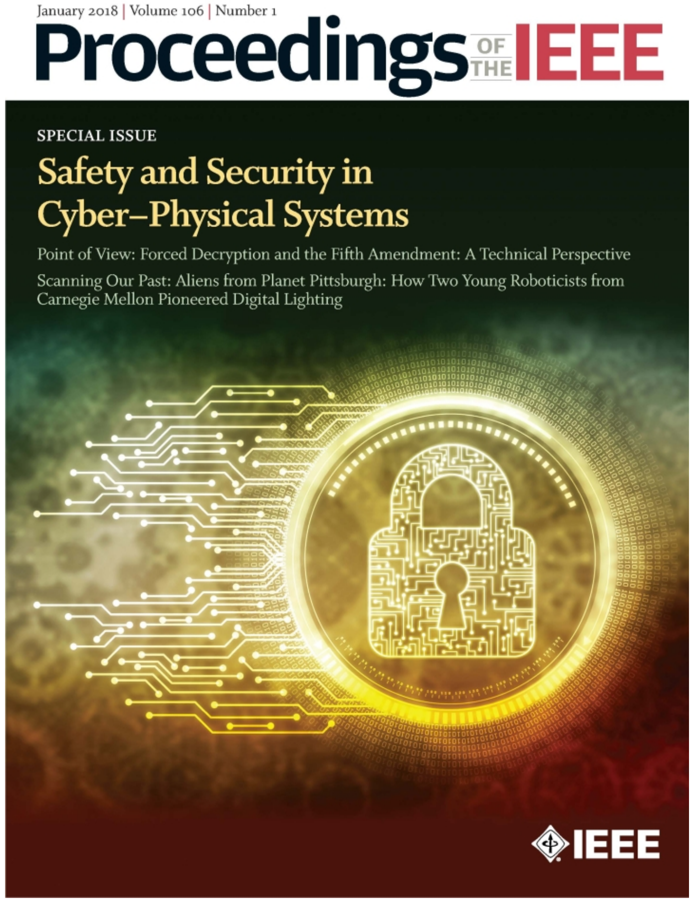
Safe and Secure Cyber-Physical and IoT Systems

Marilyn Wolf and Dimitrios Serpanos

Safe and Secure Cyber-Physical Systems

- Security: integrity and confidentiality of information.
- Safety: release of energy.
- Safety and security are traditionally handled by very distinct groups of people.
- These two characteristics are intertwined in cyber-physical systems.



January 2018 | Volume 106 | Number 1

Proceedings OF THE IEEE

SPECIAL ISSUE

Safety and Security in Cyber–Physical Systems

Point of View: Forced Decryption and the Fifth Amendment: A Technical Perspective

Scanning Our Past: Aliens from Planet Pittsburgh: How Two Young Roboticists from Carnegie Mellon Pioneered Digital Lighting

IEEE

Safety and security

- We can no longer treat safety and security as separate disciplines.
- Combination of real-time embedded systems and physical plants intertwines safety and security.
  - Cyber-physical systems (CPS).
  - Internet-of-Things systems (IoT).

Safety and security interactions

- Safety practices run counter to security updates.
  - Physical systems cannot be arbitrarily stopped.
  - Stopping and restarting may impose significant costs.

- Security practices run counter to safety practices.
  - Security provides a changing threat surface.
  - Design requirements change as threats evolve.

Themes

- Safety and security are inseparable in CPS and IoT systems.
- Neither safety nor security disciplines offer all the answers.
- Safety and security vary in their use of short-term *vs*. long-term approaches and in the use of prevention *vs*. remediation. The new field of safe and secure systems should operate at all time scales and from the earliest stages of design to updates.
- System designers must accept the fact that there is no end to design process due to evolving Internet threats. Systems must be designed to be adaptable to counter evolving threats.
- Suites of standardized design templates help to reduce design risks.
- Modern systems must combine design time analysis and architected safety+security features along with run-time monitoring.
- Safety and security should be assessed in part by probabilistic assertions of the health of the system.

Example threats

- An Airbus A400M crashed after takeoff at Sevilla Airport in Spain. Later analysis determined that the aircraft's engine control software had been incorrectly installed during its final assembly. That improper installation led to engine failure [Keo15].
- Analysis of the design of Toyota automobiles [Koo14] identified failures to apply well-known engineering techniques in several areas, including protection from cosmic ray-induced data errors and application of software engineering principles.
- Dieselgate [Dav15] was the result of a decision by Volkswagen management to design software in many of their diesel vehicles to provide inaccurate testing data that incorrectly gave the appearance of satisfying emissions regulations in several companies.

Dependability

- Avizienis and Laprie: **dependability** of computer system is justified reliance on its ability to provide its intended service.
- A **fault** may cause a system failure expressed as an **error**.
- Faults may be **physical** or **human-made**.
- Faults may be **permanent** or **transient**.

Safety

- **Safety** used for physical safety in this book.
- Related to absence or minimization of hazards that may harm life or property.
- Leveson argues that reliability is a property of components, safety is an emergent property.

Vulnerabilities

- Vulnerability in safety: design flaw or improper use of a system.
- May result in a safety hazard, security threat, or both.

Fault models

- Digital fault models are widely used for digital hardware.
    - Stuck-at 0/1 model models fault as output always stuck at a given value.
    - Stuck-at open/short models provide different behavior.
    - Transient models include bit flips, glitches.

System failure analysis

- Failures often occur because of a cascade of conditions and events.
- Common methods:
  - Functional Hazard Assessment (FHA).
  - Fault Tree Analysis (FTA).
  - Failure Mode Effects Analysis (FMEA).

