



HOW TO DELIVER

# AMAZING CUSTOMER EXPERIENCE

A GUIDE

*tell me more!* →



THROW AWAY THE DRAWING BOARD  
THAT YOU KEEP GOING BACK TO →



# Table of Contents

---

## 03 Be the Vanguard

- + The Customer-first Era
- + It's on IT's Shoulders
- + The Challenges of Delivering CX
- + Change Your Ways to Lead the Way

## 07 Assessing Your Needs

- + Before You Begin
- + Ready to Go

## 10 Delivering Amazing CX

- + Up + Down the Customer Journey
- + The Right Platform for You
- + System Integrations
- + Requirements and Feedback
- + Cloud
- + Mobile (and Desktop and Web...)

## 17 Where It All Comes Together

---

SECTION I

**BE THE  
VANGUARD**

100%



Tell me  
more ↗

# The Customer-first Era

“The customer is always right” is an adage that’s maybe just a few seconds younger than the age-old concept of business itself. Like many mottos, no one should take this literally. No one is right all the time. The purpose is to ensure that you keep the customer at the heart of what you do as a business. Always. We are living in a time where keeping your customers at the front and center of your business strategy has never been more essential.

So how is your company preparing for the customer-first era?

You’ll most likely hear of organizations heavily investing in customer experience (CX). CX and its benefits—retention, engagement, brand prominence, competition one-upmanship—are undoubtedly the current driving force in organizations. This reality has only been solidified by our pandemic-altered world. Good, engaging, easy customer experience is no longer a nice-to-have. **It’s a necessity.**

← Agreed!

For organizations around the globe, the CX trend also presents an opportunity. It’s a chance for your business to improve and innovate the way customers interact with your business, perhaps even creating new channels of revenue or ways of reducing costs.

To put a finer point on it: This is IT’s opportunity to spearhead the organization’s journey to customer-centricity.

## It’s on IT’s Shoulders

Your entire organization needs to think and act customer-first, but ultimately it comes down to technology and the people who are implementing it, developing it, and capitalizing on it.

Which makes you, your developers, your architects, your QA team, those folks over in DevOps and everyone else in your department crucial to a business’s customer-centricity.

Think about the way modern customers expect to interact with businesses today. To meet them where they’re at, your business needs:

### 1. The ability to digitalize everything – really.

Automate the manual, digitize the analog, eliminate the in-person. Using tech to make processes more streamlined.

### 2. To be mobile-first

Nearly 50% of organizations expect to be improving their mobile experiences over the next five years.<sup>i</sup>

### 3. Customer-expected digital touchpoints + modalities

52% of companies are spending only 30% of their budget on innovation.<sup>ii</sup>

yikes →

### **That’s all on you.**

You’re in charge of how systems integrate with each other. You and your department own how software is developed: what languages are used, what features are prioritized, deployment options. You’re in charge of which mobile framework to build with, which architecture to build on. You decide what goes into your tech stack and which resources to allocate to build experiences for new modalities.

# The Challenges of Delivering CX

The challenge that comes with being customer-first is that customers are humans, and humans change. What worked for them once may change down the line. What works for someone in finance may not work for someone in legal. What works in the office changes in the field. A modality that works for someone in their 20s might not be right for someone in their 60s. Developing a multi-modal experience that represents your organization's business for an ever-changing audience is much like trying to build a castle on quicksand.

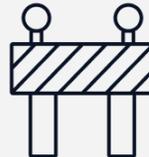
Giving customers what they want is tough in the customer-first era if you're developing bespoke solutions using traditional development methods. It's even tougher if you're outsourcing that development. Such development methods are a recipe for:



**SLOWER TIMES TO MARKET**



**COST INCREASES**



**LIMITED FEATURES + UIs**



**LACK OF INNOVATION**

## Risk and Resources

Adding more digital touchpoints means more maintenance. Of those IT professionals, only 14% spend 70% or more of their budget on innovative solutions. The rest are focused on maintenance.<sup>ii</sup>

Not only does innovative CX require more resources that you don't have, but it also comes with risk. From that same survey, 64% of respondents cite that new programming languages and frameworks required to create or improve these digital touchpoints result in higher security risk.

