.** Such a division will be key to reducing errors.

One of the most popular multilayer architectures is a three-layer architecture. Having clearly defined architecture makes work easier and faster as staff have more autonomy over their work and the data flow. It makes testing more efficient and increases the quality of management of the overall product.



WHY COMPANIES UNDERESTIMATE MOBILE APP ARCHITECTURE

In recent years, the approach to mobile app architecture has improved. Developers pay more attention to proven patterns and standards, but some developers don't follow best practice. Many mobile apps have been created with low-quality source code, without architecture, or are based on antipatterns. Custom apps are hard to maintain and develop further. Products made with poor foundations therefore can drastically increase the time and cost of development.

WELL-SELECTED ARCHITECTURE FOR MOBILE APPS

Well-chosen architecture saves a lot of problems with the maintenance and development of a product. It also gives structure to the development process and the project. Having structure is very important during developing and maintaining a project, as it improves and fixes non-existent or poor architecture. This is why we believe that selecting appropriate architecture should be an integral part of the planning stage of every software project.

INTEGRATIONS

Mobile app integration is the process of connecting the apps on your mobile device to the existing systems to exchange data and optimize workflows. As companies continue to focus on digital transformation, mobile app integration will continue to be one of the best ways to demonstrate innovation across your organization.

BENEFITS AND ADVANTAGES OF MOBILE INTEGRATION

Mobile integration enables you to create a robust service framework with an easy-to-use interface so your team can work efficiently from anywhere. You can manage a complex ecosystem of digital tools in one easy-to-use interface. This minimizes the need to transfer data manually which reduces errors and delays.

USER CASES FOR MOBILE INTEGRATION

Mobile integration in phone systems allow you to hand off VoIP calls between a smartphone and a desk phone seamlessly. You can also access contacts across both devices, enjoy WIFI connectivity to untether workers from the desk phone, and charge mobile devices directly from the desk phone. You can connect your unified communications platform with other business apps to consolidate and manage your workflows.

ADVANCED FEATURES OF MOBILE INTEGRATION

Mobile integration delivers advanced telecommunications features, which allow you to:

• Seamlessly switch between VoIP desk phone and softphone app on your mobile device without dropping the calls.

• Route calls to multiple numbers on a single device to streamline communication flow.

• Create temporary numbers (for contractors, for example) and have the calls forwarded to their own devices.

• Execute policies and permissions, such as granular call control and call distribution from a centralized dashboard.

• Facilitate group collaboration with features such as video conferencing and screen sharing, no matter where team members are and what devices they use.

ACCEPTANCE CRITERIA

The purpose of acceptance criteria is to know when the user story is done from the product perspective. Acceptance criteria defines the functionality, behavior, and performance required by the features so that the product owner accepts them.

It defines what is to be done so that developers know when the user stories are complete.

User story example 1

As a user, I can change my subscription plan so that I can try other subscription plans in my budget.

Acceptance criteria example

• If a user is changing to a subscription with a higher value, the user value of the new subscription should be charged from the user's payment details immediately.

• If a user is changing to a subscription with a lower value, the new subscription becomes active at the end of the current subscription.

• When updating their subscription, the user should receive a confirmation email.

User story example 2

As a user, I want to sign up using my phone number so I can create an account on the platform.

Acceptance criteria example

• The user enters and verifies their phone number before proceeding to enter their first name, last name, and email.

• The user will see a countdown of 30 seconds on the verification page before being able to click on the 'try call' option.

• Upon verification, the user can enter the first name, last name, and email.

• The user device ID is stored in the database along with the user's information when completing registration.

With these criteria in the above two examples, it will be apparent to the development team what needs to be implemented to complete the user's story.

USER FEEDBACK

Uncovering what your customer wants from your app is challenging. The only way to fully understand what they want is to listen to their feedback.

How do app developers know what features to include, retain, or remove? How do they know what needs to be fixed or improved? How do they understand the motives of the app users? The answer to all these questions lies in getting feedback from app users and leveraging this feedback to improve customer experience.

