Systemic Theoretic Process Analysis (STPA) Used for Cyber Security

EFCOG

April 21, 2021

- Gregory Pope, CSQE
- Group Leader SQA

Lawrence Livermore National Laboratory

LLNL-PRES-821532-DRAFT

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

To Answer These:

Can STPA be used to identify Cyber Security Requirements?

Can Secure Software Be Developed with Agile?

Explicit versus Implicit Software Requirements

Explicit Requirements

1. User adds records
2. User deletes records
3. User modifies records
4. User merges multiple records
5. Bla, bla, bla

Implicit Requirements

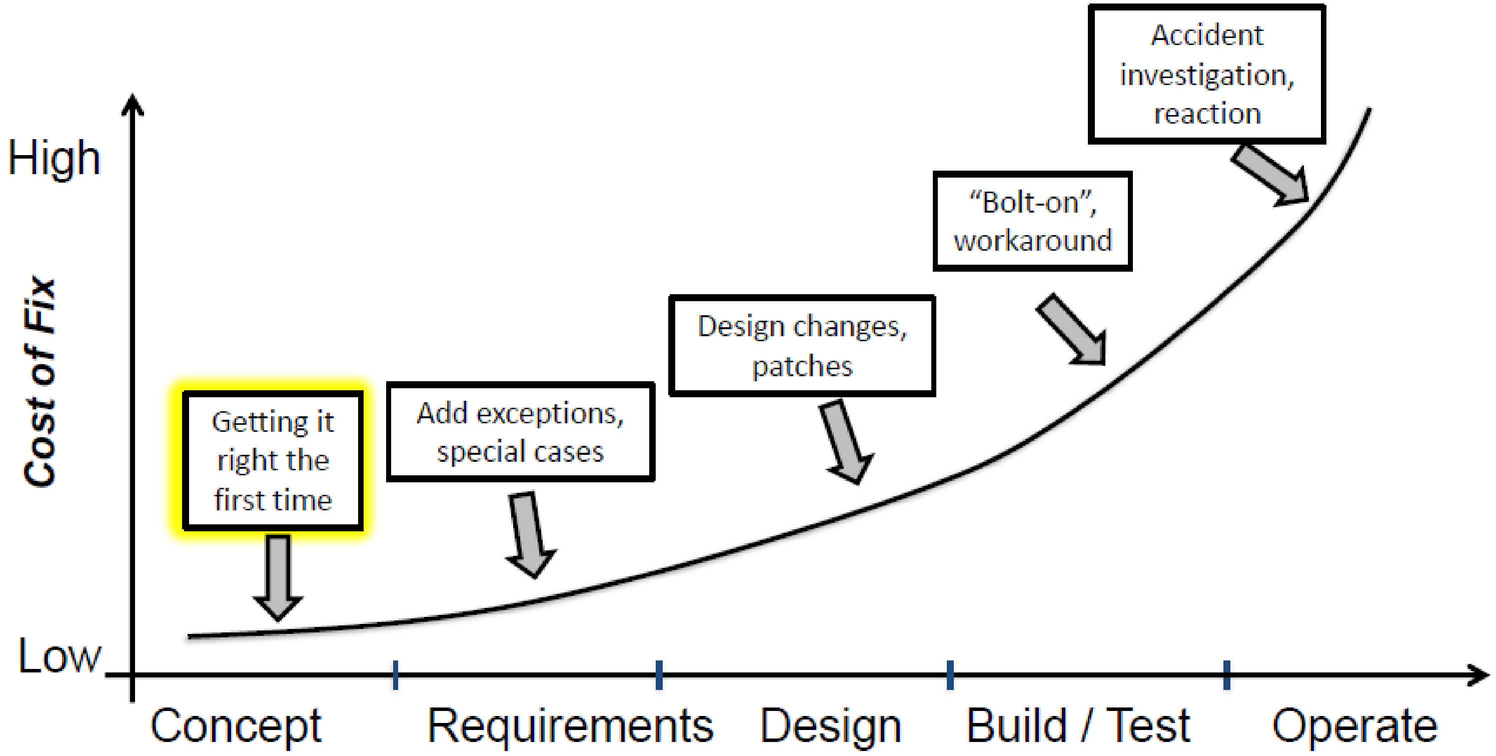
- Make it easy to use
- Make it scalable
- Make it secure

Security

- Customer may not be able to explicitly state what they want in terms of security requirements, but ……

They know they want their software to be secure.

Cost to fix problem vs. when found



When are these flaws being discovered?
vs.
When are they created?

