



Guide to End-to-End IoT Product Design



LEARN ABOUT IOT TECHNOLOGIES & DEVELOPMENT PROCESSES

This **Internet of Things (IoT)** primer covers the key *technologies* of the typical IoT solutions including:

- Connected devices
- Cloud applications and managed services
- Web portals
- Mobile applications

Throughout the document are descriptions of what to expect in terms of the *development process* including:

- Requirements gathering and your value proposition
- Design and development processes
- Quality assurance (QA) and testing

It also includes a description of what a good process looks like for the most important element of successful IoT System development – quality assurance. Sprinkled throughout the document are real-world examples of what Cardinal Peak encounters while providing product engineering services to our clients.

By the end of this document, you should:

- Understand what your technology options are
- Know what questions you need to consider
- Appreciate what the process will look like

Good luck and if you have any questions, please [reach out](#).

CONTENTS

Learn About IoT Technologies & Development Processes.....	1
Why IoT Matters	3
What is the Internet of Things?	4
Designing Your IoT Product	4
Where To Start: The Value Proposition	5
The “Things” That Make Up the IoT	7
How the Cloud Fits in the IoT	10
Other Cloud Solutions and IoT PaaS Platforms	12
IoT at the Edge	12
Machine Learning, Artificial Intelligence and Data Visualization	12
Creating Your Mobile App for the Internet of Things.....	13
What to Expect in the Mobile App Development Process	14
Quality Assurance Testing for Your IoT Product	15
Conclusion	18



Why IoT Matters

Our world is more connected than ever. From smart cities and connected cars to door locks, household appliances, dog collars, agricultural equipment, medical devices and so much more, IoT applications and connected devices have proliferated in a variety of use cases. Whether leveraging the power of the IoT to streamline operational efficiencies, better understand customer behaviors, improve customer service or enhance decision-making, organizations across industries are increasingly utilizing web-enabled smart devices to collect, send and act on data they acquire from their environments. What was once just a technology industry buzzword — the Internet of Things (IoT) — has since evolved into a connected ecosystem of billions of devices that are collecting and sharing data.

The Internet of Things (IoT) has evolved into a connected ecosystem of billions of devices collecting and sharing data.

This white paper provides a primer on the IoT product design process as well as the technologies involved (see Figure 1 below). Typical elements in an IoT system include devices, wireless connections, cloud services and a variety of user interfaces, along with a host of supporting standards and protocols. **While no single aspect of IoT product design and development is especially difficult, the sheer breadth of the solution requires a significant number of skills and myriad technologies.** As a result, IoT solutions are complex and require systematic testing throughout the development process.

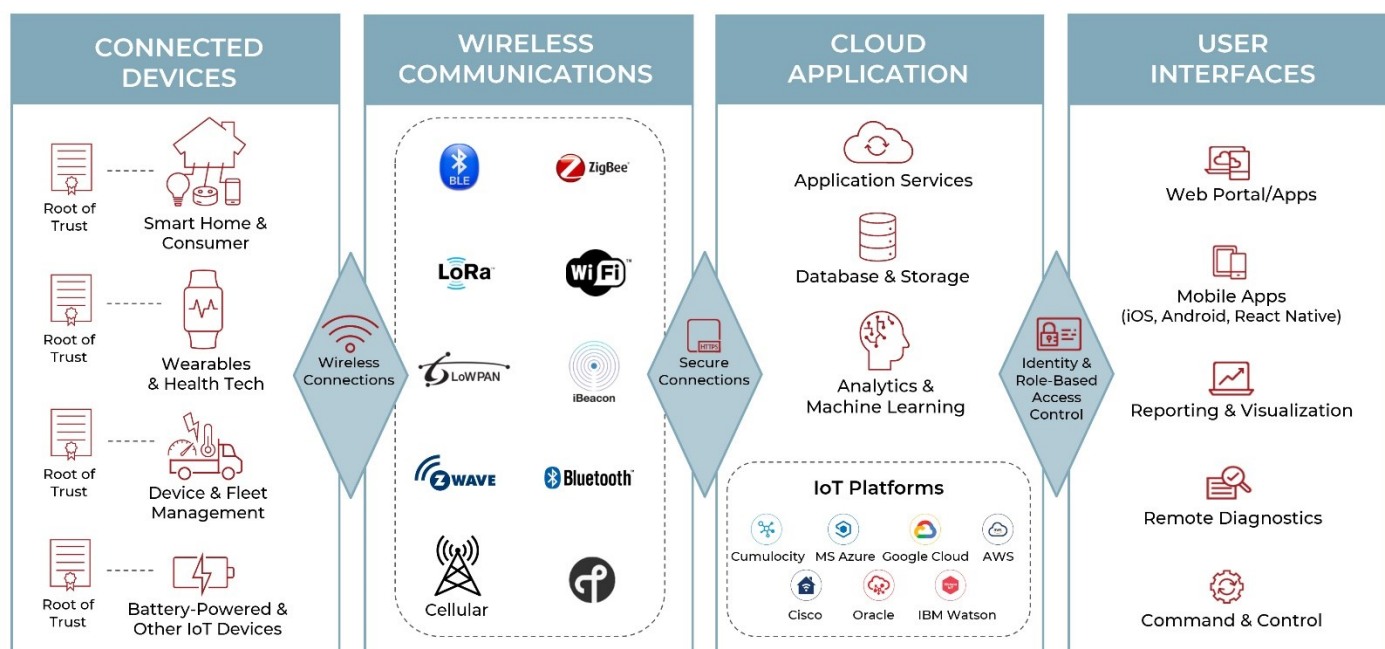


Figure 1: Elements of an Internet of Things (IoT) System

What is the Internet of Things?

