# A Hazard Analysis Technique for the Internet of Things (IoT) and Mobile

Gregory Pope, CSQE

Lawrence Livermore National Laboratory, February 21, 2017

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

# Table of Contents

# Table of Figures

## Table of Tables

## 1. Introduction

Internet of Things (IoT) is the internetworking of physical devices (also referred to as "connected devices" and "smart devices"), vehicles, buildings and other items—embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data.[1,2,3]

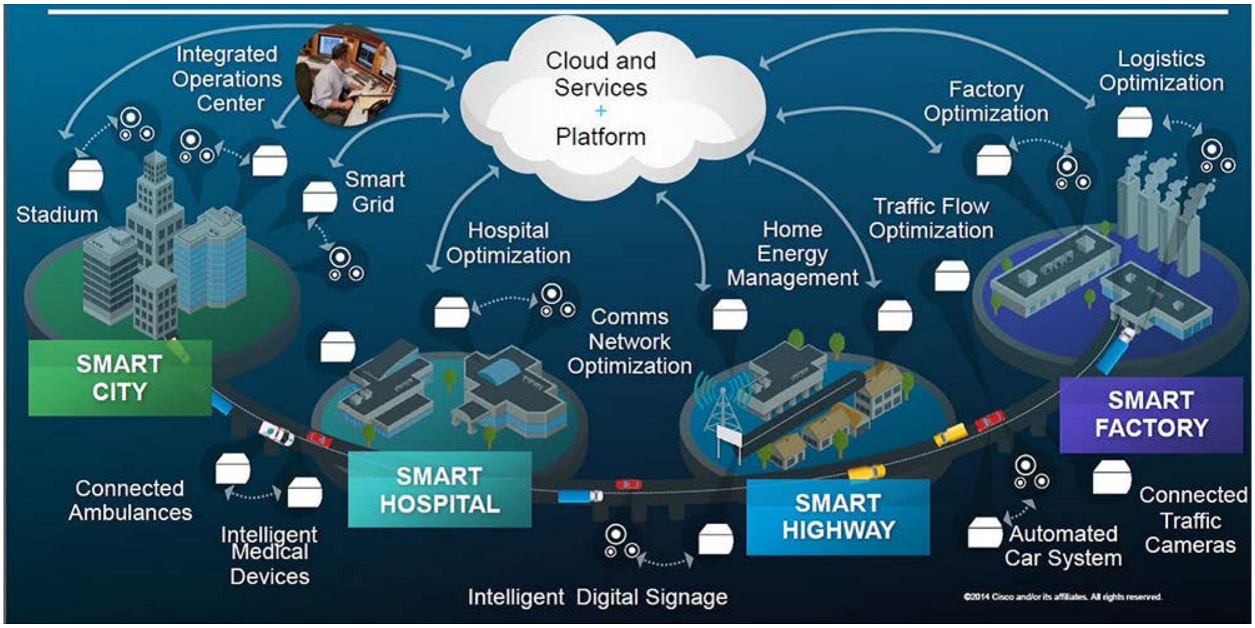

Integrated Operations Center
Cloud and Services + Platform
Logistics Optimization
Factory Optimization
Stadium
Smart Grid
Hospital Optimization
Home Energy Management
Traffic Flow Optimization
Comms Network Optimization
SMART CITY
SMART FACTORY
SMART HOSPITAL
SMART HIGHWAY
Connected Ambulances
Intelligent Medical Devices
Intelligent Digital Signage
Automated Car System
Connected Traffic Cameras
©2014 Cisco and/or its affiliates. All rights reserved.

Figure 1 - Smart Cities (Source Cisco)

The promise of the Internet of Things (IoT) includes the transformation of increased physical world activities into a virtual digital world. It is the next step in the proliferation of the internet which started with the first wave, networking of personal and business computers together, then expanded in the second wave to include mobile smart phones and tablets, and now is expanding into a third wave which ties together a wide range of physical devices and biological entities. Many devices in homes are already connected to the internet such as smart TVs, smart water meters, smart alarm systems, so the IoT is really an extension of making physical devices “smart” such as household appliances, transportation devices such as our cars, and biological entities such as animals and humans. Imagine our home consumables being monitored and replacements ordered automatically, our cars driving themselves to work and finding and reserving the closet parking space. Our health being monitored 24/7 and alerts issued and medical appointments made when needed. The concept of assigning physical devices an IP address on the internet is not really a new concept; it has been rapidly growing for years especially in medical, transportation, utilities, and financial industries. Smart debit and credit cards are more commonly used than cash. Retailers update inventories and determine item pricing as purchases are made. Patient vital signs can be sent from remote locations, including wearables.

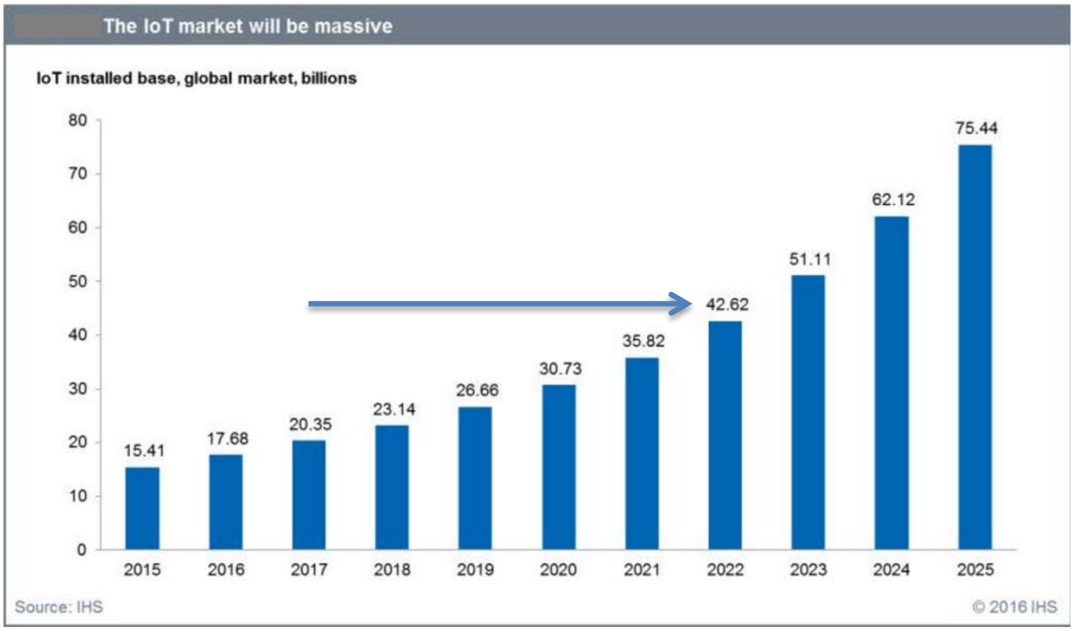
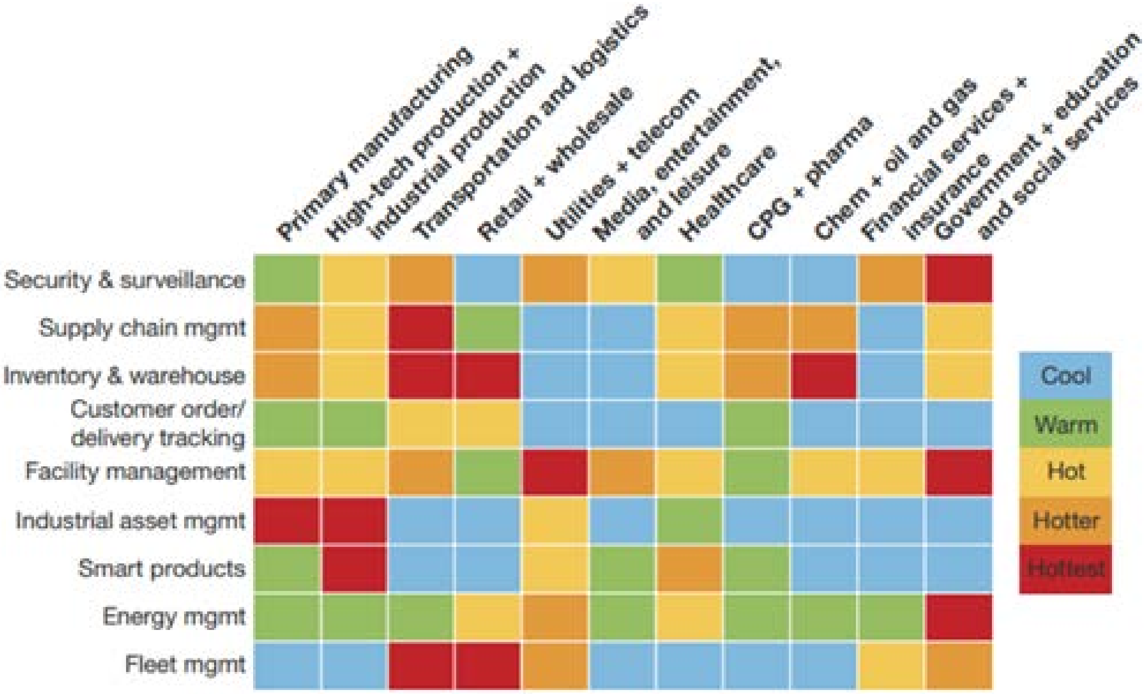


Figure 2 - Growth of IoT[4]

Like most emergent technologies, there are bound to be growing pains with IoT. As shown in figure 2, IoT is predicted to double from 2017 to 2022. The industries that are predicted to be most affected by the growth of IoT are depicted in the hot spot chart (figure 3) compiled by Forrester Research.[5]



| | Primary manufacturing | High-tech production + industrial production | Transportation and logistics | Retail + wholesale | Utilities + telecom | Media, entertainment, and leisure | Healthcare | CPG + pharma | Chem + oil and gas | Financial services + insurance | Government + education and social services |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Security & surveillance | 🟩 | 🟨 | 🟧 | 🟦 | 🟧 | 🟨 | 🟩 | 🟦 | 🟦 | 🟧 | 🟥 |
| Supply chain mgmt | 🟧 | 🟨 | 🟥 | 🟩 | 🟦 | 🟦 | 🟨 | 🟧 | 🟧 | 🟦 | 🟨 |
| Inventory & warehouse | 🟧 | 🟨 | 🟥 | 🟥 | 🟦 | 🟦 | 🟨 | 🟧 | 🟥 | 🟦 | 🟨 |
| Customer order/ delivery tracking | 🟩 | 🟩 | 🟨 | 🟨 | 🟦 | 🟦 | 🟦 | 🟩 | 🟦 | 🟦 | 🟦 |
| Facility management | 🟨 | 🟨 | 🟧 | 🟩 | 🟥 | 🟧 | 🟨 | 🟩 | 🟨 | 🟨 | 🟥 |
| Industrial asset mgmt | 🟥 | 🟥 | 🟦 | 🟦 | 🟨 | 🟦 | 🟩 | 🟦 | 🟦 | 🟦 | 🟦 |
| Smart products | 🟩 | 🟥 | 🟦 | 🟦 | 🟨 | 🟩 | 🟧 | 🟩 | 🟦 | 🟦 | 🟦 |
| Energy mgmt | 🟩 | 🟩 | 🟩 | 🟨 | 🟧 | 🟩 | 🟨 | 🟩 | 🟩 | 🟩 | 🟥 |
| Fleet mgmt | 🟦 | 🟦 | 🟥 | 🟥 | 🟧 | 🟦 | 🟦 | 🟦 | 🟦 | 🟨 | 🟧 |

Figure 3 - Heat Map of Key IoT Opportunities Varies by Industry and Application

As shown in figure 3, hot spots anticipated for IoT are in the industrial asset and fleet management sectors, inventory management, security, facility and energy management. Not really a surprise based on adoption of this technology to date. In the government sector the hottest anticipated areas are security and surveillance, facility management and energy management.

The value proposition for IoT of increased security, safety, and energy efficiency is already being realized by early adopters. Examples are;

1. Displaying and controlling home burglar alarms system from smart phones
2. Air-conditioning set to a minimum level or turned off when no motion is detected in the home for more than an hour
3. Air-conditioning can set to a minimum level or be turned off when the burglar alarm is set to away mode.
4. Verifying your driving teenager has returned home by viewing a drive way cam and motion detectors on your smart phone
5. Home carbon monoxide detector can disable gas heating system if CO threshold reached
6. While on vacation; lights, television, radios can be turned on and off to look occupied
7. Visitor can be recognized at the front doorway on a smartphone and/or recorded for playback
8. Household supplies can be reordered and shipped before running out
9. Self-driving cars and trucks offer a promise of traffic safety and energy conservation. Many newer vehicles can recognize road position and following distance already.

The IoT value proposition must be tempered by the risk of cyber security vulnerabilities and consequences of buggy software. The IoT examples given above have consequences of failure involving energy conservation, expenses, security, and safety. Therefore, the non-functional requirements for safety, security, and reliability of software used in the IoT components requires an effective hazard analysis.

This paper presents a hazard analysis technique especially developed for software and firmware that supports IoT and its vulnerability to cyber-attacks, misuse by those with ulterior motives, and the exploitation for increased profits by businesses. This paper will utilize literature searches, consulting with technical experts, and a relatively new hazard analysis technique, one especially developed for software intensive systems called Systemic Theoretic Process Analysis (STPA)[6]. The purpose of this white paper is to identify risks (or hazards for mission critical applications) for IoT in this third emergent stage of the internet so that mitigations can be applied before accidents, losses, or frustrations occur. A second purpose of this white paper is to evaluate the effectiveness of STPA as a hazard/risk analysis technique in the emerging IoT era.

## 2. Why Systemic Analytic Process Analysis (STPA)?

STPA was developed by Nancy Leveson, Professor of Aeronautics and Aerospace at MIT and discussed in both of her books: *Safeware, System Safety and Computers*, written in 1995, and *Engineering a Safer World*, written in 2011. In her books, she states “One cannot rely solely on hazard analysis methods developed for hardware when performing hazard analysis for software”

A literature search indicates that there is already ample concern about cyber security of the IoT[7]. It also revealed that many of the hazard and vulnerability analysis techniques, created originally for use with simple mechanical and analog control systems, are still in use today such as Root Cause Analysis (RCA) figure 4, Failure mode and Effects Analysis (FMEA) figure 5, and Fault Tree Analysis (FTA) figure 6, all of which were developed in the 1950's and early 1960's long before the internet and computers played a major role in society. Back then 32 thousand words of memory was a large software program and code was mostly used to solve engineering problems.

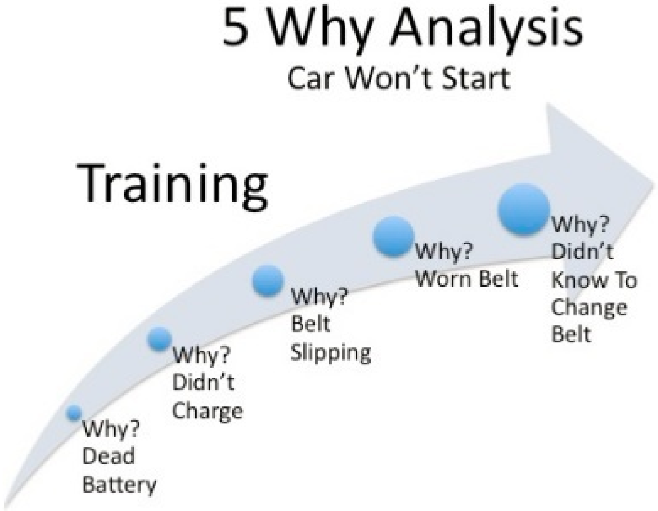
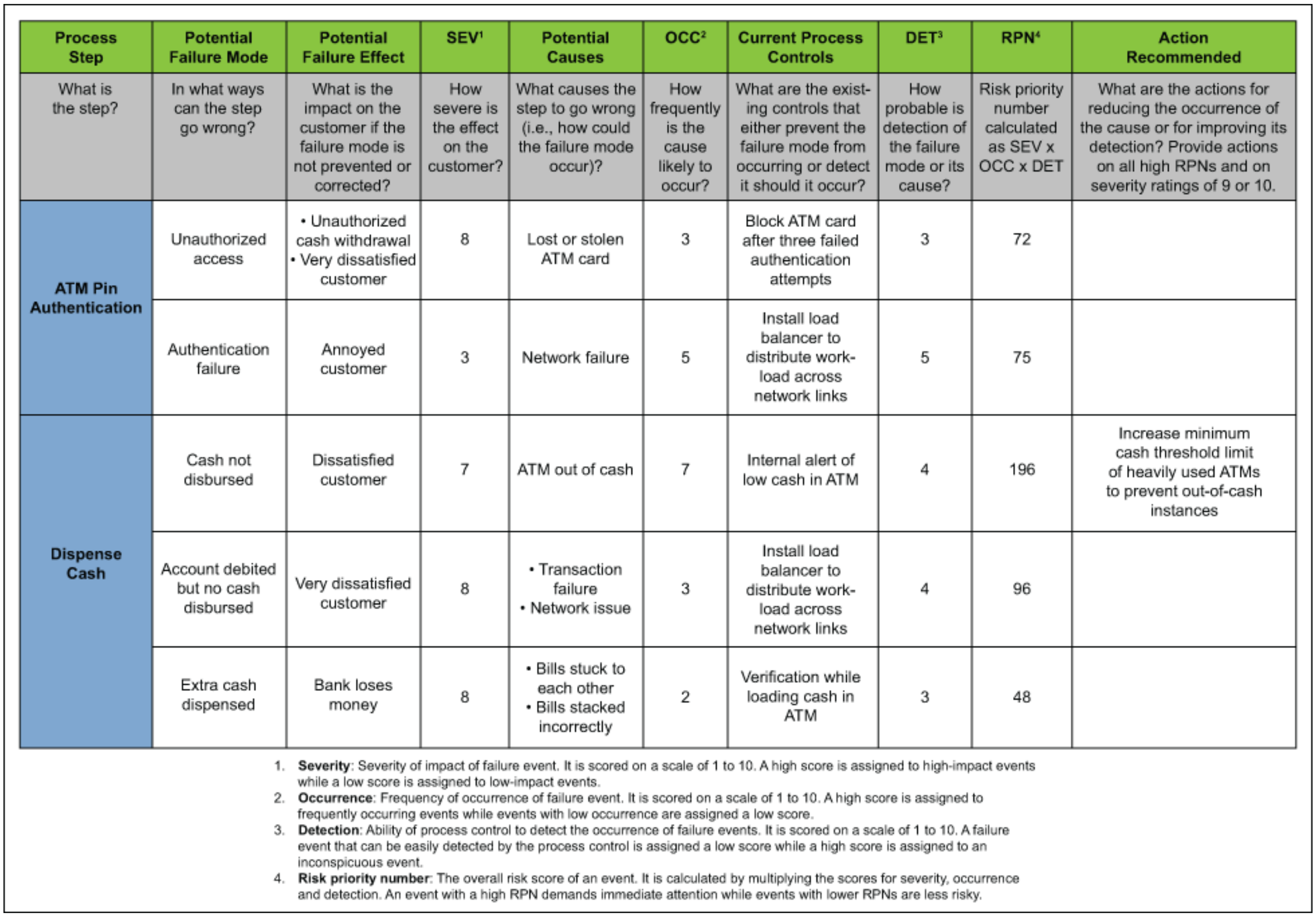


Figure 4 - Example Root Cause (Five Why) Analysis[8]



| Process Step | Potential Failure Mode | Potential Failure Effect | SEV[1] | Potential Causes | OCC[2] | Current Process Controls | DET[3] | RPN[4] | Action Recommended |
|---|---|---|---|---|---|---|---|---|---|
| What is the step? | In what ways can the step go wrong? | What is the impact on the customer if the failure mode is not prevented or corrected? | How severe is the effect on the customer? | What causes the step to go wrong (i.e., how could the failure mode occur)? | How frequently is the cause likely to occur? | What are the existing controls that either prevent the failure mode from occurring or detect it should it occur? | How probable is detection of the failure mode or its cause? | Risk priority number calculated as SEV x OCC x DET | What are the actions for reducing the occurrence of the cause or for improving its detection? Provide actions on all high RPNs and on severity ratings of 9 or 10. |
| ATM Pin Authentication | Unauthorized access | • Unauthorized cash withdrawal<br>• Very dissatisfied customer | 8 | Lost or stolen ATM card | 3 | Block ATM card after three failed authentication attempts | 3 | 72 | |
| | Authentication failure | Annoyed customer | 3 | Network failure | 5 | Install load balancer to distribute workload across network links | 5 | 75 | |
| Dispense Cash | Cash not disbursed | Dissatisfied customer | 7 | ATM out of cash | 7 | Internal alert of low cash in ATM | 4 | 196 | Increase minimum cash threshold limit of heavily used ATMs to prevent out-of-cash instances |
| | Account debited but no cash disbursed | Very dissatisfied customer | 8 | • Transaction failure<br>• Network issue | 3 | Install load balancer to distribute workload across network links | 4 | 96 | |
| | Extra cash dispensed | Bank loses money | 8 | • Bills stuck to each other<br>• Bills stacked incorrectly | 2 | Verification while loading cash in ATM | 3 | 48 | |

1. **Severity**: Severity of impact of failure event. It is scored on a scale of 1 to 10. A high score is assigned to high-impact events while a low score is assigned to low-impact events.
2. **Occurrence**: Frequency of occurrence of failure event. It is scored on a scale of 1 to 10. A high score is assigned to frequently occurring events while events with low occurrence are assigned a low score.
3. **Detection**: Ability of process control to detect the occurrence of failure events. It is scored on a scale of 1 to 10. A failure event that can be easily detected by the process control is assigned a low score while a high score is assigned to an inconspicuous event.
4. **Risk priority number**: The overall risk score of an event. It is calculated by multiplying the scores for severity, occurrence and detection. An event with a high RPN demands immediate attention while events with lower RPNs are less risky.

Figure 5 - Example Failure Modes and Effects Analysis[9]

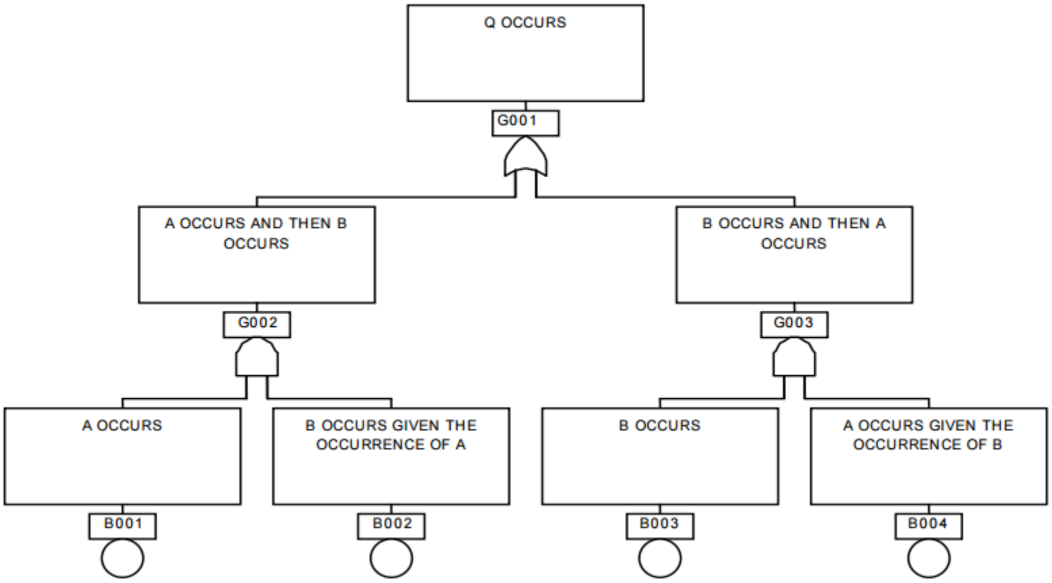

Q OCCURS
G001
A OCCURS AND THEN B OCCURS
G002
B OCCURS AND THEN A OCCURS
G003
A OCCURS
B001
B OCCURS GIVEN THE OCCURRENCE OF A
B002
B OCCURS
B003
A OCCURS GIVEN THE OCCURRENCE OF B
B004

Figure 6 - Example Fault Tree Analysis[10]

Today new cars have over 100 million line of code, airplanes over 14 million lines of code, smart phone operating systems over 12 million lines of code, and Google over 2 billion lines of code.[11] Software and the internet have become a major part of our everyday experiences. Software however fails differently than mechanical and electronic devices of the fifties and sixties. Whereas a mechanical or electronic device may experience infant mortality and then run correctly until it wears out, software gets better over time (wears in) as defects are found and fixed, with momentary failures spiking as new versions are released.  Figure 4 and 5 illustrate typical failure curves for hardware and software.

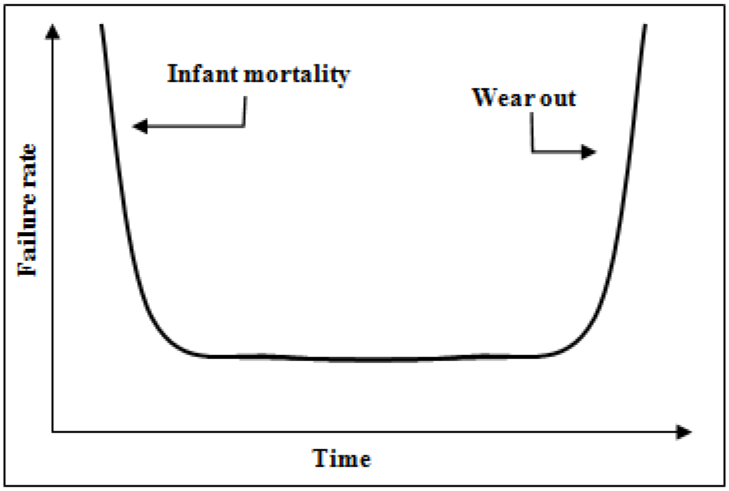


Figure 7 - Failure Curve for Hardware[12]

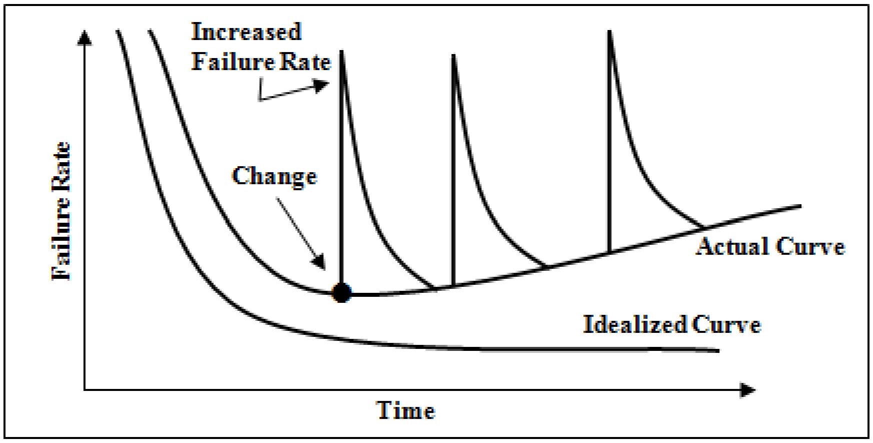


Figure 8 - Failure Curve for Software[13]

This author has analyzed several accidents involving systems that used software as a component. These accident summaries are included on a webpage for review. [14] The result of this author's analysis was that in these case studies the software did not stop working. Rather the accidents were caused by component interaction problems not explicitly covered in the requirements.  Compared to simple logic or mechanical relay controllers of the fifties and sixties, software typically has many more potentially undesired system interaction combinations that must be determined during the requirements through testing stages to be effectively mitigated. Before embarking on STPA of the IoT, a simplified example is given to demonstrate the STPA technique. The Figure 9 illustrates the five steps of STPA.

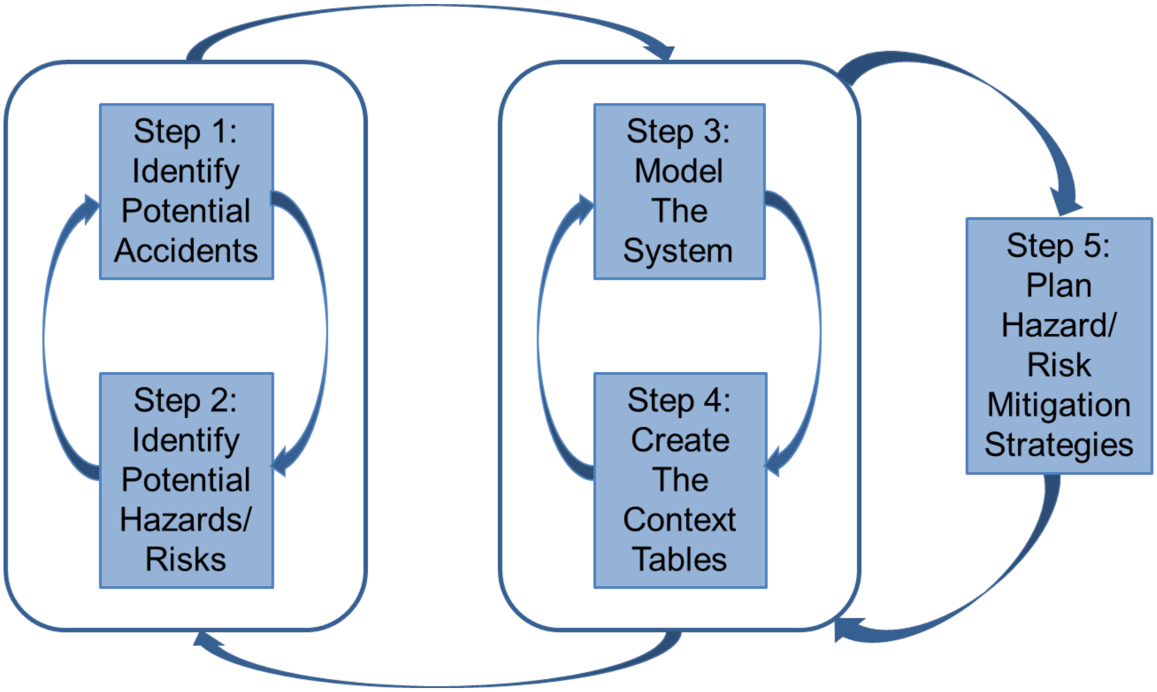

Step 1: Identify Potential Accidents
Step 2: Identify Potential Hazards/ Risks
Step 3: Model The System
Step 4: Create The Context Tables
Step 5: Plan Hazard/ Risk Mitigation Strategies

Figure 9 - STPA Overview

Step1 in the STPA process is to identify accident scenarios. For the purposes of cyber security, we will call these accidents attacks or hacks. After identifying accidents (or attacks), step 2 determines the potential hazards (or vulnerabilities) that could lead to the accident or attack. Step three requires modeling the system to be analyzed. The system can be a real-time system, as in this simple example, a non-real time system, or even an organization. Modeling at the simplest level is typically determining the interaction of a controlling device over a physical process by using sensors and actuators as shown in figure 10[15].

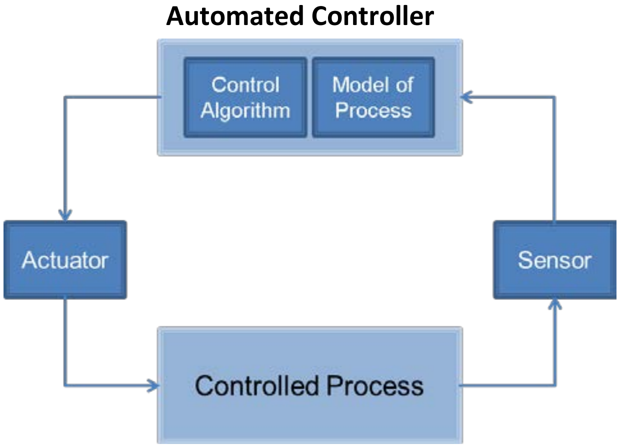


Figure 10 - STPA Control Structure Model

Note that the automated controlling device in figure 10 consists of a control algorithm and a model of the process being controlled. The states of the model of the process is determined by inputs from the sensor(s). Based on the model of the process the control algorithm commands actions using the actuator(s) to the controlled process. The automated controller can be implemented with software, firmware, PLC's, electronic circuits, or relays. If the sensor is not working properly, then the model of the physical device being controlled could be different than the actual state of the physical device. This difference could then lead the control algorithm to issue erroneous commands or issue correct commands at the wrong time. If the actuator is faulty and the sensor is working properly, the model of the controlled process could be correct and correct commands issued to the controlled process from the control algorithm will not get to the controlled process or get to the controlled process too late. Of course, the actuator and sensor could be working correctly, and a problem could occur if the model of the process or control algorithm is incorrect for the physical process being modeled. This has led to multiple accidents reusing software and assuming it will work correctly on the new physical device. When software controlled systems are involved in accidents it is most often one of the above scenarios that causes a hazard (or vulnerability) not the software failing. In fact, many accidents have involved high quality software doing exactly what is was designed to do, just doing it at the wrong time because the requirements were unclear.

To analyze the model shown in figure 10 four guide phrases are used. The guide phrases cover the different ways the model could cause a hazard or vulnerability. The guide phrases are:

1. A safety control action is not provided or is not followed.
2. An unsafe control action is provided.
3. A safety control action is provided too late or out of sequence.
4. A safety control action is stopped too soon or applied too long.

## 3. Example STPA Analysis

The simple sample problem chosen to demonstrate STPA is one that any of us have observed taking public transportation. Figure 11 shows a public transportation rail car. We would like to come up with a set of requirements for operating the doors on the rail car automatically.

