MIT MANAGEMENT SLOAN SCHOOL

INTERDISCIPLINARY CONSORTIUM for IMPROVING CRITICAL INFRASTRUCTURE CYBERSECURITY

$(IC)^3$

Striking a balance between usability and cyber-security in IoT devices

Saurabh Dutta

Working Paper CISL# 2017-12

May 2017

Cybersecurity Interdisciplinary Systems Laboratory (CISL)
Sloan School of Management, Room E62-422
Massachusetts Institute of Technology
Cambridge, MA 02142

Striking a balance between usability and cyber-security in IoT Devices

by

Saurabh Dutta
M.S. Architecture and Design, 2009
Mississippi State University

Submitted to the System Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

at the

Massachusetts Institute of Technology
June 2017

Signature of Author: ____

Saurabh Dutta
System Design and Management Program, MIT
May 10, 2017

Certified by: ____

Thesis Supervisor: Professor Stuart Madnick
John Norris Maguire Professor of Information Technologies, MIT Sloan School of Management
Professor of Engineering Systems, MIT School of Engineering

Accepted by: ____

Joan S Rubin
Executive Director, System Design & Management Program, MIT

(This page left intentionally blank)

Striking a balance between usability and cyber-security in IoT Devices

By

Saurabh Dutta

Submitted to the System Design and
Management Program
on May 12, 2017 in Partial fulfillment of the
requirements for the Degree of Master of Science in Engineering and
Management.

ABSTRACT

Today more and more physical objects are being connected to internet. The Internet of Things, or IoT, is dramatically changing the way of living and the way we interact with things and each other. Home doors can be opened remotely with a watch, cars' performance can be upgraded remotely, devices monitor health and send updates to physicians remotely. IoT technology has made some labor-intensive jobs simple and has the potential to simplify and enhance nearly every aspect of our lives. On the other hand, increased levels of high profile cyber security breaches in recent years have made it clear how important it is to make sure these devices are trustworthy and secure. While most users are aware of how critical it is to secure their laptops, mobile devices, and apps, due to the seamless ways in which IoT devices integrates into our daily lives, users are often unaware of risks associated with them.

At the same time, IoT device makers are aggressively releasing new products in a mad race to establish themselves in this emerging market. Increased pressure to differentiate on usability based functionalities has spurred products and features that are not properly vetted for security. Gartner predicts that by 2020, more than 25% of identified enterprise attacks will involve IoT, though IoT will account for only 10% of IT security budgets. As IoT continues to grow, vendors will favor usability over security and IT security practitioners remain unsure of the correct amount of acceptable risk.[1]

While on the other side of spectrum, there are some devices that are very secure but the usability has been compromised to do so. Many product designers and developers must deal with both priorities simultaneously and find them to be frequently conflicting, creating tensions.

This exploratory study introduces a framework that can be used to compare the security impact of design decisions and functionality changes to an IoT system. The main contribution of this study is analyzing existing established usability tools and concepts that is used for quick and dirty evaluation and then come up with similar exploratory tools that can be used to evaluate security and the relationship between security and usability. To understand this relationship better, prioritized list of IoT security attributes are identified by analysis of existing literature and 16 semi-structured interviews which were conducted based on purposive sampling of security experts. Ultimately the aim is to equip non-security person who wants to build an IoT product with an easy way to evaluate their product design decisions with a cyber-security lens. Thus, the results of this study is presented in the form of 2 different application that will allow them to understand underlying factors that they may need to consider to better manage risk/reward trade-offs.

Thesis Supervisor: Stuart Madnick
Title: John Norris Maguire Professor of Information Technologies
MIT Sloan School of Management
Professor of Engineering Systems, MIT School of Engineering

Acknowledgements

First and foremost, I would like to thank my family: my wife Tonima for inspiring me to go after my dreams, and supporting me in my move to Boston to attend the MIT SDM program; my daughter, Thea, for keeping me motivated to complete my studies soon.

Second, I would like to thank my thesis advisor, Professor Madnick, for encouraging me to pursue this theme and allowing me to work with the members of the MIT Interdisciplinary Consortium for Improving Critical Infrastructure Cybersecurity to make it interesting and relevant to both the academic as well as the business communities. Also, to Michael Siegel and other members of the consortium for their candid feedback and for providing me with some of the most practical and valuable advice on how to best approach this research.

To Tod Beardsley and Roy Hodgman for helping me refine my application in the unique and simple to understand manner by providing security expertise and UI development expertise respectively.

To Ger Joyce for his guidance on academic correctness.

To my academic advisor, Pat Hale and Joan S Rubin, for helping me believe in myself and encouraging me to proceed with this work.

Finally, to Blade Kotelly, Jay Brewer and my colleagues at Rapid7 for supporting my decision to attend school while working full time.

(This page left intentionally blank)

Table of Contents

CHAPTER 1: Emergence of IoT and Related Security-Usability Concerns

Introduction

There are many trade-off discussions required to bring products to market in the overall software development lifecycle. Frequently, decisions were made based on factors like time to market, overall budget, and technical feasibility. In most cases, we identified what's the minimum viable product (MVP) which we can launch within the current budget, timeline and technology at hand. Security, when discussed, was considered as part of "technology" discussions and pertained primarily to Quality Assurance tests for Cross site scripting (XSS) and other similar vulnerabilities. With the emergence of IoT (Internet of Things) systems though, cyber-security should get the same level of trade-off attention as time, budget and technical feasibility during product design decisions. In the context of this work, IoT systems refer to interrelated computing devices that have unique identifiers and have the ability to transfer data over network. Here, cyber-security means set of technologies, processes and practices designed to protect networks, computers, programs and data from attack, damage or unauthorized access.

Unlike laptops, mobile phones, software applications, IoT systems are generally too ubiquitous and embedded within a consumer's lifestyle, making it very easy for consumers to overlook the associated risks such as data breach, denial of service, etc. Thus, it falls on product managers to make cyber-security a priority for these devices. For this study, our focus is on the IoT target system, Artificial Pancreas System, and how a product manager of such device can strike a balance between usability and security.

While scholars have talked about the security-usability paradox, there is no easy way to establish a direct relationship between security and usability attributes. There is existing knowledge to explain IoT security concerns; however, if it's not easy to understand and apply by non-security professionals, practical application of this research will remain limited. While IoT devices are used across various domains, this work focuses on medical devices and, in particular, on an artificial pancreas system that diabetic patients use to control their insulin levels. This study further considers a Constructivist lens. This leads to the following:

Research Question

How might product feature prioritization concepts and methods be adopted to optimize the balance between functional usability and cyber-security when designing for IoT devices?

Objectives

1. Analyze Usability-Security paradox in various systems
2. Establish the primary beneficiary of this study
3. Map artificial pancreas system product requirements to 10 usability attributes established by previous studies
4. Establish prioritized list of IoT security attributes
5. Map artificial pancreas system product requirements to these IoT security attributes
6. Create easy to use tool to test usability-security attribute trade-offs for other IoT devices

Contribution

The main contribution of this study is analyzing established usability tools in combination with concepts that are used for quick and dirty evaluation to develop similar exploratory tools that can be used to evaluate security. Ultimately the aim is to equip a non-security person who wants to build an IoT product with an easy way to evaluate their product design decisions with a cyber-security lens. It will allow them to understand underlying factors that they may need to consider to better manage risk/reward trade-offs.

In this study, the process developed is referred as IoT Security Framework.

Initially, the system boundary is defined by conducting a Stakeholder Value Network (SVN) analysis to understand how the said IoT security framework can benefit and affect various stakeholders and identifying the primary beneficiary and persona. Persona was introduced by Alan Cooper as part of goal directed design.[2] It allows the developer to empathize and focus on user's need while they are using an artefact.[3] Product requirements are then analyzed from a functional usability feature perspective. For this study, a consolidated usability model called QUIM (Quality in Use Integrated Measurement)[4] has been used. Five semi-structured interviews were conducted consisting of two diabetic patients, two caregivers and one doctor, all of whom were sampled purposively to better understand functional usability features. Next, IoT security risks are analyzed from a general point of view and using the artificial pancreas system example that contains IoT sensors, communication and storage solutions, processing and presentation of the data, and the related interfaces in between.

Then initial heuristics for IoT security attributes is proposed based on initial risk analysis and existing scholarly literature. To prioritize the security attributes, 16 semi-structured interviews were conducted based on purposive sampling of security experts.

Finally, as practical application tool, a QFD (Quality Functional Diagram) is plotted to establish how functional usability features affect security attributes or not. Another

application, System Security Scale (SSS) is proposed which asks 10 simple questions around system's security and provide consequences and guidance on how to fix.
The target system of this study is envisioned as an artificial pancreas system with connected features, making it an IoT system, that helps diabetic patients manage their insulin levels. It also helps other stakeholders, such as caregivers and doctors, to monitor the patient remotely. Throughout the study, this target system is used as an example.

IoT Technology Evolution in Various Domains

The field of IoT devices is complex and growing. It is predicted to grow to 50 billion devices in next couple of years across various domains. See Figure 1.

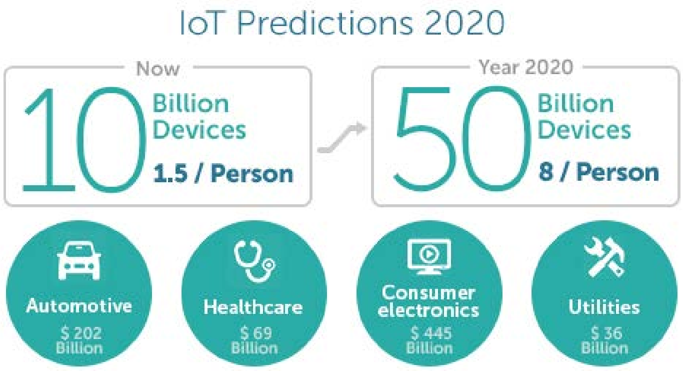


Figure 1 CISCO predicts that by 2020 there will be 8 IoT devices per person on an average across the world

Evolution of IoT Devices – Automotive Examples

Our cars are also leveraging network technologies to make driving safer and more comfortable than ever. "Smart cars" are equipped with automatic parking assist, adaptive cruise control, collision avoidance, and remote emergency response to name a few features. Dependency on these systems is becoming the new norm. Fully-autonomous vehicles are the latest and most advanced system that leverages all the previous technologies and combines them with internet-connected artificial intelligence. As per one study in 2010, world vehicle population topped 1 billion units and is growing.[5] However, driver-less autonomous could substantially reduce the number of vehicles in few decades. One urban designer hypothesizes that, "More than 90% of the time, cars are parked somewhere, taking up space, and costing money and resources. By trading private vehicles for driverless public taxis and shuttles, we could theoretically reduce the number of vehicles by 80% or more and pass the financial and environmental savings onto everyone."[6] There is no doubt that autonomous vehicles are a game changer and, with big companies like Google, Tesla, Uber and others already in road test stage, it may not be very long before these vehicles replace main stream transportation needs.

Evolution of IoT Devices – Consumer Electronics and Home Automation Examples

For human comfort and safety, a variety of mechanical and electronic systems are being used at home: from simple mechanical door locks to more complex mechanical and electronic systems like home appliances. The development of these appliances began in the twentieth-century with the disappearance of full-time domestic servants and the desire to reduce the time-consuming activities in pursuit of more leisure time.

The twenty-first century saw the rise of the “smart home” as home appliances began to leverage network technologies, combining their controls and key functions. For instance, energy distribution could be managed more evenly so that when a washing machine is on, an oven can go into a delayed start mode, or vice-versa. Increasingly, home appliances are being fitted with Internet-connected hardware that allow for remote control, automation, communication with other devices, and enhanced functionality.

Evolution of IoT Devices – Utilities and Public Infrastructure Examples

Public infrastructure is infrastructure for public use. For example, air traffic control is vital for smooth running of aviation systems. These systems include controllers that direct aircraft on the ground and through controlled airspace to prevent collisions and help expedite the flow of air traffic. Primary and secondary radar technology is used to enhance a controller's situational awareness within their assigned airspace — all types of aircraft send back primary echoes of varying sizes to controllers' screens as radar energy is bounced off their exteriors, and transponder-equipped aircraft reply to secondary radar interrogations by giving an ID (Mode A), an altitude (Mode C) and/or a unique call sign (Mode S).

However, with more and more flights in the air, radar based systems need an upgrade. NextGen proposes to transform America’s air traffic control system from a radar-based system with radio communication to a satellite-based one. As one administrator at the Department of Transportation testified before Congress, “Next Gen GPS technology will be used to shorten routes, save time and fuel, reduce traffic delays, increase capacity, and permit controllers to monitor and manage aircraft with greater safety margins.”[7] IoT devices and automated systems may, therefore, reduce the amount of information air crews need to process at any one time, thus potentially making air transportation safer.

Evolution of IoT Devices – Healthcare Examples

Our lives are becoming increasingly symbiotic with machines as we are applying robot technology for the benefit of patients in the healthcare system. Medical devices are being used for diagnosis, prevention, monitoring, treatment or even alleviation of diseases. They are also being used for replacement and modification of the anatomy: IoT devices are becoming a part of us on an anatomical level.

The first implantable device was a pacemaker designed by Dr. Ake Senning in 1958[8]. The limited battery life was a huge drawback. It was until early 1970s when first lithium iodine pacemaker was built with extended battery life of up to five years. As technology evolved, automatic mode switching (automation) was first introduced in early 1990s. It took another two decades to connect them remotely, which made it extremely easy to monitor patient health.

This next phase of the evolution of medical IoT devices will require accurate, repeatable and safe performance of these devices. Like pacemakers, other devices such as insulin pumps, defibrillators, diagnostic machines, operation room monitoring devices and surgical instruments have evolved over years and transmit vital health information from patient’s body to medical care takers.

Gartner's hype circle for emerging technologies predicts maturing technology and market adoption based on their extensive research. As shown in Figure 2 many of these IoT related technologies and applications across domains like IoT platform, smart robots, connected home and autonomous vehicles are just reaching "Peak of inflated expectations" and "Trough of Disillusionment"[9] and IoT developers and vendors do not have a lot of previous learning to lean upon.

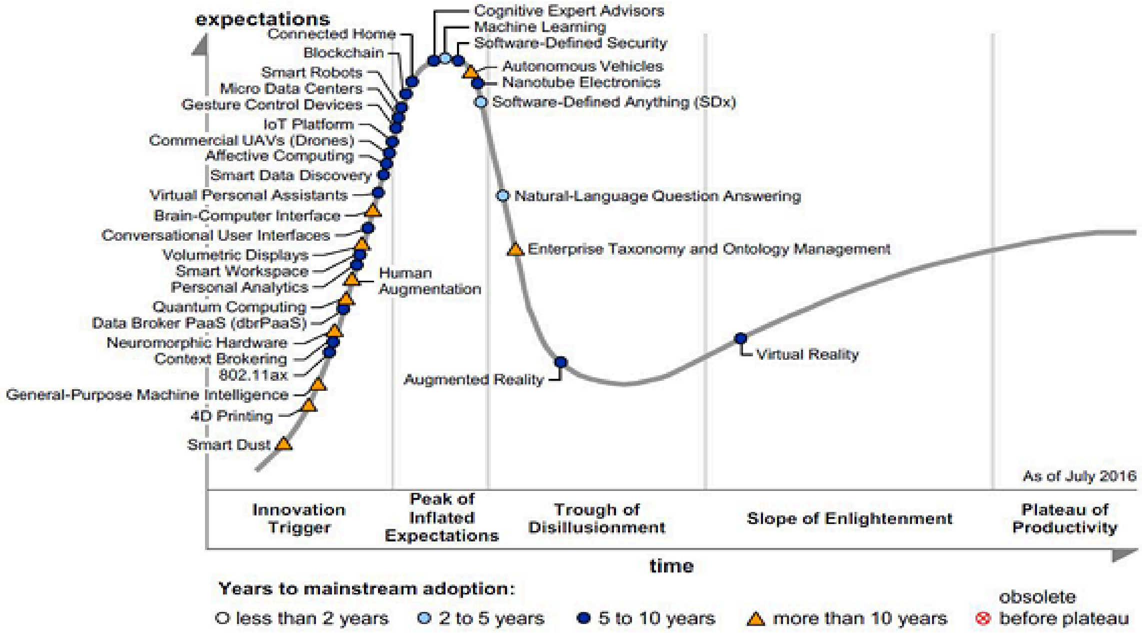


Figure 2 Gartner Hype circle for Emerging technologies, 2016

Many of these IoT device companies are start-ups.[10] All they want is to stay ahead in this race to win the IoT device war. They are neither motivated nor have resources to vet security of these devices. This study aims to fill that gap by providing an easy to follow tool that will motivate users by providing them a prioritized list of security concerns based on their specific product needs and how to reduce them.

IoT Hyper Connectivity and Cyber-Security Risks

With the emergence of cloud computing and IoT technologies, cyber security risks have grown multi-fold. Let’s consider IoT examples from healthcare: insulin pumps and implantable devices. These devices pose huge security and privacy risks as some of them can be remotely controlled. Jay Radcliff, a security researcher interested in the security of medical devices, raised fears about the safety of these devices. He shared his concerns at the Black Hat security conference. Radcliff fears that the devices are vulnerable and has found that a lethal attack is possible against those with insulin pumps and glucose monitors.[11] While these connected devices save lives, the security failure of them can cause the loss of lives as well.

Similarly, the powerful hyper-connectivity of public infrastructures like NextGen air traffic control system ADS-B brings potential vulnerability. Some security researchers have demonstrated how these signals can be spoofed to create fake planes in the sky and create chaos.[12] Nearly all public infrastructures—including rail systems, public transportation, traffic monitoring, utility services, waste and sewage disposal—are in the process of using digital technologies to make them more efficient and responsive to society’s growing needs. However, these same technologies also make them vulnerable to security risks.

In late 2016, Distributed Denial of Service (DDoS) attack against the France-based hosting provider OVH was record breaking. That attack reached over one Terabit per second (1 Tbps) and was carried out via a botnet that infected 150,000 IoT devices.[13] Less than a month later, a massive and sustained Internet attack caused outages and network congestion for many web sites. This attack was launched with the help of hacked IoT devices, such as CCTV video cameras and digital video recorders, and impacted websites from high-profile organizations, including Twitter, Amazon, Tumblr, Reddit, Spotify and Netflix.

Mirai, a malicious botnet, was used to hijack the connected IoT devices. It exploited the default usernames and passwords set by the factory before the devices were shipped to customers. Mirai can launch HTTP floods, as well as various network DDoS attacks, including DNS floods, UDP floods, SYN and ACK floods, GRE IP and GRE ETH floods, and STOMP (Simple Text Oriented Message Protocol) flood.

