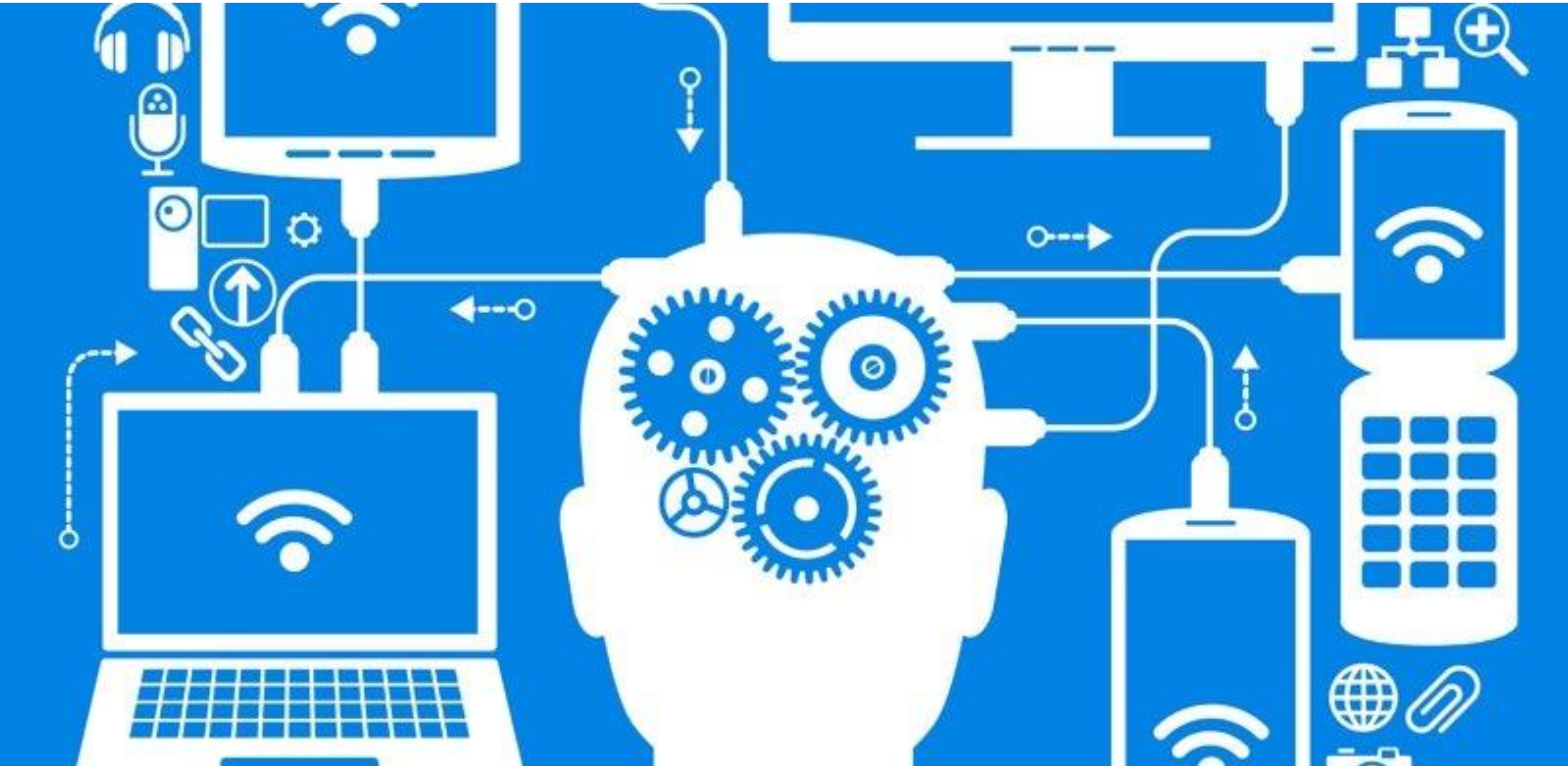
# Introduction to Software Engineering

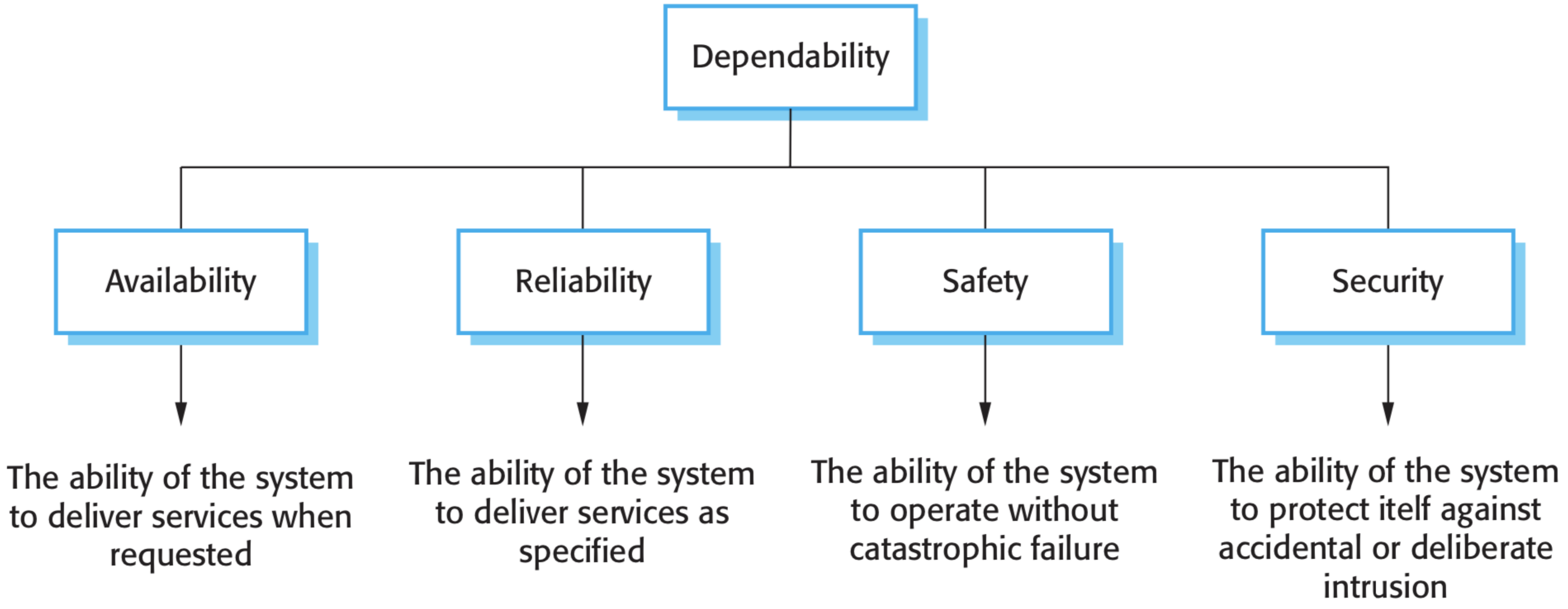
**Elishai Ezra Tsur**
Neuro & Biomorphic Engineering LAB

**Dan Zilberstein**

JERUSALEM COLLEGE OF TECHNOLOGY
LEV ACADEMIC CENTER



# Advanced concepts in Software Engineering: Dependability Attributes


Dependability
Availability
Reliability
Safety
Security
The ability of the system to deliver services when requested
The ability of the system to deliver services as specified
The ability of the system to operate without catastrophic failure
The ability of the system to protect itelf against accidental or deliberate intrusion

## Advanced concepts in Software Engineering: Dependability Threats

***Fault*** is a defect in a system.

***Error*** is a discrepancy between the intended behavior of a system and its actual behavior inside the system boundary.

***Failure*** is an instance in time when a system displays behavior that is contrary to its specification.

## Advanced concepts in Software Engineering: Dependability Means

***Prevention*** is preventing faults being incorporated into a system.

***Removal*** can be sub-divided into two sub-categories: Removal During Development and Removal During Use.

***Forecasting*** predicts likely faults so that they can be removed or their effects can be circumvented.

***Tolerance*** deals with putting mechanisms in place that will allow a system to still deliver the required service in the presence of faults, although that service may be at a degraded level.