Figure 11 - Public Transportation Rail Car with Automated Doors

The requirements for the automated doors are given as follows in figure 12:

1. Doors shall remain open at station when train is completely stopped and aligned with platform
2. Doors shall remain closed when train is moving
3. Doors shall remain open when someone is in the doorway
4. Doors shall open in an emergency

Figure 12 - Requirements for the automated Train Door

As with any high-level requirements, it may be prudent to analyze these requirements to see if any lower level requirements can be derived. For instance, the terms “doorway” and “person” may require some additional definition. A person can vary greatly in height, the doorway may extend beyond the physical door both inside and outside the rail car. Therefore, two derived requirements are:

Doorway: A volume of space extending from the top edge of the door frame to the floor of the car and 6 inches to either side. See figure 13:

Someone: A person or object that is detected within the doorway volume.

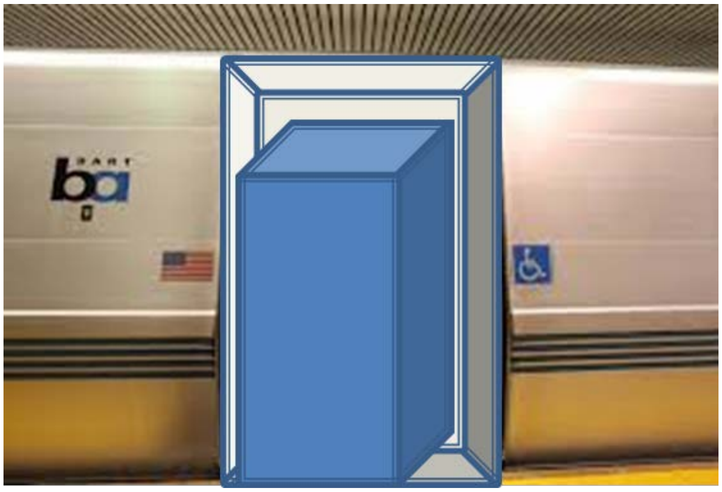


Figure 13 - Derived Requirements

These derived requirements would handle a person in a wheelchair or a large suitcase blocking the door. For the purposes of this example, the first step in STPA is to identify potential accidents:

A1 - Passenger falls out of moving train

A2 - Passengers not able to escape train during emergency

A3 - Passenger steps off stopped train not at platform

A4 – Doors close on passenger in doorway

The second step in STPA is to identify potential hazards that could cause the accidents lists above:

H-1 Doors open when the train is moving (A-1)

H-2 Doors close while person in the doorway (A-4)

H-3 Doors stuck closed during emergency (A-2)

H-4 Doors stuck open (A-1, A-3)

H-5 Doors open when not aligned to platform (A-3)

The third step in STPA is creating a model of the sample problem using the components discussed in figure 10, which are shown in figure 14. Note that other inputs have been added from the train to indicate motion, platform position, and emergency indicator.

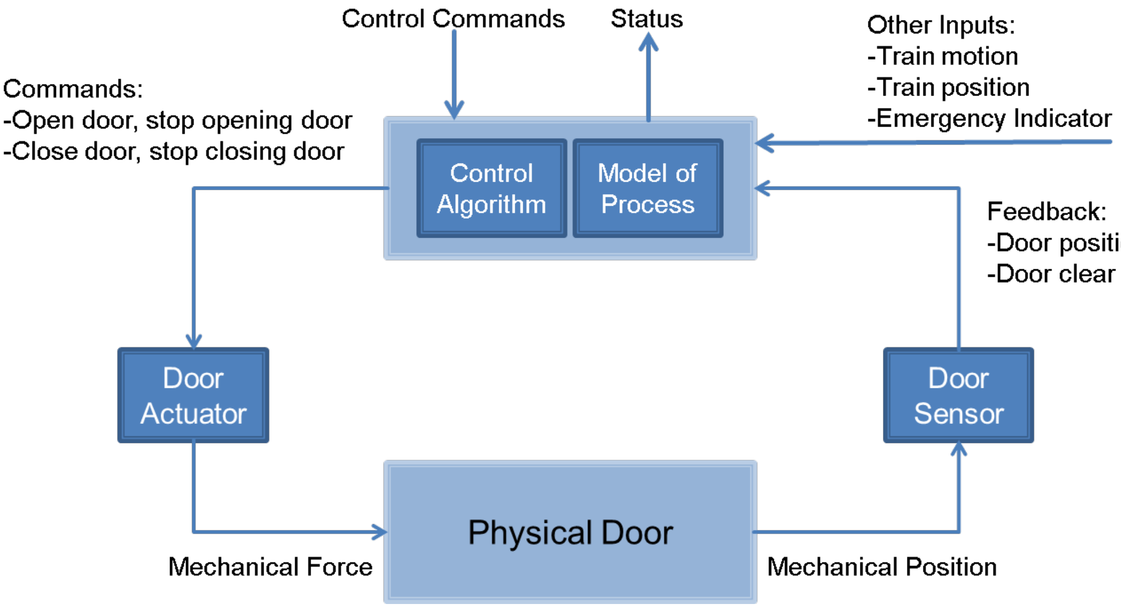
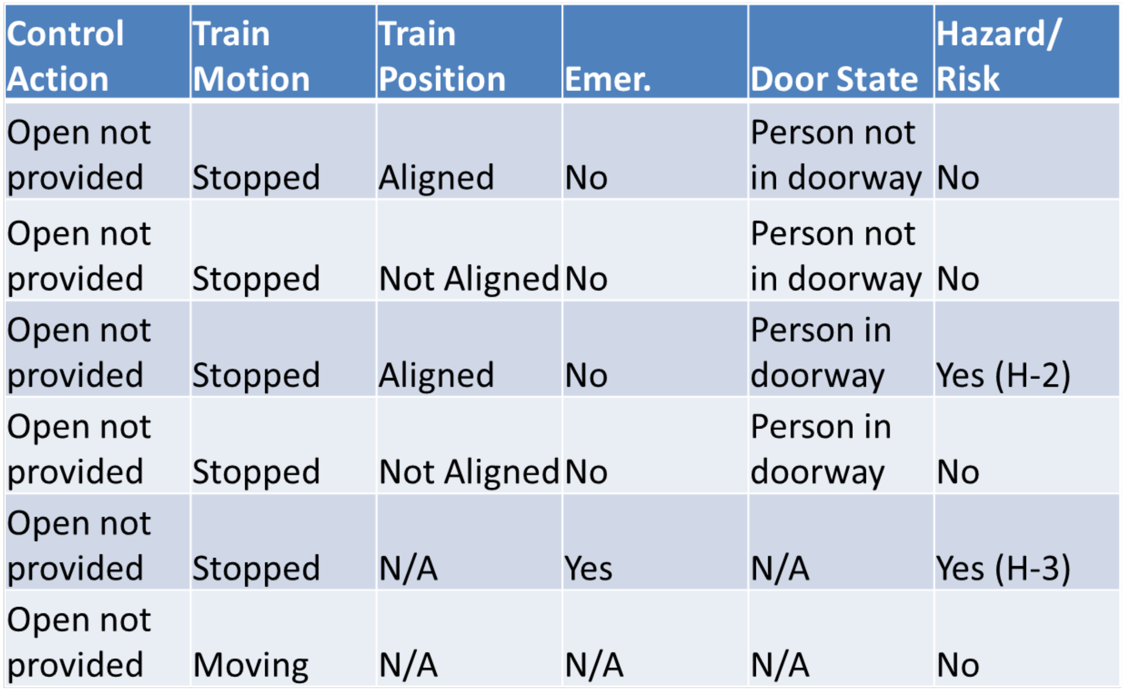


Figure 14 - Model of Rail Car Door Problem

To create the context table required in step 4 of STPA it is useful to apply guide phrases introduced earlier to the model shown in figure 14. This can lead to the discovery of additional hazards (or vulnerabilities.

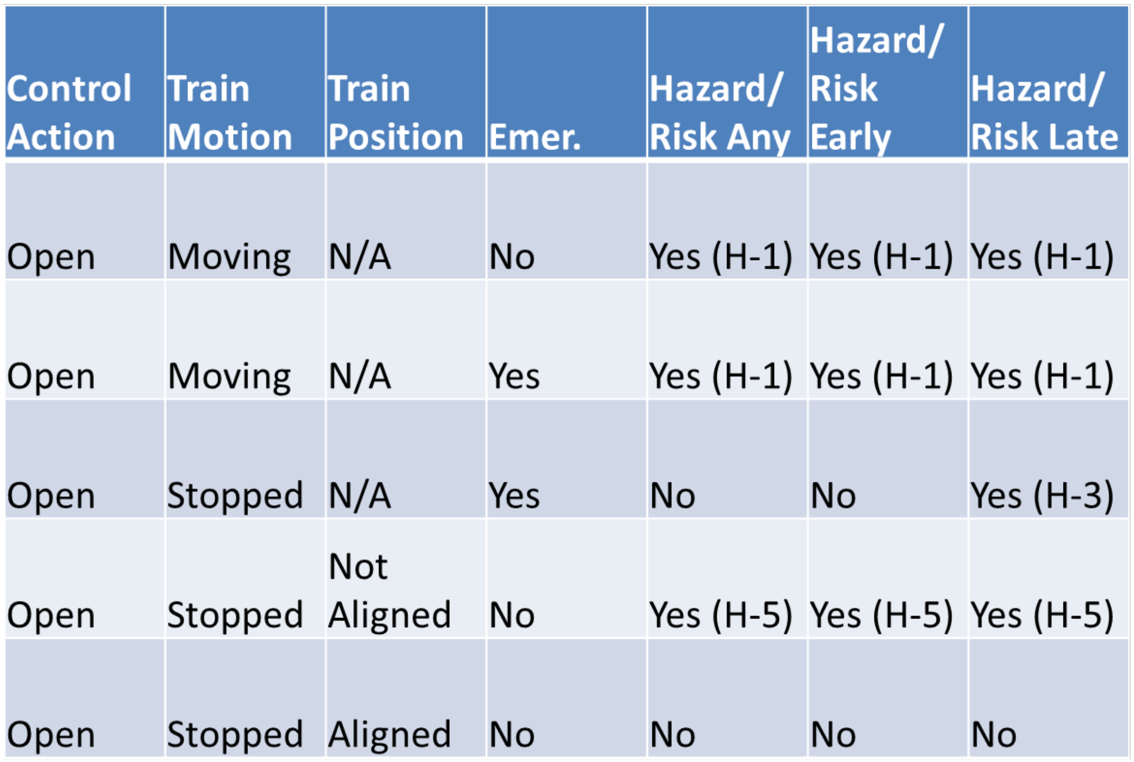
1. A safety control action is not provided or is not followed.
2. An unsafe control action is provided.
3. A safety control action is provided too late or out of sequence.
4. A safety control action is stopped too soon or applied too long.

Table 1 and 2 show context tables derived from applying the guide phrases to the model of the example problem. The context table is built using the combinations of control actions and system states. In this simple example, there is a relatively small number of control actions and system states. For larger systems, such as the IoT, it will be useful to decompose the system into sub-system models and components before applying the guide phrases and creating context tables. The context tables shown in tables 1 and 2 are high level or prefatory. This level may be sufficient for analysis. However, context tables can also be implemented on spreadsheets and databases facilitating automation of the context analysis.



| Control Action | Train Motion | Train Position | Emer. | Door State | Hazard/ Risk |
|---|---|---|---|---|---|
| Open not provided | Stopped | Aligned | No | Person not in doorway | No |
| Open not provided | Stopped | Not Aligned | No | Person not in doorway | No |
| Open not provided | Stopped | Aligned | No | Person in doorway | Yes (H-2) |
| Open not provided | Stopped | Not Aligned | No | Person in doorway | No |
| Open not provided | Stopped | N/A | Yes | N/A | Yes (H-3) |
| Open not provided | Moving | N/A | N/A | N/A | No |

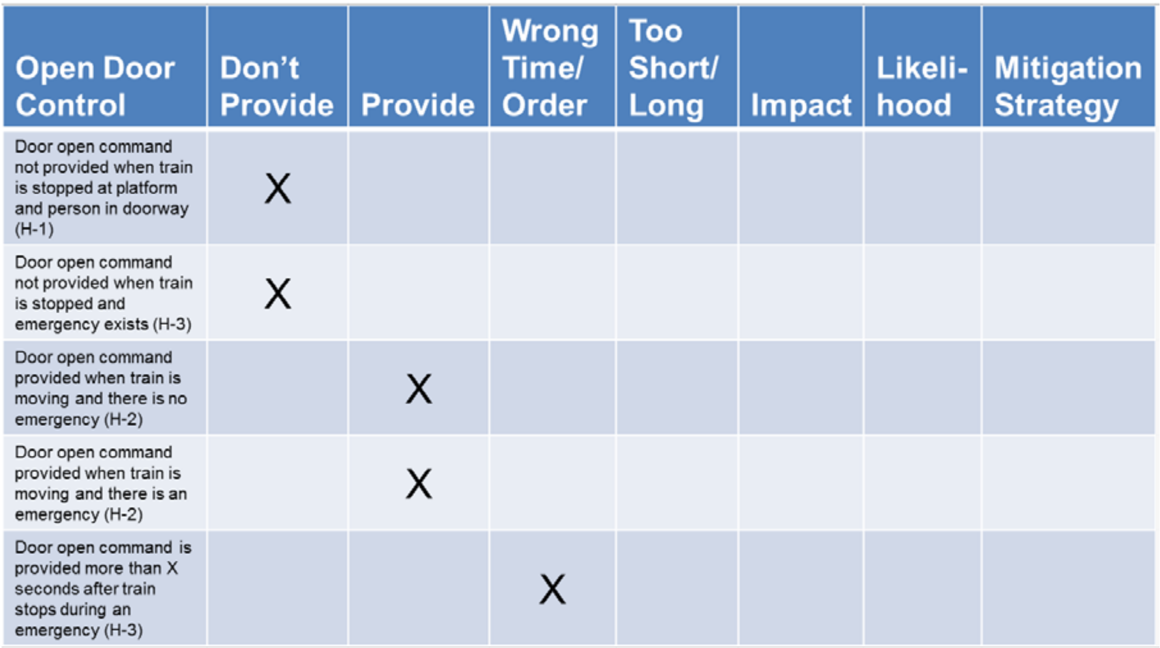
Table 1 - Context Table, Open Command Not Provided



| Control Action | Train Motion | Train Position | Emer. | Hazard/ Risk Any | Hazard/ Risk Early | Hazard/ Risk Late |
|---|---|---|---|---|---|---|
| Open | Moving | N/A | No | Yes (H-1) | Yes (H-1) | Yes (H-1) |
| Open | Moving | N/A | Yes | Yes (H-1) | Yes (H-1) | Yes (H-1) |
| Open | Stopped | N/A | Yes | No | No | Yes (H-3) |
| Open | Stopped | Not Aligned | No | Yes (H-5) | Yes (H-5) | Yes (H-5) |
| Open | Stopped | Aligned | No | No | No | No |

Table 2 - Context Table, Open Command Provided

Having completed the context tables for the example problem, step 5 of STPA requires a Hazard/Risk (or Vulnerability/Risk) mitigation strategy such as the one started in figure 15.



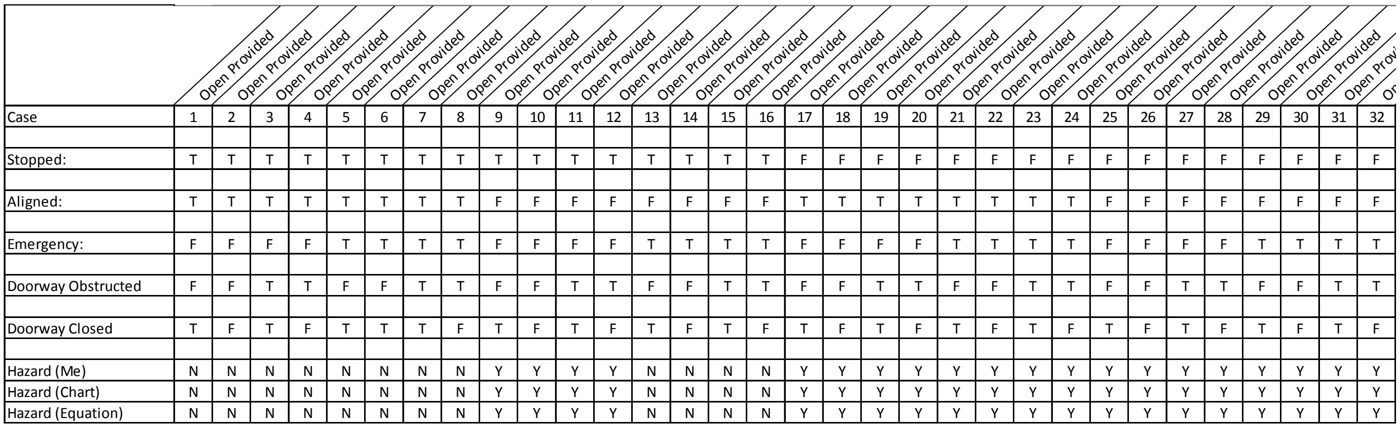
| Open Door Control | Don't Provide | Provide | Wrong Time/ Order | Too Short/ Long | Impact | Likeli-hood | Mitigation Strategy |
|---|---|---|---|---|---|---|---|
| Door open command not provided when train is stopped at platform and person in doorway (H-1) | X | | | | | | |
| Door open command not provided when train is stopped and emergency exists (H-3) | X | | | | | | |
| Door open command provided when train is moving and there is no emergency (H-2) | | X | | | | | |
| Door open command provided when train is moving and there is an emergency (H-2) | | X | | | | | |
| Door open command is provided more than X seconds after train stops during an emergency (H-3) | | | X | | | | |

Figure 15 - Hazard/Risk Mitigation Strategies

What has been presented so far is a primer on how STPA works for a train door example problem using prefatory STPA. However, STPA analysis may also be more exhaustive, even for the problem of a train door operation.

## 4. Exhaustive STPA Using a Spreadsheet and Visual Basic Macros

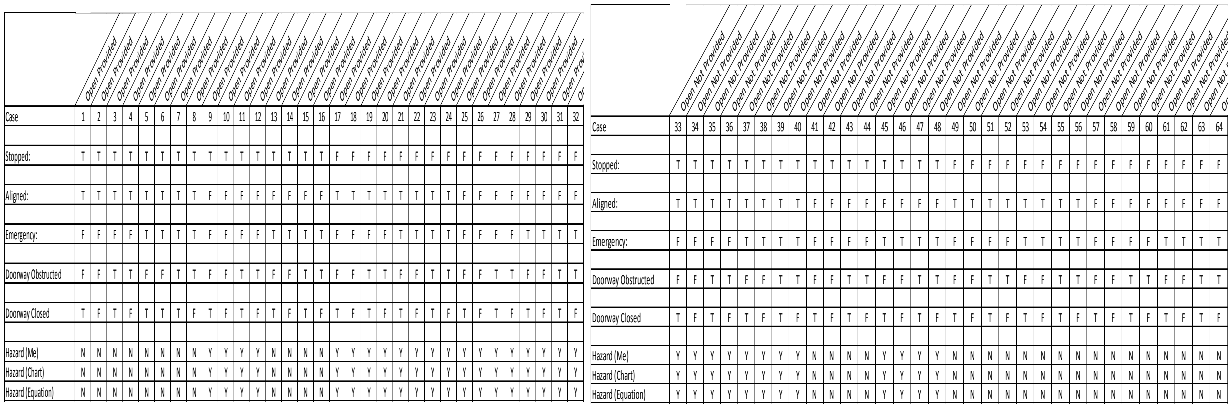
To give an example of a more exhaustive STPA of the train door problem a spreadsheet will be used. Using the Excel worksheet format all the combinations of train door states (32 combinatorial cases) can be analyzed against the guide phrases. Shown in figure 16 are the 32 cases in columns for the "Open Provided" guide phrase. The rows consist of the binary conditions which are derived from the requirements.



| | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provi | Or |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Case | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Stopped: | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Aligned: | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Emergency: | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Doorway Obstructed | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Doorway Closed | T | F | T | F | T | T | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hazard (Me) | N | N | N | N | N | N | N | N | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| Hazard (Chart) | N | N | N | N | N | N | N | N | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| Hazard (Equation) | N | N | N | N | N | N | N | N | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |

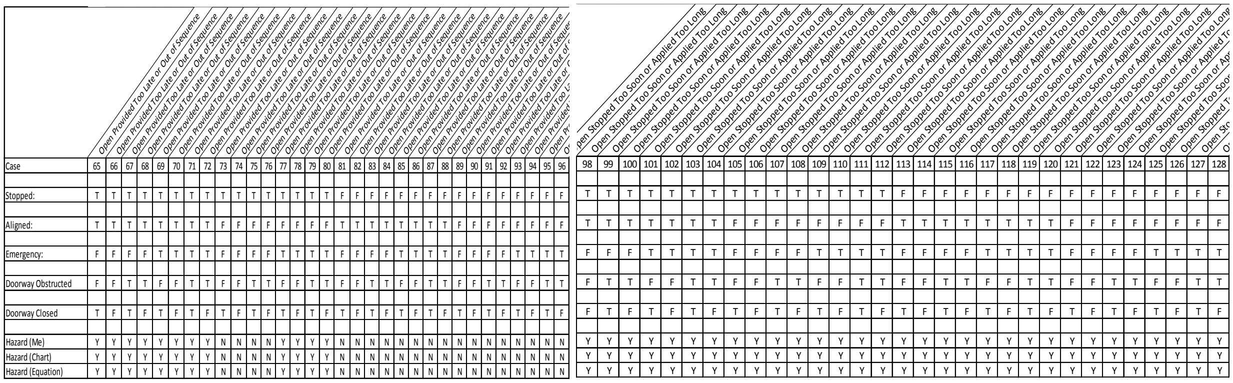
Figure 16 - Spreadsheet representation of Exhaustive STPA Cases for Analysis

An Excel workbook is then created with each guide phrase being a worksheet. With four guide phrases, there are 128 cases to analyze (32 per worksheet) for the simple train door problem. The reason the cases were divided into four separate worksheets was to be able to visually see all the cases on a single without scrolling. Figure 17 shows al four worksheets, one for each guide phrase.



| | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provi | Or |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Case | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Stopped: | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Aligned: | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Emergency: | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Doorway Obstructed | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Doorway Closed | T | F | T | F | T | T | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hazard (Me) | N | N | N | N | N | N | N | N | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| Hazard (Chart) | N | N | N | N | N | N | N | N | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| Hazard (Equation) | N | N | N | N | N | N | N | N | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |

| | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not Provided | Open Not | Or |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Case | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Stopped: | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Aligned: | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Emergency: | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Doorway Obstructed | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Doorway Closed | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hazard (Me) | Y | Y | Y | Y | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
| Hazard (Chart) | Y | Y | Y | Y | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
| Hazard (Equation) | Y | Y | Y | Y | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |



| | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence | Open Provided Too Late or Out of Sequence |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Case | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 |
| Stopped: | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| Aligned: | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F |
| Emergency: | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T |
| Doorway Obstructed | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T |
| Doorway Closed | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F |
| Hazard (Me) | Y | Y | Y | Y | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
| Hazard (Chart) | Y | Y | Y | Y | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
| Hazard (Equation) | Y | Y | Y | Y | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |

| | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long | Open Stopped Too Soon or Applied Too Long |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Case | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 |
| Stopped: | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| Aligned: | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F |
| Emergency: | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T |
| Doorway Obstructed | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T |
| Doorway Closed | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F |
| Hazard (Me) | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| Hazard (Chart) | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| Hazard (Equation) | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |

Figure 17 - One worksheet for each guide phrase

For a set of requirements that can be expressed in binary form the number of exhaustive cases to analyze would be calculated by the equation given in table 3:

Number of cases = $2^{s}$ x g,

*where s is number of binary conditions in the requirements and/or identified using STPA and g is the number of guide phrases.*

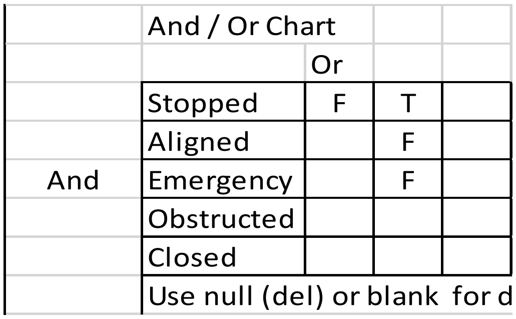
Table 3 - Formula for the number of STPA cases

Therefor in our train door example s = 5 and g = 4, and the number of cases are 128. More complex requirements would therefore generate exponentially more cases. While analyzing 128 cases may be manually feasible for a simple example analyzing more complex systems lends itself to automation.

Automation was accomplished using Visual Basic macros built into Excel. The Visual Basic code reads the worksheet rows and columns of the context table and analyzes them against either Boolean equations or and/or tables. The Boolean equations indicate the hazard conditions for each guide phrase. The Boolean equation representing hazardous conditions is compared to all combinations of conditions to look for matches. If the conditions in the context table case make the Boolean equation true, then that case is marked as a hazard. The equations are written as string types in Visual Basic, making it easier to directly compare to the letters T or F or blank on the spreadsheet. Terms that are not in the equations are assumed to not matter.

A second approach used on the automated worksheets was to construct and/or charts to express the hazard conditions. When the conditions in the and/or table match the conditions in a context table case then the case is marked as a hazard. Having two different methods to find hazardous cases (Boolean equation and and/or chart) was implemented to partially verify the correctness of the hazards. The Boolean equation and and/or charts are shown below for each guide phrase. are shown with their Boolean equation equivalents in tables x to y.

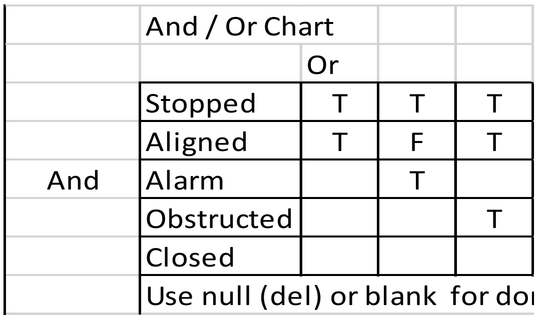
*Open Provided When: Stopped = "F" Or (Stopped = "T" And Aligned = "F" And Alarm = "F")*



| | | | | |
|---|---|---|---|---|
| | And / Or Chart | | | |
| | | Or | | |
| | Stopped | F | T | |
| | Aligned | | F | |
| And | Emergency | | F | |
| | Obstructed | | | |
| | Closed | | | |
| | Use null (del) or blank for d | | | |

Table 4 - Open Provided and/or Chart

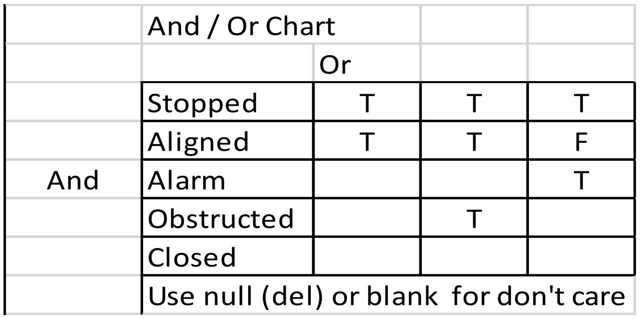
*Open Not Provided When: (Stopped = "T" And Aligned = "T") Or (Stopped = "T" And Aligned = "F" And Alarm = "T") Or (Stopped = "T" And Aligned = "T" And Obstructed = "T")*



| | | | | |
|---|---|---|---|---|
| | And / Or Chart | | | |
| | | Or | | |
| | Stopped | T | T | T |
| | Aligned | T | F | T |
| And | Alarm | | T | |
| | Obstructed | | | T |
| | Closed | | | |
| | Use null (del) or blank for doı | | | |

Table 5 – Open Not Provided and/or Chart

*Open Provided Too Late or Out of Sequence: Stopped = (Stopped = "T" And Aligned = "T") Or (Stopped = "T" And Aligned = "T" And Obstructed = "T") Or (Stopped = "T" And Aligned = "F" And Alarm = "T")*



| | | | | |
|---|---|---|---|---|
| | And / Or Chart | | | |
| | | Or | | |
| | Stopped | T | T | T |
| | Aligned | T | T | F |
| And | Alarm | | | T |
| | Obstructed | | T | |
| | Closed | | | |
| | Use null (del) or blank for don't care | | | |

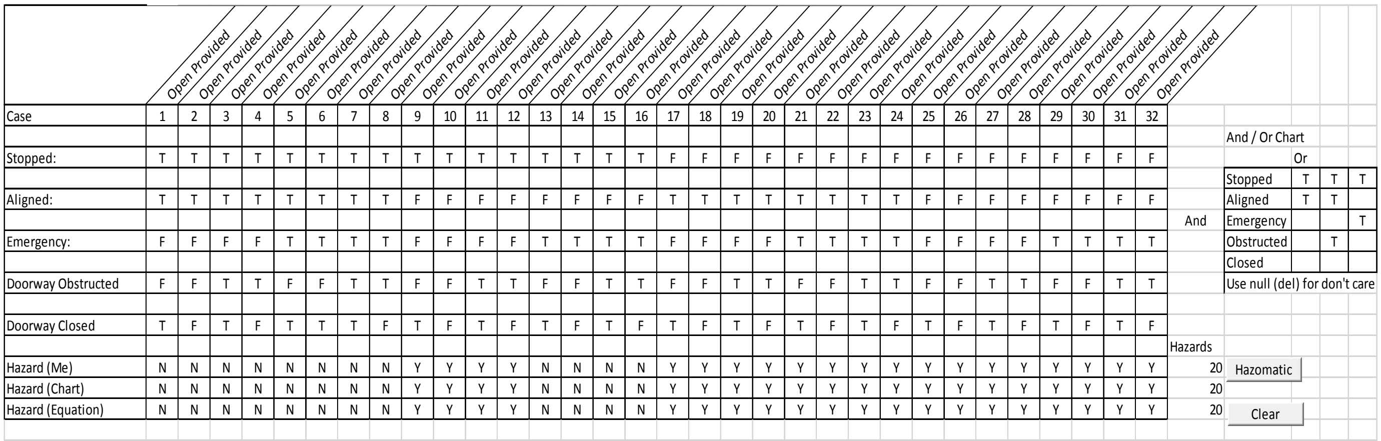
Table 6 – Open Provided Too Late or Out of Sequence and/or Chart

*Open Stopped Too Soon or Applied Too Long (Stopped = "T" And Aligned = "T") Or (Stopped = "F" And Aligned = "F") Or (Stopped = "T" And Alarm = "T")*

| | | | | |
|---|---|---|---|---|
| | And / Or Chart | | | |
| | | Or | | |
| | Stopped | T | F | T |
| | Aligned | | | |
| And | Alarm | | | T |
| | Obstructed | | | |
| | Closed | | | |

Table 7 - Open Applied Too Long or Stopped Too Soon and/or Chart

In each of the four guide phrase worksheets the results of the STPA hazard analysis are shown in three ways. The first way is this author's manual analysis in the third from the bottom row of the worksheet. The results of the and/or comparison are in the second to the bottom row, and the results of the Boolean equations are in the bottom row. If done correctly the results of the and/or table comparison and Boolean equations should be identical (bottom two rows). Every effort was made to keep the Visual Basic code simple to lessen the possibility of mistakes, but no simpler. Also, each VB function was tested in isolation (unit tested) to assure each was operating as expected. One of the concerns of doing automated STPA analysis is increasing the likelihood of adding an error in the code used to execute the hazard analysis. The VB code has a built in static analyzer to help assure correct syntax as it is being coded. This VB code could also be generalized to allow varying sizes of worksheet rows and columns and and/or tables, but constant values were used in this example to reduce the risk of errors. Figure 18 shows the worksheet for the Open Provided guide phrase cases with the automation added. The automation consists of a button to clear the automated analysis rows (bottom two rows) and a button to do the analysis as well as the and/or table. The analyze button is labeled "Hazomatic". Pushing the button causes the VB code (Appendix C) to execute on the context table and produce analysis in the bottom two rows.

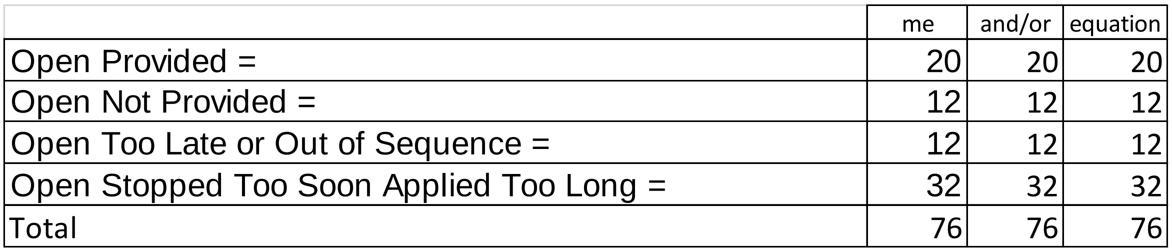


| | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | Open Provided | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Case | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | |
| Stopped: | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | |
| Aligned: | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F | T | T | T | T | T | T | T | T | F | F | F | F | F | F | F | F | |
| Emergency: | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T | F | F | F | F | T | T | T | T | |
| Doorway Obstructed | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | F | F | T | T | |
| Doorway Closed | T | F | T | F | T | T | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | T | F | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Hazards |
| Hazard (Me) | N | N | N | N | N | N | N | N | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | 20 |
| Hazard (Chart) | N | N | N | N | N | N | N | N | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | 20 |
| Hazard (Equation) | N | N | N | N | N | N | N | N | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | 20 |

| | | | | |
|---|---|---|---|---|
| | And / Or Chart | | | |
| | | Or | | |
| | Stopped | T | T | T |
| | Aligned | T | T | |
| And | Emergency | | | T |
| | Obstructed | | T | |
| | Closed | | | |
| | Use null (del) for don't care | | | |

Figure 18 - Example of worksheet with automation added.