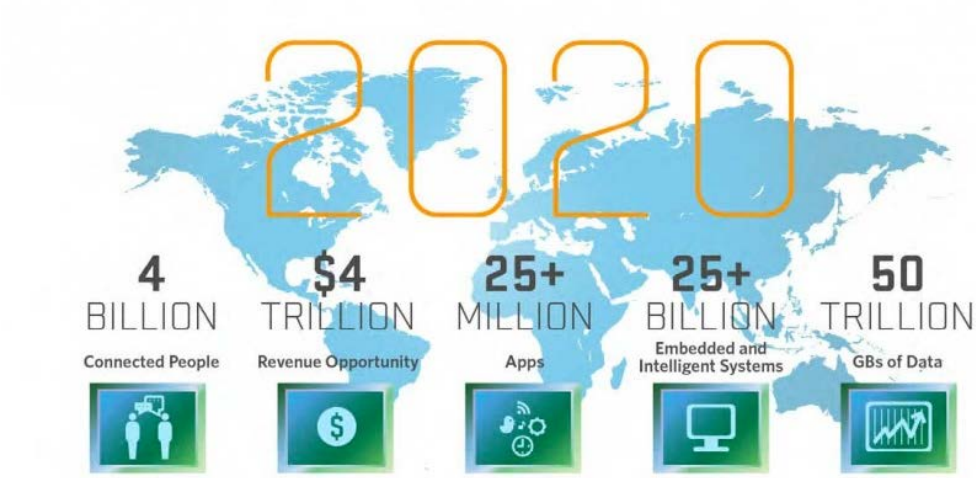
Cloud Computing Success Turning Everything into Internet of Things

In the last few years, there has been a dramatic shift in the evolution of business computing. This evolution has helped to drastically increase employee productivity, and it enables the business to do things they could never do before. Organizations are redefining how they work, using technology. This ranges from communications, collaboration, voice, video, and business intelligence. These advantages also bring new risks and security issues that were previously unimaginable.

McDonalds has used technology to transform their business. Every order and every menu item is optimized in real-time around the world. They can do this because they process all transactions through a centralized analytics engine. 8,600 transactions per second are processed in their commerce platform from all stores around the globe, making every second count.[14] This is something they couldn’t have done before because it would have been out of date using traditional methods. The cost savings, operational efficiencies, and ability to scale their business this fast would have been impossible without the advent of cloud computing.

Just a couple of years ago, any kind of documentation work required parties to be physically present wherever their signatures were needed. However, digital products like Docusign are completely changing the way signatory requirements are accomplished. Moreover, people traditionally collaborated on documents using products like Microsoft Word and emailing revised files. Now, however, companies can have employees jointly editing documents in real time with technology from products like Google Docs or Box. This dramatically reduces editing churn and improves output.

The examples mentioned here won't scale without cloud computing. The cloud has enabled technology evolution to be far more dramatic, effective, and accessible to any company. Previously, these kinds of tools would only be accessible to massive multi-national companies, but now anyone can take advantage of such innovation with a few clicks. Amazon Web Services, by far the largest public cloud provider, is seeing massive adoption. They have millions of active customers across many companies. From huge enterprise IT companies like Cap One, to Commonwealth Bank of Australia, Shell, BP, GE, Schneider Electric, Samsung – every imaginable vertical business is using AWS and similar cloud services now. Figure 3 below shows how unprecedented amount of data will flow between connected devices in next couple of years predicted by a research firm.



Source: Mario Morales, IDC

Figure 3 IDC predicts by 2020 there will be 25+ billion IoT devices transacting 50 trillion GBs of data

Let's consider robots – traditionally, they were built to do a very specific repetitive task effectively that needed robots to have that intelligence inbuilt. For example, a company called iRobot builds robots for household cleaning chores. But now the robots are connected to cloud, turning them into IoT. Previously, robots were custom built to do a single specific task. Now, these robots can have access to all possible artificial intelligence in the cloud, making them far more effective and smarter to do a wide array of complex work that needs creativity to apply multiple concepts together.

This conversion of everyday things into IoT has exploded and the market predictions from various global research companies reflects that.

Summary

Internet of Things has great potential to make huge positive impacts across different domains but with such great power, it’s important to implement it responsibly. If power of IoT goes into wrong hands, it can create havoc. Thus, it’s important to make these systems resilient to any cyber-attacks.

CHAPTER 2: Security Usability Paradox

Figure 4 Uber- A ride hailing app

A few months ago, I was taking an Uber (Figure 4) home from the airport with my wife. We recently bought a house and were looking for some renovation work and discussing a few ideas on the way. The very next day, I received a call from an unknown number—the caller said, “Hello Mr. Dutta, I am [caller’s name] calling from [company name]. I would love to discuss the home renovation project you are planning to undertake in your home.” At this point, his words started blurring as my mind was racing in different directions on how did this guy know all these details? Was it the city that informed them? Was it UBER? The timing was suspicious. That got me thinking more on this.

Security and usability

Everyone loves UBER. It’s quite easy to hail an UBER ride at a tap. But can UBER be a privacy risk? Can we say that we have compromised our privacy for a better experience and usability?

I started digging deep into this question. I looked into CAPTCHA- "a program that protects websites against bots by generating and grading tests that humans can pass but current computer programs cannot. For example, humans can read distorted text as the one shown below, but current computer programs can't."[15]I realized that in websites when security measures like CAPTCHA are added, it makes the website more secure, but the conversion rates for those websites drops significantly as usability is reduced.

Looking at health care systems, in certain types of insulin pumps, a physician has all the vital information, including the patient’s blood glucose level at the moment when they step into the clinic. To enable this, the insulin pump has an always-on Bluetooth sensor. This convenience comes at the cost of high security risk, where it’s possible to tamper with the device remotely with serious consequences.

Finding a balance

Such examples promote a common belief that that security and usability are two antagonistic goals within system design. Simson Garfinkel, in his doctoral thesis at MIT, argued that there are many instances within which security and usability can be synergistically improved.[16] This is possible by revising the way that specific functionality is implemented in many of today’s operating systems and applications. Garfinkel further explains that, in every case considered, it is shown that the perceived antagonism of security and usability can be scaled back or eliminated by revising the underlying designs on which modern systems are conceived. The errors in system design, computer user interfaces, and interaction design can lead to common errors in secure operation.

By identifying and correcting these errors, users can naturally and automatically experience more secure operation. IoT devices can benefit hugely from an established set of design frameworks which are optimized for security operations.

Patterns for better or worse

While such incidents are scary, IoT devices make our lives easier. The potential for IoT is limitless. However, while security is a potential risk, we cannot afford to not seize the opportunity to exploit IoT capabilities to its fullest. What we need is discipline governance or a rule book on how to securely use these products.

Garfinkel refers to these frameworks as simple patterns. Developers and the organizations that employ them must analyze their risks, the cost of proposed security measures, and the anticipated benefits. Be it security or usability, neither should be added to a system as an afterthought. Instead, security and usability must be designed into systems from the beginning. By providing pre-packaged solutions to common design problems, patterns can address this deficit.

A great example of a Usability Pattern is “Copy and Paste” or “Drag and Drop.” These patterns have dramatically changed the usability of computer systems. Similarly, security patterns, such as using the Secure Socket Layer (SSL) to “wrap” clear text protocols and Email-Based Identification and Authentication for resetting passwords, have allowed developers untrained in security to increase the security of their systems. Patterns that align security and usability of IOT devices can create that much-needed rule book for IoT developers.

IoT systems must be viewed as socio-technical systems that depend on the social context in which they are embedded to function correctly. The security mechanisms will only be able to provide the intended protection when people understand and can use them correctly.

It is worth noting that patterns are powerful and can positively or negatively impact a system's security. The IEEE center of secure design has highlighted notable examples of bad IoT security patterns.[17]

Baby Duck Authentication

When a baby animal is born, it assumes that the first animal it sees must be its mother, and it must be an animal of the same type. Internet of things can do something similar, such as accept a connection on a USB at boot time, join a network with a well-known SSID, accept a connection on a well-known URL, socket, or port, or trust any device that connects to a special debugging port (e.g., a JTAG port). Since the device trusts it's mother, convincing the device to do so is usually the first step in undermining its innate security.[18] While this may sound super convenient from usability stand point, it's a very risky pattern.

Secret Handshake

The Secret Handshake does not require any physical manipulation like pressing a special button. While the device is online and working normally, it is always ready to complete a Secret Handshake that indicates membership in the club of privileged users. Secret Handshakes can include specially crafted packets based on timing, ports, IPs, and payloads or holding a power line at a certain voltage for a specific interval. After validating the Secret Handshake, the device might be willing to accept a firmware update, reset to its last known good configuration, enter a special mode (such as sleep or wake), or trigger an unintended action.[18]

All the above examples demonstrate how small usability feature and convenience can come with huge risk if the wrong patterns get established.

EXAMPLE: CAPTCHA Security-Usability

As per one moderated study by distil networks[19], when CAPTCHA was present, people were on average 12% less likely to continue to the content. This number was 27% for mobile devices only. CAPTCHA can be very difficult for users. As per another study CAPTCHAs can be particularly difficult for foreigners. Whether the length of strings used in a scheme is predictable or not can have interesting implications for both its security and usability. The use of color in a CAPTCHA can have an impact on its usability, security, or both. The study showed how characters in google CAPTCHA can be confusing as shown in Table 1. Also, Table 2 shows distortion, content and presentation issues with CAPTCHA.[20]

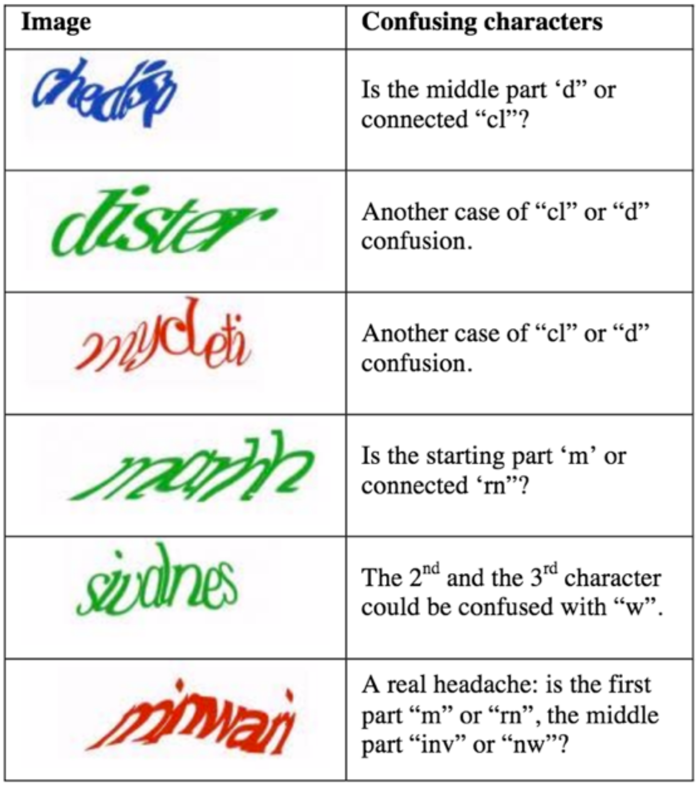
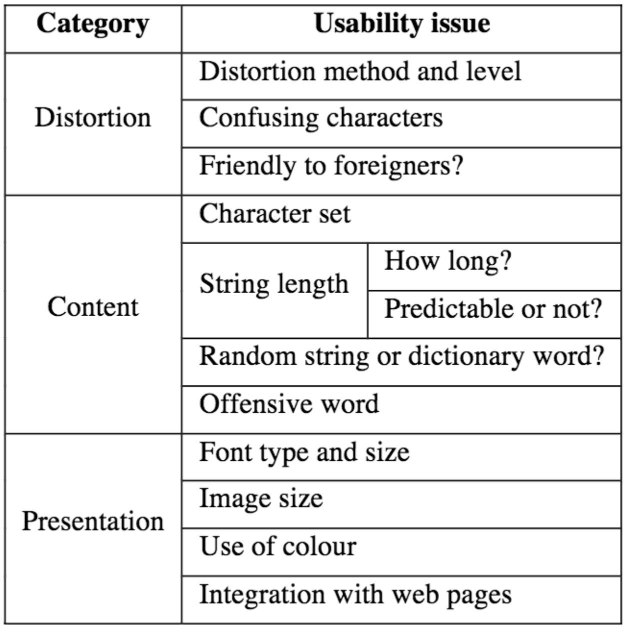


Image	Confusing characters
	Is the middle part ‘d” or connected “cl”?
	Another case of “cl” or “d” confusion.
	Another case of “cl” or “d” confusion.
	Is the starting part ‘m’ or connected ‘rn”?
	The 2nd and the 3rd character could be confused with “w”.
	A real headache: is the first part “m” or “rn”, the middle part “inv” or “nw”?

Table 1 Confusing characters in the Google CAPTCHA



Category	Usability issue	
Distortion	Distortion method and level	
	Confusing characters	
	Friendly to foreigners?	
Content	Character set	
	String length	How long?
		Predictable or not?
	Random string or dictionary word?	
	Offensive word	
Presentation	Font type and size	
	Image size	
	Use of colour	
	Integration with web pages	

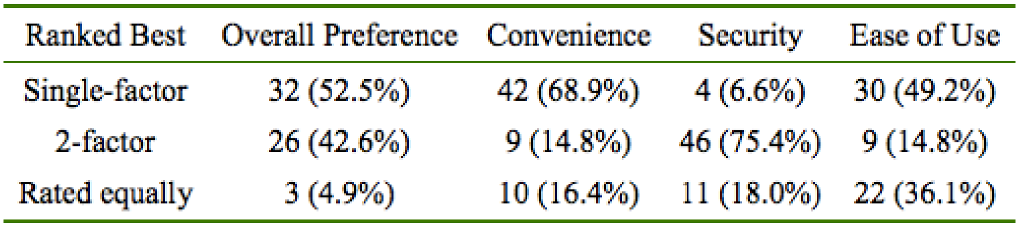
Table 2 Usability issues with CAPTCHAs

EXAMPLE: Two-Factor Authentication

Two-factor authentication is a security solution requiring the verification of two different modalities of authentication components. Typical components include: knowledge (e.g., passwords), possession (e.g., bankcard) and physical attributes (e.g., fingerprint). Two-factor solutions provide enhanced security by combining more than one authentication type, such that if a customer's password is compromised, the second factor will provide an extra barrier to fraudulent entry.[21]

In a controlled experiment with 61 banking customers, a knowledge-based, single-factor authentication procedure based on practices commonly used in the financial services industry was compared with a two-factor approach where, in addition to the knowledge-based step, a one-time passcode was generated using a hardware security token.

This experiment investigated user perceptions of the usability and security of single-factor and two-factor authentication methods which showed how usability gets degraded with better security in the automated telephone banking sector.[22] While the results were derived from multiple factors, Table 3 shows preference rank from those 61 users as an example. It is very clear that users preferred the ease of use of single factor authentication but also wanted the security provided by 2 factor authentications.



Ranked Best	Overall Preference	Convenience	Security	Ease of Use
Single-factor	32 (52.5%)	42 (68.9%)	4 (6.6%)	30 (49.2%)
2-factor	26 (42.6%)	9 (14.8%)	46 (75.4%)	9 (14.8%)
Rated equally	3 (4.9%)	10 (16.4%)	11 (18.0%)	22 (36.1%)

Table 3 Preference rank for the 2 Factor Authentication single factor authentication.[22]

EXAMPLE: Bluetooth Security Usability

Bluetooth technology is an industry-standard to connect devices in proximity. While it enhances the user experience due to its wireless portability, many researchers have identified security loopholes. For example, spoofing through keys or a "Man in the middle attack" is how the identification and encryption keys are stolen before the start of a session which can be then used to impersonate and communicate. Another type is spoofing through Bluetooth address. Each Bluetooth device has its unique ID. While a user thinks, they are connecting with a trusted device, an intruding device can change its address to match a trusted device address and get unauthorized access.[23]

Summary

At first glance, it will always look like security and usability are antagonistic, but by setting good practices, patterns and principles, security and usability can be improved synergistically. We will look into this further in our study ahead.

CHAPTER 3: Defining System Boundary and Stakeholders

For ease of explanation, in this chapter "IoT Security Framework" represents the proposed applications that is derived from this study. The proposed IoT framework can benefit or affect a lot of different stakeholders. In this chapter, system's approach is used to analyze relationship. Typically, it starts with identifying stakeholders and beneficiaries. Then needs of those stakeholders are characterized and interpreted into set of prioritized goals to establish metrics.

Stakeholders and Beneficiaries Needs

A stakeholder in an organization (corporation, government, project, etc.) is "any group or individual who can affect or is affected by the achievement of the organization's objectives"[24]. Beneficiaries are those who benefit from the proposed IoT framework. The outcome directly or indirectly addresses their needs. Beneficial stakeholders receive and give while charitable beneficiaries receive benefit, but do not provide resources to the project. Problem stakeholders give resources to the project, but get little or no direct benefit. In this scenario, the "project" is IoT security framework.

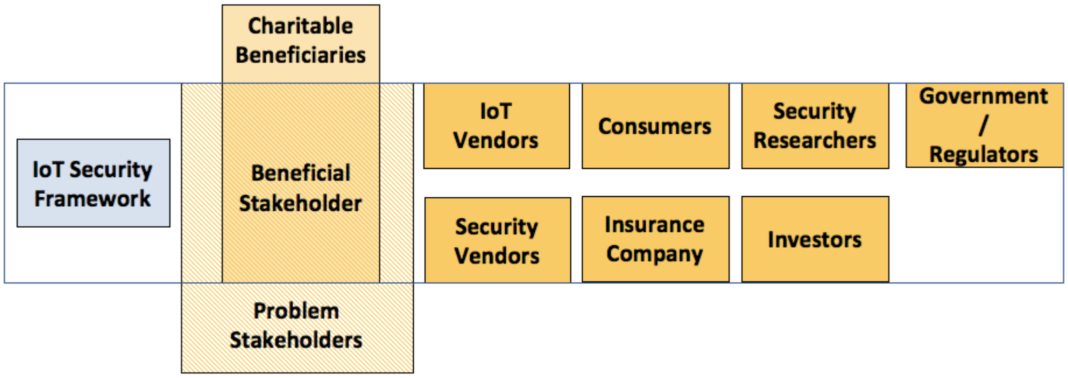


Figure 5 shows qualitative assessment of stakeholders and beneficiaries for the proposed IoT security framework

To make the framework really work and improve IoT security, the thinking is expanded and one can see that all the beneficiaries become stakeholders. Figure 5 shows project deliverables need endorsement from IoT vendors to have a meaningful impact so they are the project's beneficial stakeholder. Also, endorsement from consumers, security researchers and regulators etc. will indirectly help IoT vendor to endorse this framework. Thus, they become beneficial stakeholders too. No problem stakeholders were identified.

Since, our goal is to define a IoT security framework, it is our focal organization for this study. Figure 5 identified that there are at least 7 potential beneficial stakeholders but this needs to be narrowed down further to conduct a focused study. For example, this study can further explore how a consumer or customer of a IoT device can benefit from this IoT Security Framework to make buying decision or it can evaluate how IoT vendors can use it to improve their product security posture. So, a need analysis is done to identify which stakeholder may be benefited most. Need is a product attribute that exists in the mind of beneficiary. It is the overall desire or want or a wish for something which is lacking. It also includes opportunities to fill unexpressed or unrecognized needs. The Figure 6 shows the identified primary and secondary benefit of this framework to various beneficiaries. The primary beneficiaries are IoT device end customer, the vendor (including product managers, developers of the device), government, Investors, security vendors and Insurance company. Clearly IoT vendors seems to get the most value from the said IoT security framework. These are explained in further details in next section.

