outsystems

The Digital Transformation
Foundation Playbook

Accelerating the Digital Journey
With a Low-Code Digital Factory

Contents

Introduction

> *Every successful organization has to make the transition from a world defined primarily by repetition to one primarily defined by change. This is the biggest transformation in the structure of how humans work together since the Agricultural Revolution.*

—Bill Drayton, Social Entrepreneur

Who Should Read This Guide?

If you believe low-code offers an interesting opportunity—perhaps you have already started your low-code journey—this guide will help you understand how to take the next step. It is based on the experiences of OutSystems customers just like you who have been coming to us to resolve their challenges since 2001 and the solutions we developed together.

So, if you're ready to try a methodology that has been time-tested and proven by customers in almost every industry, this guide is what you're looking for.

What Is This Guide For?

This guide introduces a new methodology called a low-code digital factory that organizations can apply to their digital transformation initiatives from beginning to end. It walks you through, in detail, the steps you can take to address three concepts that are at the core of any digital transformation initiative: use cases for digital transformation, a framework for change, and the stages of the digital journey.

Use Cases for Digital Transformation

There is a wide range of low-code platforms available for facilitating digital transformation. This complicates most digital transformation efforts. Often, digital transformation is put into a bi-modal box or confined to a specific project like migrating data, even though the opportunity is much larger. In fact, there are three distinct use cases for digital transformation: digital operations, digital experiences, and digital core.

Digital Operations

You can start your digital transformation from within. These are the applications that support your internal business operations. The goal is to replace what has been cobbled together and what relies on manual effort, email, spreadsheets, and so forth. The typical solutions are dashboards, workflows, web portals, mobile applications, and small-to-medium-sized database apps. Digital operations needs include the rapid development of mobile and web responsive apps, simple development of forms and workflows, fast integration with systems of record, rapid change and deployment, and built-in productivity insights and dashboards for managers.
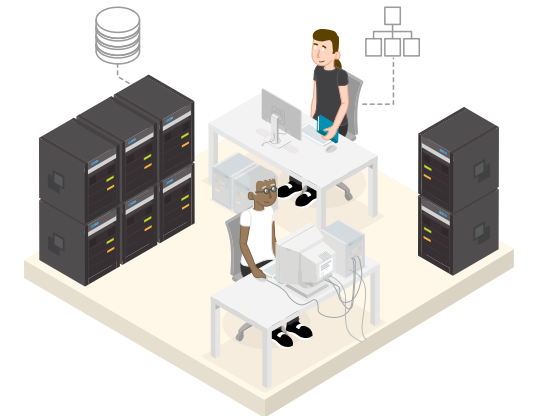
Digital Experiences

Another use case is tackling what customers experience digitally. The typical solutions are customer mobile apps, customer portals, and business portals and can include field operations apps and even apps for sporting events. Building digital experiences requires rapid prototyping and development, pixel-perfect design, massive scalability, and security, including for offline data. In addition, these experiences are instrumented so you have insights for your continuous improvement efforts.
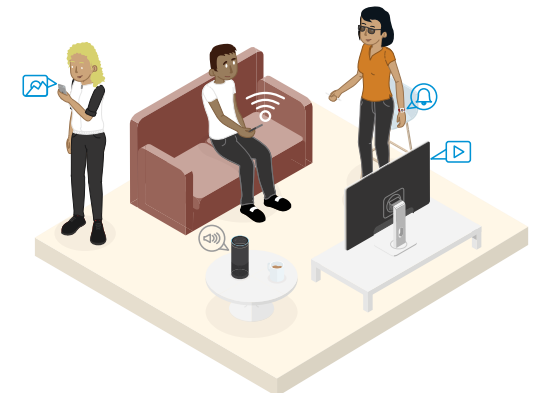
Digital Core

The final use case for digital transformation addresses the combination of aging systems and highly customized packages. These core systems are not always able to keep up with the demands of the business. Replacing them could take several years. In these situations, hardly anything is typical, but a few examples of what our customers have built are terminal management systems in the logistics and distribution industry, ERP systems in the consumer products industry, and clinical trial management in the life sciences sector.

Concept 1:

Three high-level use cases represent the scope of what can be accomplished with a low-code platform: digital operations, digital experiences, and digital core.

A Framework for Change

Digital transformation for any organization is a significant undertaking that puts an end to "business as usual." Such upheaval is best addressed by a framework called the low-code digital factory, which balances the structure that prevents your initiatives from hitting a wall with the agility that enables you to respond to change. This framework has four dimensions: structure, talent, ecosystem, and process (STEP).
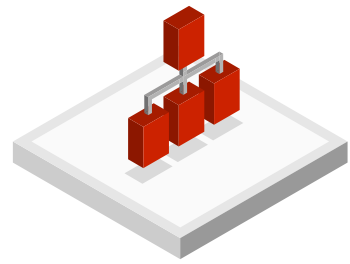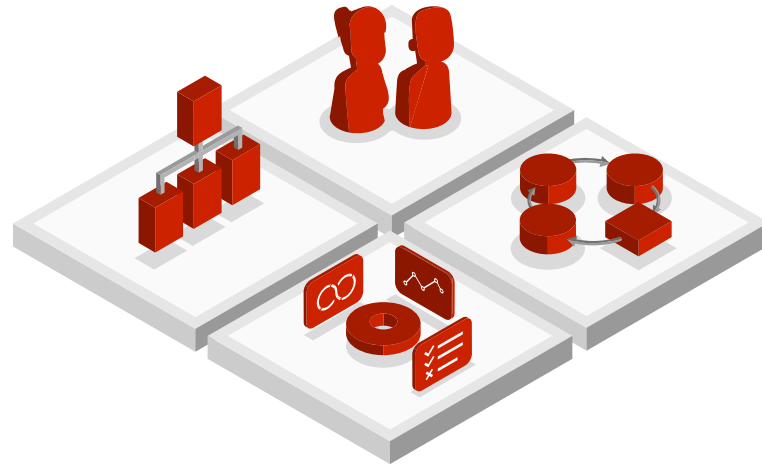
Structure

The digital factory offers an organizational structure designed specifically for successful low-code development and delivery. It addresses questions of team size, key team roles, where architecture fits, governance, and testing.
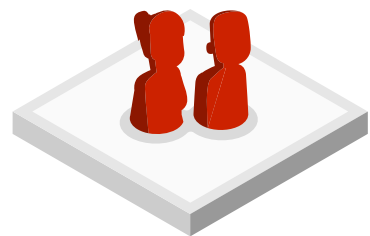
Talent

The promise of low-code is that organizations can build sophisticated solutions—even consumer mobile experiences—without the need for highly specialized resources. The talent dimension of the digital factory offers a careful approach for identifying and sourcing the roles for successful delivery of low-code projects.
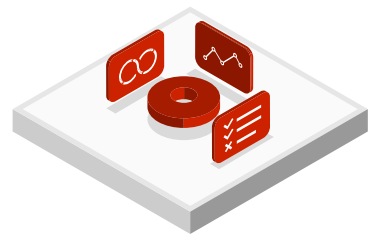
Ecosystem

Low-code platforms do not exist in isolation. They integrate with systems of record, data services, service buses, cloud services, lifecycle, DevOps tools, and more. The ecosystem dimension provides guidance for considering the sequence for tackling integration.
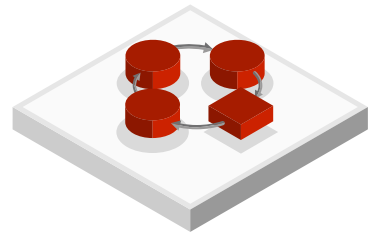
Process

No change management framework would be complete without consideration of the processes that need to be modified or established to support new ways of delivering software. The process dimension of the digital factory prescribes how different parts of an organization should interact, where low-code fits with agile, and how business and IT work together to ensure success.