Figure 19 shows the summary of the three different methods used to analyze the cases:



| | me | and/or | equation |
|---|---|---|---|
| Open Provided = | 20 | 20 | 20 |
| Open Not Provided = | 12 | 12 | 12 |
| Open Too Late or Out of Sequence = | 12 | 12 | 12 |
| Open Stopped Too Soon Applied Too Long = | 32 | 32 | 32 |
| Total | 76 | 76 | 76 |

Figure 19 - Summary of Hazards found using three methods.

After conducting exhaustive analysis on the train door problem, questions arose that require clarification of the original requirements that were given originally in figure 12. Analysis of 128 cases yielded 4 clarifications to the requirements. The revised requirements are shown in figure 20 with the clarifications highlighted:

Doors shall open at station when train is completely stopped and aligned with platform.

Doors shall remain closed when train is moving

Doors shall remain open when someone is in the doorway, the train shall not move until the doors close.

The train shall remain stopped and a door open command issued if an object is detected in the doorway when doors are closed

An emergency shall stop the train, Doors shall open in an emergency after the train has stopped

Figure 20 - Revised Train Door Requirements

The example exercise using the train door problem illustrates the steps required of STPA for both prefatory and exhaustive analysis. It illustrated that numerous combinatorial cases can be generated for even a small (five variable) problem. Also, that automation of STPA analysis may be helpful or even necessary for problems with a larger number of analyzed variables. It also shows in the train door case that exhaustive analysis may only produce a small number of clarifications to the requirements, but these clarifications may be critical to insure safety or vulnerability prevention.

## 5. STPA Used on the Internet of Things (IoT)

The main intent of this paper is to conduct a Hazard Analysis on the Internet of Things (IoT) to expose vulnerabilities that could materialize.

To accomplish this task a simplified logical model is used as shown in figure 21:

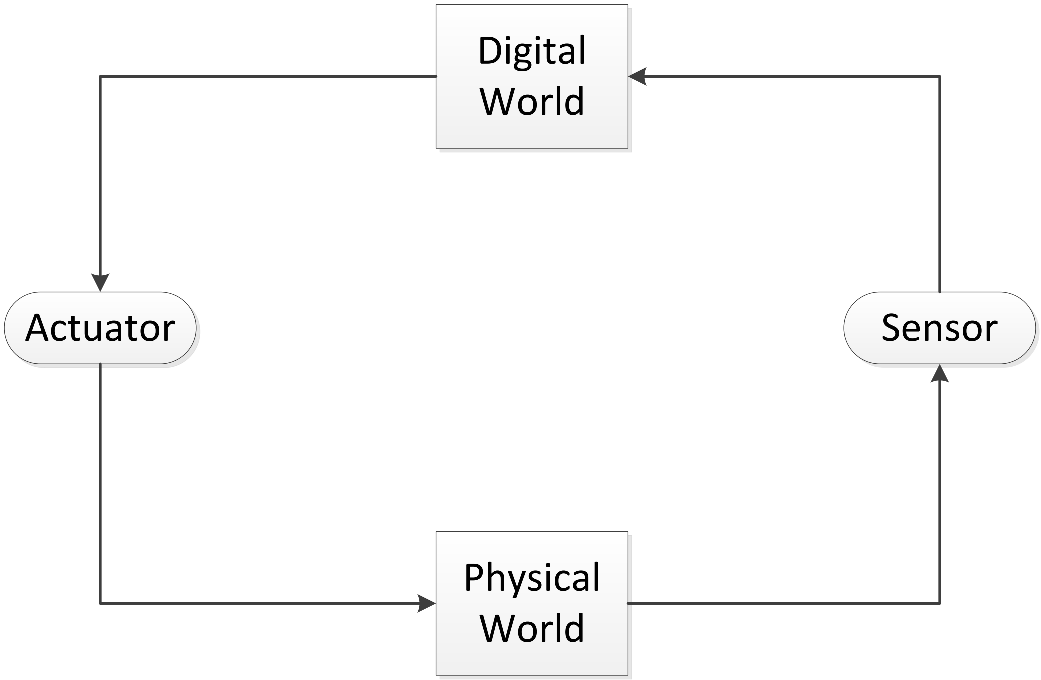


Figure 21 - Logical Model of the Internet of Things

The purpose of the internet of things as described in the introduction of this paper is to allow IP addresses to be assigned to a wide range of physical devices as well as humans and animals. In its most simplified form, the physical world and the digital world will be a system with actuators and sensors to control and receive feedback from the devices. However, the physical view of how this might be accomplished is considerably more complex. Figure 22 illustrates the interconnection of several components in the IoT system, forming three major sub-systems, IoT development, IoT Internet, and IoT Mobile. This physical representation will be used in a prefatory STPA analysis of the IoT.

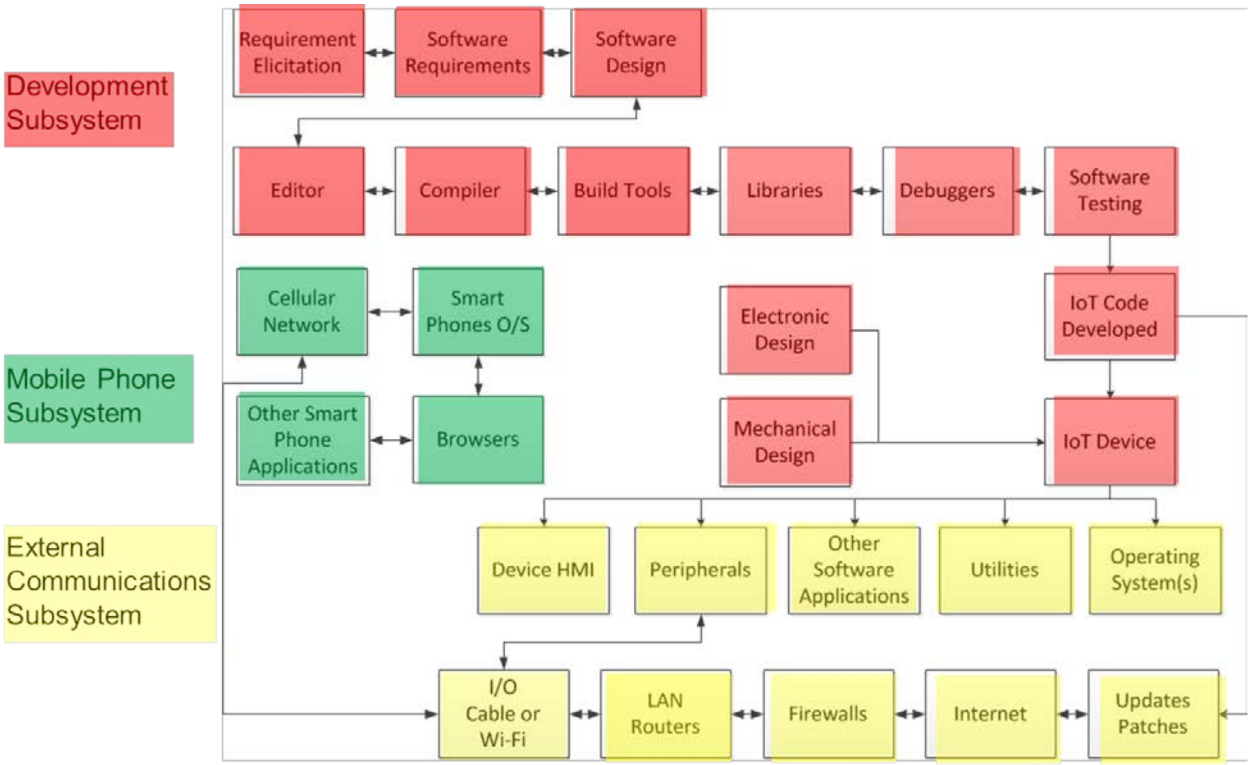


Figure 22 - Physical Representation of the Internet of Things.

The prefatory STPA analysis is done on each pair of interconnected components. This is illustrated in Figure 23 for the first two components of the IoT system, Requirements Elicitation and Software Requirements for the development subsystem.

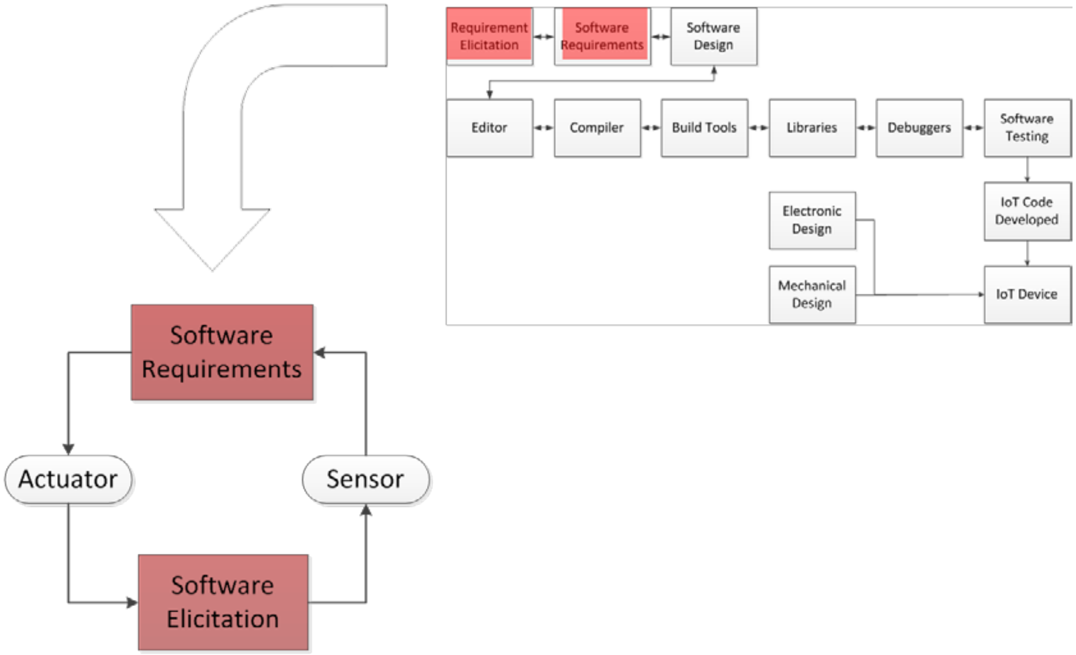


Figure 23 - STPA of the first pair of components in the development subsystem.

For the first pair of components in the development subsystem the slightly modified guide phrases are applied to identify vulnerabilities:

A resource or action required for correct operation is not provided or is not followed.

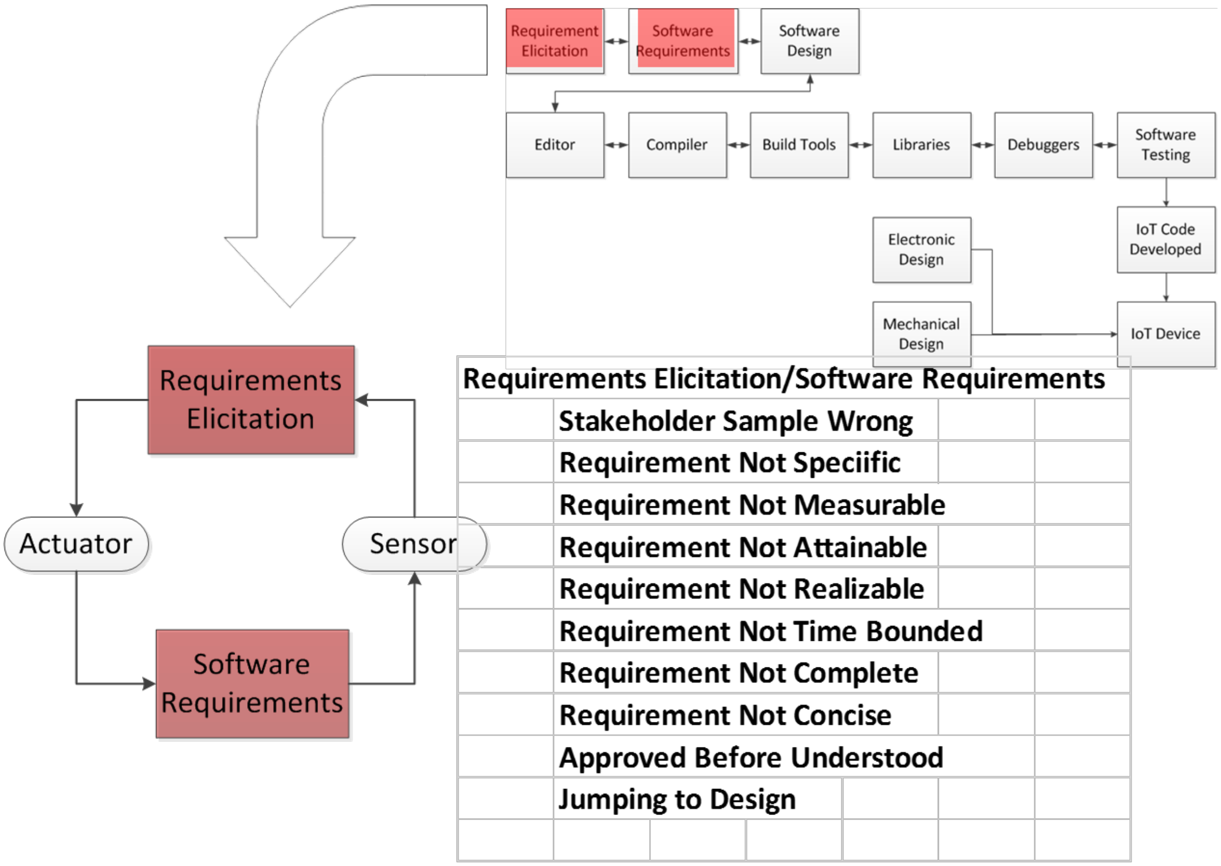
An incorrect resource or action is provided that leads to a hazard/risk.

A potentially correct resource or action is provided too late, or out of sequence.

A correct resource or control action is stopped too soon or applied too long.

Note that the linkage between the two components contains an actuator and a sensor symbol. These are not actual physical devices but instead represent information flowing from the stakeholders being elicited to the list of requirements for the IoT device and the sensor the ability to ask questions of the stakeholders based on their stated requirements and usage. In most cases the communication is bidirectional between pairs of components and the Actuator (sends information) and Sensor (receives information) analogy holds.

Figure 24 shows the result of applying the guide phrases and identifying vulnerabilities between these two components[16]:



| Requirements Elicitation/Software Requirements | |
|---|---|
| | Stakeholder Sample Wrong |
| | Requirement Not Speciific |
| | Requirement Not Measurable |
| | Requirement Not Attainable |
| | Requirement Not Realizable |
| | Requirement Not Time Bounded |
| | Requirement Not Complete |
| | Requirement Not Concise |
| | Approved Before Understood |
| | Jumping to Design |

Figure 24 - Identified Vulnerabilities for first pair of components in the development subsystem

The prefatory STPA analysis process continues for each pairing of components in the three major subsystems. The pairwise analysis of each component set is shown in Appendices A for the Development Subsystem, the External Communication Subset, and for the Mobile Phone Subsystem. The pairwise analysis allows a deeper consideration of vulnerabilities that can be present.

The results of the prefatory STPA analysis of the physical representation of IoT in Appendices A is as follows in table 8:

| Physical Model of IoT | Vulnerabilities |
|---|---|
| Development Subsystem | 110 |
| Internet Subsystem | 64 |
| Mobile Subsystem | 27 |
| Total | 201 |

Table 8 - Number of Identified Vulnerabilities of STPA Analysis for IoT Physical Model

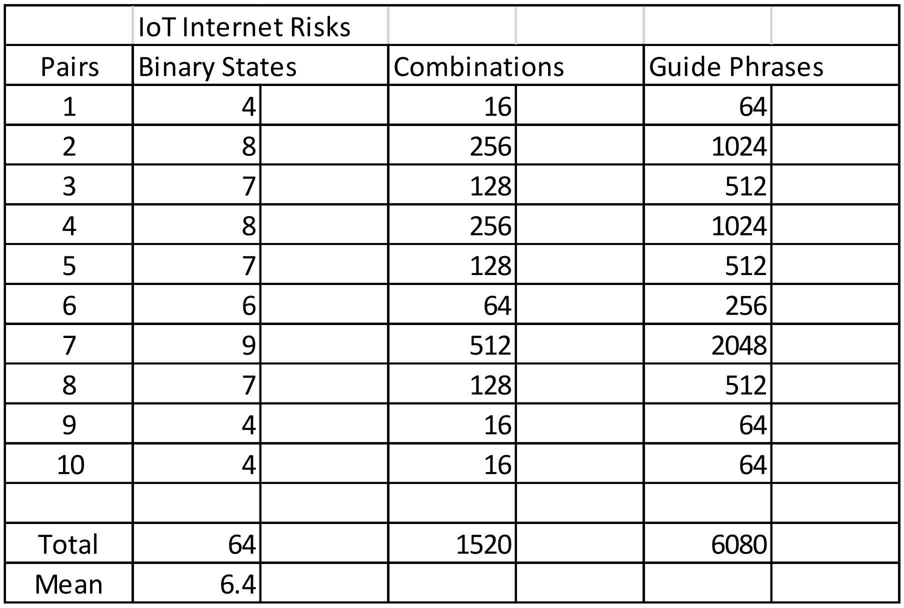
As can be seen in table 8, a summary of the prefatory STPA analyses, 201 vulnerabilities identified were in all three IoT subsystems. Each of these vulnerabilities can be viewed as a binary event that will happen or not. Using the simple formula derived in table 1 from the train door example earlier we can estimate the number of cases required for an exhaustive analysis of all possible combinations. Using the formula in table 1 yields a very large number as shown in the calculation below:

Combinations = $2^{201}$ x 4 = 1.29E+61

Therefore, an exhaustive STPA analysis such as done for the train door example is not possible within a human life time for the IoT. A second approach to quantifying the amount of analysis needed is to use the combination formula on each IoT subsystem component pair and then sum those cases. This is shown in tables 9 through 11

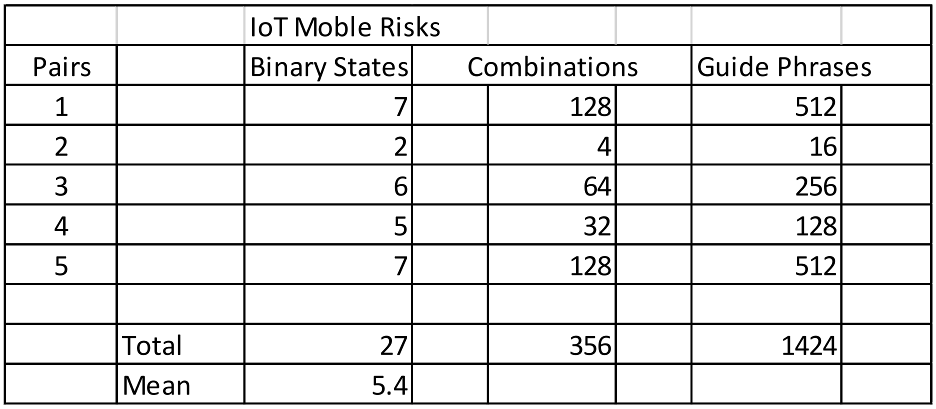
| | SW Development Risks | | |
|---|---|---|---|
| Pairs | Binary States | Combinations | Guide Phrases |
| 1 | 10 | 1024 | 4096 |
| 2 | 10 | 1024 | 4096 |
| 3 | 9 | 512 | 2048 |
| 4 | 13 | 8192 | 32768 |
| 5 | 11 | 2048 | 8192 |
| 6 | 7 | 128 | 512 |
| 7 | 5 | 32 | 128 |
| 8 | 7 | 128 | 512 |
| 9 | 6 | 64 | 256 |
| 10 | 8 | 256 | 1024 |
| 11 | 11 | 2048 | 8192 |
| 12 | 13 | 8192 | 32768 |
| | | | |
| Total | 110 | 23648 | 94592 |
| Mean | 9.2 | | |

Table 9 - Software Development Risks Summed



| | IoT Internet Risks | | | | | |
|---|---|---|---|---|---|---|
| Pairs | Binary States | | Combinations | | Guide Phrases | |
| 1 | 4 | | 16 | | 64 | |
| 2 | 8 | | 256 | | 1024 | |
| 3 | 7 | | 128 | | 512 | |
| 4 | 8 | | 256 | | 1024 | |
| 5 | 7 | | 128 | | 512 | |
| 6 | 6 | | 64 | | 256 | |
| 7 | 9 | | 512 | | 2048 | |
| 8 | 7 | | 128 | | 512 | |
| 9 | 4 | | 16 | | 64 | |
| 10 | 4 | | 16 | | 64 | |
| | | | | | | |
| Total | 64 | | 1520 | | 6080 | |
| Mean | 6.4 | | | | | |

Table 10 - IoT Communications Risks Summarized



| | | IoT Moble Risks | | | | | |
|---|---|---|---|---|---|---|---|
| Pairs | | Binary States | Combinations | | | Guide Phrases | |
| 1 | | 7 | | 128 | | 512 | |
| 2 | | 2 | | 4 | | 16 | |
| 3 | | 6 | | 64 | | 256 | |
| 4 | | 5 | | 32 | | 128 | |
| 5 | | 7 | | 128 | | 512 | |
| | | | | | | | |
| | Total | 27 | | 356 | | 1424 | |
| | Mean | 5.4 | | | | | |

Table 11 - IoT Mobile Risks

Table 12 sums up the totals from each IoT subsystem (bottom right corners)

| Physical Model of IoT | Vulnerabilities | Combinations |
|---|---|---|
| Development Subsystem | 110 | 94592 |
| Internet Subsystem | 64 | 6080 |
| Mobile Subsystem | 27 | 1424 |
| Total | 201 | 102096 |

Table 12 - Number of Combinations of IoT Vulnerabilities

Summing the number of identified vulnerability combinations ("or"ing them) reduces the number of combinations needed to analyze to a little over 100,000 combinations. By doing this we are making a simplifying assumption that the vulnerabilities in each IoT subsystem component are independent from each other. The speed at which the train door spreadsheet STPA analysis executed was roughly 32 cases per second (with a 3.2 GHz Intel Xeon CPU, 8 GB RAM, 64-bit mode).  This number of cases (102,096) at 32 cases a second on a spreadsheet would take a little less than one hour to analyze however the size of the IoT physical system makes the exhaustive approach unappealing.

## 6. STPA Surrogate Approach to Vulnerabilities

We can simplify the IoT analysis further by using average vulnerabilities per pair of components represented by surrogates. The average number of vulnerabilities for the pairs of components in each subsystem is shown in table 13.

| Physical Model of IoT | Vulnerabilities | Combinations | Average/Pair |
|---|---|---|---|
| Development Subsystem | 110 | 94592 | 9.2 |
| Internet Subsystem | 64 | 6080 | 6.4 |
| Mobile Subsystem | 27 | 1424 | 5.4 |
| Total | 201 | 102096 | 7.0 |

Table 13 - Average number of vulnerabilities found using STPA

The STPA surrogate approach used is accomplished by substituting a single component pairing for each of the three IoT subsystems. The vulnerabilities listed for the surrogate are selected from the analysis completed for each pairing in the represented subsystem. For example, in Table 13 if all 110 identified vulnerabilities in the development subsystem are used to develop a vulnerability list for the development system surrogate. The surrogate list of vulnerabilities does not repeat duplicate vulnerabilities. For instance, the vulnerability of not having an updated version of software to eliminate a known vulnerability occurs in multiple sub-system component pairings but is only listed once in the surrogate. Also, a group of individual pairing vulnerabilities may be summarized at the surrogate level, for instance requirements not being specific, measurable, attainable realizable time bounded, complete, and concise is summarized as “requirements incorrect”. The surrogate STPA for the three IoT subsystems are shown in figures 25 – 27. The STPA analysis using the surrogate approach provides a more reviewable and useful summary of the more detailed analysis of each component pair.

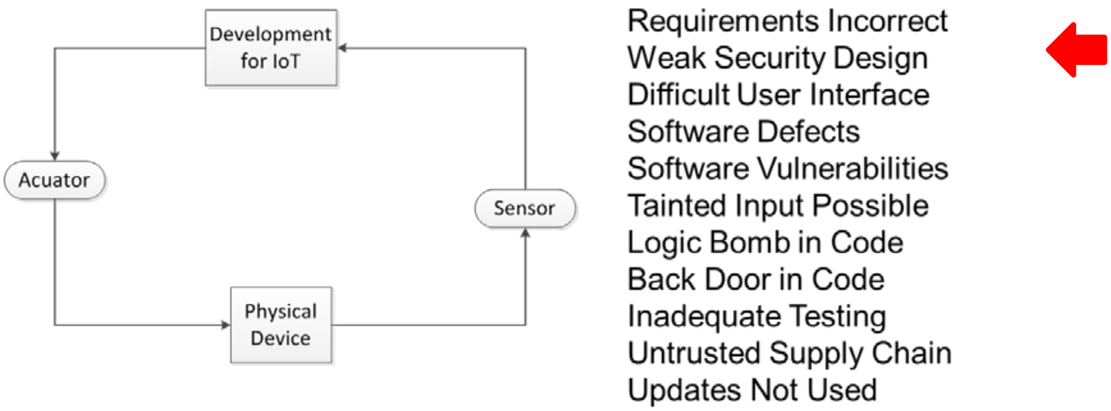


Figure 25 - Surrogate STPA for IoT Development Subsystem

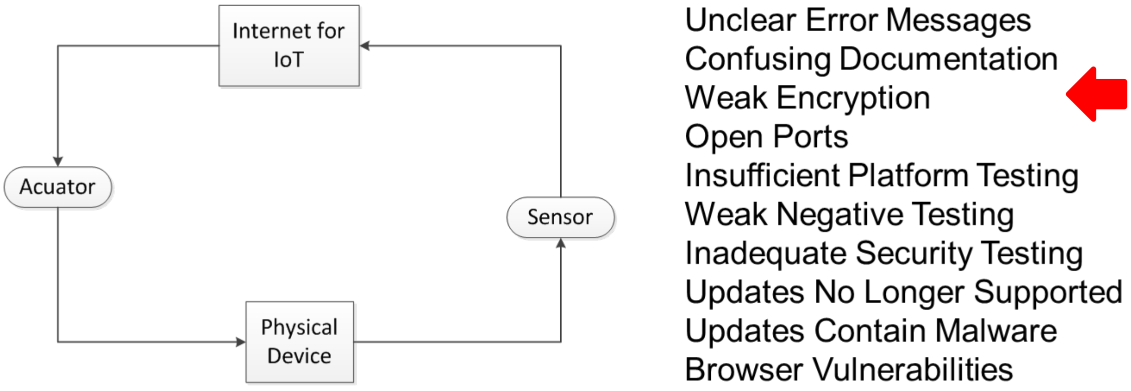


Figure 26 - Surrogate STPA for IoT Internet Subsystem

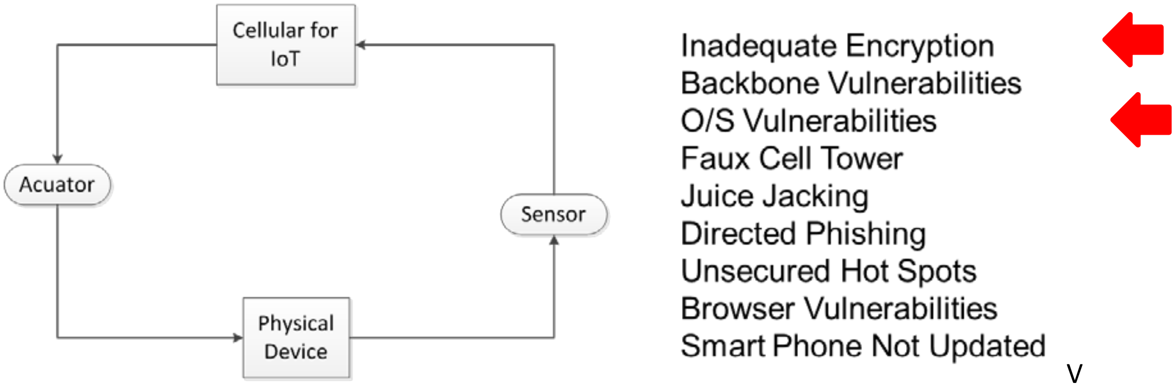


Figure 27 - Surrogate STPA for IoT Internet Subsystem

## 7. STPA Domain Expert Review

One of the values of STPA is the ability to have the analysis reviewed by domain experts to get additional input on Hazards, Vulnerabilities, and Risks. The expert used for a quick review of the STPA analysis for IoT had two major concerns:

1. The first concern was that the low price of the IoT device (under $150) would not support the costs of continuous updates to mitigate cyber vulnerabilities. At some point the device would no longer be updated and notification that it has a security vulnerability may not be publicized. Just the operating systems alone (some stripped down version of Linux or Windows) would at some time no longer be supported by updates from the manufacture or licensor.

2. The second concern was that the hardware components chosen for the IoT device would be as low cost as possible and not include the memory and CPU resources to support good encryption, at least AES level. AES is fully supported in personal computers and smart phones, but these cost considerably more than $150.

The vulnerabilities listed by the cyber security expert were annotated with red arrows in figures 25-27 as they also turned up with the STPA analysis. The domain expert's value was to explain how these discovered vulnerabilities might be likely to occur. Without the STPA analysis, the discussion with the domain expert may have never taken place. STPA offers an excellent launching point for discussions with other domain experts.

## 8. IoT User Interface and Updates Challenge

One of the STPA potential vulnerabilities found was the IoT user interface. Figure 28 shows a sample set of possible IoT devices in our homes by 2022. Many tragic accidents have occurred or made worse because the user was confused by the user interface. With so many potential manufactures making IoT devices it is imperative that user interface standards be developed so users can have the "same" look and feel with numerous devices. Also, these user interfaces should be integrated onto a single application. Along these lines some early UI standards are emerging from Goggle and Apple. Google having the "Works with Nest"[17] and Apple having "Apple Home Kit"[18]. Without these standards, a homemaker will need significant IT skills to just keep the household running.

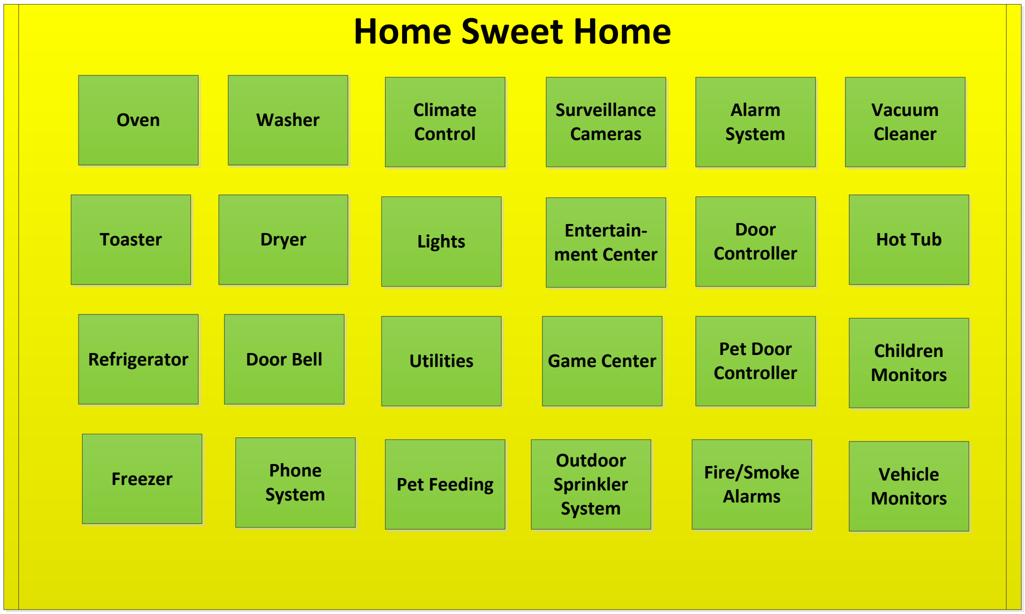


Figure 28 - Household IoT devices in 2022

Another concern found with STPA (and the domain expert) was the need to upgrade devices to fix cyber vulnerabilities. Figure 29 shows that for the physical IoT representation 22 different components will need to be updated, for each fielded IoT device. Updates need to be automated or semi-automated (ask first) and vendors need to continue to support their IoT products and supply chains:

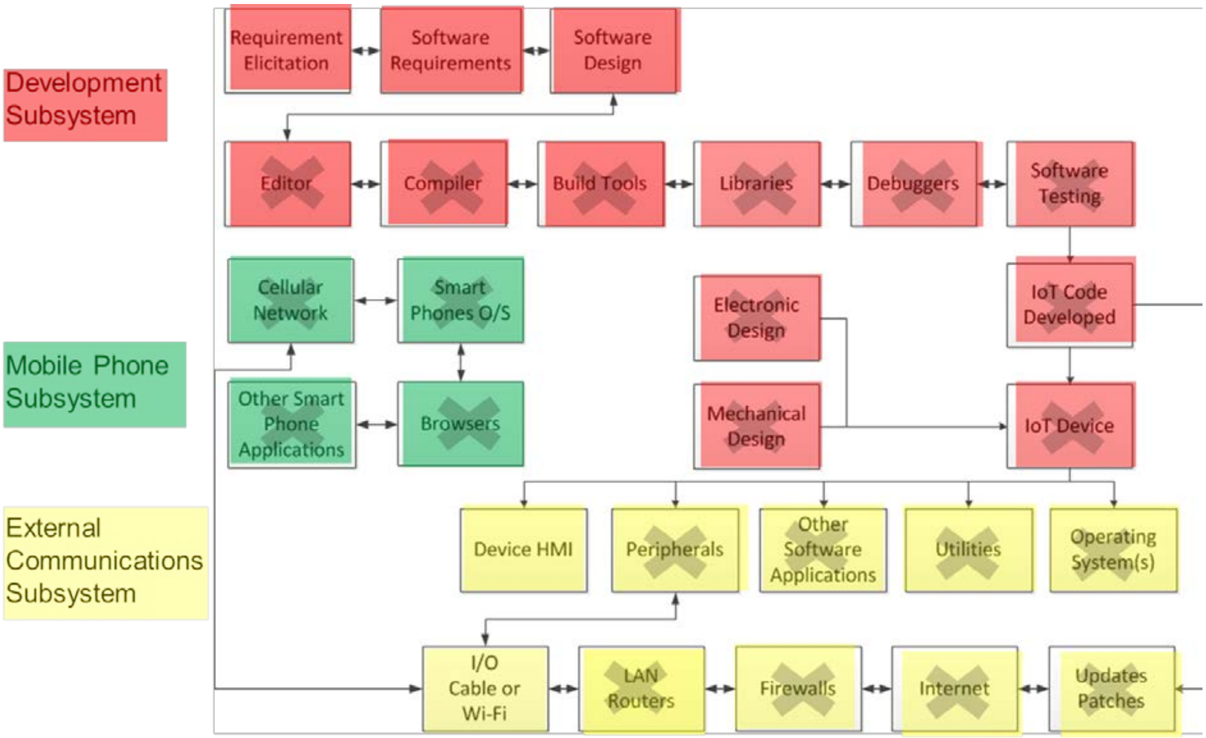


Figure 29 - Updating the IoT Physical Model and Supply Chain

The update and obsolescence concern existed before the internet of things. Software vendors who have been in business for over a decade have great difficulties supporting older versions of their software products. Even large companies such as Microsoft suspend support of their older operating systems. Updates to correct vulnerabilities in Windows NT and Windows XP are no longer available from Microsoft. However, if an IoT device vendor chose Windows NT or XP as their on-board operating system any vulnerabilities in the IoT device would remain, inviting hackers to exploit them. The IoT vendor could update the operating system in the IoT device, but usually new operating systems take up more memory than older ones, and most IoT devices do not provide extra memory due to cost, space, power, and heat restrictions.