The IoT is effectively a system of interconnected systems. Today, almost any electronic device can be made with enough computing power to communicate with other devices over the same networks that traditional computers use. These computing devices produce, collect and transfer data over a network without requiring human interaction or intervention to “talk” to one another.

The “things” of the Internet of Things are computing devices that communicate with other devices and computers connected to the network. Embedded with sensors, software and other technologies, the “things” that make up the IoT can be anything from a wearable fitness device on a person, a lightweight sensor attached to the ear of a farm animal, a connected vehicle or any other gadget that can be assigned an Internet Protocol (IP) address and transfer data over a network.

The number of IoT devices worldwide is predicted to almost triple from 2020 to 2030, to more than 25 billion products.

With the world becoming increasingly connected, IoT systems have experienced explosive growth in recent years. In fact, market and consumer data expert Statista predicts the number of IoT devices worldwide will almost triple from 8.74 billion in 2020 to more than 25.4 billion in 2030. **Now that we can connect everyday things to the internet, seamless communication is possible between people, processes and things.**

Designing Your IoT Product

At Cardinal Peak, we have accrued significant experience providing end-to-end IoT engineering services to a wide variety of customers over our 20-plus year history. This white paper is a primer on all the major components of an IoT system. Throughout this paper, we describe the process of developing an IoT product and the technologies used in IoT products as both are important to understand prior to launching into your own product design project. The overall design process is illustrated below.

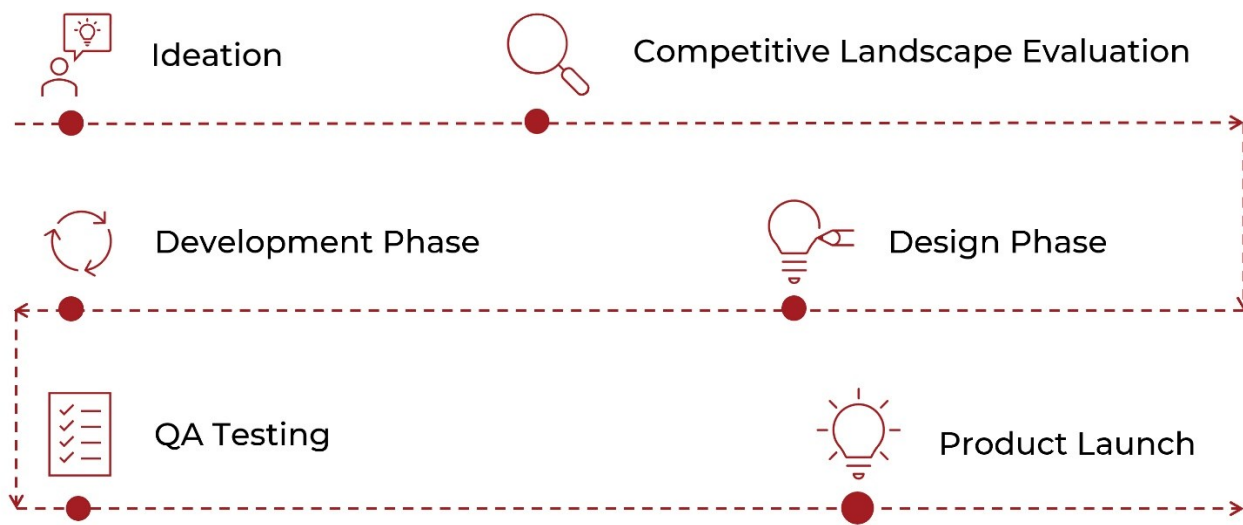


Figure 2: IoT Product Design and Development Process

Where To Start: The Value Proposition

Like all products, internet-connected devices start with the value proposition. We encourage starting with the value proposition(s) that can be satisfied by the cloud.

Why? Because the cloud is where you have the most data and therefore the opportunity to generate the most useful value proposition, as well as the “secret sauce” that sets your offering apart. Consider the classic Data, Information, Knowledge and Wisdom (DIKW) pyramid, or the wisdom hierarchy, but think of “wisdom” as intellectual property (IP), such as patents or trade secrets or ML/AI algorithms, as shown in the figure below. The peak of pyramid is your secret sauce, whether that be know-how, trade secrets, algorithms or patents.