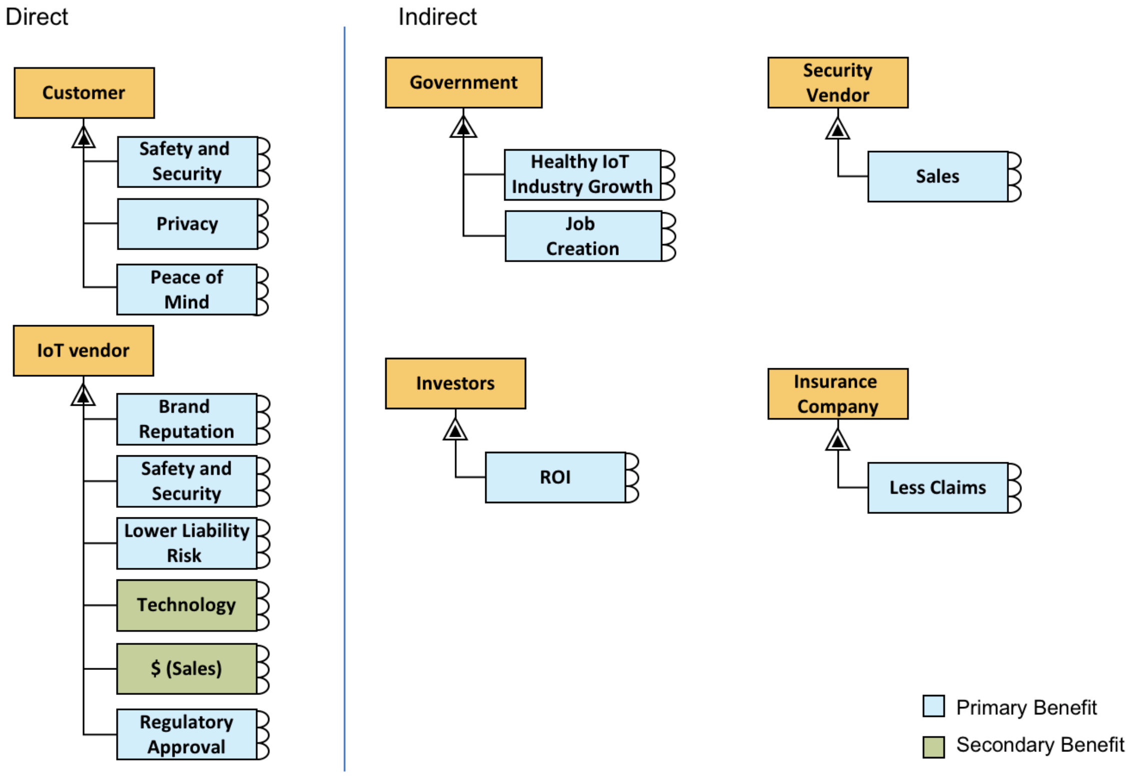


Figure 6 Beneficiaries and their needs

Stakeholder Value Network

A stakeholder value network (SVN) is a multi-relational network consisting of a focal organization, the focal organization's stakeholders, and the tangible and intangible value exchanges between the focal organization and its stakeholders, as well as between the stakeholders themselves.[25] We can analyze the intensity of benefit based on **Kano Analysis**[26] which prioritize the needs into “must have”, “should have” and “Might have” to create **benefit ranking**. Based on benefit ranking, we can make trade-off decisions on product design and development.

In Figure 7, SVN analysis is conducted to identify the benefits and needs shared between previously identified beneficiaries. Since, one of the primary objective of this study is to create easy to use tool to test usability-security attribute trade-offs for other IoT devices, the IoT Security Framework is identified as the focal point. Then, the various stakeholders are connected to the focal organization color coded by intensity of benefits.

"Must Have" relationships are identified between IoT device maker and proposed IoT security framework which deals with transfer of knowledge primarily while the other must have relationship is between IoT Device makers and Consumers which deals exchange of value (Money against Device or application).

"Should Have" needs are established between the following: IoT Security Framework needs security vendors for knowledge transfer. Security vendors also helps IoT device makers to make their product secure in exchange of money. IoT device makers also needs to produce value and provide return on investment to their investors who paid device makers money in 1st place to set up business. Certain type of consumers of these IoT devices like medical devices may have insurance in exchange of insurance premium.

"**Might Have"** needs are established between the following: IoT device makers needs to maintain compliance that is set by regulatory bodies. These regulatory bodies are managed by government which also provides workforce for law enforcement. Cyber-criminal needs to be put to justice by law enforcement. This is though quite far from the focal organization and is thus beyond system boundary.

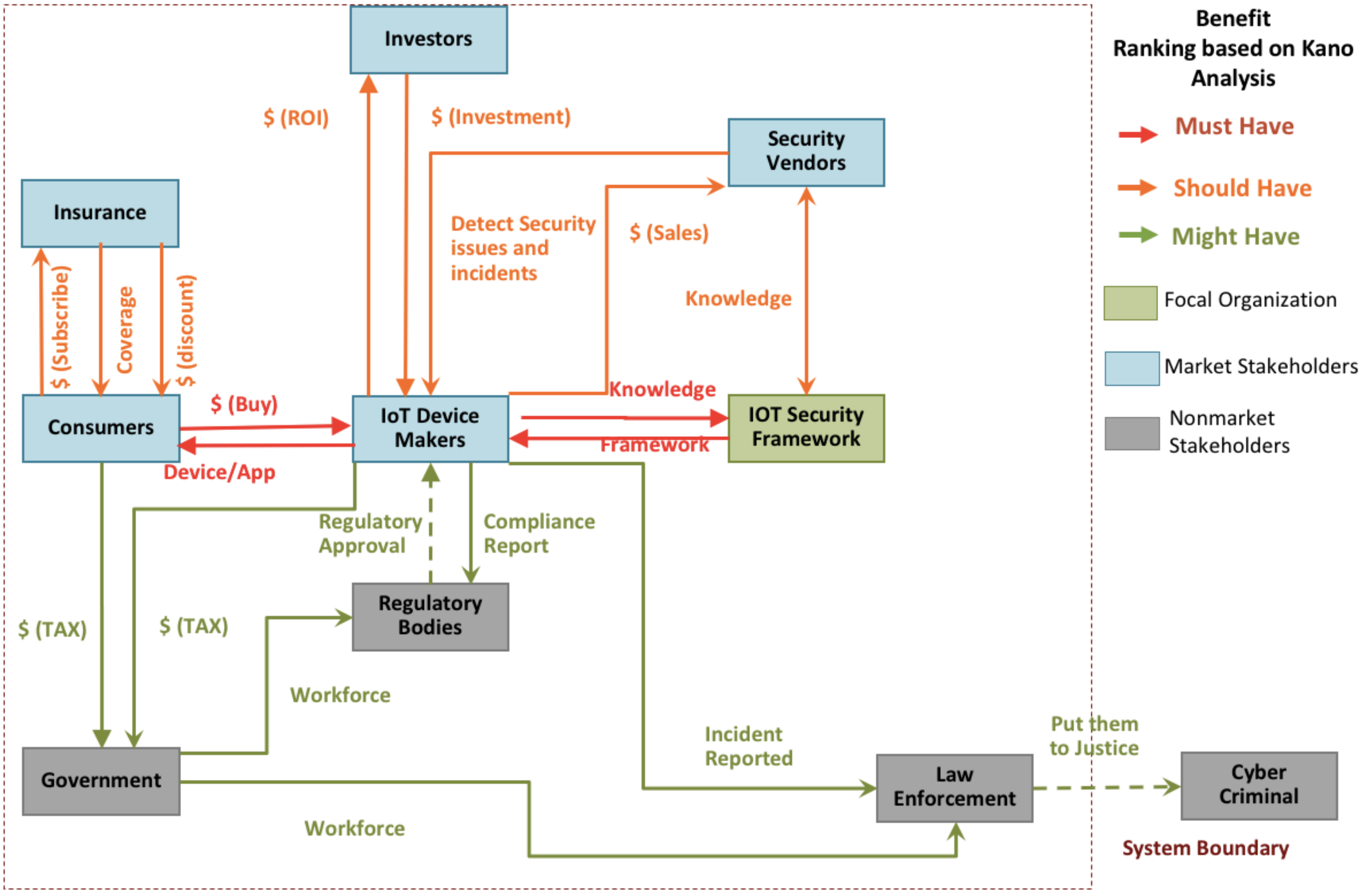


Figure 7 SVN Analysis of IoT Security Framework which shows benefit ranking

Stakeholder Value Network- Artificial Pancreas System

The heuristics for identifying the product requirement involved 5 semi-structured interviews that were conducted consisting of 2 diabetic patients, 2 caregivers and 1 doctor, all of whom were purposively sampled to better understand functional usability features. The other data point was class lecture and notes from Morales [27] who researched extensively around artificial pancreas system needs while working on such devices at his previous employer.

Figure 8 shows the primary stakeholders for an artificial pancreas system and Figure 9 shows the benefit for the direct stakeholders.

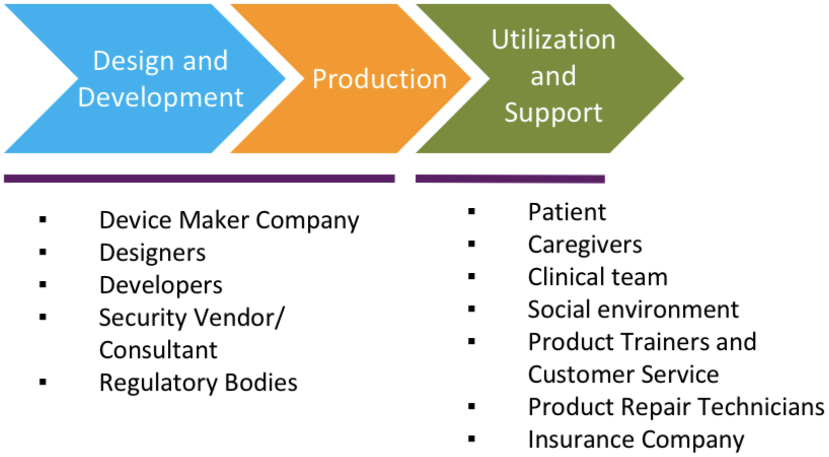


Figure 8 Various identified stakeholders during lifecycle of artificial pancreas system

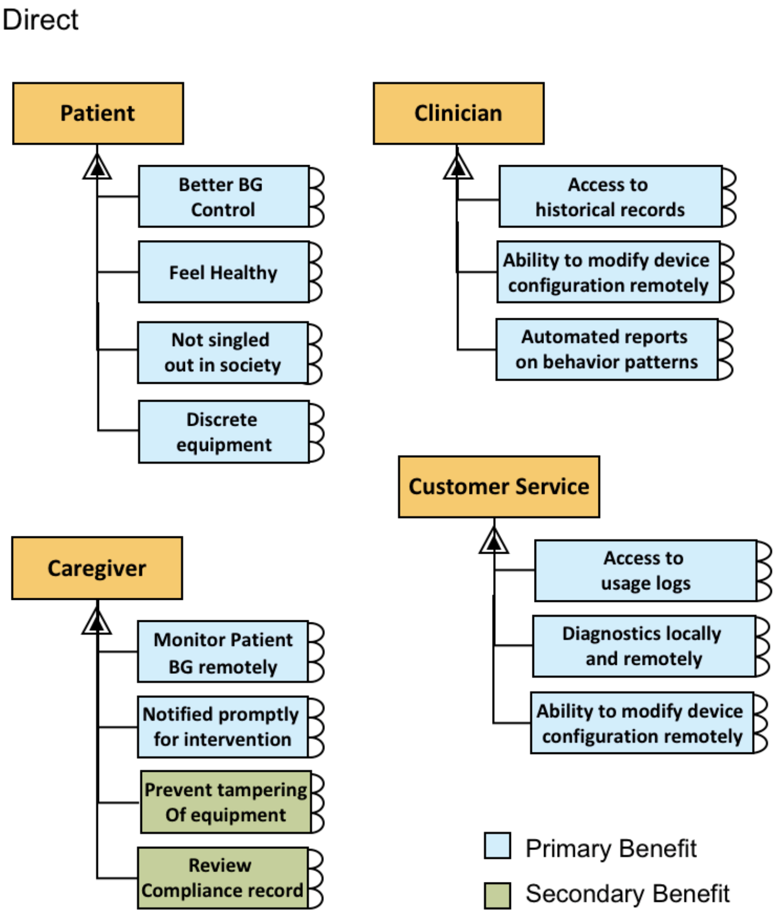

Direct
Patient
Better BG Control
Feel Healthy
Not singled out in society
Discrete equipment
Clinician
Access to historical records
Ability to modify device configuration remotely
Automated reports on behavior patterns
Customer Service
Access to usage logs
Diagnostics locally and remotely
Ability to modify device configuration remotely
Caregiver
Monitor Patient BG remotely
Notified promptly for intervention
Prevent tampering Of equipment
Review Compliance record
Primary Benefit
Secondary Benefit

Figure 9 Primary direct beneficiaries of artificial pancreas system and their needs

Once the stakeholders and beneficiaries are defined, the SVN analysis is done by keeping IoT security Framework as the focal organization but changing the context from a general IoT device to a more specific artificial pancreas system. Based on the principles of kano analysis and discussion with artificial pancreas system specialist Mr. Carlos Morales [27] from Johnson and Johnson, SVN analysis in figure 10 shows the following:

"Must Have" relationship between patient, device maker, care giver and clinical team at one side while another strong relationship is identified between the proposed IoT security framework and the device maker. Patient pays device maker, clinical team, care giver, insurance company and customer service in exchange of the actual device, medical services, health monitoring, coverage and services respectively from those stakeholders.

"Should Have" relationships is established between IoT security framework and Security vendors through knowledge transfer. Also, Investor invest on Artificial pancreas maker in exchange of return on investment (ROI). Security vendor provides security tools to artificial pancreas maker in exchange of money to secure the pancreas system. Patient is in should have relationship with clinical team, customer service to get services against payment. Also, patient gets medical insurance coverage by insurance body against subscription payment.

"Might Have" relationship is established between artificial pancreas maker and regulatory bodies where the former might have to provide compliance report to regulatory bodies to get regulatory approvals.

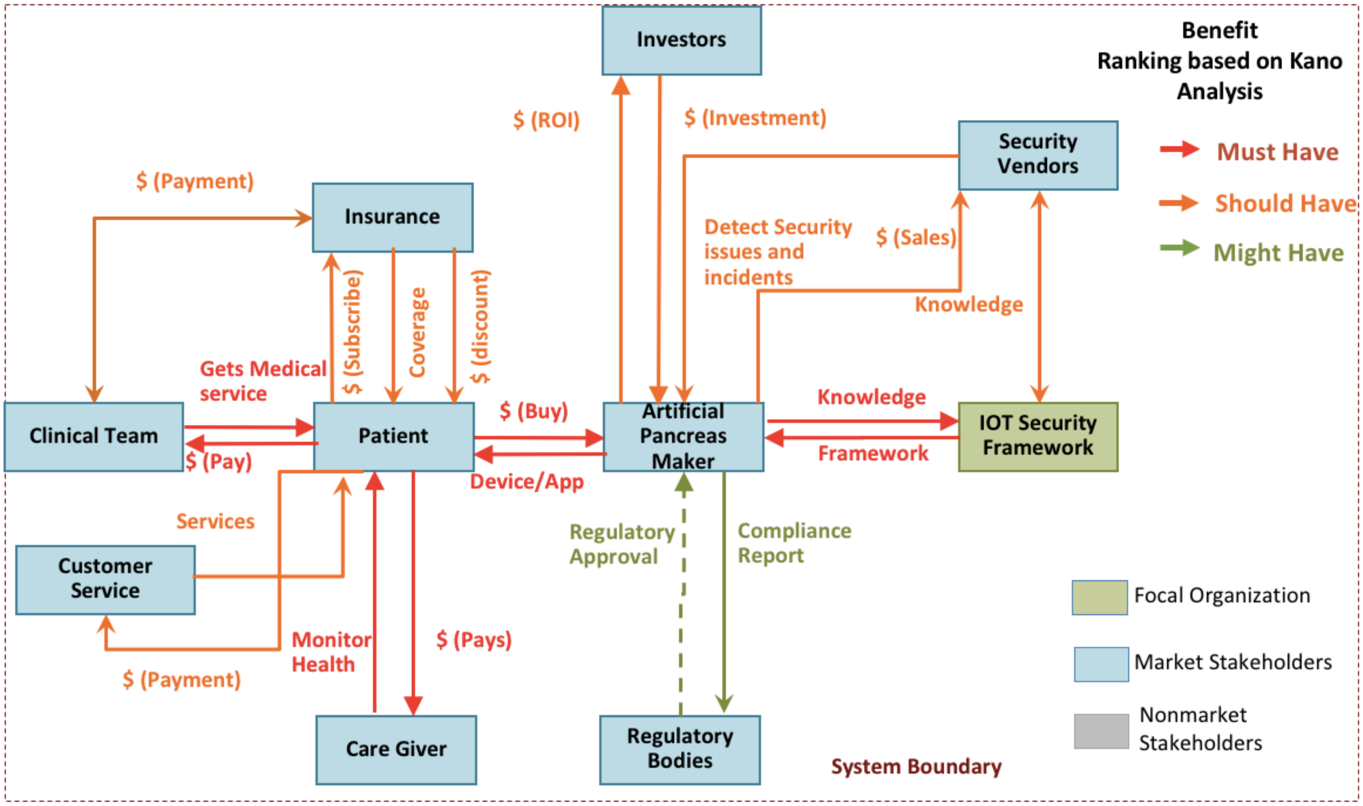


Figure 10 SVN Analysis of IoT Security Framework for artificial pancreas system showing benefit ranking

Findings

Although multiple stakeholders can benefit from the proposed IoT security framework, IoT device maker is one of the primary stakeholder by benefit ranking priority. The other stakeholder with equally high benefit ranking is consumers. As the thought process was expanded, security vendors, Insurance companies, law enforcement, government and regulatory bodies also became direct or indirect beneficiaries. Although not within system boundary, it is worth noting that this IoT security framework ultimately can affect Cyber criminals.

Detailed artificial pancreas SVN analysis showed similar results with Device maker being the primary stakeholder along with the consumers where the consumers are the patient and his/her care giver and clinical team.

Stakeholder in Focus

This study primarily focuses on IoT device maker for all further analysis going forward. As part of future work, this study briefly mentions how end users of IoT devices can benefit from this IoT framework but further analysis is not in scope.

Persona and Goals

Here is a hypothetical persona of a typical user who this study assumes may be using this IoT security framework

Role: Product manager/ Developer of IoT device maker

Goal: Need to ship new features and make product successful commercially.

Problem: Needs to understand the security implications to feature requests before implementation.

Solution: IoT framework makes John aware of potential security issues and better protect the company from releasing products that introduce unanticipated risks to their customers. It gives the persona a leg to stand on when negotiating feature requests from customers, developers, and management.

How: IoT security framework can be used to compare various designs to come up with the optimized option which does justice to both functionality and security. Ideally it will help make faster and more informed trade-off decisions.