## Advanced concepts in Software Engineering: Dependability (other possible attributes)

***Repairability*** System disruption minimized if the system can be repaired quickly. For that to happen, it must be possible to diagnose the problem, access the component that has failed, and make changes to fix that component: **open source software** makes this easier but the reuse of components can make it more difficult.

**Maintainability.** Maintaining the usefulness of a system by changing it to accommodate new requirements.

## Advanced concepts in Software Engineering: Dependability

**Survivability.** The ability of a system to continue to deliver service whilst under attack and, potentially, whilst part of the system is disabled. Work on survivability focuses on identifying key system components and ensuring that they can deliver a minimal service. **Strategies**: resistance to attack, attack recognition, and recovery from the damage.

**Error tolerance.** System has been designed so that user input errors are avoided and tolerated.

## Advanced concepts in Software Engineering: Dependability

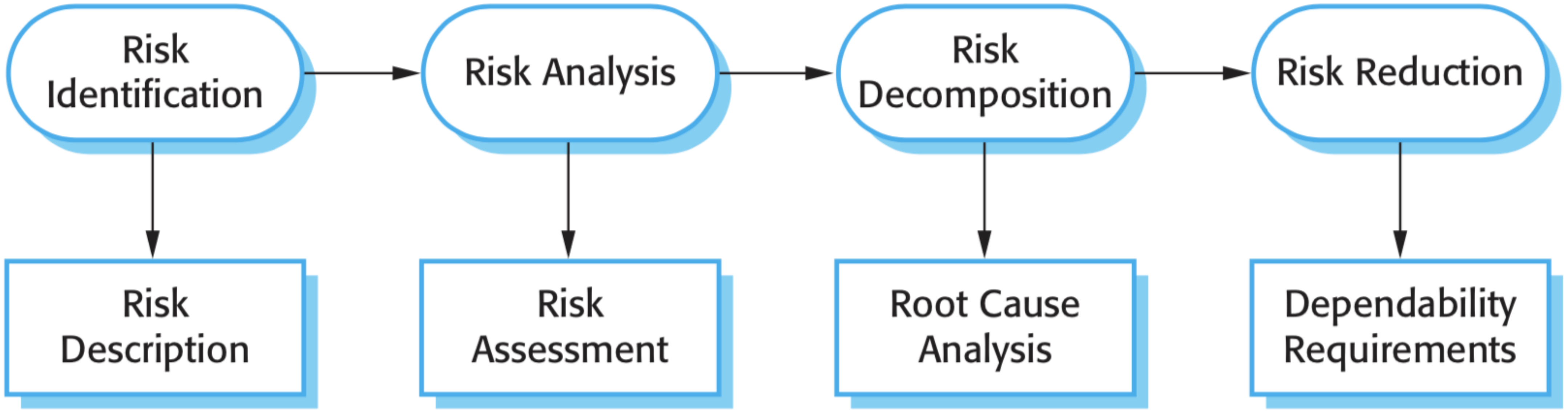
*In September 1993, a plane landed at Warsaw airport in Poland during a thunderstorm. For nine seconds after landing, the brakes on the computer-controlled braking system did not work. The braking system had not recognized that the plane had landed and assumed that the aircraft was still airborne. A safety feature on the aircraft had stopped the deployment of the reverse thrust system, which slows down the aircraft, because this can be dangerous if the plane is in the air. The plane ran off the end of the runway, hit an earth bank, and caught fire.*

*The inquiry into the accident showed that the braking system software had operated according to its specification. There were no errors in the program. However, the software specification was incomplete and had not taken into account a rare situation, which arose in this case. The software worked but the system failed.*

*This illustrates that system dependability does not just depend on good engineering. It also requires attention to detail when the system requirements are derived and the inclusion of special software requirements that are geared to ensuring the dependability and security of a system.*