Adapted from Young, 2014

Cyber-Attacks are a Big Deal

- 94% of malware was delivered through email
- 34% of data breaches that occurred were due to insiders
- 17% of data breaches involved malware
- Over 80% of security breaches were a result of phishing attacks
- 60% of security breaches occurred due to unpatched vulnerabilities
- Attacks on IoT devices grew threefold in early 2019

Frequency and Cost of Cyber-Attacks

- Globally 30,000 a day
- Trillions of dollars

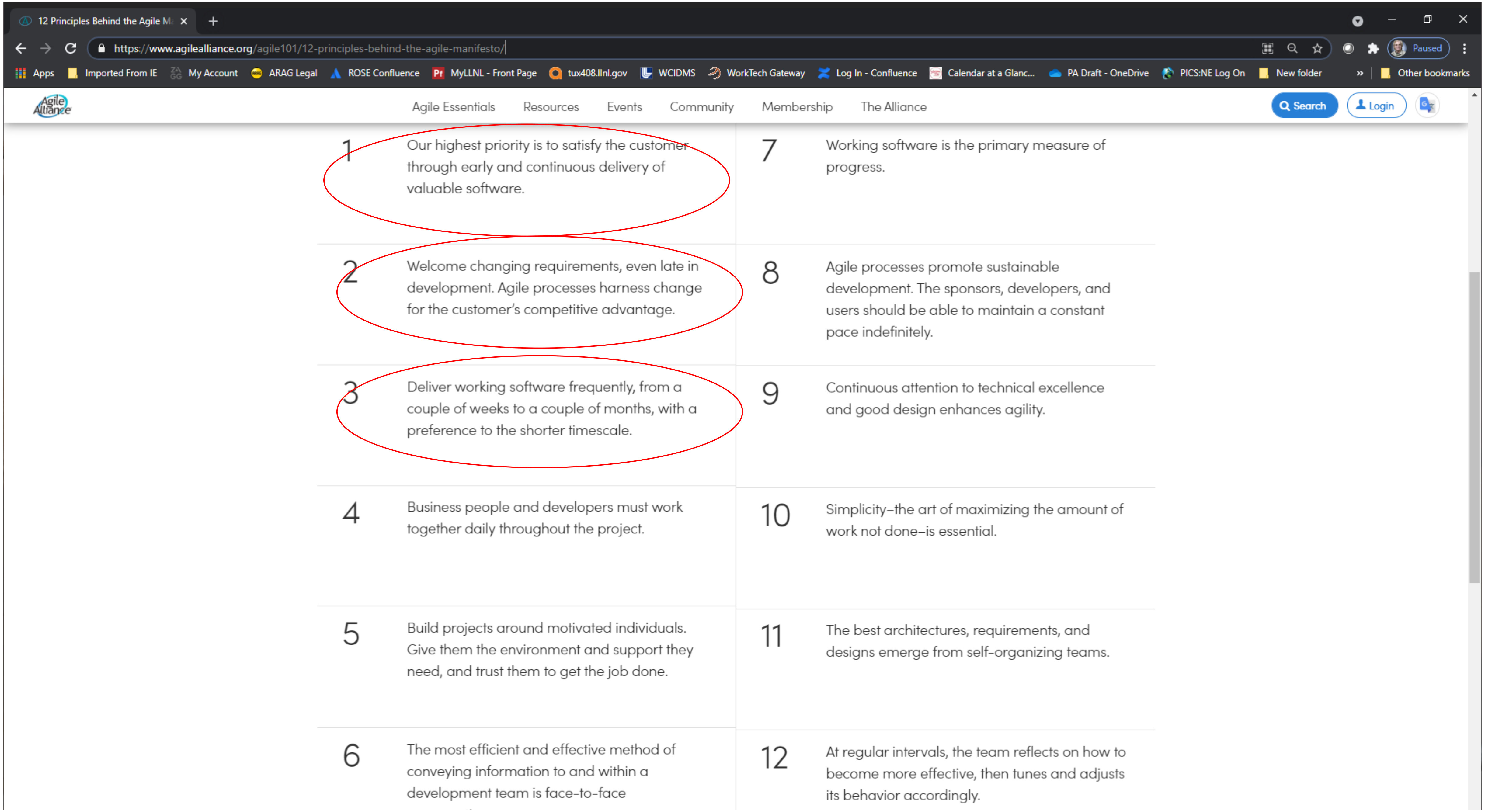
Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas



1 Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2 Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3 Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4 Business people and developers must work together daily throughout the project.

5 Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6 The most efficient and effective method of conveying information to and within a development team is face-to-face

7 Working software is the primary measure of progress.

8 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9 Continuous attention to technical excellence and good design enhances agility.