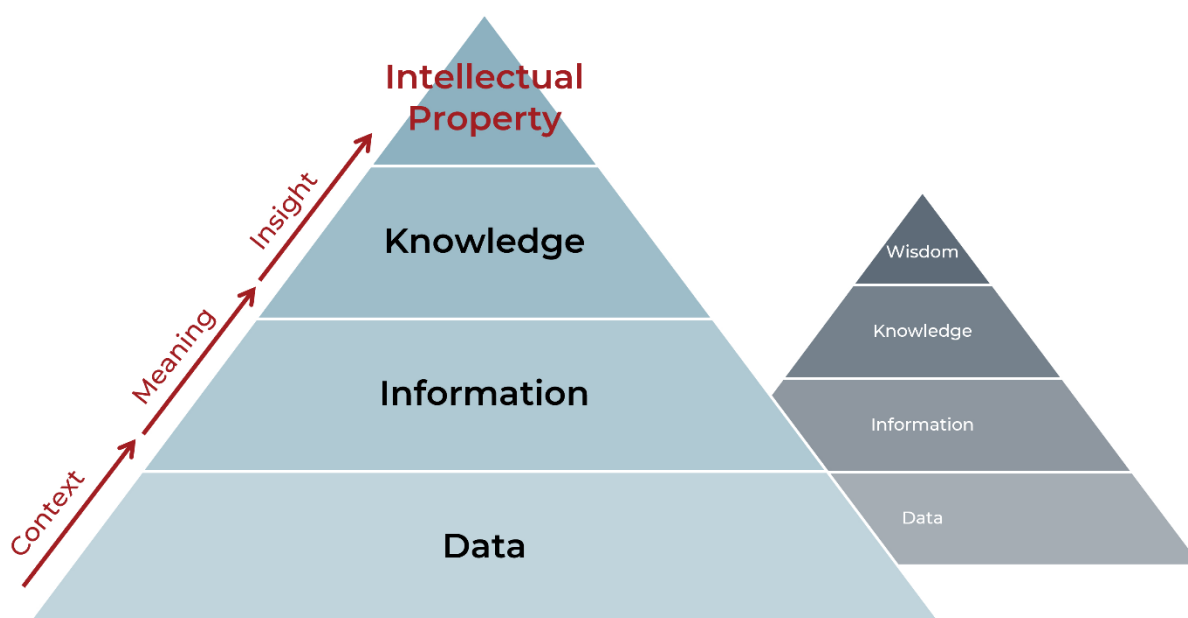


Figure 3: Data - Information - Knowledge - Intellectual Property (IP)/Wisdom

To illustrate this concept, let’s look at a simplified example that shows how connecting sensors maps to the DIKW pyramid and where you are most likely to find your unique secret sauce – by processing all the data in the cloud in some unique way.

- Data from a single IoT sensor, say an outdoor thermometer, at a single point in time is not that interesting. But imagine tens of thousands of thermometers located over a region that are collecting temperature measurements and **sending that data to the cloud**.
- Compiling all the data from the different sensors allows you to move up the pyramid and generate **information**, such as a table of temperatures. With actual information, you can start to provide more value.
- As you recognize trends in information and create actionable intelligence, you gain **knowledge** (e.g., a heat wave is rapidly moving across the country).
- If you can take that knowledge one step further and apply some machine learning or artificial intelligence, then you can potentially begin making predictions about the future and generate actual **wisdom — or IP** in this case.

For example, AccuWeather's Minutecast tells users when precipitation will start and stop at their location in the next two hours. It does this without the benefit of any weather models, but simply by looking at a series of radar images and performing image analysis.

While creating highly differentiated IP from a single temperature sensor is likely not possible, hopefully you recognize that at the nexus of the IoT system (the cloud), you have the maximum amount of data and stand the best chance of finding new and compelling value propositions.

When determining your cloud value proposition, it is important to be realistic about your likely deployment volume. If your value proposition requires a sensor every square mile, then you might not achieve success right out of the gate since success depends on a dense deployment of IoT sensors, which can be quite expensive. Instead, it might make more sense to plan to grow into some of your value propositions as you collect enough data to work your way up the DIKW pyramid to develop some IP.

Consider a client of ours' universal access control-as-a-service solution as an example of an evolving value proposition that is only possible through the cloud.



The cloud is the most likely place to find your IoT product's compelling value proposition.

Peak Experience: Connected Device Turned into Cloud Value Proposition

A customer of ours designed and developed the first Wi-Fi-connected door lock, assuming they would go on to become a leading manufacturer of Wi-Fi-connected locks. Over time, their business model changed as the company's focus shifted from one-time sale of hardware to recurring revenue from the software that supports the use of Wi-Fi-connected locks in the rental property market.

Their cloud-based access control and smart lock management software-as-a-service (SaaS) platform generates and provides codes to guests, cleaning services, maintenance teams and deliveries. Their software solution directly integrates with the backend of Airbnb, Guesty and other short-term/vacation rental property management platforms. It automatically sends access credentials to guests, tracks property entrances and exits, adds or revokes access for unplanned maintenance and more. All this ensures that there are no lost keys, no locked-out guests and no rekeying — all from a single, easy-to-use dashboard.

As this example illustrates, some of the most compelling value propositions, with recurring revenue sources, come from your cloud capabilities. The rest of your IoT system can be designed around fulfilling the cloud value propositions. The connected devices become the "things" you need to feed data into the cloud to enable the value proposition. And from there, users have user-specific portals and applications to view the outputs of the cloud and control operations.

Now that you understand the importance of identifying a successful value proposition, let's take a look at the different components of the Internet of Things ecosystem and some important connected device design and development considerations.

The “Things” That Make Up the IoT

As noted in the introduction, almost anything can be connected to the internet to change the world and how we interact with it. In the context of the Internet of Things, a “thing” is an electronic device that has a unique identifier (UID), an embedded system and the ability to transfer data over a network. IoT solutions require hardware components — whether off-the-shelf or custom-built — that serve a specific use case. **The thing will include a CPU (for logic execution), sensors (for data collection) and actuators (to act on commands).** These devices collect data and transmit it to the cloud (sometimes through a gateway or hub) to the cloud for analysis and storage. From there, the data collected can be accessed by consumers and/or businesses as actionable intelligence as shown in the following figure.

A “thing” is an electronic device that has a unique identifier (UID), an embedded system and the ability to transfer data over a network.

