universidade de aveiro

Critical software

Security Requirements

Robust Software – Nuno Silva

Mestrado em Cibersegurança

Agenda

- Motivation
- Objectives
- Security Requirements
- Security Goals
- Security Requirements Engineering
- Security Policy
- References
- Exercise

Motivation

- Security is a system property. Security is much more than a set of functions and mechanisms. IT security is a system characteristic as well as a set of mechanisms that span the system both logically and physically.
- To ensure that a software solution correctly solves a particular problem, we must initially *fully understand* the problem that needs to be solved, discover *why* the problem needs to be solved and determine *who* should be involved.
- Poorly defined requirements can cause major problems to a project and the organization.
- There are specific techniques for the requirements engineering phase which shall be considered.

Objectives

- Get to know what are security requirements and goals.
- Be able to identify and write down security requirements.
- Be able to verify/validate security requirements for completeness and appropriateness.
- Be aware of the importance of sound and complete security requirements.

Security Requirements

- Refer to the Software Security Lifecycle presentation for the main phases and relevant processes.

- This is largely achieved through a structured risk management process that involves:
  - Identifying information and related assets, plus potential threats, vulnerabilities and impacts;
  - Evaluating the risks;
  - Deciding how to address or treat the risks i.e. to avoid, mitigate, share or accept them;
  - Where risk mitigation is required, selecting or designing appropriate security controls and implementing them;
  - Monitoring the activities, making adjustments as necessary to address any issues, changes and improvement opportunities.

Security Requirements

Phase	Microsoft SDL	McGraw Touchpoints	SAFECode
Education and awareness	Provide training		Planning the implementation and deployment of secure development
Project inception	Define metrics and compliance reporting Define and use cryptography standards Use approved tools		Planning the implementation and deployment of secure development
Analysis and requirements	Define security requirements Perform threat modelling	Abuse cases Security requirements	Application security control definition
Architectural and detailed design	Establish design requirements	Architectural risk analysis	Design
Implementation and testing	Perform static analysis security testing (SAST) Perform dynamic analysis security testing (DAST) Perform penetration testing Define and use cryptography standards Manage the risk of using third-party components	Code review (tools) Penetration testing Risk-based security testing	Secure coding practices Manage security risk inherent in the use of third-party components Testing and validation
Release, deployment, and support	Establish a standard incident response process	Security operations	Vulnerability response and disclosure

Security Requirements

Source: ISO/IEC FCD 25010

Security Requirements

- Security Properties:
  - Confidentiality
  - Integrity
  - Availability
  - Non-repudiation
  - Accountability
  - Authenticity
  - Compliance
  - Privacy
  - Possession
  - Utility
  - Trustworthiness
  - Auditability

- CIA Triad
  - Confidentiality
  - Integrity
  - Availability
- Parkerian Hexad:
  - Confidentiality
  - Possession/Control
  - Integrity
  - Authenticity
  - Availability
  - Utility

  - https://en.wikipedia.org/wiki/Parkerian_Hexad

Security Goals::Threats

- Advanced persistent threat
- Backdoors
- Bootkits
- Computer crime
- Viruses
- Denial of service
- Eavesdropping
- Exploits
- Keyloggers
- Logic bombs
- Malware
- Payloads
- Phishing
- Ransomware
- Rootkits
- Screen scrapers
- Spyware
- Trojans
- Vulnerabilities
- Web shells
- Web application security
- Worms

Security Goals::Threats

- Responses to threats
- Possible responses to a security threat or risk can be:
  - Reduce/mitigate – implement safeguards and countermeasures to eliminate vulnerabilities or block threats;
  - Assign/transfer – place the cost of the threat onto another entity or organization such as purchasing insurance or outsourcing;
  - Accept – evaluate if the cost of the countermeasure outweighs the possible cost of loss due to the threat;

Security Goals::Threats

- A parallel to **Safety**: SRAC – **Safety** Related Application Condition
  - The concept of SRAC is defined on CENELEC EN50129 standard and it is the responsibility of RAMS/**Safety** Engineer to document and deliver to the user.
  - SRACs must be seen as a legal contract associated with the transfer of a device or an installation, with connotations related to **safety**.
  - Its importance requires **robustness** and its treatment must meet expectations regarding the implications that they entail.
  - SRACs clarify the **safety** responsibilities of the entities in charge of the installation, maintenance and operation, that is, of the entire service cycle of the equipment or installation.

  - Now replace Safety by Security

Security Goals::Defenses

- Computer access control
- Application security
  - Antivirus software
  - Secure coding
  - Secure by default
  - Secure by design
  - Secure operating systems
- Authentication
  - Multi-factor authentication
- Authorization
- Data-centric security
- Encryption
- Firewall
- Intrusion detection system
- Mobile secure gateway
- Runtime application self-protection (RASP)

Security Goals::Confidentiality

- The security goal that generates the requirement for protection from intentional or accidental attempts to perform unauthorized data reads. Confidentiality covers data in storage, during processing, and while in transit. NIST SP 800-27 Rev A
- Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. NIST SP 800-160

Security Goals::Integrity

- The security goal that generates the requirement for protection against either intentional or accidental attempts to violate data integrity (the property that data has not been altered in an unauthorized manner) or system integrity (the quality that a system has when it performs its intended function in an unimpaired manner, free from unauthorized manipulation). NIST SP 800-27 Rev A
- Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity. NIST SP 800-160

Security Goals::Availability

- The security goal that generates the requirement for protection against intentional or accidental attempts to (1) perform unauthorized deletion of data or (2) otherwise cause a denial of service or data. NIST SP 800-27 Rev A
- Ensuring timely and reliable access to and use of information.
- Note: Mission/business resiliency objectives extend the concept of availability to refer to a point-in-time availability (i.e., the system, component, or device is usable when needed) and the continuity of availability (i.e., the system, component, or device remains usable for the duration of the time it is needed). NIST SP 800-160