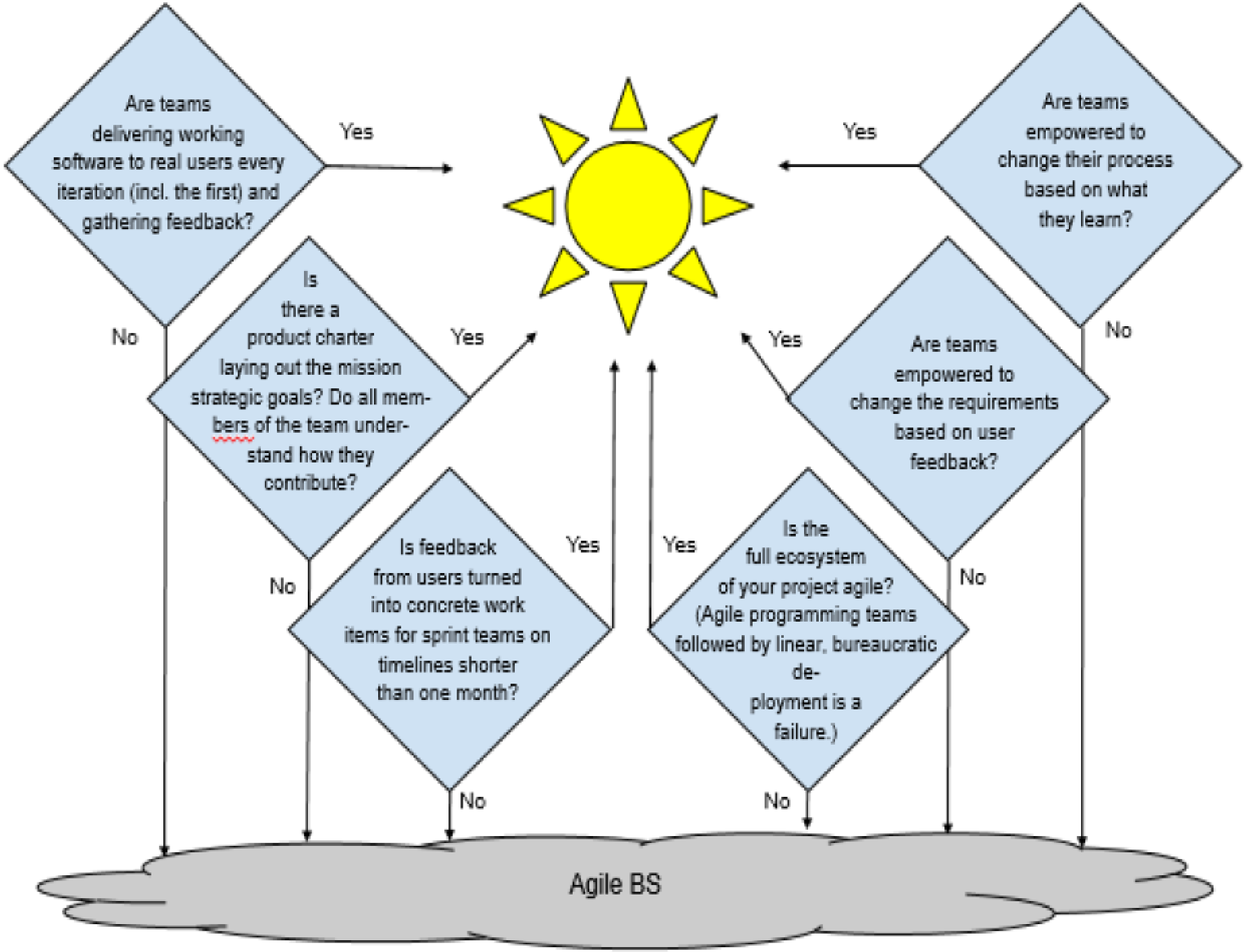
10 Simplicity–the art of maximizing the amount of work not done–is essential.

11 The best architectures, requirements, and designs emerge from self-organizing teams.

12 At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile BS Detector

Graphical version:



Are teams delivering working software to real users every iteration (incl. the first) and gathering feedback?

Yes

No

Is there a product charter laying out the mission strategic goals? Do all members of the team understand how they contribute?

Yes

No

Is feedback from users turned into concrete work items for sprint teams on timelines shorter than one month?

Yes

No

Are teams empowered to change their process based on what they learn?

Yes

No

Are teams empowered to change the requirements based on user feedback?

