Functional vs. Nonfunctional Requirements

Sponsored by:

Mark A. Monteleone, PMP, CBAP, CSM/CSPO
CPRE - Certified Professional for Requirements Engineering

**Expert Level**
Prerequisite: 3 Advanced Level certificates (2 certificates replaceable)

**Advanced Level**
Available Modules:
- Requirements Elicitation and Consolidation
- Requirements Modeling
- Requirements Management
- RE@Agile

Prerequisite: Foundation Level certificate

**Foundation Level**
Prerequisite: none

**RE@Agile Primer**
Prerequisite: none
Abstract

**Functional vs. Nonfunctional Requirements**

In this webinar, Mark Monteleone describes and compares functional and nonfunctional requirements and how to document them using declaratives, user stories, and use cases in the context of software applications.

Mark starts the webinar by briefly reviewing the source of business needs – enterprise analysis. He then lists four requirement levels.

- **Business** – needs that are required to compete or be in compliance
- **Stakeholder** – what stakeholder capabilities are needed to realize business needs
- **Solution** – what product or service requirements are needed to provide stakeholder capabilities and under what conditions are the capabilities effective, efficient, and secure.
- **Transition** – what product or service requirements are needed to ensure a smooth change in implementing the new capabilities

With that quick background, Mark focuses the webinar by further decomposing solution requirements into functional capabilities and nonfunctional conditions and formally defines the word “requirement.” He highlights how these requirements are documented by business analysts using:

- Action Verbs for functional requirements with references to business rules
- Adjectives and nouns for nonfunctional requirements with embedded metrics

Mark then shows how functional and nonfunctional requirements are documented using declaratives, user stories, and use cases. To ensure clarity, he wraps-up the webinar by providing several examples of functional capabilities paired with nonfunctional conditions using the above techniques.
Presenter Biography
Mark A. Monteleone, PMP, CBAP, CSM/CSPO

• Independent consultant
  • Experienced in PM/BA over 45 years, but it’s not the years, it's the mileage*
  • Consulted in more than 35 countries
• Author of “The 20 Minute Business Analyst” and PM/BA articles in several on-line publications such as
  • BA Times
  • BA Connection
  • Modern Analyst
  • International Association of Facilitators (IAF)
    Global Flipchart
• Graduate of Texas A&M University
  • B.S. in physics
  • M.S. in computing science
• Certified as a
  • Project Management Professional (PMP®)
  • Certified Business Analysis Professional (CBAP®)**
  • Certified Scrum*** Master (CSM™)
  • Scrum Product Owner (CSPO™)

* Indiana Jones Quote – “Raiders of the Lost Ark”
** Pronounce the “b” in CBAP, not the “P” as in CPAP
*** Never forget the “R” in scrum
Agenda (on the big screen)

1. Purpose
   • Abstract
   • Presenter Biography
   • Agenda
2. Background
   • Enterprise Analysis
   • Levels of Requirements
     • Business
     • Stakeholder
     • Solution
     • Transition
   • Traceability of Requirements
3. Decomposing Solution Requirements
   • Functional Requirements - capabilities
   • Nonfunctional Requirements - conditions
4. My Definition of Requirement
5. Documenting Functional and Nonfunctional
6. Methods of Documenting Requirements
   • Declaratives
   • User Stories
   • Use Cases
7. Examples of Paired Functional and Nonfunctional Requirements
8. Webinar Summary
9. Wrap-up with Questions

Let’s have a brief background on requirements first.
Background: Enterprise Analysis

- Understanding how the business interfaces with the ecosystem typically via a marketing study
- The study identifies changes needed to continue a successful business. These changes may involve:
  - Business Model Canvas
  - Value Proposition Model
  - Customer satisfaction
  - Competition
  - Compliance
- Based on the study, business analysts propose initiatives to accomplish needed changes (business requirements) and document them as business cases.
- In the final step, an executive committee decides to pursue selected initiatives and fund them as projects.
- In the context of each project, business analysts together with stakeholders determine requirements...
Background: Levels of Requirements

- **Business Requirements**, discovered during enterprise analysis, are changes that are needed to compete or be in compliance with government laws and regulations.