CHAPTER 4: Functional Usability Features and Product Requirements

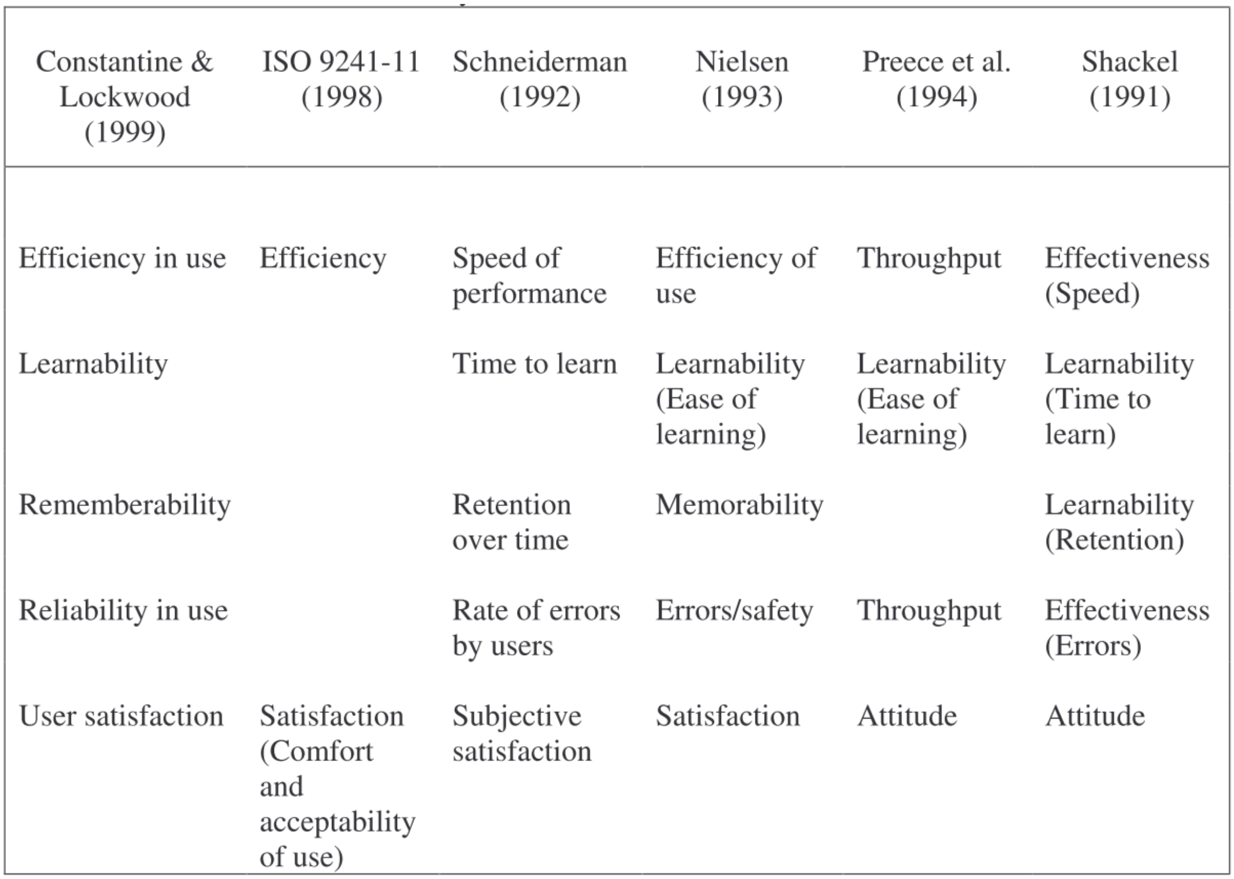
Usability as Functional Product Requirement

At a high level, any product has two types of requirements: Functional requirements specify what the system should do. Non-Functional requirements specify how the system works or how the system should behave. Usability is degree of ability of anything to be used. Generally, usability is a non-functional requirement but sometimes certain usability enhancement itself can be a new functionality. Usability features with major implications for product functionality are incorporated as functional requirements and are termed as **functional usability features**.[28] Recent studies have targeted the relationship between usability and functional requirements. Cysneiros et al. suggest identifying functional requirements that improve certain usability attributes [29]. Following such catalogue to achieve usability goals might even help to disclose new functional requirements.

General Usability Requirements

There is a lot of usability literature and frameworks which provides recommendations on how to build usable products. As an example, J. Nielsen talks about them as design heuristics[30], others call them principles of usability[31], usability guidelines[32] etc. as shown in Table 4.
Even various standards have different definitions for usability:

- “A set of attributes that bear on the effort needed for use and on the individual assessment of such use, by a stated or implied set of users”33
- “The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use”34
- “The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component”[35]



Constantine & Lockwood (1999)	ISO 9241-11 (1998)	Schneiderman (1992)	Nielsen (1993)	Preece et al. (1994)	Shackel (1991)
Efficiency in use	Efficiency	Speed of performance	Efficiency of use	Throughput	Effectiveness (Speed)
Learnability		Time to learn	Learnability (Ease of learning)	Learnability (Ease of learning)	Learnability (Time to learn)
Rememberability		Retention over time	Memorability		Learnability (Retention)
Reliability in use		Rate of errors by users	Errors/safety	Throughput	Effectiveness (Errors)
User satisfaction	Satisfaction (Comfort and acceptability of use)	Subjective satisfaction	Satisfaction	Attitude	Attitude

Table 4 Usability attributes of various standards or models[36]

In more recent times many researchers have tried to consolidate various models to create a consolidated model of measuring usability. For this study, one such model called Quality in Use Integrated Measurement (QUIM) is used. It consists of 10 usability factors that are decomposed into 26 sub factors.

QUIM 10 Usability Attributes

Here is the list of 10 usability attributes that is defined by QUIM in the consolidated model as per Seffah et al.[4]

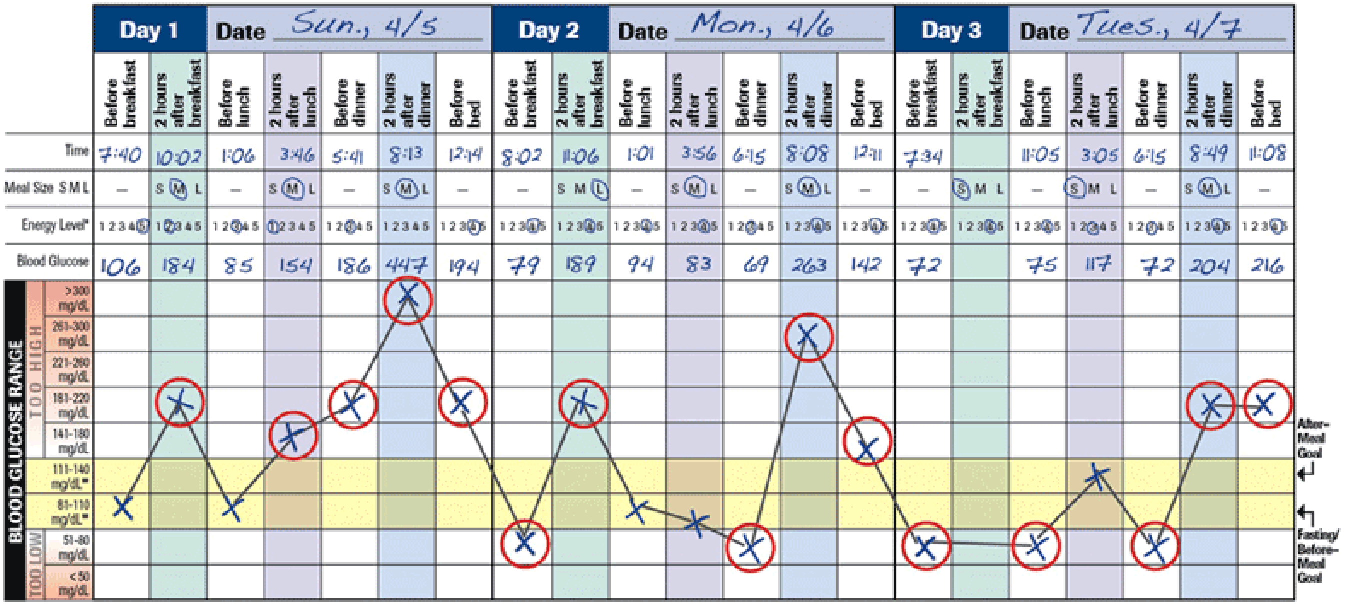
> 1. **Efficiency**, or the capability of the software product to enable users to expend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use.
>
> 2. **Effectiveness**, or the capability of the software product to enable users to achieve specified tasks with accuracy and completeness.
>
> 3. **Productivity**, which is the level of effectiveness achieved in relation to the resources (i.e. time to complete tasks, user efforts, materials or financial cost of usage) consumed by the users and the system. In contrast with efficiency, productivity concerns the amount of useful output that is obtained from user interaction with the software product…
>
> 4. **Satisfaction**, which refers to the subjective responses from users about their feelings when using the software (i.e., is the user satisfied or happy with the system)
>
> 5. **Learnability**, or the ease with which the features required for achieving particular goals can be mastered. It is the capability of the software product to enable users to feel that they can productively use the software product right away and then quickly learn other new (for them) functionalities.
>
> 6. **Safety**, which concerns whether a software product limits the risk of harm to people or other resources, such as hardware or stored information. It is stated in the ISO/IEC 9126-4 (2001) standard that there are two aspects of software product safety, operational safety and contingency safety.
>
> 7. **Trustfulness**, or the of faithfulness a software product offers to its users…
>
> 8. **Accessibility**, or the capability of a software product to be used by persons with some type of disability (e.g., visual, hearing, psychomotor) …
>
> 9. **Universality**, which concerns whether a software product accommodates a diversity of users with different cultural backgrounds (e.g., local culture is considered).
>
> 10. **Usefulness**, or whether a software product enables users to solve real problems in an acceptable way. Usefulness implies that a software product has practical utility, which in part reflects how closely the product supports the user's own task model. Usefulness obviously depends on the features and functionality offered by the software product. It also reflects the knowledge and skill level of the users while performing some task (i.e., not just the software product is considered).

Functional Usability Feature Requirements- Artificial Pancreas System

Like previous chapter, to demonstrate functional usability feature requirement gathering, an internet connected artificial pancreas system will be used. Following 3 sources are used to gather the requirements:

Firstly, Morales[27] in his guest lecture at MIT SDM program provided information on his requirement gathering methodology for artificial pancreas system at Johnson and

Johnson company. The materials he provided includes his research on day of a patient with diabetes and his caregiver. Figure 11 shows a sample schedule of a diabetic patient illustrating how within a day blood glucose levels of a patient can go up and down multiple times based on food intake.



	Day 1		Date Sun., 4/5					Day 2		Date Mon., 4/6					Day 3		Date Tues., 4/7				
	Before breakfast	2 hours after breakfast	Before lunch	2 hours after lunch	Before dinner	2 hours after dinner	Before bed	Before breakfast	2 hours after breakfast	Before lunch	2 hours after lunch	Before dinner	2 hours after dinner	Before bed	Before breakfast	2 hours after breakfast	Before lunch	2 hours after lunch	Before dinner	2 hours after dinner	Before bed
Time	7:40	10:02	1:06	3:46	5:41	8:13	12:14	8:02	11:06	1:01	3:56	6:15	8:08	12:11	7:34		11:05	3:05	6:15	8:49	11:08
Meal Size S M L	–	S [✓ M] L	–	S [✓ M] L	–	S [✓ M] L	–	–	S M [✓ L]	–	S [✓ M] L	–	S [✓ M] L	–	–	[✓ S] M L	–	[✓ S] M L	–	S [✓ M] L	–
Energy Level*	1 2 3 4 [✓ 5]	1 [✓ 2] 3 4 5	1 2 [✓ 3] 4 5	[✓ 1] 2 3 4 5	1 2 [✓ 3] 4 5	1 2 3 4 5	1 2 3 [✓ 4] 5	1 2 3 [✓ 4] 5	1 2 3 [✓ 4] 5	1 2 3 [✓ 4] 5	1 2 3 [✓ 4] 5	1 2 [✓ 3] 4 5	1 2 3 [✓ 4] 5	1 2 3 [✓ 4] 5	1 2 3 [✓ 4] 5	1 2 3 [✓ 4] 5	1 2 3 [✓ 4] 5	1 2 [✓ 3] 4 5	1 2 [✓ 3] 4 5	1 2 3 [✓ 4] 5	1 2 3 [✓ 4] 5
Blood Glucose	106	184	85	154	186	447	194	79	189	94	83	69	263	142	72		75	117	72	204	216

Figure 11 Schedule of a diabetic patient (Carlos O. Morales)

Secondly, SVN analysis of Artificial Pancreas system in chapter 2 helped to identify several others like clinician, customer service, regulatory bodies, doctors etc. as stakeholders and their needs.

Thirdly, 5 semi-structured interviews were conducted consisting of 2 diabetic patients, 2 caregivers and 1 doctor, all of whom were purposively sampled. Both of those patients used a model of artificial pancreas system that were not connected to internet (Figure 12). Due to time constraints and privacy concerns, finding further interview candidates was challenging. Although this sample size seems low, there is no consensus between experts on the number of participants for a qualitative research study. It could be argued that the number of participants within this study was close, or equal to, that recommended by several such experts. Kuzel[37] for instance, suggests that six interviews could address research questions adequately should the sample be homogeneous, as is the case of this study. Further, Romney, Batchelder, and Weller[38] assert that four individuals are enough should those individuals be highly knowledgeable within the domain in question. As the participants were purposively sampled[39] based on their knowledge of artificial pancreas systems, this is likely. To that end, it is anticipated that the findings from this study are both reliable and valid, at least from a qualitative perspective. The last point is an important one as reliability and validity are synonymous with quantitative research. In this context, reliability and validity refer respectively to the transferability of learnings and trustworthiness of participants[40].

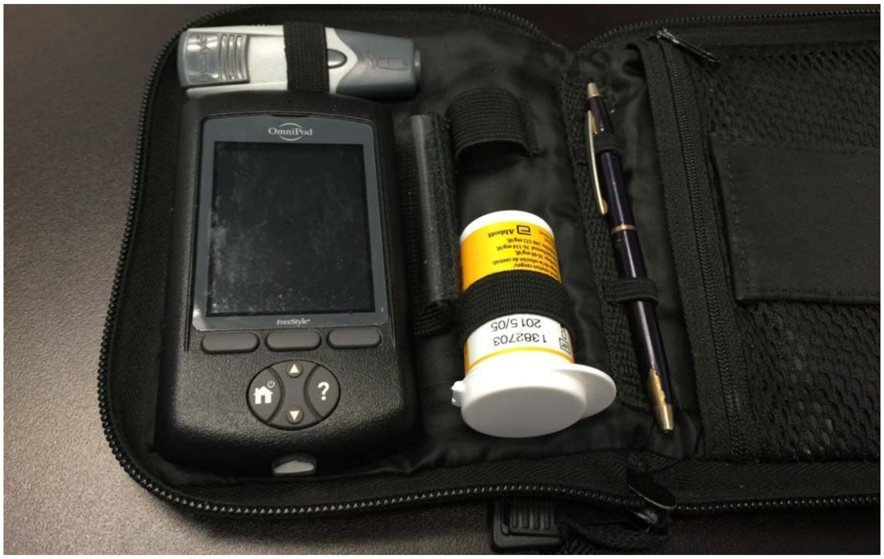
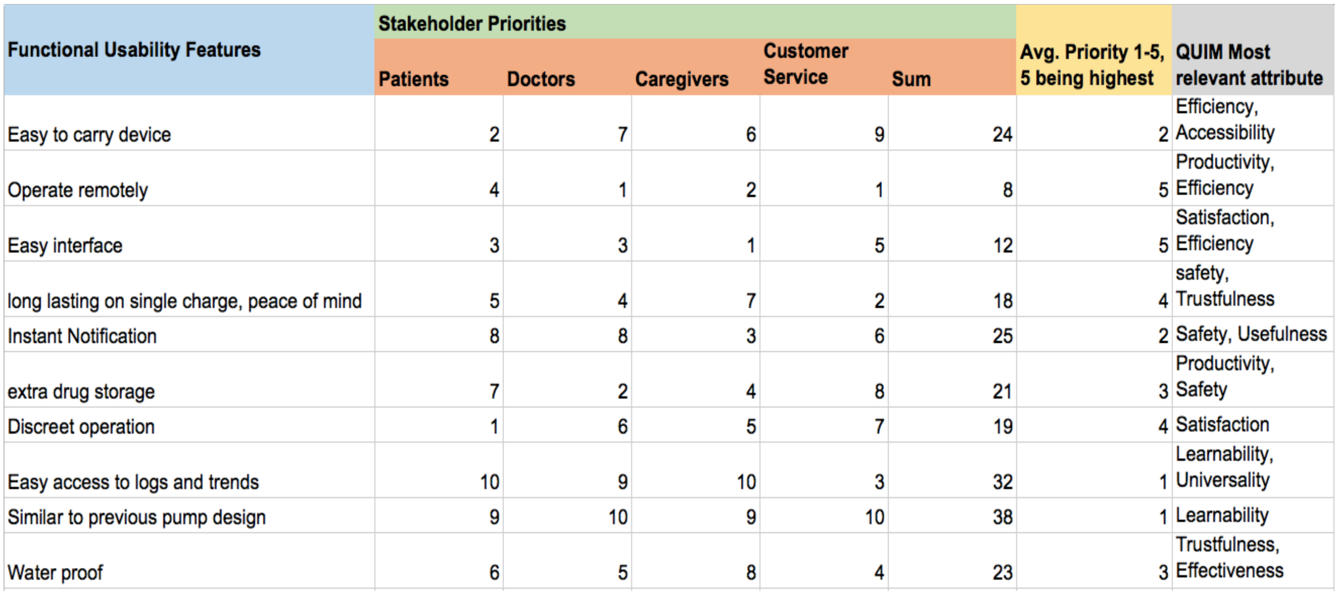


Figure 12 Current device used by one of the diabetes patient interviewed for this study

Here is list of stakeholders' top functional usability feature requirements from a proposed connected artificial pancreas system based on the research mentioned above. As shown in table 5, once the functional usability features were identified, the research participants were asked to rank features by importance between 1-10 scale where 1 is most important. In the sum column, rankings from each participant is added.

Now to determine the Average Priority, the scale used is 1-5 where 5 is the highest priority. The higher the "Sum" column is, lower the average priority rating is.



Functional Usability Features	Stakeholder Priorities					Avg. Priority 1-5, 5 being highest	QUIM Most relevant attribute
	Patients	Doctors	Caregivers	Customer Service	Sum		
Easy to carry device	2	7	6	9	24	2	Efficiency, Accessibility
Operate remotely	4	1	2	1	8	5	Productivity, Efficiency
Easy interface	3	3	1	5	12	5	Satisfaction, Efficiency
long lasting on single charge, peace of mind	5	4	7	2	18	4	safety, Trustfulness
Instant Notification	8	8	3	6	25	2	Safety, Usefulness
extra drug storage	7	2	4	8	21	3	Productivity, Safety
Discreet operation	1	6	5	7	19	4	Satisfaction
Easy access to logs and trends	10	9	10	3	32	1	Learnability, Universality
Similar to previous pump design	9	10	9	10	38	1	Learnability
Water proof	6	5	8	4	23	3	Trustfulness, Effectiveness

Table 5 Top Functional usability feature requirements for Artificial pancreas system mapped to QUIM most relevant usability attributes

Summary

Previous usability literature pointed that sometimes non-functional features like usability turns into functional usability features for some systems that has major usability implications. It is analyzed how there are many usability principles and how QUIM has consolidated them into 10 specific attributes for the sake of generalization. Then the information from class lectures, SVN analysis from chapter 2 and the interview data is used to create list of top usability requirements for a connected artificial pancreas system as an example of requirement gathering. A prioritized list of product requirements is created for proposed connected artificial pancreas. The result of this general usability attributes and specific artificial pancreas system requirements will be used in next chapters.