BENEFITS OF USER FEEDBACK

1. Better app roadmap

Feedback plays an essential role in improving the current app and drafting the app's future roadmap. As an app developer, multiple features would be on the app roadmap, but the question is which features should be prioritized.

The solution to this is providing users with a feedback mechanism. This enables developers to understand which features should be retained, which need to be fixed, and which should be prioritized in future releases.

2. Reduce costs

With great apps, come great costs. This cost is financial, time, and development resources as every single app update or fix requires resources. The development team could spend considerable time on a single bug fix. The allocation of resources is vast as it can take months to deliver a new feature.

Imagine that, after spending months developing a new feature, it fails to meet the needs of your users.

The only way to stop wasting resources is through gathering quality feedback from your users. This enables you to allocate resources to create features that add value to the customer experience. Incorporating app feedback from a closed group of users before launch allows developers to get feedback based on a minimal viable product. They incorporate this feedback to create an app that meets users' expectations from the beginning.

3. Improve app-user connection

Feedback provides an opportunity to build a two-way communication channel with your users, which leads to better engagement. When a user feels that their input is integral to the development and improvement of the app, they will be more willing to provide feedback which leads to a better app-user connection.

APP MONITORING

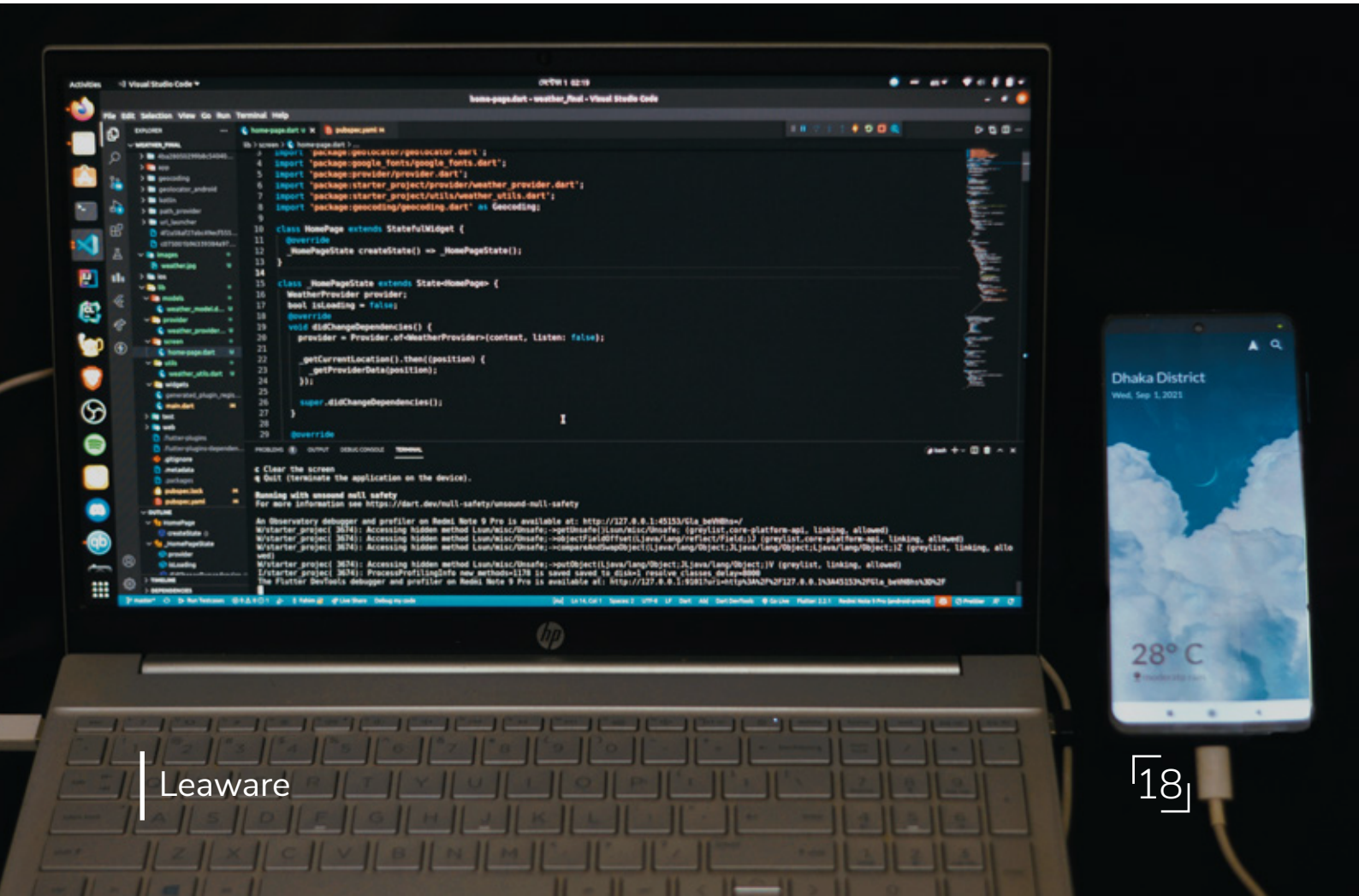
App monitoring alerts you when your main line of business apps, related databases, or email systems are not performing properly. App monitoring software gives you a visual dashboard to track usage, performance, and growth. All these are extremely important for capacity planning, meeting SLAs, and finding problems before they cause outages.