The IoT software itself must be updated when vulnerabilities are detected. A concern for a $129 IoT device is how much of the purchase price covers future upgrades and for how long? Is supplier responsible to sell IoT devices without an adequate update program to mitigate future exploited vulnerabilities? Many consumers are naïve about the need for security in their devices or assume the supplier has taken care of it for them.  An example of this naivety is the ability of the Shodan[19] search engine, a free commercial software product, which can crawl through cyber space and find open ports on the internet. This includes video cameras that do not have password protection or encryption. IoT manufactures should consider security in their designs, including AES level or encryption of better and encouragement of users to use strong passwords on these devices before allowing them to go on line.

There are numerous cases where known vulnerabilities can persist because the manufacture has discontinued support. An example is the Aristocrat[20] Mk 6 Slot Machine built in Sydney, Australia. When

Russian President Vladimir Putin essentially eliminated gambling in Russia in 2009[21] the world market was flooded with gambling machines. Some of these excess machines found their way to the Russian underground. The code within the underground purchased machines was reverse engineered to reveal the type of pseudo random number generator (PRNG) used in the Aristocrat Mk 6 slot machine. Knowing how the code worked would not allow a player to gain an advantage on the machine unless the player knew what random number cycle the machine was on. If the machine's PRNG and current cycle is known the player gains an advantage over the machine.

The scam would work by the agent entering an unsuspecting casino and locating an Aristocrat Mk 6 slot machine. The agent would capture video of two minutes of play on the machine on their smart phones camera (either by holding the smart phone facing the machine, or having the smart phone in a shirt pocket with an opening for the lens). The captured video was uploaded and received in Saint Petersburg, where it is processed to determine the machine sequence. The sequence information was downloaded to the agent's smart phone back at the casino. The smart phone on-board application would buzz the phone when a winning number was coming up on the machine. The agent then pushed the button. The agent was much more likely to hit the spin button at times when the machine had a winning combination. The only real visible behavior that tips off the casino that the system is being used is the slot machine player pushing the spin button somewhat erratically. See figure 30:

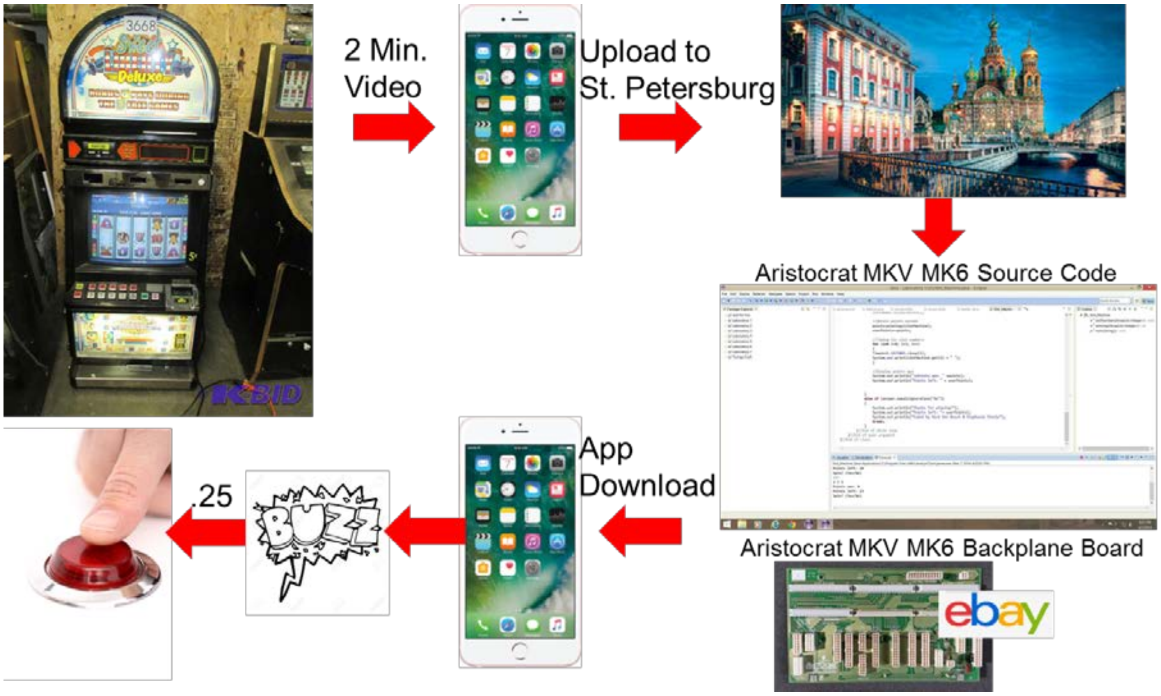


Figure 30 - Beating the Aristocrat Mk 6 Slot Machine

While this story points out the level of ingenuity used to defeat software controlled devices, the bigger point of this story, in the IoT context, is that Aristocrat Mk 6 slot machines are still used in many casinos throughout the world. The usual reason given is the replacement cost is too high. Even though machines and parts for these machines are available on eBay. In the case of IoT the same phenomenon is likely to occur, ingenuity used to defect security and obsolete devices left as easy targets for hackers[22].

## 9. Recent IoT Hacks

Remote control cars and drones have found their way onto battlefields. In Syria explosives were added to inexpensive remote control vehicles which were then sent out by the Islamic State fighters into their enemies ranks and detonated.[23] In figure 31 a captured device from the battlefield is shown.

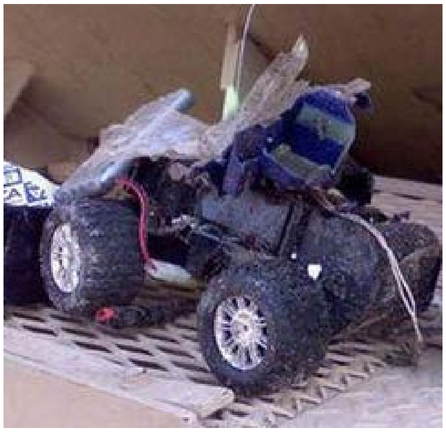


Figure 31 - Captured Remote control vehicle Used by the Islamic State as a weapon.

Another IoT concern is the increased number of IP equipped devices available to use in Distributed Denial of Service (DDoS) attacks such as the one on October 2016.[24] This attack shown in figure 32, used IoT devices including CCTV video cameras to overwhelm one of the largest Internet Service Providers. These IoT devices were not protected or relied on factory default passwords for protection. Because of this attack numerous web businesses were interrupted or disabled.

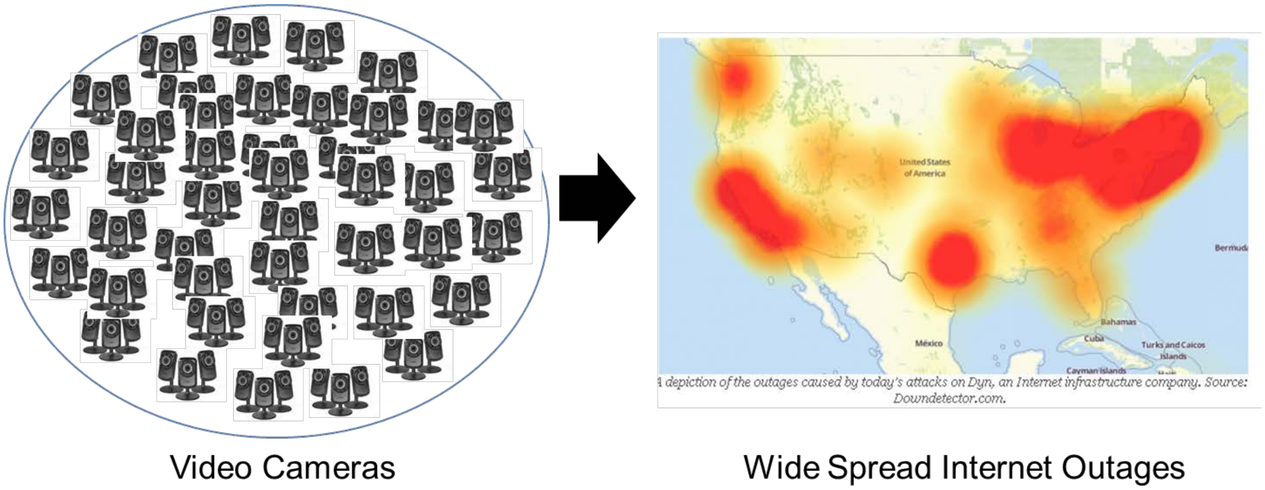

United States of America
Bermuda
México
Bahamas
Cuba
Turks and Caicos Islands
Cayman Islands
A depiction of the outages caused by today's attacks on Dyn, an Internet infrastructure company. Source: Downdetector.com.
Video Cameras
Wide Spread Internet Outages

Figure 32 - IoT Devices Used in DDoS Attack

Another attack occurred to an Austrian Hotel just before a sold-out night in January 2017. The hotel room key system shown in figure 33 was compromised and the hotel's security codes replaced with one only the hacker knew. This basically disabled the hotel's ability to issue room keys that could open room doors. The hackers demanded a ransom of 1500 euros in untraceable bitcoins to restore the hotel security key and the hotel paid. The attack has been called the ransomware of things (RoT). The amount of the ransom was small enough to incent the hotel to pay rather than cancel business. The security system software[25] and computers supporting the system have been replaced, the hotel owner is considering going back to traditional metal keys in the future.

Figure 33 - Hotel Electronic Locking System Compromised by Hackers Who Demanded Ransom

Although the Internet of Things is still an emerging technology, there are already numerous examples of the great lengths that unscrupulous people will go to exploit IoT vulnerabilities.

## 10. Future IoT Hacks and Concerns

Based on doing the STPA analysis on the IoT additional vulnerabilities are predicted in the near term. One such vulnerability is leaving default passwords in devices or putting in easy to crack passwords. This combined with the capability of turning ovens and toaster devices on and off from remote locations could lead to disruptions in the power grid. Instead of using video cameras to flood an ISP in a DDoS attack, 167,000 ovens or 454,000 toasters could be turned on within seconds of each other and add a 500-megawatt power surge to the grid as shown in figure 34. A surge of this size would be the equivalent of the capacity of a large fossil fuel or nuclear powered plant. This type of attack could cause power disruptions to wide areas and wreak havoc on the public.

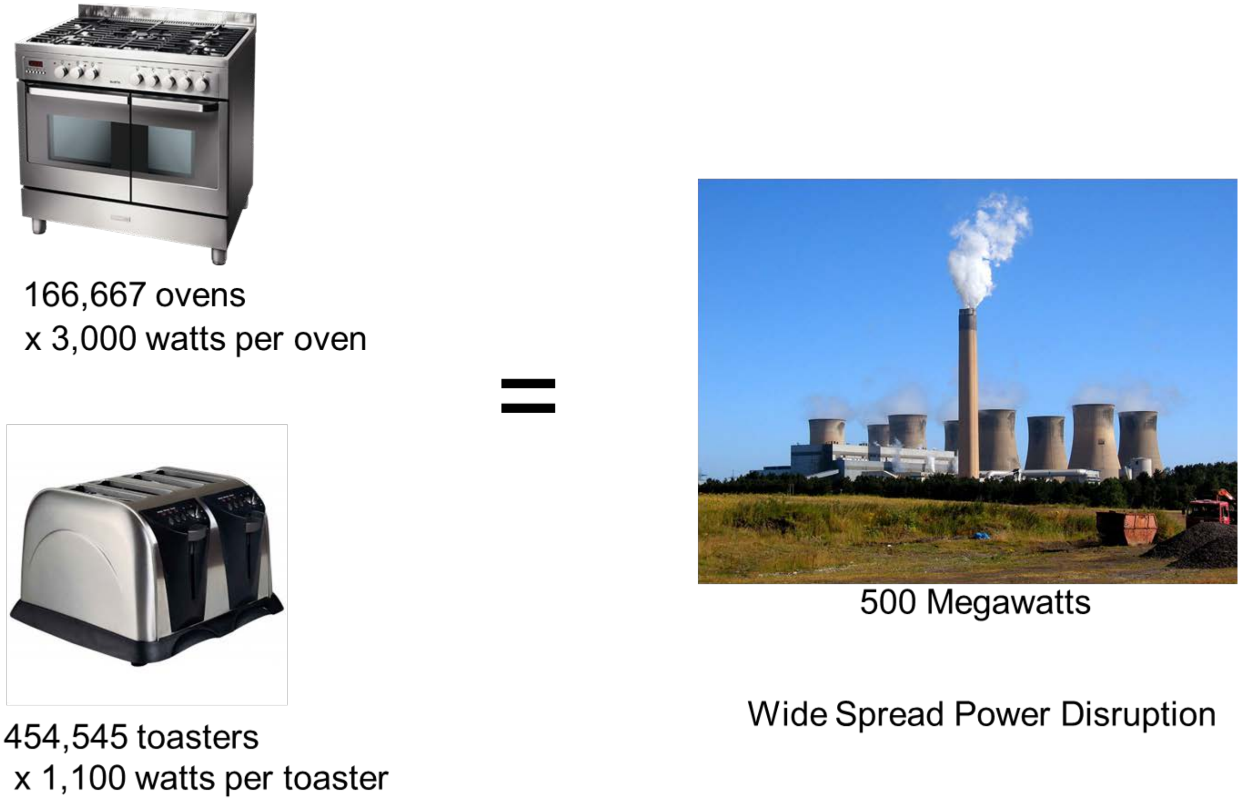


Figure 34 - Toasters Bring Down the Power Grid

Cyber buglers soon could target homes using tools such as Shodan[26] which can find security devices using default passwords or simple passwords to protecting homes, as well as the geographical location of these devices. The Cyber burglar could then check to see if the home looks vacated by querying the alarm device to see if the mode is in "away". If the house also appears empty because of lack of motion detected, the cyber burglar then disables the home alarms and opens external doors. The Cyber burglar then optionally deploys a drone over the targeted house to further scout for activity in the neighborhood and act as a look out. After arriving in the neighborhood using a self-driving car the cyber burglar leaves it a few blocks away. After looting the targeted house the cyber burglar summons the self-driving car from its parked location to provide a clean get away as shown in figure 35.

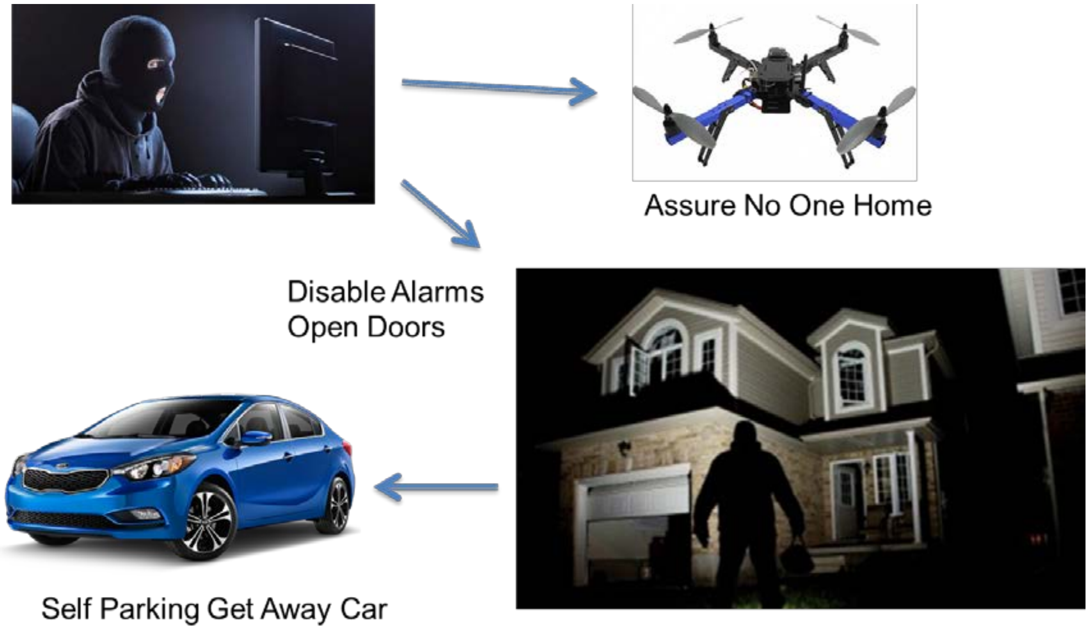


Figure 35 - Cyber Burglary

The IoT has the capability to provide online monitoring for medical patients as shown in figure 36. To track the patient’s vital signs as they go about their normal lives outside of a medical facility. This diagnostic technique would be most useful for a symptom that occurs randomly. The medical device sends the patient’s vital signs back to a medical facility where the medical staff is alarmed if the readings reach dangerous levels. The remote medical staff can contact the patient immediately or dispatch an ambulance to the location of the patient. This use of the IoT could save lives and help pinpoint hard to diagnose health conditions. However, the same information could also be used be health insurance companies to adjust medical insurance rates in near real time.

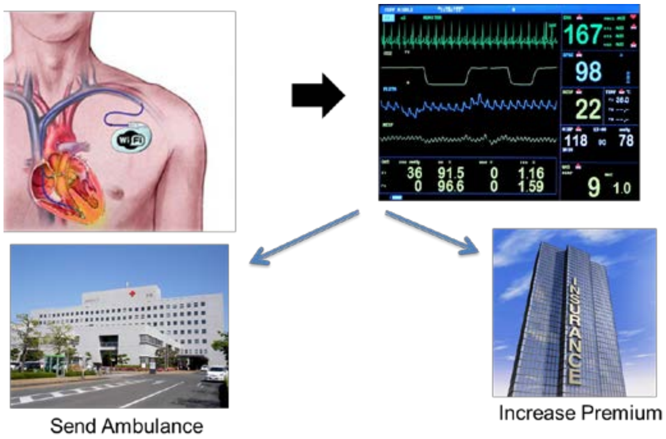
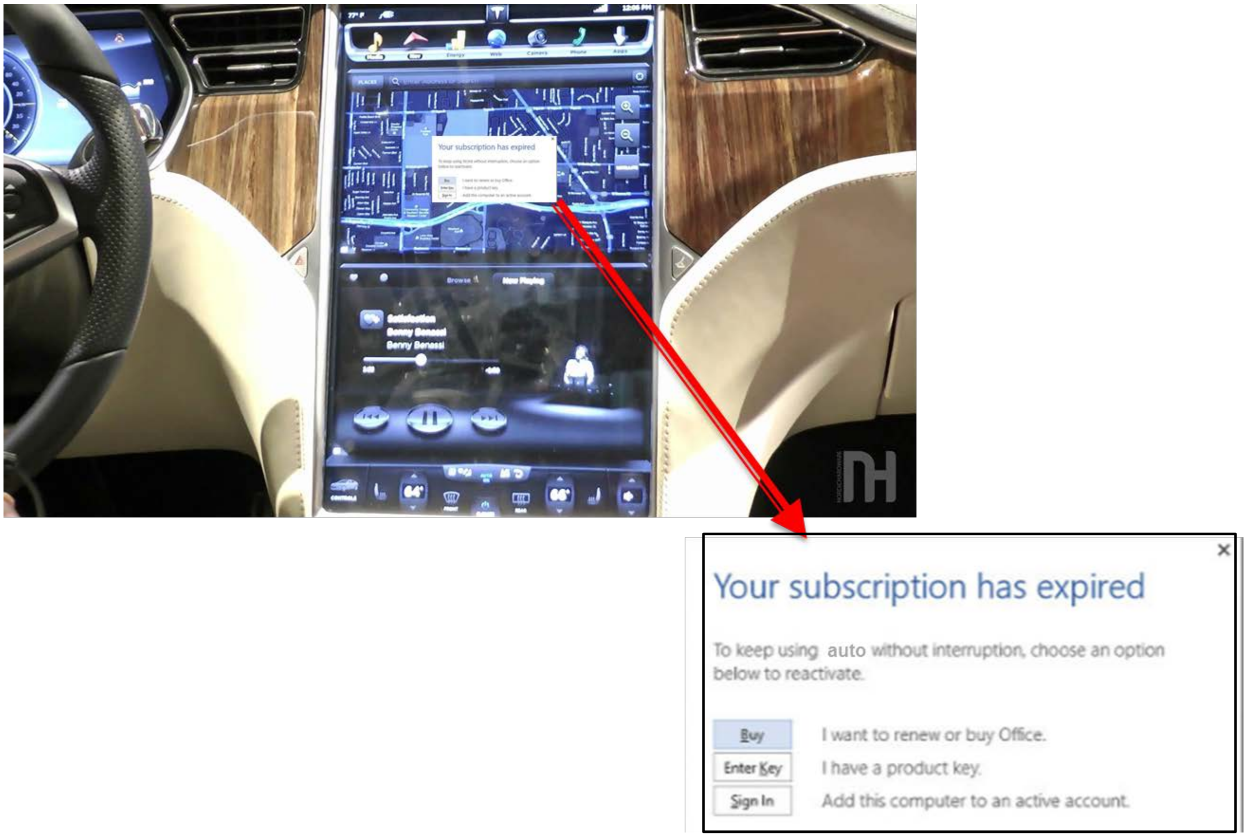


Figure 36 - Remote Patient Monitoring

Software has become common place in automobiles, reaching over 100 million lines of source code. One advantage of software controlled vehicles is upgrades can be accomplished without having to take the vehicle to the garage. An owner can get the latest updates for cyber security or added features by downloading the software updates.  Software licensing issues however could begin to cause problems in the automobile industry, see figure 37.



Your subscription has expired

To keep using  auto without interruption, choose an option below to reactivate.

Buy — I want to renew or buy Office.

Enter Key — I have a product key.

Sign In — Add this computer to an active account.

Figure 37 - Subscription Software for Automobiles

How long is a car manufacture going to continue to provide updates for vehicle cyber security vulnerabilities? Are new feature updates going to require annual subscription licenses? Does the software license transfer to a new owner when the vehicle is sold? Will vehicle software follow the model of other software offering less expensive versions with advertising pop ups (see figure 38)?

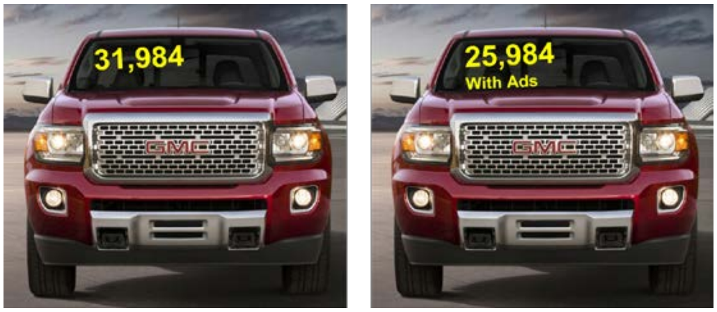


Figure 38 - Auto Pricing Options

The automobile has always attracted after-market add-ons that improve stock performance or aesthetics. As the inevitable third-party vehicle software market grows, compatibility issues may arise. Third party software suppliers could offer increased acceleration or battery life with their add-ons, but could also cause unanticipated complications with the manufactures software or vehicles components such as reduced braking distance or increased charging time. The topic of vehicular software regulation would seem to be an area for increased government intervention, as has been done in the past to assure seat belts and crumple zones and other safety innovations are built into vehicles.

Dash buttons that allow convenient reordering of consumable supplies for the household could also be misused by the younger home members who are not aware of their purpose. A young child being toilet trained may find pushing the Charmin Dash Button amusing, that is until a truckload of toilet paper shows up in the garage along with a large invoice, see figure 39.

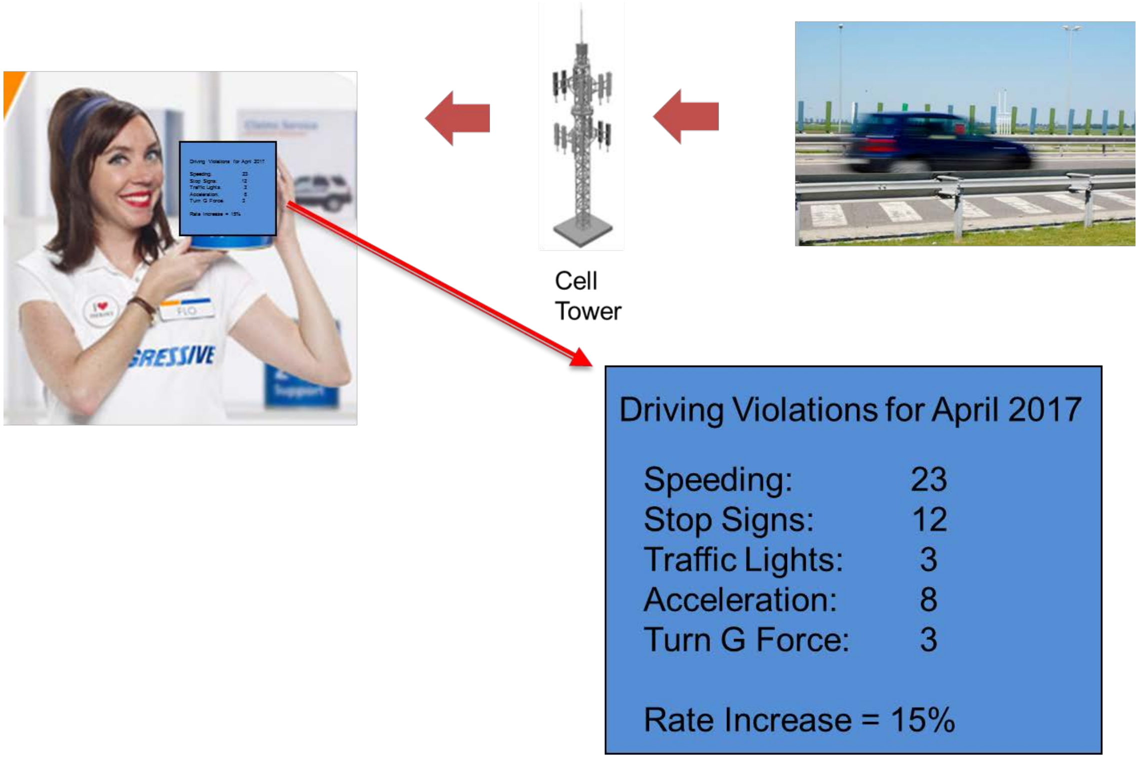
Figure 39 - Unwanted Supplies Showing Up In The Garage

Point of sale devices could find their way into new fields, such as pay as you go medicine and dentistry. Imagine the incentive to pay for anesthesia just prior to surgery or a tooth extraction with a portable credit card reader. See figure 40.

…and how did you want to pay for your anesthesia today Martha?

Figure 40 - Just in Time Anesthesia Encourages Payment

Automobile insurance companies could monitor driving habits of insured motorists and adjust rates monthly based on comparing information from the instrumented vehicle with maps of the region, see figure 41.



