

Get Your Users "IN" the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

Missed Requirements, Again?



Get Your Users "IN" the Doghouse and Keep Yourself Out:
Eliciting **Nonfunctional Requirements**


Session Learning Objectives What to Expect

		
Define nonfunctional requirements	Understand these vital, yet overlooked requirements	Classify common nonfunctional requirements

 bringing your business into focus® Understanding Nonfunctional Requirements - 2

Get Your Users “IN” the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

Nonfunctional Definition #1






Objective #1

Define nonfunctional requirements

bringing your business into focus® **Understanding Nonfunctional Requirements - 3**

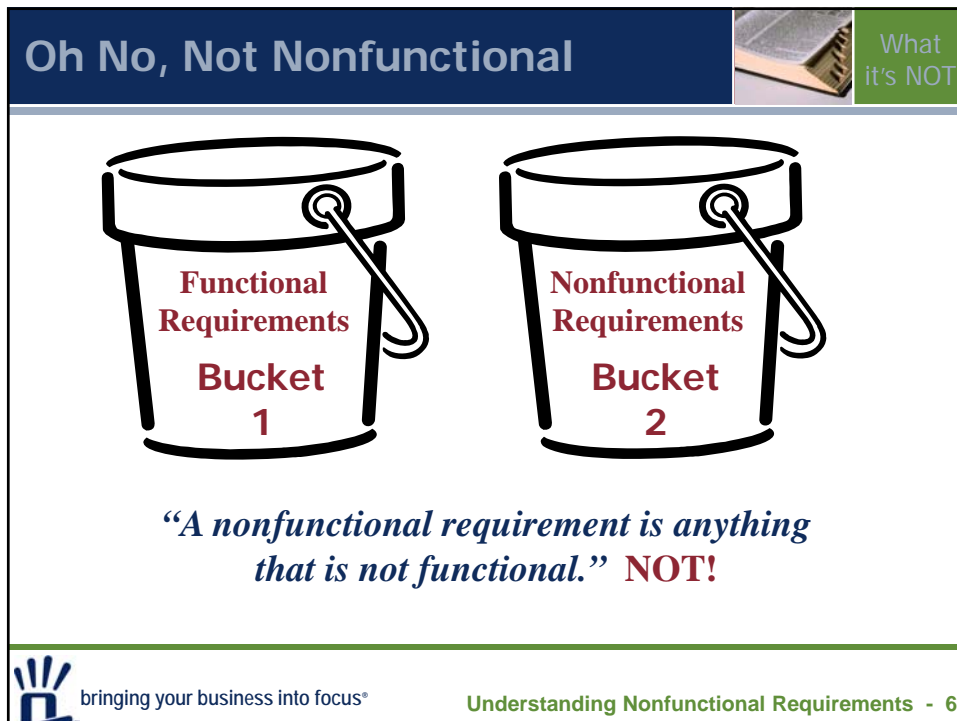
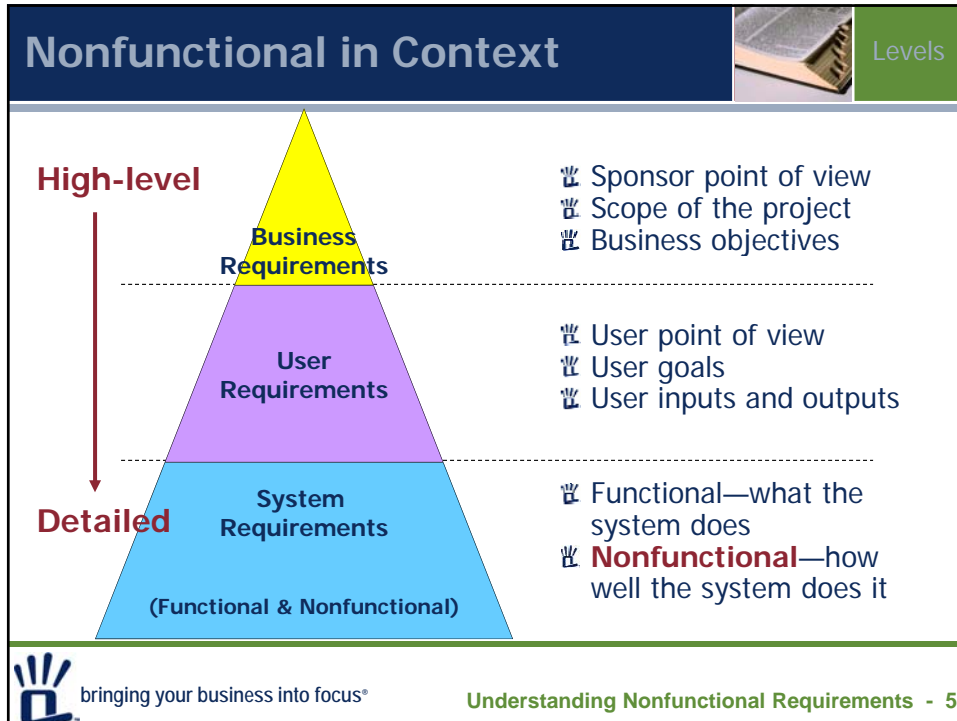
Levels of Requirements Levels



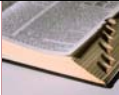
Business Level	Defines project scope. Identifies business benefits. Establishes high-level objectives.
	