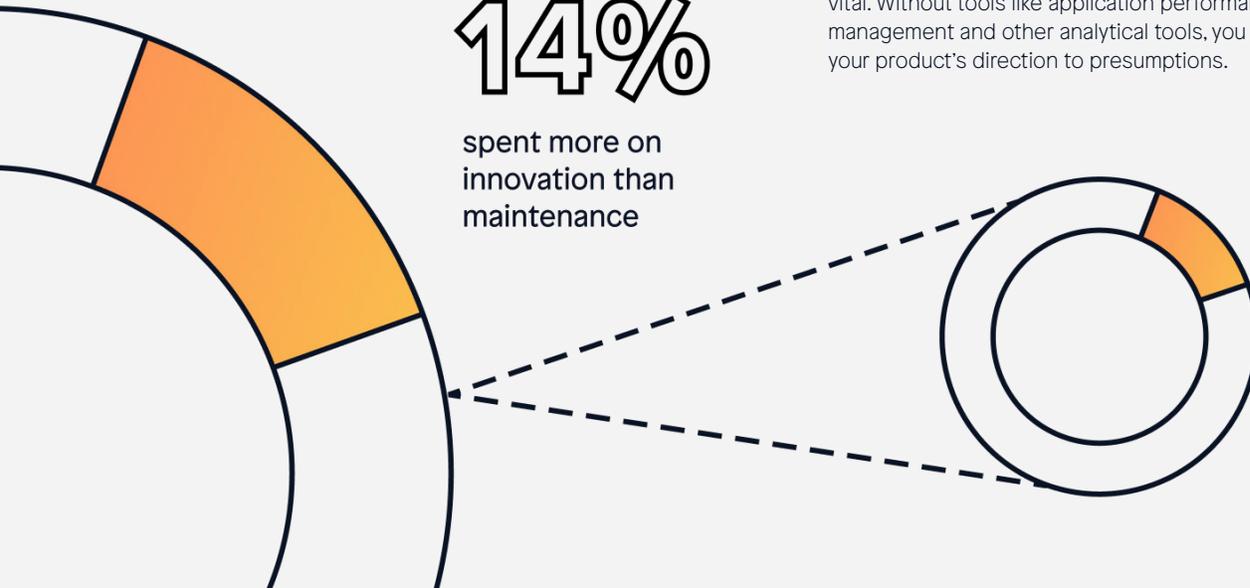
## Data and Analysis

Beyond risk and resources, developing for today's customers is challenging if you don't have the methods in place to gain an understanding of them. Part and parcel of a successful customer-focused solution are your feedback-gathering methods. Customer experience is all about constantly improving and continuously iterating based on customers' needs. If your development and deployment methods don't match that, then you aren't delivering CX that your audience (rightfully) demands. Slower feedback means longer release cycles driven by guesswork, because you're not exactly sure what the users wanted in the first place.

The ability to see how your applications perform and how your customers put them to use is also vital. Without tools like application performance management and other analytical tools, you leave your product's direction to presumptions.

**14%**

spent more on innovation than maintenance



# Change Your Ways to Lead the Way

*Game Changer*

To become the vanguard of your business and lead the charge toward a customer-first mindset, you need to think of your software development methods as a work in progress. **Many organizations are turning to low-code to change the way they develop.** 85% of respondents from the “Win with Customer Experiences” survey are looking to low-code development to accelerate and improve app design. 78% are using it to better understand the customer and implement faster feedback loops.<sup>iv</sup>

## The 4 Ps

No singular tool will change the way you build and deliver customer experiences. In addition to finding the right development platform, you also need to change your processes, activate the right people, and take the time and care to prioritize your application portfolio.



### PEOPLE

Not only do you need to better arm the people building the software with the right tools, but they need the right information. That comes from the folks making the software requests, the people setting the software requirements, and the customers and employees who are using the software.



### PROCESS

Yes, the right processes around building software need to be put in place, but this goes beyond being Agile or Lean. CX is never done. You need to establish feedback-gathering processes to better gather qualitative and quantitative data. You need iterative processes to ensure that you're always improving to meet customer expectations.