Driving Violations for April 2017

| | |
|---|---|
| Speeding: | 23 |
| Stop Signs: | 12 |
| Traffic Lights: | 3 |
| Acceleration: | 8 |
| Turn G Force: | 3 |

Rate Increase = 15%

Figure 41 - Flo Knows the Internet of Things

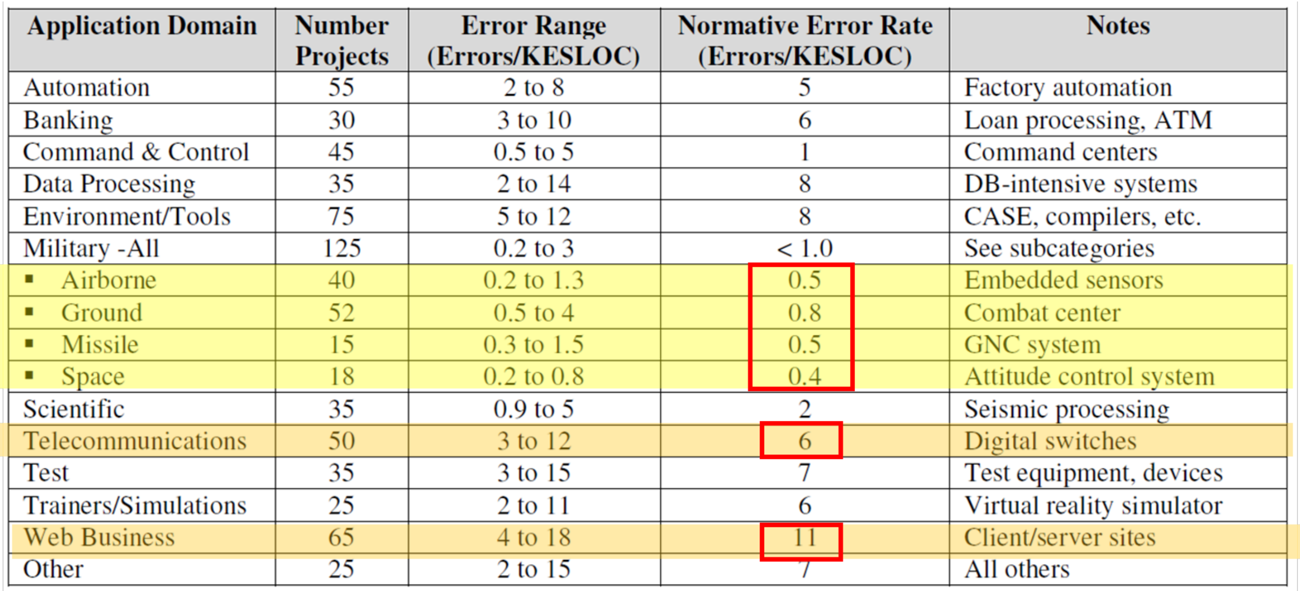
Are self-driving cars going to lull drivers into a state of inattention and complacency so they will be unable to take back control of their vehicle in an emergency? There is growing concern in the FAA[27] about this happening to commercial airline pilots and tragic accidents that have occurred when pilots have attempted to take over manual control from automation.[28] The underlying philosophies of how automation is implemented in automobiles could vary from one manufacture to another. In the aircraft industry, this is already an issue. Boeing fly by wire aircraft automation philosophy is to let the pilot overrule the automation. Airbus on the other hand has the automation philosophy of not letting the pilot do anything harmful to the aircraft. In the future when renting a car, the automobile driver will be faced with more challenges than simply locating the light switch and turn signals, now the driver must also be aware of the type of automation that the car has on board and how to use it or disable it. The impact of devices that can jam GPS signals [29] now becomes much more serious when vehicle automation is depending on it for position information. Such devices are currently marketed on the internet. In figure 42, will the truck driver, distracted with his iPad, realize there is an error message on his dash board display in time to regain control of the vehicle, or will the blue screen of death take on a more literal meaning?

```
                         Windows

A fatal exception 0E has occurred at 0028:C562F1B7 in VXD ctpci9x(05)
+ 00001853. The current application will be terminated.

*  Press any key to terminate the current application.
*  Press CTRL+ALT+DEL again to restart your computer. You will
   lose any unsaved information in all applications.

                    Press any key to continue _
```

Figure 42 - New Meaning for Blue Screen of Death

## 11. IoT Software Quality Requirements

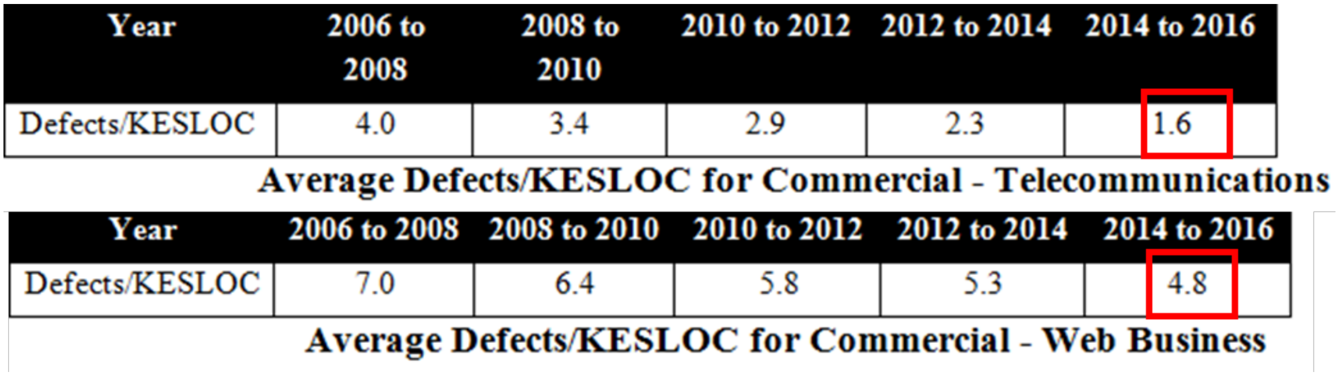
The quality of the software used in IoT devices and the supporting IoT infrastructure also contributes to the device’s vulnerability to hackers. Quality meaning the software correctly implements robust security requirements and is free of structural code errors. Structural code error make up about 38% of all possible code errors and are issues such as not initializing a variable before using it, misuse of pointers, mismanagement of memory, not carefully checking user supplied data for correctness before using it. Industry studies are conducted to evaluate the error rates of software for different types of applications. Table 14 shows error ranges for software compiled in 2004 by Donald Reifer[30].The error rates are shown in errors per thousand lines of equivalent code (KSELOC). The E or equivalent lines of code factor normalizes the source code language. Source code language syntax like C require more lines of code to accomplish a given function than for instance Java source code. The equivalence factor allows error rates to be compared between codes written in different languages.



| Application Domain | Number Projects | Error Range (Errors/KESLOC) | Normative Error Rate (Errors/KESLOC) | Notes |
|---|---|---|---|---|
| Automation | 55 | 2 to 8 | 5 | Factory automation |
| Banking | 30 | 3 to 10 | 6 | Loan processing, ATM |
| Command & Control | 45 | 0.5 to 5 | 1 | Command centers |
| Data Processing | 35 | 2 to 14 | 8 | DB-intensive systems |
| Environment/Tools | 75 | 5 to 12 | 8 | CASE, compilers, etc. |
| Military -All | 125 | 0.2 to 3 | < 1.0 | See subcategories |
| ▪ Airborne | 40 | 0.2 to 1.3 | 0.5 | Embedded sensors |
| ▪ Ground | 52 | 0.5 to 4 | 0.8 | Combat center |
| ▪ Missile | 15 | 0.3 to 1.5 | 0.5 | GNC system |
| ▪ Space | 18 | 0.2 to 0.8 | 0.4 | Attitude control system |
| Scientific | 35 | 0.9 to 5 | 2 | Seismic processing |
| Telecommunications | 50 | 3 to 12 | 6 | Digital switches |
| Test | 35 | 3 to 15 | 7 | Test equipment, devices |
| Trainers/Simulations | 25 | 2 to 11 | 6 | Virtual reality simulator |
| Web Business | 65 | 4 to 18 | 11 | Client/server sites |
| Other | 25 | 2 to 15 | 7 | All others |

Table 14 - Software Error Range by Application Domain

The telecommunications and web business application domains are highlighted to indicate the error rates. They are considerably higher than the error rates for military software of the mission critical variety. Table 15 is based on error rates at product release, so the defect rates indicate defects found during development. Donald Reifer was kind enough to supply more recent data for the commercial telecommunications and web business application domains in 2016 as shown in table 15:



| Year | 2006 to 2008 | 2008 to 2010 | 2010 to 2012 | 2012 to 2014 | 2014 to 2016 |
|---|---|---|---|---|---|
| Defects/KESLOC | 4.0 | 3.4 | 2.9 | 2.3 | 1.6 |

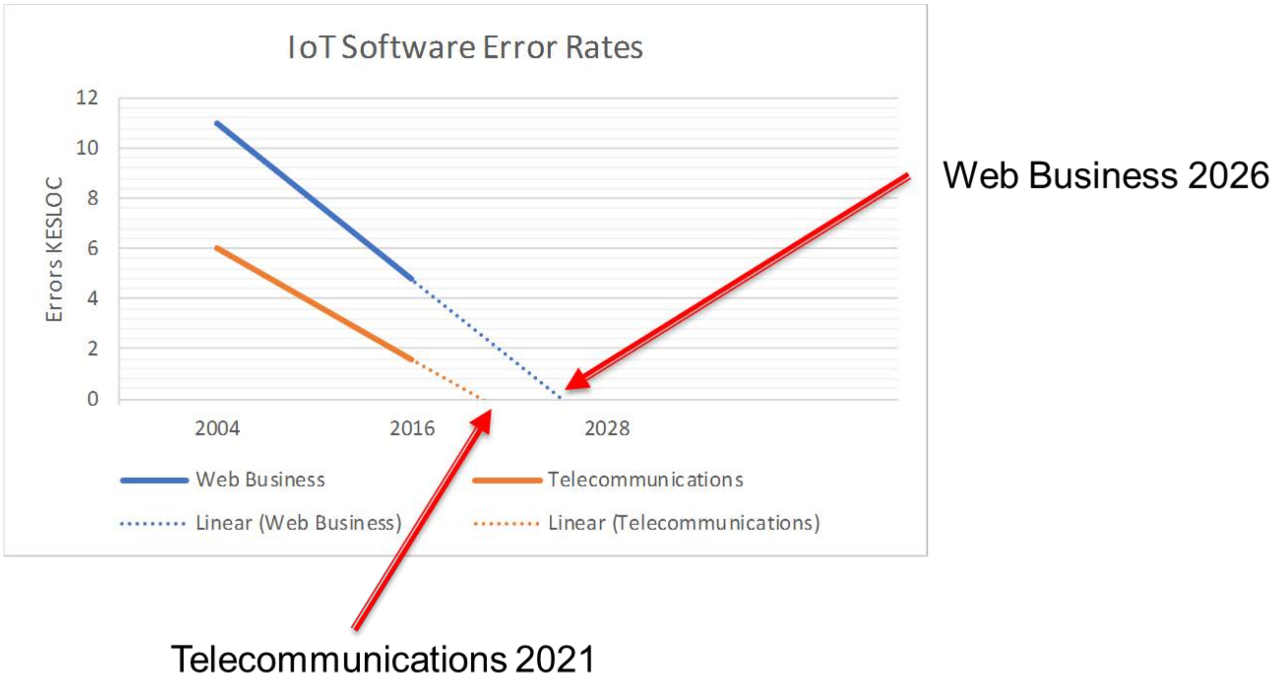
**Average Defects/KESLOC for Commercial - Telecommunications**

| Year | 2006 to 2008 | 2008 to 2010 | 2010 to 2012 | 2012 to 2014 | 2014 to 2016 |
|---|---|---|---|---|---|
| Defects/KESLOC | 7.0 | 6.4 | 5.8 | 5.3 | 4.8 |

**Average Defects/KESLOC for Commercial - Web Business**

Table 15 - 2016 Error Rates after Being Fielded for One year

Note the considerable improvement of these two IoT supportive application domains, commercial-telecommunication dropping from 6 to 1.6 and web business dropping from 11 to 4.8. Some of this improvement is because error rates are now measured after the software has been in the field for one year, whereas in 2004 the error rate was measured at release. While these are good trends and moving in the correct directions, Table 16 indicates Telecommunications Software will not be to mission critical quality levels until  about 2021 and web businesses will not be ready until 2026 unless we accelerate the importance of software quality for IoT. Mission critical quality levels would certainly seem to be appropriate for IoT transportation, medical, and security types of applications.



Web Business 2026

Telecommunications 2021

Table 16 - Trend to Reach Mission Critical Quality Levels

## 12. Specialized Tools Available

Progress of lowering error rates in software for IoT can be accelerated by use of specialized tools available to address the risk areas of IoT. Researching available tools for software development of the type needed for IoT and maintenance of this software yielded a list of 175 tools currently available. The tools are listed in Appendix B by which IoT sub-system component they support, number 1 to 26 in figure 43.

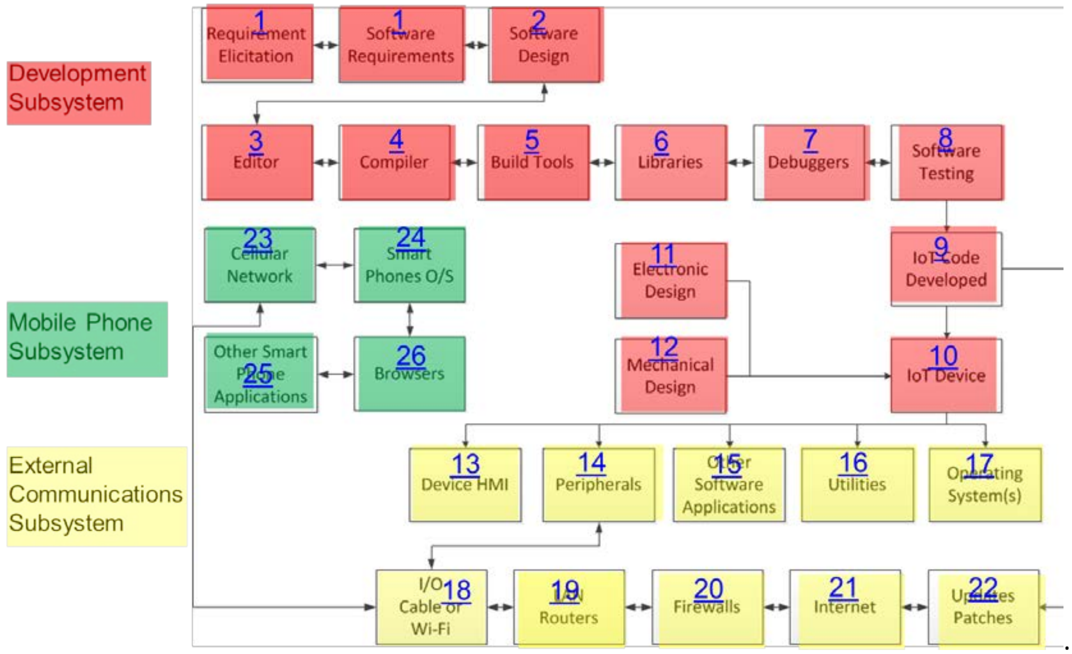


Figure 43 - Index of Specialized Tools Available to Support IoT Software Quality

## 12. IoT Mitigations

Based on the STPA of IoT several mitigations should be considered to reduce the likelihood of cyber vulnerabilities:

1. Do not use inexpensive household routers for IoT applications as the software is buggy.
2. Use strong (AES or better) encryption everywhere
3. Assure IoT systems are immune from jamming devices
4. Replace default factory user names and passwords with strong passwords, change periodically[31]
5. Assure firmware and software updated to latest versions
6. Assure media and updates are from trusted sources
7. Don’t use public recharge stations
8. Don’t use unsecured public wi-fi
9. Alarm if IoT devices are disabled or fail
10. Assure continued update support is part of the purchase price
11. Wait until IoT industry matures
12. Don’t allow remote help desks without confirming source
13. Static and dynamic code analysis on IoT Source and Binary codes
14. Assure IoT applications are security tested and support security with updates
15. Segment home networks (IoT, Personal, Business)
16. Purchase new device every few year for lower cost devices
17. For higher cost devices design the “smart” component hardware to be upgradable.

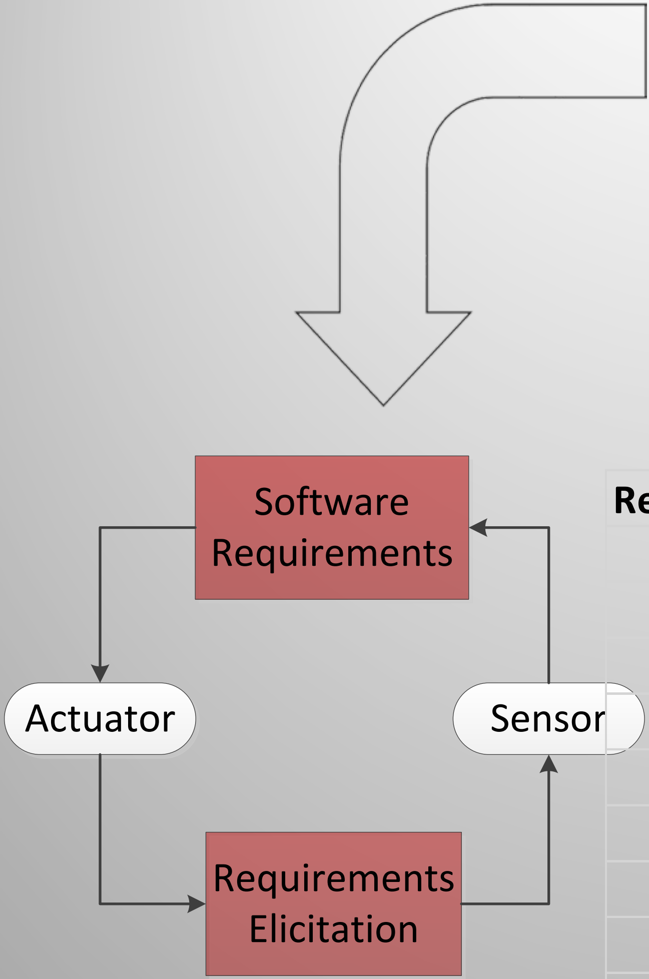
In the detailed STPA for IoT shown in this paper 201 vulnerabilities hazards, and risks were identified (appendices A). However, many of these vulnerabilities were duplicated in multiple components of the subsystems. A surrogate technique was used that identified 30 vulnerabilities, hazards, and risks and permitted is a small enough set of variables to use of automated analysis, if desired. The paper distinguished between prefatory analysis and exhaustive analysis, noting that exhaustive analysis is best suited to automated techniques.  Even the surrogate analysis for IoT ($2^{30}$ x 4 = 4.29E9) would take over a year of continual running on a spreadsheet to try all combination cases, however it could be accomplished in about 20 minutes on a medium scale parallel (1000 nodes) super computer. Larger systems hazard analysis would seem to be a good candidate for large scale computing.

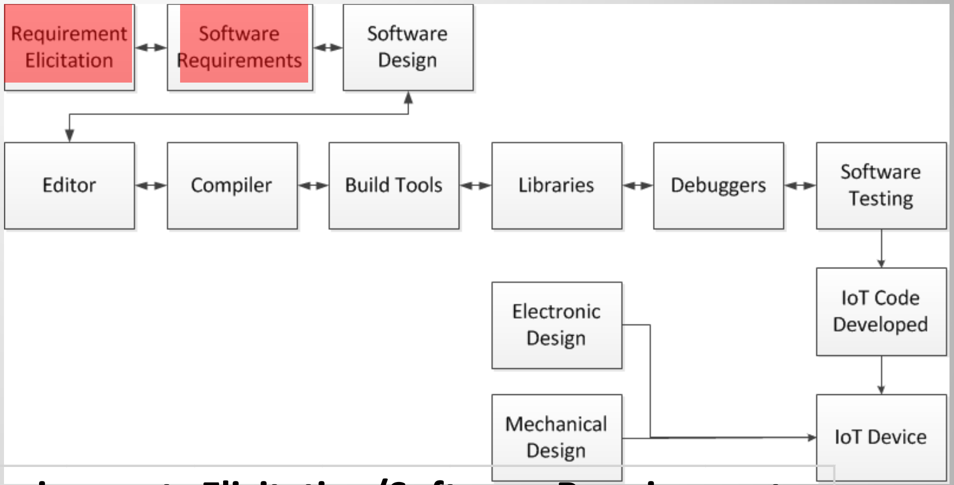
## 13. Conclusion

In conclusion, STPA is a useful addition to other techniques, such as literature searching and expert consultation for conducting analysis on hazards, vulnerabilities, and risks of IoT. The analysis should be included as part of the system development process for software destoned for the IoT and mandatory for applications that involve regulated industries where safety and large financial loss are potential consequences of failure.

# Appendix A -IoT STPA Analysis

# Development Sub-Subsystem Risks



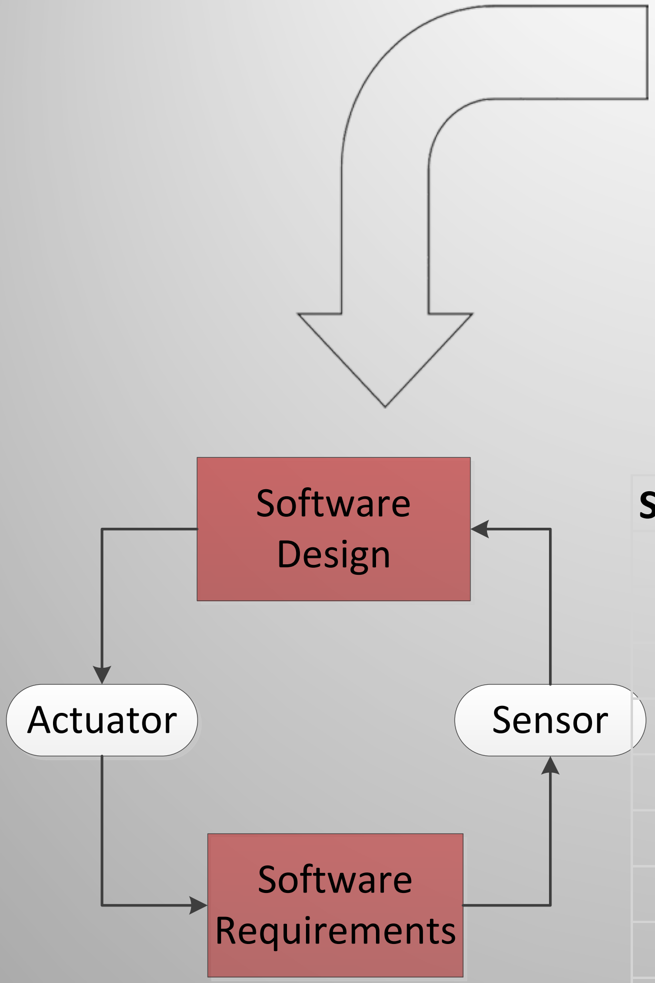


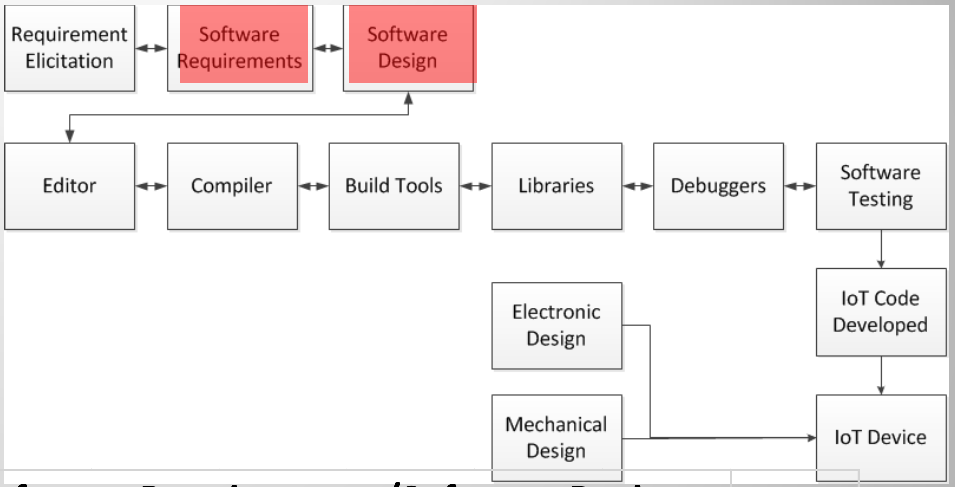
**Requirements Elicitation/Software Requirements**

- **Stakeholder Sample Wrong**
- **Requirement Not Speciific**
- **Requirement Not Measurable**
- **Requirement Not Attainable**
- **Requirement Not Realizable**
- **Requirement Not Time Bounded**
- **Requirement Not Complete**
- **Requirement Not Concise**
- **Approved Before Understood**
- **Jumping to Design**

https://www.linkedin.com/pulse/top-8-mistakes-requirements-elicitation-aaron-whittenberger-cbap

# Development Sub-Subsystem Risks



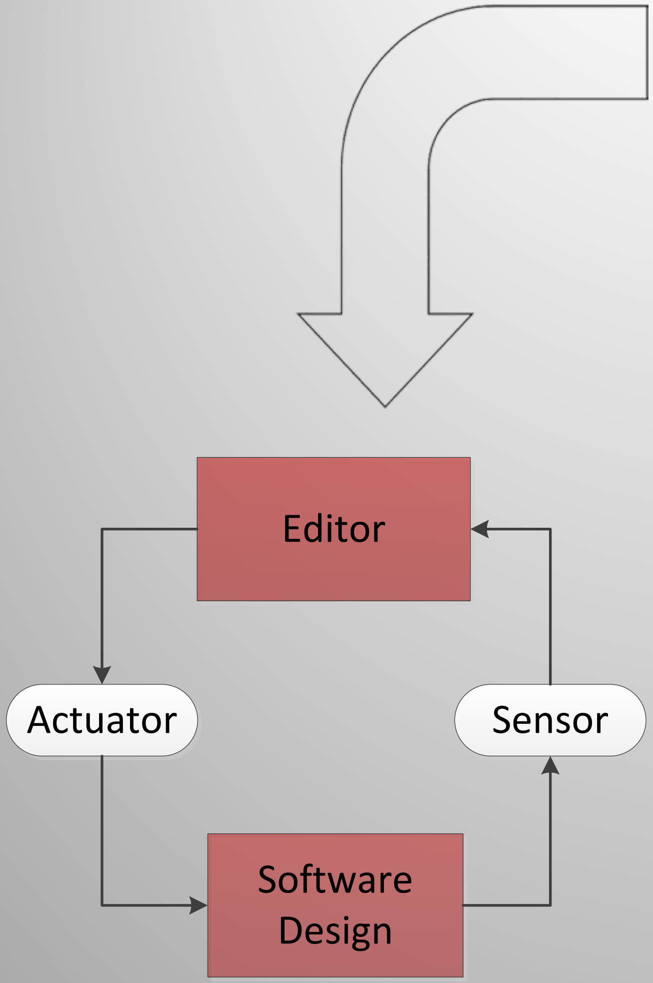


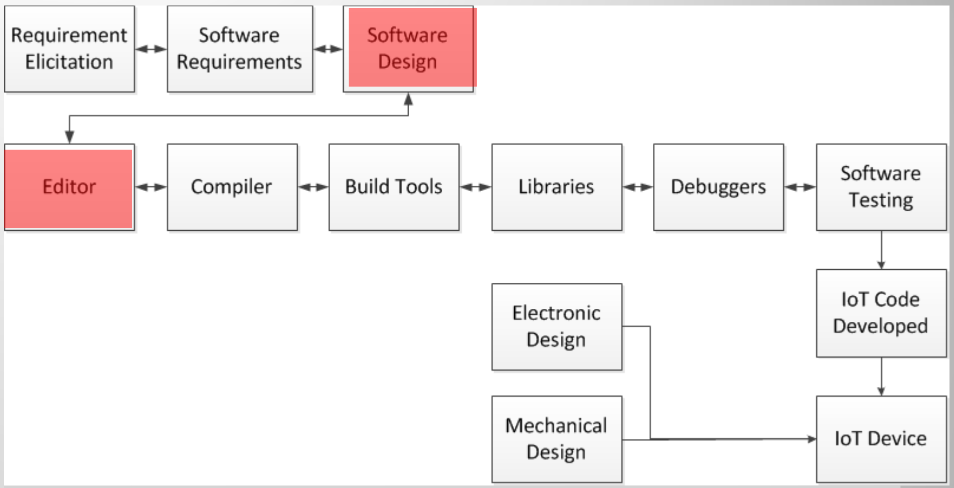
**Software Requirements/Software Design**

- **Missing Functional Requirements**
- **Missing Non-Functional Requirements**
- **Missing Security Requirements**
- **Missing Safety Requirements**
- **Exception Cases Not Handled**
- **Unclear**
- **Unprioritized**
- **Incomplete**
- **Unconsumable**
- **Unreflective of business goals**

http://www.seilevel.com/requirements/5-reasons-software-projects-fail-hint-its-often-due-to-incomplete-incorrect-requirements

# Development Sub-Subsystem Risks



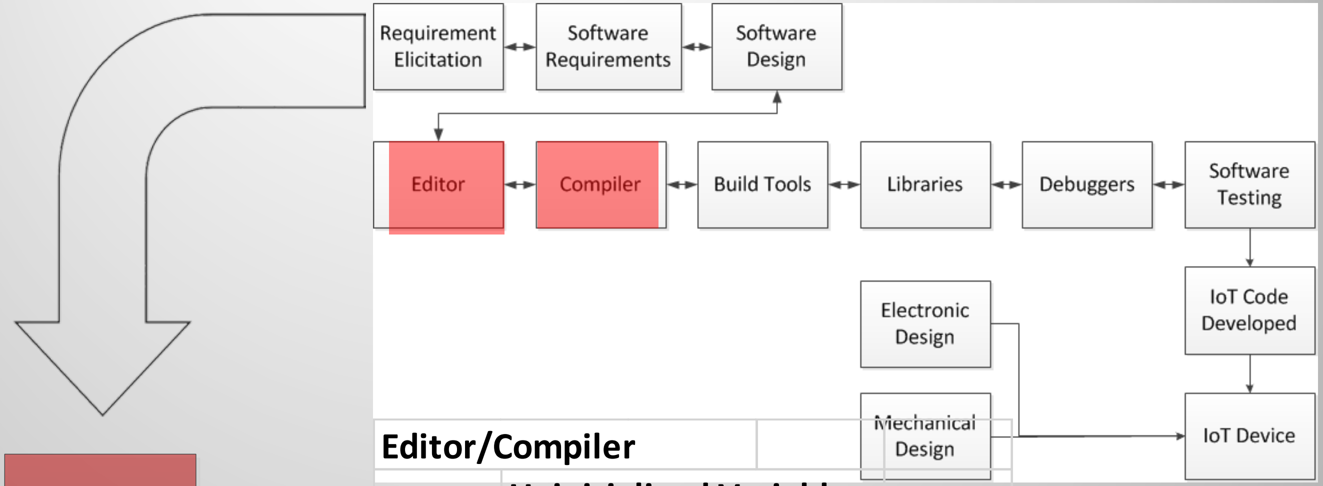


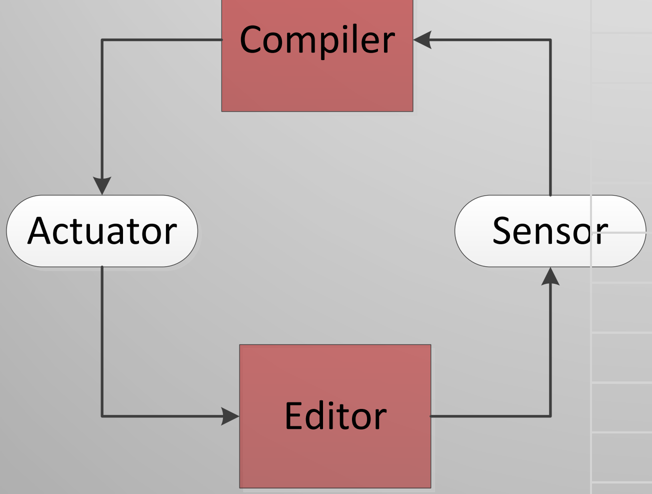
**Software Design/Editor**

- **Lacks Intuitiveness**
- **Weak  Exception Handling**
- **Human Machine Interface not well Defined**
- **Requirement Creep**
- **Weak Scalability**
- **Weak Coupling**
- **Weak Cohesion**
- **Sparse Commenting**
- **Lack of Meaningful Naming**

http://www.pcworld.com/article/146282/article.html

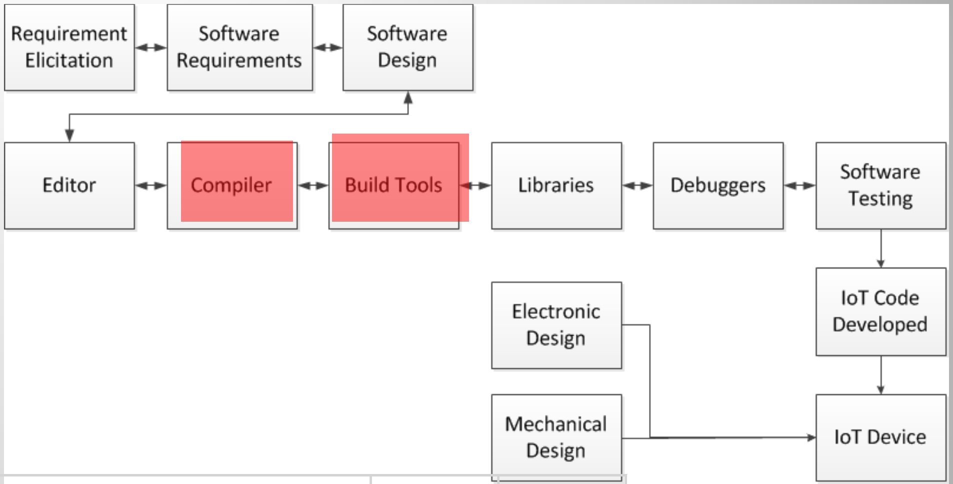
# Development Sub-Subsystem Risks

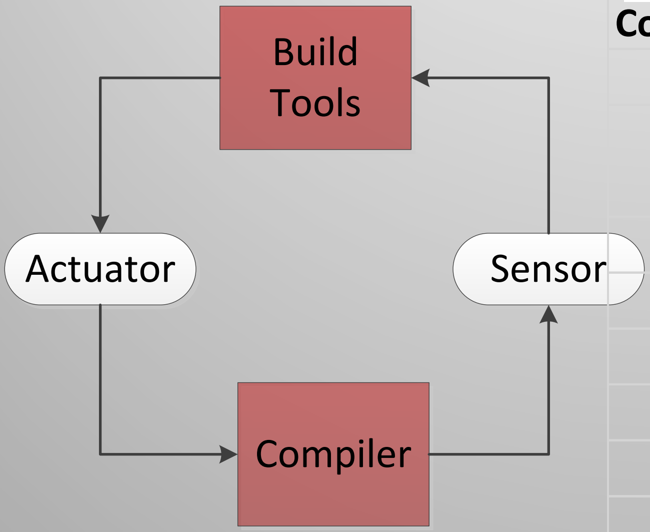




| Editor/Compiler | | | |
|---|---|---|---|
| | Uninitialized Variable | | |
| | Un Released Memory | | |
| | Array overflow | | |
| | Buffer Overflow | | |
| | Tainted Input | | |
| | Null pointer | | |
| | Stale Pointer | | |
| | Unreachable Code | | |
| | Back Door | | |
| | Logic Bomb | | |
| | Missing Requirements | | |
| | Misunderstood Requirements | | |
| | Syntax Error | | |

# Development Sub-Subsystem Risks

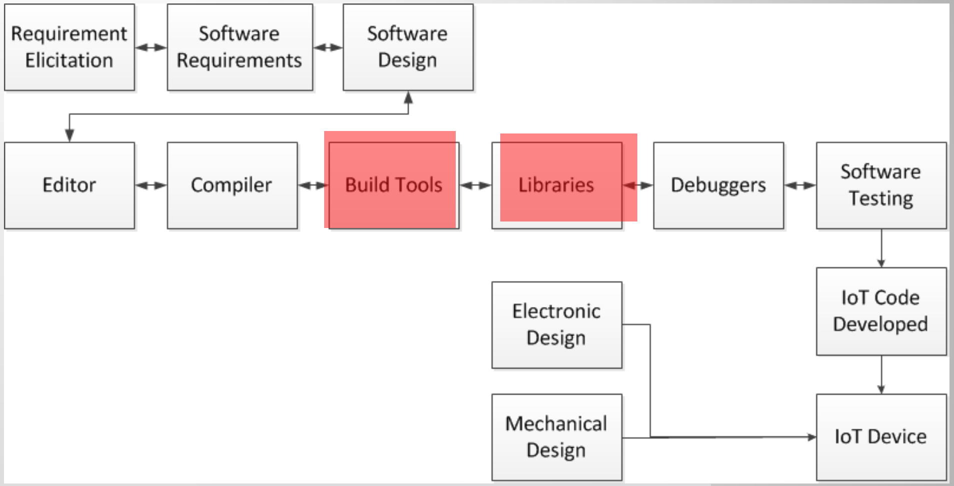


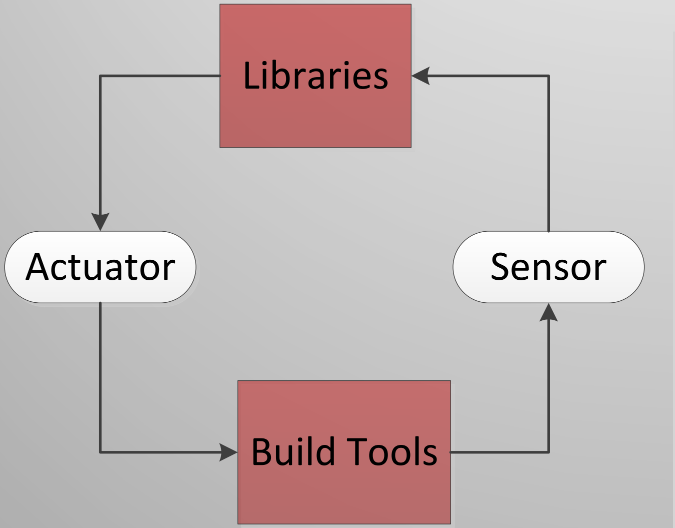


**Compiler/Build Tools**

- **Compiler Warnings Ignored**
- **Wrong Test Baseline**
- **Inadequate Resources**
- **Missing or Wrong Includes**
- **Wrong Options Set**
- **Non-Standard Features Used**
- **Exceptions Not Caught**
- **Unclear  Error Messages**
- **Errors Ignored**
- **Warnings Ignored**
- **Compiler Not Updated**