Yes

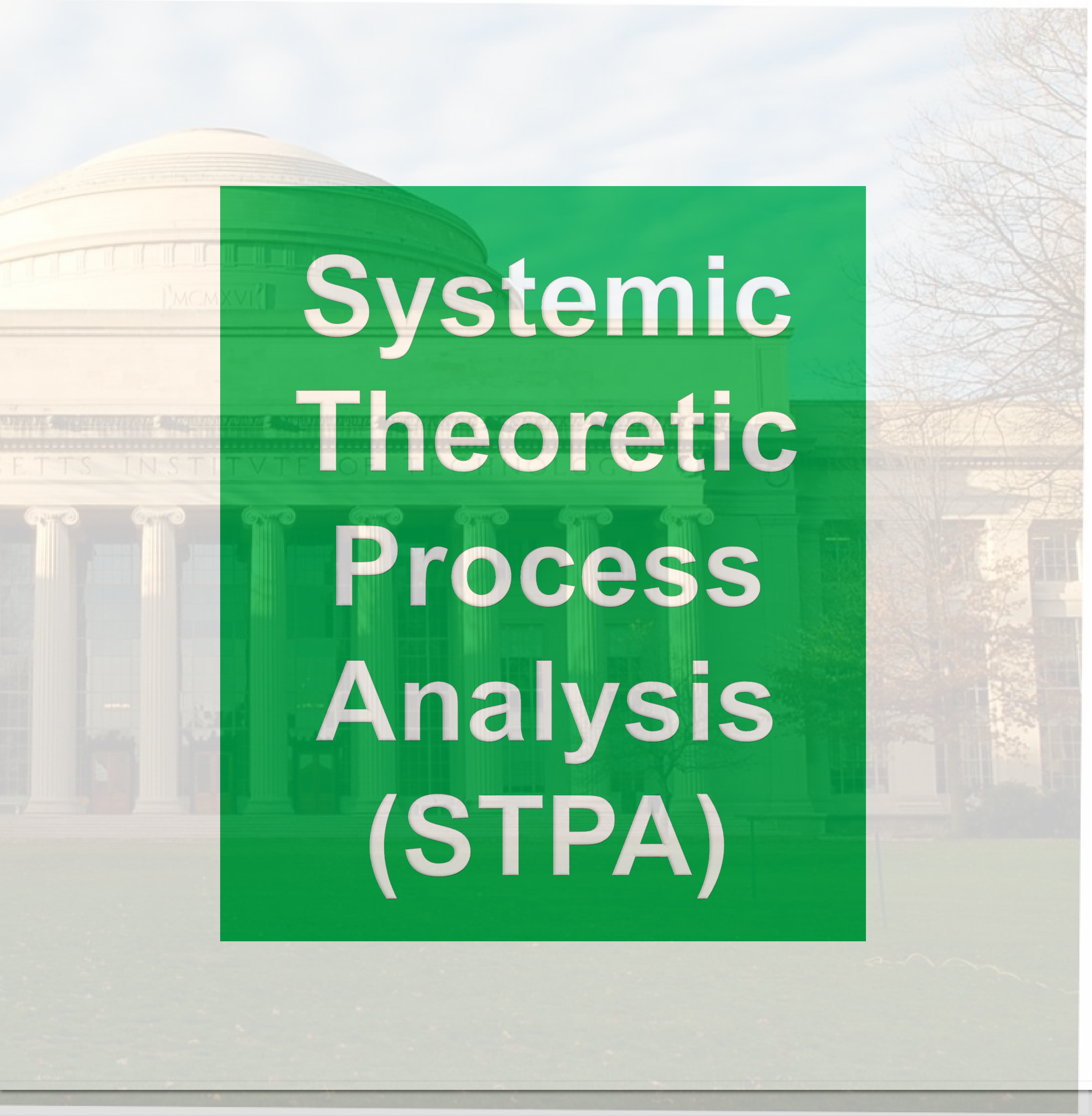
No

Is the full ecosystem of your project agile? (Agile programming teams followed by linear, bureaucratic deployment is a failure.)

Yes

No

Agile BS



Systemic Theoretic Process Analysis (STPA)

- Originally developed for hazard analysis of software-controlled systems
- Nancy Leveson PhD, Professor MIT
- John Thomas PhD MIT
- An alternative to FTA, FMEA, RCA
- Call cyber-attacks hazards and use STPA
- Make implicit security requirements explicit
- Used at the beginning of the software development lifecycle

Identify Hazards and Losses

Hazards

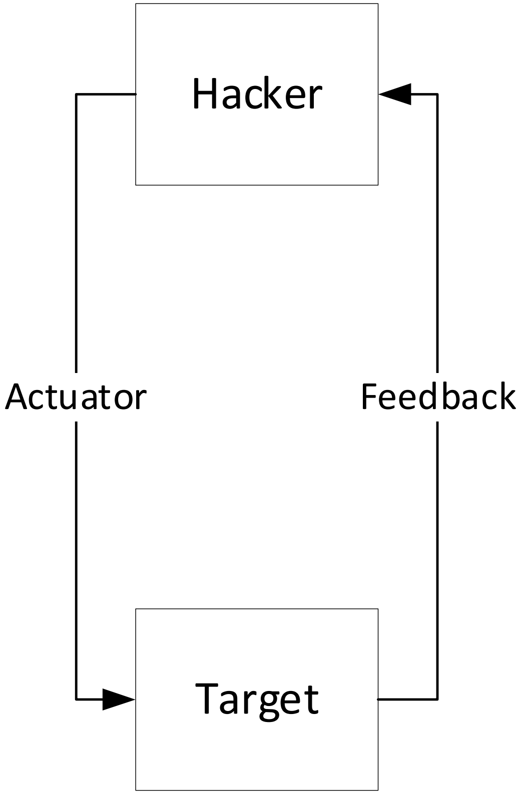
- H1. Malware
- H2. Phishing
- H3. Man-in-the Middle
- H4. Denial of Service
- H5. SQL Injection
- H6. Buffer Overflow
- H7. Zero Day Exploit
- H8. DNS Tunneling

Losses

- L1. Loss of Life or Injury
- L2. Financial Losses
- L3. Loss of Sensitive Information
- L4. Loss of Trade Secrets
- L5. Loss Public Trust
- L6. Loss of Business Operation

