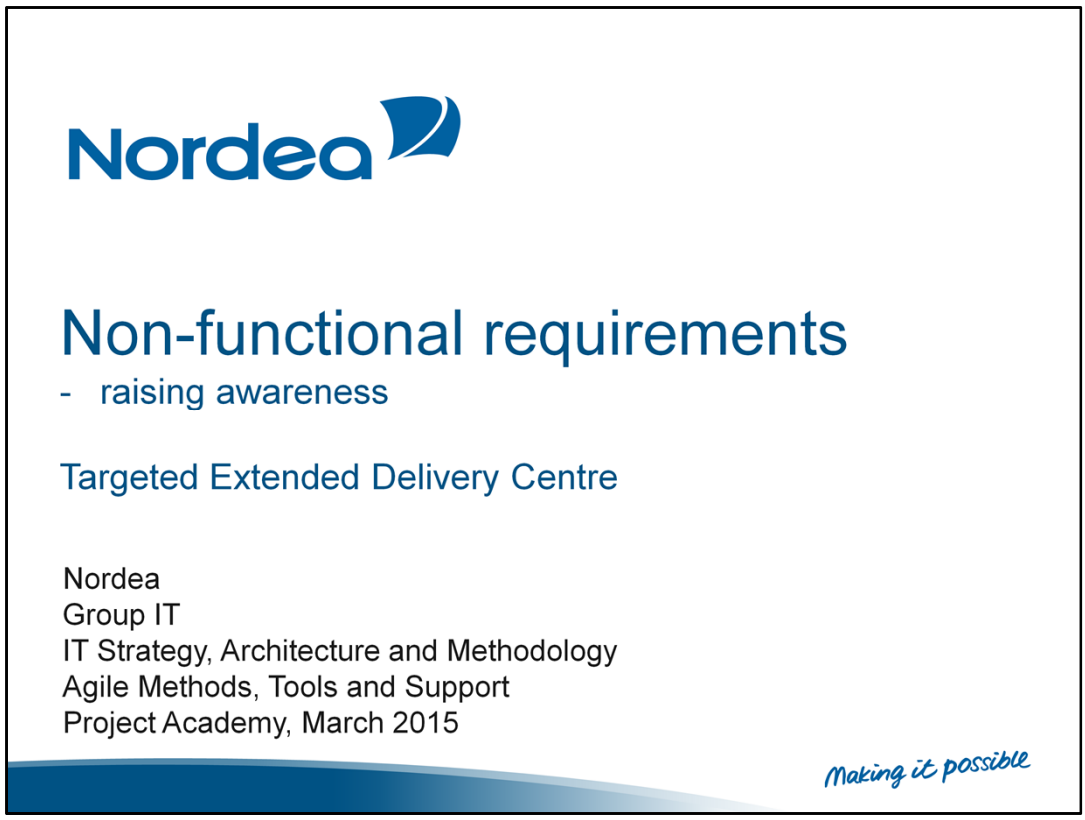

Nordea
Non-functional requirements
- raising awareness
Targeted Extended Delivery Centre
Nordea
Group IT
IT Strategy, Architecture and Methodology
Agile Methods, Tools and Support
Project Academy, March 2015
Making it possible

Comments regarding the training material to EDC

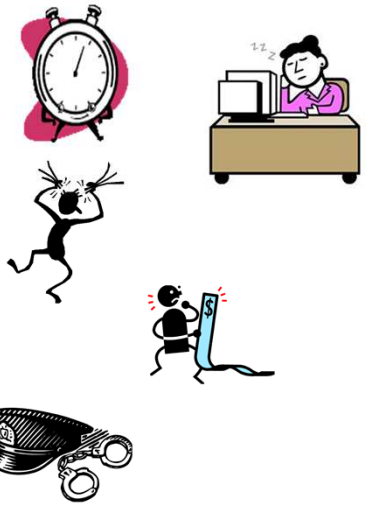
- This material is a “self-study-package” especially targeted to the EDC consultants in India
- The aim is primarily to raise the awareness of the importance of non-functional requirements
- Handing out slides from the original presentation without any kind of speech or comments from one of the instructors is *not* our recommended way of working
- Thus to fulfil the need of a “self-study-package” to EDC, each slide is accompanied with notes to explain the main points

Nordea

2

Why spend time on non-functional requirements?

- Insufficient non-functional requirements may lead to a system that works according to the functional requirements but there is a risk of:
  - Poor response time which makes the system in-efficient to use
  - Loss of data which makes it difficult to trust the result delivered by the system
  - In-efficient user interface that makes the system difficult to use
  - Increased costs due to less availability than expected
  - Serious implications due to not fulfilling legal regulations



Nordea

3

There is a lot of focus on non-functional requirements at the moment.

Non-functional requirements are important because they may drive a system's foundations - in other words, its architecture. Because these requirements are system-wide, they are by definition extremely important. In addition, they can be more significant than the functional requirements specified in use cases.

Gathering non-functional requirements is hard to do, and yet it is really important!

By improving the non-functional requirements we aim at

- Saving money
  - System architecture is designed from the beginning to meet the requirements

- Increasing quality and customer satisfaction
  - Avoid unstable applications in our production environments – and thus avoid unhappy customers and bad publicity
  - Detailed non-functional requirements are a way of making quality measurable

Nordea

4

We might just as well be honest and admit the reasons (mentioned on the slide) for focusing more on non-functional requirements than earlier.

Non-functional requirements in an agile project

- Working with non-functional requirements are equally important no matter what approach you have to the development work!
- In an agile project non-functional requirements have to be defined in the very beginning of a project to secure the right architecture

Defining non-functional requirements should be done very early in the lifecycle of an agile project!

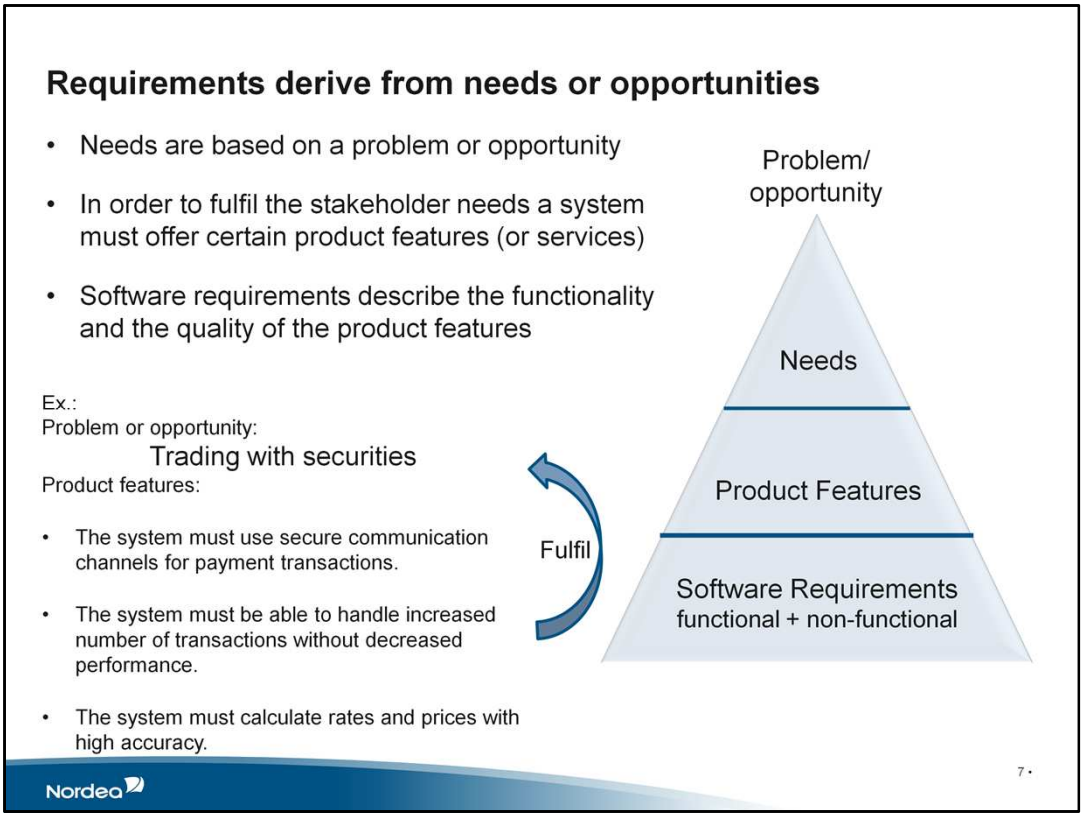
Nordea

5

You might be thinking: “We are working agile, so this is not relevant for us”. But you are so wrong!
Working with non-functional requirements are equally important no matter what approach you have to the development work!
It will be very expensive if you have to change the architecture later on during the project lifecycle.

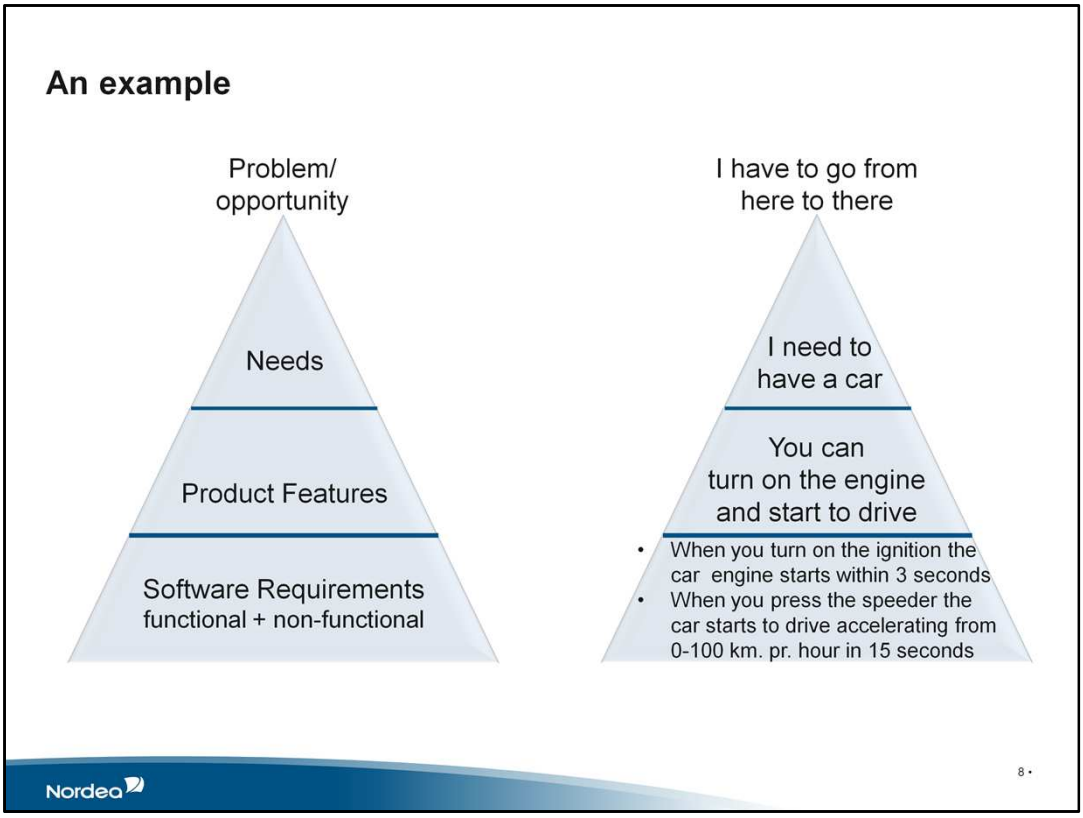
Nordea

Requirements derives from needs or opportunities!



It is important to keep in mind that all system requirements – functional as well as non-functional – derive from some kind of stakeholder's need. And the needs are caused by a problem or a business opportunity. No requirement should be "invented" – they are all based on stakeholder needs.

