NON-FUNCTIONAL REQUIREMENTS

NON-FUNCTIONAL REQUIREMENTS

- System properties and constraints
  - e.g. reliability, response time and storage requirements
- Non-functional requirements may be more critical than functional requirements
  - if not met, system is pointless

NFR & ARCHITECTURE

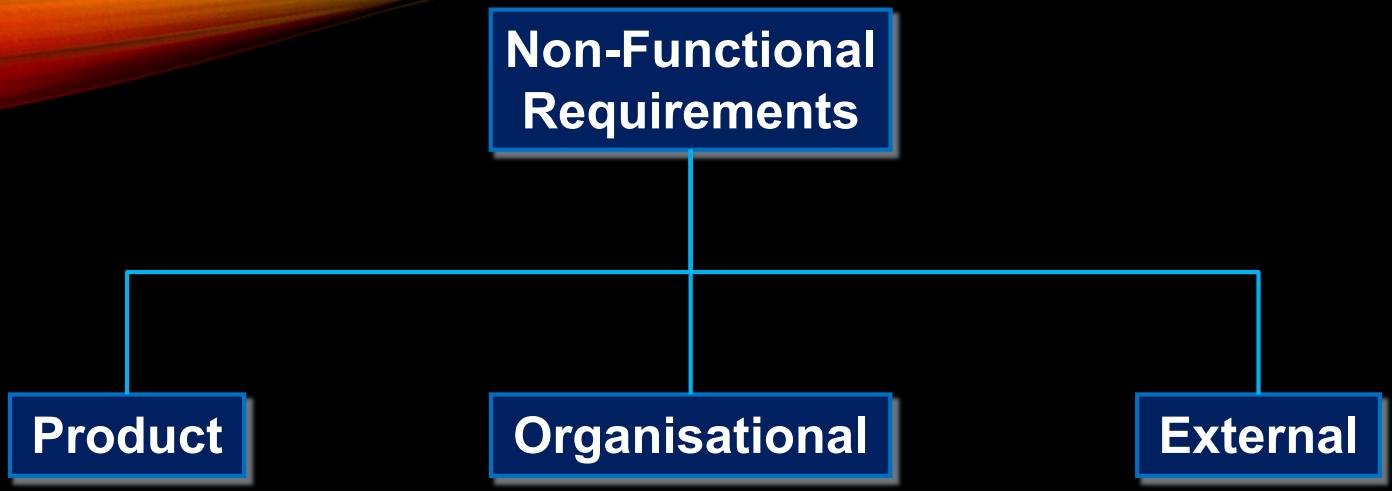
- Non-functional requirements may affect overall architecture
  - rather than just individual components
- e.g. performance requirements may require a design that minimises communication between components