- **Stakeholder Requirements** are capabilities that are needed to realize business changes.

- **Solution Requirements** are product or service capabilities and conditions needed to support stakeholder requirements.

- **Transition Requirements** are processes needed to ensure a smooth change in implementing solution requirements.
Background: Traceability of Requirements

- **Business Requirements** are changes that are needed to compete or be in compliance with government laws and regulations.
- **Stakeholder Requirements** are capabilities that are needed to realize business changes.
- **Solution Requirements** are product or service capabilities and conditions needed to support stakeholder requirements.
- **Transition Requirements** are processes needed to ensure a smooth change in implementing solution requirements.
1. Purpose
   • Abstract
   • Presenter Biography
   • Agenda

2. Background
   • Enterprise Analysis
   • Levels of Requirements
   • Business
   • Stakeholder
   • Solution
   • Transition
   • Traceability of Requirements

3. Decomposing Solution Requirements
   • Functional Requirements - capabilities
   • Nonfunctional Requirements - conditions

4. My Definition of Requirement

5. Documenting Functional and Nonfunctional

6. Methods of Documenting Requirements
   • Declaratives
   • User Stories
   • Use Cases

7. Examples of Paired Functional and Nonfunctional Requirements

8. Webinar Summary

9. Wrap-up with Questions

OK, with that background, let’s now talk about functional and nonfunctional requirements.
Decomposing Solution Requirements

- Business Requirements
- Stakeholder Requirements
- Solution Requirements are product or service capabilities and conditions needed to provide stakeholder requirements
  - **Functional** - describes software capabilities needed to provide a service or produce a product in support of a higher level requirement
    - May be constrained by business rules, but documented separately from the capability as a best practice (allows for independent update)
  - **Nonfunctional** - describes the conditions that capabilities need in order to be effective, efficient, and secure (quality of service)
    - Equally important as functional requirements
    - Clarified by qualitative/quantitative metrics and documented with the condition
    - Typically applies to all functional requirements
    - Focus of infrastructure personnel
- Transition Requirements
My Definition of Requirement (software context)

In general, a requirement is an unfulfilled need that must be achieved to realize a quality product or service. A feature is a fulfilled need(s), i.e., an implemented requirement(s).

Stakeholders (users) describe their requirements as capabilities that they need to support stated business requirements. Business analysts decompose their needed capabilities and document them as solution functional and nonfunctional requirements using

- action verbs
- possible constraining business rule(s)
- possible condition(s) with metrics

“I want access to customer accounts per BR35 with sub-second response time.”

BR35. Access to customer accounts is limited to the manager over those customers. This capability needs to be tested with a manager who is over certain customers and with a manager who is not.

“The application needs to respond in sub-seconds.” This condition applies to the entire application, not just one capability.
Documenting Functional and Nonfunctional Requirements

- **Functional Requirements - capabilities**
  - Written using action verbs with references to business rule tags
  - General action verbs for capabilities
    - Access
    - Create
    - Read
    - Update
    - Delete
    - Entry
    - Store
    - Display
    - Print
    - Calculate
    - Transmit
    - Track

- **Nonfunctional Requirements - conditions**
  - Written using adjectives and nouns with metric specifics
  - General areas used for conditions (metrics)
    - Performance (response time)
    - Availability (when and at what percentage)
    - Reliability (percentage)
    - Portability (use alternatives)
    - Capacity/scalability (user volume)
    - Maintainability (update ease)
    - Compatibility (common connection)
    - Usability (intuitive action)
    - Security/audit (authority, restrictions)
    - Data retention (time duration)
    - Backup/restore, disaster/recovery (time)
    - Training (help online or manuals)
    - Documentation (references)
Methods for Documenting Requirements (software context)

• Different Methods can be used to document Requirements
  • Declarative (IEEE Standard)
  • User Story
  • Use Case
Methods in Documenting Requirements (software context)

IEEE Standard Format (used for functional and nonfunctional Requirements)

- **Template**
  - The `<system>` shall/must `<behavior>`.
  - The `<system>` shall/must `<behavior>` if /where/using `<conditions>`.
  