The example
Trading with securities is a need and the product features listed are offered to fulfil the need.
The product features are not yet specific enough to be software requirements.

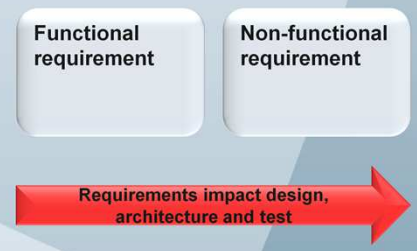


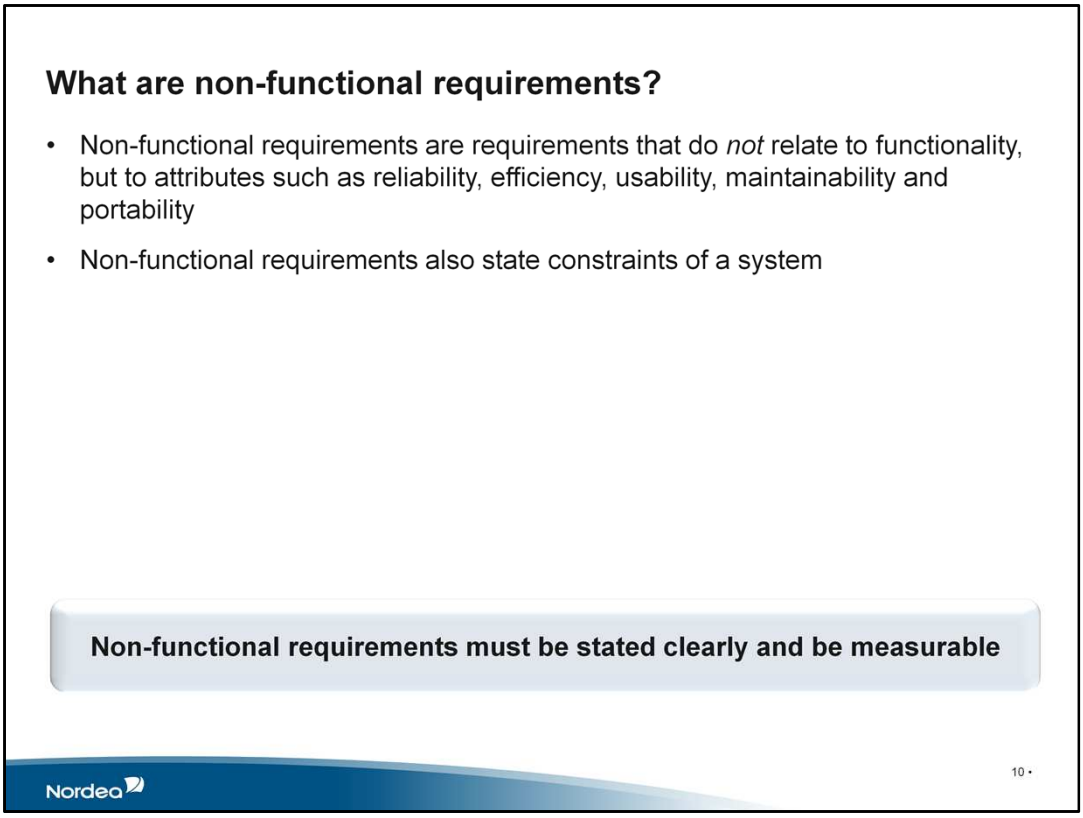
The pyramid to the left shows the ”model” – the pyramid to the right shows an example.

Nordea

What are non-functional requirements?

Functional versus non-functional requirements



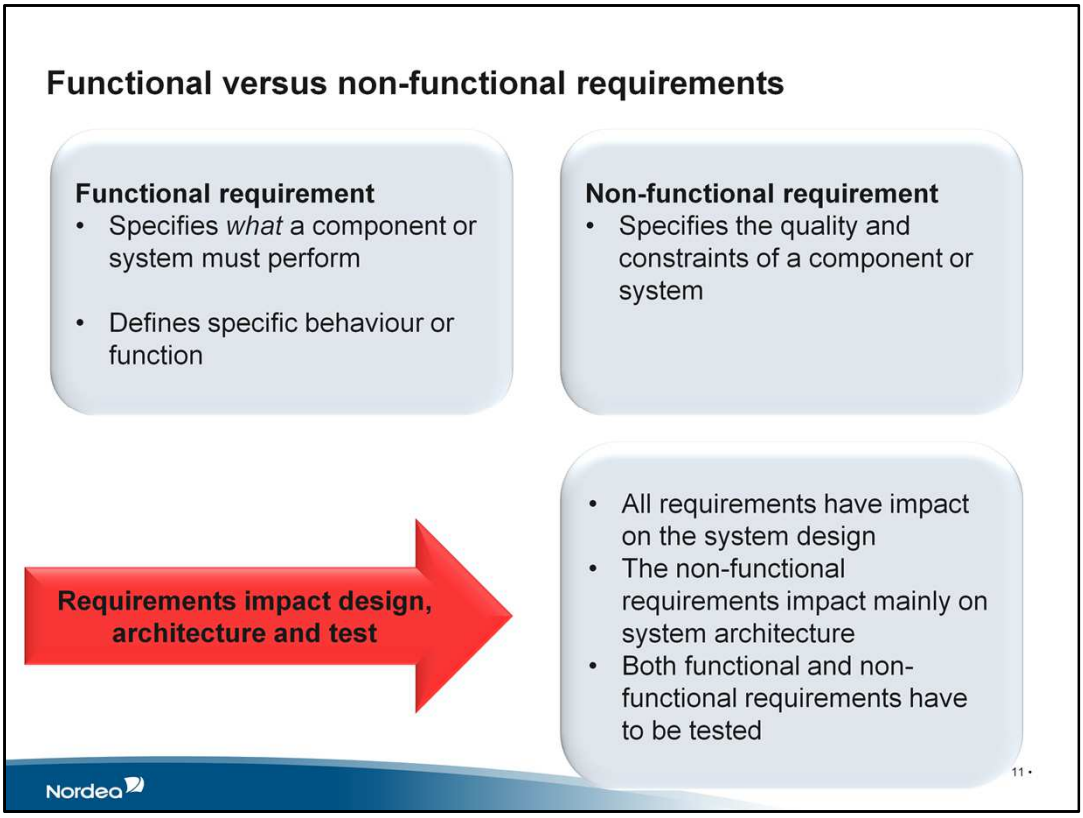


Constraints can also be seen as limitations.
An example of a constraint: A certain development process must be followed.

Sometimes you hear other terms for non-functional requirements:
- Supplementary requirements
- Quality requirements or
- Quality requirements and constraints

In Nordea the term non-functional requirements covers **quality requirements and constraints**.

Just like functional requirements non-functional requirements must be stated clearly and be measurable!



In this and the following slides we will see some examples that hopefully will help you understand the difference between functional requirements and non-functional requirements.

Depending of the non-functional requirements certain hardware might be required to be able to handle the desired performance.

The non-functional requirements have impact on design, architecture and test.
Testing non-functional requirements requires more than testing functional requirements – skills, knowledge, competences, tools, etc.

Functional versus non-functional requirements

Functional requirement
- When you turn on the ignition the car engine starts
- When you press the speeder the car starts to drive

Non-functional requirement
- When you turn on the ignition the car engine starts within 3 seconds
- When you press the speeder the car starts to drive accelerating from 0-100 km. pr. hour in 15 seconds
- The car must be able to seat 4 persons
- The car must be able to drive off road

Car # 1

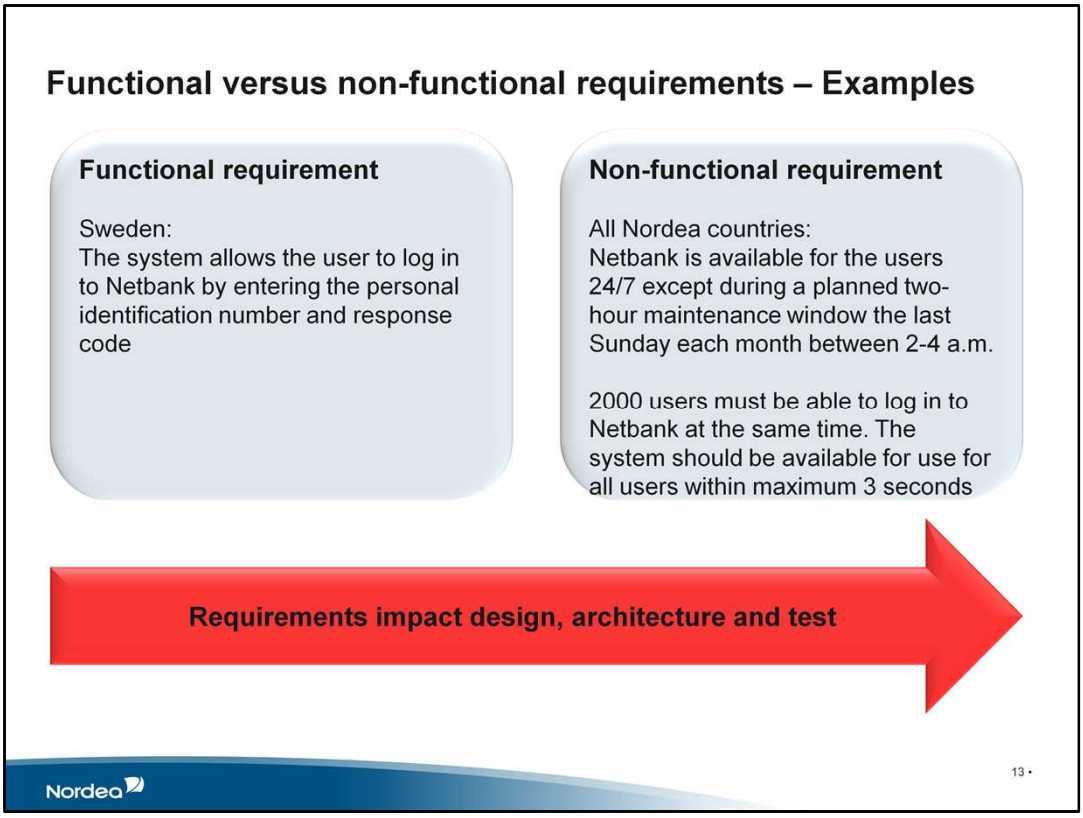
Car # 2

Car # 3

Nordea

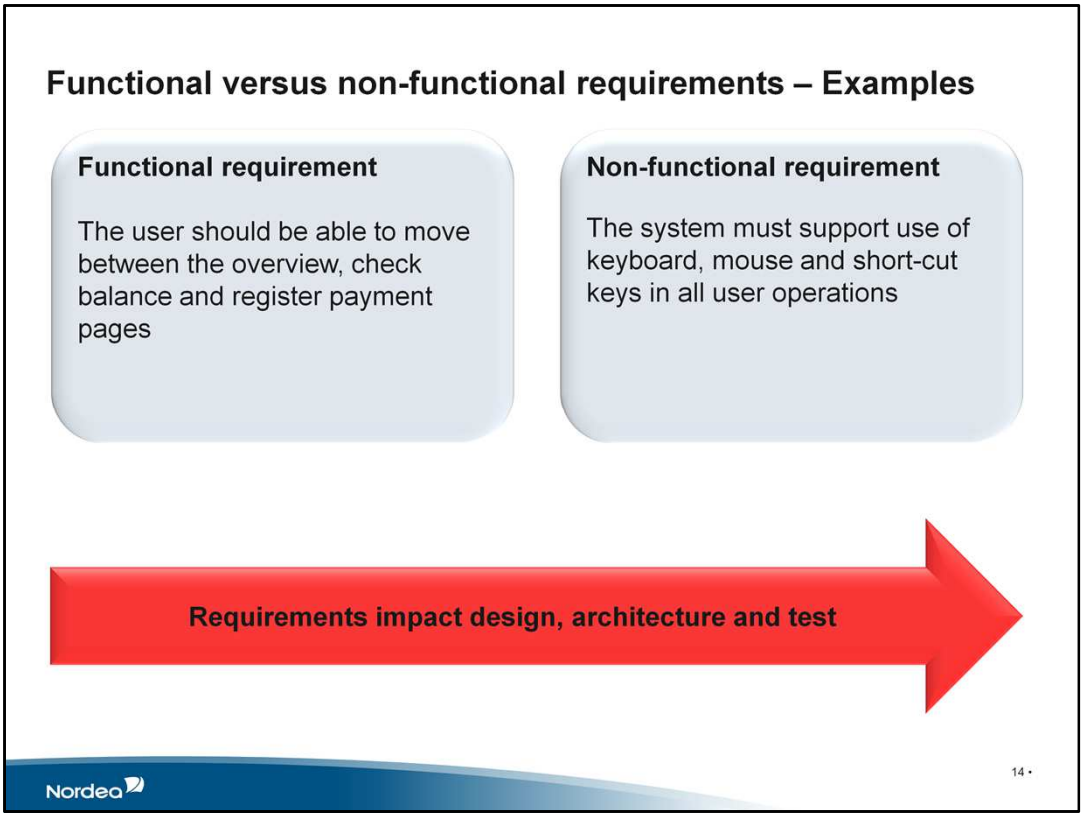
12 •

All the cars can fulfil the functional requirements.
Only Car # 3 can fulfil all the requirements.