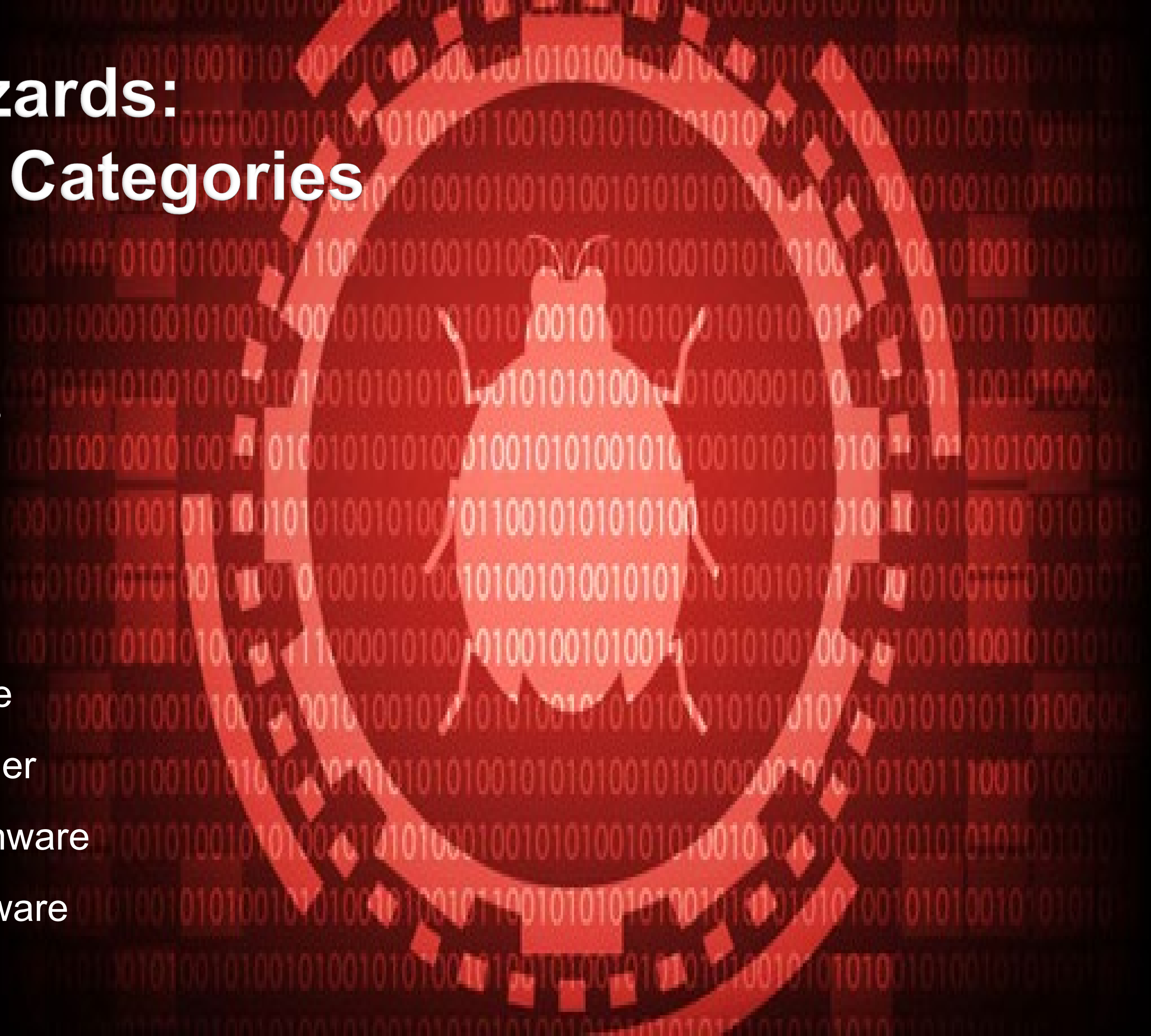
Model of Control Structure: Cyber Security as a System

Hazards
H1. Malware
H2. Phishing
H3. Man-in-the Middle
H4. Denial of Service
H5. SQL Injection
H6. Buffer Overflow
H7. Zero Day Exploit
H8. DNS Tunneling



L1. Loss of Life or Injury
L2. Financial Losses
L3. Loss of Sensitive Information
L4. Loss of Trade Secrets
L5. Loss Public Trust
L6. Loss of Business Operation

Sub- Hazards: Malware Categories



- H1.1 Adware
- H2.2 Bots
- H1.3 Rootkits
- H1.4 Viruses
- H1.5 Worms
- H1.6 Trojan
- H1.7 Spyware
- H1.8 Keylogger
- H1.9 Ransomware
- H1.10 Scareware

Identify Unsafe Control Actions: 17 Cyber-Attack Types Used

Attack Types
H1. Malware
H1.1 Adware
H1.2 Bots
H1.3 Rootkits
H1.4 Viruses
H1.5 Worms
H1.6 Trojans
H1.7 Spyware
H1.8 Keylogger
H1.9 Ransomware
H1.10 Scareware
H2 Phishing
H3 Man in the Middle
H4 Denial of Service
H5 SQL Injection
H6. Buffer Overflow
H7 Zero Day Exploit
H8 DNS Tunneling