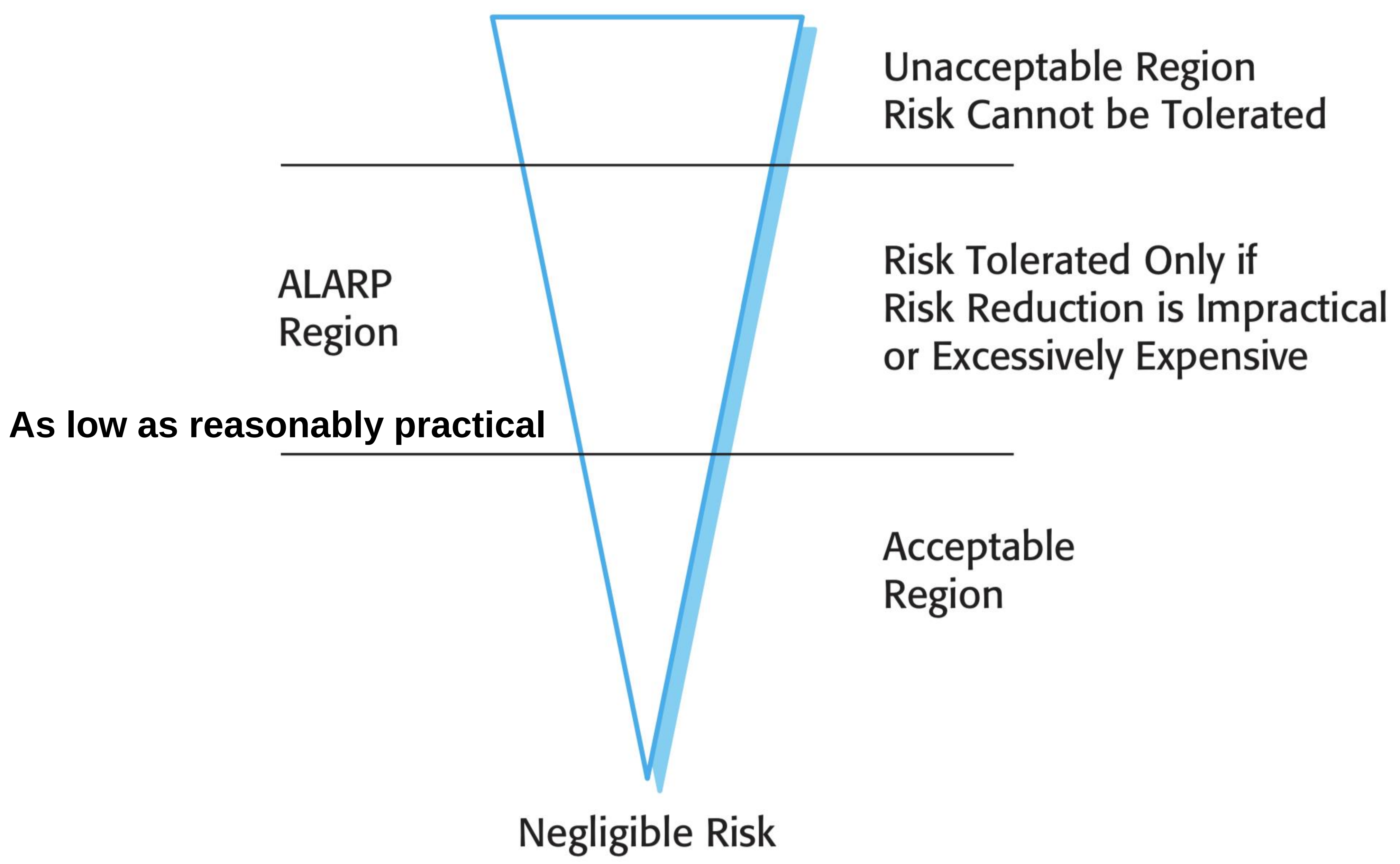
# Advanced concepts in Software Engineering: Dependability (in safety-critical systems)