Functional Hazard Analysis worksheet

Hazard ID	Life cycle phase	Activity	State/Mode	Function
Identifier	Phase analyzed by risk assessment	Actions performed within life cycle phase	System state or mode for the hazard	System function

Functional failure	Hazard Description	System Item(s)	Causal Factor Description	Mishap
Detailed description of failure mode	Detailed description of failure conditions	Portion of the system	Causes of failure	Description of failure

Effect(s)	Existing Mitigations	Software Control Category	Initial MRI	Software Criticality Index
Effects on life, limb, property	Existing means to mitigate failure	Degree of autonomy of software function	Initial risk assessment	Criticality

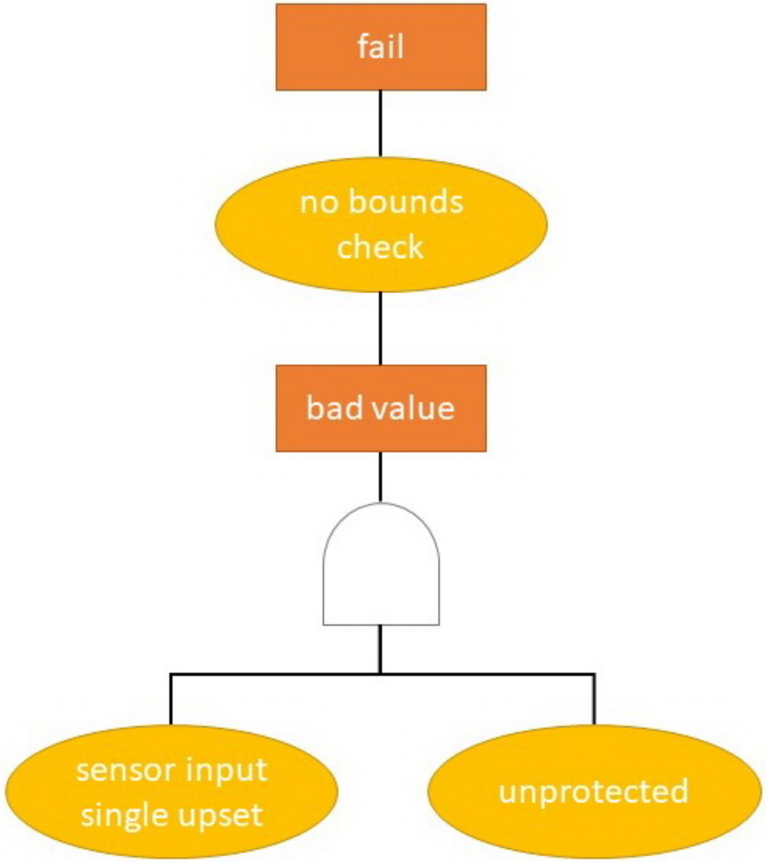
Target MRI	Causal Factor Risk Level	Recommended Mitigations	Comments	Follow-On Actions
Projected risk after mitigation	Potential for causal factors to occur	Methods to reduce risk	Relevant additional information	Further work to better understand risk

FHA methodology

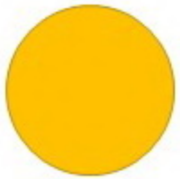
- System architecture data is analyzed to create a functional hierarchy, block diagrams, and a function/item matrix.
- The impact of the failure of each system function is analyzed for hazards.
- Safety-significant subsystems and interfaces are identified.
- Existing and recommended mitigations are identified.
- Safety-significant functions are decomposed to components. Component failures are related to subsystem hazards.
- Risk levels and software criticality indexes are identified and assigned. Follow-on actions are specified.
- A final FHA report is prepared.

Fault Tree Analysis

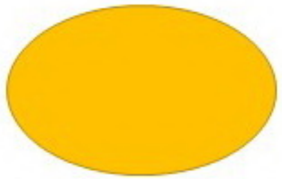
- Originally developed at Bell Labs.
- Typically proceeds from undesired event backward.
  - Precondition events can be combined to create successor event.