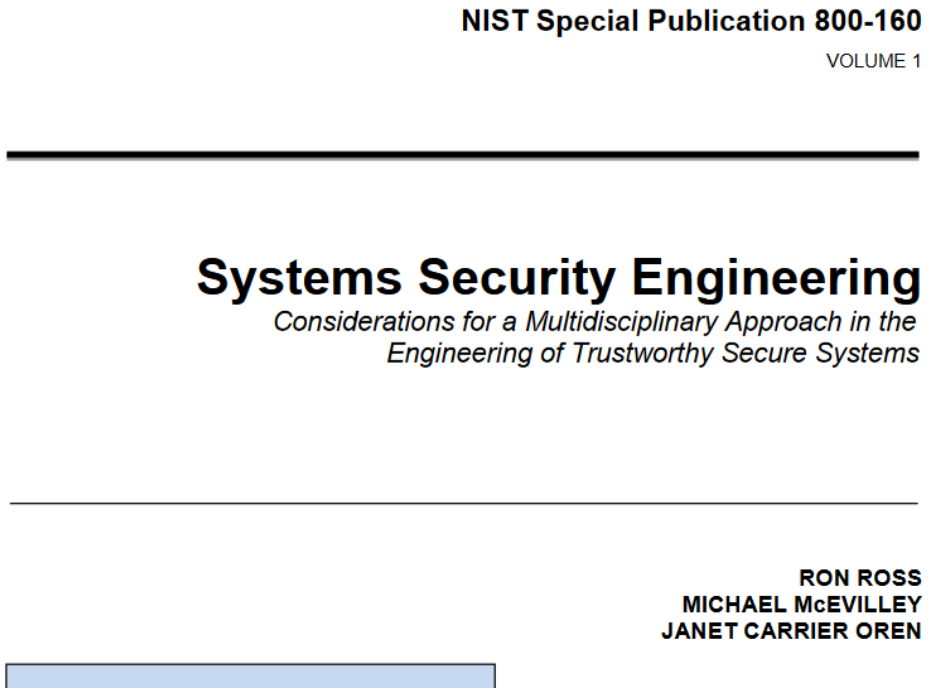
What about other goals/properties?

- Identified from the Threat Modelling or the Risk Analysis
- Non-repudiation
- Compliance
- Accountability

- All shall be captured during the Security Requirements Engineering phases.

Security Requirements Engineering

- Annex G of NIST SP 800-160



NIST Special Publication 800-160

VOLUME 1

Systems Security Engineering

Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems

RON ROSS
MICHAEL McEVILLEY
JANET CARRIER OREN

- https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160v1.pdf

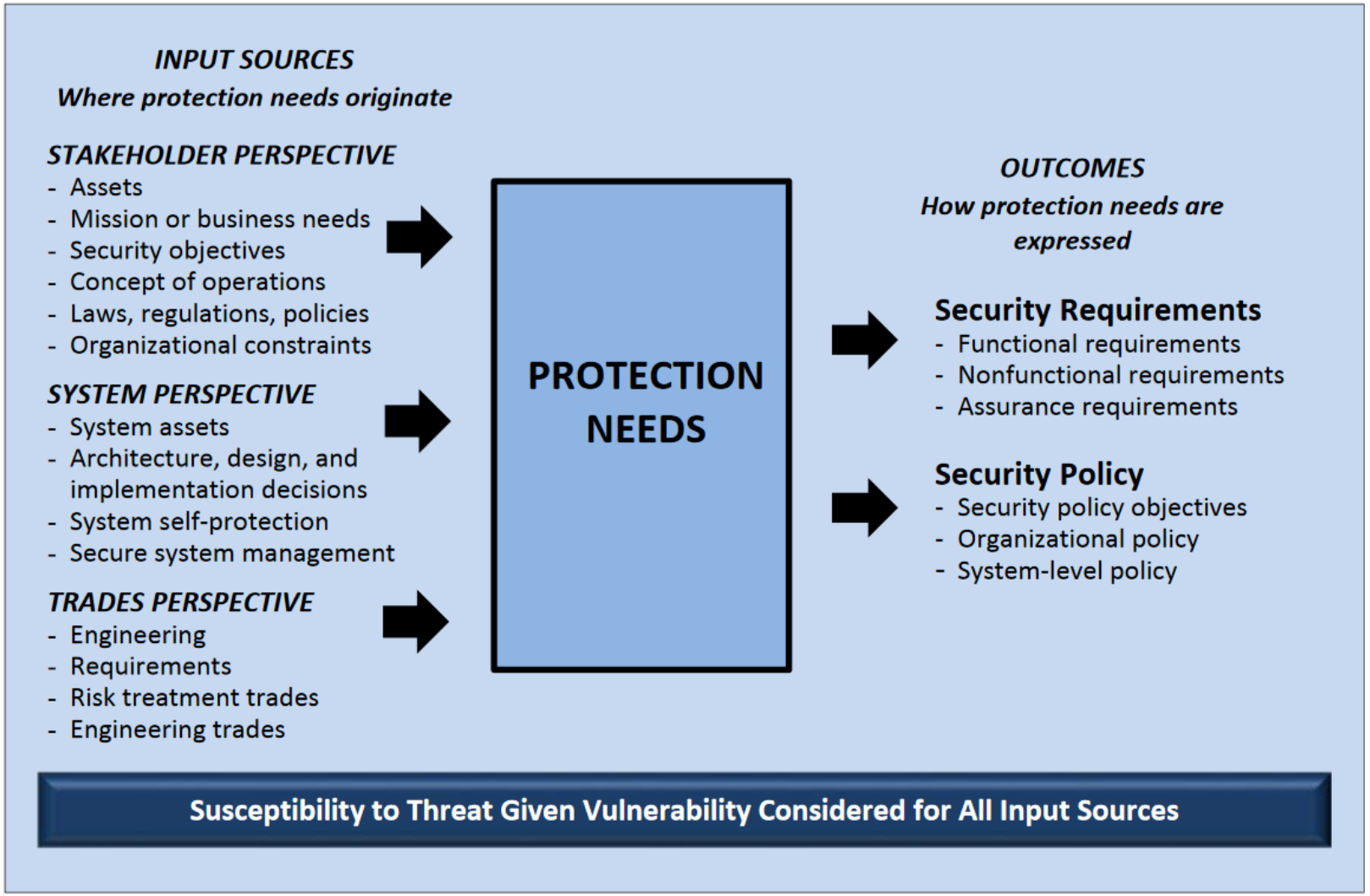
Security Requirements Engineering

- **Stakeholder perspective**: Mission/business needs; operational performance objectives and measures; life cycle concepts; laws; regulatory, statutory, and certification criteria; governing policies; loss and risk tolerance; accreditation, approval, and other independent authorization criteria; and all associated concerns and constraints.
- **System perspective**: System self-protection capability; system architecture, system design, and system implementation decisions; developmental, fabrication, manufacturing, and production standards; secure system management; technical performance objectives and measures; and all associated concerns and constraints.
- **Trades perspective**: Requirements trades; engineering trades; risk treatment trades; and other life cycle trades.