CHAPTER 5: IoT Security Analysis

“Art is never finished, only abandoned”—Leonardo da Vinci

One can always go back and touch up a painting, rewrite a lyric or melody. Art is never done. It is never complete. It can always be refined. As User Experience designers, we iterate on our product design, collect feedback, iterate again. When we achieve, something called MVP (Minimum Viable Product) or MDP (Minimum Desirable Product), the product is released for users to use. Then we again get back to loop- collect feedback, iterate design.

In Cyber security, this concept is even more relevant. “It is sometimes depressing to see how I remediated 25% of all vulnerabilities in our network over last 30 days and still my metric does not reflect that because during the same period of last 30 days, another 21% of total volume of new vulnerabilities are discovered in our network” said one of the security expert interviewed who runs the vulnerability management program for a mid-size company. He further added “You can never get rid of all security risks in your network. Instead you need to prioritize top security risks and mitigate them”

The aim for this study is to do just that. What is the top 10 things, one must consider when designing and developing an IoT product?
The study heuristics involved deep dive into relevant literature study and based on various security concerns discussed in various literature- a list of 15 primary IoT security attributes is complied. Then a security threat analysis for the identified target system- Artificial Pancreas system is conducted.

Next, 16 security experts were interviewed to stack rank those security attributes and came up Top 10 security issues that covers all attributes. We will talk about relevant application in chapter 5.

IoT Security Concerns

Security concerns for IoT devices are relatively a recent phenomenon but it still has a fair amount of relevant existing scholarly literature. Since one of the purpose of this study is to prioritize most concerning aspects of IoT security, a comparative study of literature study is created as shown in Table 6 depicting common security concerns and principles discussed.

Braz et al. (2007)[41]	**Babar et al. (2010)**[42]	**Alam et al. (2011)**[43]	**Rihai et al. (2013)**[44]
Confidentiality	Confidentiality/ Secure storage	Confidentiality	Privacy
Integrity	Integrity/ Secure storage	Integrity	Safety
Availability	Availability	Availability	Reliability
Authentication	Data Authentication	Authentication	
Authentication	Identity management	Authorization	Identification
Access Control	Access Control	Access Control	Responsibility
Trustfulness	Secure network access, content and execution environment	Trustworthiness	Trust

Table 6 A comparative study of literature identifying common security concerns

This study also referred to IoT security guidance published by the Open Web Application Security Project (OWASP)[45] which is a worldwide not-for-profit charitable organization focused on improving the security of software.

The findings from literature study is then discussed with security experts. To prioritize the security attributes and create set of guidance, 16 semi-structured interviews were conducted based on purposive sampling of security experts.

What can go wrong?

Primary effects of security attack or accident can be broadly divided into the following 3:

Confidentiality

Confidentiality is the protection of information, especially when shared over a publicly accessible medium such as air for wireless. Confidentiality can be achieved through encryption. Different existing symmetric asymmetric encryption schemes can be leveraged to ensure confidentiality.[43] For example, hackers can get access to home monitoring camera and use that to blackmail or a patient's vital health data can be compromised if an artificial pancreas system is hacked.

Integrity

Integrity involves the protection of data and makes sure that no unauthorized modifications occur. Integrity on protection of sensor data is crucial for designing reliable and dependable IoT applications. One way this is ensured is by message authentication codes (MAC) using one way hash functions.[43] For example a home or hotel door lock can be hacked to have unauthorized access for theft. Similarly, it can be catastrophic if hackers have access to control a patient's insulin dose and it can potentially kill the patient.

Availability: Availability, which is specific to IoT, ensures that information is available when required. For example, in a smart home if the attacker knows the consumption monitoring service, he can launch the denial-of-service (DoS) attack by just trying to send false service requests and the sensor nodes are incapable of handling huge number of requests due to resource limitations. Since any transmission (i.e., receiving or sending) consume power, the node will eventually run out of its battery and make it unavailable.[43] For example, due to an attack or malfunction, user may lose remote control functionality of their furnace resulting in frozen pipes. In case of an artificial pancreas system, care givers may lose access to monitor patient's insulin level which can be potentially dangerous specially in the case of kids who do need more supervision.

IoT Security Attributes

Here is the introduction to list of top security attributes identified. The next section discusses how some of these attributes affects an artificial pancreas system. We will discuss them holistically in much more detail in next chapter with practical applications.

Physical Security

“When I am in a penetration testing assignment for a IoT vendor, my rule of thumb is that in any scenario, the device should not be able to harm people, property and the surrounding environment.” said one of the security expert during the interview.

When referring to IoT systems, physical security is all about making sure people, property, surrounding environment and the device itself is not harmed in case of accident or attack. Physical security also refers to safety of the system physically for example if the IoT device itself can be damaged or stolen?

Remote Control

Wireless technologies are becoming more popular around the world and the consumers appreciate this wireless lifestyle [56] Technologies like WiFi which is a wireless networking using RF (Radio Frequency) and BLE (Bluetooth Low Energy) are used in IoT devices widely for ease of use. Improper encryption can lead to data leak or access to the device remotely.

Maintenance

It is critical for IoT devices to allow for regular maintenance including patching and upgrades. Gartner predicts that as IoT continues to grow, vendors will favor usability over security and IT security practitioners remain unsure of the correct amount of acceptable risk.

Authentication

Authentication involves the mutual verification of routing peers before they share route information and ensures shared data origin is accurate. For example, both the service provider and service consumer needs to be assured that the service is accessed by authentic user and service is offered by an authentic source.

Authorization

It consists of access polices that explicitly assign certain permissions to subjects. The IoT environment needs to provide fine-grained, re-useable, dynamic, easy to use polices defining and updating mechanism.43

Input Validation

One of the security expert interviewed mentioned that the first commandment of secure programming is, "Thou shalt not trust user-supplied input." [46] All applications require some type of user input. User input could come from a variety of sources, an end-user, another application, a malicious user, or any number of other sources. Input validation can be used to detect unauthorized input before it is processed by the application.[47]

Cleaning

Cleaning involves sanitization and data validation which is conducted to ensure that a program operates on clean, correct and useful data. It uses routines, often called "validation rules" that check for correctness, meaningfulness, and security of data that are input to the system. Rather than accept or reject input, it is changed to an acceptable format. Any characters which are not part of an approved list can be removed, encoded or replaced.[48]

Transport Security

Device can be "tricked" into sending data to unintended, unauthorized endpoints thus it needs to be ensured that all applications are written to make use of encrypted communication between devices and between devices and the internet using accepted encryption practices.[45]

Sensitive Data

If a device stores and transmits PII (Personally identifiable information), collect passwords or any similar data that can be misused, it is dealing with sensitive data. The loss of personal sensitive data can cause financial loss, a ruined credit rating, and years of hassles as he or she struggles to recover from identity theft and thus needs to be handled responsibly.

Data Storage

Data storing securely involves preventing unauthorized people from accessing it as well as preventing accidental or intentional destruction, infection or corruption of information

Encryption

No data should be stored in clear text. They are converted into a code with standard encryption practices to prevent unauthorized access. In IoT devices, it is advisable to not store data locally but always stream it to a powerful server where it can be easily encrypted.[49]

Auditing

The IoT environments need to know when their services are accessed, who is making the service request, when the request is happening.

Error Investigation

In case of an attack or accident, error investigation is crucial to understand what went wrong so that it can be prevented from causing further damage and reoccurrence.

Logging

Logging services is critical for not only troubleshooting and maintenance, but can also be the last line of defense when it comes to feature abuse and system compromise.

Transparency

While it may not be practical for a completely open source model for every feature and application, software should be reviewable by an independent auditor which has no incentive to ignore or elide over security defects in implementation. Security by obscurity may not be always best solution and some transparency and open source concepts can help improve security in a long run. [71]

Artificial Pancreas System- Risk Analysis

System Decomposition and Explanation

The Artificial Pancreas System closely mimics the glucose regulating function of a healthy pancreas. Mainly they consist of following devices: a continuous glucose monitoring system (CGM) and an insulin infusion pump. A blood glucose device (BGD) is used to calibrate the CGM. A computer-controlled algorithm connects the CGM and insulin infusion pump to allow continuous communication between the two devices.[50]

An Artificial Pancreas System can not only monitor glucose levels in the body but can be used to adjust the delivery of insulin to reduce high blood glucose levels (hyperglycemia) and minimize the incidence of low blood glucose (hypoglycemia).

Figure 13 shows a system decomposition of the artificial pancreas system.

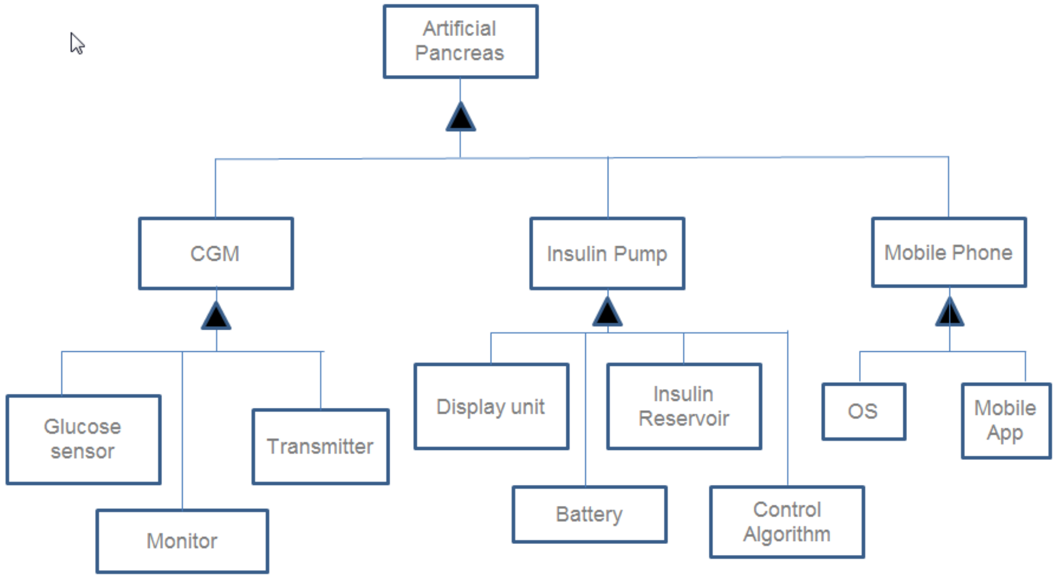


Figure 13 2 level decomposition of artificial pancreas system

Figure 14 is another way to slice and dice the system into form and function and explains various components of the artificial pancreas system that is needed to conduct attack analysis. In Figure 15, the primary focus of this study is on the subsystem marked by dark lines. This system represents the most direct control structure for patient safety. The new system added the Patient's cell phone into the system control, which transformed the system into an IOT device.

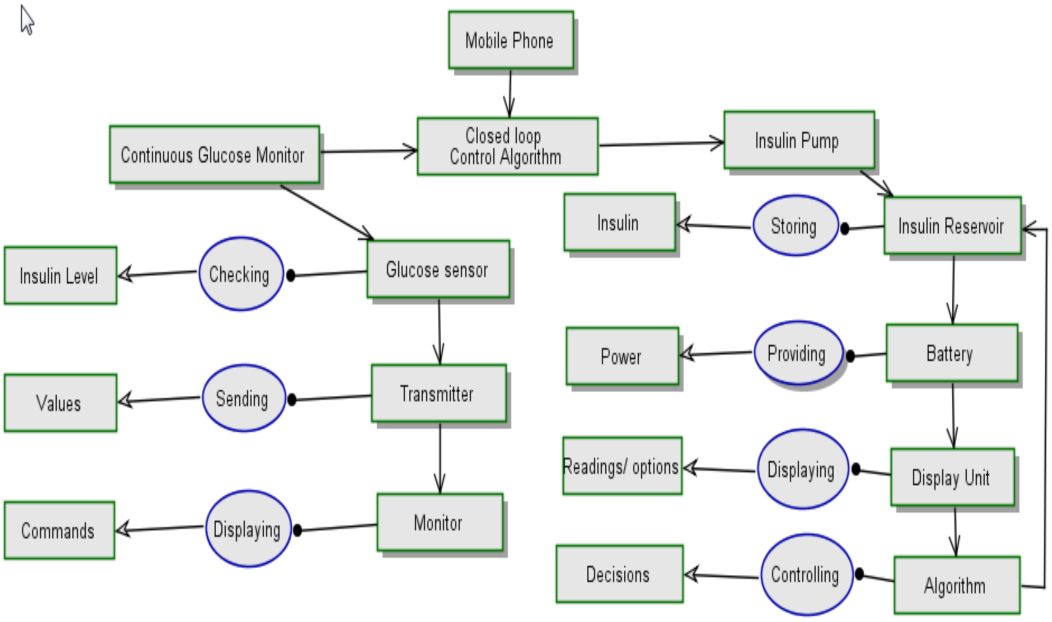


Figure 14 Form and Function diagram depicting how mobile phone interfaces with artificial Pancreas System

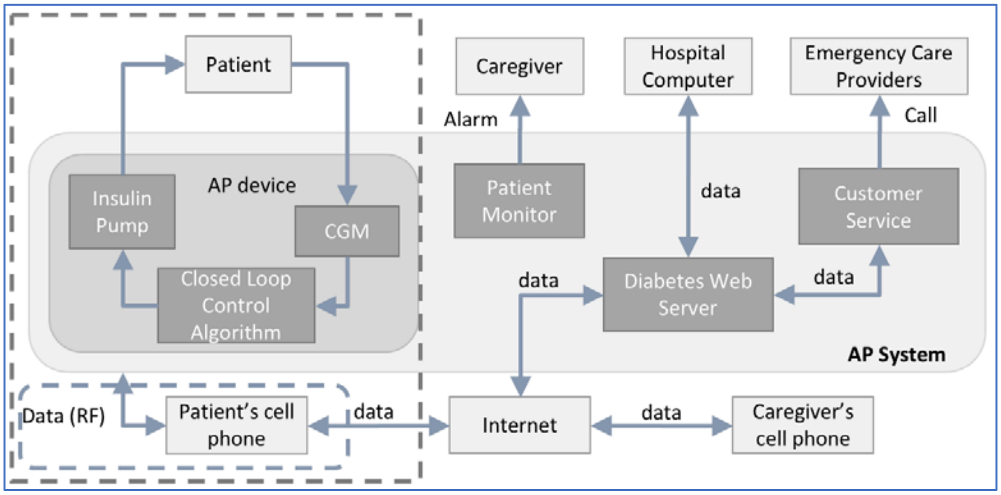


Figure 15 Shows how data flows between various stakeholders through components of artificial pancreas system

Security Attack Analysis

While researching artificial pancreas system, the study found there has been some research around hacking into it. Here is the attack analysis:

Artificial Pancreas System

What are the functions delivered by the systems, primary and secondary?	Continuous Glucose Meters (CGM): A sensor placed under the patient's skin (subcutaneously) measures the glucose in the fluid around the cells (interstitial fluid) which is associated with blood glucose levels. A small transmitter wirelessly sends information to an external receiver.[50] Insulin pump: Based on the instructions sent by the controller, the pump delivers insulin to the tissue under the skin; Special USB dongles used to program Insulin Pumps and download history data; Special wireless remotes used to deliver insulin
How does the system deliver the functions?	CGM has wireless sensors attached to special wire inserted into tissues. The sensor measures resistive value of interstitial fluid to measure sugar levels. The data is then wirelessly transmitted to external meter device for graphing.
How will you decompose the system?	First, there is a java based program that uses a wireless peripheral device to configure all the settings on the device. Second, there is a blood glucose meter that can communicate the results of a blood strip test to the insulin pump. This makes it more convenient for a diabetic to enter those values into the insulin pump. Third, this insulin pump also has a CGM functionality, allowing the use of a CGM sensor that works the same as the stand alone CGM device mentioned above. All three use the same wireless interface on the insulin pump.
How does the system connect to IoT network?	Wireless communication interface. The RF transmitter is always listening even if the remote option of the pump is turned off.
What is the attack?	McAfee’s Barnaby demonstrated that he can hack into pump through the RF transmitter listening channel. “Once the hacker sets foot in the targeted machine, he can then disable the

	warning function or/and make it disperse 45 days' worth of insulin all at once – a dose that will potentially kill the patient." The other attack is to wirelessly connect to the pump and use the java based program to change the configuration setting. "An attack would need little time, as the changing of a configuration setting would only take moments. For example, the setting that controls the ratio of insulin given at meal time could be altered. If a user is supposed to get 1 Unit of insulin per 5 grams of carbohydrate eaten, the attack could change that to 1 unit of insulin per 3 grams of carbohydrates eaten. This is a significant enough difference to cause a diabetic to become hypoglycemic roughly 60 to 90 minutes after eating."
Which functions do the attack affect?	The dose amount of insulin delivered to patients.
Which functions do the attack utilize?	The wireless interface that allows external control of the device for convenience - change configuration settings, change amount of insulin.
Who discovered the attack and how?	Rapid7 researcher Jerome Radcliffe (a computer security researcher and with lots of hardware wireless experience). His research demoed attack to change amount of insulin.[51] McAfee researcher Barnaby also researched Pacemaker and Implantable Medical Device (IMD), and found that he can remotely change the voltage of IMD, even deliver new firmware to many IMD devices at same time (a worm to infect many patients at once).[52]
What is the reaction to the attack from the security researchers/community?	Jerome's suggestion: • New RF chips have crypto on board, use it - (it can be unpractical, crypto requires more power use, more time)

	• Use IR rather than RF – inconvenient and have smaller range, but more secure • Verify New Configuration • Setting a Passcode - (this is not practical since it can impose danger in emergency when patients forgot passcode or are unconscious). • Keep range limited o One pump uses 13mhz On-Off Keying (OOK). This is much more than needed for artificial pancreas devices and thus increases remote control risks. • Blocking o Researchers are working on RF blocking for stopping RF OOK Pacemakers from malicious interference

Table 7 Security attack analysis- Artificial pancreas system

Findings and Related Security Attributes

1. **Remote control**: Analysis identified possible remote attack venues. Of course, the entire path should be secured as much as it is practical).
2. **Physical Security**: It seems that the key is the configuration and amount of insulin. One suggestion is to enforce constraints on maximum and minimum amount of insulin to give to patients within a certain period. These safety constraints should be enforced in control algorithm and mobile phone app. Also, it will be great to provide feedback to users when safety constraints are violated. This would prevent fatal attacks.
3. **Transport Security**: But then the question is if malicious attack can intercept the data from the sensor and send the system misleading data, or change the system configuration to manipulate the amount of insulin, not dangerous but incorrect, how would the system detect that?

Also, to harden and secure the device, it is necessary to be able to update/patch the firmware, OS, and application software as vulnerabilities are discovered. (Security attribute: Maintenance). In next chapter, we will tally these casual analysis findings with the proposed tool and verify how the tool works out for this use case.

CHAPTER 6: Application and Future Work

Here, the findings from this study is presented in an application tool format which an IoT device product manager can use to prioritize their security needs while introducing functional usability features.