# Development Sub-Subsystem Risks



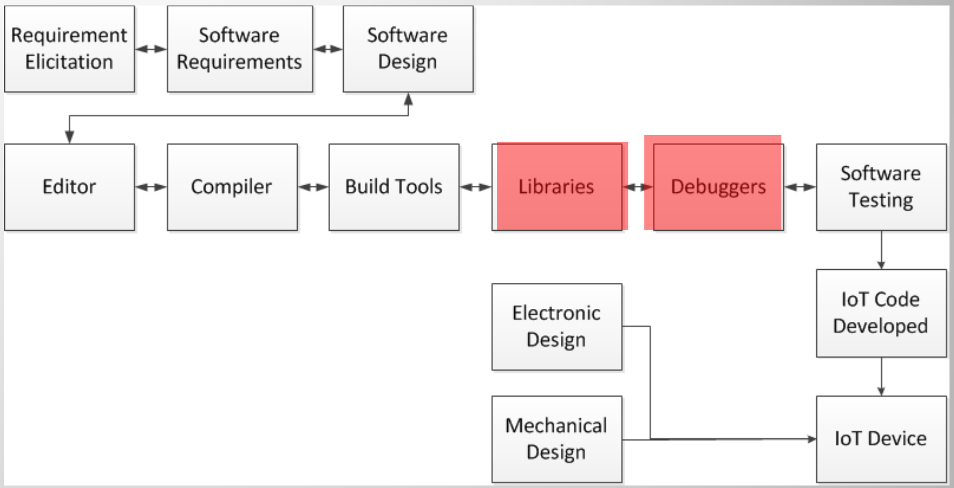


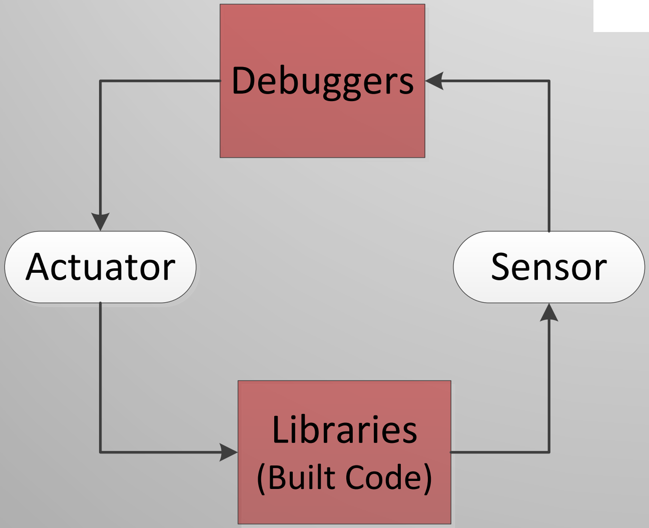
**Build Tools/Libraries**

- **Wrong Library Version**
- **Misnamed Library**
- **Library Vulnerability**
- **Library Not Updated**
- **Library Back Door**
- **Untrusted Supply Chain**
- **Library Not Updated**

https://msdn.microsoft.com/en-us/library/8x5x43k7.aspx

# Development Sub-Subsystem Risks

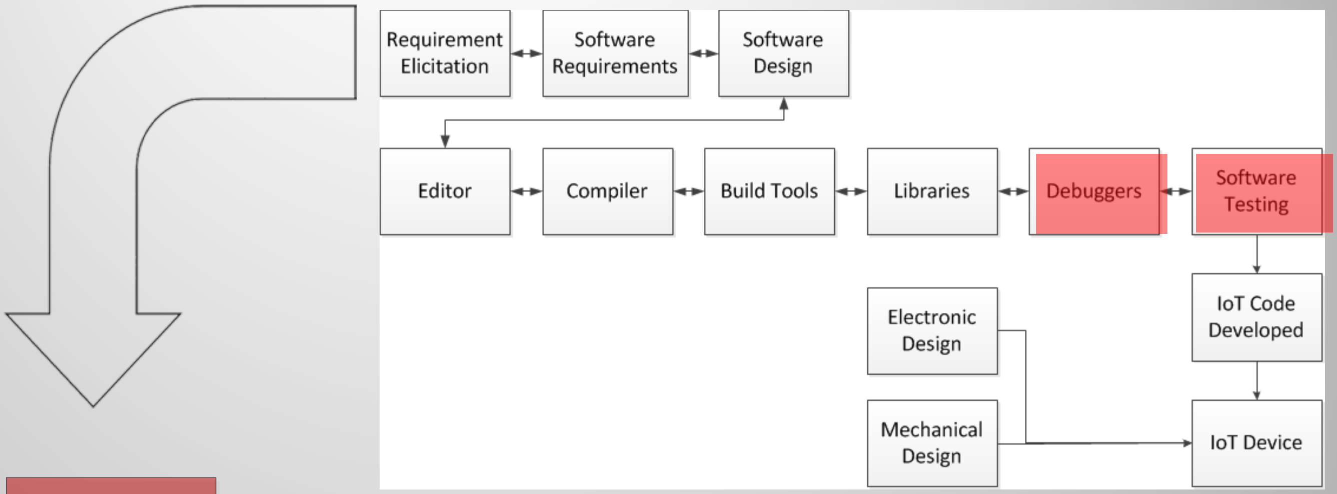


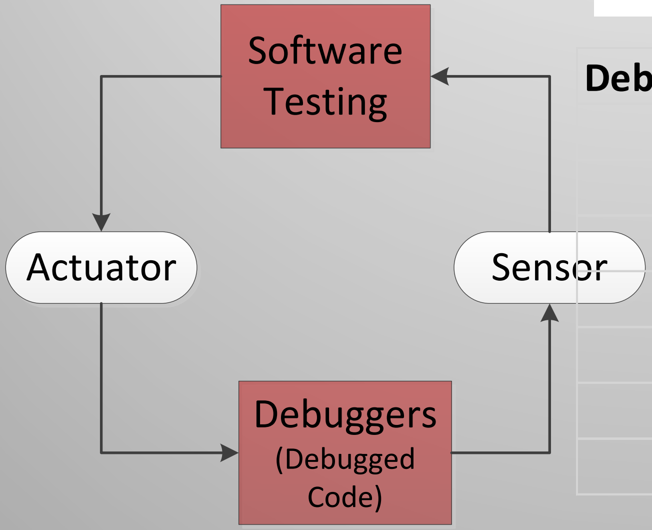


**Libraries/Debuggers**

- **Heisenberg Effect**
- **Diagnostics Residual**
- **Intermittent Problem**
- **Residual Debug Code**
- **Unrealistic Simulators**

# Development Sub-Subsystem Risks

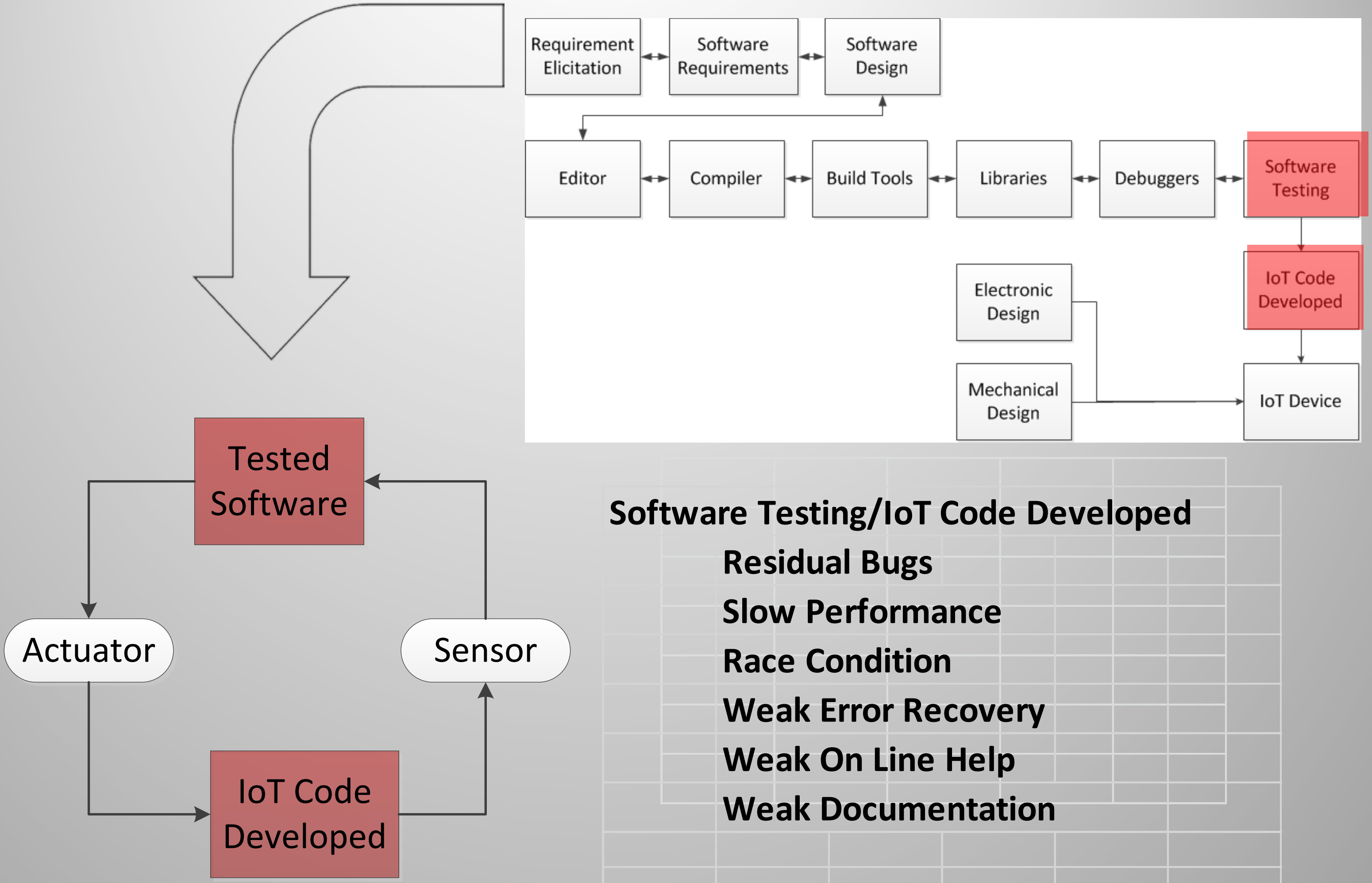




**Debuggers/Software Testing**

- **Low Coverage**
- **No/Few Security Tests**
- **Rushed Testing**
- **No/Few Requirements Traceability**
- **Redundant Testing**
- **No/Few Performance Tests**
- **Weak Platfoirm Testing**

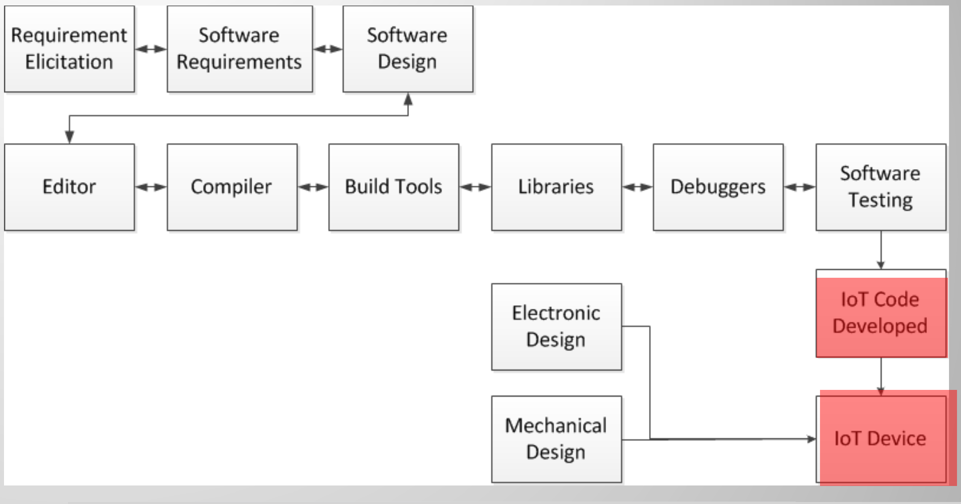
# Development Sub-Subsystem Risks

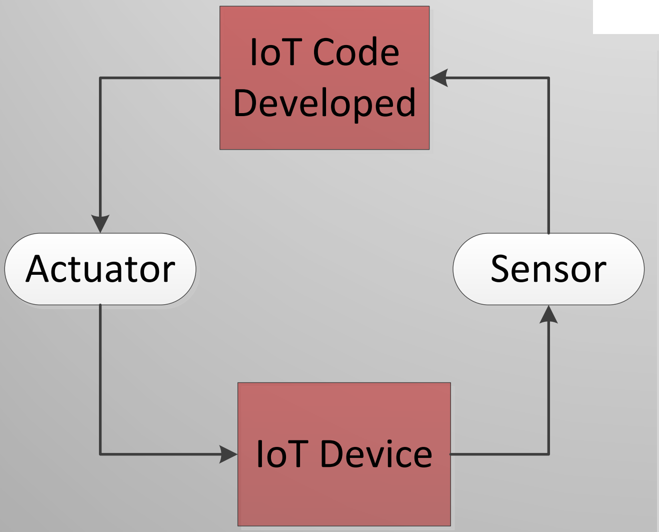


**Software Testing/IoT Code Developed**

- **Residual Bugs**
- **Slow Performance**
- **Race Condition**
- **Weak Error Recovery**
- **Weak On Line Help**
- **Weak Documentation**

# Development Sub-Subsystem Risks

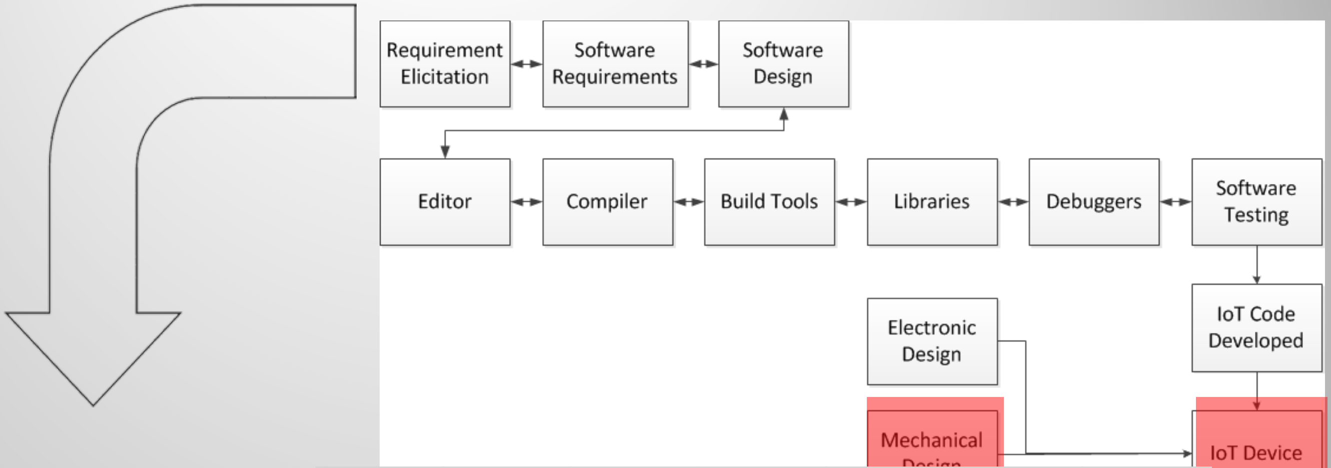


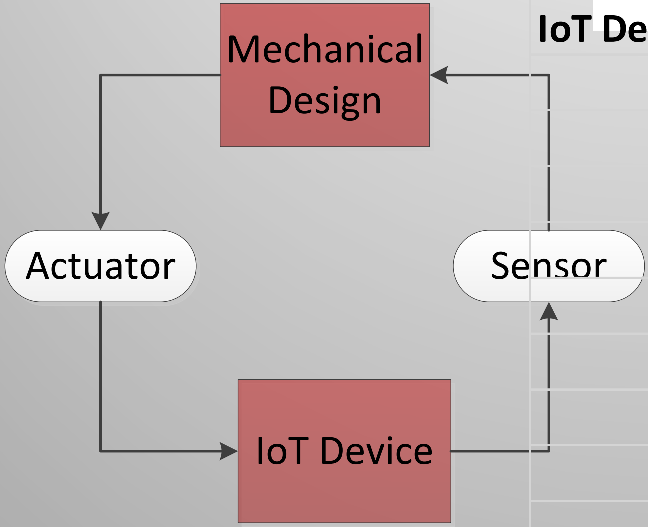


**IoT Code Developed/IoT Device**

- **HW/SW Compatibility**
- **Operating System Vulnerability**
- **Unsupported Operating System Version**
- **Slow Performance**
- **Race Conditions**
- **Confusing Install Procedure**
- **Missing Resources**
- **Tainted Install Media**

# Development Sub-Subsystem Risks

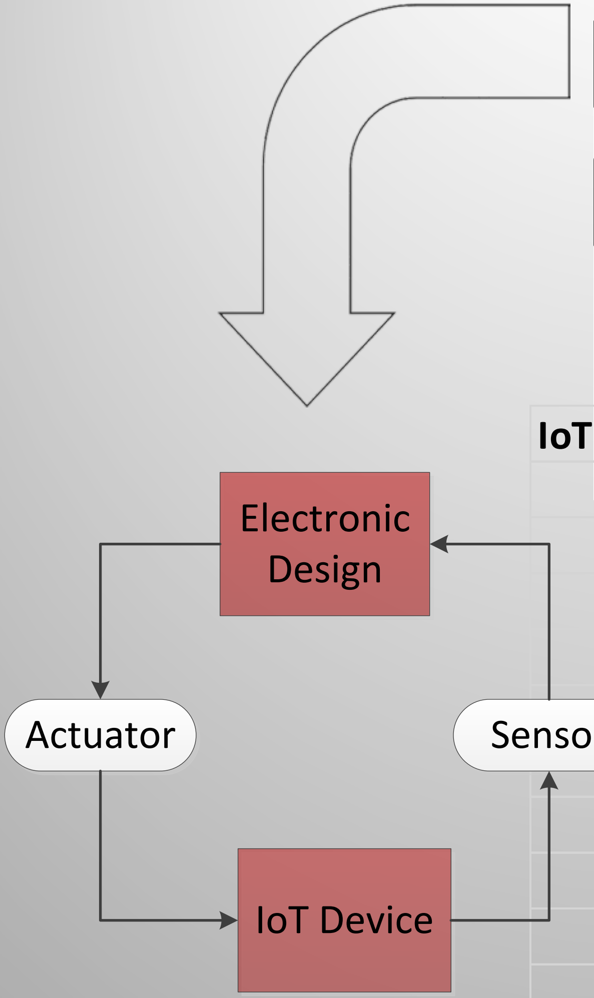


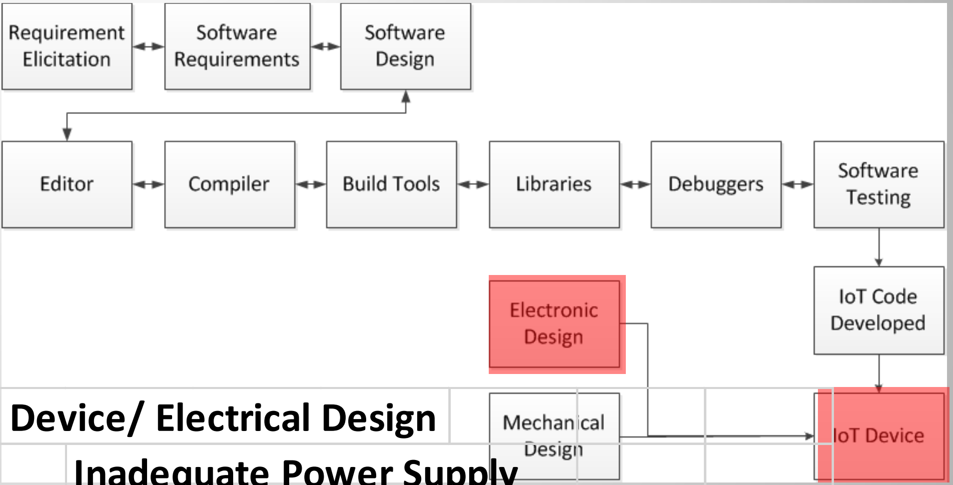


**IoT Device/Mechanical Design**

- **Insufficient Air Flow**
- **Confusing Labeling**
- **Harness Rub**
- **Poor Serviceability**
- **Fan Cooling Needed**
- **EMI/EMC Shielding**
- **Insufficient Vibration Resistance**
- **Insufficient Shock Resistance**
- **Insufficient Environmental Resistance**
- **Manufacturing Defect**
- **Supply Chain Defect**

# Development Sub-Subsystem Risks

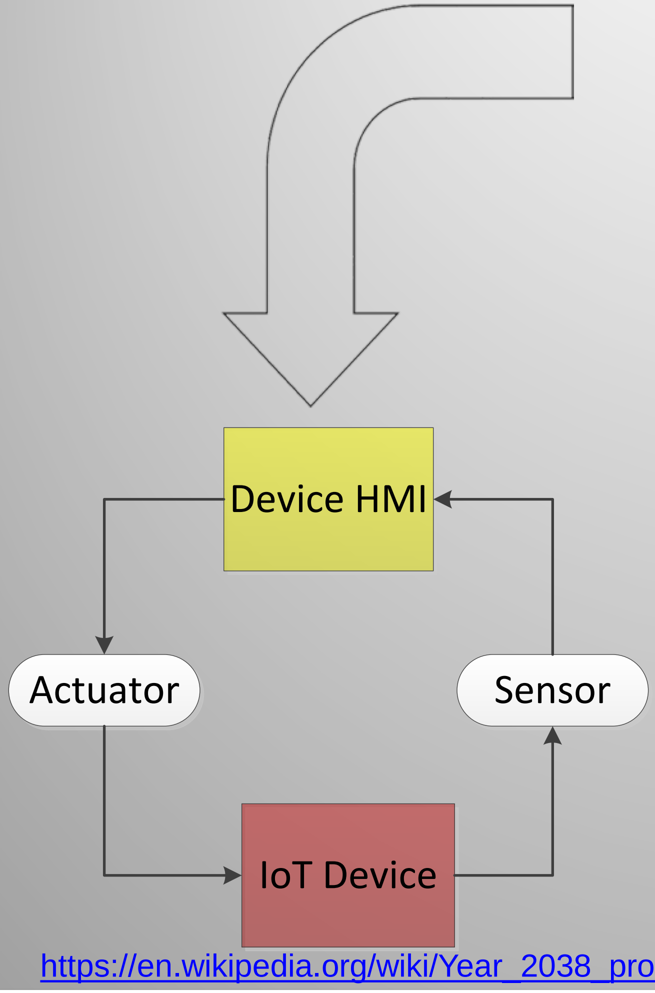


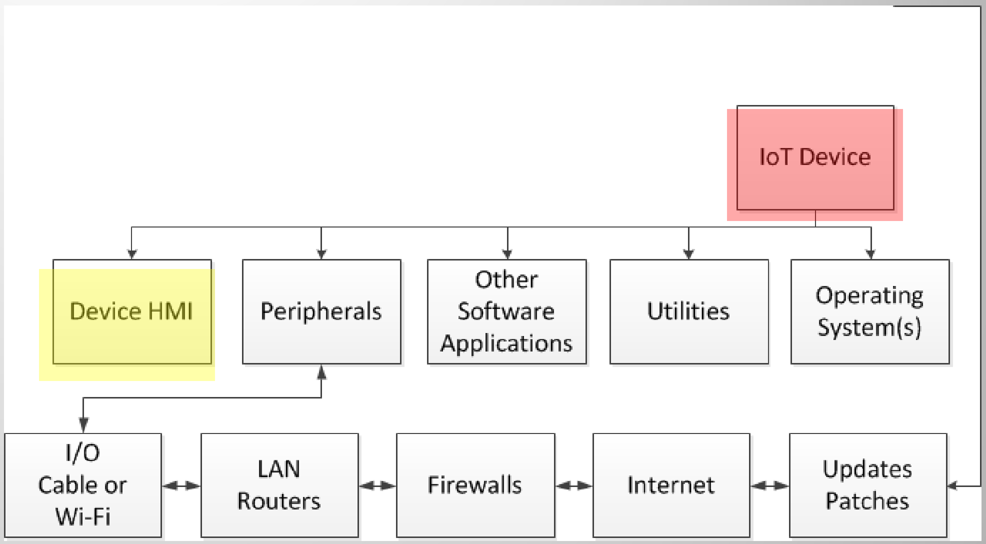


**IoT Device/ Electrical Design**

- **Inadequate Power Supply**
- **Inadequate Component Reliability**
- **Inadequate Surge Protection**
- **Inadequate Noise Suppression**
- **Counterfeit Parts**
- **Slow Processing**
- **Lack of Fault Detection**
- **Inadequate Shock Protection**
- **Inadequate Static Discharge Protection**
- **Version Control Issues**
- **Weak Serviceability**
- **Weak or No Upgrade Ability**

# External Communications Sub-Subsystem Risks





**IoT/Device HMI**

**Confusing User Interface**

**Weak Exception Handling**

**Unclear Error Messages**

**Unlike Incumbent**

**Lack of Documentation**

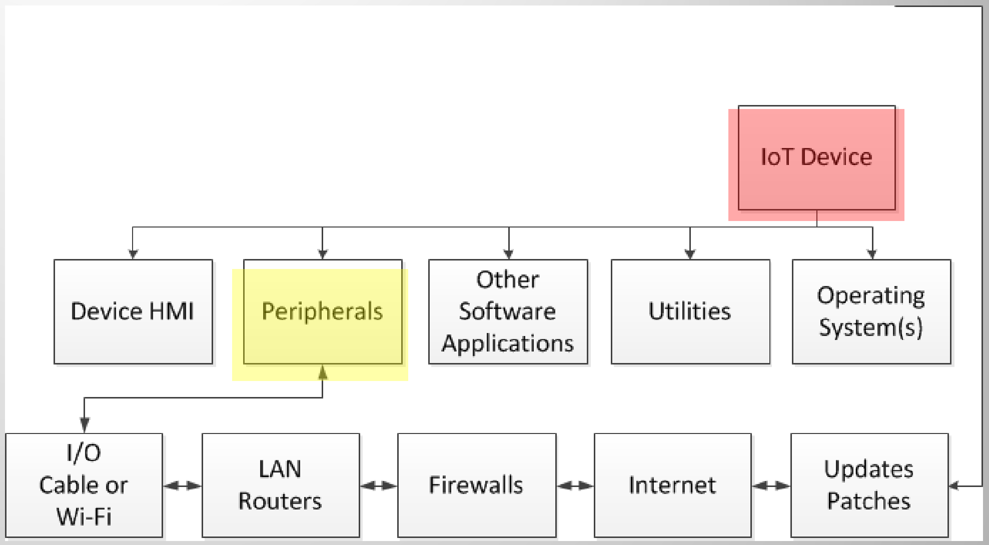
**Confusing Documentation**

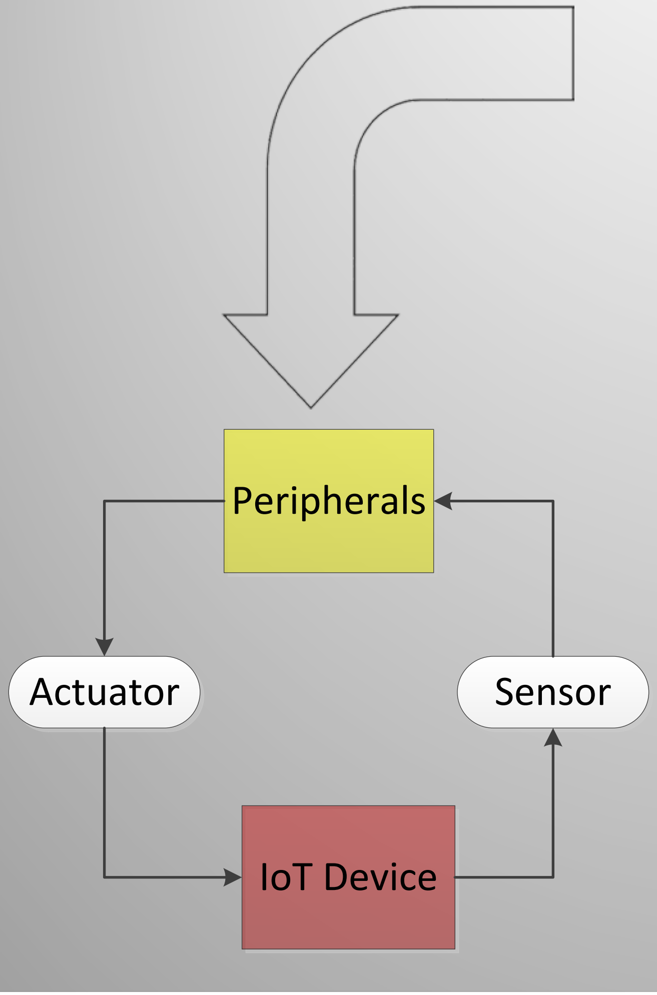
**Secret Codes Remain**

**Unix Jan 19, 2038 Bug /32 bit**

https://en.wikipedia.org/wiki/Year_2038_problem

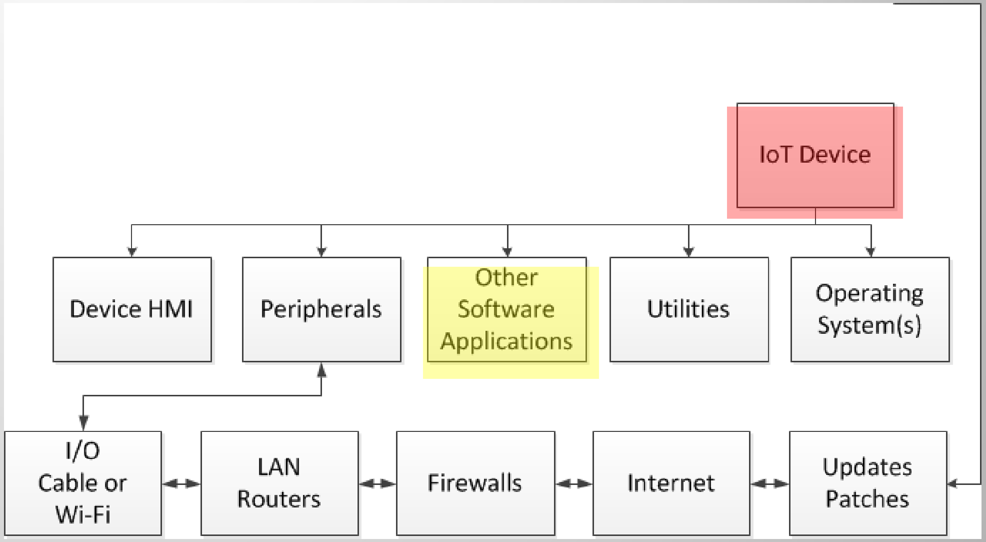
# External Communications Sub-Subsystem Risks

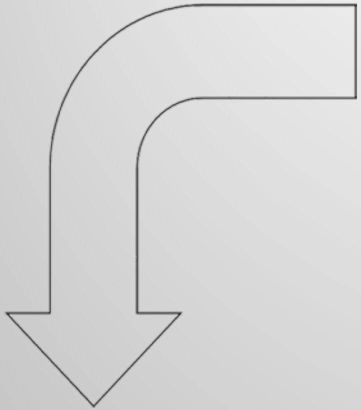


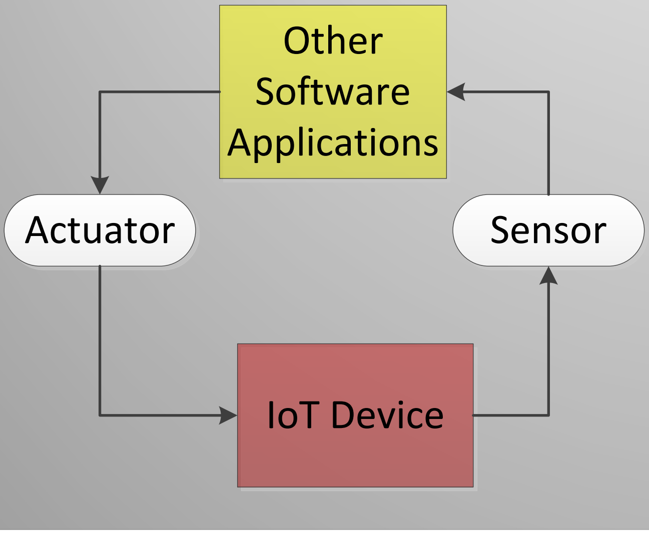


| | |
|---|---|
| **IoT/Peripherals** | |
| | **Current Drivers Not Available** |
| | **Encryption Not Available** |
| | **Inadequate Installation Guide** |
| | **No Guest Device Controls** |
| | **Open Unused I/O Ports** |
| | **Driver Not Updated** |
| | **No Longer Supported** |
| | **Buggy Software** |

