



7 Best Practices For Secure Application Development

INDEX

Modern application development risks.....	3
App development without a plan.....	4
Scope creep and lack of focus	5
Unrealistic app development goals	7
Lack of consistent, clear communication	8
Arbitrary platform selection.....	10
Lack of post-launch planning for ROI	12
Inadequate resources.....	13
Conclusion	15



MODERN APPLICATION DEVELOPMENT RISKS

Businesses rely on custom applications to build internal tools, web platforms, and mobile apps. However, many developers increase business and security risks due to inefficient processes. Code security is a particular risk.

There are many risks to be found in modern application development. Still, development risks can be reduced, if not eliminated, by following DevSecOps practices designed to identify those risks and resolve them before they create problems.

Let's look at some of the most prominent risks and the best practices to follow to mitigate them.

RISK: **APP DEVELOPMENT WITHOUT A PLAN**

A great idea isn't enough to develop a great application. You won't get anywhere without a product strategy regardless of defined specifications.

Without planning, you may discover too late that your application isn't well suited to current market conditions. Maybe it has competitive deficiencies or isn't marketed to the right users.

BEST PRACTICE: **ALIGN APPLICATION DEVELOPMENT WITH A PRODUCT ROAD MAP**

A [product road map](#) is a high-level summary that visualizes product direction over time. Development is never one and done. You need to set and understand multiple factors for deployment and follow-up workflows to ensure long-term success, including:

- Advocacy
- Constraints
- Dimension
- Estimation
- Ideation
- Priorities
- Publication
- Sequencing
- Synthesis
- Validation

Collaborate with your development team to create a strategy that encompasses everything your team needs to build and launch a successful application. Your plan should include future updates and maintenance, as well. Benefits of aligning your application development with a product road map include cost savings, informed and aligned cross-functional teams, and a quick response to your application.

The road map communicates your capabilities' classification, sequence, and alignment while letting your users gain insight into what you are building and why.

RISK: **SCOPE CREEP AND LACK OF FOCUS**

You have a brilliant idea for the perfect app. It sounds great, so you get your team going on it full steam ahead. But you might run into a problem or two if you don't know what you don't know. Here is what you should be able to articulate before beginning:

- What users want
- Your organization's business goals
- The problem(s) to be solved by this application
- The final results you want to see
- The benefits created for your organization

Building unneeded and extraneous features increases development time and cost. If you want to increase the probability of a tangible and quick return on investment (ROI), you need to devote your efforts to a product that will achieve that aim.

BEST PRACTICE: **FOCUS ON A MINIMUM VIABLE PRODUCT**

Before you ever begin software development, prioritize the most critical functionality that generates the quickest ROI. Don't add anything else. Instead, work on a minimum viable product (MVP) to showcase and test out your idea. You want a realistic first version of your app that solves your business problem. Do the following:

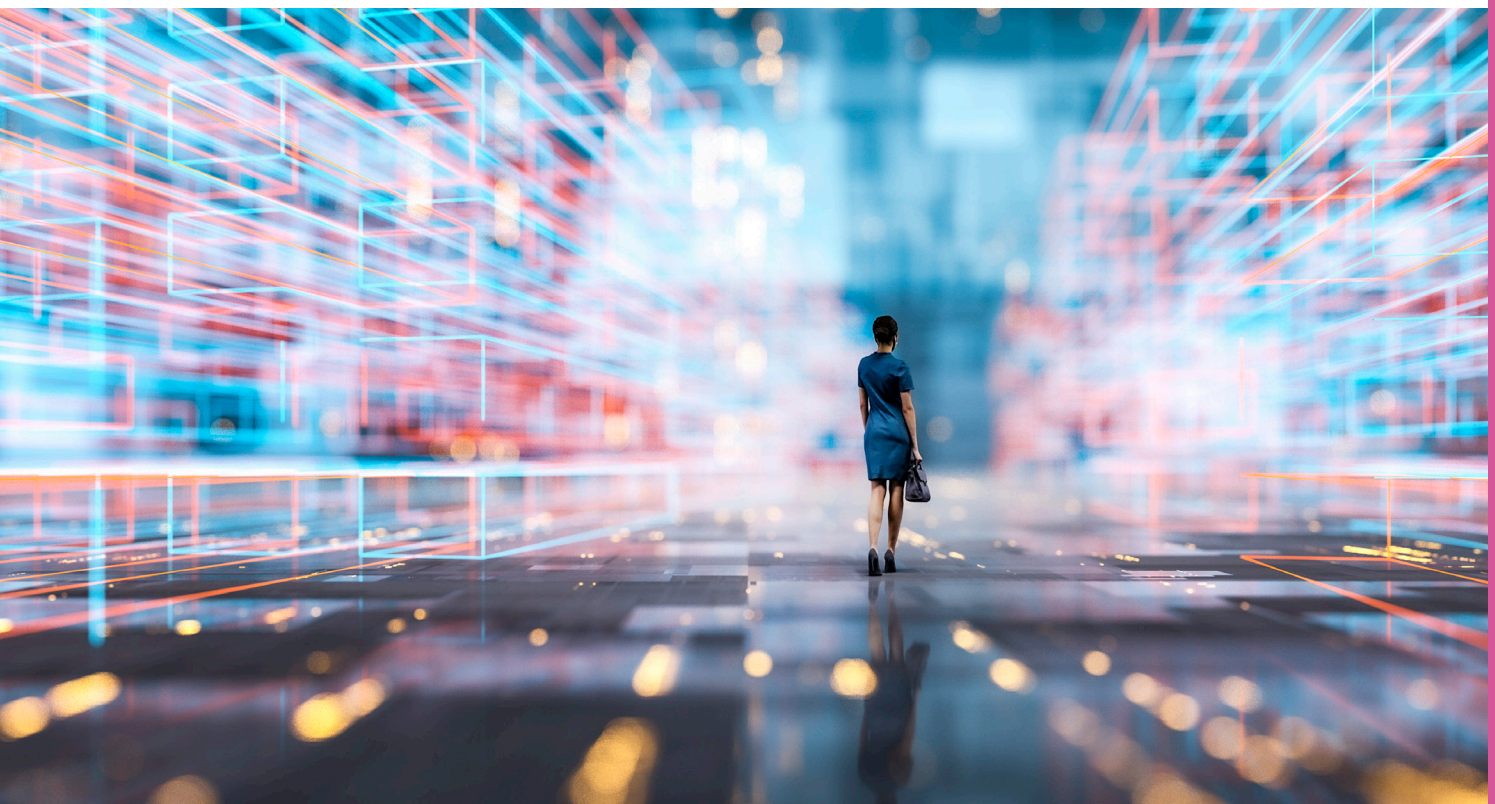
- Define your proposed app's primary purpose
- Define three to five other business issues it should solve
- Determine the target audience for your app, including their demographics and preferred operating system or platform
- Look for similar apps on the market and determine how closely they align with your idea
- Read reviews of potential competitors to learn what users do and do not like

Find the number of downloads of your competing app to determine popularity

Once you know precisely what you want your application to do and have discovered competing solutions, you can decide whether your idea merits further consideration. If you move forward with development, begin by writing down the features and — if applicable — how you will monetize your app.

After developing your MVP, hand it over to users to gain feedback on the included features. Begin design iteration and adjust feature work dynamically based on the feedback received. Now that you have a better idea of the features that deliver the best ROI, you can put more development sources on those features.

Using an MVP to gain end-user feedback reduces the risk of launching an application with unwanted features, saving resources and development costs. The MVP also reduces broader operational risks for business-critical applications by identifying and eliminating risks early in the development process and placing a workable version in the field. At the same time, you work on the full product.



RISK: **UNREALISTIC APP DEVELOPMENT GOALS**

When developing custom applications, you won't recognize success if you don't have realistic, measurable goals. There is no such thing as the perfect app that generates tons of revenue and provides optimum value right out of the gate. Unrealistic expectations simply set you and your team up for failure.