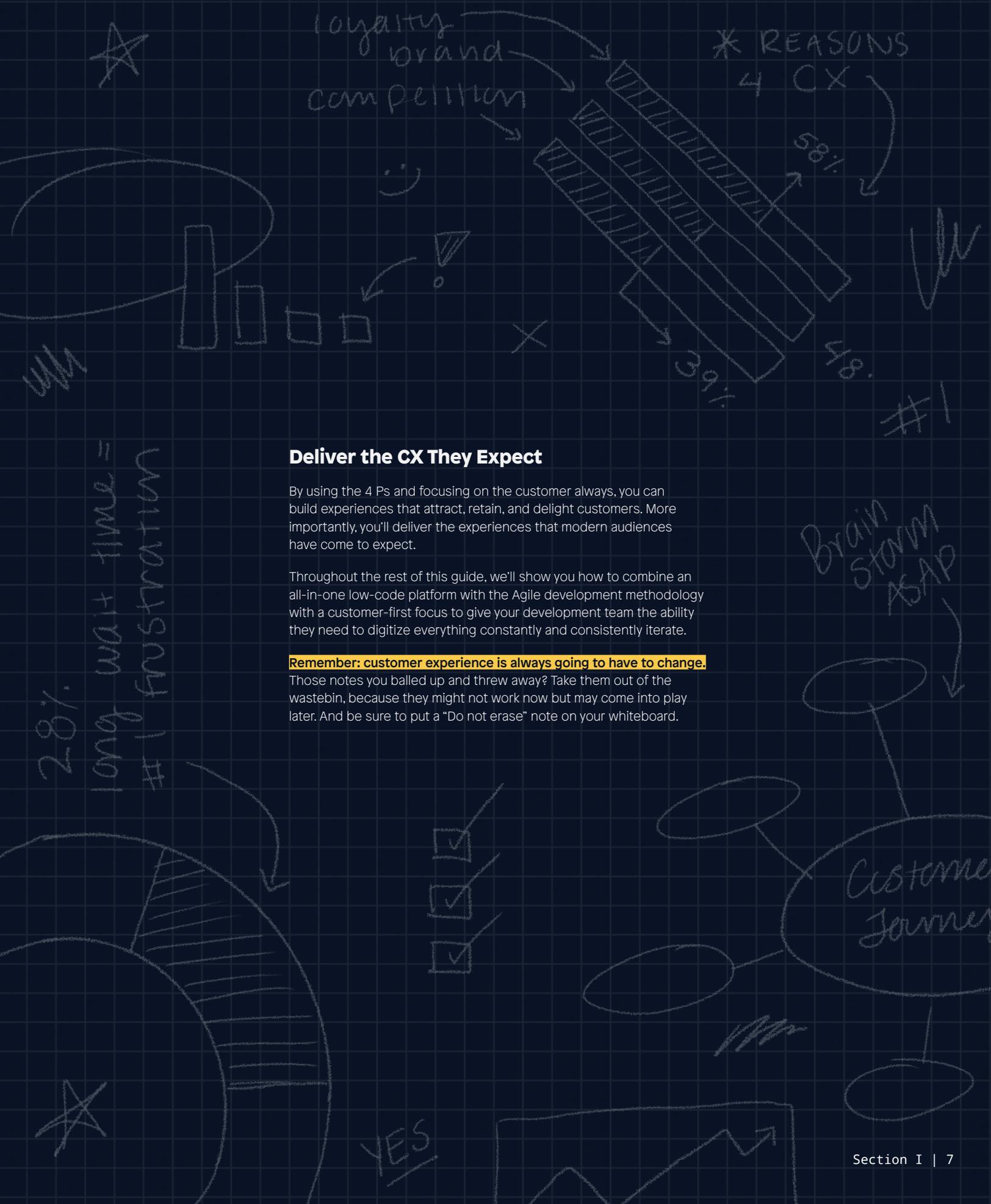
### PORTFOLIO

Using that data you gather helps inform you of what to prioritize – which digital touchpoints to build in the customer journey and which modalities to invest in. You also need to consider how your systems interconnect to truly deliver a seamless experience across your CX.



### PLATFORM

The right development platform is key to creating a customer-first IT strategy. You want a platform that helps you accelerate application development, easily integrates with third-party tools, connects with the rest of your tech stack, and also provides robust deployment options.



## Deliver the CX They Expect

By using the 4 Ps and focusing on the customer always, you can build experiences that attract, retain, and delight customers. More importantly, you'll deliver the experiences that modern audiences have come to expect.

Throughout the rest of this guide, we'll show you how to combine an all-in-one low-code platform with the Agile development methodology with a customer-first focus to give your development team the ability they need to digitize everything constantly and consistently iterate.