# External Communications Sub-Subsystem Risks



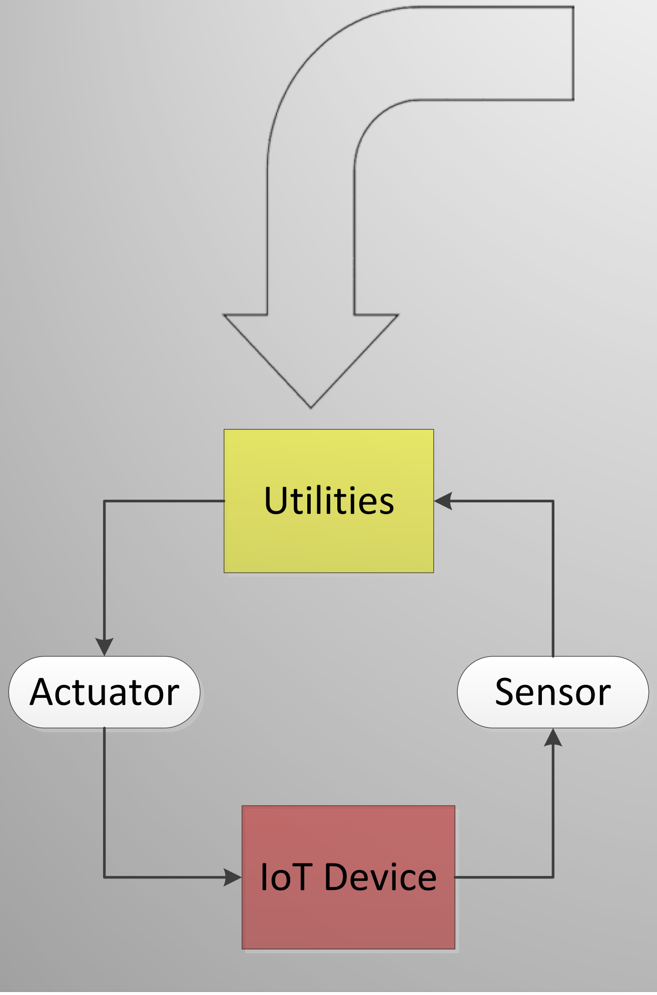


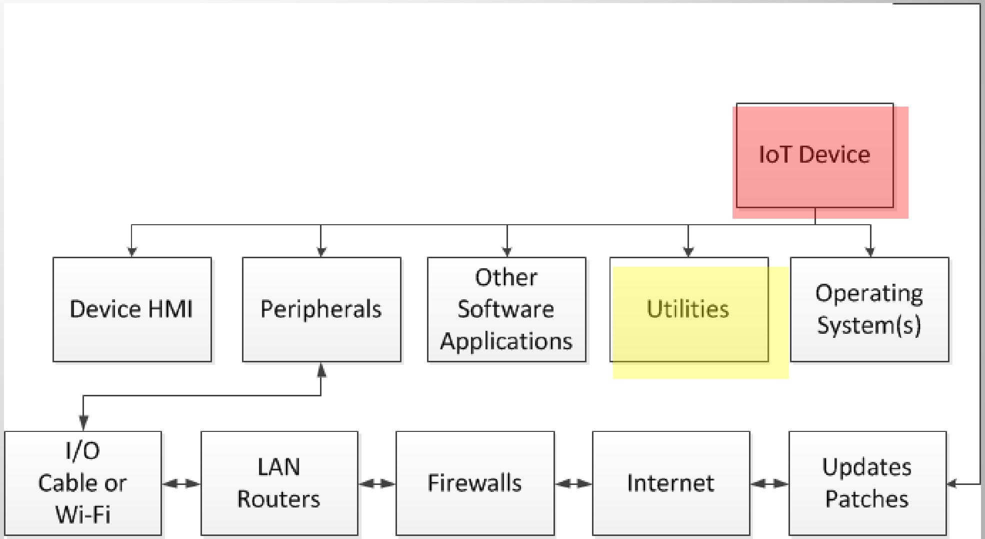


**IoT/Other SW Apps**

- **File Names Not Unique**
- **Versions Not Compatible**
- **Supply Chain Not Secure**
- **Trojans Present**
- **Back Door Present**
- **Not Enough Memory**
- **Rogue Applications**

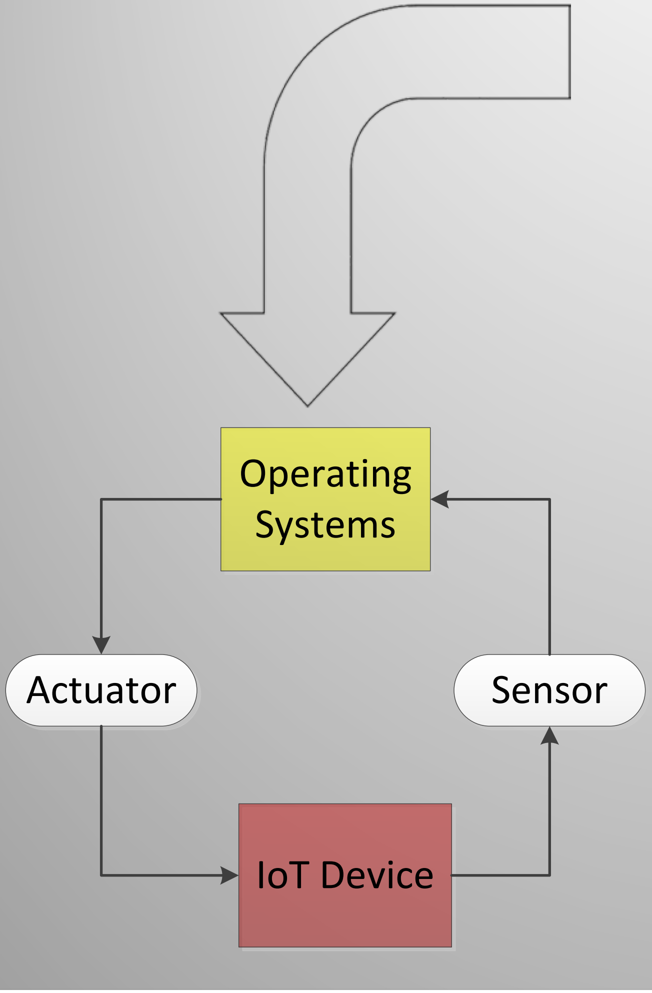
# External Communications Sub-Subsystem Risks

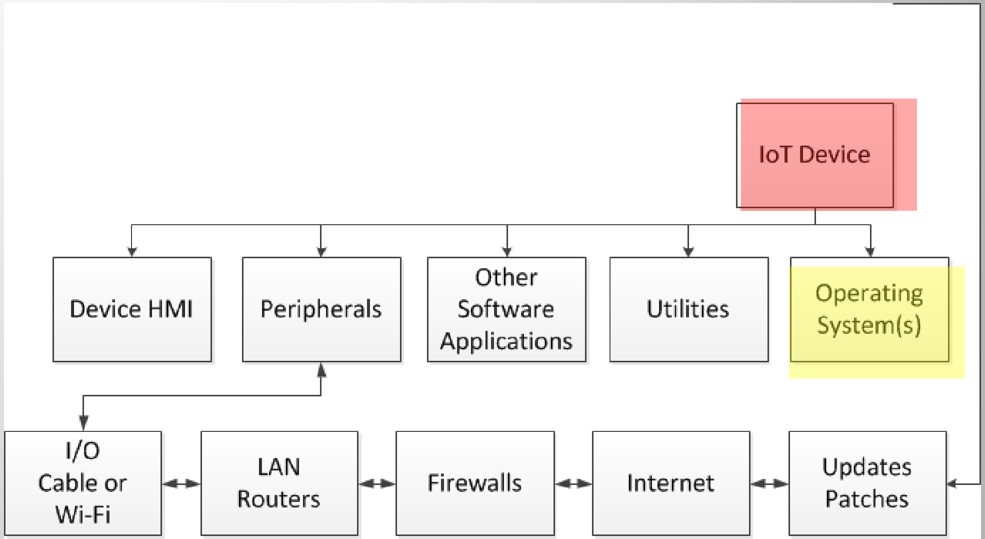




| | |
|---|---|
| **IoT/Utilities** | |
| | **Wrong Version** |
| | **Known Vulnerabilities in Version** |
| | **Version Not Updated** |
| | **Wrong Numerical  Precision** |
| | **Wrong Data Format** |
| | **Counterfeit Item** |
| | **No Upgrade Path** |
| | **Browser Vulnerabilities** |

# External Communications Sub-Subsystem Risks

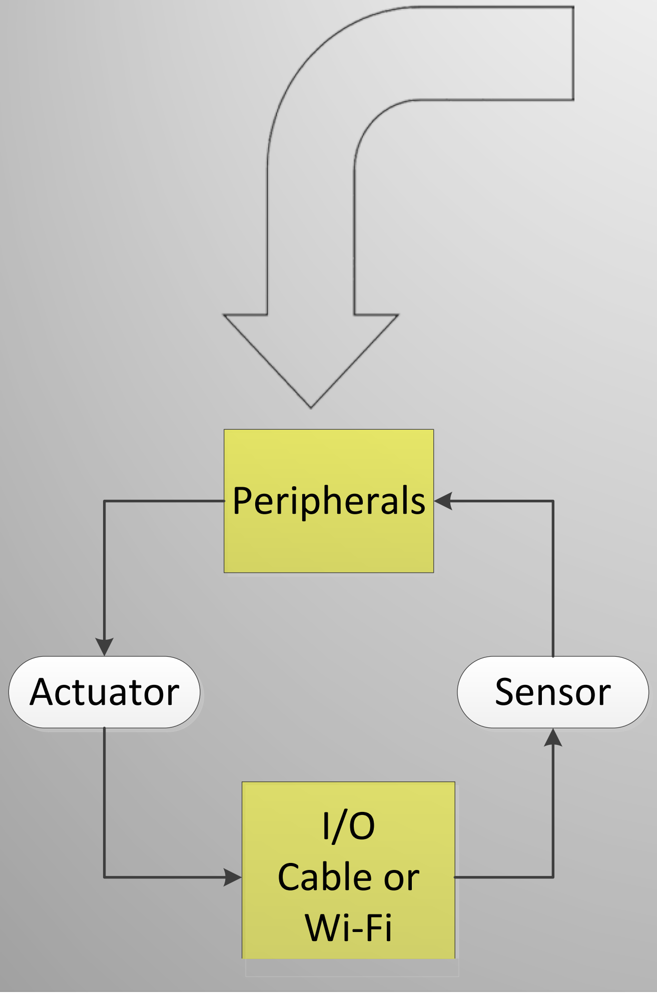


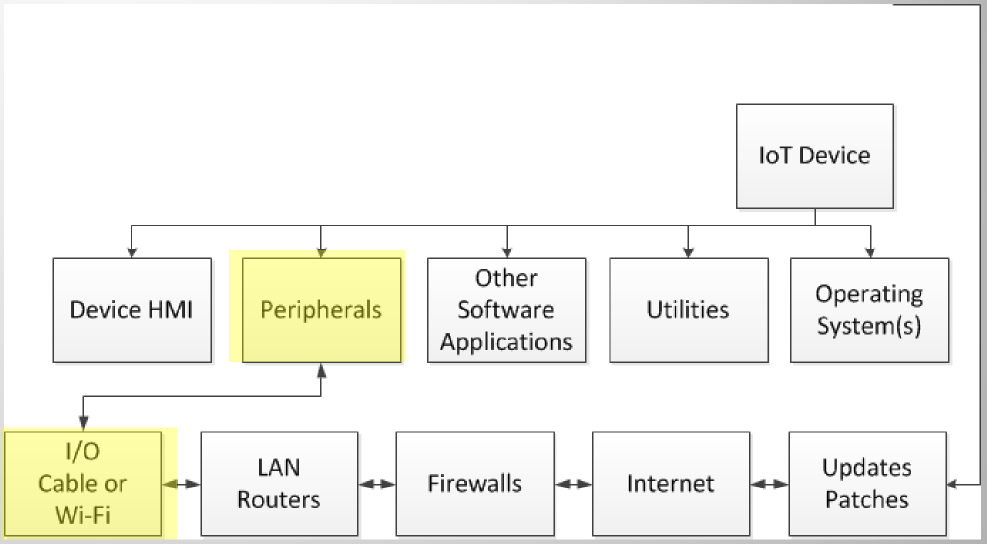


**IoT/Utilities**

- **Wrong Version**
- **Known Vulnerabilities in Version**
- **Version Not Updated**
- **Wrong Numerical  Precision**
- **Wrong Data Format**
- **Counterfeit Item**
- **No Upgrade Path**
- **Browser Vulnerabilities**

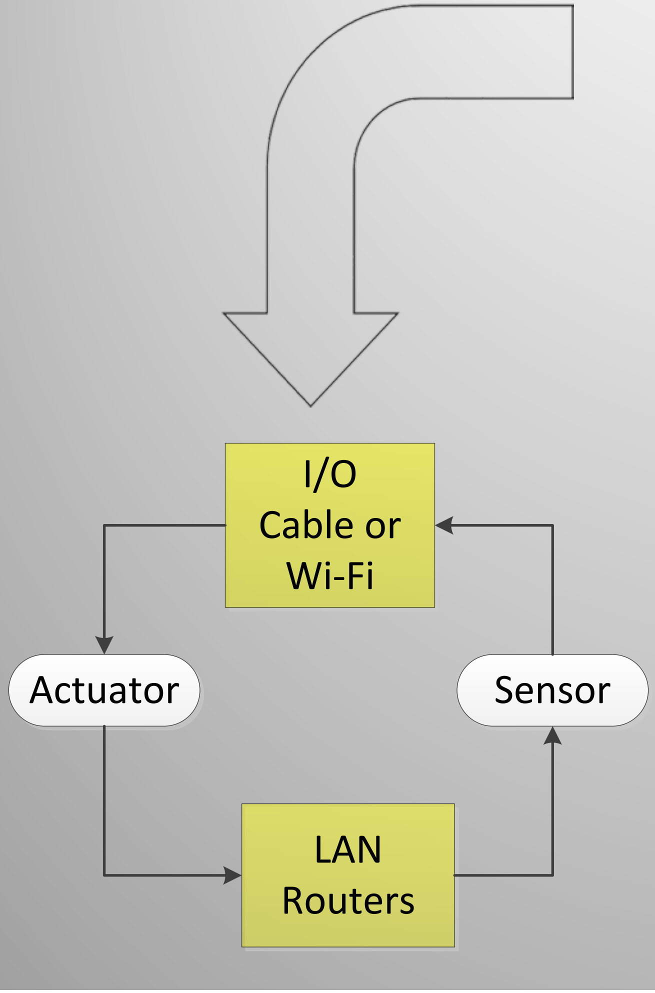
# External Communications Sub-Subsystem Risks

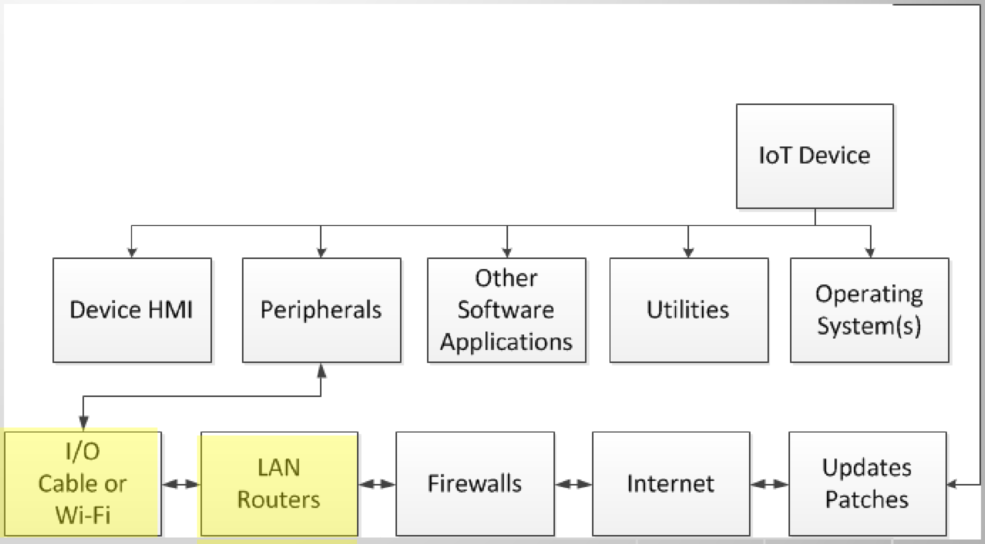




| **Peripheral/ IO Cable or Wi-Fi** | |
|---|---|
| | **Not Encrypted** |
| | **Weak Encryption** |
| | **Network Not Password Protected** |
| | **Open I/O Ports** |
| | **Driver Not Updated** |
| | **Trap Door in Driver** |
| | |

# External Communications Sub-Subsystem Risks

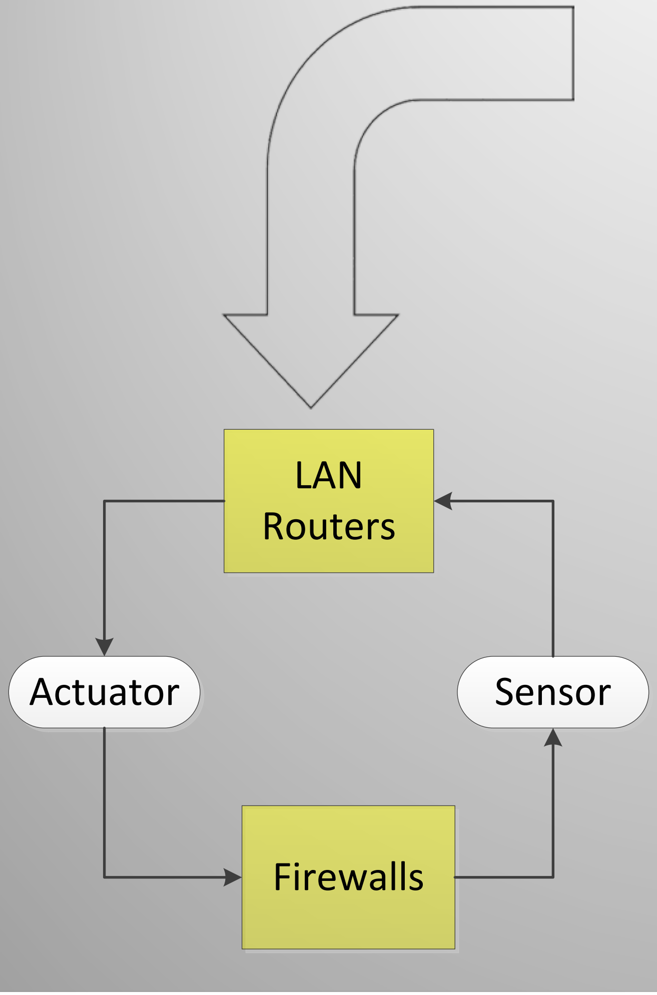


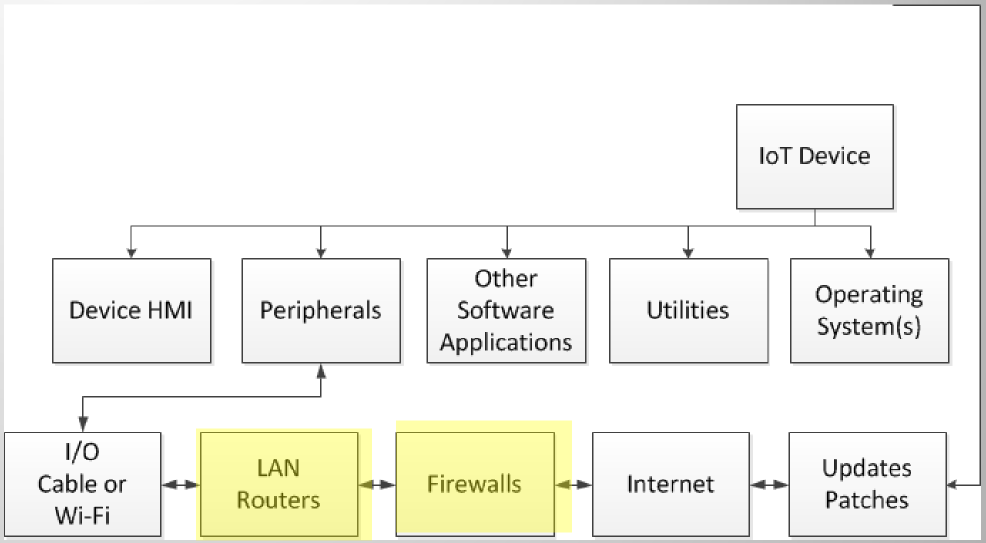


**I/O Cable or Wi-Fi / LAN Routers**

- **No / Weak Encryption**
- **Buggy Router Software**
- **Router Firmware Not Updated**
- **Router Updates No Longer Available**
- **Back Door Left In Router**
- **Updates Contain Virus**
- **Default User and Password**
- **Ports Left Open (7547)**
- **Services Exposed TR-64 -69**

# External Communications Sub-Subsystem Risks

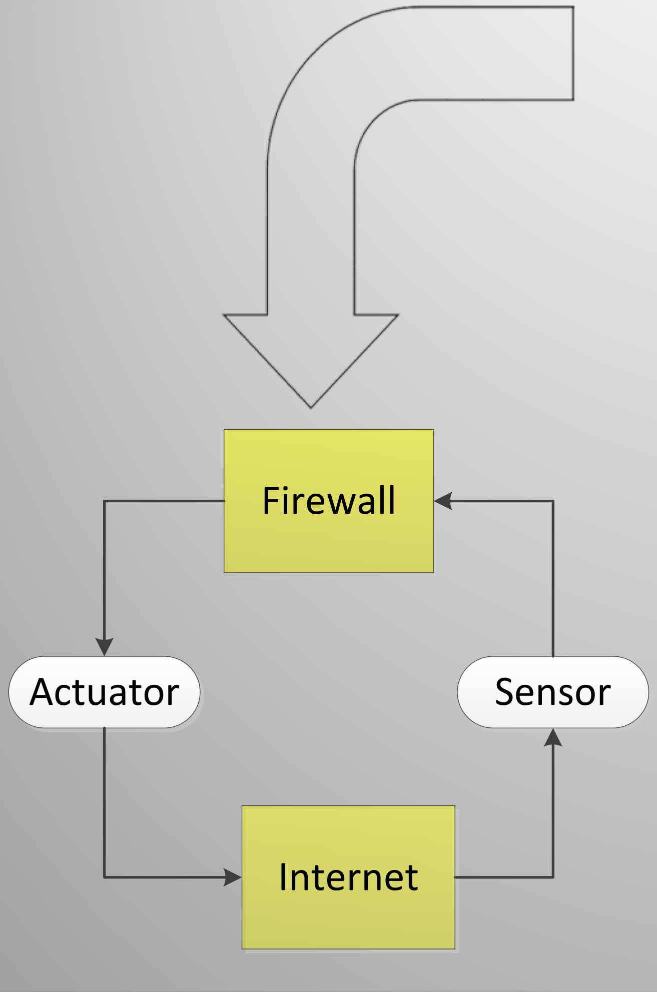


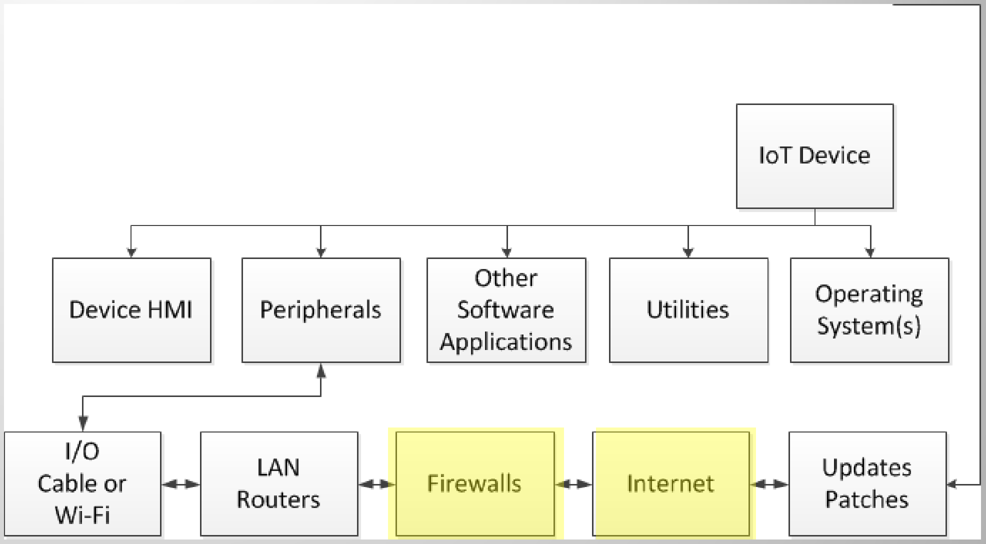


**LAN Routers/Firewall**

- Ports Left Open
- Wrong Security Policy
- Firewall No Longer Supported
- Firewall Contains Back Door
- Capacity Insufficient
- VPN Not Available
- Not Updated

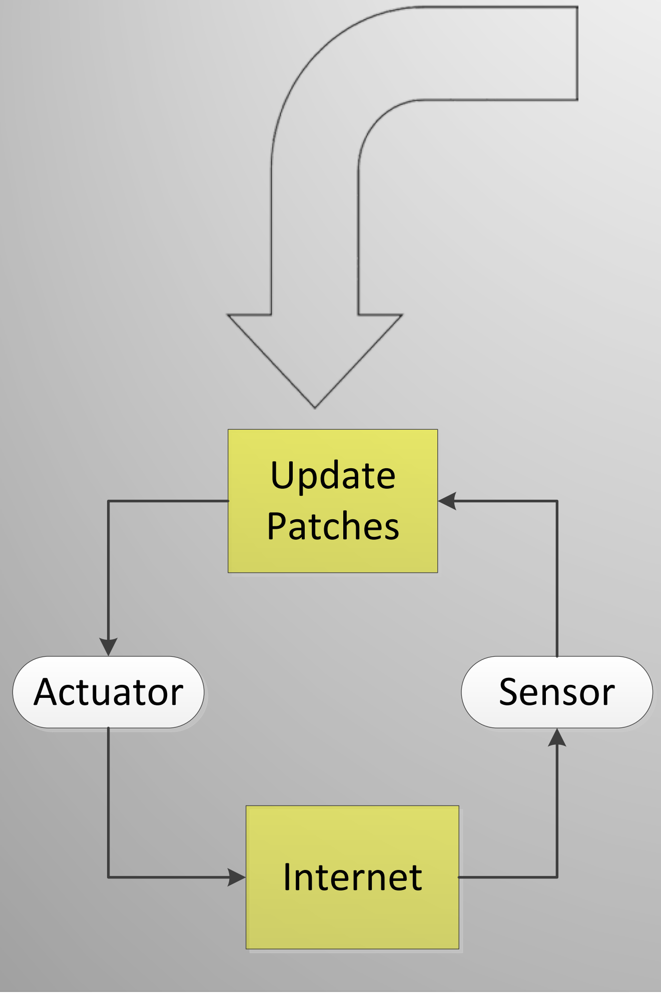
# External Communications Sub-Subsystem Risks

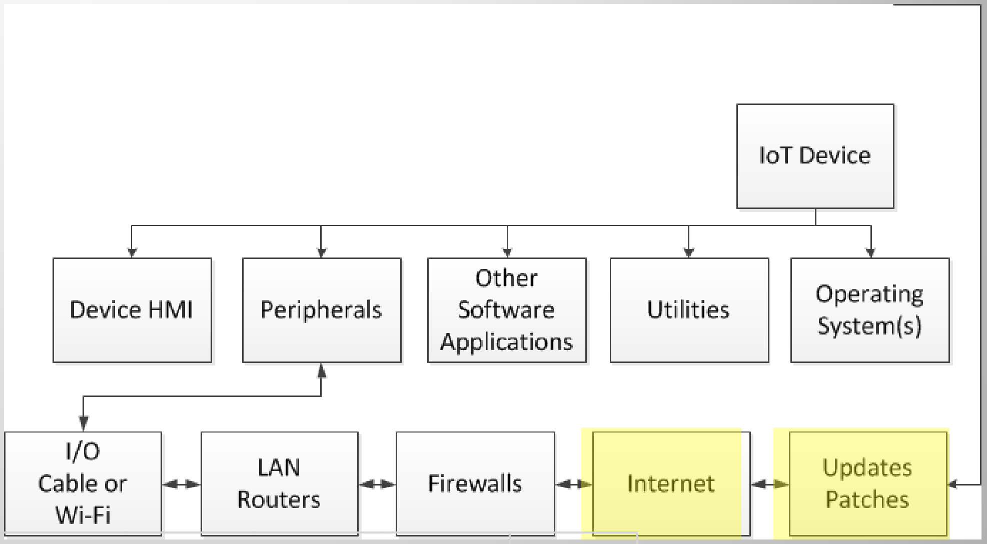




| | | | | |
|---|---|---|---|---|
| **Firewall/Internet** | | | | |
| | **Capacity Insufficient** | | | |
| | **VPN Not Available** | | | |
| | **No Port Scanning Detectors** | | | |
| | **Required Port Blocked** | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# External Communications Sub-Subsystem Risks

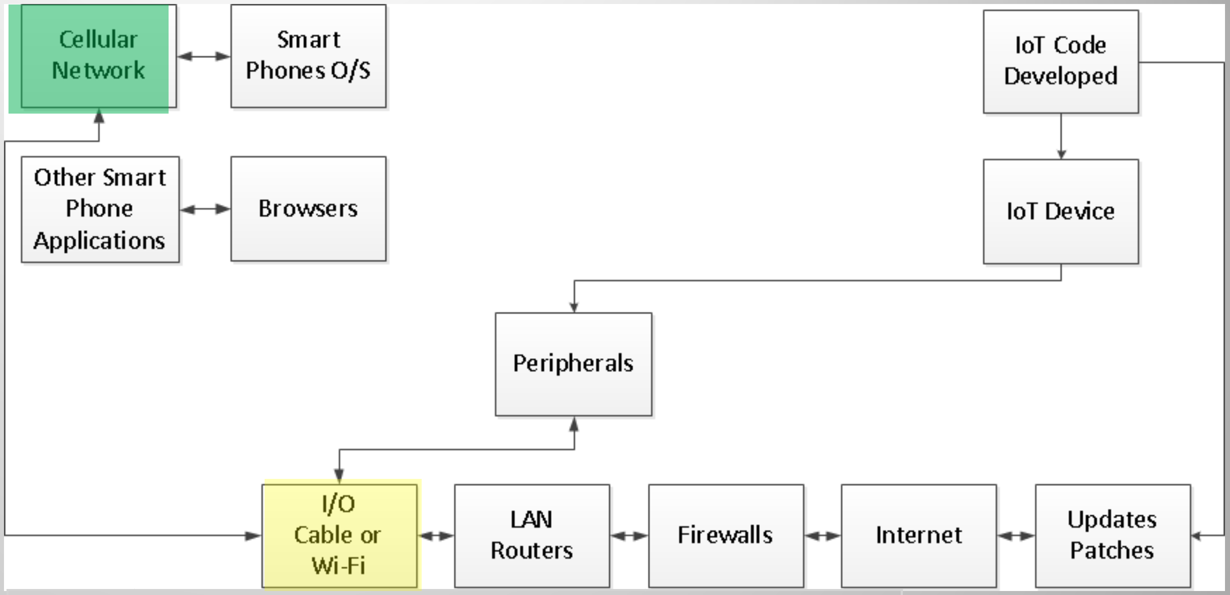


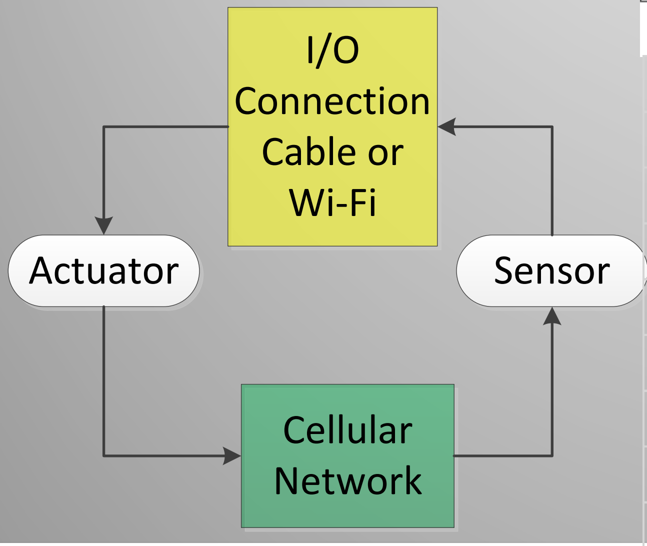


**Internet/Update Patched**

- **Patch Infected With Virus**
- **Patch Has Vulnerability**
- **Device No Longer Supported**
- **Updates Not Done**