APP MONITORING OPTIONS

The app monitoring field is large. There are some mature, well-used open-source options, such as Nagios and Hyperic which provide powerful monitoring solutions for businesses. In addition to these open-source options, there are several commercial options available as well.

SolarWinds has a powerful APM module for their Orion integrated management system, which can monitor your apps without an agent installation. There are also options from ManageEngine, which works similarly to SolarWinds.

App monitoring is an essential piece of the puzzle of ensuring your infrastructure stays operational: you know how your apps are running, how they're being utilized, and how they're growing. With this data, you can plan equipment acquisition easily, find problems with your apps more quickly, and know that your clients can complete transactions smoothly.



DEVELOPMENT

The development stage starts early on. Once an idea gains some maturity in the conceptual stage, a working prototype is developed which validates functionality, assumptions, and gives an understanding of the scope of work.

As development progresses, the app goes through a set of stages. In the initial stage, the core functionality is not tested. The app is likely to be very buggy at this point and non-core functionality will not exist. In the second stage, much of the functionality proposed is incorporated. Ideally, the app has been lightly tested, but some issues could remain.

During this stage, the app is released to a certain group of external users for further testing. After the bugs in the second stage are fixed, the app will move to the deployment stage where it will be ready for release.

If you have a complex project where user requirements change regularly, make use of an agile methodology. This methodology helps with flexible planning, progressive development, early deployment, and constant improvements. A large app can be broken down into smaller modules and an agile methodology can be applied to each of these small parts.

SPRINTS

In mobile app development, a sprint is a set period during which specific work must be completed and made ready for review. Each sprint begins with a planning meeting.

During the meeting, the product owner (the person requesting the work) and the development team agree on what work will be accomplished during the sprint. The development team has the final say on determining how much work can be accomplished during the sprint and the product owner has the final say on what criteria needs to be met for the work to be approved and accepted.

The team will determine what work will be delivered by the end of the sprint before the cycle starts. To provide the most value, the requested features and enhancements are broken down and prioritized into four categories:

- **Priority 1:** High Impact, low effort investment.
- **Priority 2:** High Impact, high effort investment.
- **Priority 3:** Low impact, low effort investment.
- **Priority 4:** Low impact, high effort investment.

Large workflows can be planned out over multiple sprint cycles, while simultaneously working to deliver smaller features at the end of every sprint. This way, even if a larger feature takes three or four sprints to accomplish, value-creating features or enhancements are regularly released to users.

VERSIONING

When a product is released, it should be allocated a version number. When new features are released, the version number increases. You may be wondering where exactly in the development process this happens and how it is tracked.