**Remember: customer experience is always going to have to change.**

Those notes you balled up and threw away? Take them out of the wastebin, because they might not work now but may come into play later. And be sure to put a "Do not erase" note on your whiteboard.

*This* →

SECTION II

# ASSESSING YOUR NEEDS

*~~~~~*

*great* ↗

# Before You Begin

Where do you start on the journey toward customer-centricity? You need to look inward first. Before putting in the work to change how your customers interact with your business, you need to assess how they're currently doing that.

Get a paper and pen or go to your favorite whiteboard and start answering these questions:

## What are the processes your employees use?

Think about customer onboarding processes, payment transactions and processing, customer service routing, approval processes, incident tickets, requests for service, and scheduling.

## How do you feel about those processes?

Are they complicated? Analog? Paper-driven? Have you received anecdotal feedback that they're inefficient, tedious, and potentially decreasing customer loyalty? Or maybe they just need a tweak or an update to meet accessibility requirements.

## What sort of touchpoints are along the customer journey?

Does your CX cover all the scenarios in which your users want to interact? Do you offer phone, web, mobile, chat and/or voice? What about something more immersive like augmented reality? Are there areas where your customers should be able to self-serve? How do those touchpoints interact with each other?

## What processes do you use?

How do you take in requests, gain feedback, and ensure requirements are being met? Is there room for improvement there?

## How do you develop for each touchpoint?

Take a look at your software development kits. What are you using? Are you developing at the speed you and your customers need to develop with? How do your developers handle integrations and APIs? Do you employ DevOps for source control, testing, deployment, etc.? Do you need more skills around niche technologies like AR, mobile, or AI?

# Ready to Go

This self-assessment gives you a full picture of how your customers are and are not interacting with your business, and how your software development methods coincide with that.

With today's customer experience, it may feel like you have to develop for everyone ... and everyone wants everything. But, prioritization is key to success here. It's important to do this upfront work because getting a lot of these answers is going to help you figure out where you need to begin and what you should focus on versus ignore.



*yes*

SECTION III

**DELIVERING  
AMAZING  
CX**

*This hits  
home*

# Up and Down the Customer Journey

You've assessed your business and its customer experiences. You've prioritized what needs to happen. Now you just need the "How" of it all. Let's dive into exactly how you're going to build out different modalities across all your touchpoints along the customer journey.

Keep in mind the 4 Ps.



## The Right Platform for You

*But how???*

The key to developing different modalities without burying your entire department in technical debt is to find a development platform that lets you **build what your customers need while at the same time respecting the tech stack you're currently invested in.**

Many organizations are turning to low-code platforms: Gartner predicts that by 2025, 70% of new applications developed by enterprises will use low-code tech.<sup>1</sup> This is because many—one-third, according to the Win with Customer Experiences survey—are struggling to find and retain developers and designers. It's tough to train and retrain developers on highly esoteric SDKs and a heavy catalog of programming languages, which means you either focus on cultivating (and retaining) a highly diverse team of incredibly talented developers or you find a low-code development platform that allows your developers to build and integrate with a number of technologies.

While 70% of applications will be developed with low-code, 33% of organizations, per the survey, intend to adopt low-code specifically to build digital customer experiences.

### Why Low-code?

Low-code abstracts away a lot of the hard coding that can be time-consuming and often redundant. The right low-code development platform can also take away the pain of integrating with other systems and third-party software, which is crucial when you're building complex multi-modality, multi-touchpoint digital customer experiences. The right low-code platform also allows you to extend applications with custom code too, so you're able to build out robust, feature-rich applications more quickly.

The right low-code platform goes beyond technology. It augments your ability to change how your business thinks about projects and development pace. Built-in Agile tools like sprint and stories allow you to **collaborate more smoothly, and share requirements and changes more easily.** Features like these allow for those in the business to better communicate needs and requirements to those in IT.

*Collaborate w/ Biz*

The right low-code platform also allows you to build both cloud-native and native mobile, so that deployment is streamlined and creating mobile experiences across devices isn't an Android/iOS nightmare.



# Systems Integrations

To serve today's customers well, you need to make sure your integration strategy aligns with your customer-first strategy.

Your customers expect that no matter the device or modality or where they are in the journey of interacting with your business, you give them what they need to accomplish their goals. Because every user has different preferences, that means creating many modalities for the same touchpoint. Sometimes the modalities have to be able to transfer from one to the other (e.g., desktop to mobile, or chat to phone).