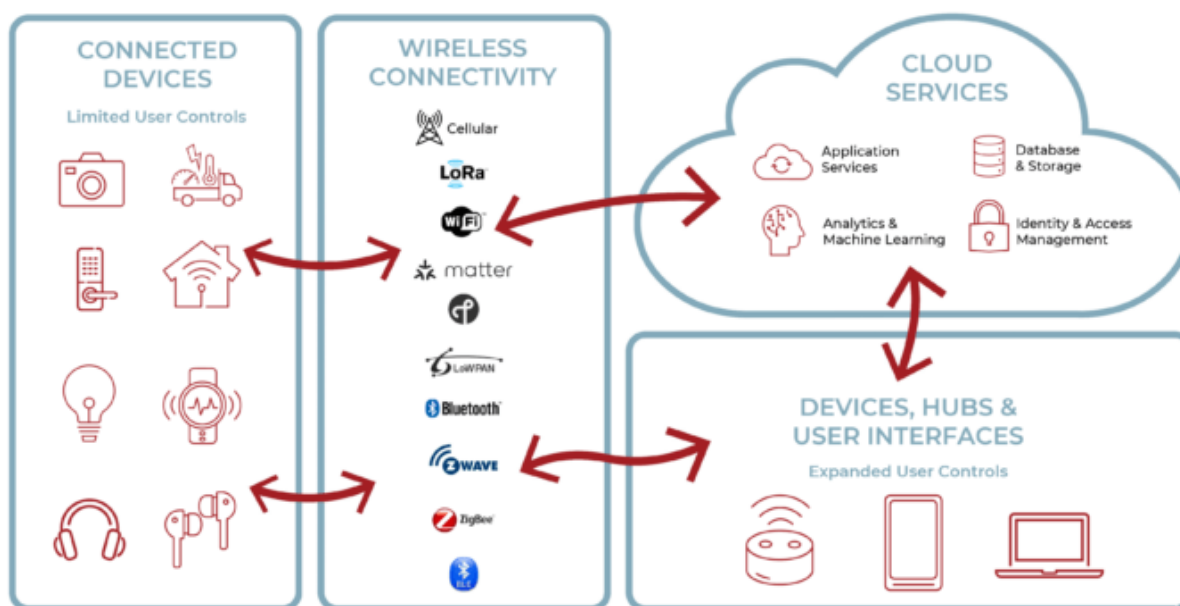


Figure 4: Connected Device Design

FAQ: Do you really need to design a custom “thing” to find your value proposition?

At Cardinal Peak, we’ve found that it isn’t *always* necessary to design your thing from the ground up. If you’re able to white label an existing product, then doing so can reduce your nonrecurring engineering (NRE) cost and speed your time to market. The downside with this approach is that you don’t own your product. If you need to make any changes, then you have to negotiate with your vendor to update it. If your “thing” is common (undifferentiated) in the market, then white labeling is likely a good path to travel down unless your sales volume is going to be so astronomically high that you want to wring every penny out of the design. If you are interested in white-labeled products, then you should look for an ODM (Original Design Manufacturer). For example, if you were looking for a white labeled smart speaker solution, then you could Google “white label smart speaker” or “smart speaker ODM.” Keep in mind that ODM’s will interview you to see if you are qualified to be their customer. They will ask a lot of questions on sales volume and your business plan.

FAQ: What if you need to design a custom “thing”?

If your “thing” is unique, then you will have to design a custom product from the ground up. Cardinal Peak helps customers design custom IoT products, so please reach out if you need support. **When designing a custom product, you can start by considering the following questions.**

Table 1: Considerations for Designing IoT Products

The Challenge	What problem are you solving?
Power	Will your product be powered by a battery, solar power or plugged into a wall outlet?
Communications	Whether transmitting information over short or long ranges, how will your product communicate with other devices?
Volume	How many units need to be produced and deployed to realize your value proposition?
Measurements	What parameters is your device measuring and how often is it taking measurements?
Controls	What do users need to control and how will they do so?
Accuracy	Will users expect your product to take highly accurate measurements?
Durability	Will users expect your thing to work in harsh environments?
Security	Will the data collected by the device be private and secure from cyberattacks?
Product Design Cost	How much money do you have for product design and development?
Business Model	What is your expected sales volume? What manufactured cost is necessary in order to be profitable?

One significant consideration to highlight is security. Every IoT white paper will cite the jarring statistics about the number of IoT devices with major security vulnerabilities. We will simply tell you what you need to do. First, do not make a device that doesn't support OTA (over-the-air) updates. Just don't do it. **Security vulnerabilities are inevitable, so firmware needs to be updated.** That is a fact of life, and every organization needs to come to terms with it. Consequently, you will need cloud tools for fleet management and software updates. Next, you need to think about the “root of trust.” The root of all security is the root of trust, a source that can always be trusted within a cryptographic system. In a nutshell, the root of trust boils down to who can your device talk to and how does that system know your device is who it says it is?

You might be able to reduce your NRE costs by selecting the right partners. Many silicon vendors are offering increasingly integrated solutions. Not only is additional functionality being rolled into single chips, but vendors are providing more pre-licensed software tools too. Vendors often license modules from other companies and include those licenses within the price of their parts. **While their volume can get you a very good price, there is no one-size-fits-all solution, so finding the right supplier depends on your product and the problem you're solving.**

**Think of your suppliers
as strategic partners.**

Our advice is to think of suppliers as your strategic partners. The more tailored their solution is to your needs, the lower your NRE will be — and the more difficult it will be to switch vendors later (as such, “strategic partners”). Partners are a double-edged sword that can decrease your time to market and reduce your costs but limit your flexibility later. At the end of the day, you want to ensure you're selecting a partner that will be there in the long run and that will value your business.