BEST PRACTICE: **SET ATTAINABLE APPLICATION DEVELOPMENT GOALS**

What do your users want and need from the first version of your MVP? Once you know the answer to that question, you can set reasonable development operations goals. You have a better idea of the goals, resources, and time required to develop the next version of your MVP based on user feedback.

Write the technical specifications clearly, so each developer understands the requirements. Specifications provide you with guidance on the scope of your development process and help you determine the right architecture for your solution.

Employ SMART criteria: specific, measurable, achievable, relevant, and time-based objectives to keep your goals realistic and attainable. Another approach is ACES – achieve what you want to do, conserve what you have to keep doing, eliminate everything that blocks achieving the goal, and steer clear of what's unneeded.

You can also resolve just one of multiple problems. For example:

- Achieving launch day monetization potential immediately
- Acquiring a considerable user base quickly
- Streamlining your internal process by improving operational efficiencies

Each of these goals is fully attainable on its own. Attempting to meet all three is unachievable and unrealistic.

Once you set realistic and achievable goals, you have a focused definition in place for success. Be practical, have clear objectives, and follow your schedule. You have laid your MVP development foundation.

RISK: **LACK OF CONSISTENT, CLEAR COMMUNICATION**

Once the specifications are delivered to the developers, it's easy to assume that what you agreed to is going to be built the way you want it.

Unfortunately, the development team's assumptions may not match yours without consistent, ongoing communication. Differences in interpretation lead to unexpected changes in product performance. Then, your team can waste resources moving in the wrong direction and going back to correct the problem.

Sometimes, stakeholders fall out of the communication loop. They are inadvertently excluded from communications and meetings. Or, they signal a willingness or desire to remain on the sidelines and expect the development team to work without input.

BEST PRACTICE: **HOLD FREQUENT, FORMAL CHECK-INS**

The solution to miscommunication is to create clear channels and habits of communicating progress, problems, and issue resolution. Develop a schedule of regular meetings and attend the daily stand-up, if possible. The minimum time you should allow to pass without at least one touchpoint is two weeks.

For example, if your team uses Agile and Scrum, participate in the sprint reviews that typically occur every two weeks. The goal is regular project monitoring to eliminate surprises for stakeholders outside the development team. Include key contributors from across the organization. Marketers need to keep up with development to ensure a smooth roll-out. Designers need to ensure ongoing development aligns with design considerations. Bring in a representative from other areas that could add or receive value from regular progress reports and interaction with the development team.

Establish a technical liaison to keep communication flowing if you are working with an external development team. Having regular communication can help resolve issues faster. Dedicate the time for managing an outsourced project. Don't assume the external vendor is moving in the right direction.

If you have a code repository that would help the external developer, your liaison can point the way so the project isn't slowed by lack of access. As you build your liaison role, keep in mind how your organization communicates with a vendor. These are excellent tips for any manager and group:

- **Don't micromanage.** Overly prescriptive instructions can reduce productivity, efficiency, and quality. Too much time is spent on compliance.
- **Take care to nurture relationships.** Workers who feel distrusted, undervalued, or disrespected have no incentive for optimum performance or communicate valuable insights.
- **Clearly define the liaison's role and responsibilities.** You can also establish milestones and metrics to measure success.
- **Emphasize transparency.** Hiding problems from outsourced partners because you believe them to be irrelevant can create significant problems down the road.
- **Don't artificially limit the relationship.** If you only share what you think the outsourced team needs to know, vision becomes misaligned, missing information harms performance, and you miss innovation and relationship-building opportunities.
- **Don't make assumptions.** Eliminate shorthand in your communications because you think the recipient knows what you mean. Making people guess what you want leads to mistakes and inefficiencies as well as the potential for improvement.

Give as much attention to communication with external vendors as you do for your internal teams.



RISK: **ARBITRARY PLATFORM SELECTION**

Some organizations pick the platform they have always used. Others go with the one the developer is familiar with. There is no one-size-fits-all selection. Instead, you need to choose whether you need a native app, a cross-platform app, or a responsive app. Every platform has advantages and disadvantages. Decide which is best for you by considering user group management needs, integration requirements, and security, as well as your choice between native or cross-platform.