Security Requirements Engineering

- Transformation of Protection Needs into Security Requirements and Policy
  - Security requirements specify security capability, performance, effectiveness, and the associated verification and validation measures, as well as constraints on system requirements.
  - Security policy consists of a well-defined set of rules that govern all aspects of the security-relevant behavior of system elements.

Security Requirements Engineering



Security Requirements Engineering

- A requirement is a condition or capability that must be met or possessed by a system or system element to satisfy a contract, standard, specification, or other formally imposed document [IEEE 610.12]
- **Functional requirements** specify capability and behavior while **nonfunctional requirements** specify quality attributes of the capability and behavior.

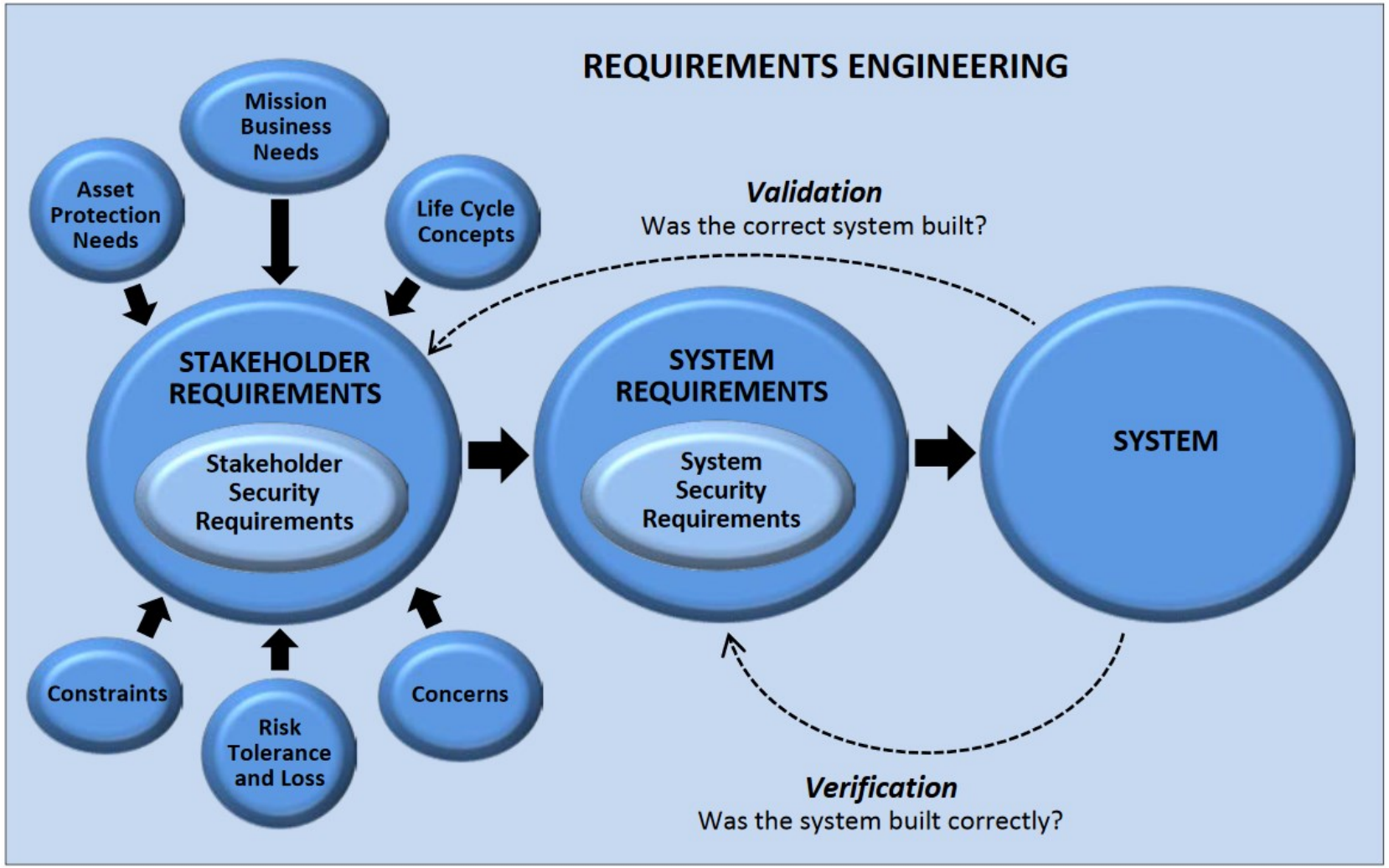
Security Requirements Engineering

- Stakeholder Requirements and Security Requirements
  - operational, protection, safety, and other needs (e.g. usability, human factors, form factor, and operational performance) and expectations of stakeholders, including legal, policy, regulatory, statutory, certification, policy, and other constraints for solutions that support the mission or business.
  - Provide the protection needed for the mission or business, the data, information, processes, functions, human, and system assets; the roles, responsibilities, and security-relevant actions of individuals that perform and support the mission or business processes; the interactions between the security-relevant solution elements; and the assurance that is to be obtained in the security solution.

Security Requirements Engineering

- System Requirements and System Security Requirements
  - (security) define the protection capabilities provided by the security solution; the performance and behavioral characteristics exhibited by the security solution; assurance processes, procedures, and techniques; and the evidence required to determine that the system security requirements have been satisfied.
  - specify the capability and quality attributes of the delivered system.