AIMS OF VERSIONING

Before getting to the when and how, let's start with why. The main aims are:

- **Everyone knows what features are available in the version of the app they have.**

- **Everyone knows what version of the app they have if they are reporting bugs or requesting support.**

DISPLAY THE VERSION NUMBER IN YOUR APP

For everyone to know what version number of the app they have, you need to display it to them in your user interface. The home screen, main menu, or information screen are suitable places to provide this information.

QUALITY

In mobile app development, it's a good idea to test early and often. Doing this will keep your final costs low. The further in you go into the development cycle, the fixing bugs can become costly. Refer to the original design and planning documents while trying various test cases.

App testing is vast, so make sure your team covers all the necessary facets of it. The app should be tested for usability, compatibility, security, interface checks, stress, and performance. In user acceptance testing, you discover whether your mobile app works for your intended users or not. To test this, give your app to a few people in your target audience and ask pertinent questions about its performance.

Once your app passes the user acceptance test, you know that your solution "works." To further test the product, you can make your app available for a beta trial, either by selecting those from previously identified groups or making it openly available for participants. The feedback you receive from beta users helps you to find out whether the app's functions operate well in a real-world situation.

TESTING STRATEGY

Cross-platform testing

Sometimes extra fixes are required if your app is interacting with other apps. Gathering and understanding project requirements, business objectives, various language platforms, and user needs are essential to creating an appropriate cross-platform testing strategy.

Feature functionality

Mobile apps usually interact with several features, both built into devices and built into the app. These interactions should be recorded and thoroughly tested. It's not necessary – and not advisable – to run functional testing across a plethora of mobile devices. Instead, test on a single device and then test platforms during compatibility testing.

TEST PLANS

A test plan is a document that describes the scope of testing, test strategy, objectives, effort, schedule, and resources required. It serves as a guide to testing throughout the development process. The scope of work is defined at the beginning of the testing process. A project team should clearly understand what features and functions there are to be tested and which ones are out of scope. To determine the scope of testing, the project specification, budget, and customer's requirements should be considered.

The main objective of unit testing is to verify whether every unit operates as intended. A function, procedure, method, or even the entire module can be considered a separate unit. Unit testing can be conducted manually, but automated testing is more common.

Entry criteria

- **The planning stage has finished.**
- **Testable units are available.**
- **All functional requirements have been defined.**
- **The unit testing environment has been set up.**

Exit criteria

- **All planned test cases have been covered.**
- **All bugs found have been reviewed.**
- **The performance of key modules have been tested.**

Logging tests and reporting

Developers fix defects found in each unit. If these defects are related to other modules and units, they must be reported.

DEPLOYMENT

Your app is ready to be fully released. Choose a date and create a formal launch. For different app stores, the policies of launching an app are different. Keep in mind that app development doesn't end when it launches. As users start to use your app, you will start receiving feedback this will need to be incorporated into future versions of the app.

Every app needs updates and new features. Typically, as soon as the first version of the app is released, a new development cycle begins, which makes budgeting the resources to maintain your product important. Apart from the money invested in building a digital product, keep in mind that it's a long-term commitment.