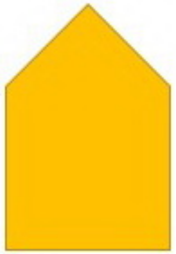
FTA symbols



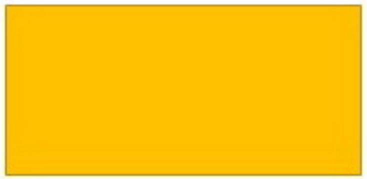
Basic event



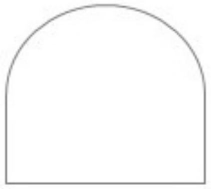
Conditioning event



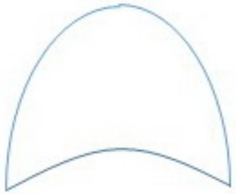
External event



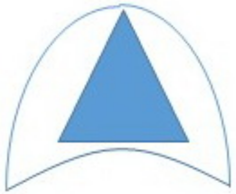
Intermediate event



AND

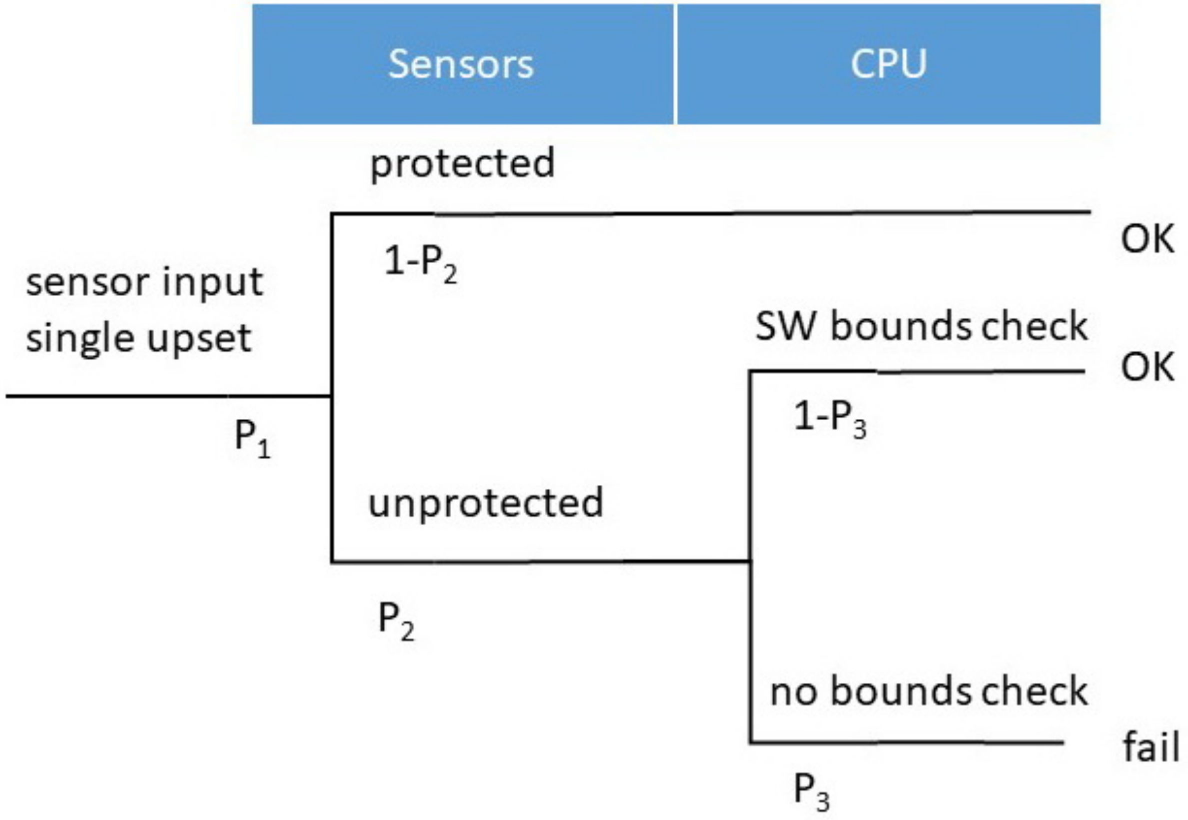


OR



XOR

Event tree


Sensors
CPU
protected
1-P_2
OK
sensor input
single upset
SW bounds check
OK
1-P_3
P_1
unprotected
P_2
no bounds check
fail
P_3

Failure Mode Effects Analysis worksheet

Function	Potential Failure Mode	Potential Effect(s) of Failure	S	Potential Cause(s) of Failure
Function being analyzed	Detailed description of failure conditions	Results of failure	Severity	Causes of failure

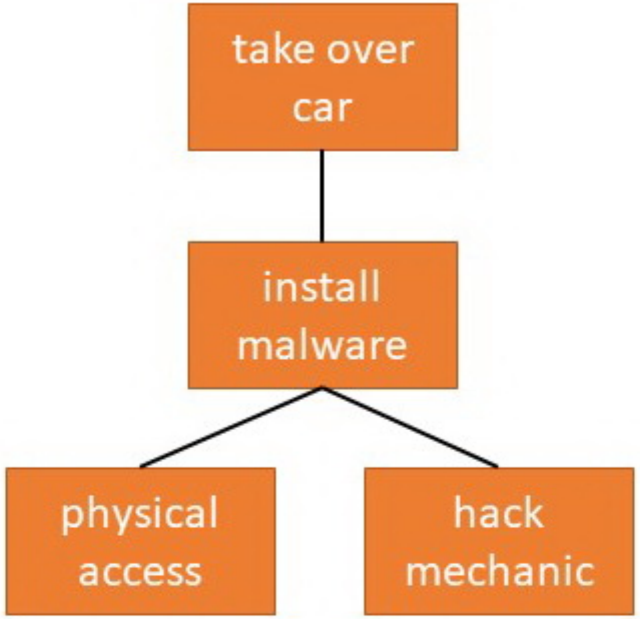
O	Current Process Controls	D	RPN	CRIT
Probability of failure due to occurrence of this failure	Existing means to mitigate failure	Ability to detect cause or failure mode	Risk Priority Number = $S \times O \times D$	Initial criticality assessment

Recommended Action(s)	Responsible Party and Target Completion Date
Actions to take	Who is responsible, when task should be finished

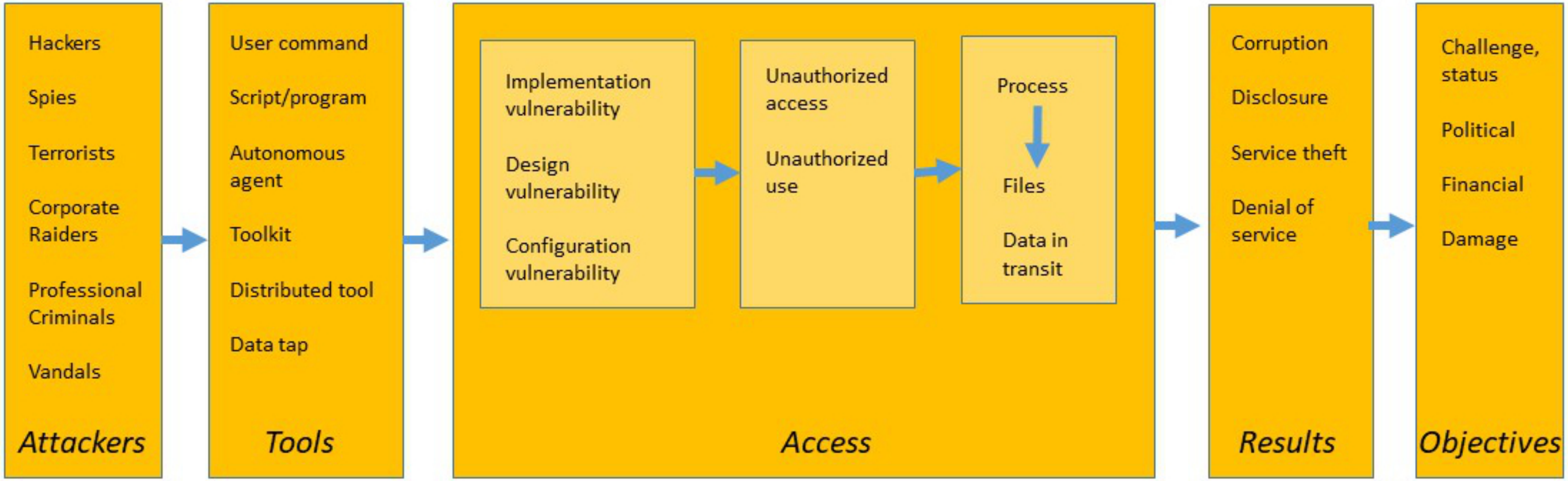
Action Taken	S	O	D	RPN	CRIT
How issue was resolved	Final severity	Final occurrency probability	Final detection ability	Final Risk Priority Number	Final risk assessment