Security Requirements Engineering



Security Requirements Engineering

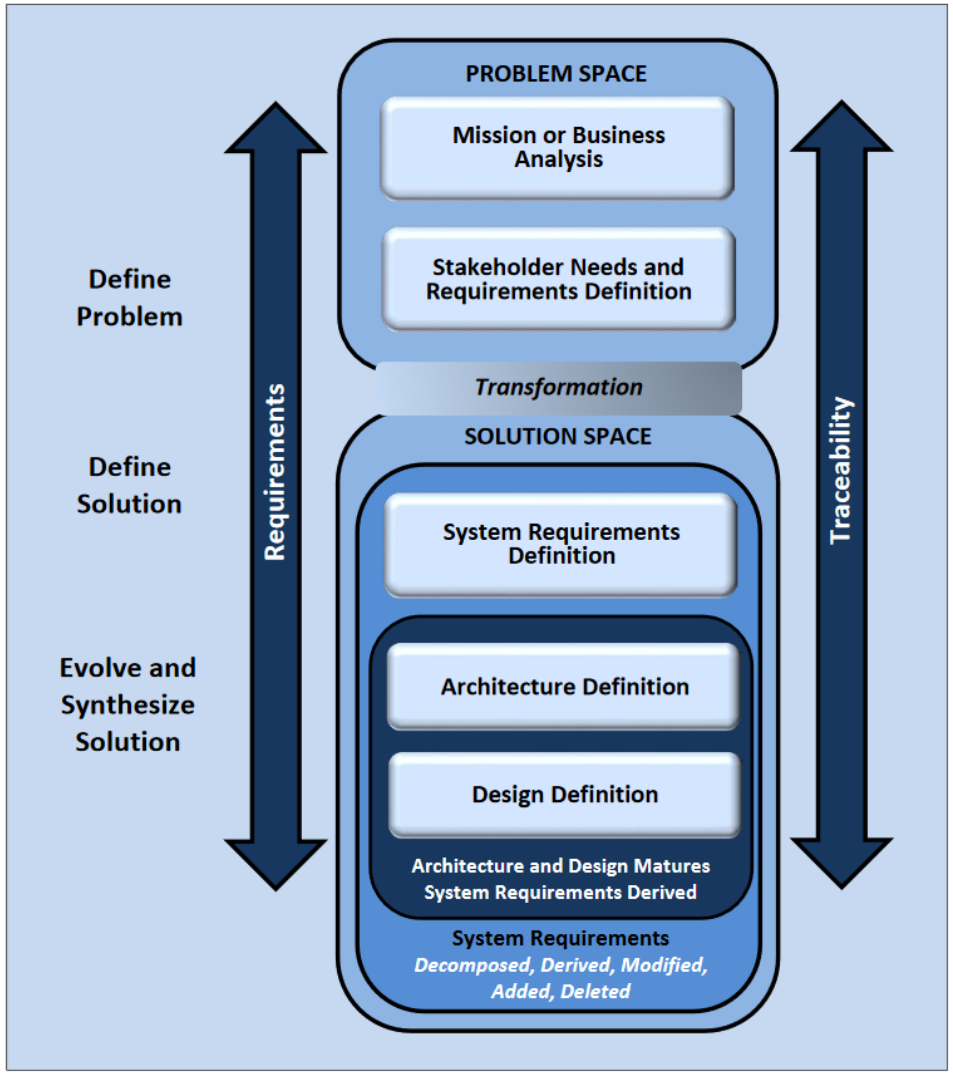
universidade de aveiro

Critical software

- Types of Security Requirements
  - **Security functional requirements** – the capability for the system to protect itself.
  - **Security nonfunctional requirements** – security behavior, performance, strength-of-function, and quality characteristics and attributes.
  - **Security assurance requirements** – techniques and methods to verify that the functional and nonfunctional requirements are met.