  - Shall/Must – mandatory
  - Should/May – optional
  - Will – provided by another system

- **Examples (FRx – Functional Requirement, NFRx – Nonfunctional Requirement)**
  - **FR1.** The ATM shall accept withdrawal requests if the amount requested conforms to business rules BR11 and BR12.
  - **NFR1.** The ATM shall require a PIN and password for access.
  - **FR2.** The Accounting System **will** provide the interface to all customer accounts.
  
  - **BR11.** Withdrawal requests must be less than $300.
  - **BR12.** An Account must retain a balance of $50.
Methods in Documenting Requirements (software context)

User Story Format (used for functional and nonfunctional)

**Unique Title:** Action Verb <optional qualifier> Noun

**Conversation:**

“As a <user role>, I want <action feature> so that <benefit received>.”

**Goal**

**Actor**

**Priority:** High, Medium, Low

**Business Value Estimate:** $, $$, $$$, $$$$
Methods in Documenting Requirements (software context)

User Story Format (used for functional and nonfunctional)

Acceptance Criteria:
- Given <input> when <user action> then <result>
- Given <input> when <user action> then <result>
- Given <input> when <user action> then <result>

Constraints:

Black Box Test
- Positive Test
- Negative Test

Business Rule
Expected Result
Use Case Format (used for only functional requirements)

1.0 The use case starts when a customer requests to order a computer
2.0 The customer enters the product code for the desired computer per BR20
3.0 The system supplies a product description and price for the matching product code
4.0 The system requests credit card information
5.0 The customer enters credit card information per BR24
6.0 The customer submits payment
7.0 The system decides payment is authorized by the credit card company
   7.1 The system marks the order as confirmed
   7.2 The system forwards payment information to the Accounting System
   7.3 The system returns an order confirmation to the customer
   7.4 Use case ends
7.e The system decides the payment is unauthorized by the credit card company
   7.e.1 The system returns error to customer
   7.e.2 Use case ends

Nonfunctional requirements are documented as declaratives on a separate list.
Agenda (on the big screen)

1. Purpose
   • Abstract
   • Presenter Biography
   • Agenda
2. Background
   • Enterprise Analysis
   • Levels of Requirements
     • Business
     • Stakeholder
     • Solution
     • Transition
   • Traceability of Requirements
3. Decomposing Solution Requirements
   • Functional Requirements - capabilities
   • Nonfunctional Requirements - conditions
4. My Definition of Requirement
5. Documenting Functional and Nonfunctional
6. Methods of Documenting Requirements
   • Declaratives
   • User Stories
   • Use Cases
7. Examples of Paired Functional and Nonfunctional Requirements
8. Webinar Summary
9. Wrap-up with Questions

How about some more examples of functional and nonfunctional requirements?
Review Format (user story) of Paired Functional and Nonfunctional Requirements Examples

**Functional**

Capability with associated Business Rule Tag (BRxx)
(create, read, update, delete, transmit, calculate, print, display, enter, store, access, track)

**NonFunctional**

(quality of service)

Condition with imbedded Metric
(performance, availability, reliability, portability, capacity/scalability, maintainability, compatibility, usability, security/audit, data retention, backup/restore, disaster/recovery, training, documentation)
Capability with associated Business Rule Tag (BRxx)
(create, read, update, delete, transmit, calculate, print, display, enter, store, access, track)

As a customer, I want to **pay via a credit card**.

Given an unauthorized company credit card, when I attempt to pay by credit card, then the charge is not accepted.

**Acceptance Criteria**

**Functional**

Condition with quantitative imbedded Metric
(performance, availability, reliability, portability, capacity/scalability, maintainability, compatibility, usability, security/audit, data retention, backup/restore, disaster/recovery, training, documentation)

As a customer, I want a sub-second response so that I pay via a credit card quickly.

**Nonfunctional (quality of service)**
Functional

Capability with associated Business Rule Tag (BRxx)
(create, read, update, delete, transmit, calculate, print, display, enter, store, access, track)

As a customer, I want to buy movie tickets on-line so that I can avoid a long line.

Given a noncurrent movie, when I attempt to buy a movie ticket, then a ticket is not issued.