All of these modalities require different user experiences and interfaces. More of those means more technologies. More technologies mean more tenuously tied-together systems. These bespoke

experiences can't be delivered at scale if you're developing them in a bespoke manner.

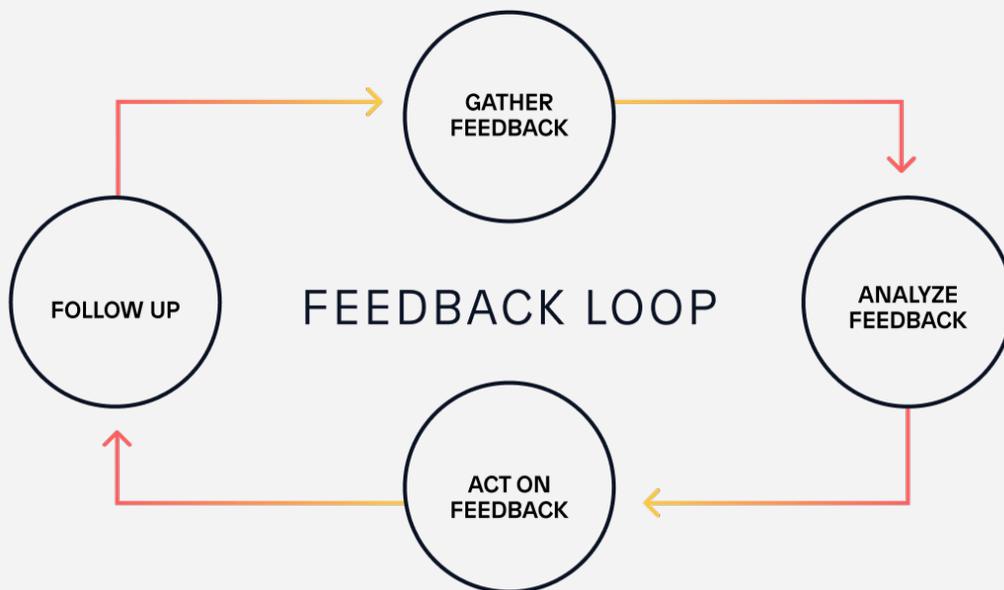
Using APIs and microservices is a way to create reusable building blocks to be used again and again across different solutions. But creating those integrations to many different systems in and outside of your company can be difficult, which is why using a low-code platform with an entire marketplace of prebuilt integrations is extremely helpful.

With low-code, you can more easily connect together the systems you'd need to create digitized, more efficient workflows, or better connect a multi-modality customer service system.

# Requirements and Feedback

The key to creating a great customer experience is knowing that it's never done. **You're no longer building a solution. You're managing the customer experience.** There's always a new device or a new preference that people desire when interacting with businesses. To truly understand your customers' wants and needs requires getting feedback in a consistent manner so you can deliver new features in an iterative manner.

*CX = Iteration*





## Feedback Gathering

An alarming pitfall that 47% of organizations fall into when trying to deliver new CX is that they ignore or are unable to track customers' needs and feedback.

The connection feedback has to delivering outstanding CX is obvious. Whether CX is good or not is based all on your users' opinions, not yours. Willfully ignoring feedback or not having anything in place to gather will never get you to the point where you're meeting customers' needs.

Feedback can be both qualitative and quantitative.

Sources of qualitative feedback can be win/loss interviews, conducted by your marketing or product teams. A win/loss program allows you to speak with customers that both decided to invest in your product and those who went in another direction. These interviews can prove invaluable. You're able to collect anecdotal evidence on what's working with people who like your product and the way they interacted with your business. You're also able to get insight as to why your product may not work for some.

You can learn a lot from win/loss interviews. Perhaps there's a missing feature. Maybe there's a geography where a feature you think is a great offer isn't really necessary or feasible. Responses to win/loss discussions start out as anecdotal – but when you make this feedback process programmatic **you'll eventually start to accrue significant enough data to view through a quantitative lens.**

more  
more!  
more!

Customer surveys and customer advisory boards are also good sources of anecdotal evidence for what your customers desire.

In aggregate, the results of these interviews and programs can reveal the potential for new products, new channels, or even new lines of business.