Stages of the Digital Journey

There are three digital transformation stages. The low-code digital factory adapts for each of them, so moving successfully through them is easy.

Foundation

This stage focuses on enabling the first project team and successfully delivering the first project, which can address a digital operations or a digital experience use case. Getting an early win is critical because it builds confidence in low-code and paves the way for subsequent projects.
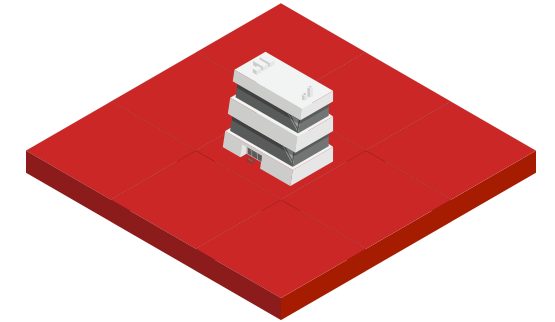
Center of Excellence

As the first team successfully delivers more projects, the natural progression is to increase or scale the use of low-code. Building a low-code center of excellence enables more teams, makes it easier to integrate low-code with other internal platforms, and adds governance as a focus. During this stage, organizations establish a self-sustaining low-code capability.
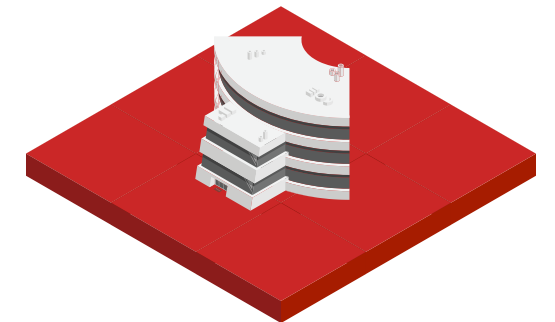
Full Digital Enterprise

With the appropriate supporting structure in place, organizations can move and grow low-code across multiple lines of business. The journey to becoming a full digital enterprise allows you to embrace all of the use cases for low-code. With a truly agile digital core, businesses are equipped to adapt quickly to changing market conditions and competitive pressures.
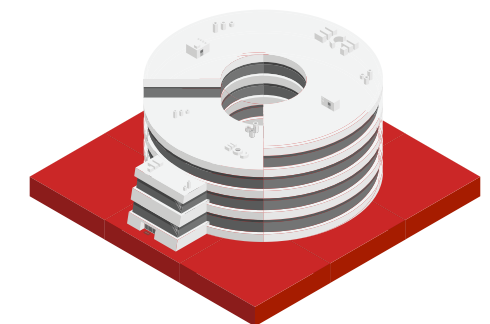
Concept 2:

A low-code digital factory promotes the successful adoption of low-code technology. It provides recommendations for organizational structure, identifying and sourcing the right talent, fitting low-code into the broader technology ecosystem, and optimizing processes to take advantage of the benefits low-code provides.

Concept 3:

Organizations go through three distinct stages of low-code maturity. At the foundation level, the focus is on enabling the first team and delivering the first successful project. At the center of excellence stage an organization opens low-code to multiple teams and creates a self-sustaining low-code capability. The full digital enterprise scales the low-code capability across multiple lines of business.

A Firm Foundation:
The Low-Code Digital
Factory Setup

The best way to ensure the adoption of almost any kind of new methodology is to show that it works.

The foundation stage of digital transformation will get a lot of people onboard and silence a lot of critics. It produces a working application that can make tasks or life easier--or even impress consumers. And, it does it fast. The low-code digital factory will enable you to set that firm foundation and speed you to your successful first app.

Setting up your low-code digital factory for the foundation stage incorporates the STEP dimensions: structure, talent, ecosystem, and process. We're going to walk you through those dimensions so you'll be ready to see results in a short period of time. The methods described take into account the digital operations and the digital experiences use cases, because the foundation stage is applicable to both.
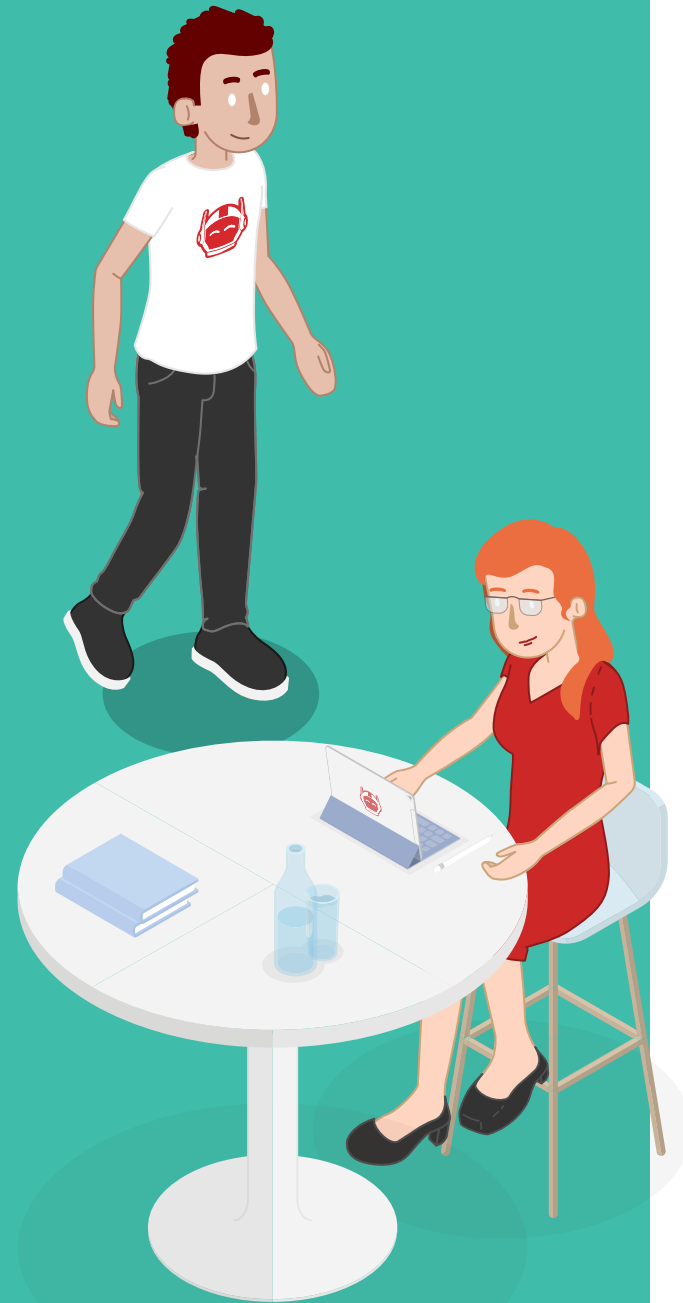
Structure