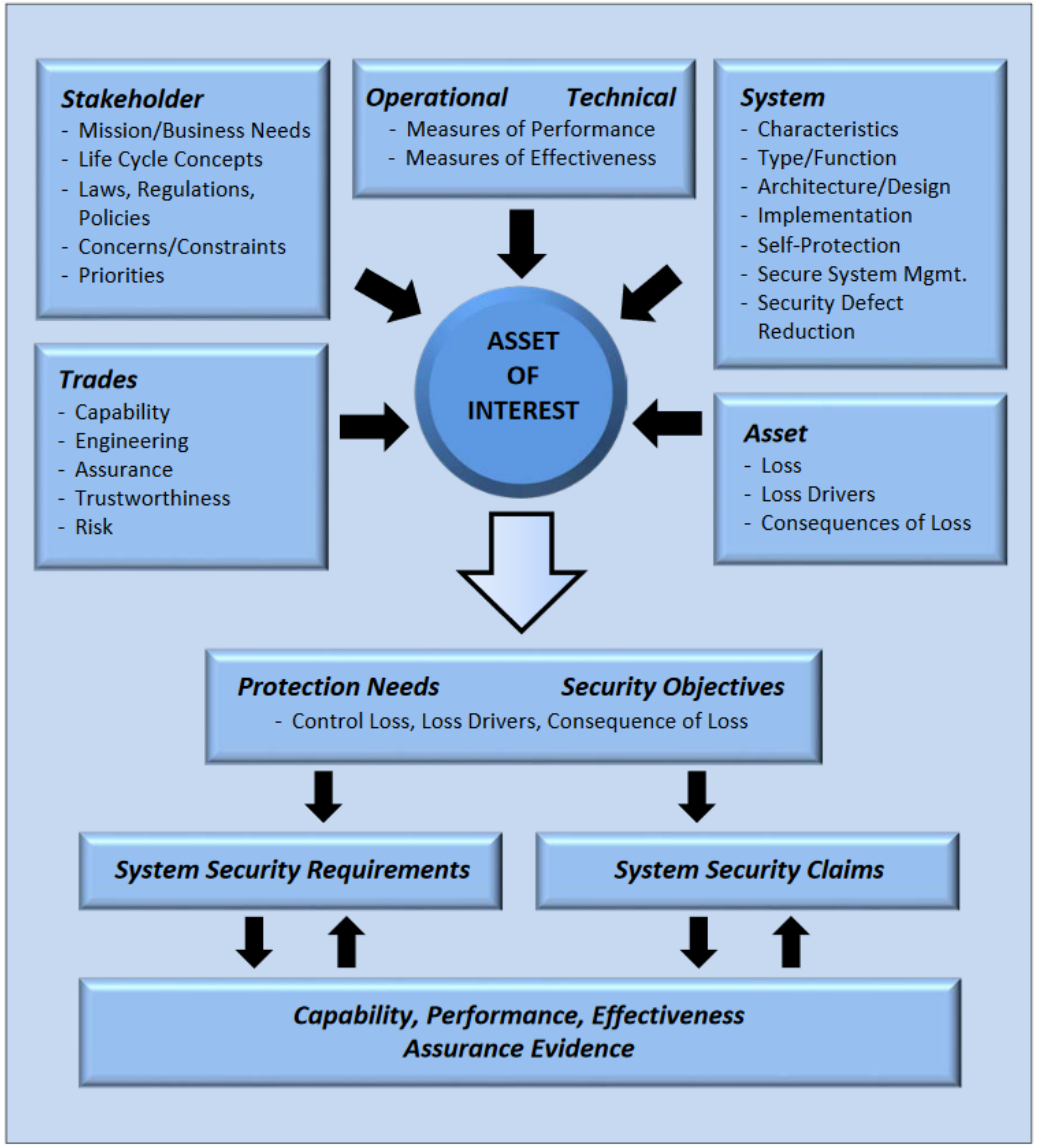
Security Requirements Engineering

- Security Requirements in the Life Cycle Processes
  - See also Security Lifecycle presentation



Security Requirements Engineering

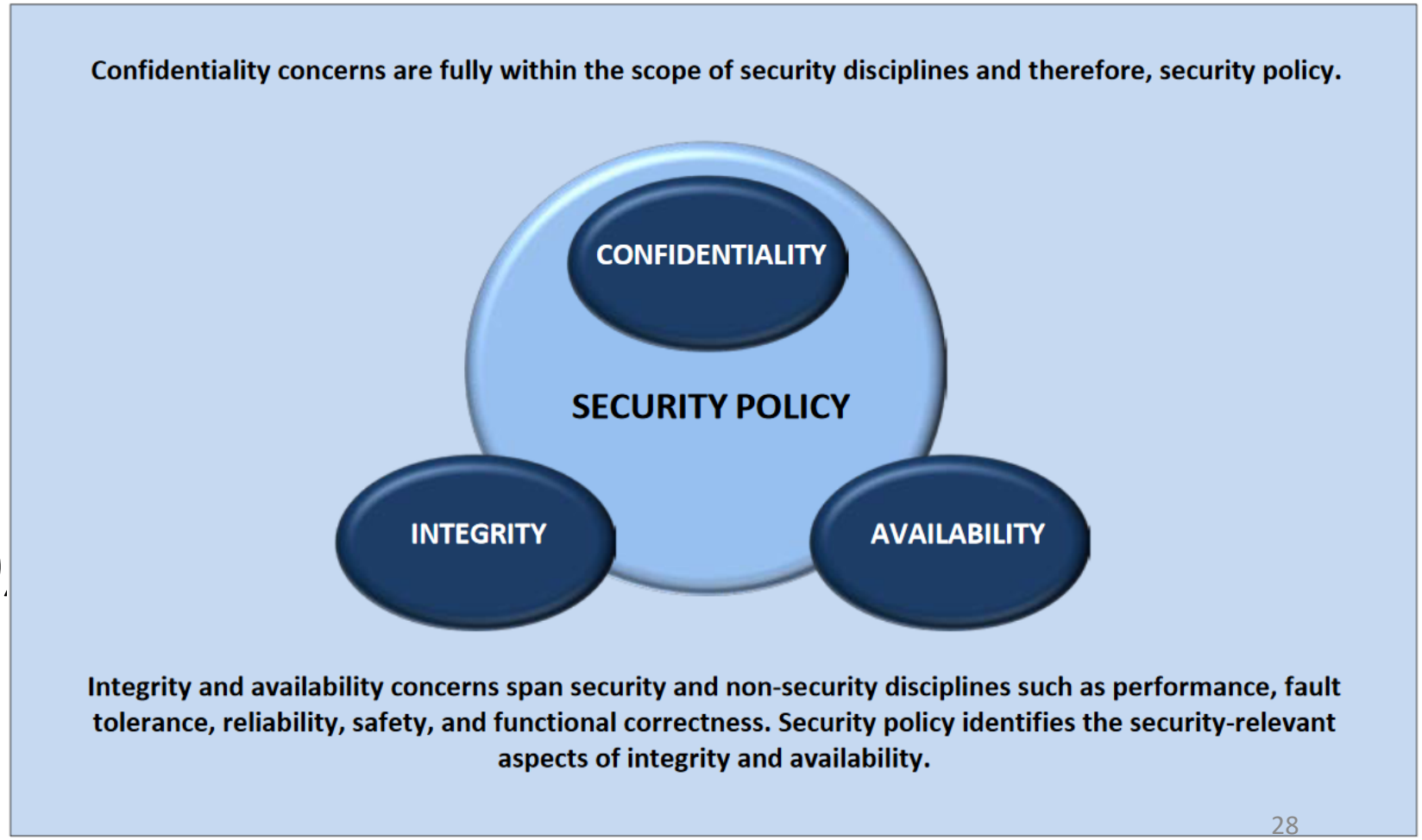
- Factors considered in security requirements analysis conducted as part of requirements engineering