There are also more technical ways you can elicit feedback. Building a chat function within your product or your customer portal can prompt users to give you their thoughts. Also, looking into usage metrics can tell you where customers drop off in the customer journey. Putting in performance measuring tools can really give you insight into how your customers are interacting with your business.

## Agile



The Agile development methodology should be intrinsic to anyone's CX delivery plans. The software development methodology is an iterative and responsive approach to software development. There are two qualities of Agile that play well into managing the customer experience. Agile fosters high levels of communication and collaboration and it provides a setting to effectively adapt software planning and delivery based on needs.

Using the Agile methodology lets teams put ceremonies in place to take in employee and customer feedback and plan the next round of features based on that feedback, which is turned into requirements, which are then met and deployed (or redeployed). After that, the feedback cycle starts again.

Agile is particularly important because requirements miscommunication and alterations are a big factor—37% and 36%, respectively—in impeding delivery times.

## Design Thinking



There are different ways to build out requirements for a solution, but one that is very customer-focused is the design thinking process. Design thinking relies on observing how your customers connect with your product. It takes into account their feelings, their surroundings, and their goals to help understand the mindsets of your customers.

There are seven activities in the design thinking process: immersion, sense making, alignment, emergence, articulation, learning in action, and pre-experience. According to Jeanne Liedtka of the Harvard Business Review, "Though ostensibly geared to understanding and molding the experiences of customers, each design-thinking activity also reshapes the experiences of the innovators themselves in profound ways."<sup>vi</sup>

Design thinking urges you to reframe the questions you're asking and why you're asking them. By reframing your questions, you're often able to think differently about the problem you're developing a solution for.

# Cloud

Your experiences, the touchpoints along your customer journey, need to be future-proofed and resilient. This means you need to know your deployment methods so you can offer applications that are available and easily updated.

You need to know your cloud strategy. Are you planning to run on the cloud? If so, does your business or industry require to use a certain type of cloud? Private? Public? Hybrid?

Depending on your choice, deploying to the cloud has a lot of benefits to your operations, which in turn have a lot of benefits to your customers.

Going cloud native can also be a boon to your CX. Cloud native applications use a microservice architecture that allocates resources to each service the app uses. Cloud native applications are also able to better scale due to their architecture and are cost effective and easy to manage. You can make high-impact changes more safely and simply with a managed microservices app. Your application is much more resilient and fewer resources are spent managing infrastructure. This means more resources can be focused on building the solution and adding business value to it.

## Resilient and Secure

Mendix applications are cloud native by default, meaning they're resilient right out of the box. You can deploy in any environment and scale without re-architecting. You can also deploy your apps with just one click, meaning that the iterative process you establish is completely supported even when it's ready to deploy.

While it's not an engaging UI or a cutting-edge experience, **an important part of CX that cloud offers is the sense of security and safety it can provide you and your users.** Public clouds, for instance, offer high uptime guarantees, meaning minimal disruptions to services as well as disaster recovery.

Mendix is recognized by many entities as being compliant when it comes to cloud security. It holds certifications for ISO/IEC 27017 and 27018, key international standards for information security controls for cloud services and protection of personally identifiable information in public clouds, respectively. Mendix also holds a CSA STAR certification, having completed the CSA STAR level one self assessment, which is based upon a comprehensive list of cloud control objectives.



## Mobile (And Desktop and Web...)

Good customer experience is never a question of mobile or desktop or web. You need to offer all options, and those options need to speak to each other and integrate without friction. You'll find a person interacting with your business first on their phone through a mobile website. Then later down the journey, perhaps they finish a transaction on their desktop. Maybe their work brings them to an area without a signal and they have to work offline on their mobile device.

People and the context in which they interact with your business are always switching and you have to be sure to develop for every scenario to make sure you're meeting expectations.