> *Plot is what happens in your story. Every story needs structure, just as every body needs a skeleton. It is how you 'flesh out and clothe' your structure that makes each story unique."*
>
> —Caroline Lawrence, Author

Using low-code to accelerate digital transformation will impact existing approaches to building software. The low-code digital factory offers an optimal organizational structure for supporting low-code application development and delivery. However, before you start looking for the ideal developers to work on the project, you need to have an adoption team in place at the outset, one that includes support from the very top, a champion to keep enthusiasm high, and an expert who has done it all before.

Adoption Team

Change doesn't happen on its own. Not only that, but it creates discomfort because people are no longer doing things the way they've always been done. Therefore, there needs to be a force to push change across an organization (picture pushing a ball up a hill). Without this force, also known as an adoption team, the smallest setback can result in an entire program being scrapped. The adoption team is composed of a few key roles.
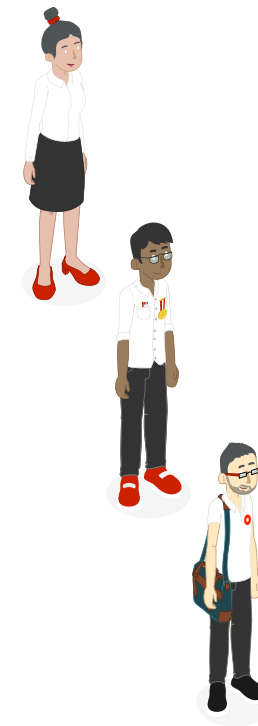
Executive Sponsors
Organizational change requires support from the very top. An executive sponsor who can articulate a compelling vision and find the necessary funding is an essential ingredient of any change program.

Adoption Champion
If the digital transformation journey were a movie, the adoption champion would be the hero, driving the success of the program and building consensus. Enthusiasm and resilience are required traits.

Adoption Expert (OutSystems or OutSystems Partner)
Every hero needs guidance from somebody who has been on the journey before and can provide valuable advice and direction. Working hand in hand with the adoption champion, the adoption expert's experience and support ensure unwelcome surprises are avoided.

The good news is that, once the ball has been pushed up the hill, it starts rolling on its own and picking up speed.
This is the nature of change and the reason that adopting low-code across an organization needs a focused team at the outset.

outsystems

The Smallest Low-Code Digital Factory

While every organization has its own unique structure, there are common functions involved in the delivery of software, and they can be broken down into six distinct areas. Since building an app at the foundation stage involves a single team, the starting point for a low-code digital factory is simple. And, you can ensure the successful delivery of the first app if you follow recommendations such as those in this diagram.

The product owner acts as a proxy between the development team and the business user. However, formal project management practices are typically not needed at this stage. Instead, business users are more likely to be directly involved with the development team.

The tech lead addresses the architectural and operational tasks, even though they are generally limited at this stage. Tech leads should know how to design an app-level architecture, control code quality with peer code reviews, and understand how technical features map to business requirements.

Business

Demand & Governance

Development

Operations

User Experience

Architecture

Business User	Product Owner	Tech Lead	Developer	Adoption Champion	Adoption Expert

Using low-code, a development team of just three people is typically able to deliver 6 new internal apps or major app releases in a year. It takes only 3 weeks to get this team up and running.

3 Developers

6 Apps

3 Weeks

Tailoring Structure for Digital Experiences

For the digital experience use case, the foundation stage requires additional resources. At this point, the team structure changes to include more roles related to making sure the app is intuitive, comfortable, and brilliant for internal users, consumers, or both.

A Customer-Centric Mindset

You can't focus on the customer without understanding the mindset. Identify and include key users throughout the design and development process and ask for their comments and opinions as new solutions are delivered.

The Expanded Role of Testing

First impressions make the difference between success and failure for an app. Performance and cross-device testing are critical activities that need to be planned for. Adding a dedicated and knowledgeable testing resource to the team and giving them the right tools to   ensure success.
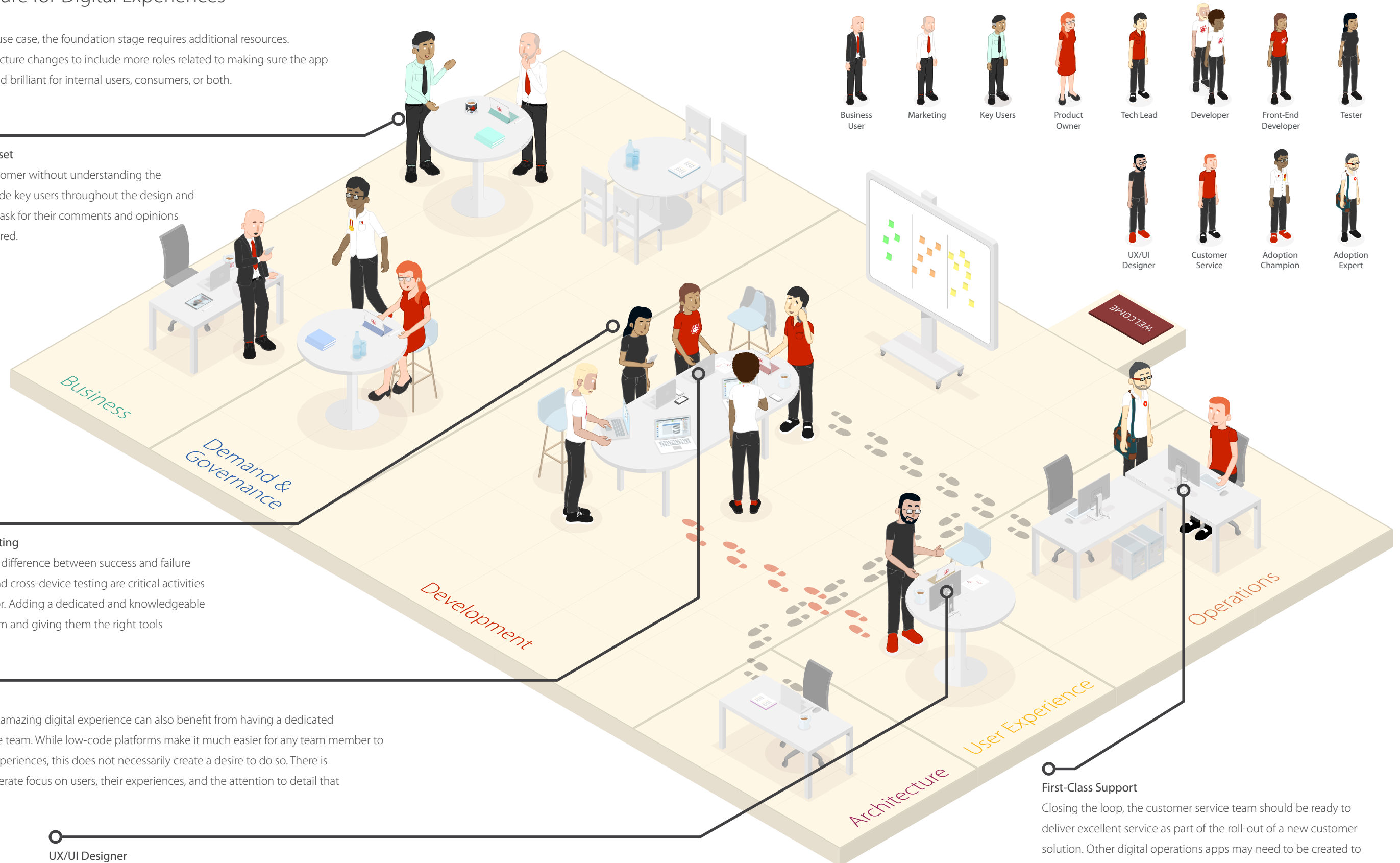
Front-End Developer