Application 1: System Security Scale

The goal of this study is to empower IoT device product manager by making him aware of potential security issues and better protect the company from releasing products that introduce unanticipated risks to their customers while adding new usability features. This framework or tool must be universally applicable and easy to use. One of the tool researched as a case study is a well-established usability tool called "System Usability Scale"

System Usability Scale

The System Usability Scale (SUS) provides a "quick and dirty", reliable tool for measuring the usability. It consists of a 10-item questionnaire with five response options for respondents; from Strongly agree to Strongly disagree. Originally created by John Brooke in 1986, it evaluates a wide variety of products and services, including hardware, software, mobile devices, websites and applications. [53] [54]

SUS generally provides high-level subjective view of usability and is thus often used in carrying out comparisons of usability between systems. The tool was created more than 40 years ago and still holds relevancy and popularity among usability practitioners because of its ease of use and the way they were written to keep the questions at a very high level. This tool is meant to be a "quick and dirty" tool instead of an exhaustive one. Here are the 10 questions which forms SUS:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

The scoring is based on a 0-100 scale with anything below 60 is assigned a grade "F" and anything above 90 is grade "A" as seen in Figure 16.

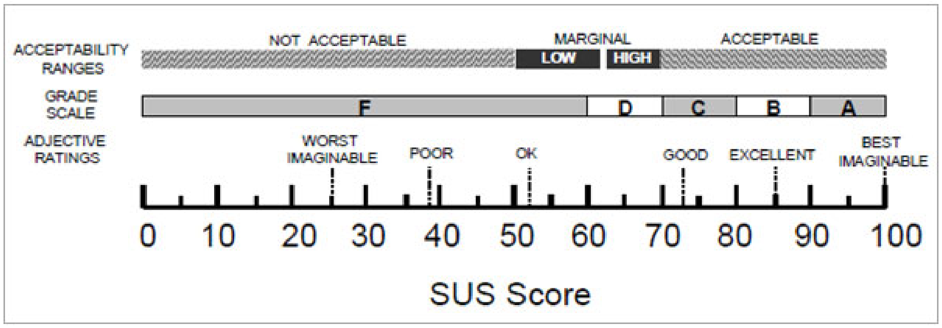


Figure 16 SUS scoring mechanism

Once security attributes were prioritized and consequences were understood based on the purposive interviews with 16 security expert and literature studies, the information was organized into set of 10 questions which have equal weightage with every "Yes" is a positive point and "No" is a negative point on the proposed "System Security Scale". The questions are deliberately kept very generic so that it can be applied to any IoT device. The aim of this study is not only make the IoT product manager aware of security concerns but also equip him and his development team with some easy to understand guidance on how to mitigate those security concern. The table below shows the summary of questions and recommended improvement followed by detailed guidance for each question:

System Security Scale and Guidance

System Security Question	Affects (C,I,A)	Improvement Recommendation	Security Attributes
1. Is it impossible for the feature to affect the health and safety of people or property?	availability	Provide safety guarantees for failure conditions	physical security
2. Does the feature require a local, physical interface to access it?	availability	Lock down all control and data input interfaces	remote control
3. Can authorized users or devices patch or update the feature in the future?	integrity	Build and maintain a patch / update service	Maintenance
4. Can only authenticated, authorized users or devices access the feature?	availability confidentiality	Construct and enforce authentication and authorization policies	authentication authorization

5. Is all received data automatically inspected and validated?	availability integrity	Validate all input	cleaning input validation
6. Are data transmissions encrypted and mutually authenticated?	confidentiality integrity	Use secure transport techniques	transport security
7. Does the feature avoid storing personally identifying information, tokens, or passwords?	confidentiality integrity	Be deliberate and careful with secure storage of credentials	sensitive data
8. Is any stored data only accessible after authentication by an authorized user or device?	availability confidentiality	Consider encrypting data at rest	data storage encryption authorization
9. Does the feature routinely log use and errors in a way that authorized users can inspect the logs?	logging integrity	Store log data securely	auditing error investigation
10. Is the source code available for inspection by a third party?	integrity	Adopt open source principals where appropriate, and accepted vulnerability disclosure practices	transparency

Table 8 Final set of SSS questions and guidance

Here are the security attributes and guidance created based on the study:

1. **Is it impossible for the feature to affect the health and safety of people or property?**

(Physical Security, Availability)
Improvement: Provide safety guarantees for failure conditions

In cases where the feature can fail accidentally, a secure design must also consider the possibility that the feature can fail due to malicious action, and should fail in a way that presents the least likely harm to people, property and the surrounding environment. For example, a fire suppression system might not allow remote users from exercising the system when no danger is actually present, but a failure in the sensors that detect smoke and fire should not prevent that authorized user from triggering the fire suppression system. A balance must be struck, and documented, that details the expected failure condition of any component of a system that exercises physical actions.

Another consideration is preventing the actions of a **malicious user**. When engineering for physical safety, the actions of a directed attacker are often not considered when determining reasonable safety precautions in the design phase of a project. Automobile safety, for example, includes design features such as seat belts, crumple zones, and anti-lock brakes, and all envision an accidental loss of control of the vehicle.

In designing secure software that controls a physical device, we must assume that the authorized user account may be compromised by a malicious user. Depending on the network capabilities of the device, this user may be remote, far away from the nearby device, and therefore, will not personally suffer the consequences of physical danger.

It is possible to mitigate the scenario involving a malicious user by designing a feature to never trust remote input for personally or environmentally dangerous activities using local-only, physical input interfaces. [45]

In cases where a remote interface is essential to the feature, a cryptographically strong network connection (Q6) and input inspection (Q5) is required. This can protect against a man-in-the-middle (MitM) attack [55] which alters the user's intentions during transit, as well as offer strong authentication and session management to prevent user accounts from being compromised by a malicious actor.

2. Does the feature require a local, physical interface to access it?

(Remote Control, Availability)
Improvement: Lock down all control and data input interfaces

When designing features for devices that lack local input mechanisms, such as a monitor, keyboard, and mouse (KVM), such features are necessarily interacted with remotely. For a traditional networked feature, this interaction can take place over a wired Ethernet connection to a packet-switched, routed network, or an 802.11 wireless network connection (either mediated through a shared access point or directly via peer-to-peer networking). Features may also be exercised over Bluetooth

Low Energy (BLE), radio frequencies (RF), or other physically distant communication channels.

In the case of all electromagnetic-based communications (WiFi, BLE, or RF), it should be assumed that the effective range of an attacker can be much farther than the expected use case, thanks to the use of high-gain antennae and high-powered transmission equipment.[56] Other energy media, such as magnetic, optical, and laser systems, typically have shorter ranges, although these, too, can be extended to surprising levels by innovative attackers.

Therefore, all electromagnetic-based interfaces should be considered remote interfaces, and all such communications should provide strong encryption (See Q6), input inspection (See Q5), and, where appropriate, authorized user authentication [43] (See Q4). These secure usability considerations are critical for both protecting data transmitted to and from the feature, as well as regulate access to the feature. For example, it doesn't make sense to have long range RF chip in an artificial pancreas system which can expose to remote attacks.

3. Can authorized users or devices patch or update the feature in the future?

(Maintenance, Integrity)
Improvement: Build and maintain a patch / update service

Failing to provide for a mechanism to update a feature can lead to a situation where a device is permanently vulnerable to a post-production discovered issue during normal operation, therefore putting the integrity of the system at risk.

In software development, shipping defects is all but certain, and maintenance is required to fix, patch, or remove features that introduce new issues. These updates can be delivered in a variety of ways depending on the specific product or service, and can be characterized as in-band or out-of-band, and manual or automatic.

The most successful means to distribute patches and updates are automatic, in-band updates. A process runs on the system that periodically checks for updates, and if there is one available, it is obtained and applied, and any affected services are restarted, all without user interaction. Most workstation-based web browsers apply this strategy, as well as mobile applications that are configured for automatic updates.

Some strategies involve a manual process, where an authorized user must intentionally seek out an available update, obtain it, apply it, and restart any affected services. Many operating systems are configured this way by default, primarily to avoid unscheduled reboots.

Some products have no direct access to an update source, and updates must be obtained over an "external" channel, such as a portable disk delivered to the user via physical mail, or, in the case of regulated industries, an authorized technician (rather than the authorized user), must apply the update.[57] This is the common case for automobiles and medical devices, for example.

While providing updates is an integral component of a secure maintenance program, care must be taken to ensure that updates are from a cryptographically-verified trusted source [45] (see Q6), are locally verified as authentic and complete before updating (see Q5), and that a post-update procedure provides an evidence that the update was, in fact, successfully applied (see Q9).

4. Can only authenticated, authorized users or devices access the feature?

(Authentication, Authorization, availability, confidentiality)
Improvement: Construct and enforce authentication and authorization policies

Authentication involves the mutual verification of routing peers before they share route information and ensures shared data origin is accurate. Both the service provider and service consumer needs to be assured that the service is access by authentic user and service is offered by an authentic source. [43]

Pre-authenticated interactions with a device are sometimes unavoidable. After all, the act of authenticating is necessarily a pre-authenticated interaction. Extra care must be taken to ensure that the data provided by an unauthenticated user is safe to handle and process (see Q5), and that unauthenticated interactions are as restricted as possible. [58]

If a feature is intended to be exercised by anonymous users, those interactions should, where possible, be limited to read-only access to stored files and memory, and only then, should be limited to reading only public, non-sensitive data. There should be no circumstance where an unauthenticated user can read personally sensitive data (see Q8).
If write access must be granted to anonymous users, it should be assumed that those users will, accidentally or intentionally, provide malformed or malicious data, or data

intended to overflow the storage capacity of the device. Therefore, a balance must be struck between logging normal operations (see Q9) and maintaining the stability and security of the logging system itself.

Finally, if remote access to the feature is possible (see Q2), and that remote access includes unauthenticated access, it should be assumed that the user may not be the intended, authorized user of the device. [59]

5. Is all received data automatically inspected and validated?

(Input Validation, Cleaning, availability, integrity)
Improvement: Validate all input

User supplied data can't be trusted without proper vetting. This is especially true when input is being passed directly from an untrusted source to any sort of comparison or rendering function, such pre-authentication login page (see Q4). For example, if a username is to be compared against a database of valid usernames, and rendered back to the user on a successful or failed login screen, that username must be ensured to be safe for not only the comparison function and rendering function, but also for any logging functions (see Q9). If a username contains an unexpected character that is a meaningful terminator or sequence of characters, such as a NUL, a semicolon, or an HTML tag, this could lead to unexpected results.[60] A failure to check and prevent these characters from being passed in an unsafe way is the root cause of many classes of vulnerabilities, from buffer overflows (BOF) to cross-site-scripting (XSS) to SQL injection (SQLi).

Some vulnerabilities may be non-obvious and not rely on controlling the execution flow of a feature or application. For example, a feature that registers new users should ensure that usernames contain only expected characters (such as alphanumeric characters only), of the expected length (of one to twenty characters), and not contain reserved or misleading values (such as "Administrator" or "Welcome").

Features that accept input in the form of data files should also make sure that the expected data matches what is actually received. This is made more complex due to the attack technique of creating polyglot files, which are files that may be rendered as several types of data, depending on the context. In web application programming, for example, it is often possible to construct files that appear as an image file, due to the presence of "magic" bytes in the file's header, as well as an executable script in a language like PHP. Therefore, filtering mechanisms must be sophisticated enough to

distinguish between allowed file formats (using a whitelist approach) and prohibited formats (using a blacklist approach).
It is important to note that even authorized, authenticated users should be subjected to input validation, in order to prevent against a privilege escalation attack where a regular, non-privileged user attempts to gain administrative control over the feature, application or underlying operating system (see Q2 and Q3).

Data that is received from a non-human user, such as another feature or device, must also be validated before processing or storage. In many cases, such data is naively trusted as coming from a sensor or other component without sufficient, cryptographically assured authentication (see Q6).

6. Are data transmissions encrypted and mutually authenticated?

(Transport security, Confidentiality, integrity)
Improvement: Use secure transport techniques

While encryption over remote interfaces is critical in protecting data from eavesdropping or alteration in transit, equally important is the ability to guarantee that the destination of the data is, in fact, the intended destination. If a feature or device can be "tricked" into sending data to unintended, unauthorized endpoints, this can lead to a compromise of sensitive data such as personally identifying information (see Q7) or remote logging data (see Q9). Such techniques for compromising data usually involve spoofing, or impersonating, the address of the intended endpoint over a network.[61] For example in an artificial pancreas system, if malicious attack can intercept the data from the sensor and send the system misleading data, or change the system configuration to manipulate the amount of insulin, it can be life threatning.

While transport security is especially important in cases where an application or device is intended to be used over inherently untrustworthy networks, such as the internet or shared, public WiFi networks, encrypted communications should also be used on nominally "private" networks.[62] This can help defend against a compromise of the network infrastructure, especially when those networks rely on radio frequencies with no practical means of defending against injection or impersonation attacks (see Q2).

7. Does the feature avoid storing personally identifying information, tokens, or passwords?

(Sensitive Data, Confidentiality, integrity)
Improvement: Use secure transport techniques

The best defense against a data breach is to avoid storing useful data in the first place, closely followed by a well-defined, well-documented mechanism to limit the damage from a breach. In the event of a failure of authentication controls (see Q4) or transport security (see Q6), a device which stores no personally identifiable information is necessarily a less attractive target for attackers. [63]

In the case where a feature is designed to interact with another, third-party application, device, or other system, such access should be controlled using expiring tokens, which are used solely to authorize the access from this particular endpoint. In this way, tokens can either not be reused at all, or if they are, can be easily disabled once a compromise is detected.[64] Storing and using unique usernames and passwords should be avoided. Generalized account access should be minimized whenever possible. Modern single sign-on (SSO) services, for example, use special-purpose authorization tokens, rather than storing and forwarding passwords themselves, which helps control access in the event of a security failure.

As far as non-token, non-password data is concerned, personally identifying information should be avoided whenever practical. Some features, of course, require the collection and storage of PII such as a user's legal name, phone number, address, geolocation data, and other specific pieces of information. In these cases, the transmission or publication of this data should be conducted over encrypted channels (see Q6), and storing this data locally should be similarly encrypted (see Q8).

8. Is any stored data only accessible after authentication by an authorized user or device?

(Data Storage, Encryption, Authorization, Availability, Confidentiality)
Improvement: Consider encrypting data at rest

Encrypted data storage is critical to maintain reasonable security in the event the physical hardware becomes lost, is stolen, or is otherwise in the physical control of an adversary. Minimally, sensitive data such as passwords, personally identifying information (see Q7), and access logs (see Q9) should be stored in an encrypted format to prevent unauthorized use and tampering. [65]

Many applications, such as those that operate on modern smartphones and tablets, rely on the operating system for data encryption capabilities, since smartphones and tablets are much more likely to be lost or stolen than traditional desktop PCs or rack-mounted servers. [66] These devices require authentication on first power-on, and this tends to be the accepted compromise between security and usability.

Sometimes, upon installation, it is possible to check the operating environment for flags or other indicators that lower-level encryption services are available.[67] However, if this is impossible, and encryption services are not reasonably guaranteed by the operating system or platform, features and applications should provide their own encryption services using standard encryption techniques.

In other words, if it is possible for encryption to be disabled on a given platform, and it is impossible to determine if encryption services are available, it should be assumed that platform is unencrypted by default, and the feature should defend against this by requiring a decryption key or password for data storage access. [42]

9. Does the feature routinely log use and errors in a way that authorized users can inspect the logs?

(Auditing, Error Investigation, Logging, Integrity)
Improvement: Store log data securely

Providing reasonable, local logging services is critical for not only troubleshooting and maintenance, but can also be the last line of defense when it comes to feature abuse and system compromise. Of course, system logs themselves need to be protected by reasonably secure authorization (see Q4) and encrypted storage (see Q8), or else an adversary may be able to glean personally sensitive information (see Q7) about the users or the environment the feature operates in.

The data format of logs should be resilient to tampering. One approach to this is to include a cryptographically strong hash of each log entry that depends on the correct hash of the prior log entry, and any log deletion action should itself be logged.[68] The chosen format of log files should also include unambiguous timestamps for each event, and the events should be accessible and understandable by authorized users (and only authorized users). [69] Logging activities do come with a cost in terms of processing and storage, so care should be exercised to choose an appropriate logging fidelity; too much data is sometimes just as bad as not enough.

Finally, to defend against local log file corruption or tampering, logs should be stored both locally, and have the capability to be transmitted remotely, using normal transport encryption (see Q6). The decision to employ a push or pull model of log transmission is an implementation-specific detail, and will depend on what is most appropriate for the given application.

10. Is the source code available for inspection by a third party?

(Transparency, Integrity)
Improvement: Adopt open source principals where appropriate, and accepted vulnerability disclosure practices

There are many valid arguments in favor of closed, proprietary software models, not the least of which is the defense of intellectual property, trade secrets, and other "secret sauce." However, it is rare that "security by obscurity" is a sufficient defense against even casual adversaries.[70] In many cases, critical software vulnerabilities are discovered by accident by otherwise non-expert users, so keeping a feature's source code secret tends to discourage those experts that are most able to help resolve a vulnerability before they become widely exploited.[71]

While it may not be practical for a completely open source model for every feature and application, software should be reviewable by an independent auditor which has no incentive to ignore or elide over security defects in implementation. Open source, of course, enables this sort of in-depth review by experts, but it should be noted that an open source model of software development does not guarantee independent audits for secure coding.

