

Requirement Types

1. Why should we care?
2. What are they?
3. How should we use them?

RTP IIBA Chapter, April 26th, 2008

Razvan Radulian, VP of Marketing

Founder & President of Why-What-How Consulting, LLC



Objectives/Agenda

- Establish the **core/common language**, to facilitate communication and avoid waste and confusion:
 - inside the team:
Business Analysts, Project Managers
 - with other partners:
Customers, Users, Technical team, Ops, Support, Training, etc.
- The **What & Why**:
 - What are the Requirement Types?
 - Why do we need them?
 - Who needs them? When do we need them?
- The **How**: principles and practices (high-level)
 - Who will define them? How?
 - Who will use them? How?
- **Practice** (hands-on exercise)
- Conclusions, **lessons to take away...**



Core definitions: A Requirement...

BABOK 2.0 (DRAFT):

1. A condition or capability needed by a stakeholder to **solve a problem** or **achieve an objective**.
2. A condition or capability that must be met or possessed by a **solution or solution component** to satisfy a contract, standard, specification, or other formally imposed documents.
Note: “solution...” replaces old “system...” in BABOK 1.6
3. A **documented representation** of a condition or capability as in (1) or (2).



... or, in plain English

Merriam Webster's dictionary:

1. something required:
 - a: something wanted or needed : necessity <production was not sufficient to satisfy military *requirements*>
 - b: something essential to the existence or occurrence of something else
2. condition <failed to meet the school's *requirements* for graduation>

Ralph Young^{#2}

“a statement that identifies a **capability, characteristic, or quality factor** of a system in order for it to have value by a user or a customer to **solve a problem or achieve an objective**”

#2 “The Requirements Engineering Handbook”, Ralph Young, 2004

Principles and practices

Whatever language you choose to adopt, you should:

- Adapt it to the specifics of the business, organization, project...
 - “Fit-for-purpose” principle, rather than “One-size-fits” all
 - Practices (unique, specific) ≠ Principles (“global”, common)
 - “Best Practices” is meaningless, unless they are “Our Practices”
- Agree upon it by all project team members
 - Caution: project language (may be) ≠ sub-team language (e.g. Business Analysts vs Technical team)
 - Synonyms are excellent to reconcile different languages
- Use it consistently in all communication within/about the project

Definitions: Other terms

System = Process + People + Products

- System \neq IT System
- System $>$ Product
- Product = Tools?
- System = Solution?

Project & System/Products

- Product \neq Project

... and finally:

Requirement Types



What are they?

List known Requirement Types...



Requirement Types: some examples

- Business Requirements
 - Stakeholder Requirements
 - User Requirements
 - Customer Requirements
 - System Requirements
 - Process Requirements
 - Regulatory Requirements
 - Product Requirements
 - Quality Requirements
 - Data Requirements
 - Business Rules
 - Assumptions
 - Constraints
 - Technical Requirements
 - Design Requirements
 - Functional Requirements
 - Non-Functional Requirements
 - Scope
 - High-level Requirements
 - Detailed-level Requirements
 - Usability Requirements
 - Project Requirements
 - Documentation Requirements
- ... and the list can go on and on!



Emerged need:

Organize and simplify...

Solution: Categories & Criteria...

By the **target audience**:

- Stakeholder Requirements
- User Requirements
- Customer Requirements
- Regulatory Requirements

By **levels of details**:

- Scope-level Requirements
- High-level Requirements
- Detailed-level Requirements
- Project Requirements

Note:

Remember **Progressive Elaboration** (aka. Iterative and Incremental development)?

By the **domain**:

Business

- Business Requirements
- Business Rules

System/Product:

- Process Requirements
- Quality Requirements

Project:

- Assumptions
- Constraints
- Documentation Requirements

Technical:

- Data Requirements
- Design Requirements
- Functional Requirements
- Non-Functional Requirements
- Usability Requirements

... and, yet, the list **STILL** can go on and on!

Criteria: Multiple Perspectives...

	Scope-level	High-level	Detailed-level
Customers	<ul style="list-style-type: none">• Vision, Scope<ul style="list-style-type: none">– Project Charter– Assumptions	<ul style="list-style-type: none">• Business Reqs<ul style="list-style-type: none">– Process Model– Business Rules	System Reqs/model
Users	System Scope	<ul style="list-style-type: none">• User Reqs<ul style="list-style-type: none">– Use Cases diagrams	<ul style="list-style-type: none">• Use Cases• Activity Diagrams• Interfaces• Usability
Technical team	<ul style="list-style-type: none">• Product Scope<ul style="list-style-type: none">– Features	<ul style="list-style-type: none">• Functional Reqs• Non-functional Reqs• Architecture	<ul style="list-style-type: none">• System Use Cases• Design Reqs
Project team	<ul style="list-style-type: none">• Project Scope• Milestones• Constraints	<ul style="list-style-type: none">• WBS• Work Packages• Budget	<ul style="list-style-type: none">• WBS Dictionary• Schedule• Risks



Story: The blind men and the elephant, ...with a **twist!**

Short version of the story:

<http://www.britishcouncil.org/languageassistant-primary-tips-six-blind-men-and-the-elephant.htm>

Moral of the story:

To understand the Whole, you need to integrate all
Perspectives/parts

The **TWIST**:

It's not enough to integrate the perspectives,
you have to agree on what the perspective types are

Hint (analogies... remember those dreadful SAT questions?):

Requirements are to Requirement Types
like

Perspectives are to Perspective Types

So...

what's so important about that?