Security Policy

- CIA Objectives

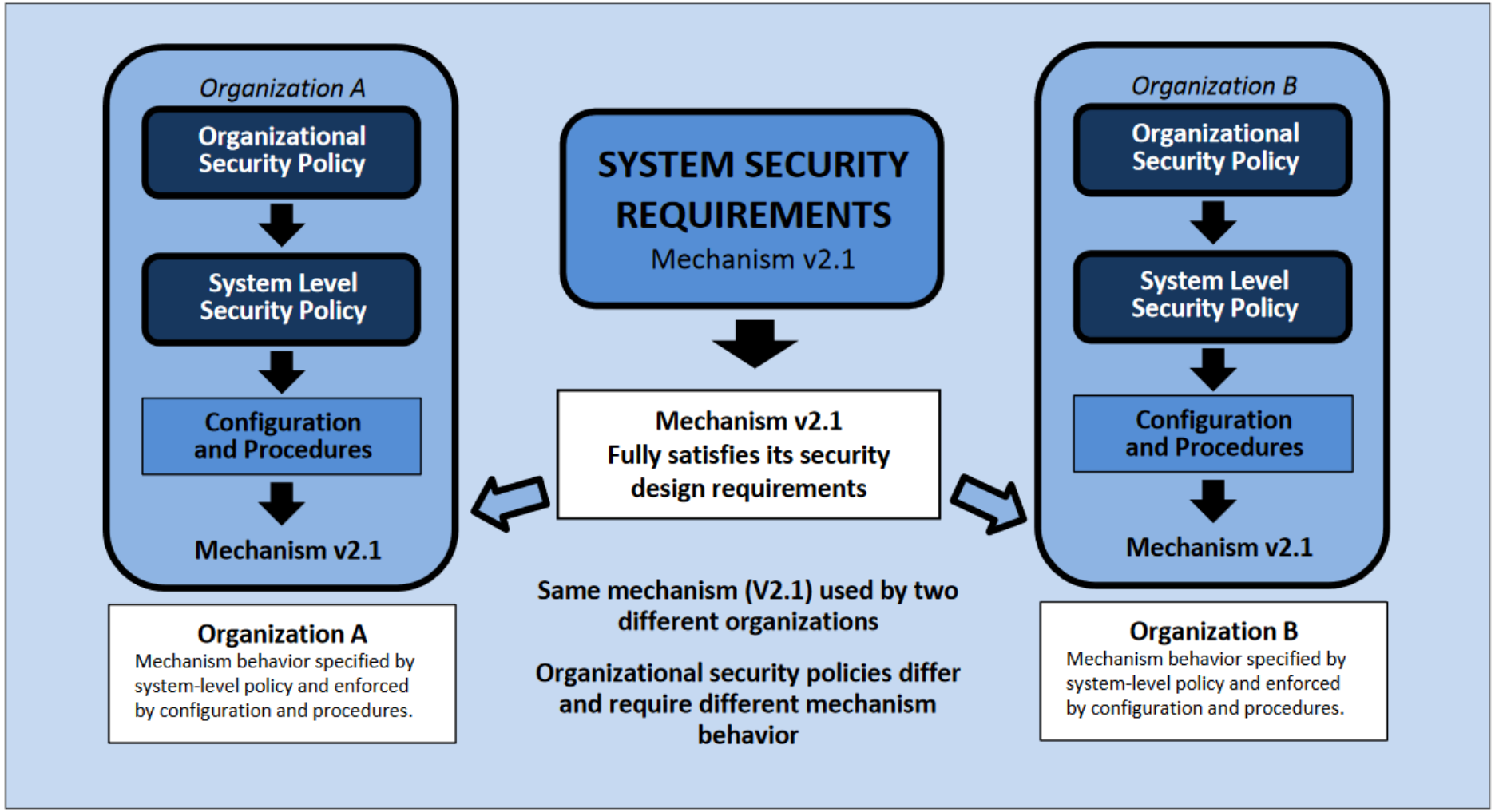
- See NIST SP 800-160, Annex G.3



Security Requirements and Policy Summary

- **Security requirements** determine the capability for security mechanisms to behave in some manner;
- **Security policy** determines the behavior that is deemed “secure” behavior; and
- For a **mechanism** to be deemed **secure**, the requirements for the capability of the mechanism must be **consistent** with the **security policy enforcement rules**; the mechanism must satisfy the security requirements; and the mechanism must be configured to behave in a manner defined by the **organizational security policy**.

Security Requirements and Policy Summary



References

- Cherdantseva Y. and Hilton J.: "Information Security and Information Assurance. The Discussion about the Meaning, Scope and Goals". In: *Organizational, Legal, and Technological Dimensions of Information System Administrator*. Almeida F., Portela, I. (eds.). IGI Global Publishing. (2013)
- *"Engineering Principles for Information Technology Security"*, *SP 800-27, Rev A, https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-27ra.pdf*
- Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems, SP 800-160, November 2016, NIST, https://doi.org/10.6028/NIST.SP.800-160v1
- https://en.wikipedia.org/wiki/Information_security

References

- Writing Good Requirements, Ivy Hooks, https://reqexperts.com/wp-content/uploads/2015/07/writing_good_requirements.htm
- NISTSP 800-160 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160v1.pdf

Exercise #2

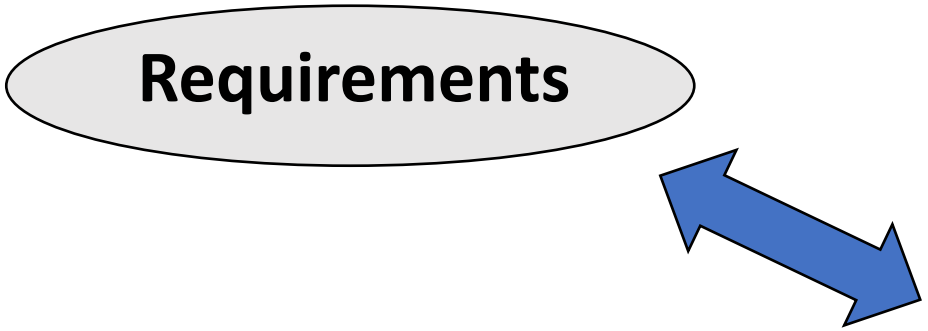
- Create a Security Requirements Catalogue
- Let’s write >15 requirements: At least 3 requirements related to Confidentiality, 3 to Integrity and 3 to Availability
- At least one requirement applicable to the user and one applicable to the development process/organization
- All you need to consider is: requirements applicable to your project!

- Size: 1 to 3 pages.