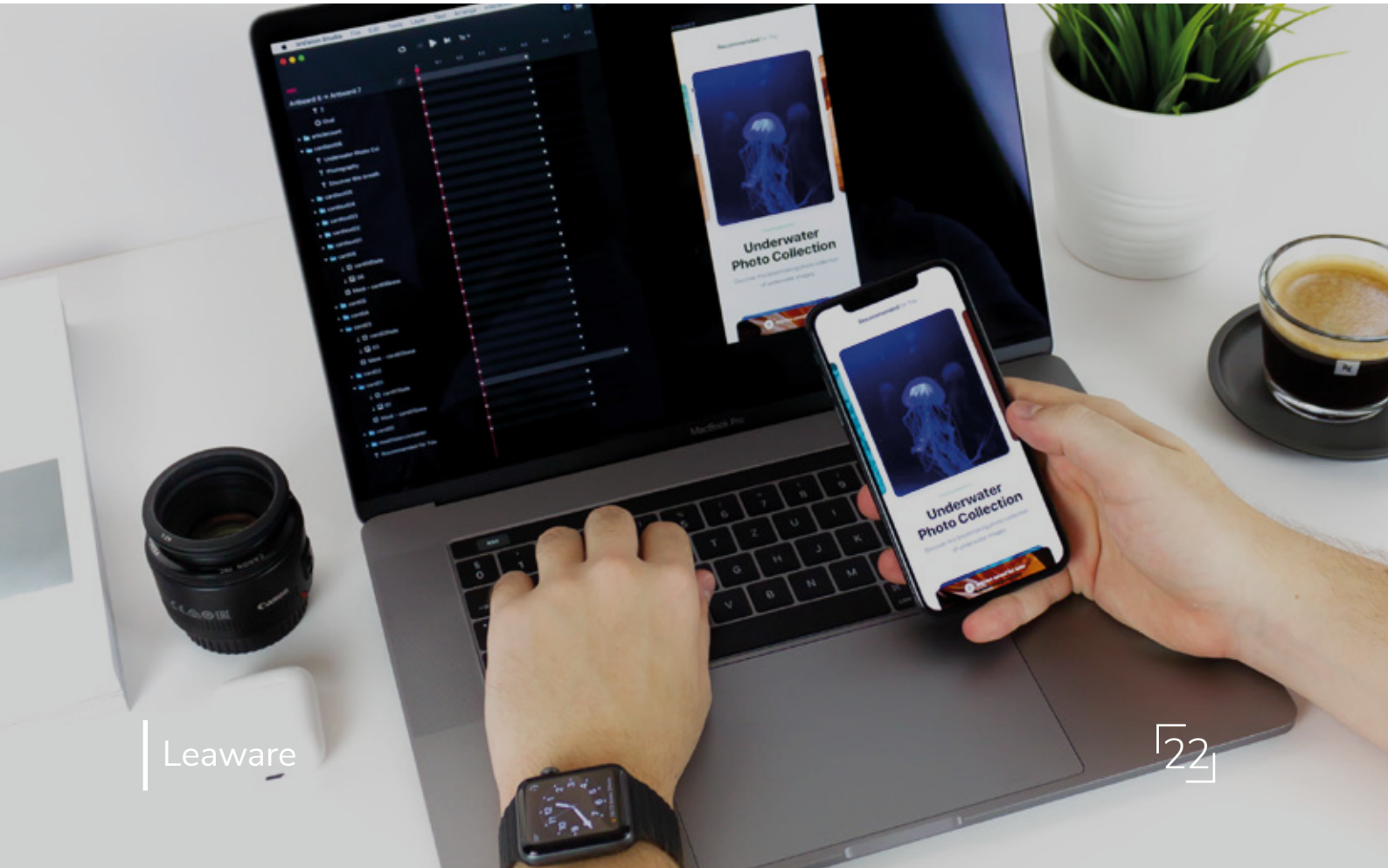
DOCUMENTATION

Process documentation is an internal, living document that details the tasks and steps needed to launch a new process. From something as simple as onboarding new staff to bigger goals like changing team structure, it's important to properly document and track the progress of new processes. You can also create process documents to streamline current processes. You may be surprised to learn just how many you have in your organization already, from deploying new tools to customer-facing communication.

As well as continuing to align teams, process documentation serves as a roadmap for team members which helps to clarify the steps needed to create new processes. It also eliminates confusion between team members, serving as a go-to resource to refer to.

PROCESS DOCUMENTATION VS. PROCESS MAPPING

While the two terms sound similar, there are some key differences between process documentation and process mapping. Process documentation focuses on creating a written document that outlines key details, while process mapping focuses on visualizing the process. While process documentation does include a visual representation, it's quite different from the in-depth visual of a process map.

