

New York
Seattle
Minneapolis

How to Future-Proof Your IoT Software Solution

Danny Aponte, *Senior Director of Software Engineering*

July | 2017

Designing IoT Software that Lasts

Software users' ever-changing expectations are high, and they continue to grow. Whether you are working on a mission-critical enterprise system or this is your first entrepreneurial venture, there's a great deal to consider before you bring your program or app to market.

In great software, user experience is not just about the interface, quality assurance is not making sure just the "go-right" path is working, and documentation isn't a benign afterthought.

Any successful new product is created for the end-user. Software has a sophisticated level of involvement, which puts experience at the forefront. This suggests that your amazing new solution will best be developed with an iterative approach that focuses on the user early and often during the process.

The development of precise and security-critical applications requires a rigorous process. The collaborative software stack discussed in this paper offers a **team environment** for business professionals and software developers, wherein ongoing testing and iterative development is the key to success.

IoT is impacting the software design processes across business sectors. The powerful opportunities it presents spans industries and affects both B2B and B2C markets. Gartner reports that "by 2020, more than half of new business processes and systems will incorporate some element of the Internet of Things." As embedded software becomes more and more mainstream, technology and business leaders may find it's necessary to re-evaluation their software development methodologies.

Software Collaborators are needed to develop solutions that simply work, and work correctly. A goal-driven collaborator process goes beyond agile, making it bendable and fluid enough to produce incrementally-improving software.

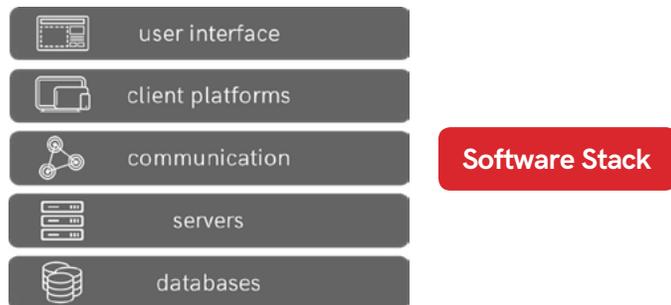
Health Care Improves with IoT Software

Application developers have been experiencing tremendous opportunities in the healthcare market with the growth of automated centralized solutions. The Internet of Things (IoT) plays a role in many of the new applications that are finding success in the medical industry. Embedded software, such as patient sensors, address a new way for medical professionals to track patients' vital signs and activity and then stream the information to the medical center's data storage infrastructure.



Traversing the Software Stack

The components that are necessary for winning software are essentially the same across industry, user, and the application's purpose. Companies can choose the best development path by first understanding the full range of the software stack.



User Interface

One might argue that the user interface is the most critical component of software development. Why? Because your end-user interacts with the front-end design. It's their first impression. *Perhaps their only impression.* It's valid to say that if your user is presented with an unappealing and complicated screen, there's a chance you lose them before the application has a chance to show what it can do. So, from that perspective, yes, interface design makes a major contribution to the user experience.

The user interface must be simple, yet attractive, and the layout must draw attention to important content headlines. *Interactive objects are of foremost consideration.* Clarity is key. Great designers know that the user should not be required to think. Menus, buttons, forms, and other call-to-action items should be easy to identify and easier to use.

Circling back to the ranking importance of each stack component, what happens when a user clicks on a registration button and nothing happens? Read on.

Client Platform

Software has an obligation to perform perfectly and equally well on various operating systems and across all screen sizes available to the user.

In the not so distant past, designated operating systems determined how desktop software was built. Simply put, programs were either developed to work on Windows or a Mac (ok, mostly on just Windows). IoT and cloud computing has completely changed this paradigm. If you're developing cloud-based software, your team needs to understand the role of Platform as a Service and Infrastructure as a Service technologies (such as Amazon Web Services or Microsoft Azure).

Another challenge is assuring your software will work on mobile platforms. True cross-platform mobile development ensures that the native application operates consistently across Android, iOS, and other mobile operating systems, while still meeting the unique experience demands of each platform. Comprehensive testing is critical.

Client platform performance is another component that affects the overall user experience. Getting back to the point previously made, an exquisite and fine-tuned user interface is worthless if it doesn't perform well on your user's device.