Optional Support Slides

Requirements engineering is the process of establishing

- the services that the customer requires from a system
- the constraints under which it operates and is developed



The descriptions of the system services and constraints

that are generated during the requirements engineering process

Examples (requirements iteration)

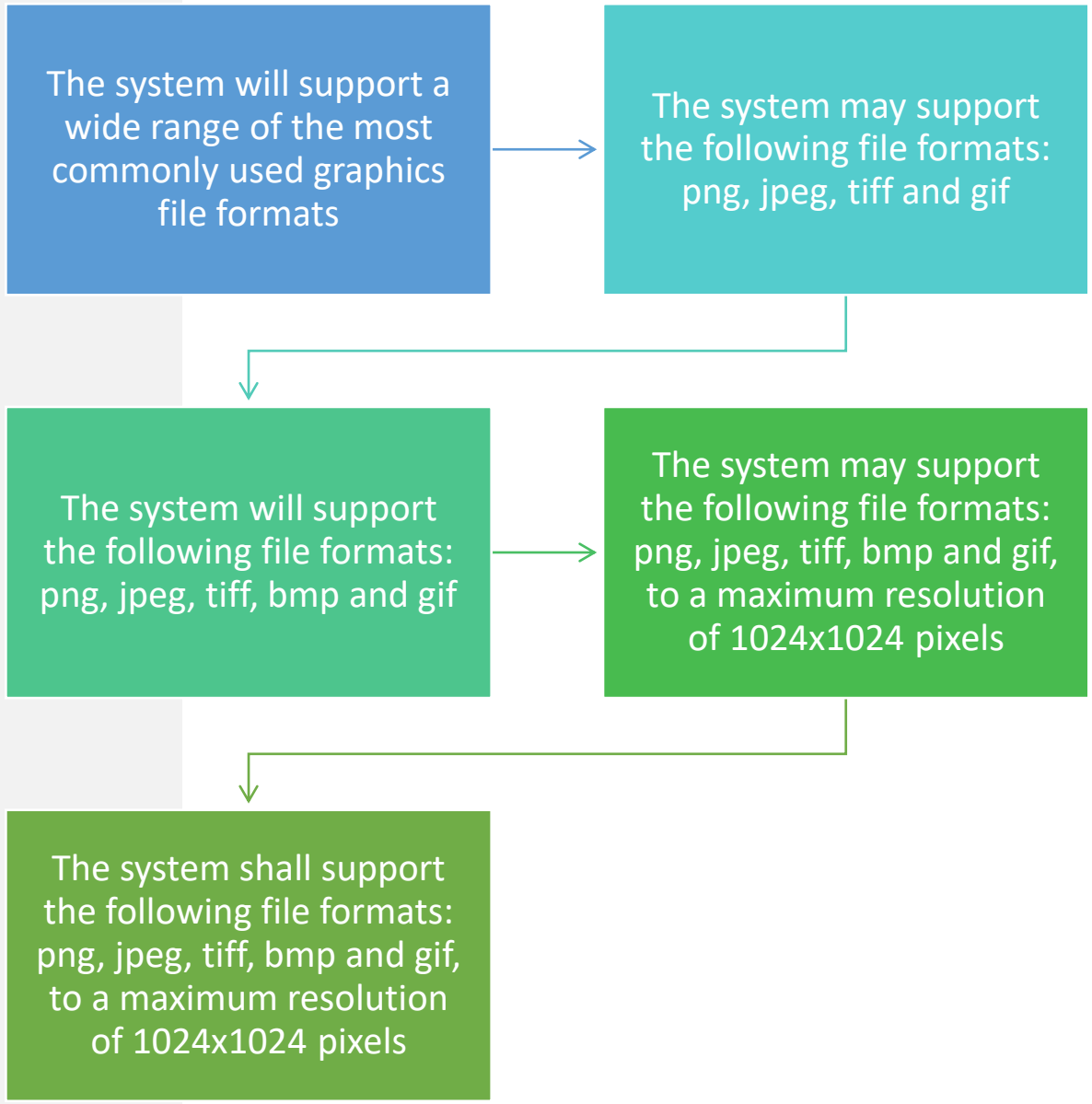
- The system will support a wide range of the most commonly used graphics file formats
- The system may support the following file formats: png, jpeg, tiff and giff
- The system will support the following file formats: png, jpeg, tiff and giff
- The system may support the following file formats: png, jpeg, tiff and giff, to a maximum resolution of 1024x1024 pixels
- The system **shall** support the following file formats: png, jpeg, tiff and giff, to a maximum resolution of 1024x1024 pixels

Examples (requirements iteration)

1. graphics file formats
2. tiff and giff

3. The system will support
4. The system may
5. The system shall

Examples (requirements iteration)



Examples of Functional Requirements

universidade de aveiro

Critical software

A. All users will access the system using a user id and a password
B. The system shall support the following document formats: PDF, RTF, Microsoft Word 2010 and ASCII text
C. Every transaction shall be allocated a unique identifier
D. The system shall have a mechanism to help recover a user's password

Which are security related? (A, C, D)

Requirements Precision

- Problems arise when requirements are not precisely stated (see 2 previous slides)
- Ambiguous requirements are interpreted in different ways by developers and users
- For example, '**recover password**' from the previous slide:
  - **User intention** – mechanism which allows the user to view the password after going through an authentication procedure
  - **Developer interpretation** – allowing the user to reset their password so that it can be set again (e.g. using email link)
- Thus, requirements shall be defined as precisely and clearly as possible

Requirements Completeness and Consistency

- Ideally, requirements should be both complete and consistent:

  Complete

  - They should include descriptions of all facilities required

  Consistent

  - There should be no conflicts or contradictions in the descriptions of the system facilities

- In reality, it is very difficult/impossible to produce a complete and consistent requirements document

Examples of Non-Functional Requirements

universidade de aveiro

Critical software

- Product requirement
  - All encryption shall use the Advanced Encryption Standard version x.
- Organisational requirement
  - The system development process and deliverable documents shall conform to the process and deliverables defined in coding and documentation standard XYZ.
- External requirement
  - The system shall not disclose any personal information about customers apart from their name and reference number to the operators of the system.
- Performance requirement
  - The system shall respond to a user's request for information in less than 1 second during "peak-time" and 0.1 seconds during "normal time".