Finally, an open source model of development does have a debilitating effect on short-term secrecy. When vulnerabilities are discovered, and patches are created, the act of patching an open source feature is effectively publically disclosing a vulnerability before end users can acquire and apply the patched feature (see Q3). This is especially true for libraries that are intended to be used in a supply chain of downstream technology providers, in cases where end-users of the feature cannot reasonably apply patches and updates themselves. [72]

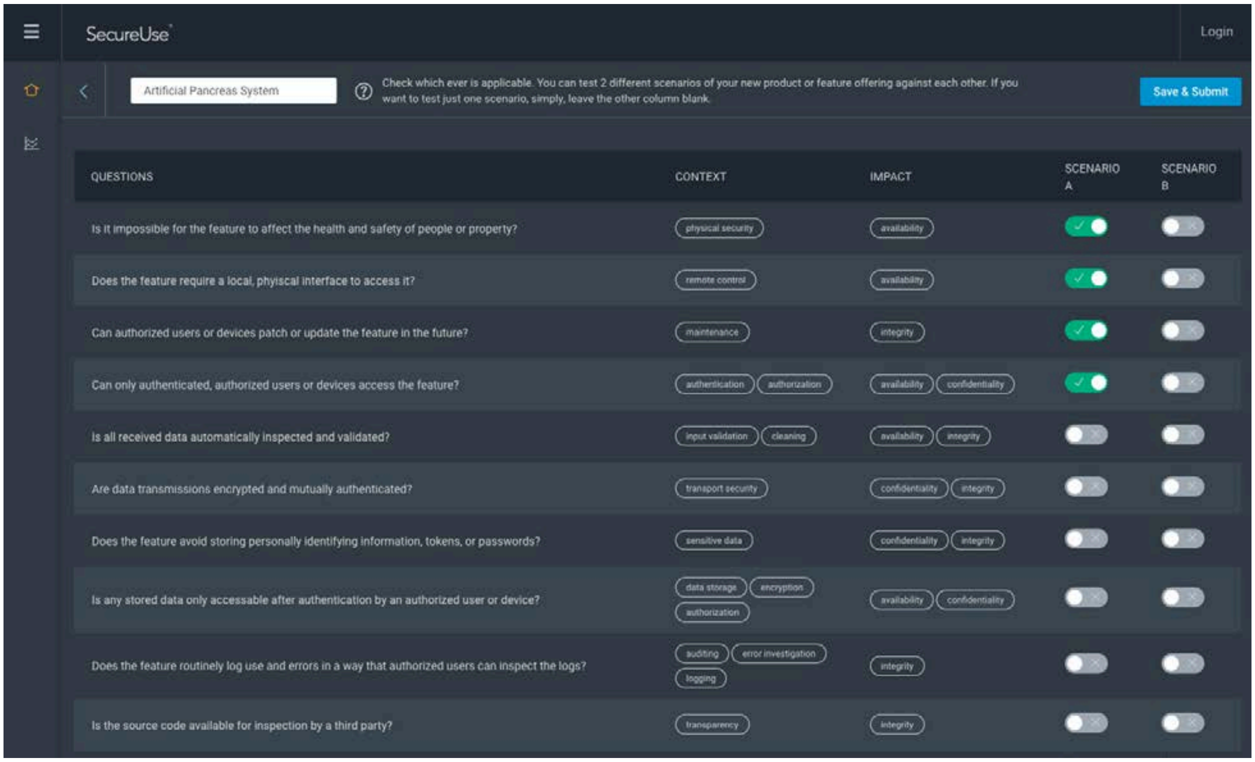
In these cases, short-term secrecy about a given vulnerability and the patch should be employed, and secure communication channels established with downstream providers, shortly before patches are applied to a publicly reviewable codebase. [73]

SecureUse Prototype- Evaluating Connected Artificial Pancreas System

SSS (System Security Scale) questions are added in a A/B test like format- A/B testing is comparing 2 version of anything. It can be a website, application or a product compared on certain attributes. It doesn't even have to be 2 items to compare. A single product can be also tested for its security resiliency against the 10 questions. The prototype product created for this study is named "SecureUse" that can be used to identify security priorities. Here are 2 examples:

1. Test a new product idea- User wants to connect a physical device to the internet like a regular home lock. So, Scenario A is a regular home lock while Scenario B is an internet connected home lock which can be unlocked through user's smart phone remotely.
2. Test a proposed new feature in an existing system. Compare the new proposed feature- insulin push capability through mobile phone in artificial pancreas system (Scenario A) to the current state of the system where the artificial pancreas system do not have any connectivity feature and insulin push is controlled by a physical button on the device (Scenario B)

Elaborating on the artificial pancreas system, based on previous findings in this study, here is an example of how this test may look like if a user test a scenario where Scenario A is a regular artificial pancreas system which is not connected to Bluetooth or internet vs. Scenario B which is envisioned as a IoT device connected to Bluetooth and internet. Hovering on each question also provides user with additional context on the security questions in a layman's language so that someone without security knowledge can answer the questions and get guidance on security. Figure 17 shows the screenshot of SecureUse prototype where user provides input. For reading actual questions, see Table 8. Figure 18 shows the output received based on the questions answered. For reading actual text, see System security scale and guidance number 5 above in this chapter.

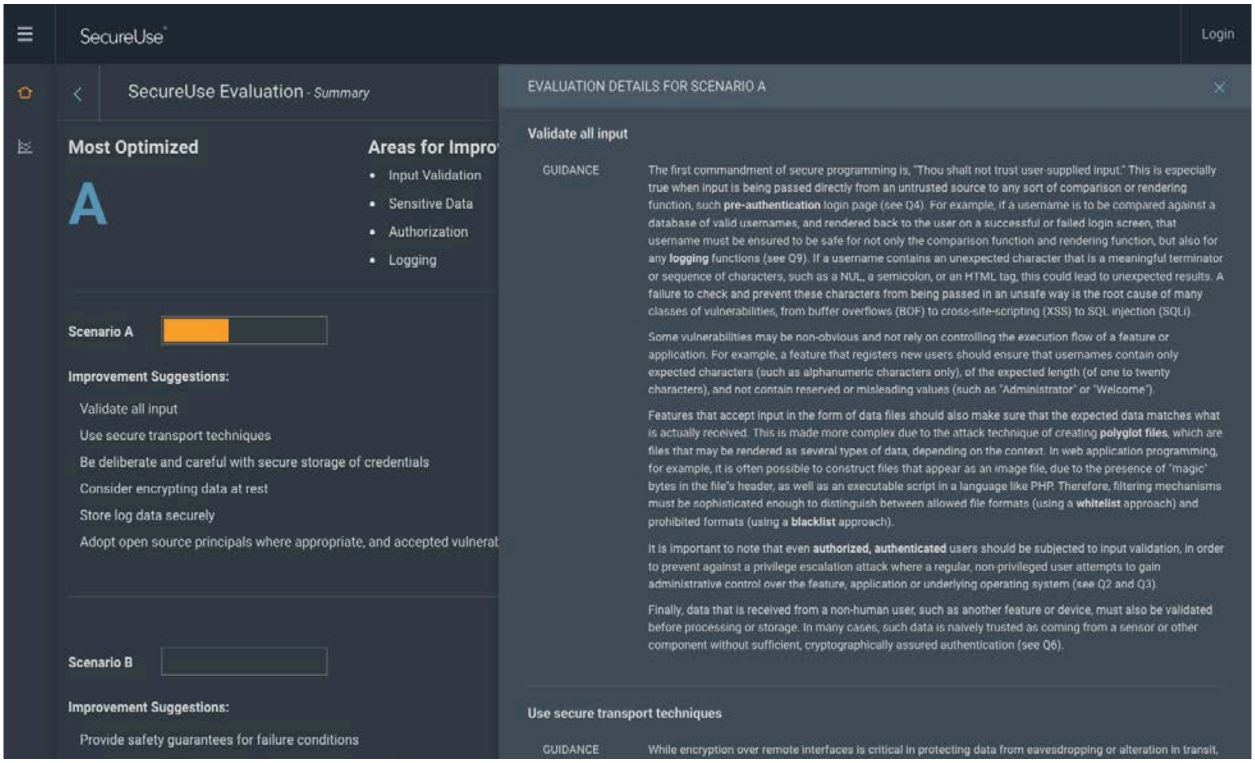


Artificial Pancreas System

Check which ever is applicable. You can test 2 different scenarios of your new product or feature offering against each other. If you want to test just one scenario, simply, leave the other column blank.

QUESTIONS	CONTEXT	IMPACT	SCENARIO A	SCENARIO B
Is it impossible for the feature to affect the health and safety of people or property?	physical security	availability	✓	✗
Does the feature require a local, phyiscal interface to access it?	remote control	availability	✓	✗
Can authorized users or devices patch or update the feature in the future?	maintenance	integrity	✓	✗
Can only authenticated, authorized users or devices access the feature?	authentication authorization	availability confidentiality	✓	✗
Is all received data automatically inspected and validated?	input validation cleaning	availability integrity	✗	✗
Are data transmissions encrypted and mutually authenticated?	transport security	confidentiality integrity	✗	✗
Does the feature avoid storing personally identifying information, tokens, or passwords?	sensitive data	confidentiality integrity	✗	✗
Is any stored data only accessable after authentication by an authorized user or device?	data storage encryption authorization	availability confidentiality	✗	✗
Does the feature routinely log use and errors in a way that authorized users can inspect the logs?	auditing error investigation logging	integrity	✗	✗
Is the source code available for inspection by a third party?	transparency	integrity	✗	✗

Figure 17 Shows user answering questions to compare a current artificial pancreas system to a proposed connected device



SecureUse Evaluation - *Summary*

Most Optimized

A

Areas for Impro

- Input Validation
- Sensitive Data
- Authorization
- Logging

Scenario A

Improvement Suggestions:

Validate all input

Use secure transport techniques

Be deliberate and careful with secure storage of credentials

Consider encrypting data at rest

Store log data securely

Adopt open source principals where appropriate, and accepted vulnerat

Scenario B

Improvement Suggestions:

Provide safety guarantees for failure conditions

EVALUATION DETAILS FOR SCENARIO A

Validate all input

GUIDANCE

The first commandment of secure programming is, "Thou shalt not trust user-supplied input." This is especially true when input is being passed directly from an untrusted source to any sort of comparison or rendering function, such **pre-authentication** login page (see Q4). For example, if a username is to be compared against a database of valid usernames, and rendered back to the user on a successful or failed login screen, that username must be ensured to be safe for not only the comparison function and rendering function, but also for any **logging** functions (see Q9). If a username contains an unexpected character that is a meaningful terminator or sequence of characters, such as a NUL, a semicolon, or an HTML tag, this could lead to unexpected results. A failure to check and prevent these characters from being passed in an unsafe way is the root cause of many classes of vulnerabilities, from buffer overflows (BOF) to cross-site-scripting (XSS) to SQL injection (SQLi).

Some vulnerabilities may be non-obvious and not rely on controlling the execution flow of a feature or application. For example, a feature that registers new users should ensure that usernames contain only expected characters (such as alphanumeric characters only), of the expected length (of one to twenty characters), and not contain reserved or misleading values (such as "Administrator" or "Welcome").

Features that accept input in the form of data files should also make sure that the expected data matches what is actually received. This is made more complex due to the attack technique of creating **polyglot files**, which are files that may be rendered as several types of data, depending on the context. In web application programming, for example, it is often possible to construct files that appear as an image file, due to the presence of "magic" bytes in the file's header, as well as an executable script in a language like PHP. Therefore, filtering mechanisms must be sophisticated enough to distinguish between allowed file formats (using a **whitelist** approach) and prohibited formats (using a **blacklist** approach).

It is important to note that even **authorized, authenticated** users should be subjected to input validation, in order to prevent against a privilege escalation attack where a regular, non-privileged user attempts to gain administrative control over the feature, application or underlying operating system (see Q2 and Q3).

Finally, data that is received from a non-human user, such as another feature or device, must also be validated before processing or storage. In many cases, such data is naively trusted as coming from a sensor or other component without sufficient, cryptographically assured authentication (see Q6).

Use secure transport techniques

GUIDANCE

While encryption over remote interfaces is critical in protecting data from eavesdropping or alteration in transit,

Figure 18 Show result of comparison along with guidance to look out for certain security issues followed by guidance on how to go about them

Future Work

Not in scope of this study but as part of future work, it would be great to explore possibility of an actual metric derived from a combination of SUS (System Usability Scale) and SSS (System Security Scale). SUS which has set of 10 questions has an established metric with a range of 0-100. SSS has a potential to be converted into a similar metric. The resulting quantitative metric can be verbally stated in the form of the ubiquitous blood pressure rate. For instance, 70/85 or seventy over eighty-five would signify that both security and usability levels are high. For ease of comparison with other options, the results would be visualized within a Tornado Chart as shown in Figure 19. The ideal solution being a design that scored highly on both sides of the chart. For instance, within Figure 27, Option 7 clearly satisfied each evaluator, including representative users, more so than other options. [74]

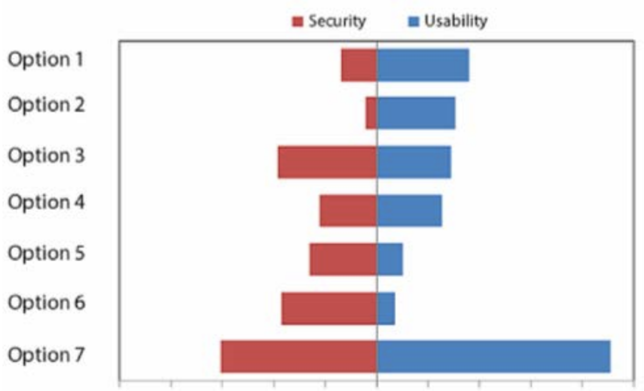


Figure 19 Proposed Tornado chart showing Security-Usability combined metrics

Application 2: Security- Usability QFD

QFD (Quality Functional Diagram) is a product development methodology whose objective is to "deploy" the Voice of the Customer [75] throughout the product development process. It helps to engage cross-functionality in an orderly, truly participative way, enlisting the support of all major functions within the organization toward a common view by creating series of matrices, the first and most common of which is referred to as the House of Quality (HOQ). These matrices help map customer needs to product features which is used to prioritize product's design specifications.[77]

A rigorous view of QFD employs not just one matrix (the original House of Quality), but multiple matrices namely Customer needs to performance measures, Performance measures to features or solutions, Features or solutions to parts specifications and Parts specifications to manufacturing processes. [77]

There are other variations of QFD- one popular variation is called Pugh Concept selection developed by Professor Stewart Pugh.[76] That variation attempts to evaluate different product concepts against the key performance measures, with the objective of incrementally moving toward an ideal concept.

As a part of this study, a new variation is being explored which attempts to evaluate functional usability features to security requirements, with the objective of identifying prioritized security concerns. In this study, we used artificial pancreas system as an example. Figure 20 shows the QFD plotted with following information:

Left side- Functional Usability Requirements
Left side rows are used to list the prioritized product features that needs to be evaluated against security concerns. In the diagram, there is a prioritized list of functional usability features for a connected artificial pancreas system which is discussed in detail in *Chapter 4, Table 9 Top Functional usability feature requirements for Artificial pancreas system mapped to QUIM most relevant usability attributes.* Note that the requirements are measured on a scale of 1-5 where 5 is the most important requirement. This is reflected on the 1st column to right. The next column shows the priority in percentage.
Note: This is the variable in the template. For any other project, these requirements need to be updated as per project need.

Ceiling- Security Requirements: This is list of IoT security attributes grouped by relevance as discussed in Chapter 6, *Table 10 Final set of SSS questions and guidance.* The row directly below the ceiling shows which security attributes need to be maximized and which ones need to be minimized to optimize security of the system.
Note: These security requirements are proposed as static attributes of this template and can be applied to any project.

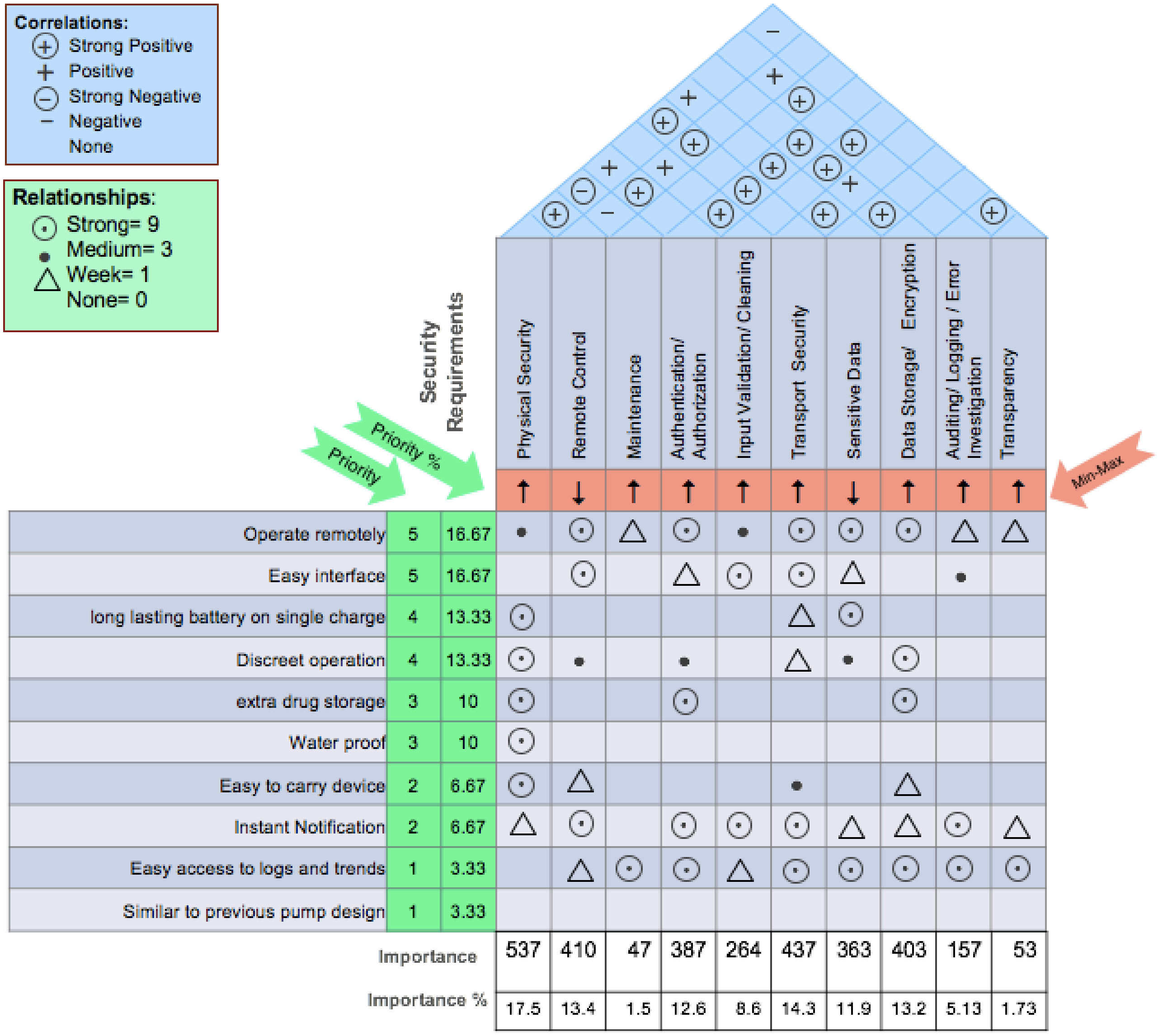
The Roof: This is the triangle shaped cap in the diagram. This shows a matrix describing the correlation between the security attributes. It can show how the security requirements affects each other. The correlation between them is classified into 4 categories:

- Strong Positive: For example, improving encryption and data storage will also improve sensitive data security equally
- Positive: For example, better transport security will affect authentication and authorization security somewhat positively
- Strong Negative: No strong negative identified here.

- Negative: For example, improving security for remote control feature may make maintenance of a device by customer service more difficult due to accessibility
- None: Many security attributes may not correlate to each other at all

Lower level / Main body: This is where the relationship is mapped between a functional usability feature and security requirements which is basically left side and ceiling in the HOQ/QFD.

- Relationship is mapped on a scale of Strong, Medium, Weak, None which are represented by various symbols and are assigned numeric value 9,3,1 and 0 respectively based on standard QFD procedure. So for example, in security requirement 1st column "Physical security" is related to each functional usability requirement represented by a relationship value.
    - Operate remotely relationship= Medium (3)
    - Easy interface relationship = None (0)
    - Long lasting battery on single charge relationship = Strong (9) and so on…
- Now the importance of each security requirement is calculated by Summing up (Priority % of each functional usability feature Multiplied by security requirement relationship assigned value). So, as an example in column 1 for physical security, the calculation will look like this: (16.67 x 3) + (13.33 x 9) + (13.33 x 9) + (10 x 9) + (10 x 9) + (6.67 x 9) + (6.67 x 1) = 537
- Then the importance is converted to importance percentage. For example for the 1st column in diagram below, 537/ 3058 x 100= 17.5%



	Priority	Priority %	Physical Security	Remote Control	Maintenance	Authentication/ Authorization	Input Validation/ Cleaning	Transport Security	Sensitive Data	Data Storage/ Encryption	Auditing/ Logging / Error Investigation	Transparency
			↑	↓	↑	↑	↑	↑	↓	↑	↑	↑
Operate remotely	5	16.67	●	⊙	△	⊙	●	⊙	⊙	⊙	△	△
Easy interface	5	16.67		⊙		△	⊙	⊙	△		●	
long lasting battery on single charge	4	13.33	⊙					△	⊙			
Discreet operation	4	13.33	⊙	●		●		△	●	⊙		
extra drug storage	3	10	⊙			⊙				⊙		
Water proof	3	10	⊙									
Easy to carry device	2	6.67	⊙	△				●		△		
Instant Notification	2	6.67	△	⊙		⊙	⊙	⊙	△	△	⊙	△
Easy access to logs and trends	1	3.33		△	⊙	⊙	△	⊙	⊙	⊙	⊙	⊙
Similar to previous pump design	1	3.33										
Importance			537	410	47	387	264	437	363	403	157	53
Importance %			17.5	13.4	1.5	12.6	8.6	14.3	11.9	13.2	5.13	1.73

Figure 20 HOQ/QFD showing matric of security vs. usability for artificial pancreas system

Findings

In this example, based on importance % score, top 3 security attributes that needs to be taken care of are the following:

1. Physical security
2. Transport security
3. Remote control

This top 3 priority matches with a previous casual attack analysis we conducted for the target system in chapter 5 (Findings and Related Security Attributes). Thus, this proof of concept for the proposed exploratory QFD is verified to work well at least in this scenario.