The process of creating an amazing digital experience can also benefit from having a dedicated front-end developer on the team. While low-code platforms make it much easier for any team member to build sophisticated user experiences, this does not necessarily create a desire to do so. There is no replacement for a deliberate focus on users, their experiences, and the attention to detail that is part of this process.

UX/UI Designer

Understanding the customer journey and developing an optimized experience requires specialized skills. Adding a UX/UI designer to the team structure ensures the needs of the customer are being considered and that the result exceeds their expectations.

First-Class Support

Closing the loop, the customer service team should be ready to deliver excellent service as part of the roll-out of a new customer solution. Other digital operations apps may need to be created to support new digital experiences.

Business User
Marketing
Key Users
Product Owner
Tech Lead
Developer
Front-End Developer
Tester
UX/UI Designer
Customer Service
Adoption Champion
Adoption Expert

Business

Demand & Governance

Development

Architecture

User Experience

Operations

WELCOME

Talent

"*Talent is the No. 1 priority for a CEO. You think it's about vision and strategy, but you have to get the right people first.*"
—Andrea Jung, President and CEO of Grameen America

The low-code market has grown dramatically, but is still relatively less understood when compared with traditional coding. Identifying talent, either internally or in the job market, is difficult without a better understanding of the skills required. To help organizations know what to look for we present profiles for the key low-code roles. Refer to the appendix for descriptions of the different competency areas and skill ratings used in this section.

Tailoring Talent for a Digital Experience

As you move to creating digital experiences, other roles become important to your team, such as user experience and testing. Specialized skills are needed to ensure new digital experiences are robust and make an impression on customers. People in your organization or on your team might already have these skills, which means you can transfer them to low-code, but you need to resist the temptation to have developers on the team try to fill multiple roles. The additional roles should be fully focused on the digital experience task at hand and are best handled by resources with the right background. Here are the profiles that are needed.

Foundation-Level Resources

The skills necessary for starting the low-code journey are likely to already exist in your current teams. The following profiles act as a guide for mapping capabilities and skill levels to team members. In addition, low-code platform vendors generally provide online learning materials to get teams up to speed in just a few weeks, along with programs to ensure the necessary support is available. By taking advantage of these and the fact that it takes less time to learn low-code technology, teams often begin delivering business value in a matter of weeks.

Developers

Skills: Analysis & Design, Web App Architecture, User Experience, SQL Programming, Development Languages, UI Development Languages, Integrations, Performance, Troubleshooting & Debug

Developers work under the supervision of an experienced tech lead to design, build, and test collaborative applications that provide business solutions.

In the market, you should target developers with more than 3 years of software development experience, using .Net, Java, or Javascript.

This role will master OutSystems technology combined with modern software engineering practices for developing innovative web and mobile business applications.

Tech Lead

Skills: Analysis & Design, Web App Architecture, User Experience, SQL Programming, Development Languages, UI Development Languages, Integrations, Performance, Troubleshooting & Debug

The tech lead will drive the implementation of innovative solutions that solve complex technical and business issues. The tech lead uses an agile approach to establish the connection between technical and business teams.

Target professionals with 2+ years of team leadership and customer-facing experience.

Where you have experienced developers and an architecture team already in place, explore promoting internally. If going to the market, target senior developers looking for leadership opportunities.

UX/UI Designer

Skills: User Research, Information Architecture, Low-Fidelity Mockup, Visual Design, User Testing, Customer Management, Mobile, Usability, Business Process, Customer Journey

The UX/UI designer provides a clickable prototype that visually demonstrates the look and feel of the user experience, thereby increasing the understanding of the application's value. Most importantly, this role maps the customer journey.

Good UX UI designers typically have analytical minds and are pragmatic thinkers. You can look for designers with a UI and UX background, but you can also look for candidates with psychology or journalism degrees because they tend to be investigators who ask questions.

Front-End Developer

Skills: Analysis & Design, Web App Architecture, User Experience, SQL Programming, Development Languages, UI Development Languages, Integrations, Performance, Troubleshooting & Debug

The front-end developer is responsible for implementing user interfaces and adding interactive elements to applications that will engage the user. They should have experience with building scalable front-ends.

Your target should be seasoned front-end and UI development professionals, with skills in HTML, CSS, JavaScript, and responsive web design; be well-versed in the technology and tools used for digital experiences; and be able to communicate effectively with developers.

Tester

Skills: Analysis & Design, Web App Architecture, User Experience, SQL Programming, Development Languages, UI Development Languages, Integrations, Performance, Troubleshooting & Debug

Testers define test strategies, write test cases, and use automated and manual methods to test applications.

Target professionals with at least 3 years of software QA, test planning, and test writing experience, at least 1 year of test automation experience, familiarity with continuous integration testing

frameworks (for example, Selenium, Robot, Jenkins and Apache JMeter), and knowledge of device farms.

A good tester should be curious, detailed oriented, skeptical, and customer-centric with strong interpersonal skills.