BEST PRACTICE: **INCLUDE PLATFORM CHOICE DURING PLANNING**

Ask your development team for input on the best platform for your project. Find one that has the technical advantages you need plus acceptable limitations. Check user reviews and research when researching your platform choices. Other considerations include:

- Ability to run application security testing
- After-launch updates
- Development cost
- Features required for mobile app and user experience
- Platform feature support
- Security risks
- User expectations such as speed

You have one choice to make that may supersede the rest. Are you going with single platform development or multi-platform development? Each has pros and cons.

Single Platform Development

The native iOS platform is closed – Apple controls everything. The vendor has strict guidelines but you need to prepare for changes. You should be ready to update based on the latest iOS regularly.

It's easy to see that the native Android platform is the top mobile development platform in the world. A large pool of users is available, and the platform is more flexible than iOS. Java is the language of choice and it's easy to find a skilled developer.

You should consider developing for a single native platform if:

- The project is particularly difficult
- You desire to be “stylish”
- You need to develop a custom solution
- You want the app to have a native look and feel
- You want to deliver the best user experience
- You want to spend less on testing through devices

Cross-Platform Development

Cross-platform applications have the advantage of running everywhere, so you only need to write it once. However, there are drawbacks. Compatibility is not equal on all platforms. Native apps run better than cross-platform apps, which are not optimized for any single platform. The app may not look as aesthetically pleasing on any platform.

Cross-platform performance is highly dependent on the framework as every platform has its limitations. Again, there is no perfect one-size-fits-all solution.

You should consider developing a cross-platform application if:

- Cost is an important consideration
- The project is simple and straightforward
- You are developing a promotional project
- You don't need a “stylish” user interface (UI)
- You require multi-platform development

RISK: **LACK OF POST-LAUNCH PLANNING FOR ROI**

Your responsibility for the app doesn't end with its development. Early planning should determine how you achieve rapid, strong ROI with the app. Just because you think the application is brilliant doesn't mean users will find the app or use it optimally.

Product launch is not an ending but a beginning. You need to look at the application as a holistic experience. Don't assume that the future brings only a few bug fixes or new features.

BEST PRACTICE: **A ROBUST POST-LAUNCH PRODUCT PLAN**

During early product planning, consider post-launch sustainability and ROI. The post-launch plan considers the supporting workflows required for a successful product. Every part of your organization plays a part in post-launch, including marketing, sales, customer service and support, and other operational areas:

- Marketing needs to build the personas of the target audience to reach your users effectively. They need to perform competitor research, prepare promotional materials, create web content, and work on lead generation.
- Sales is involved in app store optimization, reviews, paid and referral marketing, and social media and influencer marketing.
- Customer service involves user engagement and retention, onboarding users, generating push notifications, and in-app messaging. They link users to helpful communications and communities.

The plan should also address other areas, including marketing automation and app monetization. Your new application needs a village to help it thrive.

Your approach depends on your user base. Post-launch execution varies between marketing an application to new users and rolling it out as a productivity tool to internal users. However, there is some overlap in terms of promotion and support. Both internal and external user bases are critical sources for usability studies and feedback. User prompts, surveys, app store reviews, and incentives for feedback are all valuable methods of improving the app throughout its lifecycle.

As you drive toward launch, keep all parties in the loop to prepare for support and minimize post-launch risks. Your post-launch plan must drive regular, tangible progress that delivers measurable ROI and app adoption after the initial launch.

RISK: **INADEQUATE RESOURCES**

You may have seen a poster that shows a product can meet only two of three goals — **good, fast, or cheap**. It's up to you to decide which of those aligns with your business goals.

A small prototype can be good and cheap. You can invest more later as you receive feedback. If you are presenting to investors, you might be fast and cheap to achieve financing. However, a high-quality product is never fast or cheap.

What would be the result if you realized after half of the project is complete that you don't have the right people to execute your planned application properly? Probably, you will be late and over budget. What if your organization lacks the management-level expertise to understand the relevant technology? Planning becomes a roadblock.