Communication

A scalable communication interface connects systems across a room or across the globe. This is how software does its job - through the design of the application's information flow.

New digital innovations in communications are born nearly every week. When making these technology decisions, the available options far exceed the desirable ones. You need an expert on your software team who can stay abreast of this ever-changing world. With this, the

ideal solution for your product can be extracted through a *precise vetting system*.

Great software designers can confidently and successfully use the best of these revolutionary infant technologies while avoiding the transient ones that are here today, gone tomorrow. To leverage communication innovations, development should use a "Swiss Army Knife" style. Understanding how a new interface can integrate with your current technology choices will result in an efficient and versatile communication platform.

Servers

Enterprise frameworks and distributed systems are built with cutting-edge technologies. Traditionally known as full-stack development, expertise in the server portion of the software stack requires skills in both front-end and back-end technologies.

Even if it isn't obvious, most applications used daily are connecting to a back-end cloud server in one form or another, and many use IoT technology. Whether you're interacting with a web-based GUI, or you're using a mobile app that syncs its data across all your devices, the server infrastructure is a critical component of the overall experience.

Building with the future in mind entails thinking through how the system will scale when your solution requires 1,000, 10,000 or even 100,000 end points. It requires not

only robust, modular coding, but also IT infrastructure and failover expertise. Above all, your quality assurance disciplines must be top-notch to ensure that your solution is ready to handle the broad spectrum of failure scenarios generated by your global user base.

Databases

As a foundational element to any connected solution, the underlying data models and storage technology must be built to last. This absolute requirement is well understood in the industry. But in today's fast-paced agile world with emerging IoT opportunities, the key is to start small, leaving room to grow.

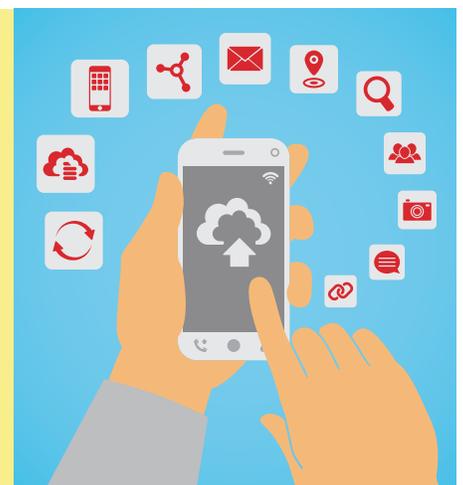
As with each stack component, database development is a discipline that requires a specific area of expertise. Cloud storage addresses the ever-growing need for data. Robust architectures, such as SQLServer, mongoDB, and PostgreSQL can supply secure and widely available data storage to all your connected client devices. Meanwhile, mature micro-infrastructures like Realm and SQLite power your mobile application to cache data for fast retrieval in the palm of a user's hand. Whichever solution is determined for your software program, be sure that security is in the vanguard of its data system.

In many cases, your database will live on for the life of your solution, even while mobile and front-end technologies are changed out. Be sure that interoperability and scalability are set goals during the architecture stage.

Considering a Custom Cloud-Based Application for Your Business Processes?

An IDC Report, *Custom Applications Powered and Enabled by Cloud Computing*, calls out the benefits of leveraging cloud-based custom application development services for a custom solution over packaged software or in-house development.

- ▶ Faster end-to-end process execution for higher levels of automation
- ▶ More effective utilization of resources on innovation rather than application development
- ▶ Reduced ongoing maintenance and upkeep costs by internal staff





Traditional Software Firms

There are a few routes you can take to complete a software development project. Be prepared to interview potential software developers to find a solution provider who best fits your needs.

Software firms often have various areas of expertise, which can veer to either front-end usability or back-end functionality, or somewhere in between.

Front-end Developers

These are the design-based firms. Their focus is typically on the first two stack components: the user interface and client platform stages of development. The result can be a compelling software interface that looks beautiful; but without an equal focus on the bottom layers of the stack, you may find that not enough attention is given to structure and functionality.