Goals and Requirements

- **Non-functional requirements** may be very difficult to state precisely and imprecise requirements may be difficult to verify.
- **Verifiable non-functional requirement**
  - A statement using some measure that can be objectively tested
- **Goal**
  - A general intention of the user such as ease of use
- Goals are helpful to developers as they convey the *intentions* of the system users

Examples - Goals

- **An example of a bad system goal**
  - The system should be easy to use by experienced controllers and should be organised in such a way that user errors are minimised.
- **An example of verifiable non-functional requirement**
  - Controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by trained users shall not exceed two per day.

Requirements Measures

Property	Measure
Speed	Processed transactions/second User/Event response time Screen refresh time
Size	K Bytes Number of RAM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

Requirements Interaction

- **Conflicts between different non-functional requirements are common in complex systems**

- Username/Password mechanism should be easy for user to remember
- All passwords must be hard to guess and ideally require upper/lower case letters and special symbols to ensure high security

Which is the *most critical* requirement?

Natural Language Requirements

universidade de aveiro

Critical software

- Lack of clarity
  - Precision is difficult without making the document difficult to read
- Requirements confusion
  - Functional and non-functional requirements tend to be mixed-up in the same document
- Requirements amalgamation
  - Several different requirements may be expressed together
  - Requirements tend to be extensive and comprise multiple objectives
  - Leads to problems with testing/debugging

Requirements Best Practices

- Use standard format for all requirements (requirements template, requirements guidelines)
- Use language in a consistent way. For example, use

  shall for mandatory requirements (that must be supported),

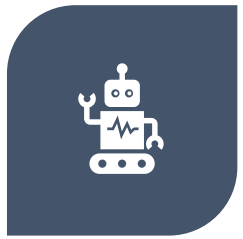
  should for desirable requirements (that are not essential).
- Use **text highlighting** to identify key parts of the requirement
- Avoid the use of computer jargon
- Make documents **self contained** (e.g. include glossaries and complete examples)

Requirements Best Practices

S – SPECIFIC

M – MEASURABLE

A – ACHIEVABLE / ATTAINABLE



R – RELEVANT

T – TIME-BOUND

Requirements Best Practices

Avoid:

- Always
- Some
- Appropriate
- Handle
- Etc.
- It
- Should
- Support
- But not limited to
- And/or
- Easy

Requirements Best Practices

Avoid:

- Making bad assumptions
- Writing implementation (HOW) instead of requirements (WHAT)
- Describing operations instead of writing requirements
- Using incorrect terms
- Using incorrect sentence structure or bad grammar
- Missing requirements
- Over-specifying

Requirements Best Practices

Instead, use:

- The System shall provide ...
- The System shall be capable of ...
- The System shall weigh ...
- The Subsystem #1 shall provide ...
- The Subsystem #2 shall interface with ...
- The Administrator shall …
- The User shall be able to …

Security Example

- **Security Quality Sub factors and Associated Measures**

Security Quality Sub factor	Associated Security Measures
Confidentiality	Access control; Physical protection; Security policy
Integrity	Access control; No repudiation; Physical protection; Attack detection
Availability	System recovery; Physical protection; Attack detection
Accountability	Non-repudiation; Attack detection

- Source: https://www.pmi.org/learning/library/importance-of-security-requirements-elicitation-9634

Security Example

- **Security Measures and Associated Mechanisms**

Security Measure	Associated Security Mechanisms
Access Control	Biometrics; Certificates; Multilevel security; Passwords and keys; Reference monitor; Registration; Time limits; User permissions; VPN
Security Policy	Administrative privileges; Malware detection; Multilevel security; Reference monitor; Secure channels; Security session; Single access point; Time limits; User permissions; VPN
Non-repudiation	Administrative privileges; Logging and auditing; Reference monitor
Physical Protection	Access cards; Alarms; Equipment tagging; Locks; Offsite storage; Secured rooms; Security personnel
System Recovery	Backup and restoration; Configuration management; Connection service agreement; Disaster recovery; Off-site storage; Redundancy
Attack detection	Administrative privileges; Alarms; Incident response; Intrusion detection systems; Logging and auditing; Malware detection; Reference monitor
Boundary Protection	DMZ (Demilitarized Zone); Firewalls; Proxies; Single access point; VPN

Security Example

- **Example of a Security Requirements Catalogue**

Type	Requirement Description	Comments
Authentication	The system shall have authentication measures at all the entry points, front panel, or inbound network connection.	To avoid unauthorized access
	The system shall support Windows domain authentication, and when authenticate to AD (Active Directory), shall support NTLMv2, as well as Kerberos protocol, and shall avoid transmitting username/password on the wire when authentication to the AD.	Currently many systems use Single Sign-On (SSO) using Kerberos and Windows domain.
	The system shall support Smartcard, USB token (two factors) authentication.	Improving the security using smartcard technologies and facilitating the physical access in case of using SSO.
	The system shall support Proximity card, magnetic card authentication.	To adequate some technologies as NFC (Near Field Communication), for example,
	The system shall support authentication based on device local authority.	In case of physical access.
	The system shall support cloud-based authentication.	Allow cross domain Single Sign On
	The system shall support authentication to the external 3rd party authentication server, and protect the user credential during authenticating to the external server.	Security network aspects must receive special attention.
	The system shall support multiple authentication approaches at the same time.	It is necessary a strong monitoring and auditing

Security Example

- **Example of a Security Requirements Catalogue**

Availability	The backup system shall store the recover data in a network system.	To help in case of failure or intruder action.
	The system shall do mirroring to allow data to be available in physically separated locals (separate site when the application is on the Web).	Help minimize the risk of one single point of failure.
	The system shall apply load balancing.	Help minimizing the impact of potential system failures.
Integrity	The system shall ensure all data provided by application has consistency (either create a new and valid state of data, or, return all data to its state before a transaction was started).	To avoid an unauthorized person or system alter data inadvertently or intentionally.
Auditing	The system shall keep historical records (logging) of events and processes executed in or by an application.	Define more specific security loggings to allow recreating a clear picture of security events.
Non-Repudiation	The system shall implement cryptographic methods such as generating digital signatures or digital fingerprinting.'	Help the application and system avoiding repudiation.