Example Analysis: Adware

H1.1 Adware

M1.1 Run Software to Detect and Remove Known Adware and Potentially Unwanted Programs (PUP)

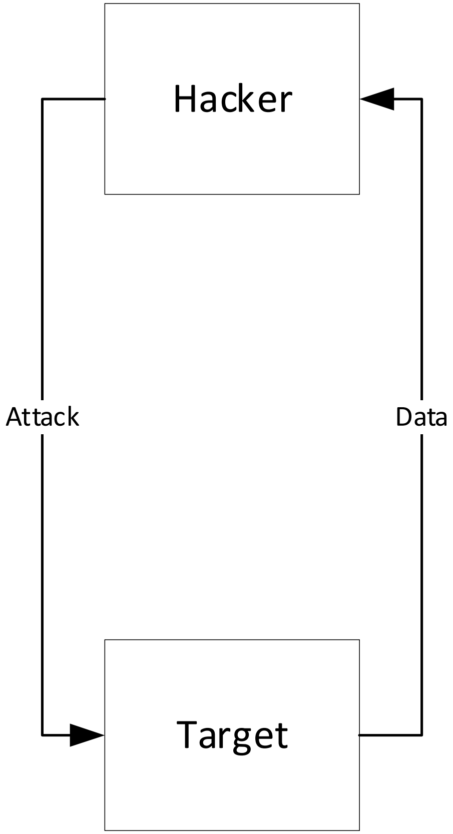
M1.4.1 Assure Virus detection software is running and is up to date

M1.4.2 Detect higher than expected CPU and RAM Memory usage

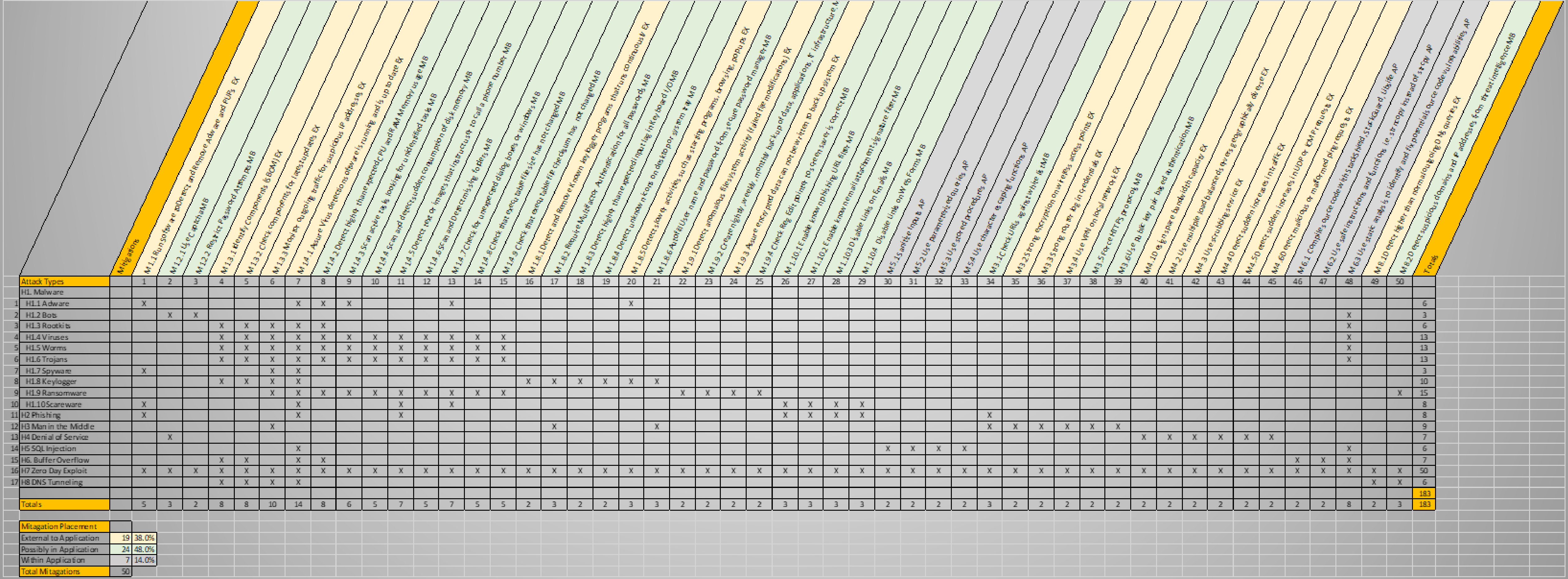
M1.4.3 Scan active tasks looking for unidentified tasks

M1.4.7 Check for unexpected dialog boxes or windows

M1.8.5 Detect slower activities such as starting programs, browsing, pop ups.



