

# LECTURE 15

## Software requirements Engineering

Hazard Analysis and Safety-Critical System Requirements

### **Instructor:**


Dr. Natalia Chaudhry,

Assistant Professor, PUCIT, University of the Punjab, Lahore.

- Requirements not only have to deal with events which are unexpected; they also need to anticipate the consequences of things going wrong.
- Safety-critical systems are systems whose failure could result in loss of life, significant property damage, or harm to the environment. Examples include avionics, automotive safety systems, nuclear power plant controls, and medical devices.
- **Characteristics:**
  - High reliability and availability
  - Strict regulatory compliance
  - Real-time operation
  - Redundancy and fail-safes

## Problems and Issues in Safety Critical Systems

- As complex systems can not be controlled with 100% confidence the possibility that the system may depart from a safe state has to be anticipated.
- This brings in several new sets of requirements which more conventional systems do not need to worry about:
  - Requirements to detect unsafe states: monitoring.
  - Requirements for diagnostic support: how to interpret the cause of the problem.
  - Requirements to contain the immediate consequences of failure and protect people: e.g. evacuation procedures, shutting down damaged systems, fail-safe processes.
  - Recommendations for returning the system to a safe state, including repair of damaged components.
  - Methods for dealing with the longer term consequences of failure: injury, environmental pollution.

- 
- Hazard Analysis is a systematic process to identify potential hazards, assess their risks, and develop strategies to mitigate those risks.
  - It ensures that all potential dangers are recognized and addressed early in the system development life cycle.

## Steps in Hazard Analysis:

### **Hazard Identification:**

- Identify possible hazards through brainstorming, historical data analysis, and expert consultations.

### **Risk Assessment:**

- Evaluate the probability and severity of each hazard.
- Classify risks to prioritize mitigation efforts.
- Use risk matrices to quantify and categorize risks.

### **Hazard Mitigation:**

- Develop strategies to eliminate or reduce hazards.
- Implement design changes, safety mechanisms, and operational procedures.
- Ensure redundancies and fail-safes are in place.



**practical example of hazard analysis in the context of  
developing an autonomous vehicle.**

## Step 1: Hazard Identification

### 1. Brainstorming:

- Gather a team of experts including software developers, safety engineers, and domain experts.
- Identify potential hazards related to the vehicle's operation. For instance:
  - Sensor failure
  - Software bugs
  - Obstacle detection errors
  - Unintended acceleration

### 2. Historical Data Analysis:

- Review data from past incidents involving autonomous or semi-autonomous vehicles.
- Common hazards identified include issues like unexpected pedestrian crossings, adverse weather conditions affecting sensor performance, and GPS signal loss.

### 3. Expert Consultations:

- Consult with industry experts and regulatory bodies to understand potential hazards.
- Use insights from these consultations to refine the list of potential hazards.



## Identified Hazards:

- Sensor failure (e.g., LIDAR, cameras)
- Software bugs leading to incorrect decision-making
- Communication failures between components
- GPS signal loss or spoofing
- Battery failure or fire
- Unexpected pedestrian or animal crossing
- Adverse weather conditions affecting sensor accuracy



## Step 2: Risk Assessment:

**Goal:** Evaluate the probability and severity of each identified hazard to prioritize mitigation efforts.

### Steps:

- **Risk Matrix:** Use a risk matrix to classify each hazard based on its likelihood and impact.

### Example Classification:

- **Sensor Failure:** High probability, high severity (loss of situational awareness)
- **Software Bug:** Medium probability, high severity (incorrect vehicle actions)
- **Battery Failure:** Low probability, high severity (fire hazard)
- **Unexpected Pedestrian Crossing:** Medium probability, medium severity (potential collision)

### Step 3: Hazard Mitigation:

**Goal:** Develop and implement strategies to eliminate or reduce the risks associated with each hazard.

#### Mitigation Strategies:

##### Sensor Failure:

- **Redundancy:** Use multiple types of sensors (LIDAR, radar, cameras) to provide overlapping coverage. If one sensor fails, others can compensate.
- **Health Monitoring:** Continuously monitor sensor health and functionality. Trigger alerts and fallback procedures if a failure is detected.

##### Software Bugs:

- **Rigorous Testing:** Implement extensive testing protocols, including unit tests, integration tests, and real-world simulations.
- **Formal Verification:** Use formal methods to mathematically verify the correctness of critical algorithms.
- **Fail-Safe Modes:** Design the vehicle to enter a safe state (e.g., slow down and pull over) if a software anomaly is detected.

## Battery Failure:

- Thermal Management:** Design advanced thermal management systems to prevent overheating.
- Battery Health Monitoring:** Continuously monitor battery health and performance to detect signs of degradation or failure early.
- Fire Suppression:** Equip the vehicle with fire suppression systems to contain any battery fires.

## Unexpected Pedestrian Crossing:

- Advanced Detection Algorithms:** Develop algorithms that can quickly and accurately detect pedestrians and other obstacles.
- Predictive Modeling:** Use predictive models to anticipate the behavior of pedestrians and other road users.
- Emergency Braking:** Implement emergency braking systems that can rapidly stop the vehicle if a collision is imminent.



# **Safety-Critical System Requirements**

## **1. Functional Requirements:**

- Define what the system should do, focusing on safety aspects.
- Examples: Automatic shutdown in case of a detected fault, real-time monitoring.

## **2. Non-Functional Requirements:**

- Performance: Ensure timely responses to critical events.
- Reliability: Maintain high uptime and quick recovery from failures.
- Security: Protect against malicious attacks that could cause safety hazards.

## **3. Regulatory Requirements:**

- Compliance with standards and regulations specific to the industry.
- Examples: ISO 26262 for automotive safety, DO-178C for avionics software, IEC 61508 for industrial safety.

# Techniques and Tools

## 1. Model-Based Design:

- Use models to simulate system behavior and identify potential hazards.
- Tools: MATLAB/Simulink, SysML.

## 2. Formal Methods:

- Apply mathematical techniques to verify correctness and safety properties.
- Tools: Model checkers, theorem provers.

## 3. Safety Analysis Tools:

- Software tools that facilitate hazard analysis and risk management.
- Examples: Reliability Workbench, PHA-Pro.


## Case Studies

### **1. Therac-25:**

- A radiation therapy machine that caused patient deaths due to software errors.
- Lessons: Importance of rigorous testing, hazard analysis, and fail-safes.

### **2. Toyota Unintended Acceleration:**

- Issues with electronic throttle control led to unintended acceleration incidents.
- Lessons: Need for thorough risk assessment, real-time monitoring, and robust design.



**Understanding and implementing Hazard Analysis and Safety-Critical System Requirements are crucial for developing systems that we can trust with our lives.**

**By identifying hazards early, assessing risks accurately, and designing robust safety mechanisms, we can create systems that operate safely and reliably.**