Back-end Developers

This group can create great systems with thorough and accurate reporting and an emphasis on capabilities. This focus is great, as long as product design and user experience aren't compromised. Keep in mind that if the user interface isn't extremely intuitive, users won't fully utilize the wonderful algorithms that these developers build.

A Balanced Team

A development team that understands the need for a compelling yet simply intuitive design, and pairs it with a robust, sophisticated back-end can have the best of both worlds. In today's extremely fast-paced, instant-gratification software environment, it's critical that your software team can slice vertically through the stack, and not horizontally.

The Finished Product

Many software development firms produce a great product. The important consideration is whether they can deliver the set goals of each stack component and be sure that those mini-goals culminate into the broader vision of the software solution. Otherwise, it won't meet your user's needs.

It's up to you to make sure the user experience isn't lost along the way. If you find the need to engage additional resources to complete the project, you run the risk of diluting the integrity of the team process. Project costs will increase and your product launch can be delayed.

Collaborators' Approach to Software Development

A process that leverages a team of Software Collaborators puts speed and structure around software development.

Testing and gathering critical feedback is what makes product development a success. Ideally, you want to get a version of the software in as many users' hands as soon as possible and be prepared to address their feedback. Traditional linear development methodology, known as waterfall, can fall short in this area. The formal process and change controls used with waterfall fail to meet the reality of rapidly-changing user requirements. The drawback of waterfall projects lie in the lack of ongoing customer feedback. If your requirements aren't on point, expectations will not be met.

Iterative, cross-platform software collaboration is more advantageous due to its **flexibility** and **direct connection** to the end user. An iterative approach is micro-goal driven, with an easy process in place to adjust to changing requirements. This improves your team's ability to accurately and efficiently manage costs, and it provides the quickest route to market.

How is this accomplished?

The compound annual growth rate (CAGR) for IoT applications and analytic software is expected to increase 40% by 2020.

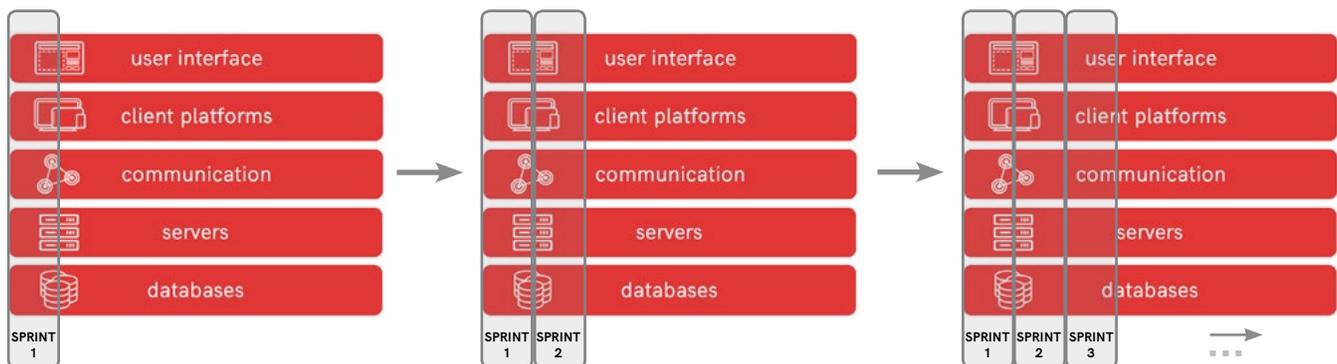
Source: Boston Consulting Group (BCG)

Born out of the Agile methodology, iterating through vertical slices in the software stack is not a sequential, straight-line process. Every component in the stack is addressed in a series of continuous, quick-moving sprints that deliver a market-ready product at the end of each. The project moves in manageable pieces. There are no surprises.

Communication is vital to collaboration. Developers are in ongoing contact with the customer through daily scrums, biweekly planning meetings, and biweekly reviews. This accelerates decisions, including changes, throughout the process. There is no opportunity to go down deep rabbit holes or misstep on the scope implementation. **This level of governance keeps the variance narrow.**

Effective Agile Methodology

IPS's team is well-versed in all areas of the stack. Their "vertical slice" approach provides a great framework for efficiency in building full software solutions and incrementally improving software.