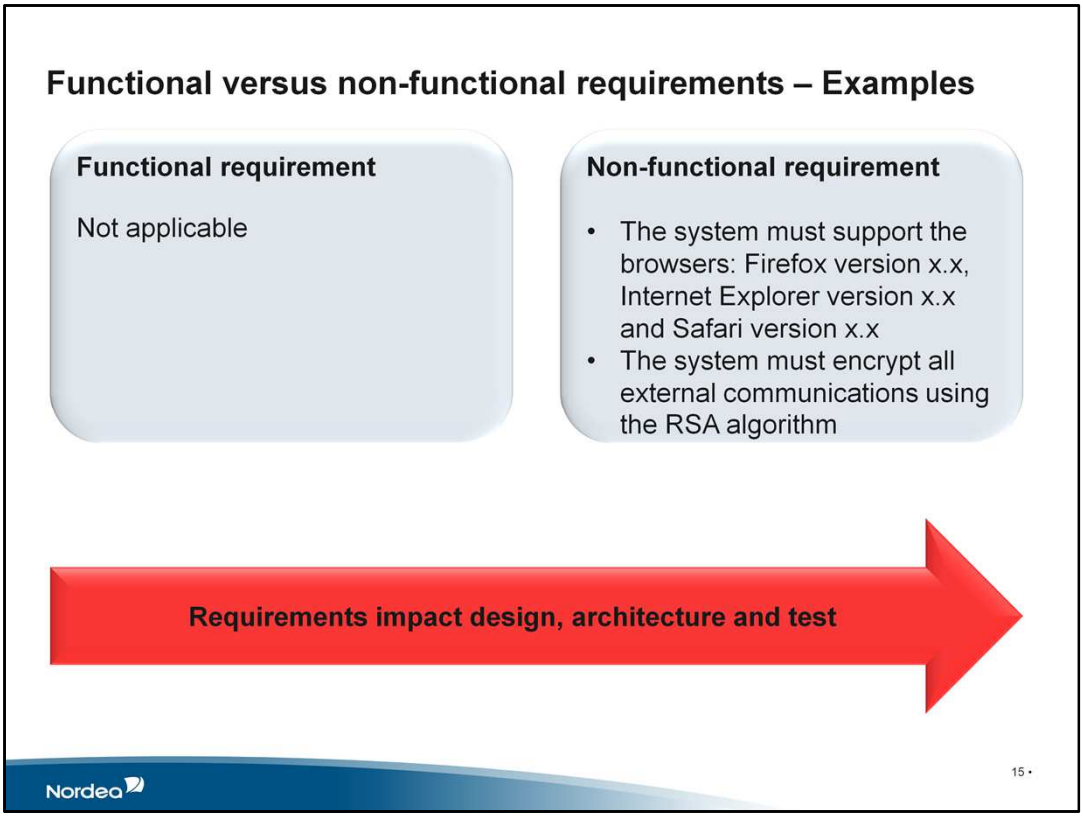
In Nordea we might have extra complexity due to different solution in some countries depending on the local laws in the different countries. Here is an example of different log-on procedures.

“Available 24/7” is a **reliability** requirement.
“2000 users” is a **performance** requirement.



The functional requirements state that the user must be able to move around within the different parts of the system.
The non-functional requirements state the different possibilities for doing so.

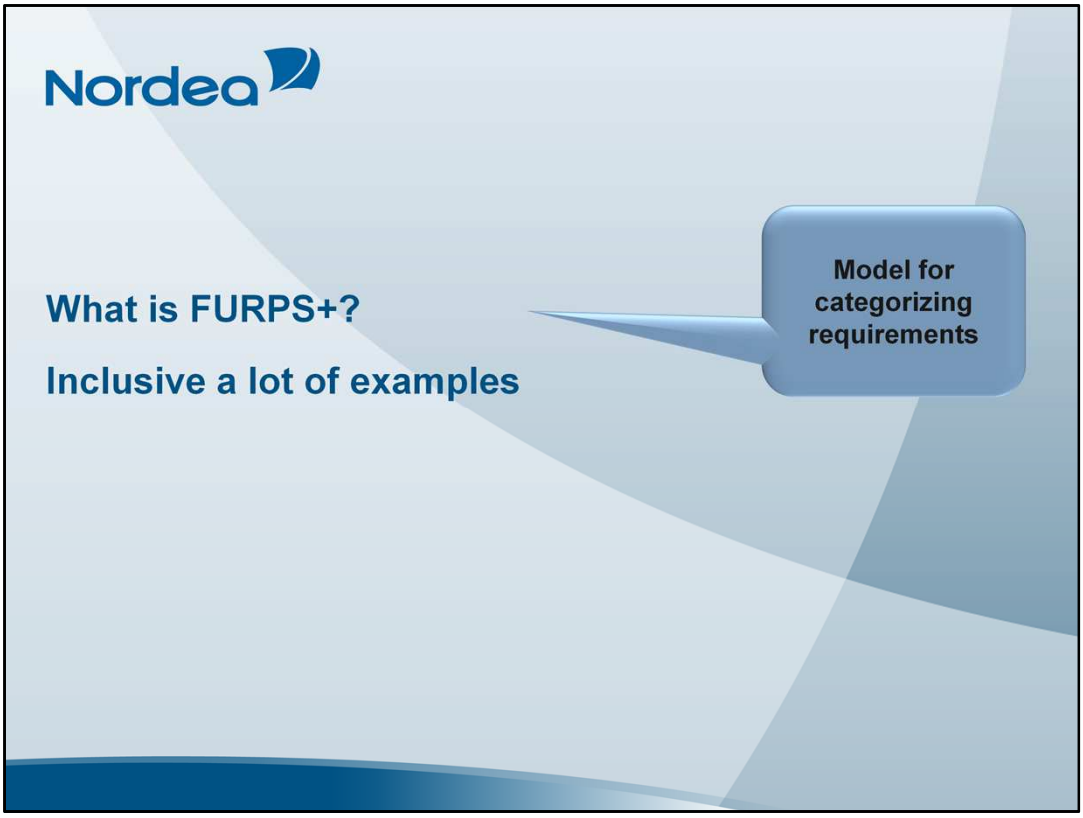
The non-functional requirement is a **usability** requirement.

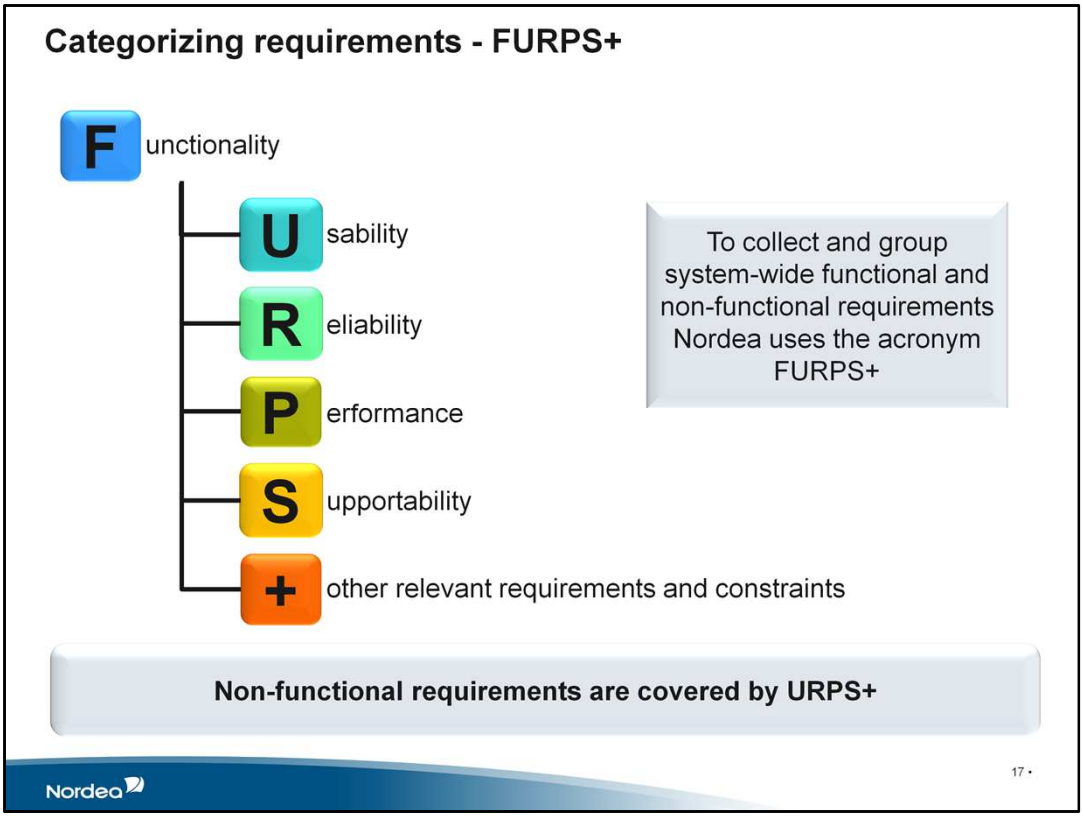


These non-functional requirements are examples of **constraints**.

(RSA is an algorithm for public-key cryptography).

Note: There might also be requirements that are neither functional nor non-functional system requirements but are still very important requirements for the project to fulfil – even though they are not software requirements.


Nordea
What is FURPS+?
Inclusive a lot of examples
Model for categorizing requirements



In Nordea FURPS+ is the chosen model for categorizing requirements.

FURPS+ are used for categorizing all kinds of requirements.