Attack tree

- Introduced by Schneier.
- Branches are OR-combination by default but may be marked as AND.
- Nodes can be labeled for likelihood, cost of attack, special equipment, *etc.*



Howard’s model of computer and network attacks



Hackers

Spies

Terrorists

Corporate Raiders

Professional Criminals

Vandals

Attackers

User command

Script/program

Autonomous agent

Toolkit

Distributed tool

Data tap

Tools

Implementation vulnerability

Design vulnerability

Configuration vulnerability

Unauthorized access

Unauthorized use

Process

Files

Data in transit

Access

Corruption

Disclosure

Service theft

Denial of service

Results

Challenge, status

Political

Financial

Damage

Objectives

Microsoft STRIDE model

- Part of Security Development Lifecycle.
- Threats:
  - Spoofing.
  - Tampering.
  - Repudiation.
  - Information Disclosure.
  - Denial of Service.
  - Elevation of Privilege

NIST

- NIST Framework for Improving Critical Infrastructure Cybersecurity defines activities for managing cybersecurity goals and processes.
  - Asset vulnerabilities are identified and documented.
  - Threat and vulnerability information is collected from a variety of sources.
  - Internal and external threats are identified and documented.
  - Potential impacts on business and the likelihood of those events are identified.
  - Risk is assessed based on threats, vulnerabilities, likelihoods, and impacts.
  - Risk responses are formulated and prioritized.

- *NIST Guide to Industrial Control Systems (ICS) Security* proposes risk management operating at organization, business process, and IT/ICS levels.

Intrusion kill chain (Hutchins *et al*.)

- Reconnaissance identifies targets.
- Weaponization makes use of a Trojan to deliver a payload.
- Delivery sends the weapon to the target.
- Exploitation triggers the intrusion code on the target.
- Installation provides a persistent presence of the adversary on the target.
- Command and control (C2) provide remote information on and control over the attack code.
- Actions on objectives perform the desired operations on the target.

Cyber kill chain

- Gartner defined cyber kill chain.
- Modified cyber kill chain for industrial control systems (Armando and Compagna).
  - Stage 1 prepares and executes a cyber intrusion.
  - Stage 2 develops and executes attack on industrial control system:
    - Attack development and tuning, validation, and the attack proper.
  - Constituent actions for attack phases: enabling includes triggering and delivering; initiating the attack includes modifying and injecting; supporting the attack includes hiding and amplifying

Certification

- May be provided by:
  - Companies (Underwriters Laboratory).
  - Professional societies (IEEE, SAE).
  - Other organizations (ISO).
  - Government agencies (FAA).

Aircraft certification

- U. S. certification based on FAR Part 21.
- Aircraft certification:
    - Type certificate for design.
    - Production certificate for production.
    - Airworthiness certificate for production and maintenance of the aircraft.
- Type certification:
    - Design information, inspection and maintenance plans, flight tests.
    - Supplemental type certificates can certify post-manufacture modifications.
- SAE ARP4761 provides guidelines for safety assessment of civil aircraft.
- FAR Part 23 governs maintenance and alteration.
    - Any item permanently attached to the aircraft must be certified.
    - Airframe and Powerplant Mechanic is certified to sign off maintenance.

Certification and systems analysis

- FAR 25.1309 requires that airplane systems and associated components must be designed so that a failure that would prevent continued safe flight and landing is *extremely improbable* and that the occurrence of any other failure conditions that would reduce the capability of the airplane or ability of the crew to cope with adverse operating conditions is *improbable.*

Fail-safe design

- Designed integrity and quality including life limits.
- Redundancy or backup systems.
- Isolation of systems, components, and elements.
- Proven reliability so that multiple, independent failures are unlikely to occur during the same flight.
- Failure warning or indication.
- Flight crew procedures for use after failure detection.
- Checkability or the ability to check a component’s condition.
- Designed failure effects limits to limit the safety effects of a failure.
- Designed failure path to control and direct the effects of a failure so as to limit its safety impact.
- Margins or factors of safety.
- Error tolerance.

DO-178C

- Used for avionics software certification in U. S., Canada, Europe.
- Modified V process:
    - Functional requirements, hazard and safety analysis, and functional allocations identify the functions allocated to software and their development assurance level.
    - Software is refined from planning, through requirements, design, coding and integration.
    - The products of software development at various stages are fed into a system safety assessment that may update the system functional and hazard/safety analysis.
    - Software is verified, then the system is verified, with the results of software verification feeding into the system safety assessment.

- Design Assurance Level (DAL):
- Level A is catastrophic, generally with airplane loss and possible deaths.
- Level B is hazardous, reducing system performance or the ability of the crew to operate the aircraft.
- C is major, significantly reducing the safety margin or increasing crew workload.
- D is minor, slightly reducing the safety margin or increasing crew workload.
- E is anomalous behavior that has no safety effect on the aircraft or pilot workload.

ASTM F3269-17

- Defines best practices for certification of UAVs with complex functions (machine learning, *etc.*).
    - Run-time assurance architecture includes recovery control functions.

Medical devices

- Software components of medical devices or software-as-a-medical device must be validated.
- Software used in device production or manufacturing quality systems must be validated.
- Recalls between 1992-1998:
  - 3140 total.
  - 242 attributable to software, 192 due to software defects introduced by changes after the initial release.