Ecosystem

Technological change is never an isolated phenomenon. This revolution takes place inside a complex ecosystem which comprises business, governmental and societal dimensions. To make a country fit for the new type of innovation-driven competition, the whole ecosystem has to be considered.

—Klaus Schwab, Exec. Chairman of the World Economic Forum

The successful adoption of any new technology depends on how well it fits into the existing ecosystem of tools. An amazing solution that doesn't integrate with a key internal system won't be used widely or regularly. Avoiding becoming the next shelfware victim requires a deliberate strategy. As with organizational structure, however, starting small early on reduces the time it takes to get value from the solution.

The Smallest Low-Code Ecosystem

This diagram maps the dependencies in an ecosystem for the foundation stage when OutSystems is at the center. It illustrates a simple approach whereby any of the tools illustrated integrate with OutSystems and are readily available as downloads.

Proactive Monitoring
From business SLAs to network performance issues, OutSystems has built-in monitoring capabilities that identify issues before they become a problem.

Extra Automation
The OutSystems behavior-driven development (BDD) component rounds out testing needs so you have everything required for internal app development.

Tracking Demand
While built-in capabilities provide enough features for most project tracking needs, using JIRA with the pre-built connector adds an extra level for those who need it.

Simple DevOps
At the foundation level, using built-in deployment capabilities means no delays in getting the first app out. You can also connect JIRA to OutSystems to capture app feedback, closing the loop with the user.

Analytics · Agile Project Management (Jira Software) · Test Automation (BDD Framework) · Code Analysis · Application Lifecycle · Accelerators · Enterprise Theme · Integration Connectors · User Management · App Feedback · Case Management (Jira Software) · Infrastructure Monitoring (Infra Monitor)

Business · Demand & Governance · Development, Arch & UX · Operations

Expanding the Ecosystem for Digital Experience

The dependencies expand when addressing user experience. The following diagram shows tools for that expanded ecosystem based on the successes of a wide range of customers. OutSystems integrates with or has connectors to these tools.
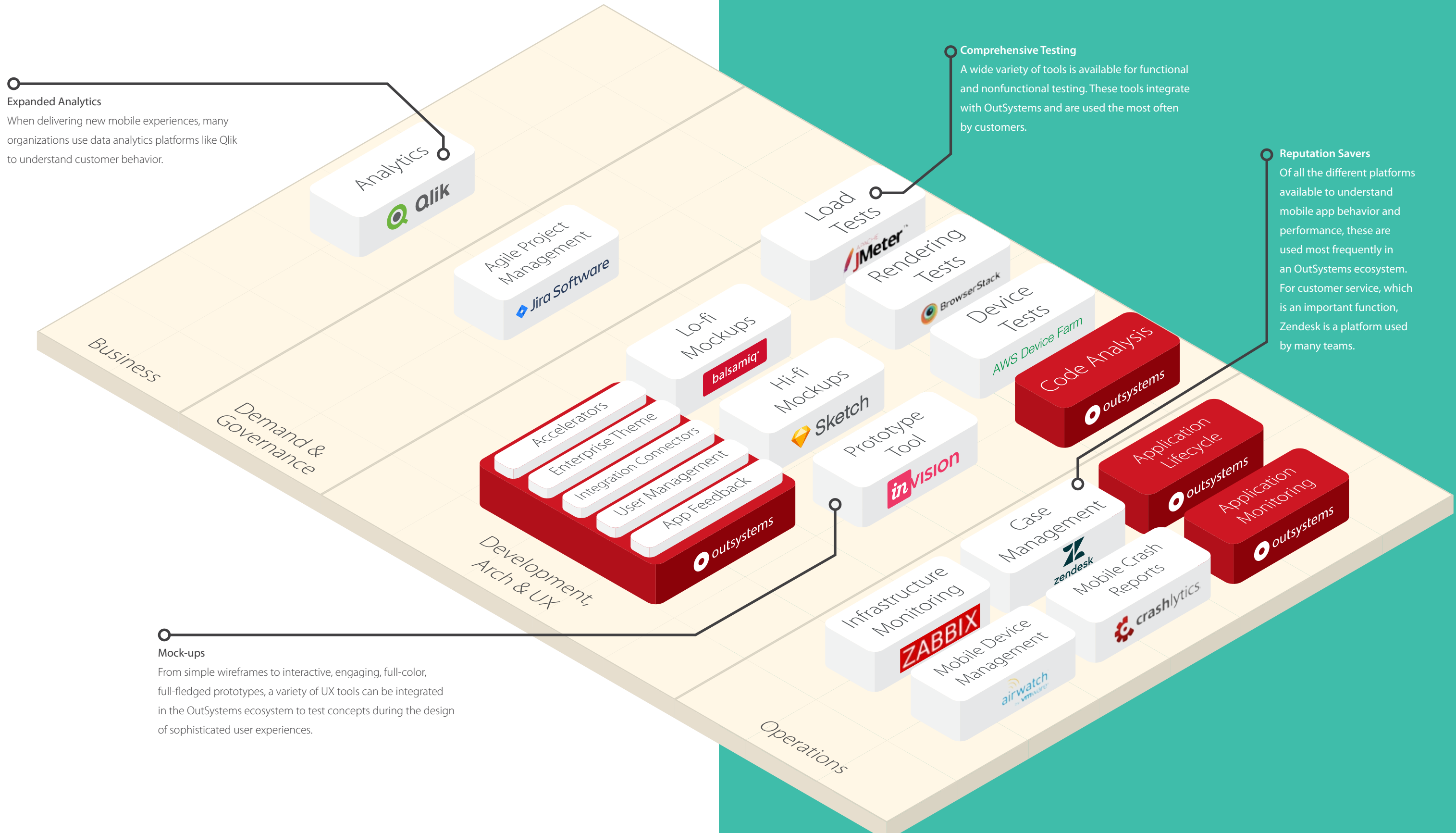
Expanded Analytics

When delivering new mobile experiences, many organizations use data analytics platforms like Qlik to understand customer behavior.

Comprehensive Testing

A wide variety of tools is available for functional and nonfunctional testing. These tools integrate with OutSystems and are used the most often by customers.

Reputation Savers

Of all the different platforms available to understand mobile app behavior and performance, these are used most frequently in an OutSystems ecosystem. For customer service, which is an important function, Zendesk is a platform used by many teams.

Mock-ups

From simple wireframes to interactive, engaging, full-color, full-fledged prototypes, a variety of UX tools can be integrated in the OutSystems ecosystem to test concepts during the design of sophisticated user experiences.

Business

Demand & Governance

Development, Arch & UX

Operations

Analytics — Qlik

Agile Project Management — Jira Software

Load Tests — JMeter

Rendering Tests — BrowserStack

Device Tests — AWS Device Farm

Lo-fi Mockups — balsamiq

Hi-fi Mockups — Sketch

Prototype Tool — inVision

Code Analysis — outsystems

Application Lifecycle — outsystems

Application Monitoring — outsystems

Accelerators

Enterprise Theme

Integration Connectors

User Management

App Feedback — outsystems

Case Management — zendesk

Mobile Crash Reports — crashlytics

Infrastructure Monitoring — ZABBIX

Mobile Device Management — airwatch

Process

"There is nothing so useless as doing efficiently that which should not be done at all."