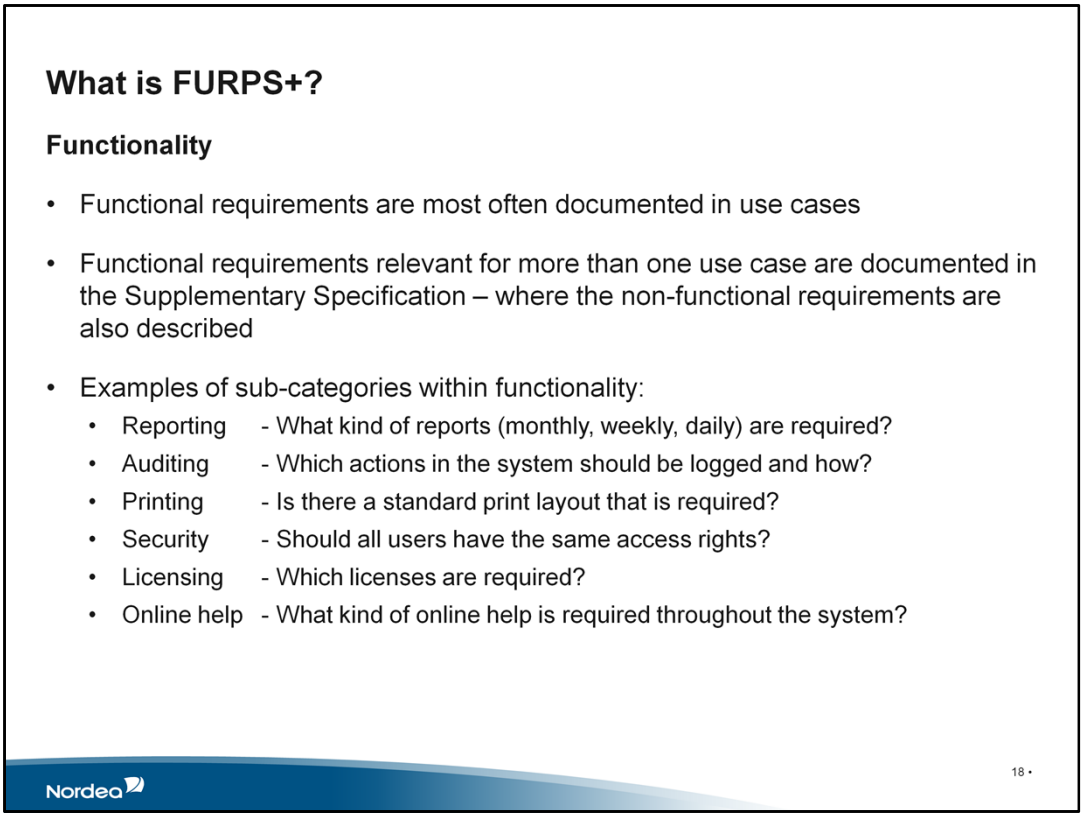
URPS+ are the non-functional requirements.

URPS are quality requirements, + is constraints.

The categories are primarily meant as a support in remembering or finding all requirements.

Meaning: What category the different requirements belong to is not so important. What is important is to find them all.

We will talk a lot more about these categories on the following slides.



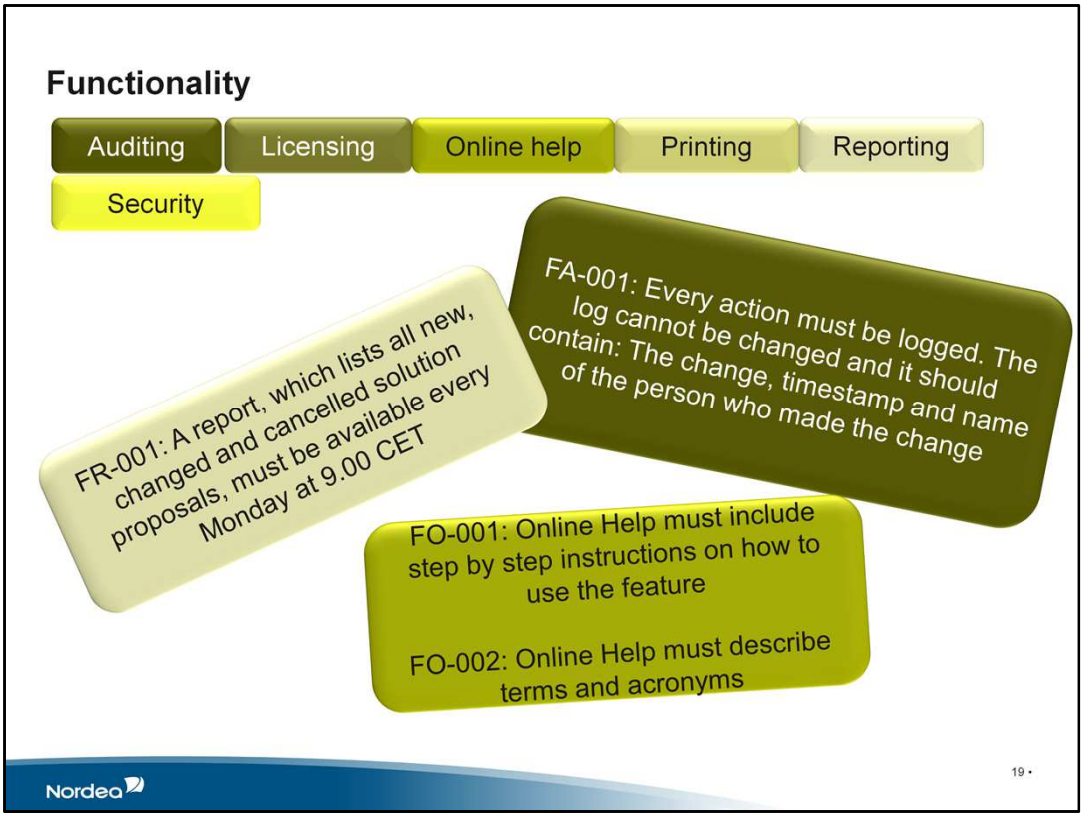
The F in the FURPS+ acronym represents all the system-wide functional requirements that we would expect to see described. These usually represent the main product features that are familiar within the business domain of the solution being developed.

The system wide functional requirements can be very technically oriented. This is another reason why people have trouble capturing them and stating them as requirements - people may be very familiar with the business domain concepts, but not so familiar with technology concepts.

Functional requirements that you may consider to be also **architecturally significant system-wide functional requirements may also include**

- **localization**
- **mail**
- **system management**
- **workflow**

Each of these may represent functionality of the system being developed and they are each a system-wide functional requirement.

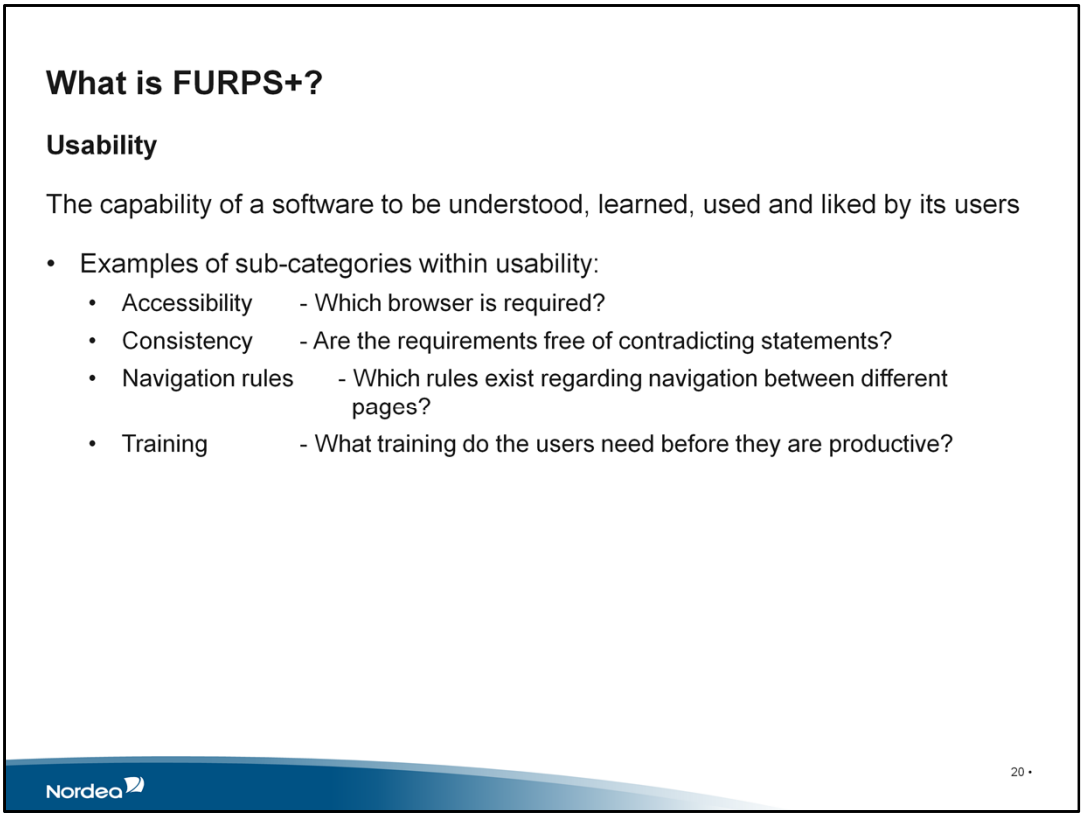


Many of the functional requirements are documented in use cases, however some are not applied to a specific use case. Therefore they should be included in the Supplementary Specification.

Requirements should have unique IDs in order to increase traceability. **The abbreviation can be decided within each project.** What we have used here is only an example of what the IDs could be like.

FR - Functionality Reporting
FA - Functionality Auditing
FO - Functionality Online Help

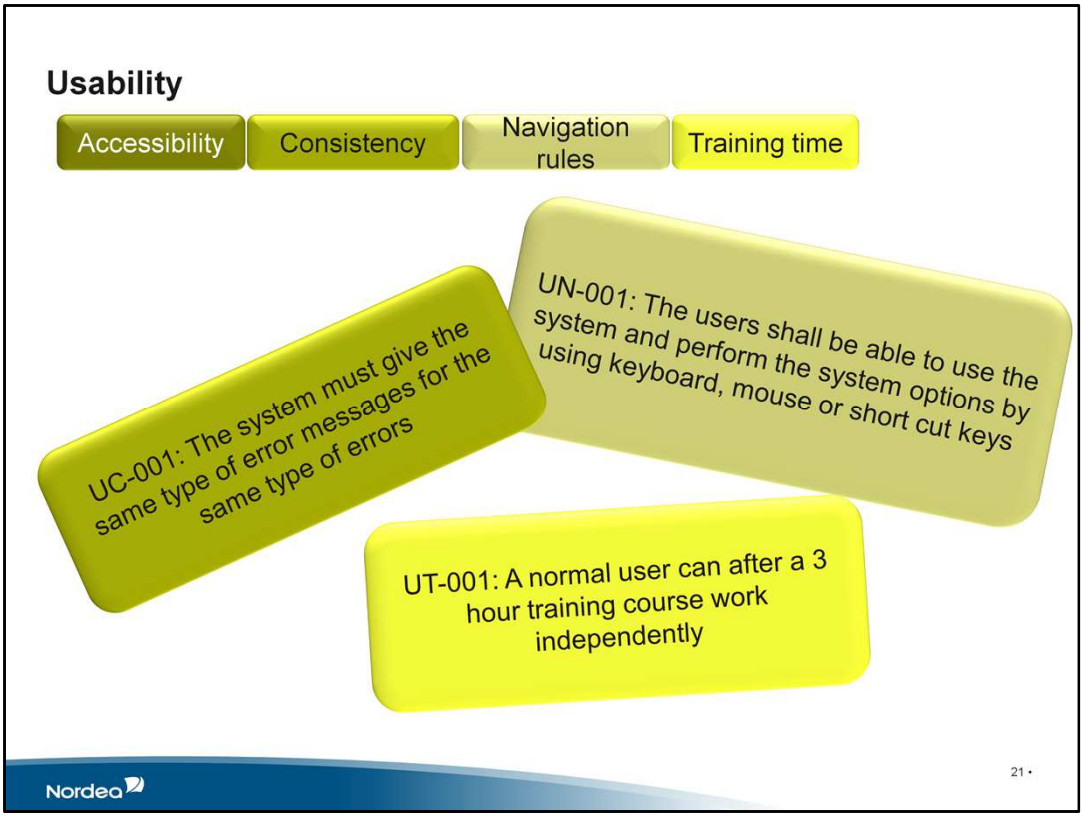
All examples – on requirements in this category and the others - are taken from the Supplementary Specification Guidelines.



Usability is about how easy it is for the users to use the system. This can be tested by having real users trying to use the system and report the users behaviour.