ISO 9000

- Family of quality standards.
- Principles:
  - Customer focus to understand the needs of customers and align organizational objectives accordingly.
  - Leadership to establish a vision and challenging goals.
  - Engagement of people to use their abilities and make them accountable.
  - A process approach to activities.
  - Continual improvement of organizational capabilities.
  - Evidence-based decision making.
  - Manage relationships with suppliers.
- ISO/IEC 15504 Automotive SPICE is process assessment model for automotive embedded computing systems:
  - Primary life cycle processes for supplier/customer interface.
  - Organizational life cycle processes.
  - Supporting life cycle processes.

Safety design processes

- Software System Safety Handbook (Joint Services Computer Resources Management Group).
  - Safety planning by customer is an iterative process.
  - Alternates between requirements/safety policy and software safety plans.

- ISO 26262 is for functional safety of automotive E/E systems.
  - Guidelines only.
  - Subsystem’s hazards are identified and safety goals are specified.
  - QM for goals that can be achieved using a standard quality management system, levels A-D otherwise, with A least critical.
- ASTM F3153 describes system-level test process for avionics systems safety.

Software safety processes (Tighlman *et al.*)

- The software critical index of each safety-significant function is determined.
- Software requirements hazard analysis derives the software requirements necessary to provide a safe implementation and mitigate hazards.
- Software architectural hazard analysis is conducted on architectural documents and requirements. It is performed before the preliminary design review.
- Software design hazard analysis expands the analysis to consider the planned implementation; it reviews each identified hazard and looks for new hazards. This step is performed before the critical design review.
- Code level hazard analysis analyzes safety-significant variables, typing, code flow analysis, and error processing.
- Operator documentation safety review reviews user documents for adequacy and to identify additional hazards introduced by the documents.
- Software safety testing verifies and validates all software safety requirements.
- Formal review gives evidence for the process and resultant risk level of the design.

Safety case

- Alternate safety analysis methodology.
  - Goal-based or evidence-based.
  - Part of burden-of-proof on designers and managers.

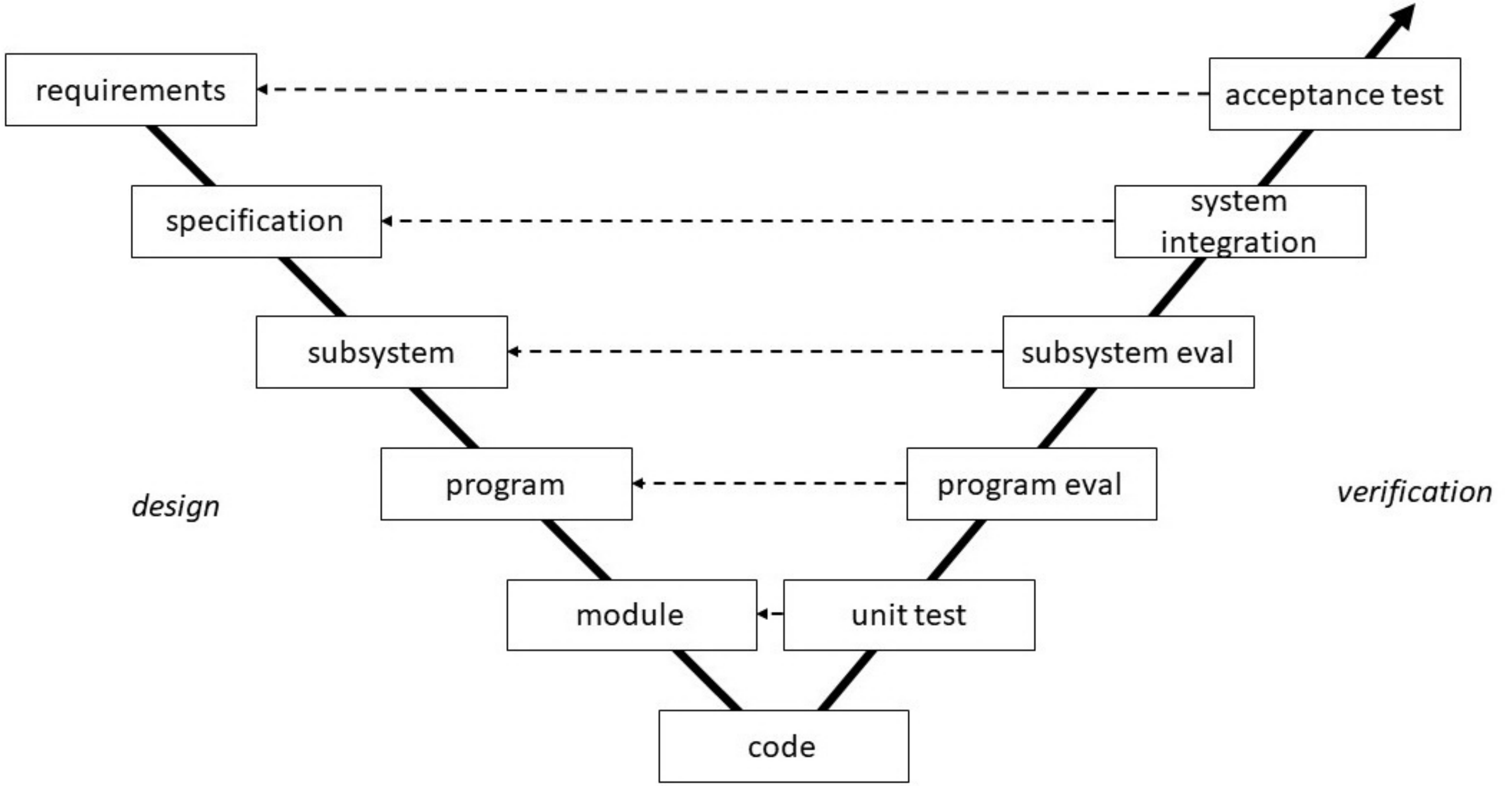
- The aim of the safety case.
- The audience for the safety case and why it is being written.
- The scope of the document.
- A description of the system and its environment.
- If created for a modification the system, a justification for the change.
- A safety argument.
- Supporting safety evidence.
- Caveats, assumptions, and limitations.
- Conclusions.