BEST PRACTICE: **STAFF AUGMENTATION AND OUTSOURCING**

Outsourcing and augmenting your staff with contractors is a cost-effective way to ensure you have all the resources you need to produce a successful application. You can execute on deadlines and budget allocations. Bring your due diligence when selecting a development team to hire, then use all the best practices above to keep the ball rolling.

Bringing in outside talent can bring expertise in key technologies to your in-house team. Perhaps you need to bolster project management. In that case, bring in a management partner to take care of your custom development project. As you begin initial planning for app development, perform a detailed evaluation of the developers needed to execute your design. Appropriate staffing has a substantial impact on your overall project timeline.

Benefits of Staff Augmentation

Staff augmentation is your best choice if you need specific developers or expertise over an undetermined time. However, if you need people to take overall responsibility for executing the development of a particular project, outsourcing is a cost and time-effective selection.

As you research the best staff augmentation or outsourcing group, look at reviews from others who use the group. Familiarize yourself with the group's technical and domain expertise.

You want people with experience in the most relevant, innovative technologies to build and implement your brilliant new app.

Ask the vendor questions about their development process, such as:

- Will they provide links and test apps?
- Will you have a team dedicated to your project?
- Will you have access to the code?

Lock down how the group handles reporting and communication. Also, determine if they have experience with similar applications.

Benefits of Outsourcing

Outsourcing helps obtain skills that are unavailable internally. You limit your risk by creating flexibility. You don't need perfect resource planning initially if you plan to outsource or augment, reducing the potential for delays.

Outsourcing also has the advantage of engaging a complete, project-based team. You have several types of outsourcing to choose from:

- **Location outsourcing:** This is defined by the physical distance between your organization and the outsourcer. You can bring talent onsite, outsource onshore to a provider in the same country as you, or offshore the work to a vendor in another country altogether. Nearshoring is the same as offshoring, except the vendor is in a country that shares geography or time zones with you.
- **Relationship outsourcing:** This is available in several styles. Staff augmentation allows you to keep ownership and responsibility for the entire project. A managed team enables you to share the responsibility and ownership of development projects, as all agree on who owns specific responsibilities.
- **Project-based outsourcing:** This leaves the majority of responsibility with the outsourcing provider. The provider owns the project from beginning to end.
- **Contract-based outsourcing:** This can be charged for a fixed price or for time and materials. The time and materials contract method is very flexible and can be scaled easily. If you haven't set a project scope or the scope is likely to change, charging for time and materials doesn't require you to set a price ahead of time.
- **Fixed-price outsourcing:** This is typically used in smaller projects with clear and unchanging requirements. The limited scope allows the vendor to set a firm price in advance. However, if complications or changes arise, you will be required to provide additional payment.

CONCLUSION

Modern application development comes with multiple risks, of which the seven issues above are some of the most common. There is no path to the perfect product plan because application development is complicated. You cannot account for every contingency. However, you can plan carefully and test as much and as early as possible to discover and resolve issues and mitigate risk. The earlier you can identify a delay, issue, or opportunity, the more leverage you have for completing a successful project.

Proactive planning lets you dynamically alter development details to achieve your core goals. A crucial part of planning and development is application security testing. [Security](#) is increasingly on the minds of business leaders, consumers, and the media.

Kiuwan provides the best security tools to ensure secure custom application development. Our solution integrates security into the DevOps process to bake security into your app from the start. You reduce your risk by finding and solving security issues throughout development.

Choose Kiuwan for your next application development project so you can identify and mitigate risk early. Bring your project to a successful launch and lifecycle by using best practices and early risk mitigation across all app development.

REQUEST A TRIAL AT [KIUWAN.COM/REQUEST-A-TRIAL](https://kiuwan.com/request-a-trial)
LEARN MORE AT [KIUWAN.COM](https://kiuwan.com)

GET IN TOUCH

Headquarters

2950 N Loop Freeway W, Ste 700
Houston, TX 77092, USA

United States [+1 732 895 9870](tel:+17328959870)
Asia-Pacific, Europe, Middle East and
Africa [+44 1628 684407](tel:+441628684407)

contact@kiuwan.com

Partnerships: partners@kiuwan.com