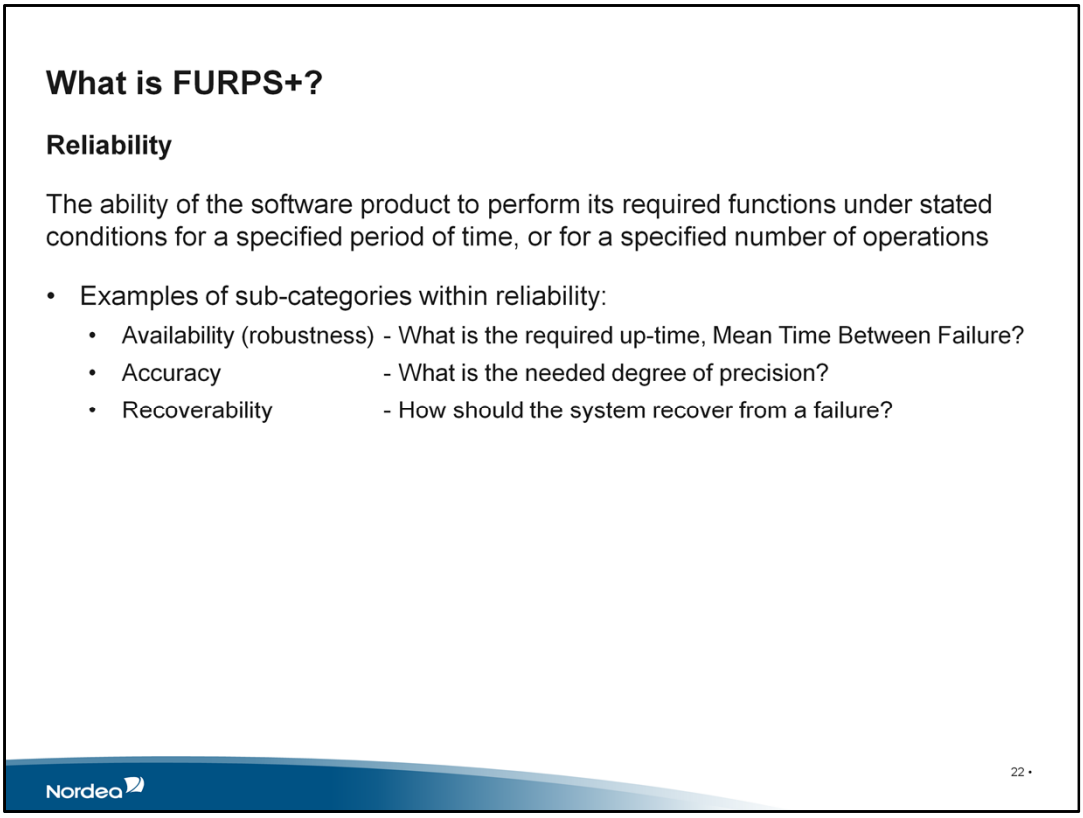
Consistency covers the degree of uniformity, standardization and freedom from contradiction among the documents or parts of component or system.

Maybe it is necessary to have separate focus on usability – depending on what you are developing.



Training time – To be a “good” requirement a definition (or at least a common understanding) of what is meant by a “normal user” is required.

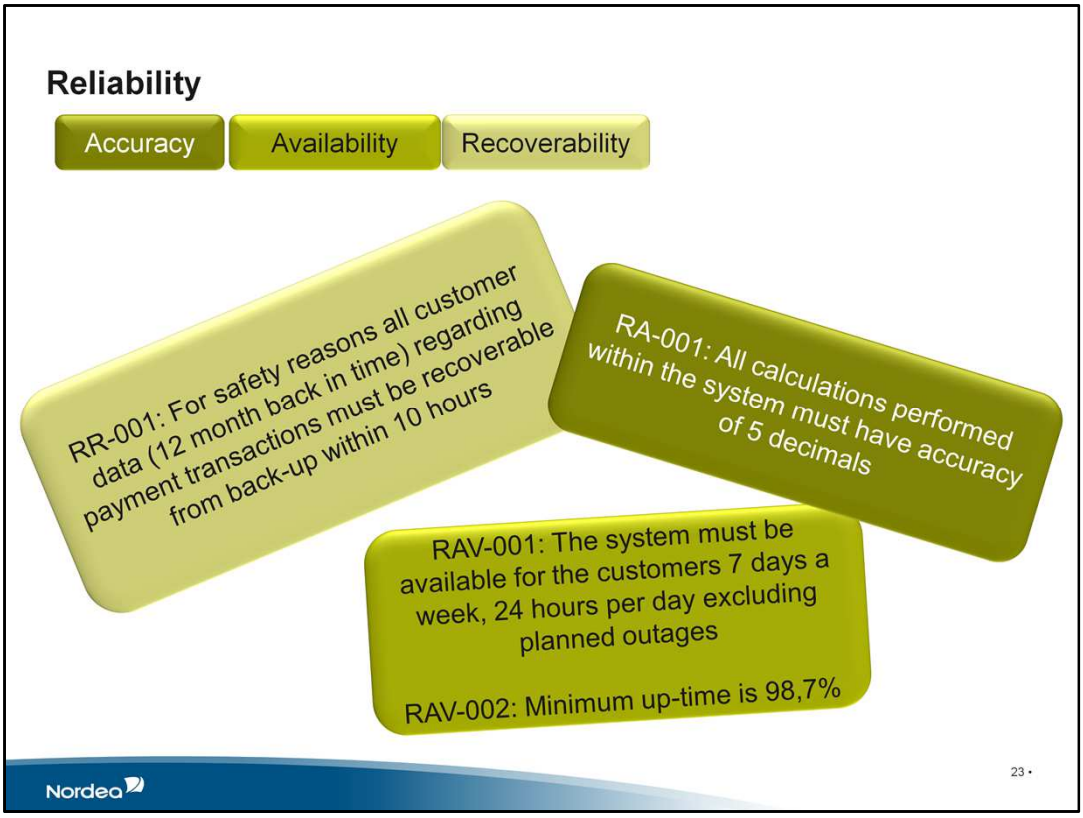
UC - Usability Consistency
UN - Usability Navigation rules
UT - Usability Training time



Availability means the degree to which a component or system is operational and accessible when required for users. Often expressed as percentage.
Mean Time Between Failure is usually written as MTBF.

Accuracy means the capability of the software product to provide the right or agreed results or effects with the needed degree of precision.

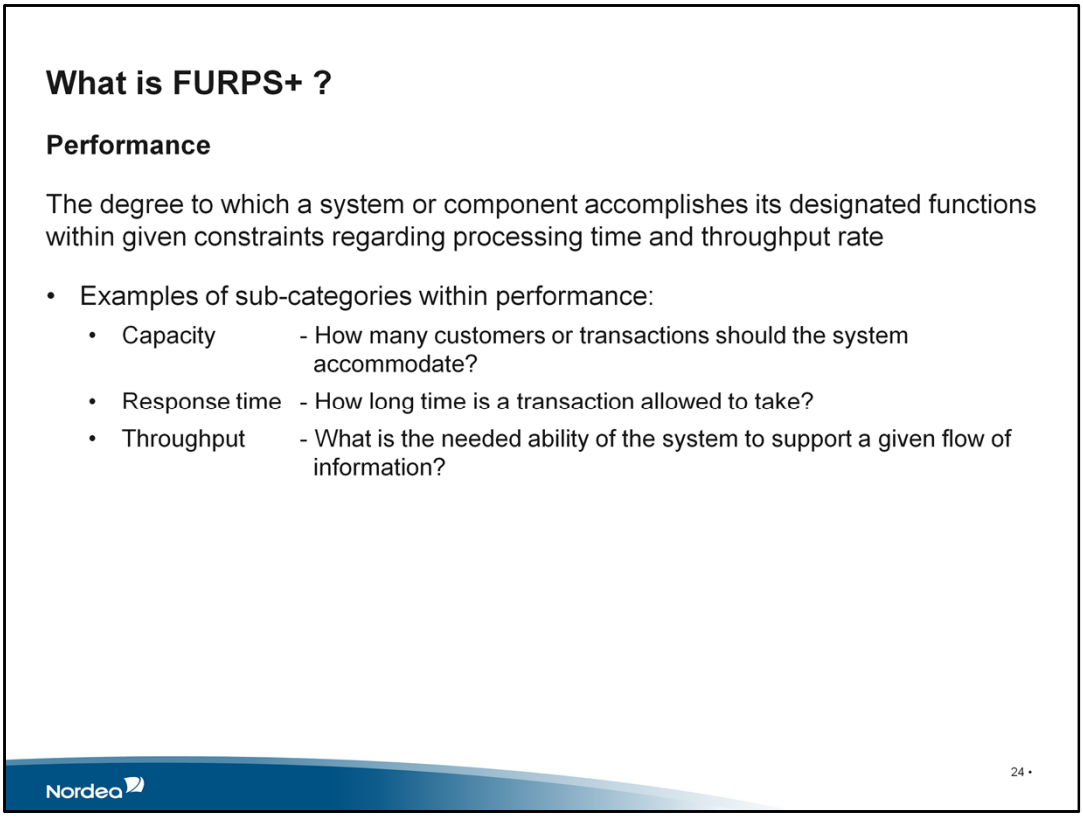
Recoverability means the capability of the software product to re-establish a specified level of performance and recover the data directly affected in case of failure. E.g. if a break down should happen during money transfer what data must not be lost.



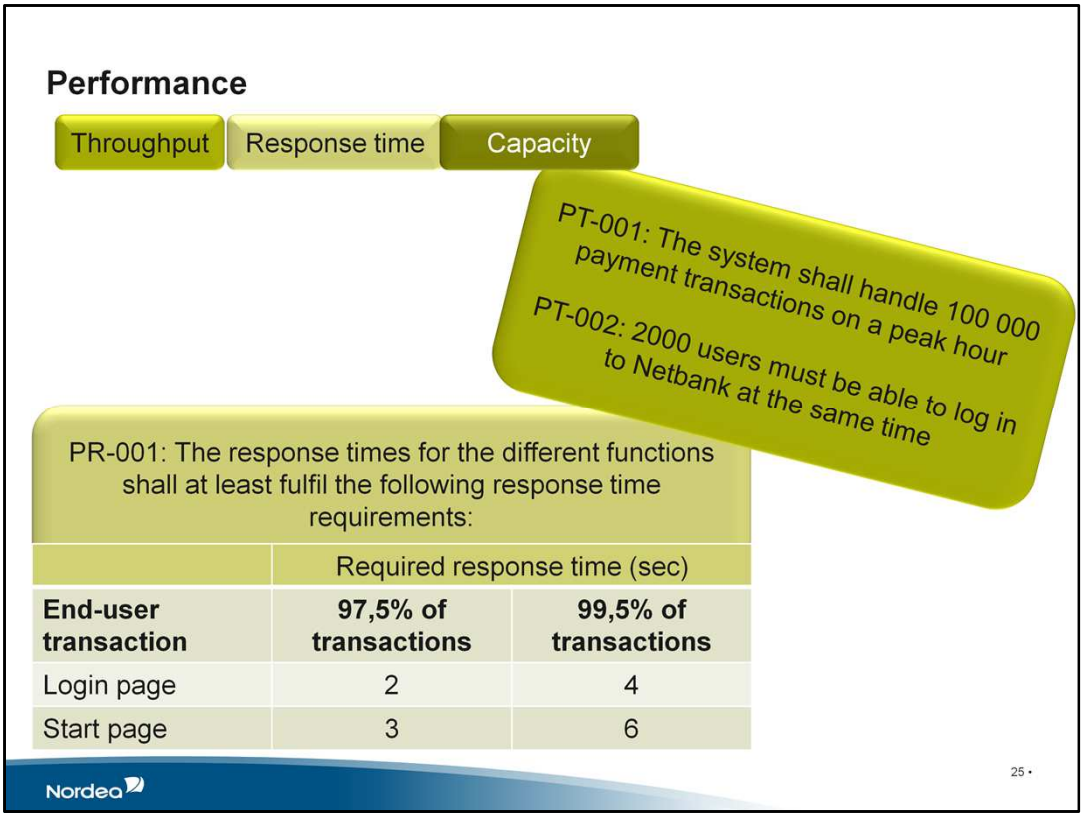
Accuracy – Even though the results of calculations are only shown on the screen with 2 decimals it might still be a requirement that all calculations should be done with 5 decimals.