STPA 17 Cyber-Attacks and 50 Mitigations



		Mitigations	M1.1 Run software to Detect and Remove Adware and PUPs EX	M1.2.1 Use CaptchaMB	M1.2.2 Restrict Password Attempts MB	M1.3.1 Identify Components (SBOM) EX	M1.3.2 Check components for latest updates EX	M1.3.3 Monitor Outgoing traffic for suspicious IP addresses EX	M1.4.1 Assure Virus detections software is running and is up to date EX	M1.4.2 Detect higher than expected CPU and RAM Memory usage MB	M1.4.3 Scan active tasks looking for unidentified tasks MB	M1.4.4 Scan and detect sudden consumption of disk memory MB	M1.4.5 Detect text or images that instruct user to call a phone number MB	M1.4.6 Scan and Detect missing folders MB	M1.4.7 Check for unexpected dialog boxes or windows MB	M1.4.8 Check that executable file size has not changed MB	M1.4.9 Check that executable file Checksum has not changed MB	M1.8.1 Detect and Remove Known keylogger programs that runs continuously EX	M1.8.2 Require Multifactor Authentication for all passwords MB	M1.8.3 Detect higher than expected input lag in keyboard I/O MB	M1.8.4 Detect unknown icons on desktop or system tray MB	M1.8.5 Detect slow activities such as starting programs, browsing, popups EX	M1.8.6 Autofill User name and password from secure password manager MB	M1.9.1 Detect anomalous file system activity (failed file modifications) EX	M1.9.2 Create nightly, weekly, monthly backup of data, applications, & infrastructure M	M1.9.3 Assure encrypted data can not be written to back up system EX	M1.9.4 Check Reg Edit pointers to screen saver is correct MB	M1.10.1 Enable known phishing URL filter MB	M1.10.2 Enable known email attachment signature filter MB	M1.10.3 Disable Links on Emails MB	M1.10.4 Disable Links on Web Forms MB	M5.1 Sanitize Inputs AP	M5.2 Use parameterized queries AP	M5.3 Use stored procedures AP	M5.4 Use character escaping functions AP	M3.1 Check URLs against white list MB	M3.2 Strong encryption on wireless access points EX	M3.3 Strong router log in credentials EX	M3.4 Use VPN on local network EX	M3.5 Force HTTPS protocol MB	M3.6 Use Public key pair based authentication MB	M4.1 Design spare bandwidth capacity EX	M4.2 Use multiple load balanced servers geographically diverse EX	M4.3 Use scrubbing service EX	M4.4 Detect sudden increases in traffic EX	M4.5 Detect sudden increases in UDP or ICMP requests EX	M4.6 Detect malicious or malformed ping requests EX	M6.1 Compile source code with StackShield, StackGuard, Libsafe AP	M6.2 Use safe instructions and functions i.e. strncpy instead of strcpy AP	M6.3 Use static analysis to identify and fix potential source code vulnerabilities AP	M8.1 Detect higher than normal outgoing DNS queries EX	M8.2 Detect suspicious domains and IP addresses from threat intelligence MB	Totals
	Attack Types		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	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	
	H1. Malware																																																				
1	H1.1 Adware		X						X	X	X				X							X																															6
2	H1.2 Bots			X	X																																													X			3
3	H1.3 Rootkits					X	X	X	X	X																																								X			6
4	H1.4 Viruses					X	X	X	X	X	X	X	X	X	X	X	X																																	X			13
5	H1.5 Worms					X	X	X	X	X	X	X	X	X	X	X	X																																	X			13
6	H1.6 Trojans					X	X	X	X	X	X	X	X	X	X	X	X																																	X			13
7	H1.7 Spyware		X					X	X																																												3
8	H1.8 Keylogger					X	X	X	X										X	X	X	X	X	X																													10
9	H1.9 Ransomware							X	X	X	X	X	X	X	X	X	X							X	X	X	X																									X	15
10	H1.10 Scareware		X						X				X		X													X	X	X	X																						8
11	H2 Phishing		X						X				X															X	X	X	X					X																	8
12	H3 Man in the Middle							X											X				X													X	X	X	X	X	X												9
13	H4 Denial of Service			X																																						X	X	X	X	X	X						7
14	H5 SQL Injection								X																							X	X	X	X															X			6
15	H6. Buffer Overflow					X	X		X	X																																						X	X	X			7
16	H7 Zero Day Exploit		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	50
17	H8 DNS Tunneling					X	X	X	X																																										X	X	6
																																																					183
	Totals		5	3	2	8	8	10	14	8	6	5	7	5	7	5	5	2	3	2	2	3	3	2	2	2	2	3	3	3	3	2	2	2	2	3	2	2	2	2	2	2	2	2	2	2	2	2	2	8	2	3	183

Mitagation Placement		
External to Application	19	38.0%
Possibly in Application	24	48.0%
Within Application	7	14.0%
Total Mitagations	50	