Peak Experience: Selecting a Strategic Partner

As a more concrete example of how you might choose a strategic partner, consider Silicon Labs (one of our design partners). Specializing in making radios for the IoT, Silicon Labs has all the major short-range radios that you would find in smart homes, including Bluetooth, Zigbee, Z-Wave and Wi-Fi as well as a few longer-range solutions, such as LoRa. One advantage of selecting Silicon Labs as your strategic partner is that the company has invested in the creation of development tools that effectively abstract the particular radio out, so the developer doesn't have to worry about whether this is a Zigbee or Z-Wave radio or one of the others. If you are planning to have different SKUs with different radios, these tools are great as most of the work is already done. For example, we had a customer that needed a Zigbee radio for the Lowes version of the product and a Bluetooth radio for the Home Depot version. Silicon Labs made it easy to create two different SKUs.

Additionally, Silicon Labs has figured out how to time-slice a radio so the same radio can talk two protocols, so you can effectively get both for the price of one. Finally, Silicon Labs development tools are the same no matter which radio you are using, so your team doesn't require expertise with each protocol. If you are looking to include multiple radios in your product or have different SKUs that use different radios, then Silicon Labs would be an excellent partner. Although, if you are just building a Wi-Fi (or any single radio) device, then Silicon Labs would be a fine vendor, but not really a strategic partner.

As you can see, when it comes to building your IoT device, you have some options and considerations.

How the Cloud Fits in the IoT

The cloud enables communication between IoT devices and applications along with managing devices and user accounts while performing business logic and data mining. The cloud provides an efficient, flexible and scalable platform for storing data and delivering the infrastructure and services needed to power IoT devices and applications.

The cloud is typically the brain of your IoT system in that it's where all the data and business logic reside, and it manages user accounts, security, and the devices themselves, including OTA updates. Some level of data storage and processing take place in the cloud, making the cloud central for drawing insights from that data. The cloud also allows for incredible scalability. When you have hundreds, thousands or even millions of things taking measurements, scaling would be challenging without cloud services such as AWS.

The cloud is typically the brain of your IoT system.

There are a variety IoT platforms out there, as well as tools (and patterns), to help you develop your cloud solution. You will want to take some time to find the right partner for your needs. Here at Cardinal Peak, Amazon Web Services (AWS) is our go-to partner for most IoT products that have differentiated features. Amazon continually improves this offering by building an ever-increasing portfolio of tools to manage common aspects of IoT systems. Amazon has many managed services to benefit connected products, including [AWS IoT Core](#), Kinesis Video, EC2, MQ, SageMaker, CloudWatch and others. AWS services are like Lego bricks in that you can build virtually anything with them far faster than building everything from scratch. Given the size of the tech giant, you can feel confident that Amazon will be a partner for the long haul. Some of the benefits of utilizing AWS for your cloud system are shown in the following figure.



Figure 5: Basic Cloud Benefits for an IoT System



Peak Experience: Developing Video IoT Systems With and Without the Cloud

If you don't have experience with cloud applications, then here is a case study that makes some of the concepts described above a little more concrete. Many IoT systems include video as a key component. Video can be challenging as it takes a large pipe to transfer video and video files can be huge. A long time ago, we developed a video IoT system from scratch. Imagine thousands of convenience stores, each with 5-10 video cameras. Much more recently, we developed a very similar system using Amazon's Kinesis Video. You can read the details in our "engineering cost savings and performance benefits Amazon Kinesis Video can unlock" blog, but the punch line is that it took 1/10 the labor to develop it using Kinesis.

With Amazon Kinesis, video and audio data storage is simplified. Cameras simply stream video to AWS using the tech giant's software development kit (SDK), and Kinesis automatically provisions and elastically scales all the infrastructure needed to ingest streaming video data from millions of devices. Plus, the tool durably stores, encrypts and indexes video data in streams, enabling organizations to ingest, buffer and process streaming data in real time. Even better, Kinesis is fully managed and runs streaming applications without requiring organizations to manage any infrastructure, thereby providing significant engineering savings and performance benefits.

While this example is focused on the common application of video, we could tell similar stories for other services that AWS delivers. Basically, when AWS recognizes a common problem that many of their customers have, they develop universal solutions that save their customers the time of having to develop everything from scratch. These well-tested solutions continue to be improved over time by AWS too.

Something else to consider if you haven't previously released a cloud application is that only one version of the program is running, so you don't need to push updates to millions of devices to fix a bug or add a feature to your software. This fundamentally changes the update cycle and allows for continuous development rather than fixed releases.

The cloud fundamentally changes update cycles because there is only one version of the program running.

Another example of the type of benefits you get from a cloud service provider is security. Developers don't develop custom code for standard security practices, they simply select the options that match your security policy in the cloud service. Examples include when a system asks you to change your password, notify you that a new device logged in on your account or prompt you to use two-factor authentication to log in. This reduces development costs and ensures everyone is using state-of-the-art security practices.

With the typical IoT system, you would develop several web portals for different stakeholders. As an example, you might have a user portal for letting users manage their devices. You will likely have an admin portal for fleet management, such as pushing OTA updates. If your business makes use of value-add resellers, then they might have a different portal that provides access to support their customers. These are all examples of the role-based-access-control web-portals that your cloud application could support.

While cloud solutions are infinitely cheaper than the old way of doing things, they are a significant monthly expense. Cloud service costs are a challenge to calculate because you pay for what you use, and each transaction has a tiny associated cost. Typically, you develop a use model to estimate costs which you use to design the most cost-efficient solution you can. Then post-release, you continue to tweak the system to reduce costs further. A good case study that you can review to see a real-world example of this type of operational cost optimization is “[scalable video cloud service using web sockets.](#)”

Other Cloud Solutions and IoT PaaS Platforms

While we highly recommend AWS, Amazon is not the only game in town. If your application integrates with Microsoft’s strong position in the office, then you might want to consider Azure as an alternative. And there are other classes of cloud solutions, too. While AWS and Azure empower you to build a custom solution, there are a variety of “IoT Platform-as-a-Service” offerings. These solutions are generally designed to address specific classes of IoT solutions, and as such limit your options, but they can also lower your initial development costs and help get you to market. A few examples of IoT PaaS solutions include Ayla Networks, Tuya and Afero. If this is a compelling solution for your need, we recommend that you review their websites for more information. You will also want to search for other competitors as there is a wide of variety of these platforms out there (hundreds).