## Risk-driven approach (remember?…)



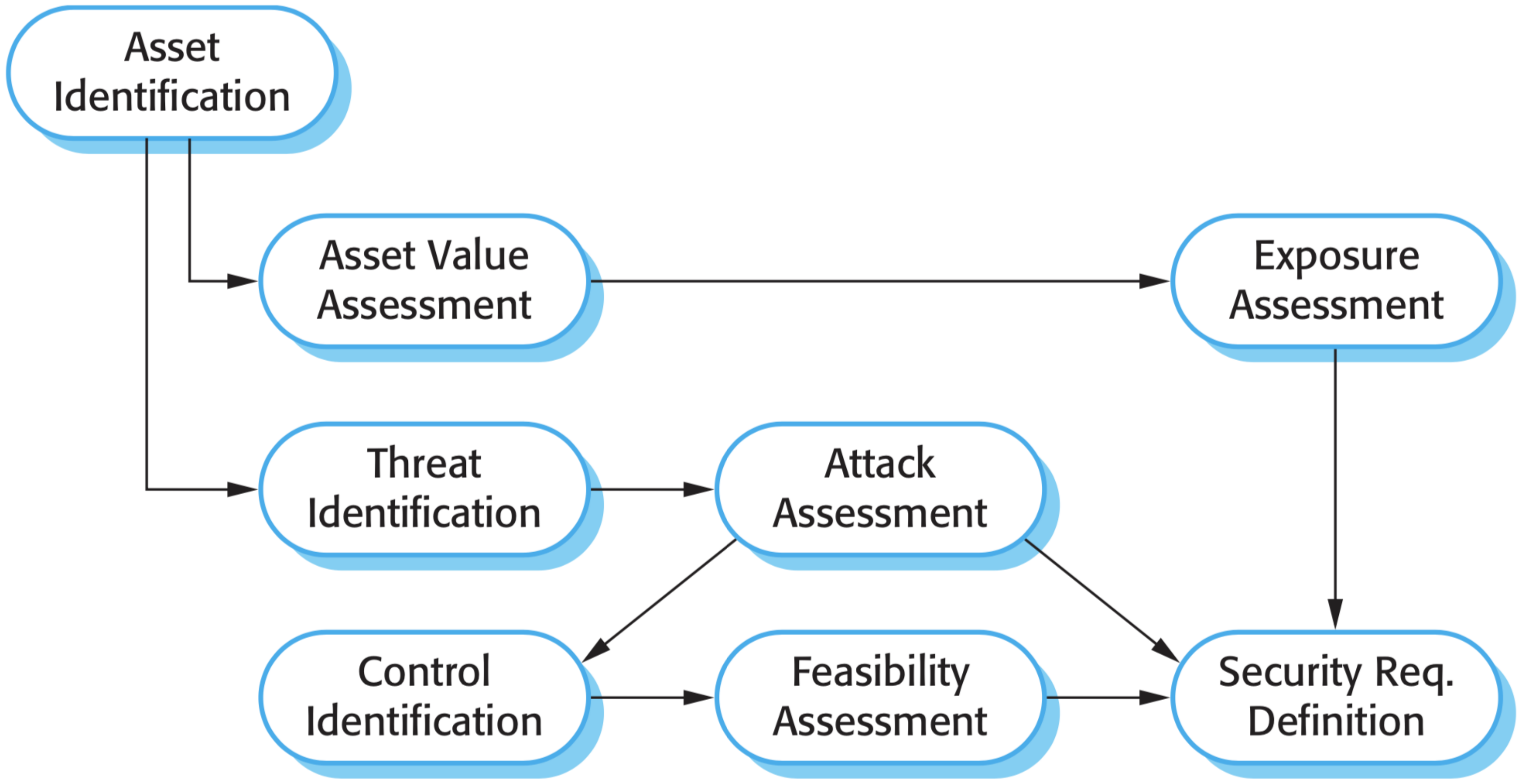
## Advanced concepts in Software Engineering: Dependability (in safety-critical systems)

### The Risk Triangle



## Advanced concepts in Software Engineering: Security

### Risk-driven security requirement



## Advanced concepts in Software Engineering: Dependability (Security)

Dependability and security requirements are of two types:

### 1. Functional requirements

The software's functional security requirements specify a security function that the software must be able to deliver. Obviously, the functional security requirements are a subset of the overall functional requirements.