Multiple perspectives/criteria

BENEFITS:

- clarity of purpose
- clarity of language/communication
- effectiveness and efficiency of approach



Well... have you paid attention?

Frm BBK 2.0 (DRFT):

1. **cndtn r cpblty ndd by stkhldr t slv prblm r chv n bjctv.**
2. **cndtn r cpblty tht mst b mt r psssd by sltn r sltn cmpnnt t stsfy cntrct, stndrd, spcfctn, r thr frmly mpsd dcmnts.**
3. **dcmntd rprsnttn f cndtn r cpblty s n (1) r (2).**



Consider: Communication “styles”

“According to a research at Cambridge University, it doesn't matter in what order the letters in a word are, the only important thing is that the first and last letter be at the right place. The rest can be a total mess and you can still read it without problem. This is because the human mind does not read every letter by itself, but the word as a whole.”

Would you like to scramble your emails to **your** manager?

http://www.glassgiant.com/text_scrambler/

Industry examples...

- Borland Requirements Definition and Management (RDM) Solution
- EDS Requirements Determination Process (RDP)
- Zachman Framework

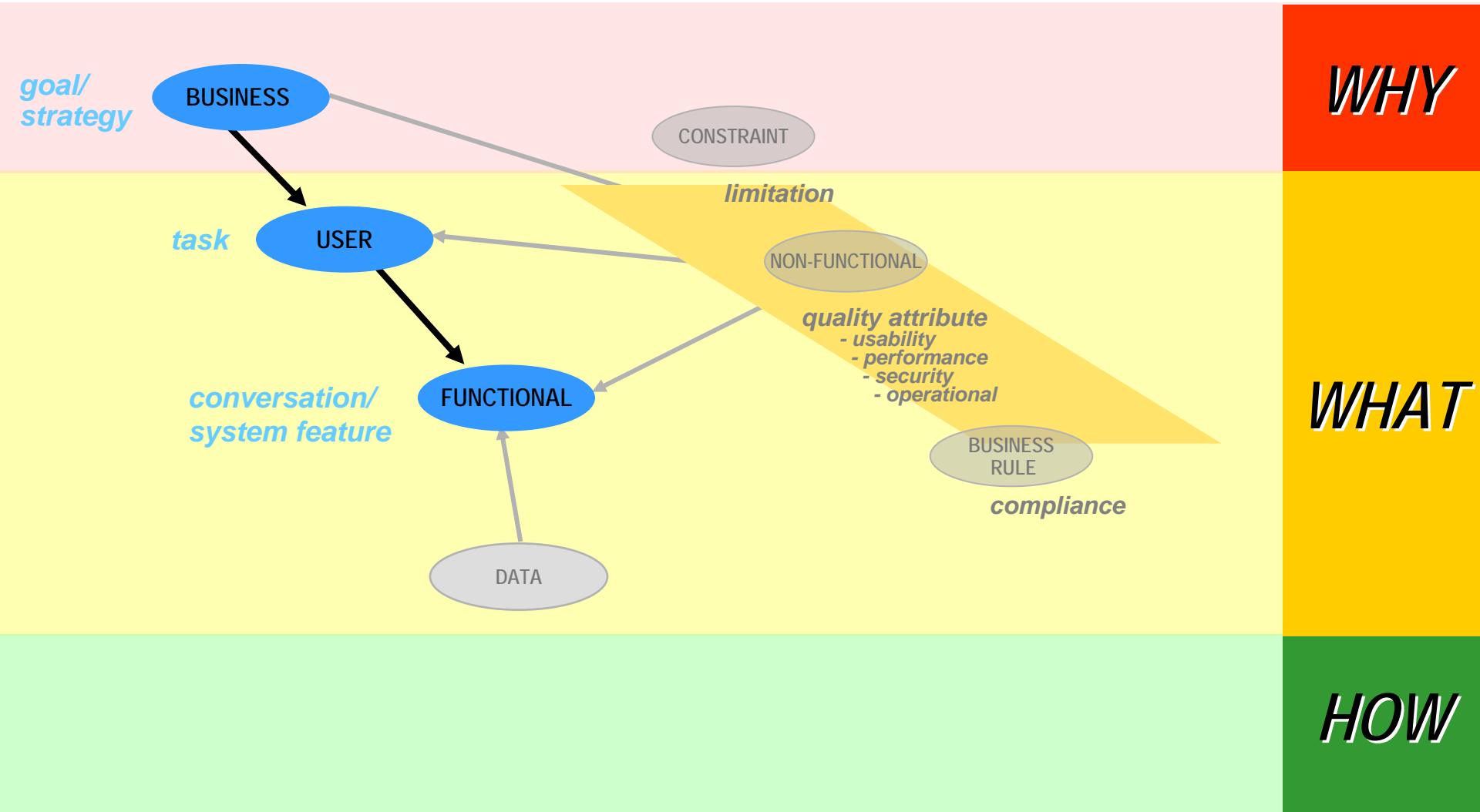
... and, of course:

- IIBA/BABOK:

**From Requirement Types
to Requirement Levels**



Borland: Requirement Structure



Adapted from Karl Wieggers, **Software Requirements**

Copyright © 2005 Borland Software Corporation. All rights reserved.