Availability - Some requirements might imply requirements not only to the software but also to hardware, surveillance, monitoring, SLA's etc.
In some situations it will be necessary to give the stakeholder stating the requirements an estimate of what it will require and cost to fulfil the requirements. Knowledge about the cost might make him lower the requirements regarding availability.

RA - Reliability Accuracy
RAV - Reliability Availability
RR - Reliability Recoverability

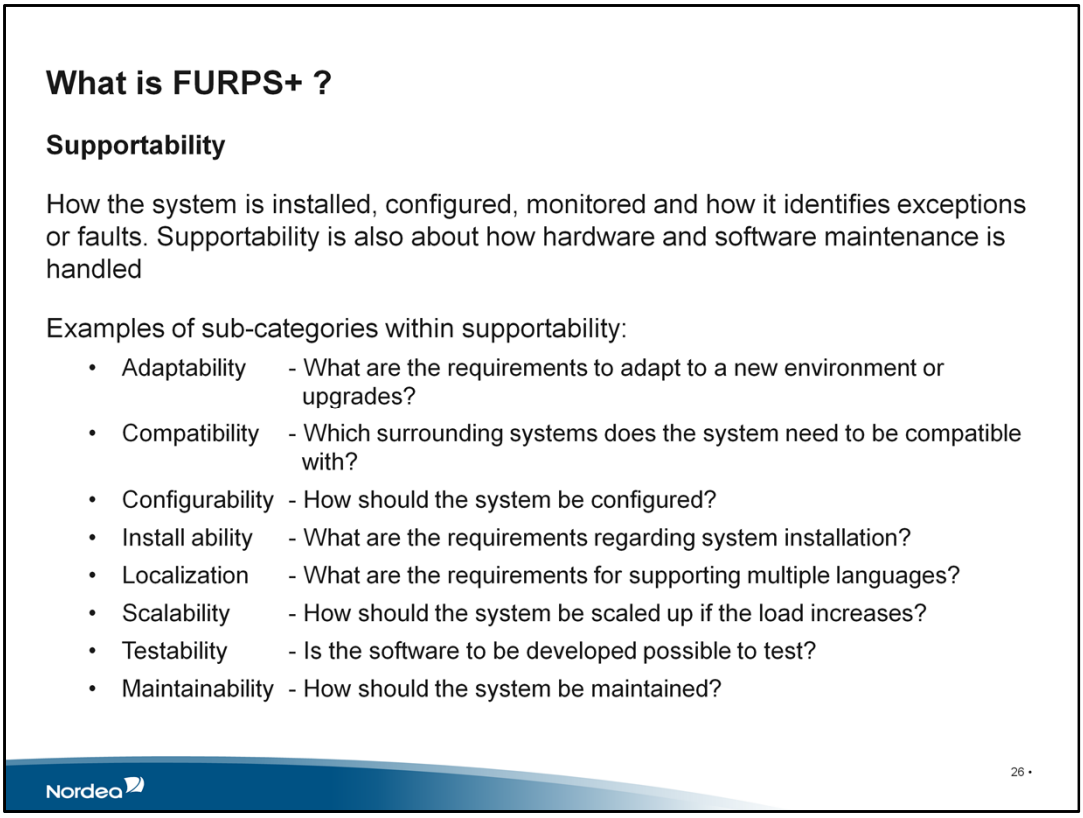


This is the category that most people think of when they hear the term non-functional requirements!



Response time – Requirements regarding response time is often described in tables like this.
When capturing requirements regarding response time keep in mind that the response time may be different from normal situations to peak hours.

PT - Performance Throughput
PR - Performance Response time
PC - Performance Capacity



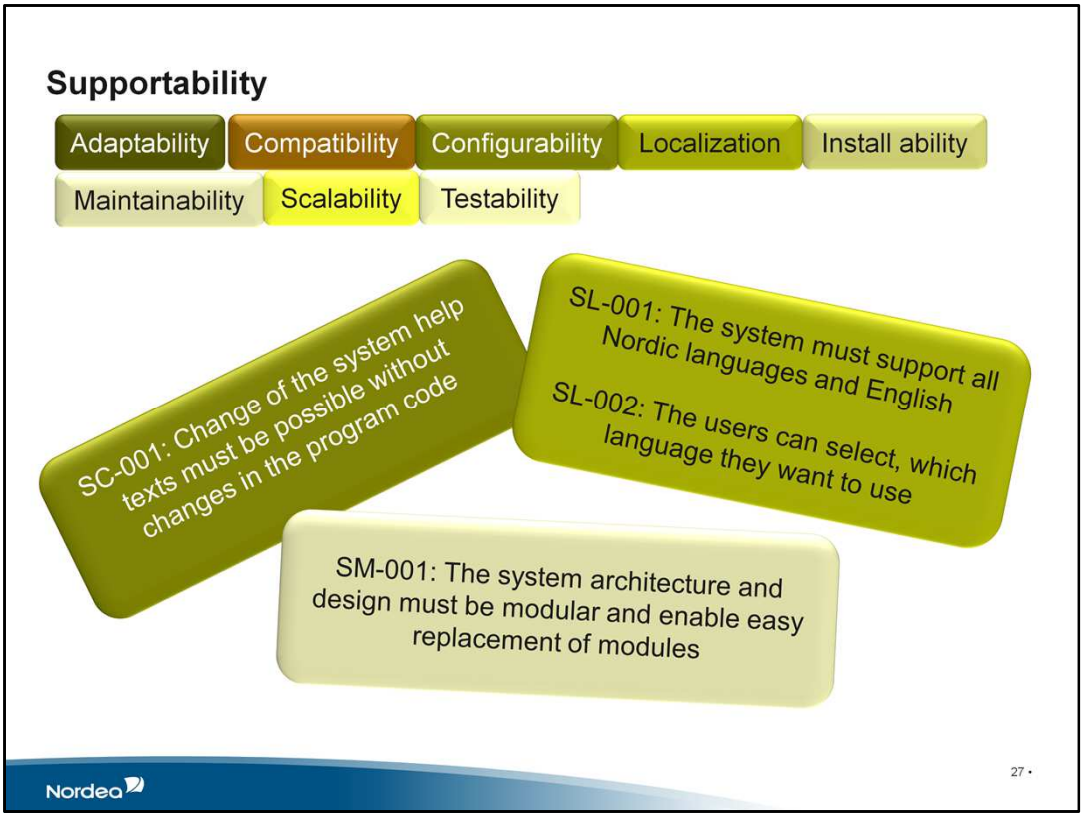
Supportability is the most technical oriented non-functional requirements category. That is probably the reason why a lot of business people find it the hardest to describe their requirements within this category.

Adaptability means the capability of the software product to be adapted for different specified environments. This without applying actions or means other than those provided for this purpose for the software considered.

Scalability means the capability of the software product to be upgraded to accommodate increased loads.

Testability means the capability of the software product to enable modified software to be tested.

Maintainability means the ease with which a software product can be modified to correct defects, modified to meet new requirements, modified to make future maintenance easier, or adapted to a changed environment.

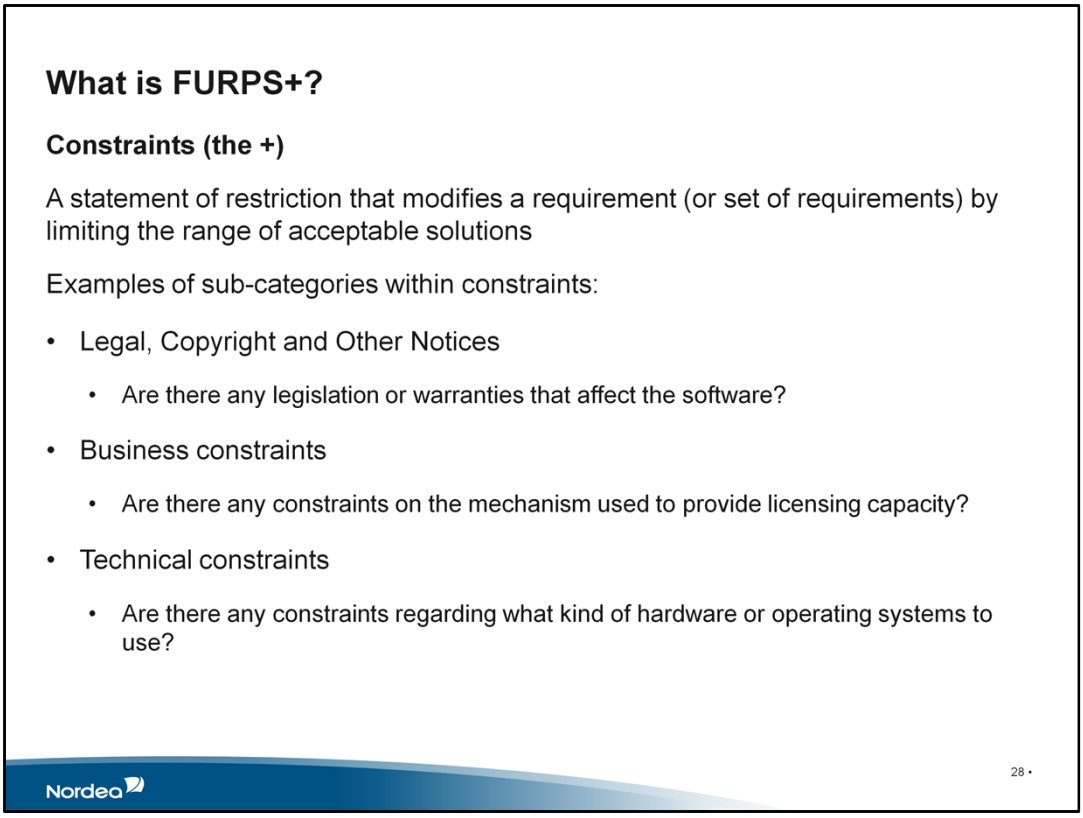


Compatibility – It is important to find out if there are requirements on compatibility of this system with previous versions of the system or with legacy systems providing the same capability?
An example could be: The system web services should be back compatible at least one version.