In between a custom-built AWS solution and a stock IoT PaaS, ThingLogix offers another type of solution. They have developed an AWS pattern that you can start using right away and then build on top of when the need arises.

While it is impossible to present all the permutations of cloud systems, hopefully this primer has given your big picture options. Please feel free to contact us to discuss your needs, as every product and every brand is unique.

IoT at the Edge



Edge computing is a distributed information technology architecture that involves the deployment of computing and storage resources at the location where data is produced. By moving away from cloud data centers that might be thousands of miles away, edge computing emphasizes reducing latency and providing more processing of data closer to the source of the data, the IoT product.

If you have been enjoying this primer and are interested in learning how edge computing can be utilized in your IoT system, please [click here](#) to sign-up to be notified when our next white paper is ready for download.

Machine Learning, Artificial Intelligence and Data Visualization



If you have been enjoying this primer and are interested in learning how machine learning (ML) and artificial intelligence (AI) can be utilized in your IoT solutions, please [click here](#) to sign-up to be notified when our next white paper is ready for download.

Creating Your Mobile App for the Internet of Things

Depending on your specific IoT solution, you may or may not need a smartphone/tablet app. End-user apps are standard for most consumer products and less prevalent for industrial IoT applications which tend to use web portals for data visualization and device management. We will let you decide whether your users will expect a mobile app for your product.

The typical end-user features of a mobile app include:

- Device registration/onboarding
- Account management
- Device control
- Data visualization

IoT smartphone apps are a little different than other types of smartphone apps, so if you are hiring another firm to develop your app, we highly recommend hiring one that is used to developing IoT apps as it will greatly increase your likelihood of success. We have found that it is critical for both app developers and the QA test team to understand how the hardware works.

As one example (of many), let's consider an app for controlling the lowly space heater. The space heater, like most embedded applications, uses what is called a state machine which basically means that the device is always in one of several possible states. There are generally logical rules that need to be met to move from one state to another, which the developers and testers need to understand. For example, when the

space heater in our example is told to start heating, it goes into a state where the coils are heated up before the fan turns on. This might take 30 seconds during which time the user could be concerned that the device did not receive the command from the app (and as a result they mash the button again and again). This is (at best) a terrible user experience and could potentially be worse if the app keeps sending more and more commands to the device as the impatient user keeps pushing buttons trying to get something to happen. A developer that understands the concept of the state machine will already have asked the right questions that allow her/him to provide cues to the user like a message that says "warming up" with a little count down timer or similar. Likewise, a QA tester familiar with IoT devices will be on the lookout for these types of issues.

Mobile apps for IoT devices require a special understanding of how the application code and hardware work together.

If you determine that your IoT device needs a mobile application, then you need to decide whether it will run on iPhones/iPad, Android devices or both. Below are the three basic technology options for smart device applications.

Native Mobile App Development

Native mobile applications are written in the programming language and use the frameworks provided by the platform owner. Cardinal Peak develops native iOS and native Android applications. The benefits of native applications include speed, access to the hardware on the smartphone, and an interface that matches the design language of the OS. If you are developing for both Android and iOS, then the disadvantage of native application development is cost as there is very little beyond basic UX that can be shared across the two platforms.

Hybrid Mobile App Development

Cross-platform mobile applications can be written in a variety of programming languages and frameworks before they are compiled into a native application that runs directly on the device operating system. While we've developed apps in several cross-platform tools, our standard is React Native. It provides the best tools to streamline development and launches applications that look and feel like native apps, running at comparable speed. Developing in React Native reduces the costs of developing two standalone apps (Android and iOS). The ability to blend Native code for accessing hardware makes it a great solution for applications that need access to the smartphone hardware and therefore cannot run in a browser.

Progressive Web App Development

Progressive web apps (PWAs) skip the app store delivery and app installations — they're web applications that utilize a set of browser capabilities and appear as a shortcut on the device home screen. Despite having limited access to hardware components when compared to native apps, PWAs do have a distinct advantage: they are device independent. A leader in audio, video and IoT app development, Cardinal Peak can support your PWA development to deliver apps with a UX on par with native apps.

What to Expect in the Mobile App Development Process

While other organizations processes may vary, our typical mobile application development starts with discovery to understand your goals, the product features and your brand's identity to guide you through user experience (UX) and user interface (UI) development.

Discovery scales depending on your application. For a simpler app, we have weekly video calls to explore a portion of the app. During the week, we develop clickable models which are used in the following week's meeting to solicit feedback. For more complex apps, we have a multi-day workshop at your facility to dig deeper. A couple weeks later, we present an array of design options for feedback.

Once the UI is laid out, the developers start their work. The number of developers depends on the complexity of the project and the desired schedule. We typically run on 2-week sprints as described in our [project execution blog post](#). With each sprint, we strive to execute vertical slices of features so both QA and the customer can start evaluating the app as soon as possible.

Once the app is feature-complete, we typically have a quality sprint (where no new features are introduced) before starting the "friends and family" alpha testing. While you pick your friends and family, we manage all other aspects of the testing. We may expand the testing if needed, then we are ready for product release.