—Peter F. Drucker, Top Management Thinker

Delivering with low-code is different from traditional hand-coding. So, a different process is needed to help your team focus on doing the right things at the right time. We begin with a process flow optimized for delivering operational apps, broken down into four categories. Activities are noted with who is responsible for completing the task, who has final accountability for the task completion, and who is consulted along the way. The appendix provides further detail for each activity in the flow.

Delivery
By enabling all layers of an application to be developed with a consistent visual approach and simplifying deployments to a single-click, the promise of cross-functional agile teams can be realized.

Run
Getting your app into production is a milestone that should be celebrated, but things don't stop there. Promoting your app, getting feedback, troubleshooting, and evaluating impact to the business are an important part of making sure your app is successful.

Demand
While project management activities are less formal at the foundation level, it's important to consider the value of new solutions and avoid building apps for the sake of it. Ideation activities identify the most valuable features and the effort required to build an app versus the value it offers.

Setup
At the foundation level, the setup requirements can all be accomplished in a matter of just a few weeks. The focus is on getting the team ready, selecting the app to be developed, ensuring the architecture is sufficient, and setting up the release pipeline.

Legend:
- **R** Responsible
- **A** Accountable
- **C** Consulted
- **i** Informed

Roles: Business User, Product Owner, Tech Lead, Adoption Expert, Adoption Champion

Setup
- Choosing and Training Demand Resource
- Choosing and Training Delivery Resources
- Backlog Analysis and 1st App Selection
- Reference Architecture Design
- Application Lifecycle Management Setup
- Infrastructure Provisioning

Demand
- Application Ideation
- Business Case and Metrics Definition
- Application Estimation

Delivery
- Application Initiation
- Application Architecture design
- Iteration Planning
- Iteration Delivery
- Iteration Tests
- Production Roll-out

Iteration 1..N

Run
- Production Release
- App Promotion
- Application Feedback Gathering
- App Troubleshooting
- Evaluate Business Case Metrics
- Portfolio Review

outsystems

Adjusting for Digital Experiences

At the foundation level for a digital experience use case, the process flow changes to include activities specific to delighting users.
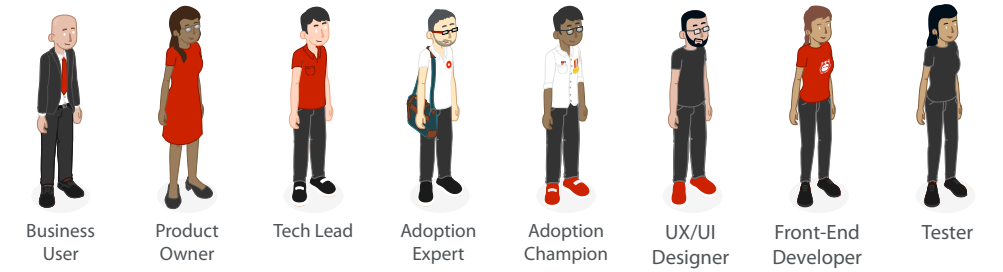
Developing the MVP

Delivering a minimum viable product (MVP) as a starting point is a lean startup concept. It is critical for new digital experiences. Validation and early feedback shape the evolution of the MVP. The result is a final product that targets customers, employees, or partners with greater precision and effectiveness.

R Responsible
A Accountable
C Consulted
i Informed

Business User • Product Owner • Tech Lead • Adoption Expert • Adoption Champion • UX/UI Designer • Front-End Developer • Tester

Setup

- Choosing and Training Demand Resources
- Choosing and Training Delivery Resources
- Reference Architecture Design
- Infrastructure Design
- Infrastructure Provisioning
- ALM Setup

Demand

- Application Ideation
- Business Case and Metrics Definition
- Application Estimation

Delivery

MVP Definition

- Mobile Application Initiation
- Mockups Low and High Fidelity
- Customer Journey
- Implement Look and Feel

Go/No Go

Iteration 1..N

- Architecture Design
- Application Architecture Design
- Iteration Planning
- Iteration Delivery
- Iteration Tests

- Quality Assurance
- Functional Testing
- Device Testing
- Load Testing
- Field Testing
- Production Roll-out

Test, Test, and Test Some More

When testing new experiences that run on a mobile device, you need to consider the variety of devices, network conditions, high traffic volumes, and offline behavior. In other words, test anything that might create a negative experience for your customer.

Run

- Evaluate Business Case Metrics
- Adoption Monitoring and Tuning
- App Troubleshooting
- Store Review Monitoring
- Performance Monitoring
- Store Release
- Production Release

Protecting Your Reputation

From bad app store ratings to unhappy social media posts, it is more important than ever to review how a product is being received in the market. The ability to adapt solutions quickly is only effective with an established feedback loop that considers a wide variety of sources.

Summary

Inevitably, in any discussion about digital transformation or low-code platforms, at least one person will ask "what's in it for us?" The best way to answer that question is to demonstrate value as quickly as possible.

Many companies who have adopted low-code have stories that include a statement like this: "We started with a small project, and we were amazed at how little time it took to get a working app. After seeing what we could do, we decided to move on to bigger projects."
They give examples of people in their companies who were skeptical at first and then won over by a quick success with lasting impact.
Or, they talk about how they'll never forget developing a critical application in just 6 weeks, especially one that used to take 6 months.

The foundation stage is based on every success story we know.
It's about keeping it simple and starting small so that everyone can see how quick and easy it is to complete a project. So, you set up your low-code digital factory to address this stage and so that it's ready to adapt as your organization moves to other stages. In other words, you'll be in the position to get started on a project like a well-oiled machine and watch it go live in a few weeks.

It's a powerful answer to "what's in it for us?"

What's Next?

This is just the beginning of your digital transformation journey supported by a low-code digital factory. In our next playbook, we cover the topic of establishing a center of excellence for the three use cases and how to adapt your low-code digital factory for that next step.

Appendix

Competence Skill Level Breakdown

For each of the low-code roles described in the talent section, there is an assessment of the required competence on a 5-point scale according to the following chart:

Skill Level: 0	*Not Applicable*

This competency is not applicable to the role.

Skill Level: 1	*Basic Knowledge*

There's a common knowledge or an understanding of basic techniques and concepts.

Focus is on learning.

Skill Level: 2	*Limited Experience*

Able to understand and discuss terminology, concepts, principles, and issues related to this competency.

Has a level of experience gained in a classroom, experimental scenarios, or both—or as a trainee on the job.

Is expected to need help with this skill.

Focus is on developing through on-the-job experience.