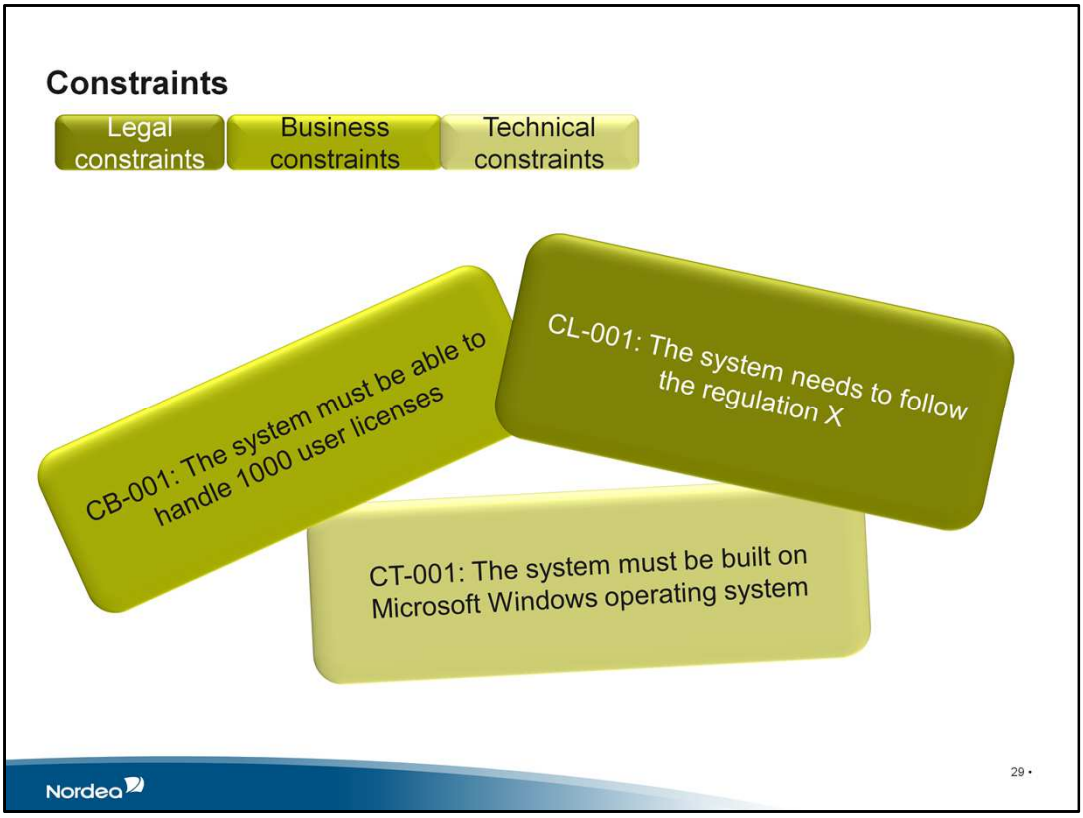
Scalability – This can be regarding increased data volumes or number of users. What volumes of users and data will the system support? This may be expressed as a profile over time.
An example could be: If load in the system increases, it must be possible to add new hardware without changing program code.

Testability – An example could be: Subsystems should be possible to be tested separately using mock-ups simulating surrounding systems.

SC - Supportability Configurability
SL - Supportability Localization
SM - Supportability Maintainability



If you look for descriptions of FURPS+ on the internet you might find different types of constraints like design constraints, implementation constraints, interface constraints, physical constraints. If it is relevant to use other or more types of constraints (sub categories) than the three listed on the slide, feel free to add them.

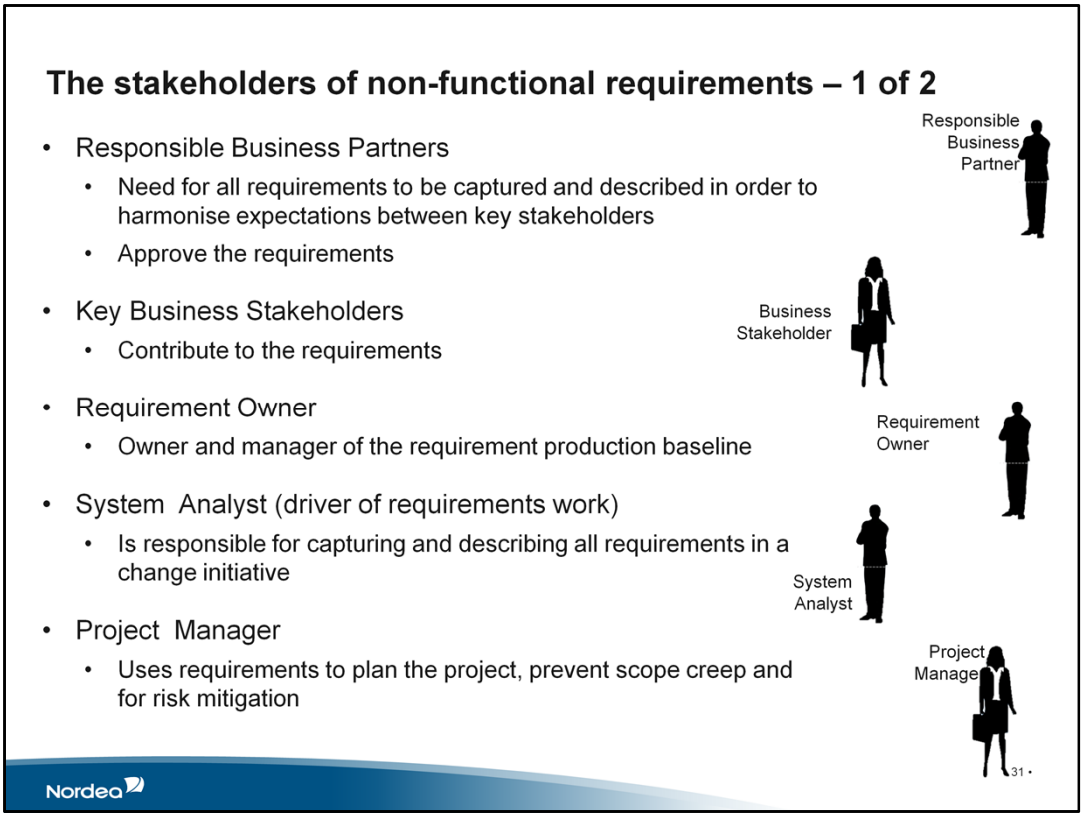


CL - Constraints Legal
CB - Constraints Business
CT - Constraints Technical

The URPS+ categories are meant to help us remember all aspect of the non-functional requirements. It is not so important what category the requirements are placed in. Therefore when capturing requirements do not spend too much time on categorizing the requirements.

Nordea

The Stakeholders of non-functional requirements



A lot of different roles are involved in the work of capturing and/or documenting requirements – or they are dependant on the result of the requirement work.

Key Business Stakeholders
Anyone from the business area giving input to the requirements can act as a Key Business Stakeholder.
The Key Business Stakeholders contribute to non-functional requirements mainly within Usability and Reliability and also system wide Functionality (not captured in use cases).
They might need support from other project participants in order to see or understand the impact requirements have on the system to be and the cost it will be to fulfil them.
But they are the ones deciding what the requirements should be.

Requirement Owner
This is a new role in Nordea! It is described in the Requirements and Test Portal on the Nordea intranet.
The Requirement Owner is the owner and manager of the Requirement Production Baseline. If the system exists already and no baseline is set, it is the responsibility of the Requirement Owner to establish the Requirement Production Baseline.
The role as Requirement Owner can be assigned to anyone from the business area who has the most impact on the requirements.

Requirement Production Baseline is a collection of the requirements that are currently in production.
Requirement Baseline is the new requirement work on-going.

Requirement Baseline is a collection of requirement descriptions both functional and non-functional. First version is often established during a change initiative and handed over to maintenance when the IT deliveries are put into production thus becoming the first Requirement Production Baseline.
In Nordea we are working on having Requirement Production Baselines established for all applications – but it will take time…

System Analyst
In SDP – System Development Process – the driver of the requirement work is called System Analyst. But sometimes the role is called IT Analyst, Senior Analyst or something else.

Non-functional requirement stakeholders

- Software Architect
  - Contributes to non-functional requirements
  - Designs the system based on functional and non-functional requirements
- Developers and system maintenance resources
  - Contribute to non-functional requirements
  - Develop the system based on functional and non-functional requirements
- Test Manager
  - Test Manager and/or test resource(s) review both functional and non-functional requirements and approve testability
  - Is responsible for allocating needed functional and non-functional test resource(s) together with the Project Manager.
  - Is responsible for following up status of functional and non-functional tests
- Responsible IT Partner
  - Might have to be involved if there are hardware or performance issues to deal with

Nordea

32

Software Architect
Contributes to non-functional requirements mainly within Performance, Supportability and Constraints

Developers and system maintenance resources
Contribute to non-functional requirements mainly within Performance, Supportability and Constraints

Nordea

How are non-functional requirements captured and documented?

How are non-functional requirements captured?

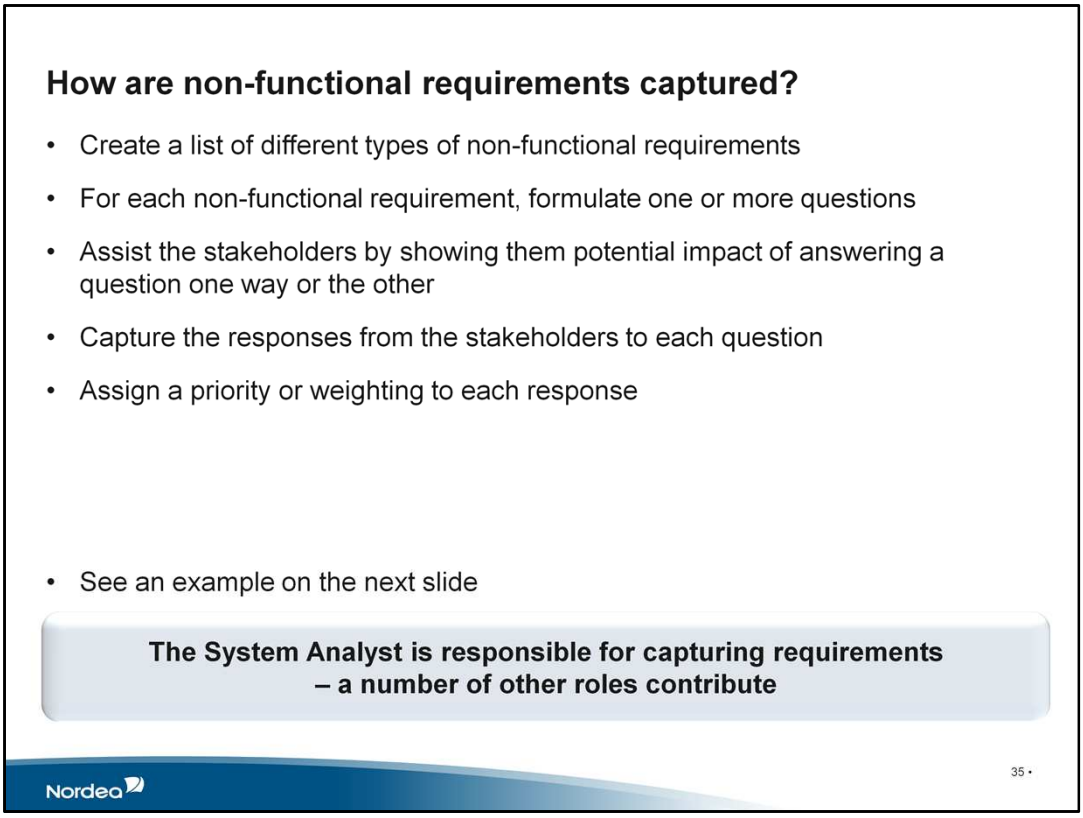
When defining non-functional requirements for a new or improved application you can conduct a **non-functional requirements workshop.**

Material to be used when facilitating the workshop is available on the portal called Step In.

Nordea

34 •

The non-functional requirements workshop gives you a **kick-start** to the work of finding and describing non-functional requirements.



This is *one* way of doing it… You can see an example of this on the next slide.

The FURPS+ categorization can help retrieve the non-functional requirements.
Remember all requirements should be based on stakeholder needs.

Note:
Performance requirements for an existing system can be derived from measurements in production.
While creating performance requirements for a new system it sometimes is impossible to specify exact needed performance figures e.g. throughput of a component, or response time for a transaction. These requirements need to be collected during testing.
Needed resources: performance tester, developer and architect.

How are non-functional requirements captured?