How to take care of those security requirements are already discussed in previous section- See Table 8 Final set of SSS questions and guidance
This kind of collaborative exercise can help to get to informed consensus across various stakeholders.

It often results in a prioritization which is highly unexpected and different from the conventional wisdom held by the company and many of the participants before engaging in QFD, thus stretching the team's thinking as to which activities are most critical toward creating a winning product or service.[77]

Product managers can use this matric as a template where the identified security requirements are pre-filled. Once they have set of proposed product usability features, they can fill up the left side and then map out the relationship to prioritize security needs accordingly.

Future Work

This introduction of new matric is exploratory. Further study is needed to find ways to standardize correlations within security requirements. This can help with mitigating risk by establishing alternative security methods. For example, if in certain system, it is not possible to patch but placing a firewall is possible, the desired result can be same although security method applied is different.

Conclusion and Next Steps

The main contribution of this work was to discover that there is a gap between the balance of usability and security specifically in artificial pancreas system. This was done by decomposing and analyzing usability and security into sub attributes. To that end, this work is a first step in an attempt to close that gap by theorizing that existing multi-dimensional tools and concepts from the fields of usability and security can be combined to strike that balance. As part of future work, I would like to explore if this concept can be expanded to any IoT device. The next step is to validate and thus generalize this work from a post positivist perspective, which leads directly to the following hypotheses:

Hypothesis: IoT products can be better optimized for security and usability, if the relationship between functional usability feature and corresponding security concern is clearly understood.

Defeating Cybercrime by creating secure, usable systems has proven to be a challenge in recent years. One of the other deficiencies that has led to this globally-recognized issue is the lack of a standard scoring system that considers both security and usability within system design. The future contribution of this work is to consider how this single metric, defined by in- formation security experts, usability experts, and representative users, can be used to assess the security and usability of a system, concentrating on IoT devices.

Reflectivity

Looking back, I would have liked to generalize my study for overall IoT devices to begin with and add more variety of people in my interviews. Instead of just interviewing security experts, it would have great if I could have talked to some IoT vendors to validate the assumptions. Also, talking to end consumers of these IoT device could have provided information like how culture, gender and other human behavior may affect security of devices.

Implications

The immediate implication of this study is the simple applications created for IoT vendors, specially product managers and developers of IoT startups who can't afford to hire security experts in their team. They can still test their design for security resiliency and follow some simple guidelines to avoid introducing unanticipated risk in their products. Future work to combine security and usability score into a unified metric can help end consumers make informed decision while buying such products. For example, like nutrient contents displayed in food packages, security usability score can be published for IoT devices. Another area where this can be used is governing bodies which can regulate IoT device quality based on its security-usability score and set up minimum standards.

References

1 Panetta, Contributor: Kasey. "Gartner's Top 10 Security Predictions 2016." Smarter With Gartner. N.p., 13 Dec. 2016. Web. http://www.gartner.com/smarterwithgartner/top-10-security-predictions-2016/ 30 Mar. 2017.

2 Cooper, Alan. The inmates are running the asylum:[Why high-tech products drive us crazy and how to restore the sanity]. Indianapolis, IN, USA: Sams, 2004.

3 Bhattarai, Ranjan, Ger Joyce, and Saurabh Dutta. "Information Security Application Design: Understanding Your Users." Lecture Notes in Computer Science Human Aspects of Information Security, Privacy, and Trust (2016): 103-13.

4 Seffah, Ahmed, Mohammad Donyaee, Rex B. Kline, and Harkirat K. Padda. "Usability measurement and metrics: A consolidated model." Software Quality Journal 14.2 (2006): 159-78.

5 Sousanis , John; "World Vehicle Population Tops 1 Billion Units," Wards Auto, Aug. 5, 2011. Web. http://wardsauto.com/news-analysis/world-vehicle-population-tops-1-billion-units, 30 Mar. 2017.

6 Arth, Michael; "New Pedestrianism: A Bridge to the Future," Carbusters Magazine, 2008. Web. https://web.archive.org/web/20091026014132/http://www.carbusters.org/magazine/33/feature3.html, 30 Mar. 2017.

7 "Impacts of the LightSquared Network on Federal Science Activities," Testimony of the Honorable Peter H. Appel, U.S. House of Representatives Committee on Science, Space and Technology, Sept. 8, 2011. Web. http://science.house.gov/sites/republicans.science.house.gov/files/documents/hearings/090811_%20Appel.pdf, 30 Mar. 2017.

8 "Greatbatch, W., & Holmes, C. F. (1991). History of implantable devices. IEEE Engineering in Medicine and Biology Magazine, 10(3), 38-41."

9 "Gartner's 2016 Hype Cycle for Emerging Technologies Identifies Three Key Trends That Organizations Must Track to Gain Competitive Advantage." Gartner. N.p., n.d. Web. 05 Apr. 2017.

10 "Internet of Things Startups." AngelList. N.p., n.d. Web. 05 Apr. 2017.

[11] Radcliff, Jay; “[Lecture Name].” [Location of Conference], Black Hat Conference. date of lecture. Lecture.

[12] Lam, Anthony, José Fernandez, and Richard Frank. "Cyberterrorists Bringing Down Airplanes: Will it Happen Soon?." 12th International Conference on Cyber Warfare and Security 2017 Proceedings.

[13] Khandelwal, Swati; “World’s Largest 1 Tbps DDoS Attack Launched from 152,000 Hacked Smart Devices,” Hacker News, Sept. 27, 2016. Web. http://thehackernews.com/2016/09/ddos-attack-iot.html. 30 Mar. 2017.

[14] "McDonald's Case Study - Amazon Web Services (AWS)." Amazon Web Services, Inc. N.p., n.d. Web. 23 Apr. 2017.

[15] "CAPTCHA: Telling Humans and Computers Apart Automatically." The Official CAPTCHA Site. N.p., n.d. Web. 03 May 2017.

[16] Garfinkel, Simson L, “Design Principles and Patterns for Computer Systems That Are Simultaneously Secure and Usable” (PhD Diss., Massachusetts Institute of Technology, 2005). https://simson.net/thesis/

[17] Hope, Paco; “Beware of these IoT Designs with Security Flaws,” Tech Republic, March 4, 2016. Web. http://www.techrepublic.com/article/beware-of-these-iot-designs-with-security-flaws/. 30 Mar. 2017.

[18] Chowdhury, Maksudul A., Janet Light, and William McIver. "A framework for continuous authentication in ubiquitous environments." Wireless Communication and Sensor Networks (WCSN), 2010 Sixth International Conference on. IEEE, 2010.

[19] "To CAPTCHA or Not to CAPTCHA." Distil Networks. N.p., n.d. Web. 07 Apr. 2017.

[20] Yan, J., El Ahmad, A. S.: Usability of CAPTCHAs or usability issues in CAPTCHA de- sign. In: Proceedings of the 4th symposium on Usable privacy and security, pp. 44-52. ACM (2008).

[21] O'Gorman, Lawrence. "Comparing passwords, tokens, and biometrics for user authentication." Proceedings of the IEEE 91.12 (2003): 2021-2040.

[22] Gunson, N, Marshall, D, Morton, H & Jack, M 2011, “User perceptions of security and usability of single-factor and two-factor authentication in automated telephone

banking," Computers & Security, vol 30, no. 4, pp. 208-220.
DOI: 10.1016/j.cose.2010.12.001

23 Hager, Creighton T., and Scott F. MidKiff. "An analysis of Bluetooth security vulnerabilities." Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE. Vol. 3. IEEE, 2003.

24 Freeman, R. Edward. Strategic Management: a Stakeholder Approach. Boston : Pitman, 1984. Print.

25 Feng, W. & Crawley, E. F.. "Stakeholder Value Network Analysis for Large Oil and Gas Projects." Research Report, Engineering Systems Division. Cambridge, MA: Massachusetts Institute of Technology. (2008)

26 Kano, N., N. Seraku, F. Takahashi and S. Tsuji: "Attractive Quality and Must-be Quality", Hinshitsu. The Journal of the Japanese Society for Quality Control, (April 1984), pp. 39 -48.

27 Morales, Carlos O.(2015) "Class lecture and notes from visiting faculty"

28 Juristo, Natalia, Ana Moreno, and Maria-Isabel Sanchez-Segura. "Guidelines for Eliciting Usability Functionalities." IEEE Transactions on Software Engineering 33.11 (2007): 744-58. Web.

29 Cysneiros, L.m., V.m. Werneck, and A. Kushniruk. "Reusable knowledge for satisficing usability requirements." 13th IEEE International Conference on Requirements Engineering (RE'05) (2005): n. pag. Web.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.10&rep=rep1&type=pdf

30 Nielsen, Jakob. "Usability Heuristics." Usability Engineering (1993): 115-63. Web.

31 Constantine, Larry L., and Lucy A. D. Lockwood. Software for use: a practical guide to the models and methods of usage-centered design;. Boston: Addison Wesley, 1999. Print.

32 D. Hix and H.R. Hartson, Developing User Interfaces: Ensuring Usability through Product and Process. John Wiley and Sons, 1993.

33 International Organization for Standardization/International Electrotechnical Commission. "ISO/IEC 9126." *Information Technology, Software Product Evaluation, Quality Characteristics and Guidelines for their Use* (1991).

34 "802.5, 1998 Edition (ISO/IEC 8802-5:1998)." (1998): n. pag. Web.

35 IEEE Standards Coordinating Committee. "IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990). Los Alamitos." CA: IEEE Computer Society (1990).

36 Seffah, Ahmed, Mohammad Donyaee, Rex B. Kline, and Harkirat K. Padda. "Usability measurement and metrics: A consolidated model." Software Quality Journal 14.2 (2006): 159-78. Web.

37 Kuzel, A. Sampling in qualitative inquiry. In Doing qualitative research, ed. B. Crabtree and W. Miller, 31–44. Newbury Park, CA: Sage. (1992). print

38 Romney, A., W. Batchelder, and S. Weller.. Culture as consensus:Atheory of culture and informant accuracy. American Anthropologist 88:313–38. (1986). print

39 Creswell, J. W. Research Design: Qualitative, Quantitative, and Mixed Methods Approaches (Fourth Edi). SAGE Publications.(2013). print

40 Golafshani, N. Understanding Reliability and Validity in Qualitative Research. The Qualitative Report, 8(4), 597-606 (2003). Print

41 Braz, Christina, Ahmed Seffah, and David M'Raihi. "Designing a Trade-Off Between Usability and Security: A Metrics Based-Model." Lecture Notes in Computer Science Human-Computer Interaction – INTERACT 2007 (2007): 114-26. Web.

42 Babar, Sachin, Parikshit Mahalle, Antonietta Stango, Neeli Prasad, and Ramjee Prasad. "Proposed Security Model and Threat Taxonomy for the Internet of Things (IoT)." Recent Trends in Network Security and Applications Communications in Computer and Information Science (2010): 420-29. Web.

43 Alam, Sarfraz, Mohammad M. R. Chowdhury, and Josef Noll. "Interoperability of Security-Enabled Internet of Things." Wireless Personal Communications 61.3 (2011): 567-86. Web.

44 Riahi, Arbia, Yacine Challal, Enrico Natalizio, Zied Chtourou, and Abdelmadjid Bouabdallah. "A Systemic Approach for IoT Security." 2013 IEEE International Conference on Distributed Computing in Sensor Systems (2013): n. pag. Web.

45 "Main Page." OWASP. N.p., n.d. Web. 07 Apr. 2017.

46 Rashid, Fahmida Y., "Swagger stumbles: Flaw enables remote code execution." InfoWorld. InfoWorld, 27 June 2016. Web. 30 Mar. 2017. <http://www.infoworld.com/article/3088569/security/swagger-stumbles-flaw-enables-remote-code-execution.html>.

47 "Input Validation Cheat Sheet." Input Validation Cheat Sheet - OWASP. N.p., n.d. Web. 02 Apr. 2017.

48 "Data Validation." Data Validation - OWASP. N.p., n.d. Web. 02 Apr. 2017.

49 "Welcome To Trusted Computing Group." Trusted Computing Group. N.p., n.d. Web. 04 May 2017.

50 Center for Devices and Radiological Health. "Artificial Pancreas Device System - What is the pancreas? What is an artificial pancreas device system?" U S Food and Drug Administration Home Page. Center for Devices and Radiological Health, n.d. Web. 05 May 2017.

51 J. Radcliffe, “Hacking Medical Devices for Fun and insulin: Breaking the Human SCADA System. “Blackhat conference” (2011): Web.https://media.blackhat.com/bh-us-11/Radcliffe/BH_US_11_Radcliffe_Hacking_Medical_Devices_Slides.pdf

52 Stilgherrian (CSO Online). Lethal medical device hack taken to next level: attacker sniffs insulin pump ID, delivers fatal dose. October 21, 2011. Available at: http://www.cso.com.au/article/404909/lethal_medical_device_hack_taken_next_level/. Accessed March 30, 2015. Google Scholar

53 Brooke, John. "SUS-A quick and dirty usability scale." Usability evaluation in industry 189.194 (1996): 4-7.

54 Affairs, Assistant Secretary for Public. "System Usability Scale (SUS)." Usability.gov. Department of Health and Human Services, 06 Sept. 2013. Web. 01 Apr. 2017.

55 Callegati, Franco, Walter Cerroni, and Marco Ramilli. "Man-in-the-Middle Attack to the HTTPS Protocol." IEEE Security & Privacy Magazine 7.1 (2009): 78-81. Web.

[56] Hager, C.t., and S.f. Midkiff. "An analysis of Bluetooth security vulnerabilities." 2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003. (n.d.): n. pag. Web.

[57]"Defense Security Service Home Page." Defense Security Service (DSS). N.p., n.d. Web. 07 Apr. 2017. <http://www.dss.mil/documents/rmf/DSS_Assessment_and_Authorization_Process_Manual-August242016.pdf>.

[58] "Federal Communications Commission." Federal Communications Commission. N.p., 29 Mar. 2017. Web. https://transition.fcc.gov/cyber/cyberplanner.pdf 30 Mar. 2017.

[59] "Chapter 7 Access Control, Authentication, and Encryption." Chapter 7 Access Control, Authentication, and Encryption. N.p., 10 Jan. 2005. Web. 30 Mar. 2017. <https://docs.oracle.com/cd/E19901-01/817-7607/aci.html>.

[60] "Don't trust user input." Don't trust user input - OWASP. N.p., n.d. Web. 30 Mar. 2017. <https://www.owasp.org/index.php/Don%27t_trust_user_input>.

[61] Venkatraman, K., J. Vijay Daniel, and G. Murugaboopathi. "Various Attacks in Wireless Sensor Network: Survey." International Journal of Science and Research (IJSR) 3.1 (2013): 2008-011. Web. 30 Mar. 2017. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.686.8840&rep=rep1&type=pdf>.

[62] Ye, Tao, Darryl Veitch, and Jean Bolot. "Improving wireless security through network diversity." ACM SIGCOMM Computer Communication Review 39.1 (2008): 34. Web.

[63] Ukil, Arijit, Soma Bandyopadhyay, and Arpan Pal. "IoT-Privacy: To be private or not to be private." 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (2014): n. pag. Web.

[64] Soldatos, John, Nikos Kefalakis, Manfred Hauswirth, Martin Serrano, Jean-Paul Calbimonte, Mehdi Riahi, Karl Aberer, Prem Prakash Jayaraman, Arkady Zaslavsky, Ivana Podnar Žarko, Lea Skorin-Kapov, and Reinhard Herzog. "OpenIoT: Open Source Internet-of-Things in the Cloud." Interoperability and Open-Source Solutions for the Internet of Things Lecture Notes in Computer Science (2015): 13-25. Web.

[65] Bethencourt, John, Amit Sahai, and Brent Waters. "Ciphertext-Policy Attribute-Based Encryption." 2007 IEEE Symposium on Security and Privacy (SP '07) (2007): n. pag. Web. 2017. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.475.2538&rep=rep1&type=pdf>.

[66] Colp, Patrick, Jiawen Zhang, James Gleeson, Sahil Suneja, Eyal De Lara, Himanshu Raj, Stefan Saroiu, and Alec Wolman. "Protecting Data on Smartphones and Tablets from Memory Attacks." ACM SIGPLAN Notices 50.4 (2015): 177-89. Web.

[67] Wolthusen, S.d. "Security policy enforcement at the file system level in the Windows NT operating system family." Seventeenth Annual Computer Security Applications Conference (n.d.): n. pag. Web.

[68] Schneier, Bruce, and John Kelsey. "Secure audit logs to support computer forensics." ACM Transactions on Information and System Security 2.2 (1999): 159-76. Web. <https://www.schneier.com/academic/paperfiles/paper-auditlogs.pdf>.

[69] Bauspiess, Fritz, and Frank Damm. "Requirements for cryptographic hash functions." Computers & Security 11.5 (1992): 427-37. Web.

[70] Mercuri, Rebecca T., and Peter G. Neumann. "Security by obscurity." Communications of the ACM 46.11 (2003): 160.

[71] Courtois, Nicolas T. "The dark side of security by obscurity and cloning Mifare Classic rail and building passes, anywhere, anytime." (2009).

[72] Arora, Ashish, Rahul Telang, and Hao Xu. "Optimal policy for software vulnerability disclosure." Management Science 54.4 (2008): 642-656.

[73] Arora, Ashish, et al. "Impact of vulnerability disclosure and patch availability-an empirical analysis." Third Workshop on the Economics of Information Security. Vol. 24. 2004.

[74] Dutta, Saurabh, Stuart Madnick, and Ger Joyce. "SecureUse: Balancing Security and Usability Within System Design." International Conference on Human-Computer Interaction. Springer International Publishing, 2016.

[75] Gaskin, Steven P., Abbie Griffin, John R. Hauser, Gerald M. Katz, and Robert L. Klein. "Voice of the Customer." Wiley International Encyclopedia of Marketing - Gaskin - Wiley Online Library. John Wiley & Sons, Ltd, 15 Dec. 2010. Web. 03 Apr. 2017.

[76] Pugh, Stuart, and Don Clausing. Creating innovtive products using total design: the living legacy of Stuart Pugh. Addison-Wesley Longman Publishing Co., Inc., 1996.

[77] Hauser, John R., Abbie Griffin, Robert L. Klein, Gerald M. Katz, and Steven P. Gaskin. "Quality Function Deployment (QFD)." Wiley International Encyclopedia of Marketing (2010): n. pag. Web.