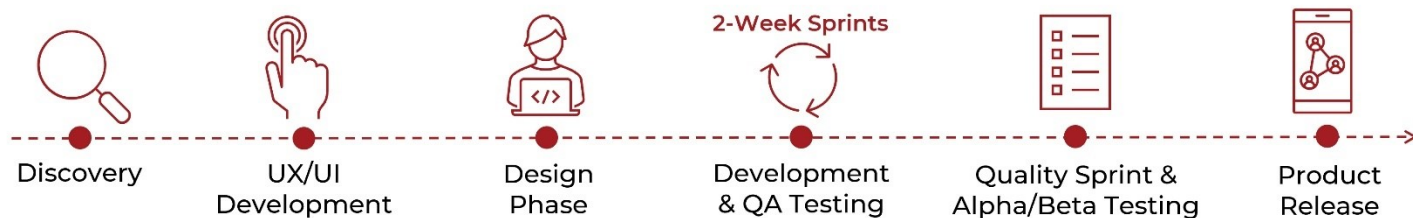


Figure 6: The Mobile App Development Process

For those curious readers that have heard the terms: front-end, back-end, SDK and API, a short description on each is provided below.

FAQ: What is the front end vs. the back end?

Most mobile apps are not standalone applications but instead provide a way for users to view and interact with data stored in cloud systems. The mobile “front end” is the app, while the “back end” is the cloud application where the data lives. For example, our Yonder app’s user interface (the music player) is the mobile front end while the back-end cloud serves up the content (the music files, artist information, social content, etc.). Sometimes, we are a little crafty and might bridge directly from the app to a device (like a video camera) to save costs, but the basic idea is the same: The mobile app is the front end and content comes from somewhere else.

FAQ: How do the front end and back end communicate?

The mobile front end obtains the data from the cloud back end via a variety of service calls, such as application programming interfaces. In some cases, these APIs may be developed and owned by the same company developing the mobile application, while in others, the API may be controlled by a third party and access is granted via a commercial agreement.

FAQ: How do mobile apps use SDKs and APIs to integrate with the cloud back end?

Back-end services are typically accessed through a variety of different types of APIs. While some back-end service APIs are available directly to the application through calls in the platform itself, many of the specialized services must be integrated into the app via a software development kit, commonly known as an SDK. Once the SDK has been added to the app via the development environment, then the application can make use of the APIs defined in the SDK.

Hopefully the description above gives you a background the technologies used in smart device apps along with an understanding of what to expect in the development process.

Quality Assurance Testing for Your IoT Product

To be successful in the market these days requires a product to reach a very high level of quality. This level of quality is not reached by accident or because you have great engineers. This high level of quality can only be reached by having a great QA process and a team that follows it.

IoT system testing can be quite challenging since so many interconnected technologies are used from hardware through connectivity to cloud and end-user applications. What might look like an app bug (because the system doesn’t do what it is supposed to when the button is pressed) could easily be a cloud or embedded software defect. A great QA process and team make the development process efficient by correctly allocating defects to the right sub-system and providing detailed information to the developer on how to reproduce the error.

Successful IoT products must meet a very high level of quality, attained only through great engineering.

The figure below shows our overall QA process from initial requirements through sustaining support.

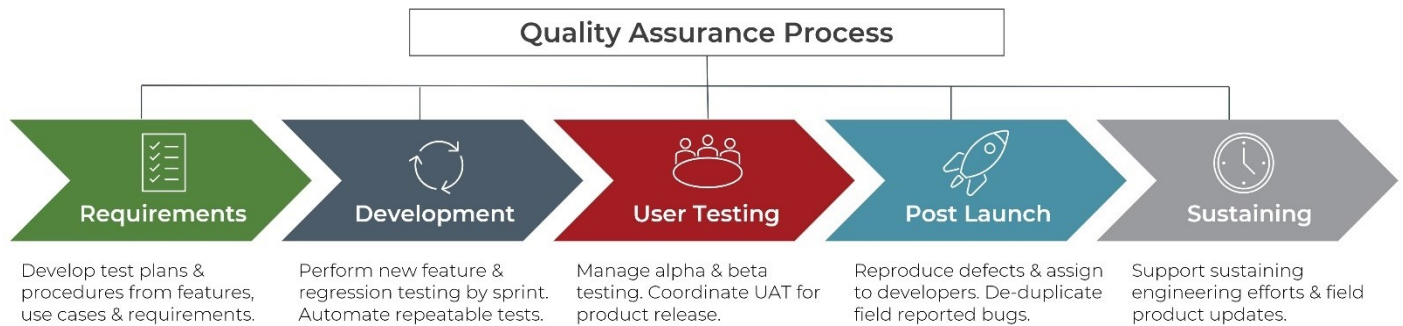


Figure 7: Our QA Process

Highlighting the second box in the figure above (“Development”), QA is part of every sprint. There are two sets of tests that QA will do with every sprint (typically 2 weeks). First, QA will test each new feature that was completed during the sprint against the requirements. If the feature passes, they close the JIRA ticket. If the feature doesn’t pass, then they point it back to the developer with an explanation of any defects and how to reproduce them. These are typically manual tests as they are only done once or twice.

In addition to new feature testing, QA will perform limited regression testing with each sprint. Regression (or integration) testing is important because sometimes when a new feature is added, it breaks something that previously worked (i.e., the software “regresses”).

You don’t want to save all the integration testing for the end of the project as that gives a false sense of where you are in the project. A process that doesn’t address quality issues right away is building technical debt which will eventually have to be repaid. Regression testing is often automated as the tests are repeated frequently both during the development phase as well as post launch when defects are fixed, or new features are added. The figure below shows how QA fits into each sprint.