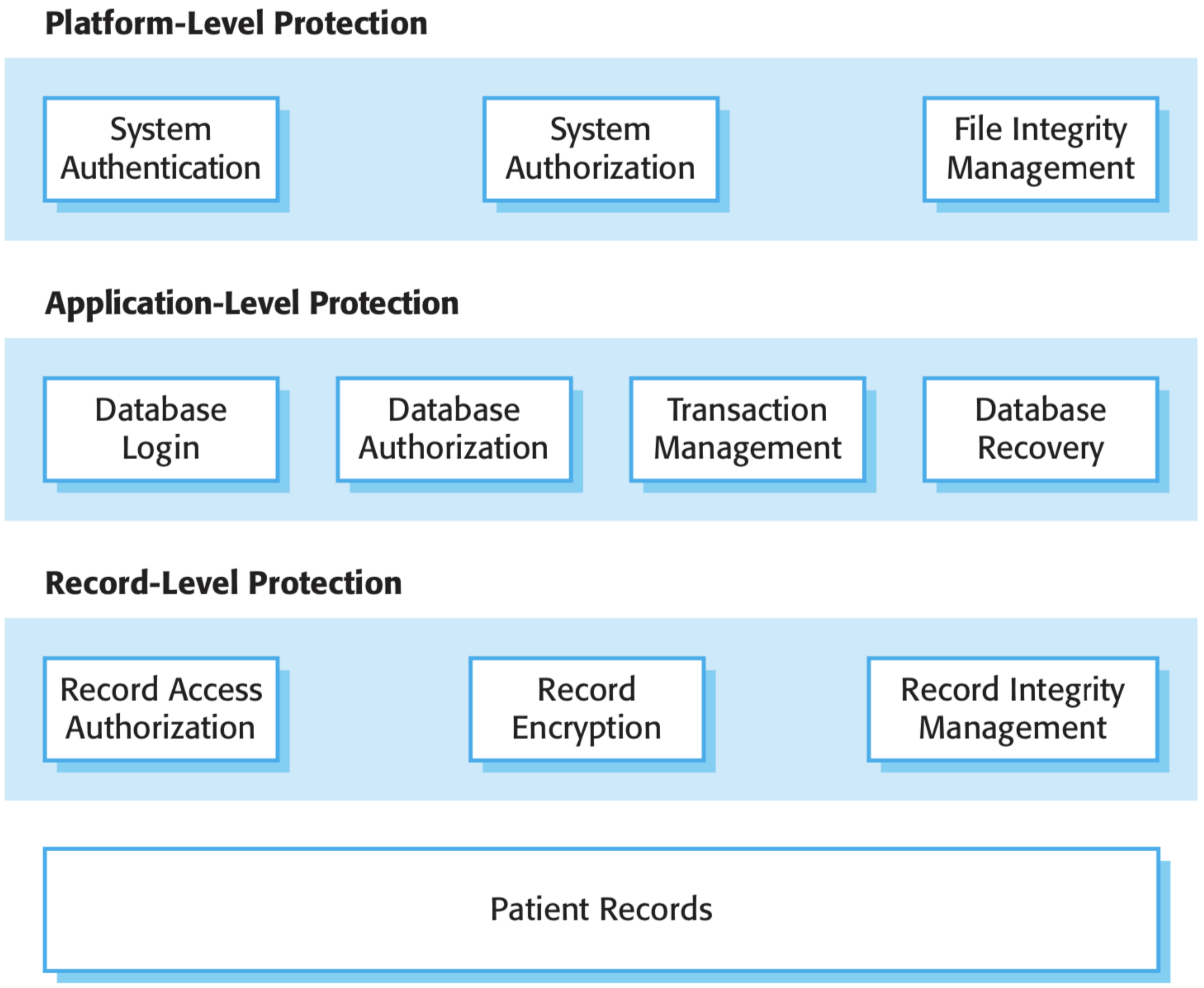
### 2. Non-functional requirements

The non-functional security requirements specify a security quality or attribute that the software must possess. There are 3 types of non-functional security requirements:

- **Security Property Requirements**
- **Constraint/Negative Requirements**
- **Security Assurance Requirements**

# Advanced concepts in Software Engineering: Security

## Security Driven Design



The top level controls access to the platform on which the patient record system runs.

Invoked when access to specific records is required