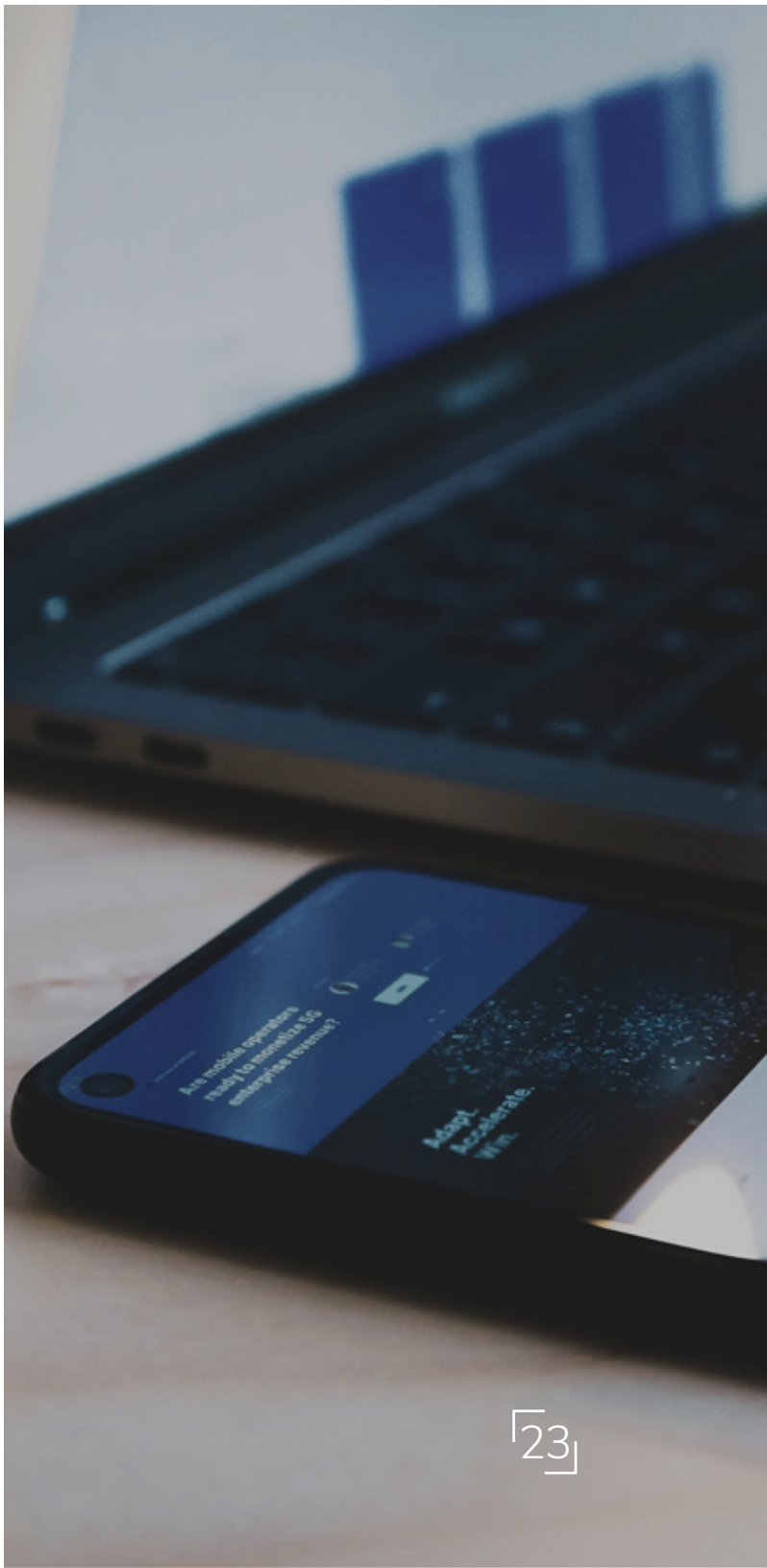


RELEASE ON PRODUCTION

There are only a couple of things that can be more frustrating in agile software development than when your release planning halts. One typical bottleneck is that product managers cannot work on the next release of an app that is being developed across different teams.