MISRA

- Coding-level guidelines for automotive software.
- MISRA Generic Modeling Design and Style Guidelines for directory names, file names, *etc.*

- MISRA C guideline examples:
  - No unreachable code.
  - A *typedef* name should be unique.
  - Macro and identifier names should be unique.
  - All *if…else* if constructs are terminated with an *else* clause.

V methodology (MISRA, ISO 26262)



Security design processes

- **Attack surface** is set of interfaces (**attack vectors**) or locations where an attacker may extract or inject data.
- Bell and LaPadula analyzed security using set theory.

Security design principles (Saltzer)

- *Economy of mechanism* reduces chances of flaws and faults that could compromise a security mechanism.
- *Fail-safe defaults* require that permission should be explicit and exclusion the default.
- *Complete mediation* ensures that every object access must be checked to ensure that the access is allowable*.*
- *Open* design forbids reliance on what is now known as security-through-security.
- *Separation of privilege* requires multiple keys for privilege.
- *Least privilege* causes programs and users to operate under the least set of privileges required to complete the task.
- *Least common mechanism* minimizes the commonality of mechanisms among users.
- *Psychological acceptability* promotes ease of use.
- *Work factor* identifies the amount of effort required for a hacker to subvert a mechanism.
- *Compromise recording* creates audit trails.

MULTICS ring policy

- Concentrates on direct modification of information, not indirect modification.
- Each information subject (module) and object (repository) is given an integrity level that does not change during its lifetime.
- A subject may modify only an object whose integrity level is less than or equal to its own.

NIST Platform Firmware Resliency Guidelines

- Security mechanisms are based on roots of trust or rooted chains of trust.
- Changeable firmware shall rely on root of trust update.
- Devices with intrusion detection shall make use of root of trust for the detection services.
- Recovery shall rely on root of trust.
- The update mechanism will be the only mechanism for updating device firmware.
- Flash shall be protected to be unmodifiable outside of an authenticated/secire update mechanism.
- Protection mechanisms cannot be bypassed.
- Write protection of field non-upgradeable memory shall not be modifiable.
- Critical data shall be modifiable only through the device or defined interfaces.
- A successful attack on firmware shall not compromise the device’s detection capability.
- The device shall perform integrity checks on critical data before use.
- Firmware recovery mechanisms shall resist attacks against critical data or primary firmware image.
- Critical data recovery mechanisms shall resist attacks.

NIST guidelines

- Incident response recommendations:
    - Creating a policy and plan for incident response.
    - Developing procedures to handle and report incidents.
    - The development of guidelines for communicating about incidents with outside parties.
    - Creation of a team structure and staffing model.
    - Establishing relationships with other parts of the organization.
    - Determining the services to be provided by the incident response team.
    - Staffing the response team and providing training.
- Guidelines for Smart Grid Cybersecurity identifies seven domains, logical interfaces are categorized based on their security characteristics.
- NIST SP 800-53 provides U. S. Federal organizations with guidance on the management of information security risk.
    - Based on a muti-tier model including organizations, mission/business processes, and information systems
- ICS security:
    - develop the security business case; build and train a cross-functional team; identify charter and scope of team; define ICS policies and procedures; implement an ICS security risk management framework; provide training for ICS staff

Zero-day vulnerability

- A **zero-day vulnerability** is not known to those who are interested in mitigating the vulnerability.
- Once known, referred to as **zero-day exploit**.
- Possible detection methods:
  - Statistical analysis of attack profiles.
  - Signatures of known exploits.
  - Analysis of exploit’s behavior relative to the target.

Compare and contrast

- Safety risk analysis is driven by requirements and is concentrated on the requirement phase.
- Security vulnerability analysis is driven by the system structure and concentrated at the architecture phase and, to some extent, at coding.

Compare and contrast, *cont'd.*

- Many existing, installed devices are insecure, creating long-term problems.
- Traditional IT applications are often transaction-oriented. CPS and IoT systems may not be easily modeled as transactions.
- Physical plant attacks:
  - Timing attacks.
  - Replay attacks.
- Safety design often assumes that system designers control characteristics of their components, may not be true for embedded software.

Incident reporting

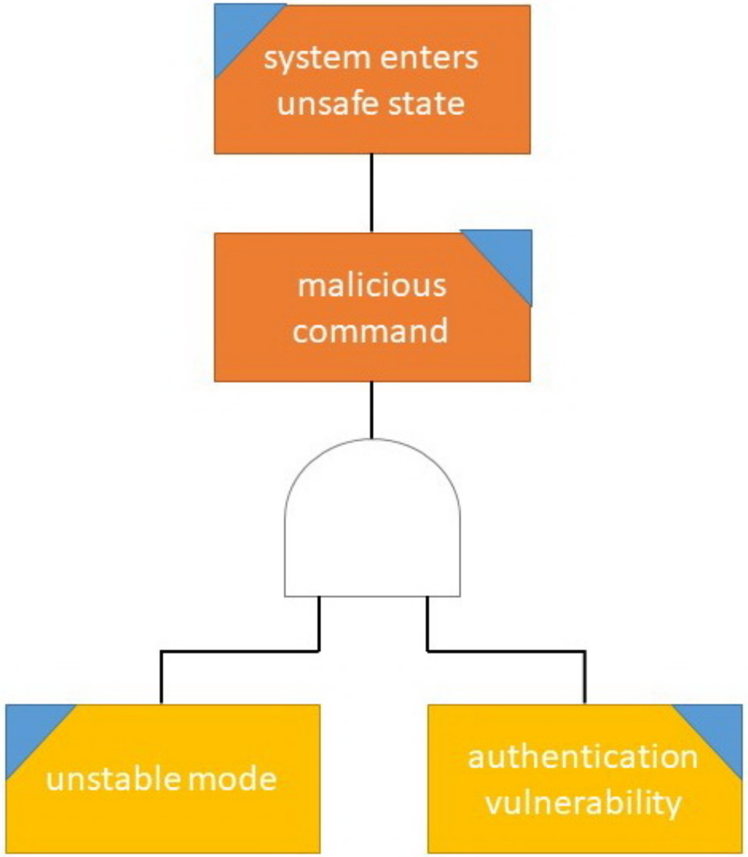
- Safety incident reporting is widely practiced, may be mandated by law or regulation.
- Computer security violations may be collected by several different organizations:
    - U. S. CERT.
    - Internet Crime Complaint Center.
    - DHS National Infrastructure Coordinating Center.
- Private organizations may not always report security breaches.
- V methodology assumes that risks are known early in the design process.