# Mobile Phone and Update Sub-Subsystem Risks

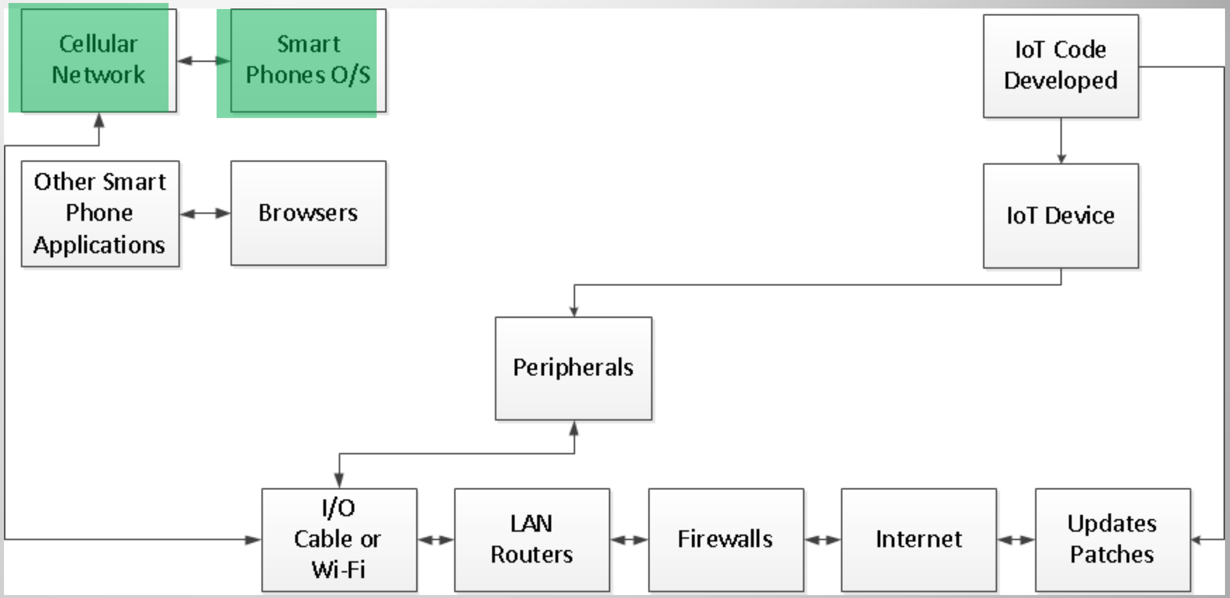


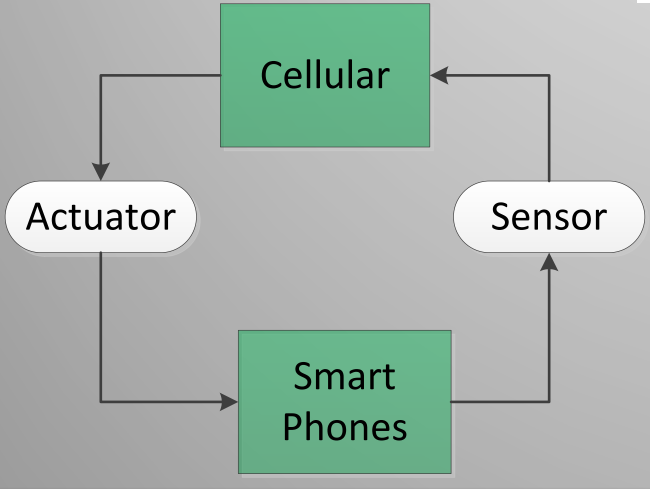


**I/O Connection Cable or Wi-Fi/Cellular Network**

- **Not Encrypted**
- **Inadequate Encryption A5/1, A5/2**
- **Backbone Vulnerabilities SS7**
- **Backdoors**
- **Public Unsecured Wi-Fi**
- **Operating System Vulnerabilities**
- **O/S Not Upgraded**

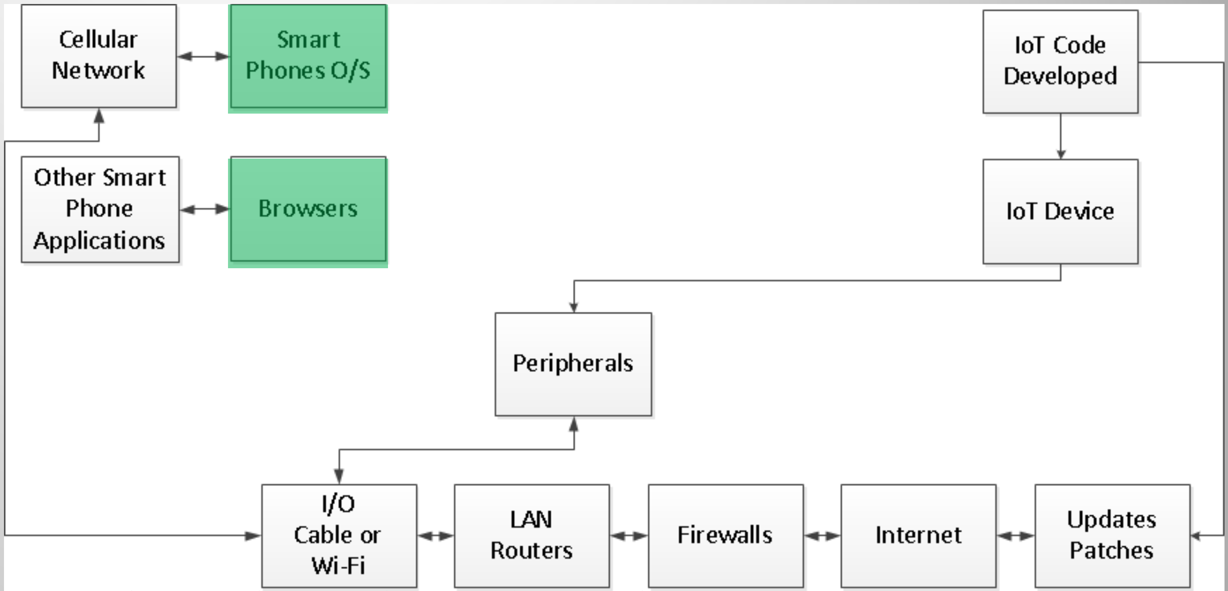
# Mobile Phone and Update Sub-Subsystem Risks

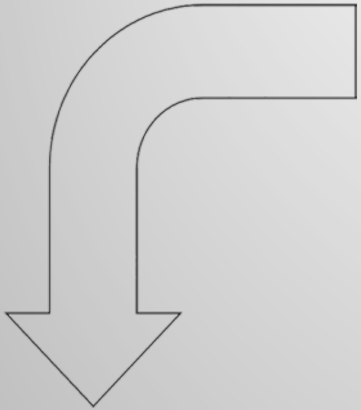


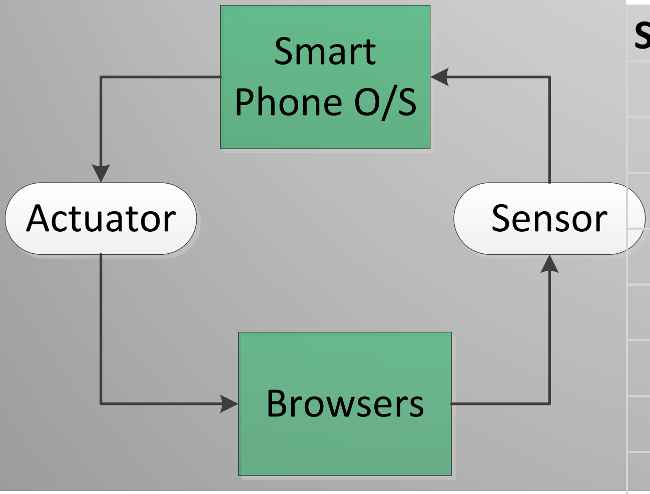


| Cellular/Smart Phones | | | |
|---|---|---|---|
| | Juice Jacking | | |
| | Faux Cell Tower | | |
| | | | |
| | | | |
| | | | |

# Mobile Phone and Update Sub-Subsystem Risks



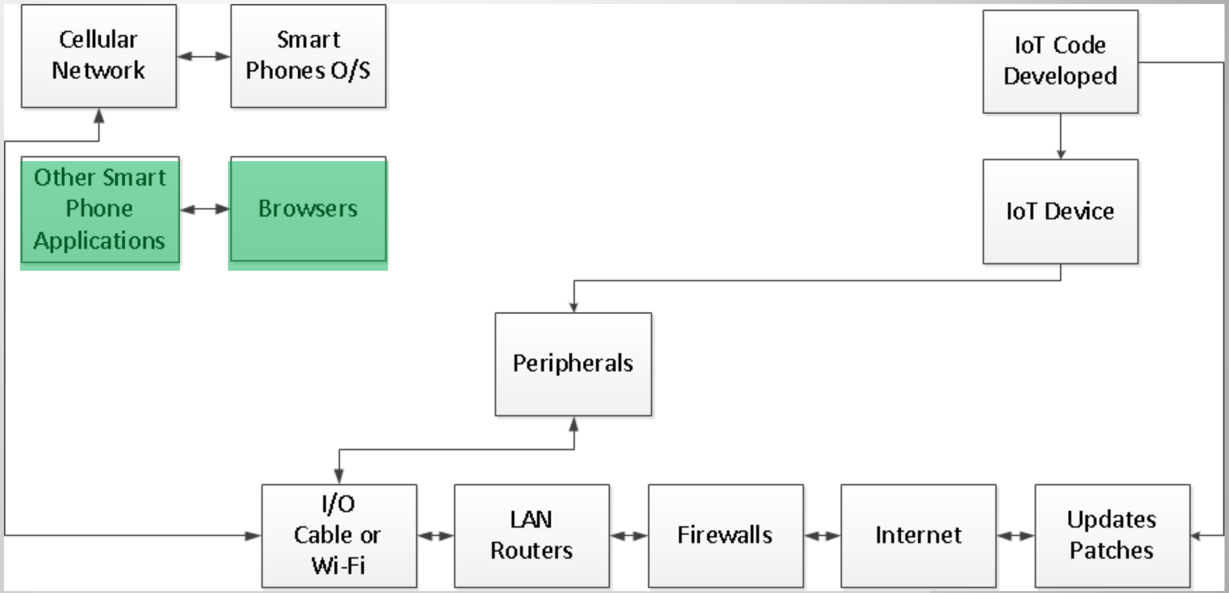


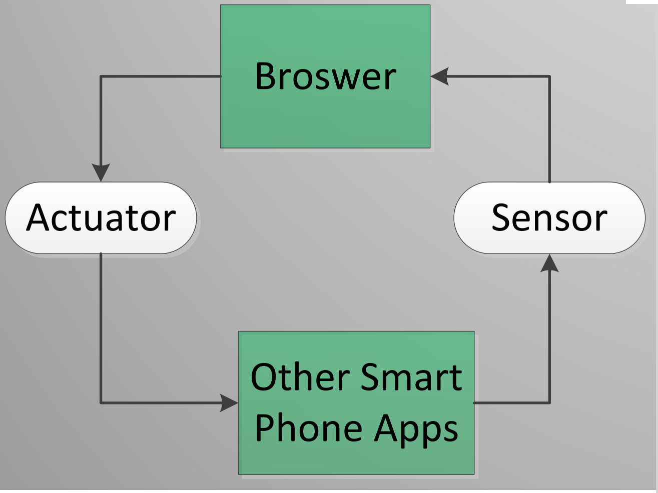


**Smart Phones/Browser**

- **Lack of Frame Busting, Tap Jacking**
- **Vulnerable Mobile Site**
- **Phishing**
- **Counterfeit Address Bar**
- **Unsecured Hot Spots**
- **Unintelligible Error Message**

# Mobile Phone and Update Sub-Subsystem Risks





**Browsers/Other Apps**

**Not Downloaded From Trusted Source**

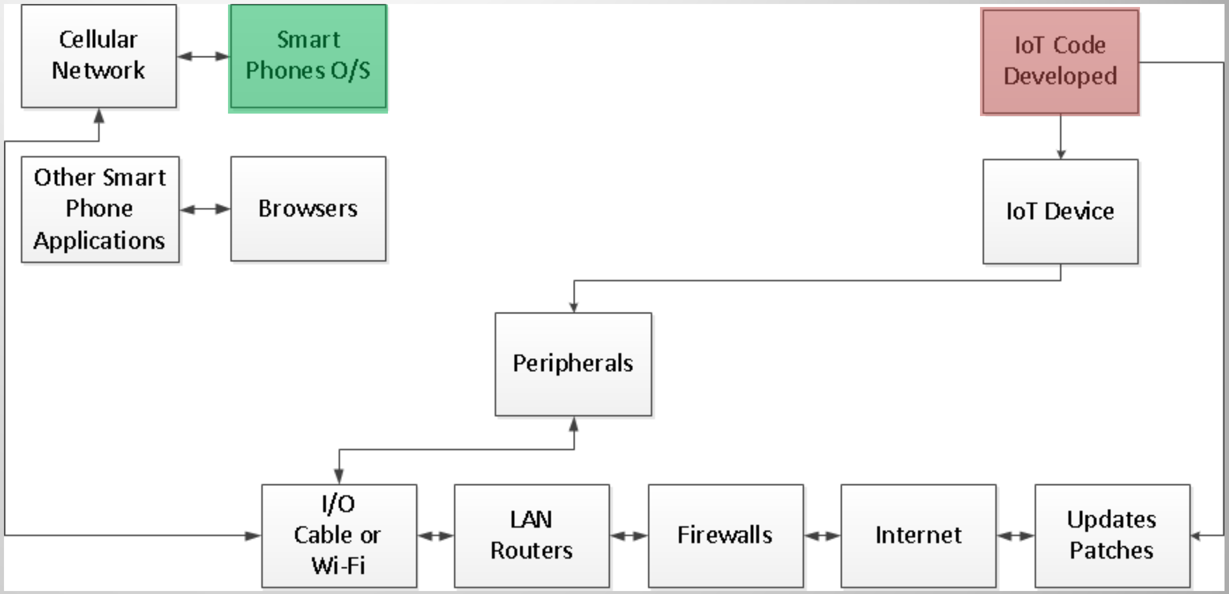
**Malicious Apps**

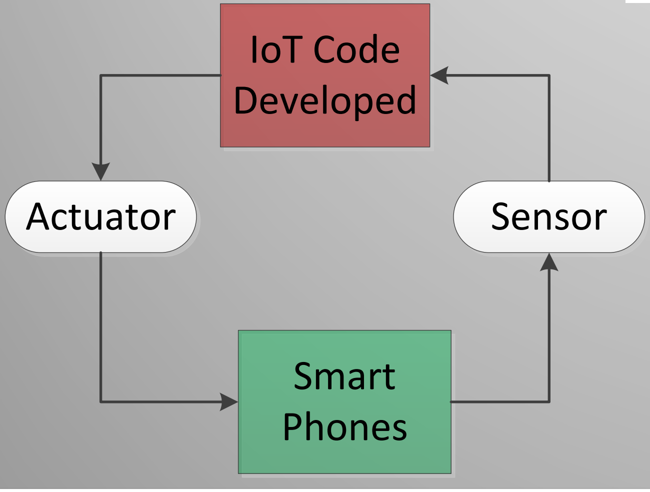
**Apps Not Updated**

**App No Longer Supported**

**Apps Not Updated**

# Mobile Phone and Update Sub- Subsystem Risks





| | |
|---|---|
| **Smart Phone/Updates** | |
| | **Counterfeit Updates** |
| | **Trap Door in Update** |
| | **Discontinued Updates** |
| | **Slow Updates** |
| | **Unencrypted Updates** |
| | **Out Of Date Firmware** |
| | **Device Not Updated** |

# Appendix B – Tools Supporting IoT Listed by Component

# Requirement and Design Tools

- Requirements
  - Rational Requisite Pro
  - Objectiver
  - CaseComplete
  - RMTrac
  - Optimal Trace
  - Analyst Pro
  - DOORS
  - GMARC
  - Jira

- Design
  - IBM Architect Designer
  - Structure Chart
  - Data Flow Diagram
  - Doxygen
  - Visustin
  - McCabe IQ
  - Design By Contract
  - Assertions

# Editors and Compilers

- Editors
  - Sublime Text
  - Notepad ++
  - Vim
  - Atom
  - Emacs
  - Eclipse
  - Code::Blocks
  - GNAT Programming Studio
  - Code Lite
  - NetBeans
  - Qt Creator

- Compilers
  - Intel
  - Gnu
  - Microsoft
  - Visual C++
  - Clang
  - CUDA
  - ROSE Custom Tool

# Build Tools and Libraries

- Build Tools
  - Cmake
  - Autoconf
  - Jenkins
  - Bamboo
  - Ant
  - CruiseControl
  - Git
  - Subversion
  - Mecurial
  - Perforce
  - and many more

- Libraries
  - Boost
  - OpenMP
  - OpenGL
  - VTK
  - and many more

# Debuggers and Software Testing

- Debuggers
  - Code View
  - Valgrind
  - MS Visual Studio
  - Intel Debugger
  - Parasoft Insure++
  - IDAPro
  - ROSE Custom Tool
  - and many more

- Software Testing
  - Cppunit
  - Junit
  - Google Test
  - Test Complete
  - HP Functional Tester
  - Selenium
  - Squish
  - ATS
  - and many more

# Developed Code and IoT Device

- Developed Code
  - Purify
  - Intel Inspector XE
  - Insure++
  - Lcov,Gcov
  - Cobertura
  - Klocworks
  - Parasoft
  - Coverity
  - ROSE Custom Tool

- IoT Device
  - Vector Software
  - HiTex
  - eggPlant
  - Xilinx SDK
  - Peta Linux Tools
  - IDAPro
  - ROSE Custom Tool

# Electronic and Mechanical Design

- Electronic
  - Spark
  - PulseForge 3300
  - Project Wire
  - TDK4PE
  - PCB Web Designer
  - NanoTech AMD 3D
  - and many more

- Mechanical
  - AutoCad
  - Autodesk CAM
  - IronCAD
  - SolidWorks
  - and many more

# Device HMI and Peripherals

- HMI
  - Active X
  - Lab Windows
  - Tk/Tcl
  - Motif
  - Visual Basic/C++
  - PowerBuilder
  - and many more

- Peripherals
  - LabView
  - FastSend
  - PacketCheck
  - Intel Admin Tools
  - NDT

# Software Aps and Utilities

- Software Aps
  - Norton
  - McAfee
  - TrendMicro
  - Malwarebytes
  - Bitdefender
  - MS Office
  - ROSE Custom Tool

- Utilities
  - CrashPlan
  - COMODO Back Up
  - AOMEI Backupper
  - Cobian Backup
  - and many more

# Operating Systems, I/O Cable of Wi-Fi

- Operating Systems
  - Windows
  - Linux
    - Embedded Linux
    - Android
  - Unix BSD
  - Mac OS X
    - iOS
  - Commercial Unix
  - Solaris/OpenSolaris
  - Custom OS

- Cable or Wi-Fi
  - Vastar
  - Fluke
  - Ideal Networks
  - AirCheck G2
  - Wi-Fi Pineapple
  - Wi-Spy DBx
  - Wi-Fi Analyzer
  - Yellowjacket-BANG

# Routers and Firewalls

- Routers
  - Angry IP Scanner
  - Ping Plotter
  - Blast
  - TCP View
  - PRTG Network Monitor
  - HP Load Runner
  - and many more

- Firewalls
  - Nessus
  - Nmap
  - Netcat
  - Tcpdump
  - Wireshark
  - CommView
  - and many more

# Internet and Update Patches

- Internet
  - ZenMap
  - Ipplan
  - Tcpdump
  - Nmap NSE Scripts
  - Dig
  - Acunetix
  - John the Ripper
  - and many more

- Update Patches
  - Tripwire CCM
  - CodeDx
  - Corporate Software Inspector
  - Open VAS
  - Retina CS
  - MBSA
  - Nexpose
  - and many more

# Cellular Network Smart Phone O/S

- Cellular Network
  - Speedtest
  - Network Signal Info Pro
  - wiggle.net

- Smart Phone O/S
  - iOS
  - Android
  - Windows Mobile
  - BlackBerry OS
  - LG webOS

# Smart Phone Aps and Browser

- Smart Phone Aps
  - McAfee
  - Norton
  - Trend
  - Panda
  - Webroot
  - iPhone N/A

- Browsers
  - Chrome
  - Firefox
  - Safari
  - Opera
  - Skyfire
  - UC Browser
  - BeFE

# Appendix C – Visual Basic Code for STPA Automation

```
'***********************************************************************************************************
'Program to automate the analysis of all combination cases and determine hazardous conditions
'Input is T or F or blank or null in columns of a spreadsheet
'T means the condition or state is true, F means the condition or state is false
'Blank or null means the condition does not matter
'Output is a Y in columns that produce a hazard condition, N if it does not
'This program assumes that there are 64 columns and 5 binary conditions with four guide phrases
'Each guide phrase is a worksheet within the workbook
'Written in Visual Basic for Applications (VBA) 7.1
'January 16, 2017 version 1.0
'Gregory M. Pope
'Lawrence Livermore National Laboratory
'***********************************************************************************************************


'***********************************************************************************************************
'Caveat
'This VBA code was written to demonstrate one of many possible ways to automate STPA for a specific text book problem
'The code has been tested multiple ways, but only for this one example problem
'This code has not been verified or validated for general use for an actual hazard analysis of a real system
'However the code may be of some value to show one way STPA can be automated within Excel
'The author and LLNL make no explicit or implicit guarantee of the code's correctness
```

```
'*************************************************************************************************************

Dim Stopped, cStopped, Aligned, cAligned, Alarm, cAlarm, Obstructed, cObstructed, cClosed, Closed, Hazard, cHazard, acHazard(3) As String

Dim Col, ccol, K As Integer

Dim Logic_Test As Boolean


'Read from the active worksheet 32 columns of conditions

Sub Read_Column()

For Col = 2 To 33

Call Motionless

Call Platform

Call Emergency

Call Doorway

'For the active worksheet pick the equation that is a non-hazardous combination of states

'If this produces a true logical test then it is not considered a hazard, if false it is a hazardous condition

Call Read_And_Or_Chart

If ActiveSheet.Name = "Provided" Then Logic_Test = Stopped = "F" Or (Stopped = "T" And Aligned = "F" And Alarm = "F")

If ActiveSheet.Name = "Not Provided" Then Logic_Test = (Stopped = "T" And Aligned = "T") Or (Stopped = "T" And Aligned = "F" And Alarm = "T") Or (Stopped = "T" And Aligned = "T" And Obstructed = "T")

If ActiveSheet.Name = "Late" Then Logic_Test = (Stopped = "T" And Aligned = "T") Or (Stopped = "T" And Aligned = "T" And Obstructed = "T") Or (Stopped = "T" And Aligned = "F" And Alarm = "T")

If ActiveSheet.Name = "Stopped" Then Logic_Test = Stopped = "T" Or Stopped = "F"

Debug.Print "________________________"
Debug.Print "      " + CStr(Logic_Test) + "       "

If Logic_Test = True Then Hazard = "Y" Else Hazard = "N"

ActiveSheet.Cells(16, Col) = Hazard

Debug.Print "______________________" + CStr(Col - 1)

Next

End Sub

'Read in motion state

Sub Motionless()


ActiveSheet.Cells(4, Col).Select


Stopped = ActiveSheet.Cells(4, Col).Value


If Stopped <> "T" And Stopped <> "F" Then MsgBox "Value must be either T or F was = " + Stopped, vbOKOnly, "Error"


Debug.Print "Stopped = " + Stopped


End Sub

'Read in platform state

Sub Platform()


ActiveSheet.Cells(6, Col).Select


Aligned = ActiveSheet.Cells(6, Col).Value
```

```
If Aligned <> "T" And Aligned <> "F" Then MsgBox "Value must be either T or F was = " + Aligned,
vbOKOnly, "Error"

Debug.Print "Aligned = " + Aligned


End Sub

'Read in Emergency state
Sub Emergency()

ActiveSheet.Cells(8, Col).Select

Alarm = ActiveSheet.Cells(8, Col).Value

If Alarm <> "T" And Alarm <> "F" Then MsgBox "Value must be either T or F was = " + Alarm, vbOKOnly,
"Error"

Debug.Print "Alarm = " + Alarm

End Sub

'Read in Doorway state
Sub Doorway()

ActiveSheet.Cells(10, Col).Select
```

```
Obstructed = ActiveSheet.Cells(10, Col).Value

If Obstructed <> "T" And Obstructed <> "F" Then MsgBox "Value must be either T or F was = " + Obstructed, vbOKOnly, "Error"

Debug.Print "Obstructed = " + Obstructed

ActiveSheet.Cells(12, Col).Select

Closed = ActiveSheet.Cells(12, Col).Value

If Closed <> "T" And Closed <> "F" Then MsgBox "Value must be either T or F was = " + Closed, vbOKOnly, "Error"

Debug.Print "Closed = " + Closed

End Sub

' Read in and/or Chart
Sub Read_And_Or_Chart()
acHazard(1) = ""
acHazard(2) = ""
acHazard(3) = ""
cHazard = "Y"
```

```
K = 0
For ccol = 36 To 38

ActiveSheet.Cells(5, ccol).Select
cStopped = ActiveSheet.Cells(5, ccol).Value
If cStopped <> "T" And cStopped <> "F" And cStopped <> "" And cStopped <> " " Then MsgBox "Value must be either T or F or null or space was = " + cStopped, vbOKOnly, "Error"

ActiveSheet.Cells(6, ccol).Select
cAligned = ActiveSheet.Cells(6, ccol).Value
If cAligned <> "T" And cAligned <> "F" And cAligned <> "" And cAligned <> " " Then MsgBox "Value must be either T or F or null or space was = " + cAligned, vbOKOnly, "Error"

ActiveSheet.Cells(7, ccol).Select
cAlarm = ActiveSheet.Cells(7, ccol).Value
If cAlarm <> "T" And cAlarm <> "F" And cAlarm <> "" And cAlarm <> " " Then MsgBox "Value must be either T or F or null or space was = " + cAlarm, vbOKOnly, "Error"

ActiveSheet.Cells(8, ccol).Select
cObstructed = ActiveSheet.Cells(8, ccol).Value
If cObstructed <> "T" And cObstructed <> "F" And cObstructed <> "" And cObstructed <> " " Then MsgBox "Value must be either T or F or null or space was = " + cObstructed, vbOKOnly, "Error"

ActiveSheet.Cells(9, ccol).Select
cClosed = ActiveSheet.Cells(9, ccol).Value
If cClosed <> "T" And cClosed <> "F" And cClosed <> "" And cClosed <> " " Then MsgBox "Value must be either T or F or null or space was = " + cClosed, vbOKOnly, "Error"
```

```
Debug.Print "cStopped = " + cStopped
Debug.Print "cAligned = " + cAligned
Debug.Print "cAlarm = " + cAlarm
Debug.Print "cObstructed = " + cObstructed
Debug.Print "cClosed = " + cClosed

K = K + 1


Call Compare_To_Chart

acHazard(K) = cHazard
Next
If acHazard(1) = "N" Or acHazard(2) = "N" Or acHazard(3) = "N" Then cHazard = "Y" Else cHazard = "N"
Debug.Print "acHazard(1) = " + acHazard(1) + "  acHazard(2) = " + acHazard(2) + "  acHazard(3) = " + acHazard(3)
ActiveSheet.Cells(15, Col) = cHazard
Debug.Print "cHazard = " + cHazard
End Sub

'Compare and/or chart to current contect table column
Sub Compare_To_Chart()


'Check if Stopped matches
```

```
If cStopped <> "" And cStopped <> " " Then
    Debug.Print "Not Null or Blank"
    If Stopped = cStopped Then
        cHazard = "N"
         Debug.Print "N"
    End If
    If Stopped <> cStopped Then
        cHazard = "Y"
        Debug.Print "Y"
        Exit Sub
    End If
End If
Debug.Print "cStopped = " + cStopped + " Stopped = " + Stopped

' Check if Aligned matches
If cAligned <> "" And cAligned <> " " Then
    Debug.Print "Not Null or Blank"
    If Aligned = cAligned Then
        cHazard = "N"
         Debug.Print "N"
    End If
    If Aligned <> cAligned Then
        cHazard = "Y"
        Debug.Print "Y"
        Exit Sub
```

```
   End If
End If
Debug.Print "cAligned = " + cAligned + " Aligned = " + Aligned


'Check if Alarm matches
If cAlarm <> "" And cAlarm <> " " Then
   Debug.Print "Not Null or Blank"
   If Alarm = cAlarm Then
      cHazard = "N"
       Debug.Print "N"
   End If
   If Alarm <> cAlarm Then
      cHazard = "Y"
      Debug.Print "Y"
      Exit Sub
   End If
End If
Debug.Print "cAlarm = " + cAlarm + " Alarm = " + Alarm


'Check if Obstructed matches
If cObstructed <> "" And cObstructed <> " " Then
   Debug.Print "Not Null or Blank"
   If Obstructed = cObstructed Then
      cHazard = "N"
       Debug.Print "N"
```

```
    End If
    If Obstrcuted <> cObstructed Then
        cHazard = "Y"
        Debug.Print "Y"
        Exit Sub
    End If
End If
Debug.Print "cObstructed = " + cObstructed + " Obstructed = " + Obstructed

'Check in Clossed matches
If cClosed <> "" And cClosed <> " " Then
    Debug.Print "Not Null or Blank"
    If Closed = cClosed Then
        cHazard = "N"
         Debug.Print "N"
    End If
    If Closed <> cClosed Then
        cHazard = "Y"
        Debug.Print "Y"
        Exit Sub
    End If
End If
Debug.Print "cClosed = " + cClosed + " Closed = " + Closed

End Sub
```

## End Notes

[1] Brown, Eric (13 September 2016). "Who Needs the Internet of Things?". Linux.com. Retrieved 23 October 2016.
[2] Eric (20 September 2016). "21 Open Source Projects for IoT". Linux.com. Retrieved 23 October 2016.
[3] "Internet of Things Global Standards Initiative". ITU. Retrieved 26 June 2015
[4] Source IHS
[5] The Internet Of Things Heat Map, 2016 Where IoT Will Have The Biggest Impact On Digital Business by Michele Pelino and Frank E. Gillett January 14, 2016
[6] http://sunnyday.mit.edu/STPA-Primer-v0.pdf
[7] https://www.scmagazineuk.com/cyber-security-of-the-fridge-assessing-the-internet-of-things-threat/article/531554/
[8] https://chapters.theiia.org/pittsburgh/Events/Documents/Root%20Cause%20Analysis%20Presentation.ppt
[9] Failure Modes and Effect Analysis (FMEA) was developed by reliability engineers in the late 1950s to study problems that might arise from malfunctions of military systems.
[10] Bill Vesely, NASA HQ
[11] ://wwhttpw.informationisbeautiful.net/visualizations/million-lines-of-code
[12] http://www.onlineclassnotes.com/2013/01/software-doesnt-wear-out-explain-this.html
[13] http://www.onlineclassnotes.com/2013/01/software-doesnt-wear-out-explain-this.html
[14] See http://www.silverbuckshot.net/Accidents Three Mile Island, Vincennes (Iran Air Flight 655), Cali (American Airlines Flight 965), Ariane 501, Mars Climate Orbiter (MCO), Mars Polar Lander (MPL), Titan/Centaur/Milstar, SOHO (Solar Heliospheric Observatory), Therac 25, ATT Phone System, Kegworth British Midlands, Asiana Flight 214
[15] Information for this section drawn from PowerPoint slides used during a Safety System Software Class presented in Seattle during the summer of 2010 by Nancy Leveson. Permission to use this information granted by Professor Leveson. Also some of the information is covered in Nancy Leveson's Book *Engineering a Safer World*, MIT Press, Fall 2011, page 164.
[16] https://www.linkedin.com/pulse/top-8-mistakes-requirements-elicitation-aaron-whittenberger-cbap
[17] https:// https://nest.com/works-with-nest/ https://nest.com/works-with-nest/nest.com/works-with-nest/
[18] http://www.apple.com/ios/home/
[19] https://en.wikipedia.org/wiki/Shodan_(website)
[20] https://en.wikipedia.org/wiki/Aristocrat_Leisure
[21] http://www.washingtonpost.com/wp-dyn/content/article/2009/07/01/AR2009070100601.html
[22] https://www.wired.com/2017/02/russians-engineer-brilliant-slot-machine-cheat-casinos-no-fix/

[23] http://www.express.co.uk/news/world/600771/ISIS-Islamic-State-remote-controlled-cars-bomb-attacks-Syria-Iraq-drones-Britain
[24] https://krebsonsecurity.com/2016/10/hacked-cameras-dvrs-powered-todays-massive-internet-outage/
[25] http://www.welivesecurity.com/2017/01/30/austrian-hotel-experiences-ransomware-things-attack/
[26] http://www.zdnet.com/article/shodan-the-iot-search-engine-which-shows-us-sleeping-kids-and-how-we-throw-away-our-privacy/
[27] https://www.panamacademy.com/are-pilots-new-threat-aviation-safety-loss-flying-skills-must-be-addressed
[28]http://www.slate.com/articles/technology/future_tense/2014/12/automation_in_the_cockpit_is_making_pilots_thinking_skills_duller.html
[29] http://www.gps.gov/spectrum/jamming/
[30] Donald Reifer, "Industry Software Cost, Quality, and Productivity Benchmarks", DoD Software Tech News, July 2004
[31] http://www.howtogeek.com/195430/how-to-create-a-strong-password-and-remember-it/