Nonfunctional (quality of service)

Condition with quantitative imbedded Metric
(performance, availability, reliability, portability, capacity/scalability, maintainability, compatibility, usability, security/audit, data retention, backup/restore, disaster/recovery, training, documentation)

As a customer, I want to buy movie tickets on-line 24/7 so that I can buy tickets at any time.
### Functional

Capability with associated Business Rule Tag (BRxx)
(create, read, update, delete, transmit, calculate, print, display, enter, store, access. track)

**Acceptance Criteria**

As a customer, I want to **update data in multiple locations**.

Given data needed at different locations, when I update data, then the same data is stored in multiple locations.

### Nonfunctional (quality of service)

Condition with quantitative imbedded Metric
(performance, availability, reliability, portability, capacity/scalability, maintainability, compatibility, usability, security/audit, data retention, backup/restore, disaster/recovery, training, documentation)

**Acceptance Criteria**

As a customer, I want data **updated consistently at all locations 99.99% of the time**.
**Functional**

Capability with associated Business Rule Tag (BRxx)
(create, read, update, delete, transmit, calculate, print, display, enter, store, access, track)

Acceptance Criteria

As a customer, I want the capability to unlock my office door from my mobile phone.

Given an ABC mobile phone, when I attempt to unlock my office door, then the door opens.

**Nonfunctional (quality of service)**

Condition with qualitative imbedded Metric
(performance, availability, reliability, portability, capacity/scalability, maintainability, compatibility, usability, security/audit, data retention, backup/restore, disaster/recovery, training, documentation)

Acceptance Criteria

As a customer, I want to unlock my office door from several types of mobile phones.
**Functional**

Capability with associated Business Rule Tag (BRxx)
(create, read, update, delete, transmit, calculate, print, display, enter, store, access, track)

As a customer, I want remote access to my orders per BR15.

BR15: All remote access needs to be through two-factor identification.

BR15 could be considered a security/audit condition.

**Nonfunctional (quality of service)**

Condition with quantitative imbedded Metric
(performance, availability, reliability, portability, capacity/scalability, maintainability, compatibility, usability, security/audit, data retention, backup/restore, disaster/recovery, training, documentation)

As a customer, I want to access my orders even if there are 1000 users logged-in.
Functional

Capability with associated Business Rule Tag (BRxx)
(create, read, update, delete, transmit, calculate, print, display, enter, store, access, track)

As a customer, I want to calculate taxes on purchases per BR16.

BR16: The tax formula ..........

Nonfunctional
(quality of service)

Condition with qualitative imbedded Metric
(performance, availability, reliability, portability, capacity/scalability, maintainability, compatibility, usability, security/audit, data retention, backup/restore, disaster/recovery, training, documentation)

As a customer, I want to easily change tax tables used in calculations.
Functional

Capability without an associated Business Rule
(create, read, update, delete, transmit, calculate, print, display, enter, store, access, track)

As a customer, I want to access my project data from various applications.

Nonfunctional (quality of service)

Condition with qualitative imbedded Metric
(performance, availability, reliability, portability, capacity/scalability, maintainability, compatibility, usability, security/audit, data retention, backup/restore, disaster/recovery, training, documentation)

As a customer, I want to access my project data via a common protocol using different applications.
**Functional**

Capability without an associated Business Rule
(create, read, update, delete, transmit, calculate, print, display, enter, store, access, track)

As a customer, I want to be able to create, read, update, and delete project records.

**Nonfunctional**

(quality of service)

Condition with qualitative imbedded Metric
(performance, availability, reliability, portability, capacity/scalability, maintainability, compatibility, usability, security/audit, data retention, backup/restore, disaster/recovery, training, documentation)

As a customer, I want an intuitive navigation method to create, read, update, and delete project records.
Functional

Capability without an associated Business Rule
(create, read, update, delete, transmit, calculate, print, display, enter, store, access, track)

As a customer, I want users to enter and store personnel records.

Nonfunctional
(quality of service)

Condition with qualitative imbedded Metric
(performance, availability, reliability, portability, capacity/scalability, maintainability, compatibility, usability, security/audit, data retention, backup/restore, disaster/recovery, training, documentation)

As a customer, I want authorized users to enter and store personnel records.
### Functional

Capability without an associated Business Rule
(create, read, update, delete, transmit, calculate, display, enter, store, access, track)

As a customer, I want to **track** current purchase orders.