Security threats

- Hierarchy of security concerns:
  - A vulnerability is a system security weakness.
  - A threat is a possible means to exploit a vulnerability.
  - An exploit is software that can be used by an attacker to take advantage of a vulnerability.
  - An attack is an implementation of a threat.

Compound threats

- A **compound threat** includes both security and safety threats.
- Allow attackers to extend their attacks into the physical world.
- Threat tree for malicious exercise of an unstable mode.



Threat analysis models

- Find cutsets of threat trees:
  - Combination of safety and security.
  - All-safety or all-security.
- Generalize risk priority number:
  - S_f, O_f, D_f for safety, occurrence, and detection factors.
  - Vulnerability priority number $VPN = S_v \times O_v \times D_v$
  - Threat priority number $TPN = RPN + VPN$

- Functional Hazard Analysis Worksheet:
  - The security vulnerability.
  - Tools and methods used for the attack.
  - Access mode for the attack.
  - Results of the attack.
  - Relationship of the attack to safety.

Functional Hazard Analysis Worksheet

Threat ID	Life cycle phase	Activity	State/Mode	Function
Identifier	Phase analyzed by risk assessment	Actions performed within life cycle phase	System state or mode for the hazard	System function

Vulnerability	Attack Tools	Access Mode	Results of Attack	Relationship to Safety
Security vulnerability	Tools and methods used for attack	Unauthorized access, use, etc.	System compromise	How security violation results in safety failure

Functional failure	Threat Description	System Item(s)	Causal Factor Description	Mishap
Detailed description of failure mode	Detailed description of threat conditions	Portion of the system	Causes of failure	Description of failure

Effect(s)	Existing Mitigations	Software Control Category	Initial TPN	Software Criticality Index
Effects on life, limb, property	Existing means to mitigate failure	Degree of autonomy of software function	Initial threat assessment	Criticality

Target TPN	Causal Factor Risk Level	Recommended Mitigations	Comments	Follow-On Actions
Projected threat after mitigation	Potential for causal factors to occur	Methods to reduce threat	Relevant additional information	Further work to better understand threat

Failure Mode Effects Analysis worksheet for threats

Function	Potential Failure Mode	Potential Effect(s) of Failure	S_f, S_v	Potential Cause(s) of Failure
Function being analyzed	Detailed description of failure conditions	Results of failure	Severity of failure, vulnerability	Causes of failure

Vulnerability	Attack Tools	Access Mode	Results of Attack	Relationship to Safety
Security vulnerability	Tools and methods used for attack	Unauthorized access, use, etc.	System compromise	How security violation results in safety failure

O_f, O_v	Current Process Controls	D_f, D_v	TPN	CRIT
Probability of failure due to occurrence of this failure/vulnerability	Existing means to mitigate failure	Ability to detect cause or failure mode/attack	Threat Priority $TPN = RPN + VPN$	Initial criticality assessment

Recommended Action(s)	Responsible Party and Target Completion Date
Actions to take	Who is responsible, when task should be finished

Action Taken	S_f, S_v	O_f, O_v	D_f, D_v	TPN	CRIT
How issue was resolved	Final failure, vulnerability severity	Final occurrency probabilities	Final detection abilities	Final Threat Priority Number	Final threat assessment

Cyber-physical kill chain

- Reconnaissance identifies both physical and cyber targets. Attack development can take into account safety risks that could be exploited.
- Weaponization may in some cases make use of physical properties of the system to deliver an attack.
- Delivery may or may not depend on physical access. In some cases, delivery may involve interfering with physical objects (attempted theft of a BMW automobile proceeded over two days [Roo18]).
- Exploitation may make use of a combination of cyber and physical methods.
- Installation may provide a persistent presence or a presence over a limited time span. Installation may also include methods, such as replay, to hide the attack.
- Command and control may allow the attacker to remotely assess physical damage and update the direction of the attack.
- Actions related to safety and security include the security actions of detect, deny, disrupt, degrade, deceive, destroy as well as the safety actions of detect, mitigate.

Safety accident reporting

- The Department of Energy maintains several databases, including Safety Basis Information System (SBIS). The Department operates a process to identify Suspect/Counterfeit and Defective Items.
- The Federal Aviation Administration maintains an accident and incident database (https://www.faa.gov/data_research/accident_incident/).
- The National Transportation Safety Board (NTSB) provides a database on accident reports on various transportation modes (https://www.ntsb.gov/investigations/AccidentReports/Pages/AccidentReports.aspx) and a database specific to aviation accidents (https://www.ntsb.gov/_layouts/ntsb.aviation/index.aspx).

Example NTSB crash report

- An executive summary provides a short description.
- A factual information section describes the crash narrative, injuries, emergency response, motorcoach, highway and grade crossing, railroad operations, motor carrier operations, motorcoach driver, weather and roadway conditions.
- An analysis section considers the motorcoach driver and train crew, the grade crossing, and emergency egress and extrication.
- A conclusions section describes finding and probable cause.
- A recommendations section provides new recommendations as well as recommendations reiterated and reclassified in the report.
- An appendix describes the investigators and parties to the investigation.
- A list of references is provided, as is a list of figures and tables as well as acronyms and abbreviations.

Software vulnerability databases

- The NIST National Vulnerability Database (NVD) (https://nvd.nist.gov/) is the U. S. government repository of vulnerability management data.
- Common Vulnerabilities and Exposures CVE® (https://cve.mitre.org) is a database of publicly known vulnerabilities. CVE data is used in NVD.
- The CERT Vulnerability Notes Database (https://www.kb.cert.org/vuls/) provides a set of Vulnerability Notes that include technical descriptions, remediation notes, and affected vendors.
- The Common Vulnerability Scoring System (CVSS) defines metrics for IT-oriented vulnerabilities.