Skill Level: 3	*Intermediate*

Able to successfully complete tasks in this competency.

Has applied this competency in situations successfully and with minimal guidance.

Is expected to use the skill independently with occasional help from a more senior resource.

Focus is on applying and enhancing knowledge or skill.

Skill Level: 4	*Proficient*

Able to perform the tasks associated with this skill without assistance.

Has consistently provided practical, relevant ideas and perspectives on process or practice improvements that may easily be implemented.

Is expected to coach others in the application of this competency by translating complex nuances into easy-to-understand terms.

Focus is on broad organizational or professional issues.

Skill Level: 5	*Advanced*

Able to provide guidance, troubleshoot, and answer questions related to this area of expertise and the field where the skill is used.

Has demonstrated consistent excellence in applying this competency across multiple projects and organizations and also created new applications for this competency, lead the development of reference and resource materials for this competency, or both.

Is expected to be considered the "go to" person in this area within and outside the organization.

Focus is strategic.

Competence Breakdown

The following table provides more detail about each of the nine competency areas identified as part of the talent dimension.

Skills	Description
Analysis and Design	
Architecture/Solution Design	• Master good architecture principles and be able to design applications using the 4-layer architecture • Guarantee good performance, scalability, and maintainability • Know how to avoid, detect, and correct circular references • Know how to use tools to monitor project architecture

Requirement Gathering	• Master requirements gathering techniques:
	  o Identify vision and drivers
	  o Clearly understand business context
	  o Identify personas
	  o Define user stories
	  o Develop business process models
	  o Build mockups

Web Application Architecture	

Web Application Architecture	• Understand web application architecture (client/server architecture):
	  o How and when to POST or GET
	  o Page rendering
	  o Database access
	  o Understand session behavior
	  o Understand browser caching
	  o View State
	  o AJAX requests

User Experience	

Understand and Apply Usability	• Understand information architecture:
	  o What it is
	  o Why to build it
	  o Organization
	  o Labeling
	  o Wait cost
	  o Navigation
	  o User location
	  o Auxiliary navigation
	• Understand SCRAP (symmetry, contrast, repetition, alignmnet, proximity). design principles and be able to apply them when building an application
	• Know the 11 UX Rules for IT Developers
	• Understand the importance of usability tests and be able to execute them
	• Use real sample data or generate realistic test data

outsystems

Analysis and Design

SQL Programming	Able to implement:
	  ○ Common table joins (left join, inner join)
	  ○ Exists
	  ○ Count
	  ○ Group by
	  ○ Top
	  ○ Row number
	  ○ Aggregate functions (SUM, AVG, MAX, MIN)
	Able to implement:
	  ○ In
	  ○ Between
	  ○ Outer Join
	  ○ SQL Case
	  ○ Subqueries
	Understand indexes and their impact on performance

Troubleshooting and Debug

App Troubleshooting	Able to analyze error logs Able to analyze monitoring logs Able to analyze performance logs Know how to troubleshoot, identify, and fix performance issues Understand indexes and execution plans Understand performance tuning tools and how to use them for tuning and monitoring

Development Languages

Algorithm and Pseudocode	Understand the problem or requirement Think through the solution and develop a design - be able to drill-down and explain it
C#, Java, Python, PHP	Understand language best-practices and most commonly used code libraries Able to implement a basic algorithm or functionality Experienced with MS Visual Studio, Eclipse, or other IDE
iOS & Android	Swift and Java

UI Development Languages

CSS	Know CSS syntax Able to explain the box model Know what a pseudo element is Able to troubleshoot an issue in the browser Know the order of import of CSS in OutSystems Know how to read and apply selectors Know the differences between CSS and CSS3 Know the difference between positioning values Able to understand the organization of a theme Able to understand vertical alignment Able to explain RESS (responsive web design + server side) and RWD (responsive web design) approaches

HTML	Know what elements are used and how by Know what APIs were made available in HTML5 Know how to add HTML standard elements to an HTML file Understand accessibility
JavaScript	Understand JS/jQuery Select an element by id Select an element by class Able to show and hide an element Able to import (and use) a jQuery plugin Able to troubleshoot an issue in the browser Know how to create a JS object Know how to create a jQuery plugin Know how to do a manual jQuery Ajax request Has created and shared a jQuery plugin in the community

Integrations

REST & SOAP	Able to consume a REST and SOAP API Able to expose a REST and SOAP API Know how to test REST and SOAP integrations Know how to handle null values for SOAP API
Third-party Integration	Know how to create extensions using the SAP Wizard for Integration Studio Know how to handle transactions between OutSystems and SAP

Performance

Assess Application Performance	Able to drill down into reports and identify bottlenecks . Use performance reports (understand difference between client-side vs server-side time) Use the browser performance tools
Optimize Application Code	Reduce the page size Reduce the size of page resources to load and cache them Reduce the data fetched from the database Cache data from external data sources
Optimize Application Infrastructure	Avoid database degradation Use maintenance plans Tune database file growth Improve application server and web server memory settings

Work Item Definition

Setup

Activity	Overview
Choosing and Training the Product Owner	The product owner must be able to interact with the business, translating their wants and needs and mapping them to user stories. The PO needs to be a skilled project manager, responsible for keeping the process going. Experience with low-code is a plus. **Responsible/Accountable:** Adoption Champion **Consulted:** Adoption Expert
Choosing and Training the Delivery Resource	The tech lead and (up to 3) developers should have some previous app development experience and the respective set of skills, and experience with OutSystems is a plus. The Tech Lead has other functions at the foundation level, but will have to have development training. **Responsible:** Product Owner **Accountable:** Adoption Expert
Backlog Analysis and 1st App selection	The first app selection is crucial for demonstrating the advantages of low-code and initiating platform adoption momentum by delivering business value early. This step needs to be made with care because choosing the wrong first application based on superficial reasoning, technical concepts, or a need to please can result in the effort's being abandoned because no value was perceived. **Responsible:** Product Owner **Consulted:** Business User(s) and Adoption Expert
Reference Architecture Design	Understanding integration needs with legacy systems and common nonfunctional requirements (NFRs), like authentication and authorization, audit trailing, or style guides is key to setting a reference architecture that guides future applications to be designed to last and evolve with cost savings. **Responsible:** Tech Lead Acting as Architect **Consulted:** Adoption Expert
Infrastructure Provisioning	With a cloud infrastructure, everything is set up and configured automatically for you. Cloud provisioning involves additional DevOps setup steps, yet simplifies everything. **Responsible:** Tech Lead Acting as DevOps Engineer **Consulted:** The responsible for the infrastructure at the IT org.
ALM setup	A proper low-code application lifecycle management practice is crucial to assure a project's success and to make sure you can keep up the speed throughout the application lifecycle. Different teams and roles need to be set up in the respective ALM tools, and the deployment pipeline and release management need to be defined. **Responsible:** Tech Lead Acting as DevOps Engineer **Consulted:** Adoption Expert