Example of Combined Attack and Mitigations

	Mitigations	M1.1 Run Software toDetect and Remove Adware and PUPs EX	M1.2.1 Use Captcha MB	M1.2.2 Restrict Password Attempts MB	M1.3.1 Identify Components (SBOM) EX	M1.3.2 Check components for latest updates EX	M1.3.3 Monitor outgoing traffic for suspicious IP addresses EX	M1.4.1 Assure Virus detection software is running and is up to date EX	M1.4.2 Detect higher than expected CPU and RAM Memory usage MB	M1.4.3 Scan active tasks looking for unidentified tasks MB	M1.4.4 Scan and detect sudden consumption of disk memory MB	M1.4.5 Detect text or images that instruct user to call a phone number MB	M1.4.6 Scan and Detect missing folders MB	M1.4.7 Check for unexpected dialog boxes or windows MB	M1.4.8 Check that executable file size has not changed MB	M1.4.9 Check that executable file checksum has not changed MB	M1.8.1 Detect and Remove Known keylogger programs that runs continuously EX	M1.8.2 Require Multifactor Authentication for all passwords MB	M1.8.3 Detect higher than expected input lag in Keyboard I/O MB	M1.8.4 Detect unknown icons on desktop or system tray MB	M1.8.5 Detect slower activities such as starting programs, browsing, pop ups EX	M1.8.6 Autofill User name and password from secure password manager MB	M1.9.1 Detect anomalous file system activity (failed file modifications) EX	M1.9.2 Create nightly, weekly, monthly back up of data, applications, IT infrastructure. M	M1.9.3 Assure encrypted data can not be written to back up system EX	M1.9.4 Check Reg Edit pointer to screen saver is correct MB	M1.10.1 Enable known phishing URL filter MB	M1.10.2 Enable known email attachment signature filter MB	M1.10.3 Disable Links on Emails MB	M1.10.4 Disable Links on Web Forms MB	M5.1 Sanitize Inputs AP	M5.2 Use parameterized queries AP	M5.3 Use stored procedures AP	M5.4 Use character escaping functions AP	M3.1 Check URLs against white list MB	M3.2 Strong encryption on wireless access points EX	M3.3 Strong router log in credentials EX	M3.4 Use VPN on local network EX	M3.5 Force HTTPS protocols MB	M3.6 Use public key pair based authentication MB	M4.1 Design spare bandwidth capacity EX	M4.2 Use multiple load balanced servers geographically diverse EX	M4.3 Use scrubbing service EX	M4.4 Detect sudden increases in traffic EX	M4.5 Detect sudden increases in UDP or ICMP requests EX	M4.6 Detect malicious or malformed ping requests EX	M6.1 Compile source code with StackShield, StackGuard, Libsife AP	M6.2 Use safe instructions and functions i.e. strncopy instead of strcpy AP	M6.3 Use static analysis to identify and fix potential source code vulnerabilities AP	M8.1 Detect higher than normal outgoing DNS queries EX	M8.2 Detect suspicious domains and IP addresses from threat intelligence MB	Totals
Attack Types		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	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	
H1. Malware																																																				
H1.1 Adware		X						X	X	X				X							X																															6
H1.2 Bots			X	X																																													X			3
H1.3 Rootkits					X	X	X	X	X																																								X			6
H1.4 Viruses					X	X	X	X	X	X	X	X	X	X	X	X																																	X			13
H1.5 Worms					X	X	X	X	X	X	X	X	X	X	X	X																																	X			13
H1.6 Trojans					X	X	X	X	X	X	X	X	X	X	X	X																																	X			13
H1.7 Spyware		X					X	X																																												3
H1.8 Keylogger					X	X	X	X									X	X	X	X	X	X																														10
H1.9 Ransomware							X	X	X	X	X	X	X	X	X	X							X	X	X	X																									X	15
H1.10 Scareware		X						X				X		X													X	X	X	X																						8
H2 Phishing		X						X				X															X	X	X	X					X																	8
H3 Man in the Middle							X											X				X													X	X	X	X	X	X												9
H4 Denial of Service			X																																						X	X	X	X	X	X						7
H5 SQL Injection								X																							X	X	X	X															X			6
H6. Buffer Overflow					X	X		X	X																																						X	X	X			7
H7 Zero Day Exploit		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	50
H8 DNS Tunneling					X	X	X	X																																										X	X	6
																																																				183
Totals		5	3	2	8	8	10	14	8	6	5	7	5	7	5	5	2	3	2	2	3	3	2	2	2	2	3	3	3	3	2	2	2	2	3	2	2	2	2	2	2	2	2	2	2	2	2	2	8	2	3	183

Mitagation Placement		
External to Application	19	38.0%
Possibly in Application	24	48.0%
Within Application	7	14.0%
Total Mitigations	50	

Opportunities for Improvement

- Mitigate Phishing (80%)
- Mitigate Lack of Updating and Patches (60%)
- Address Explicate Security Requirements Early
- Not all mitigations can happen in the application
- Stakeholders must include Network and IT Subject Matter Experts
- Nothing about Agile prohibits these mitigations

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.

Lawrence Livermore
National Laboratory