During release planning, you will soon realize that apps developed across multiple teams need an aligned approach to be released. This is where the concept of the “mobile release train” comes into play. By executing this approach, different developer teams can simultaneously release apps to different app stores.

The concept is nothing new and is part of the scaled agile framework. The mobile release train describes how to deliver software in a specific way. The mobile release train can be used by small and/or big distributed departments which work on one app, but in different teams.



PUBLISHING IN STORES

Google Play and the Apple App Store are the two largest platforms that distribute and promote apps. To avoid any issues and possible rejections, your apps should be built following the specific requirements of each of the stores. If you are ready to publish, chances are you have already built and tested your app.

Submitting an app to the Google Play store

Google Play is a library where users browse and search for apps. If a user opens your app listing, it will take them only a few seconds to decide whether it matches their needs.

You need to make your listing compelling to catch their attention and convince them to download your app. The text for your app listing should follow those listing criteria. These are all the details that are shown to customers on the Google Play store.

In the Google Play store, the short description is the first text users will see on the app listing. In the long description, describe your app accurately using the right keywords to catch your audience and optimize ASO for Google Play. Choose the correct graphic features and screenshots to reinforce your brand and show off your app's best features.

Once these steps have been completed, you can upload the binary file (the .apk file) and follow the guide on your Google console to roll out the release to publish our app. APK (Android Package Kit) is the file format used by the Android operating system to install and run apps. The APK file contains all the elements needed for your app to work on an Android device.

Submit the app to the App Store

It is highly recommended that you read the Apple Store Review guidelines before starting the submission process. Apple's review team reviews each app before approving them for release. Most Apple rejections are due to non-compliance with the guidelines. Before you submit your app for review, you will need to provide the following information:

- **Icon, app preview/ screenshots**

- **Metadata (the name of your app, its category, a detailed description, and additional keywords for ASO).**

Don't underestimate the importance of the description. The first three lines are the most important because this is what users can see without having to tap to read more. Be accurate, clear, and concise to transmit the core value of your app to the users.

MAINTENANCE AND SUPPORT OF PRODUCTION APPS

Like humans, apps need to visit the doctors (technical engineers) for a check-up. The maintenance process starts right after the user purchases particular software or technology.

To improve the performance and fault correction, utilize mobile app maintenance and improvement services. It helps the mobile app adjust to upgrades and environmental changes. When apps are continuously in action, maintenance is extremely important. For mobile app development companies who are still using legacy (traditional) technology, app and software maintenance are mandatory for them.

What are the different types of app maintenance?

Generally, there are different kinds of maintenance procedures and techniques provided by service providers to keep systems up to date. Below are key maintenance procedures:

1. Emergency maintenance

You cannot predict the problems and errors that can occur in a system. Whenever a company faces this kind of unpredictable problem, they initiate emergency maintenance and support. Emergency maintenance is one of the most common types of app maintenance as many users utilize it during crisis points. For instance, fixing an inactive internet connection in the office is an emergency maintenance situation in the corporate world.

2. Perfective maintenance

This is the implementation of new or changed user requirements in an app. In this kind of maintenance, developers do function modifications without affecting the regular behavior of the app. Perfective maintenance services make changes in the source code to bring new functionality to meet new requirements of users.

3. Adaptive maintenance

For this type of maintenance, changes are made to the app to keep them up to date with changes in the working environment such as hardware or the operating system. The term environment refers to the influences and working conditions that act on the system. The adaptive maintenance assures the compatibility of an app with the latest version of the operating system. For instance, Bluetooth drivers for Windows 8 and Windows 10 are different.

4. Preventive Maintenance

This kind of maintenance performs actions to stop the occurrence of errors. Preventive maintenance helps to lower the complexity of apps, improving the app's maintainability and understandability.

FIXING BUGS AND IMPROVEMENTS

Code optimization, code restructuring, and document updating include preventive maintenance practices. Code optimization includes changes in programs that use storage space for faster execution, while code restructuring involves structure modification to reduce the complexity of source code.