Borland:

Requirement Types Defined

Requirement Type	Definition
Business Requirement	A <u>business requirement</u> is a goal of the organization requesting the system.
User Requirement	A <u>user requirement</u> is a task that the user must be able to accomplish using the system.
Functional Requirement	A <u>functional requirement</u> is a conversation between the system and a user or another system requesting/providing information.
	A <u>functional requirement</u> is system feature that must be built into the system to satisfy the user requirements.
Constraint	A <u>constraint</u> is a limitation or restriction placed on the choices available to the project team for design and development of the system.
Non-Functional Requirement	A <u>non-functional requirement</u> is a quality attribute that the system must have. These attributes are <u>not</u> system features (functional requirements), but they do influence how the functionality of the system is implemented. Non-functional requirements usually deal with some aspect of <u>usability</u> , <u>performance</u> , or <u>security</u> or are <u>operational</u> in nature.
Business Rule	A <u>business rule</u> is a law, policy, standard or procedure by which an organization functions. It is a statement that defines or constrains some aspect of the business.
Data Requirement	A <u>data requirement</u> is information the system or user requests/provides to satisfy an interface requirement or functional requirement.

EDS:

Requirements Determination Process (RDP)

The EDS RDP methodology identifies the following phases when defining requirements for a release:



Source: <http://www.ottawa-outaouais.theiiba.org/events/>

EDS:

Requirements Determination Process (RDP)

Evolution of a Requirement– Example #2

Scope Statement

Business Objective: Increase repeat business by 25%.

High Level Requirements

Business Policy: Provide discounts to customers with high purchase volumes.

Glossary

Customer: A Customer is a person or organization who purchases products from ABC Company stores.
(*Business Rule Type: Term*)

Detailed Requirements

Functional & Non-Functional

The system shall calculate the Total Order Cost (*RULE001 Volume Discount*).

Business Rules

RULE001: Volume Discount

If a Customer's previous purchases exceed \$1,000 within the last 12 months, subtract 15% from the Total Order cost.

(*Business Rule Type: Action Enabler*)

For **lot more** examples and details visit the Ottawa IIBA Chapter website.

IIBA/BABOK 2.0 (DRAFT):

Requirement levels

1. Business Requirements
2. Stakeholder Requirements
3. Solution Requirements
 - Functional Requirements
 - Non-functional Requirements
 - Implementation Requirements



Practice it: Hands-on exercise...

Project: Develop a fountain-pen for left-handed people

Goals:

Using the Requirement Types matrix



1. Define the Scope
2. Determine the Stakeholders
3. Determine the Requirement Types
4. Determine the High-level Requirements
5. Fill in as much data as possible (as time allows) under each Requirement.

	Scope level	High level	Detailed level
Customers			
Users			
Technical team			
Project team			

References & additional reading

Professional Bodies of Knowledge: BABOK, PMBOK, SWEBOK

Books:

- Karl Wiegiers (2003): ***Software Requirements 2: Practical techniques for gathering and managing requirements throughout the product development cycle***, Redmond: Microsoft Press
- Ralph Young (2004): ***The Requirements Engineering Handbook***, Artech House
- Ian Alexander (2002): ***Writing Better Requirements***, Addison-Wesley Professional
- Elizabeth Hull, Ken Jackson, Jeremy Dick (2005): ***Requirements Engineering***, Springer

Internet:

- Borland RDM: <http://bdn1.borland.com/borcon2004/article/paper/0,1963,32193,00.html>
- EDS RDS: <http://www.ottawa-outaouais.theiiba.org/events/>
- Also, watch the RTP BizBuzz BLOG (<http://rtpbizbuzz.blogspot.com/>) for interesting discussions on related topics – some written by Razvan :-)

Acknowledgements

- **Betty Luedke**, Principal Consultant with Borland, for providing information and slides about the Borland RDM Solution
- **Laura Paton**, VP of Education for the RTP IIBA Chapter, for providing the link to the Ottawa IIBA Chapter presentation on EDS RDP
- **Anne Hartley, Sushma Ohri, Melissa Kempf**, and the whole **BlueCross and BlueShield of NC**, for hosting this event
- **All of you**, for taking the time to attend this presentation and for giving me the opportunity to share my thoughts with you.

To all, my most sincere

THANKS,

Razvan :-)

razvan.radulian@why-what-how.com

I hope to see you again at our next chapter meeting:

- “Business Rules”, September 2008
...and at the my presentation at the
- PM/BA World Congress, November 2008