Adjusting for Digital Experiences

Activity	Overview
Infrastructure Design	When developing mobile applications for consumers, the infrastructure must be designed to deal with several requirements like the number of users using the app, redundancy, and load balancing. **Responsible:** Tech Lead Acting as DevOps Engineer **Consulted:** Product Owner

Demand

Activity	Overview
Application Ideation	Find innovative solutions to achieve your business goal. Analyze the potential gains and roadblocks. To successfully deliver a project, you need to identify the requirements and high-level features. **Responsible:** Product Owner **Consulted:** Business User(s)
Business Case and Metrics Definition	Based on the ideation, you can define the business case and metrics. This will include why the application makes sense business-wise, and how can you measure if you are on target. Management requires measurement for informed decision-making. **Responsible:** Product Owner **Consulted:** Business User(s)
Application Estimation	After making the business case, you determine the effort and budget required and available to create your app. Consider the requirements and high-level features involved. **Responsible:** Product Owner **Consulted:** Tech Lead and Business User(s)

Delivery

Activity	Overview
Application Initiation	When you start developing an application, the first steps set the pace of the development phase. After the initial requirements and high-level features are recorded, you're ready to define the development requirements and build a backlog. **Responsible:** Product Owner **Consulted:** Tech Lead, Business User(s)
Application Architecture design	To make sure your application is a success, it needs to have a good architecture that allows scalability, performance, and component reusability. **Responsible:** Tech Lead, acting as Architect
Iteration Planning	Every iteration starts with planning so that everything is clear for development. Planning should include recent user feedback to promote user adoption. **Responsible:** Product Owner **Consulted:** Tech Lead

Iteration Delivery	After iteration planning, development is executed according to the backlog and to agile best practices. **Responsible:** Product Owner **Consulted:** Tech Lead
Iteration Tests	The developed features for the iteration must be tested to assure the quality of the application. This phase is required before conducting the iteration demo. **Responsible:** Tech Lead Acting as Tester/Tester **Consulted:** Tech Lead
Production Rollout	The developed application needs a plan that defines the way the application is going to be deployed and used. This includes the criteria for a successful rollout and to identify when a rollback plan should be triggered. **Responsible:** Product Owner **Consulted:** Tech Lead

Adjusting for Digital Experiences

Activity	Overview
Mobile Application Initiation	After initial functional and nonfunctional requirements, high-level features, and target devices are captured, it is time to define the user stories and build a backlog. **Responsible:** Product Owner **Consulted:** Business User or Users
Mockups - Low and High Fidelity	During the MVP definition phase, ideas should be presented first with low-fidelity mockups. Some of the mockups should be taken to high-fidelity so that project stakeholders can be aligned with the vision for the application. **Responsible:** UX/UI Designer Acting as Customer Journey Designer **Consulted:** Business User or Users
Customer Journey	During the MVP definition phase, it is necessary to define the journey, the multi-touch point mapping, the experience map value, and the business value metrics. **Responsible:** UX/UI Designer Acting as Customer Journey Designer **Accountable:** Product Owner **Consulted:** Business User or Users
Implement Look and Feel	During the MVP definition phase, it is necessary to implement the look and feel based on the high-fidelity mockups. This will give a vision for how the application will look in the future and allow a faster development process. **Responsible:** Front-end Developer **Accountable:** Tech Lead **Consulted:** UX/UI Designer Acting as Customer Journey Manager

Native Plugin Development	During the project implementation it might be necessary to implement features that require the use of mobile device capabilities. This requires native development of the plugins to be added to the application. **Responsible:** Front-end Developer **Accountable:** Tech Lead
Quality Assurance	After development, it is necessary to assure the quality of the code and the architecture. **Responsible:** Tech Lead Acting as Architect **Consulted:** Developers
Functional Testing	After development, a phase of functional testing assures that the features developed are aligned with the requirements. **Responsible:** Tester **Accountable:** Product Owner **Consulted:** Tech Lead
Device Testing	After development, a phase of device testing assures that the features developed are correctly implemented on the required mobile devices. **Responsible:** Tester **Accountable:** Product Owner **Consulted:** Tech Lead
Load Testing	After development, a phase of load testing assures that the features developed are correctly implemented to support multiple user access. **Responsible:** Tester **Accountable:** Tech Lead Acting as DevOps Engineer
Field Testing	After development, a phase of field testing assures that the features developed are correctly implemented and working in the field the way the users expect. **Responsible:** Tester **Accountable:** Product Owner

Run

Activity	Overview
Production Release	The application must be deployed in the production environment. If it is not a standalone application, it should be connected to other production systems or application configurations. This process is based on the rollout document for the release, and it ensures a successful and shorter deployment process. **Responsible:** Tech Lead Acting as DevOps Engineer **Accountable:** Product Owner
App Promotion	To assure application success and user adoption, It is very important to have the application internally promoted with users and stakeholders. Create an internal marketing campaign with video, demo with the app journey, and all internal PR work needed to show that this was the right app and solution (board, IT management, etc.). **Responsible:** Adoption Champion **Accountable:** Product Owner

Application Feedback Gathering	After an application is in production, find out from users what they think so you can improve the app in future releases or fix urgent issues. The feedback you gather can be related to functionality, user experience, or app issues. **Responsible:** Product Owner **Accountable:** Tech Lead
App Troubleshooting	If issues with the app are detected, you must be able to execute symptom analysis, identify the root causes, and mitigate the problems. Otherwise, the application will not be adopted nor will it be successful. **Responsible:** Tech Lead Acting as DevOps Engineer **Accountable:** Product Owner
Evaluate Business Case Metrics	Compare the application in production with the business case and metrics defined in the demand phase to determine the success of the application. Determine adoption rate and define a plan based on the findings. **Responsible:** Product Owner **Consulted:** Adoption Expert
App Troubleshooting	In the future, there might be a need to evolve the application, issue new releases, or deliver new features. Periodic portfolio reviews can determine and prioritize the next steps. **Responsible:** Product Owner **Consulted:** Adoption Expert

Adjusting for Digital Experiences

Activity	Overview
Store Release	Where appropriate, mobile apps should be promoted in the corresponding stores. **Responsible:** Product Owner **Consulted:** Business User or Users
Performance Monitoring	You should continually monitor the application to identify possible future constraints and act on problems that may appear even before being reported **Responsible:** Tech Lead acting as DevOps Engineer **Consulted:** Product Owner
Store Review Monitoring	After an application is in the stores, you should monitor the reviews to identify how to improve the app in future releases or to fix urgent issues. **Responsible:** Product Owner **Consulted:** Business User or Users
Adoption Monitoring and Tuning	After a mobile app is live, measure user adoption and identify actions to increase it. **Responsible:** Product Owner **Consulted:** Business User or Users

outsystems

www.outsystems.com