### Nonfunctional

(quality of service)

Condition with quantitative imbedded Metric
(performance, availability, reliability, portability, capacity/scalability, maintainability, compatibility, usability, security/audit, **data retention**, backup/restore, disaster/recovery, training, documentation)

As a customer, I want to **retain** purchase orders for up to **7 years**.
Functional

Capability without an associated Business Rule
(create, read, update, delete, transmit, calculate, print, display, enter, store, access, track)

As a customer, I want to be able to restore all current and past purchase orders.

Nonfunctional
(quality of service)

Condition with qualitative imbedded Metric
(performance, availability, reliability, portability, capacity/scalability, maintainability, compatibility, usability, security/audit, data retention, backup/restore, disaster/recovery, training, documentation)

As a customer, I want current and past purchases backed-up on-site on a daily basis.
**Functional**

Capability without an associated Business Rule
(create, read, update, delete, transmit, calculate, print, display, enter, store, access, track)

As a customer, I want all purchases backed-up off-site on a weekly basis.

**Nonfunctional (quality of service)**

Condition with qualitative imbedded Metric
(performance, availability, reliability, portability, capacity/scalability, maintainability, compatibility, usability, security/audit, data retention, backup/restore, disaster/recovery, training, documentation)

As a customer, I want all purchases backed-up off-site on a weekly basis.
Capability without an associated Business Rule
(create, read, update, delete, transmit, calculate, print, display, enter, store, access, track)

As a customer, I want to transmit accounting records to a remote location.

Condition with qualitative imbedded Metric
(performance, availability, reliability, portability, capacity/scalability, maintainability, compatibility, usability, security/audit, data retention, backup/restore, disaster/recovery, training, documentation)

As a customer, I want user training on how to transmit accounting records to a remote locale.

If the word “initial” is used in terms of training, then this Nonfunctional Requirement could be considered a Transitional Requirement.
**Functional**

Capability without an associated Business Rule
(create, read, update, delete, transmit, calculate, print, display, enter, store, access, track)

**Nonfunctional**
*(quality of service)*

Condition with qualitative imbedded Metric
(performance, availability, reliability, portability, capacity/scalability, maintainability, compatibility, usability, security/audit, data retention, backup/restore, disaster/recovery, training, documentation)

As a customer, I want to **print** sales records.

As a customer, I want user **documentation** on how to print.
Webinar Summary

• **Functional Requirements** - describes solution capabilities needed to provide a service or produce a product to support a higher level requirement
  • Written using action verbs (declaratives, user stories, use cases)
  • May be constrained by business rules, but documented separately from the capability

• **Nonfunctional Requirements** - describes the conditions that the capabilities are tied to in order to be effective, efficient and secure (quality of service)
  • Equally important as functional requirements
  • Written using adjectives and nouns (declaratives, user stories, but not use cases)
  • Clarified by qualitative/quantitative metrics and documented with the condition
  • Focus of infrastructure personnel
The CPRE is worldwide acknowledged – over 42,000 certified professionals in 77 countries

Gain the relevant skills, improve your professional profile and you will gain a higher market value.

- Work more efficiently with stakeholders during elicitation.
- **Handle functional and nonfunctional requirements**
- Learn the appropriate methods for successful negotiations.
- Be familiar with different techniques for modeling and documenting requirements.
- Be more effective at managing requirements during the entire lifecycle of a project or product.
- Know the terminology of RE
- Become part of an internationally recognized community.

[www.ireb.org](http://www.ireb.org)
Wrap-up with Questions

Thank You for Attending and to Our Sponsor
References

1. Business Model Generation by Alexander Osterwalder and Yves Pigneur
2. Value Proposition Design by Alexander Osterwalder and Yves Pigneur
3. Business Analysis Book of Knowledge (BABOK) by International Institute of Business Analysis (IIBA)

Access www.modernanalyst.com to listen to a recording of this webinar and obtain a copy of the slides.