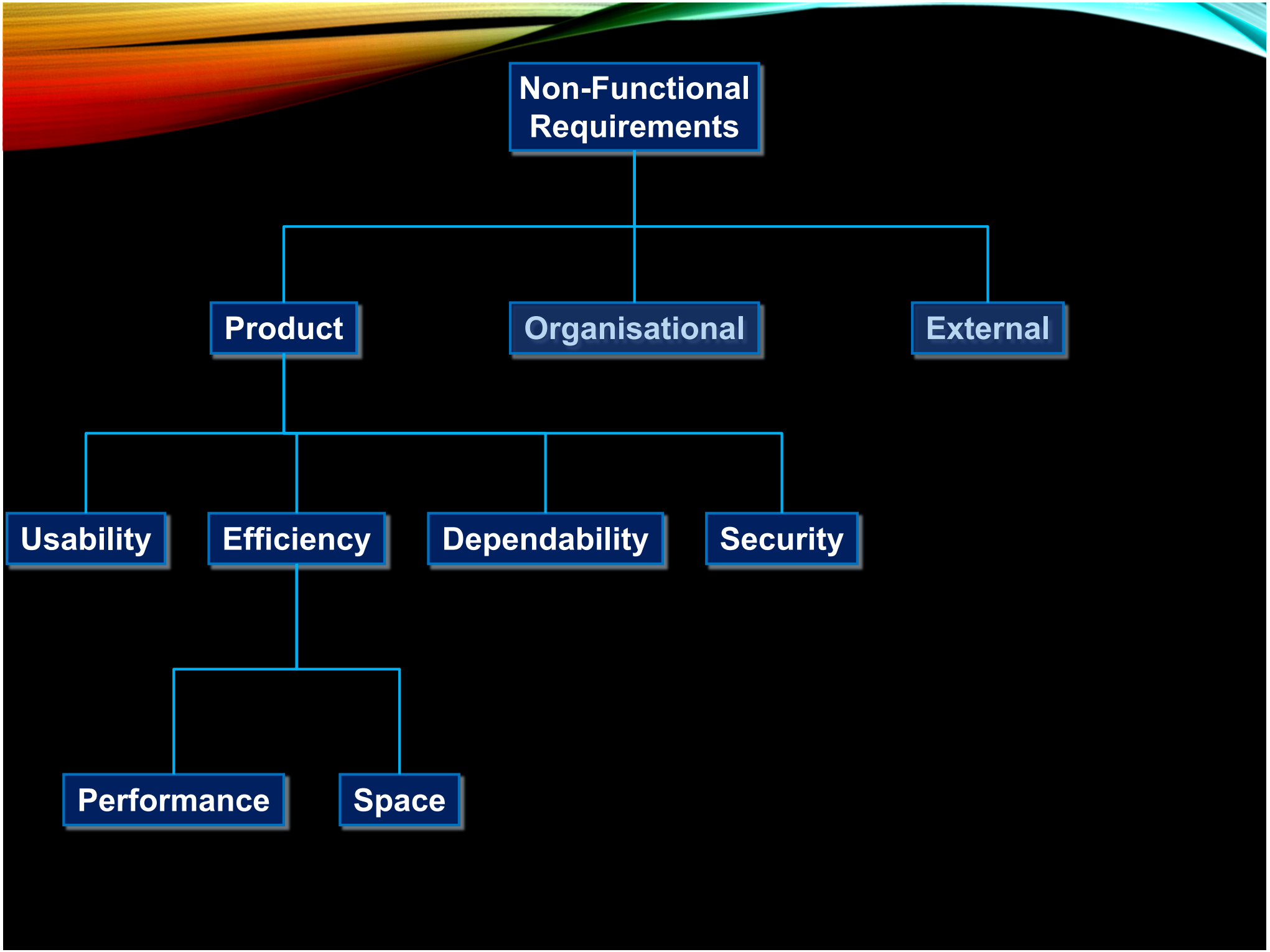
NON & FUNCTIONAL REQUIREMENTS

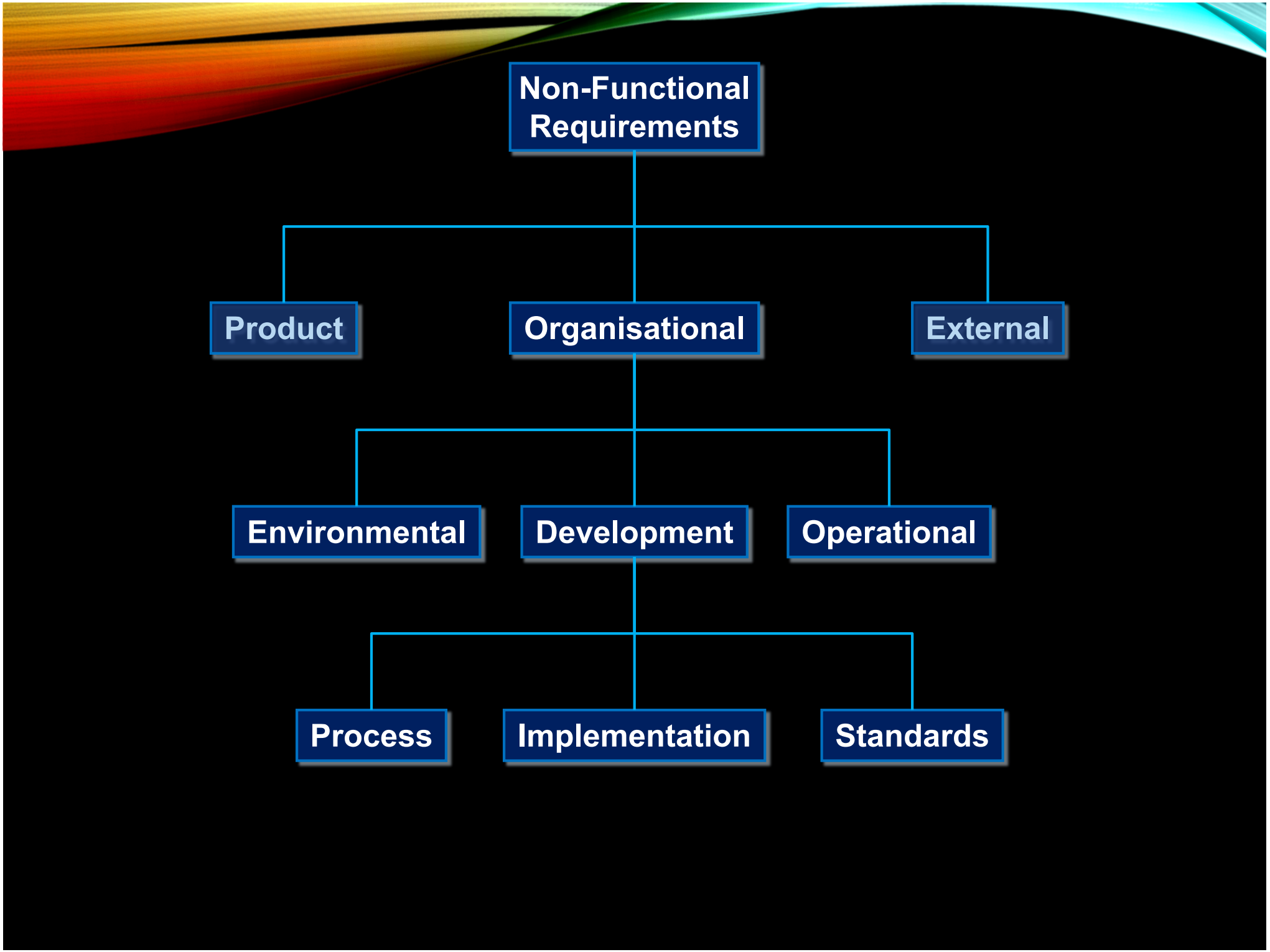
- A single non-functional requirement may:
  - generate a number of related functional requirements that define required system services
    - e.g. a security requirement
  - restrict functionality of existing functional requirements