## Mobile Development and Mobile Architecture

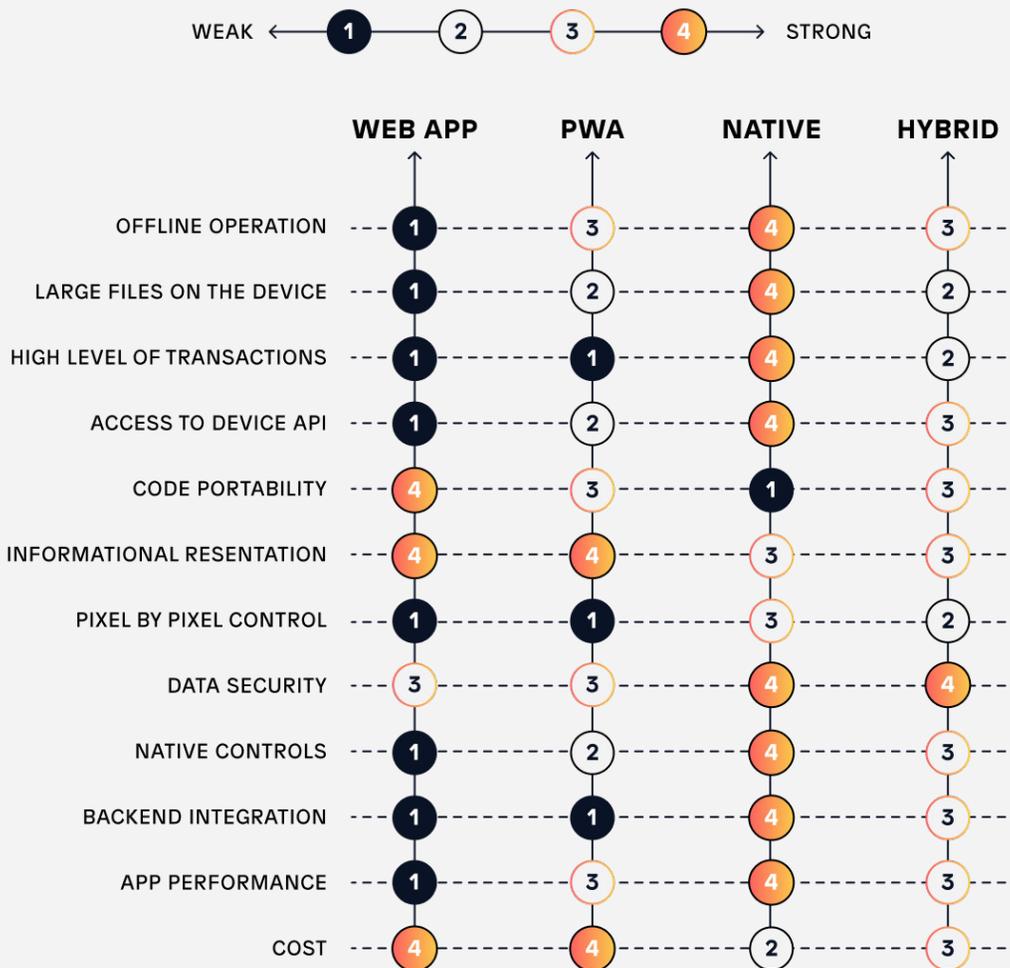
A customer-first strategy, in most cases, should go with a mobile-first mindset. That's not to discount desktop or web (those are still very needed). But mobile is vital. 68% of all website visits in 2019-2020 stemmed from mobile devices, according to a Google Analytics Benchmark report. An IDC survey suggests that, in 2020, organizations that invested in digital transformation technologies like mobile applications have seen a 20% increase in employee productivity.<sup>vii</sup> While you should treat your employees as if they're customers, too, this even

has an effect on your external constituents. 85% agree that enhancing the employee experience leads to a better customer experience.

If you're worried about which architecture best fits your customer base, here's something you need to hear right now: **Plan to use all of them.** In the report "Key Considerations When Building Web, Native or Hybrid Mobile Apps," Gartner suggests that rather than using a one-size-fits-all approach with mobile architectures, 90% of enterprises will use a combination of the three.<sup>viii</sup> So now mobile development becomes a matter of when to use the right architecture per solution, not overall.

*Could that work?*

## MOBILE ARCHITECTURAL TRADE-OFFS



Different use cases call for different architecture types. Going with a native architecture allows you to create Android or iOS-specific applications. With that, you're often developing twice over. That can be time- and cost-intensive, but the usability and security features may outweigh that.

Developing a web app gives you lots of code portability and is generally more cost-effective, but it's tough to integrate with backend systems and doesn't perform nearly as well as a native app.

A nice middle-ground is progressive web applications (PWAs). PWAs have limited device capabilities but are generally better experiences than a responsive website. With the rising popularity of PWA and cross-platform mobile frameworks like React Native and Flutter, we're seeing a decline in developers using hybrid containers—so much so that the most popular hybrid container, PhoneGap, was deprecated by Adobe in 2020.