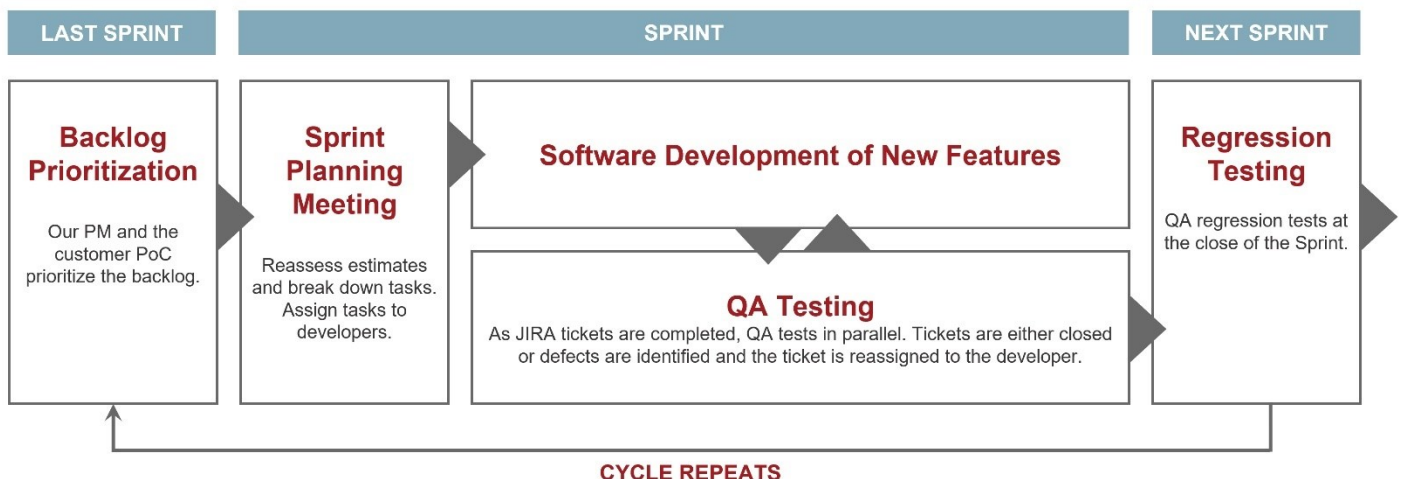


Figure 8: QA in the typical sprint development cycle

FAQ: What are the steps in the QA testing process?

The primary steps in the QA testing process are:

- Gathering and defining requirements.
- Studying and familiarizing with the product and any new feature development.
- Creating a feedback loop of test planning, test case creation, test execution, results reporting and test improvement.
- Repeating this process until the product is ready for launch.

FAQ: What are some IoT testing factors?

While this list isn't exhaustive, it highlights the myriad of factors that can go wrong and cements the value of engineering QA testing into the product development process. That's why it's critical to understand as much about your product, its value proposition, how it will work, who will be using it and how it will be used as soon as possible. With that understanding, you can detect any defects or unmet requirements early on in the development process — and ultimately rework your product to ensure the release of a higher-quality product in the end.

- Usability
- Functionality
- Security
- Compatibility
- Connectivity
- Regulatory compliance
- Real-time data
- Real-world scenarios
- Upgrades
- And more

At the end of the day, if you don't test any aspect of your IoT system, you can't be sure that it works. Our QA team comprises 15-20% of our entire staff and works continuously through every project to find and correct any issues before they become final-stage surprises. If you are building a product for commercial release, then you want to make sure you have QA built into your process. If you are hiring a design firm, then you want to explore how they treat QA. If it isn't done until the end (or they believe that your customers are testers), then you haven't found the right firm.

By testing continuously throughout the product development process, you can identify and correct any issues before they become final-stage surprises.



Conclusion

With the IoT now all around us, delivering instantaneous information and enabling us to make spur-of-the-moment decisions, the world is more connected than ever before. Promising the tremendous ability to tackle real-world challenges, the IoT delivers numerous benefits to consumers as well as enterprises. However, designing and developing products that seamlessly integrate with mobile applications and cloud systems to deliver value to users is not a simple process.

Nearly three in four IoT projects fail, and when you consider the resources that most companies invest in developing IoT solutions, that percentage is even more eye-opening. **By their nature, IoT systems are inherently complicated and require detailed knowledge of communications protocols, security standards, data collection, AI/ML and analytics.**

At the most basic level, IoT solutions consist of:

- **“Things”** — Small electronic devices with embedded communications to transfer data over a network.
- **The Cloud** — A centralized network computing facility that manages the “things” and user accounts.
- **Mobile Apps, Dashboards & Web-Portals** — Applications that allows users and operators to control their devices and view data produced.

In addition to these basic building blocks, it is our sincere hope that this white paper makes it abundantly clear that you should also consider the importance of QA testing during IoT product design and development.

At Cardinal Peak, we leverage extensive experience with IoT product design and development to empower our clients with the ability to deliver innovative solutions that meet their unique needs. With deep expertise in audio, video, voice, IoT, UX/UI, hardware, embedded software, cloud, mobile app development and quality assurance testing, we intimately understand everything that goes into launching even the most complex IoT products. We’ve worked with dozens of target technologies, and the likelihood that we’ve had success in your market segment is strong. Our proven process is tailored to bring your innovation to market.

If you need help to design, develop, launch and support an innovative, best-in-class IoT product, let us know how we can bring your innovation to life!



CARDINAL PEAK, LLC

1380 Forest Park Circle, Suite 202
Lafayette, CO 80026-3378
303.665.3962
cardinalpeak.com
info@cardinalpeak.com