Example CVE entry

- The current description provides a summary.
- Impact describes severity and metrics for CVS versions 3.0 and 2.0.
- References to advisories, solutions, and tools are provided.
- Vulnerability type is identified.
- Vulnerable software and versions are provided.

Improper authorization threats

- Authorization domain governs access to data.
    - Domain size should balance efficiency vs. protection.
    - Domains should be designed considering both safety and security.
- Software safety threats:
    - Poor numerical algorithms.
    - Hardware flaws.
    - Cosmic rays and other externally-induced faults.

Iterative threat analysis

- Safety typically operates over long time scales while security reacts quickly.
- Revisit threat analysis:
  - Several times during design.
  - Post-deployment.
- Requirements safety/security:
  - define a post-deploy schedule for vulnerability analysis;
  - develop a plan to handle zero-day vulnerabilities;
  - develop critieria under which these plans are revisited both during later design phases and after deployment.

Architecture phase

- Architectural design:
  - Uses threat analysis.
  - Updates threat analysis to include architectural information.
- Newly-identified vulnerability possibilities:
  - The new vulnerability may be a variation of one that was previously considered in the design process.
  - The new vulnerability may present new threat cases.

Threat mitigation

- Pre-deployment:
  - McGraw identifies static analysis, risk analysis, penetration testing for security, risk-based software testing.

- Post-deployment:
  - Emphasize resilience of safety-critical systems to both security and safety threats.
  - Test software updates for physical plant issues

Definitions

- Functional safety: risks resulting from faults or design flaws.
- Reliability: probability of a system being able to perform its intended function.
- Availability: percentage of time over which system is capable of performing its intended function.
- Certification: legal or regulatory process in which a system is deemed to meet certain criteria, such as safety.

Risk management

- Risk: potential for loss or injury.
- Risk can be minimized but not avoided.
  - Risk management may require trade-offs.
- Risk management approaches:
  - Design for minimum risk.
  - Incorporate safety devices.
  - Provide warning devices.
  - Develop procedures and training.

Safety management

- **Risk planning** provides an organized, managed process for the identification and mitigation of risks.
- **Risk assessment** identifies potential risks through system engineering documents and lessons learned.
- **Risk analysis** assesses the likelihood of risks and their potential consequences.

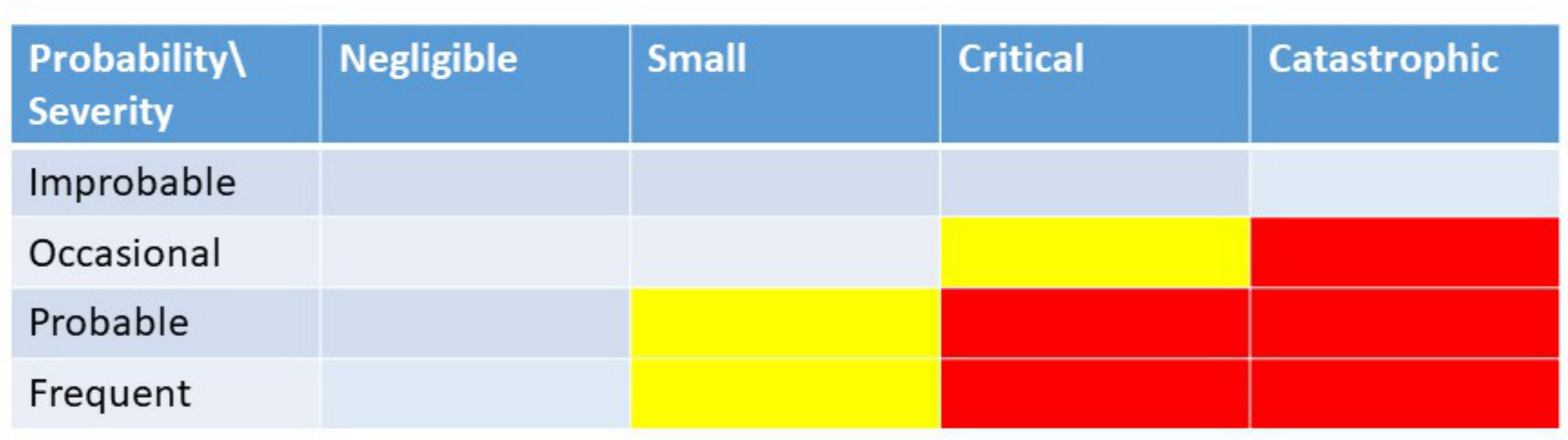
Risk model

- **Controlled** or **eliminated risk** can be managed through design process.
- **Residual risk** cannot be controlled or eliminated.
  - May come from identified or unidentified sources.
- **Unacceptable risk** cannot be tolerated.

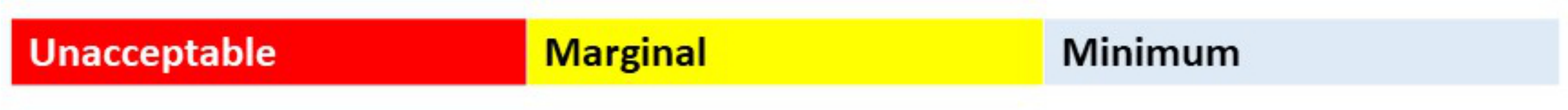
Failure modes and effects analysis

- A hazard is a precondition for a mishap.
- Failure modes and effects analysis (FEMA) identifies likelihood and severity of hazards.
  - Results can be used for risk modeling and management.

Hazard risk index matrix



Probability\ Severity	Negligible	Small	Critical	Catastrophic
Improbable				
Occasional			🟨	🟥
Probable		🟨	🟥	🟥
Frequent		🟨	🟥	🟥



🟥 Unacceptable	🟨 Marginal	Minimum

Risk priority number

- Product of three factors:
  - The severity of the risk.
  - The likelihood of occurrence of the risk.
  - The system’s ability to detect the failure mode or its cause.
- Risk priority number $RPN = S \times O \times D$.