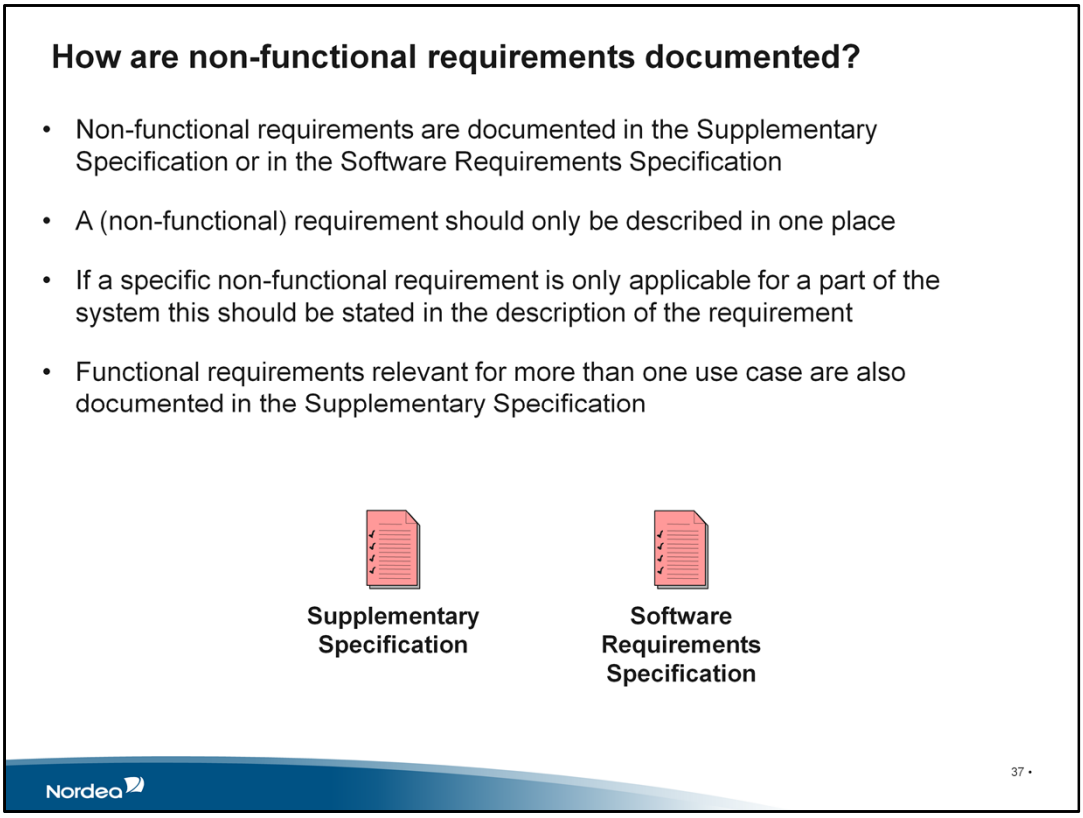
- Example of non-functional requirement questionnaire
  - The questionnaire template is called Supplementary Requirements Questionnaire and can be found in the Requirements and Test Portal on the Nordea intranet – as well as other templates to use when working with requirements

Non-functional requirement	Questions	Impact	Answers	Priority
Availability	Are there any requirements regarding the system "up time"? This may be specified in terms of mean Time Between Failures (MTBF)	The higher availability, longer time to market time.	Availability is a key product feature. The product must have a MTBF of 60 days.	High

Nordea

36 •

Adopt the language in the questions to the persons that should answer them e.g. business, architecture.



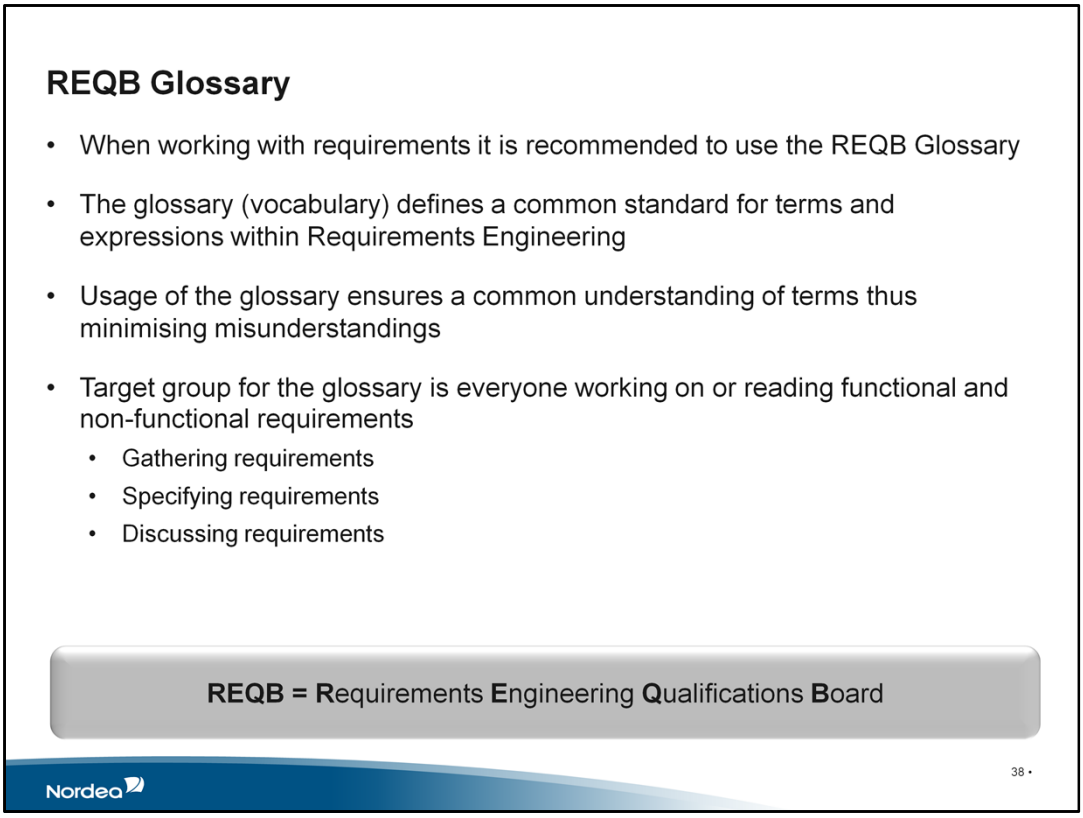
In Nordea we have 2 different requirements documentation packages to choose from:

1. **Use Cases** and **Supplementary Specification** - most beneficial when there is functionality to develop.
2. **Software Requirements Specification** - a traditional way of documenting and most beneficial when you have no functionality to develop or when functionality is already given, e.g. Batch, Commercial of the Shelf (COTS).

Projects that do not have a Supplementary Specification sometimes documents non-functional requirements in the System Architecture Document (SAD) which is *not* correct. Only links to requirement documents should be included in the SAD.

The Supplementary Specification Guidelines contains a lot of examples of non-functional requirements. These can be used as inspiration.

Note: System wide functional requirements are also documented in the Supplementary Specification.



REQB = **R**equirements **E**ngineering **Q**ualifications **B**oard is a group of international experts who work together to set an international standard (common language) for Requirements Engineering.

There are many reasons why it is a good idea to use the REQB Glossary:

- A group using terms or expressions defined in the REQB glossary "speaks" the same language
- One common language within Requirements Engineering helps to prevent misunderstandings
- It is easy to double-check the meaning of a term/expression in the glossary in case of doubts. This means no time is spent on discussing what a term or expression means
- REQB is an international standard and it is used worldwide
- ISTQB (International Software Testing Qualifications Board) is used by Test in Nordea and these two standards and glossaries are aligned. So when the requirements resources use terms or expressions as defined in the REQB glossary, it is easier for the test resources to understand what is meant and vice versa
- The quality of the requirements is raised, which leads to lower development costs and higher probability of meeting deadlines

A few examples from REQB Glossary:

- **Baseline** is a specification or software product that has been formally reviewed or agreed upon, that thereafter serves as the basis for further development, and that can be changed only through a formal change control process
- **Constraint** is a statement of restriction that modifies a requirement or set of

requirements by limiting the range of acceptable solutions

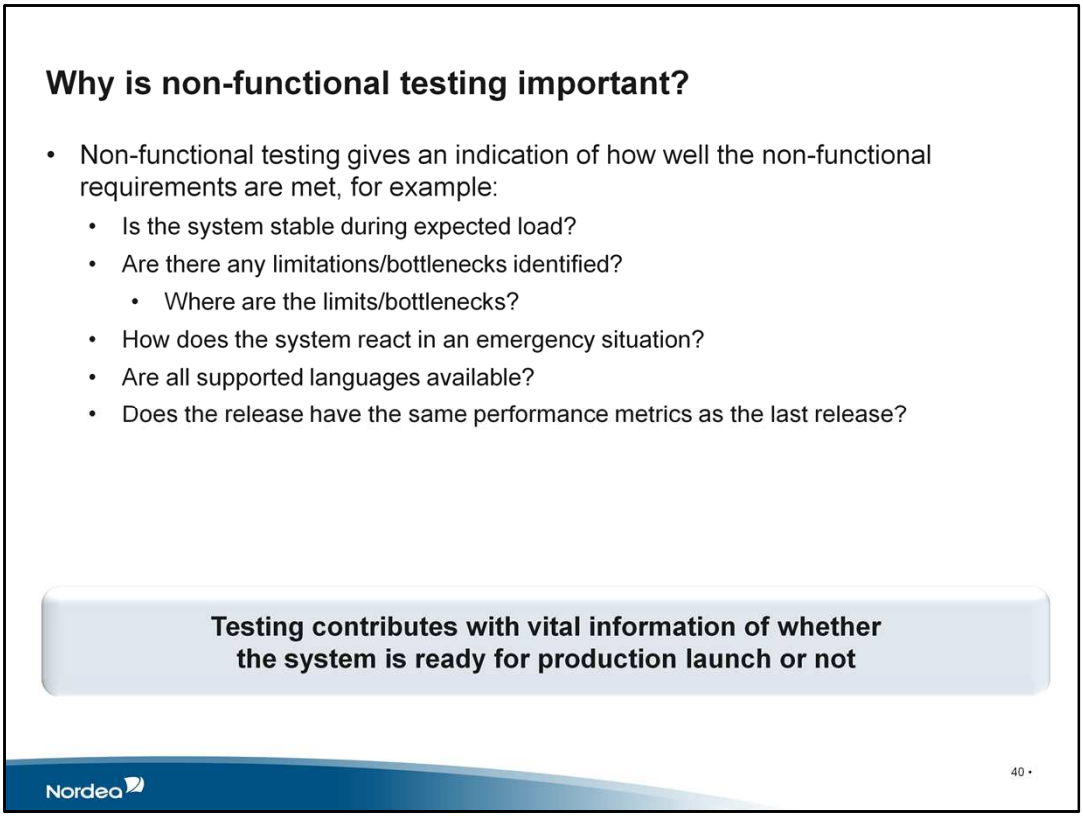
- **Elicitation** is the act of obtaining information from other people. In the context of Requirements Engineering, elicitation is the process of gathering requirements from stakeholders
- **Functional requirement** is a requirement that specifies a function that a component or system must perform
- **Non-functional requirement** is a requirement that does not relate to functionality, but to attributes such as reliability, efficiency, usability, maintainability and portability

The REQB Glossary can be found via the Requirements & Test Portal in the Nordea Intranet.

Note:
Since REQB is a glossary/vocabulary, REQB does not replace SDP or any other development method but supports a method with a common standard language.

Nordea

Why is non-functional testing important?



The result of the tests are basis for decision of whether or not to deploy to production.

What does it require to test non-functional requirements?

- Functional and non-functional requirements have been reviewed and testability approved
- Production-like test data and test environments are in place
- Test tools, e.g. Quality Center (test management tool) and Load Runner (performance test tool) are in use – contact Development Tools team
- Test cases and test sets are created and planned
- Test resources with the right competences are available - contact Testing Services

Review of requirements is an important part of testing.
This also applies to non-functional requirements!

Nordea

Where can you find more information?

When you get your Nordea user-ID and password you will be able to find all the information regarding requirements, and see the templates, guidelines and examples etc. in the Step In portal.

Step In contains descriptions of methods, tools, support and training regarding project management and system development in Nordea.
In addition to the information you will also be able to read and join conversations and discussions about relevant topics.