The Mendix low-code platform gives you the ability to build best-of-breed app-store-based and browser-based experiences from one platform. Using low-code to build native apps means you don't have to use OS-specific programming languages, opening up your resource pool to create these richer applications. You can develop once and deploy anywhere. Mendix's native apps leverage React Native to enable the same code base to run on both iOS and Android, and they look the part, too. They blend into their respective ecosystems.

You need to make sure that you're developing with mobile in mind because once you have your mobile architecture set, nearly all other modalities falls into place.

### ...And Everything Else

**The best customer experiences are the ones that are so seamless, so effortless, they become a part of everyday life.** Mark Weiser, widely regarded as the father of ubiquitous computing, once said "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." While mobile is that technology now, AR/VR, IoT, and other kinds of modalities are not far away.

Beyond mobile and the different variations, Mendix is a low-code development platform that also helps you deliver a range of experiences like conversational UIs, augmented reality and mixed reality. The same philosophy used in mobile applies to the other range of experiences that customers may expect.

With a low-code solution like Mendix, you're able to build enterprise-grade experiences without the specialized developer skillsets needed to build them. Alternatively, developers can extend Mendix applications with custom coding to open up to or extend to technologies like IoT and conversational UI.

*Future State*



SECTION IV

**WHERE IT  
ALL COMES  
TOGETHER**

*This can be done J*

Delivering customer experience is not easy.

It requires change at a level that may seem unfathomable. How does an IT department lead a transformation that affects the people, processes, technology, and the solutions you build for an entire enterprise?

That's a huge ask and, again, it's not easy. But it's not impossible either. Once you're able to put the contents of this book into action, you'll see opportunities of all kinds begin to open for your business. That's what the ability to digitize everything and become mobile-minded does.

Incorporate feedback methods, and you begin to get the information and data you need to help inform how you prioritize what to build and for whom. Institute an agile way of working and you get to put that feedback to action and push data-informed updates out faster. Abstract away the code and simplify your systems and you speed up time-to-market and concentrate on

creating business value. Expand your device and deployment options and you give users the control they want with the security that puts them at ease.

Or, in other words: Inject a customer-first mindset into your development methods, and you'll find yourself understanding your audiences better. Understanding leads to delivering experiences that consistently meet their needs or exceed their expectations.

Meeting those changeable needs leads to more loyalty and retention. Understanding also leads to finding out you need to deliver new ways of engagement, which could open a path for your business to break into entirely new, previously unexplored markets.

Venturing into the unknown of the customer-first era isn't easy, and there's no one way to travel.

**But the risk and your work are worth the reward.**

Let's go!

---

## Endnotes

- i "Win with Customer Experiences-A Survey," Mendix, Mendix, Inc., 19 Jan. 2022, <https://www.mendix.com/resources/win-with-customer-experiences-a-survey/>.
- ii Ibid
- iii Ibid
- iv Ibid
- v Wong, Jason, Kikihimo Iijima, Adrian Leow, Akash Jain, and Paul Vincent. 2021. "Gartner Magic Quadrant for Enterprise Low-Code Application Platforms." Gartner, Gartner, Inc. September 20, 2021. <https://www.gartner.com/en/documents/4005939/magic-quadrant-for-enterprise-low-code-application-platforms>.
- vi Liedtka, Jeanne. 2018. "Why Design Thinking Works." Harvard Business Review. <https://hbr.org/2018/09/why-design-thinking-works>. September 1, 2018. <https://hbr.org/2018/09/why-design-thinking-works>.
- vii Muscolino, Holly. 2021. "The Employee Experience & Customer Experience Connection." IDC Blog | Global Market Intelligence, IDC. September 17, 2021. <https://blogs.idc.com/2021/09/17/employee-experience-and-customer-experience-what-is-the-connection/>.
- viii Leow, Adrian, and Jason Wong. 2020. "Key Considerations When Building Web, Native or Hybrid Mobile Apps." Gartner, Gartner. February 14, 2020. <https://www.gartner.com/en/documents/3980961/key-considerations-when-building-web-native-or-hybrid-mo.MobileApps>; Gartner, Gartner, 14 Feb. 2020. <https://www.gartner.com/en/documents/3980961/key-considerations-when-building-web-native-or-hybrid-mo>.