The Stack Revisited: Vertical-Slice Software Collaboration

Take another look at the software development stack. The ideas hold true, but by slicing vertically you will enrich the software creation process. For example, we've already discussed the importance of scalability. What happens if your database needs to be changed during deployment? When you develop in slices, variances are negligible and the client gets exactly what they need.

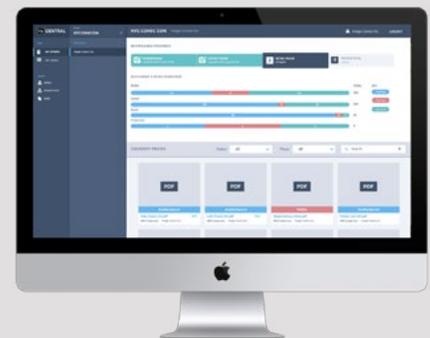
Here's how slicing works with the software stack you're already familiar with:

- ▶ **User Interface:** Wire framing should be used to model the basic structures of the program. With each sprint, usability requirements can be fleshed out for the specific functionality assigned to the slice.
- ▶ **Client Platform:** A slice calls for a specific feature on a particular device, an incremental addition to the existing capabilities of the application.
- ▶ **Communication:** Connectivity to the backend is an essential part of the slice, enabling a full round trip to the server and data storage to empower the client front-end. New IoT product complexities further intensify the importance of security testing.
- ▶ **Servers:** Server interfaces are built according to the requirements of the client, using scalable and flexible frameworks, conserving costs, and avoiding missteps in scope. This is achieved with the user always at the center of changes.
- ▶ **Databases:** Data models required by the front-end are developed as needed, but require an architectural vision to prevent the buildup of technical debt.

For each functionality set, the user interface team will build a click-through prototype that you can touch, feel and test early in the process. Then, as the two-week milestones are delivered and approved, the application becomes shippable as a revenue-generating product.

Collaboration is the Main Event

IPS' vertical slice approach was a win for T3 Central, a digital command center for corporate and trade association events. Planning events takes precise coordination. T3 Central makes it easy for event managers, with centralized workflows, reviews, approvals, planning, budgeting and reporting. This all rolls-up to an execute level dashboard where managers can see the overall event status and what needs to be focused on next to stay on time and on budget. It also keeps a full history of work in queue and work completed, and generates detailed reports to help you manage your internal and external constituents.



Centralized Logistics Tracker

T3expo®

Bring on the Software Collaborators

The expertise maintained by the IPS team allows clients to develop products that use the latest technology. **But not just any technology.** Only well-vetted development tools are used. It's important not to reinvent the wheel, but to always be watching for new wheels. Determine what is destined to be a new industry standard.

Aberdeen reports that the increased complexity forces companies to focus on a shorter software timeline schedule, and software developed in a vertical iterative project process are 58% more likely to meet their launch date than all others.

Most IPS software projects consist of an 8 to 16-week timeframe. There is a **speed-to-deliver** iterative process for release during development. High quality is at no time compromised for speed. Incremental development will minimize the possibility of poor performance or malfunction.

The IPS software team is trained to handle all capabilities at all phases of development. They each have **expertise**, not just knowledge, in each area. Cross training provides flexibility and efficient resource allocation. And the developers benefit from a constant rotation of new products. It provides them with a full breadth of team knowledge.

"Agile is what we do. All of us."

- Bob Wild, *Senior Vice President at IPS*

Developers proficient in both iOS and Android **do exist**. Being proficient at both user interfaces and back-end database schemas **is possible**.

What does all of this mean to the client? We spoke about straight-talk communication and cost efficiency, but there's more to the experience. Customers have one point of contact and transparent collaboration across the entire product lifecycle. Expertise across the entire stack is in one house, under one roof.

As IoT continues to provide extraordinary opportunities for new software, the right approach can make you a frontrunner in your industry. Nascent product development allows organizations to plan software for the future, with the user's future needs in mind. The resulting product is inseparable from the end-user, a goal that any high-quality, high-value software solution must meet.



To learn more contact Danny Aponte at daponte@ips-yes.com or visit our website intelligentproduct.solutions.