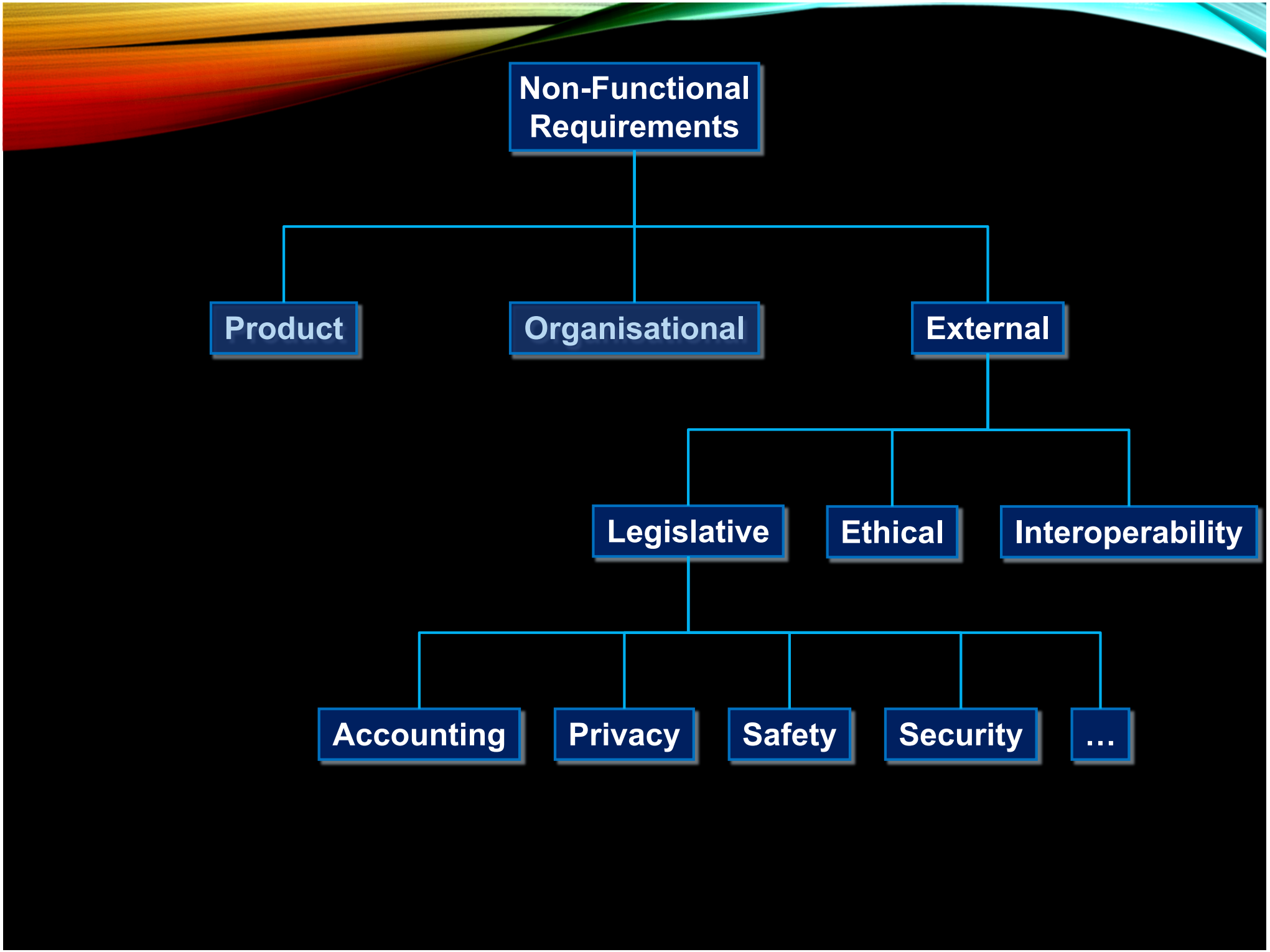
NFR SOURCES

- Product requirements
  - behavioural constraints
    - e.g. execution speed, reliability, …
- Process requirements
  - restrictions on the development process
    - e.g. standards to follow, …
- External requirements
  - factors external to the system
    - e.g. inter-operability, legislative requirements, …


Non-Functional Requirements
Product
Organisational
External


Non-Functional Requirements
Product
Organisational
External
Usability
Efficiency
Dependability
Security
Performance
Space


Non-Functional Requirements
Product
Organisational
External
Environmental
Development
Operational
Process
Implementation
Standards


Non-Functional Requirements
Product
Organisational
External
Legislative
Ethical
Interoperability
Accounting
Privacy
Safety
Security
…

NFR EXAMPLES

- Product requirement
  - The user interface for LIBSYS shall be implemented in HTML 5 and JavaScript, without any third-party libraries.
- Process requirement
  - System documentation shall contain all content specified in ISO/IEC/IEEE 15289. This content is to be organised in a searchable data repository.
- 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.

VERIFIABLE

- Imprecise requirements cannot be verified
- NFR should be a measurable statement
  - The system should be easy to use by experienced controllers and should be organised in such a way that user errors are minimised.

  vs.

  - Experienced controllers shall be able to use all system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

MEASURES

Property	Metric
Speed	Processed transactions/second, User/Event response time, Screen refresh time
Size	M Bytes
Ease of Use	Training Time, Number of help screens
Reliability	Mean time to failure, Prob. of unavailability, Rate of failure, Availability
Robustness	Time to restart after failure, % of events causing failure, Prob. of data corruption on failure
Portability	% of target dependent statements, number of target systems

REQUIREMENTS CONFLICTS

- Conflicts between different requirements are common in complex systems
  - efficiency vs. safety
  - usability vs. security

QUALITY ATTRIBUTES

Safety	Understandability	Portability
Security	Testability	Usability
Reliability	Adaptability	Reusability
Resilience	Modularity	Efficiency
Robustness	Complexity	Learnability

READING

- Sommerville
  - Chapters 10 to 14
  - Skim, don't need to read in detail

NEXT STEPS

- Tutorial
  - Use case modelling