DEVELOPING NEW FEATURES

Corrective maintenance

Corrective maintenance repairs the faults and defects found in day-to-day app functions. It deals with the errors related to app logic, design, and coding. These types of errors are called residual errors. The bug reported by users in the app generally initiates corrective maintenance.

SUMMARY

The world moves with technology, which is always evolving. Entering the new digital age requires having the knowledge that is necessary to develop an app.

The wrong choice of the development team can mean:

-The app may have bugs that affect the app's performance.

-Choosing technology that is not suitable for the digital product.

-Expensive development.

-The potential need to reprocess the app's code.

Examples of mobile apps that failed are:

-Vine: There was no concern for users' needs and they lacked a clear plan.

-Hailo: A taxi app in the UK which had poor market research and ill-advised business decisions. There was little compatibility when operational in New York.

-Rdio: A music streaming app from the creators of Skype, which had a poor concept, inferior marketing, and a weak plan for growth management.

Be well prepared

The key is management and planning when developing the digital product. Be clear about the business objective and establish solid foundations in market research to ensure that what you want to solve is aligned with the needs of potential users.

Choose the right mobile technology

To get the highest quality and performance from your app, be sure to choose technologies that are compatible with the development.

Mobile app categories:

-Native: This is for creating apps dedicated to working on mobile operating systems, such as Netflix, Evernote, Uber, LinkedIn, and WhatsApp.

-Cross-platform technologies: These create mobile apps that can run on many platforms. Examples of cross-platform mobile technologies are Xamarin, React Native, and Flutter.

The architecture of mobile apps

The architecture of mobile apps describe how elements of the solution work with structural elements. The better built and constituted it is, the higher the quality of the product.

Mobile devices capabilities

Consider using a server to provide online/offline support rather than use the app's resources. This is to ensure smoother operation.

Data cache

It is very convenient for the user to be able to use the app without internet access.

Push notifications

They can be from the server promoting something new or local, like a reminder.

Minimum version of operating system

The functionality of the app must be in line with current trends and be compatible with the operating system.

Analysis and design

Align the design and development of the product to the user's needs and ensure that it is updated in line with current trends. Be innovative and create intuitive designs.

Mockups

Mockups include the design of the app and it shows how the app will look. The model design can come from the same developer using specialized graphic design software.

Architecture

Combining technology with the app interface and the correct operating system ensures the good architecture of your digital product. One of the most crucial features of a good architecture is responsibility layer separation. Bad digital product architecture results in delays and increased costs for the development and maintenance of the code.

Integrations

This is how different apps of your digital product's system work together and complement functions.

User feedback

The most appropriate person to provide feedback about the functionalities of your digital product is the user. With this feedback, you can manage changes that will improve the user experience. Benefits like reduced development costs improve the content in the app.

App monitoring

This ensures the functionality of your app is consistent with the initial and its development is not corrupted.

The process of developing a digital product begins when the bases are completely defined and customer needs have been established. Customer interaction during development is extremely important as they approve the stages of product development and determine whether the product meets their expectations. The developer must ensure that the operation is complete and successful. The quality of the product must always be considered and bugs must be resolved. This means that customers will not encounter coding problems and the app will flow better. The tests must have a plan and a scope defined by the developer.

Documentation is extremely important to record as it evidences results and helps the product to improve over time.

When launching a product, the release date and the release platform must be planned. It is important to remember that development is a continuous process and customer feedback can be given at any point. This is done to ensure the continuous operation of the digital product.

ABOUT THE AUTHORS

Since 2010, LEAWARE has been helping start-ups and already established companies build and develop their digital solutions, making them grow, boost their businesses, and succeed.

Tom Soroka
Leaware Founder
Business Development Manager

Damian Wasilewski
Project Manager
Business Development Manager

Carlos Lopes
Marketing Specialist
Business Development Manager

LEAWARE

Contact us
ask@leaware.com
Visit our website
www.leaware.com
Or see our Clutch profile
Clutch
Follow us on social media

Thanks for reading!

To receive more of our "Mobile Apps For Non-Tech People" scan the QR code below.

We build the right software.
We build the software right.

LEAWARE