

The Internet of Things: Best Practices You Need to Know About

INTRODUCTION

Yes, the Internet of Things (or "IoT" for short) is a network of billions of sensors and devices - but it's important to not let that number fool you. The true strength of this concept rests in their ability to come together to form something much more powerful than any one of them could be on their own - something with the potential to transform not only the way your organization runs today, but to also uncover the type of insight you'll need to remain agile and moving forward for years to come. This is particularly true in the world of software development, where even seemingly small decisions can have significant implications down the road.

If you truly want to guarantee that your IoT deployment is enabling you to work "smarter, not harder," there are a few key best practices you'll need to keep in mind.

It's All About Data

As is expected with an infrastructure made up of massive amounts of tiny pieces, the Internet of Things can (and will)

generate equally massive volumes of data. With that in mind, the decision of how and where to store that data is perhaps the most important you will make - especially early on in this process.

For the best results, you need to be able to determine which data remains on the edge and which data is valuable enough to actually make the trip all the way to your enterprise core. Being able to separate "data" from "quality, critical data" as quickly as possible is one of the keys to this. Your decisions regarding where and how your data is stored will affect nearly everything, from how easily you and your teams are

Agility is truly the name of the game, particularly in the modern era of enterprise software development.

able to communicate to how many tiers a particular application requires to what sort of I/O engine you need on the backend and more.

Figure out the data side of the equation - and how much data you want to deal with once it's left the edge of the application - first.

Don't Forget About the Platform

The type of platform you choose for your IoT deployment matters a great deal, as the requirements for a network with a few hundred end users versus a few hundred thousand will obviously vary a great deal. You could get by with something like an Arduino or even a Raspberry Pi-powered platform for the former - the latter is going to require something compact and rugged that can be easily gang-programmed and inserted into the device.

Keep in mind that some embedded systems are NOT infinitely interchangeable. You need to be able to think about the long-term implications of whatever platform you choose, bringing both the software development and hardware side of the equation together to support whatever long-term goals you have in mind. Your ability to see this bigger, longer picture will impact everything - from the immediate success of a device to your ability to adapt, revise and change its design as the need arises in the future.

Security, Security, Security

Another major consideration in terms of the IoT is one of security. Keep in mind that any device connected to the Internet is a potential vulnerability just waiting to be exploited by someone who knows what they're doing. Now, think about the countless tiny sensors that are ALL connected to the Internet and sharing data with one another and you can begin to get a better understanding of just why security is so important.

Truth be told, security needs to be built into the design of the system from the very beginning. Don't be afraid to get paranoid - the consequences of not doing exactly that are far too severe. At Pegasus One, we even let pen testers have a go at our systems early and often in the process because the one thing we do NOT believe in when it comes to security is "taking too many precautions."

Contextualizing the Internet of Things

Yes, the IoT is a series of "things" all connected together - but what does that really mean in terms of what you're trying to accomplish? One of the major strengths of the IoT is that it is malleable - it can essentially be anything you need it to be with a little forethought. But that's the problem, too - without a clear vision in mind, it's easy to get lost along the way.

Think carefully about the nature of the system that you're building and the devices that sit on the perimeter and don't allow yourself to be convinced that "one size fits all" when it comes to the "things" you're introducing to the Internet.

Agility is King

Agility is truly the name of the game, particularly in the modern era of enterprise software development. In order to allow the Internet of Things to actually support this, however, you must throw out the "write once and forget it" discipline that has existed in embedded design for years. Those silos need to be shattered and nothing can exist in isolation. If you're not already teaching your embedded team all about agile development, now is absolutely the right time to start.

It's About Remaining Light

Another one of the most important best practices to consider in the context of the IoT is that you should never make your IoT-powered software (or the hardware that supports it) any bigger than it needs to be. Resist the urge to add functionality just for the sake of it - the modern era of software development just doesn't support this idea anymore.

To be perfectly honest, the "KISS" principle was never more appropriate than in the IoT. Ideally, you should be thinking about it like this. Gone are the days where solutions need to do as many things as humanly possible to gain traction - particularly in an IoT-based world. That's how you become a "Jack of All Trades, Master of None." For the best results, don't try to do a dozen things poorly - try to do one thing incredibly well. This is how you get ahead in the modern era.

Look for reasons to take things out, edit feature lists ruthlessly, and follow the writers' advice to "kill your darlings." The final product will be better for it.



Test, Test and then Test Again

Think about all of the best practices that led you to this point. You need to focus on security. You need to remain as agile as possible. You need to make "lightweight" a priority. The only way to guarantee all of these things is to test early and often, ad infinitum. Take the time to test every system with a variety of unintended use-cases, and under no circumstance neglect security testing.

Every time you make some type of change or implement a new feature, it needs to be tested - both against the use cases you can predict but especially against those you can't. Remember that in the era of

the MVP (minimum viable product), you'll be finding constant opportunities for improvement even after your initial roll-out. The key is to make use of them whenever they reveal themselves and that involves constant testing.

Make no mistake about it: your end users AND your legal team will thank you for taking the time to test.





Pegasus One

1440 N Harbor Blvd #900, Fullerton, CA 92835, USA

Phone: +1 (714) 485-8104