User Level	Focuses on business processes. Names user roles and user goals. Identifies flow of information and materials.
	
System Level	Identifies processing of inputs and outputs. Defines functions and features. Describes quality attributes of the system environment.


bringing your business into focus® **Understanding Nonfunctional Requirements - 4**

Get Your Users “IN” the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

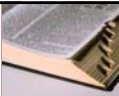







Get Your Users “IN” the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

A Handful of Definitions	 Example
<p>“<i>Nonfunctional requirement</i> – in software system engineering, a software requirement that describes not what the software will do, but how the software will do it, for example, software performance requirements, software external interface requirements, software design constraints, and software quality attributes. Nonfunctional requirements are difficult to test; therefore, they are usually evaluated subjectively.” [Thayer2000]</p>	
<p>“<i>Non-functional requirements</i> – These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process and standards. Non-functional requirements often apply to the system as a whole. They do not usually just apply to individual system features or services.” [Sommerville2007]</p>	
<p>“<i>Non-functional requirement</i> – A property, or quality, that the product must have, such as an appearance, or a speed or accuracy property.” [Robertson2003]</p>	
<p>“<i>Nonfunctional requirement</i> – A description of a property or characteristic that a software system must exhibit or a constraint that it must respect, other than an observable system behavior.” [Wieggers2003]</p>	
<p>“<i>Nonbehavioral requirements</i> – Requirements that describe the required overall attributes of the system, including portability, reliability, efficiency, human engineering, testability, understandability, and modifiability.” [Davis1993]</p>	

 bringing your business into focus®

Understanding Nonfunctional Requirements - 7

What Does It Really Mean?	 What's left?
<p>Verbiage Removed:</p>	<p>...how the software will do it.</p>
<p> General wordiness.</p>	<p>...constraints on the services or functions offered by the system.</p>
<p> Nonfunctional “examples.”</p>	<p>...property, or quality, that the product must have.</p>
<p> “Descriptive” information that doesn’t add value.</p>	<p>...property that a software system must exhibit or a constraint that it must respect.</p>
<p> Inclusion or reference to “functional requirements.”</p>	<p>...required overall attributes of the system.</p>

 bringing your business into focus®

Understanding Nonfunctional Requirements - 8

Get Your Users “IN” the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

A Simplified Definition



How
Well

Nonfunctional Requirement

—

a specification of *how well*
a software *system*
must function.



bringing your business into focus®

Understanding Nonfunctional Requirements - 9

Functional versus Nonfunctional



Differ

- | | |
|--|---|
| F Product features. | N Product properties. |
| F Describe the work that is done. | N Describe the character of the work. |
| F Describe the actions with which the work is concerned. | N Describe the experience of the user while doing the work. |
| F Characterized by verbs (active). | N Characterized by adjectives. |




bringing your business into focus®

Understanding Nonfunctional Requirements - 10

Get Your Users "IN" the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

Vital, Yet Why Overlooked? #2

Objective #2



Understand why nonfunctional requirements are vital, yet overlooked

bringing your business into focus® Understanding Nonfunctional Requirements - 11



Importance of Nonfunctional Need Both



All functional requirements may be satisfied, but if nonfunctional requirements are overlooked, the system will fail.



bringing your business into focus® Understanding Nonfunctional Requirements - 12

Get Your Users “IN” the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

Nonfunctional Challenges		Don't Agree
<ul style="list-style-type: none"><li data-bbox="402 380 1023 422">❏ No agreed upon formal definition.<li data-bbox="402 485 727 527">❏ No complete list.<li data-bbox="402 590 1201 724">❏ No single universal classification scheme, accommodating all the needs of different application domains under different situations.<li data-bbox="402 787 1079 871">❏ Different people and organizations use different terminologies.		
 bringing your business into focus®	Understanding Nonfunctional Requirements - 13	

Alternative Names & Spelling		Many Names
<ul style="list-style-type: none"><li data-bbox="402 1249 771 1291">❏ Quality Attributes.<li data-bbox="402 1344 820 1386">❏ Quality Requirements.<li data-bbox="402 1438 966 1480">❏ Nonbehavioral Requirements.<li data-bbox="402 1543 722 1585">❏ “ilities” or “ities.”<li data-bbox="402 1638 998 1722">❏ Spelled with or without a hyphen.<ul style="list-style-type: none"><li data-bbox="446 1680 974 1722">⚠ Non-functional or Nonfunctional		
 bringing your business into focus®	Understanding Nonfunctional Requirements - 14	

Get Your Users “IN” the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

Nonfunctional Complexity

Complex Nature

- Subjective**
 - Viewed, interpreted and evaluated differently by different people. Statements are often too brief and vague.
- Relative**
 - Interpretation and importance may vary depending on the particular system and organizational needs.
- Integrated**
 - Attempts to achieve one requirement can hurt or help the achievement of other requirements due to the nature of their global impact.


bringing your business into focus® Understanding Nonfunctional Requirements - 15

Nonfunctional Classification

#3

Objective #3

Making sense of nonfunctional requirements with a user-focused classification



bringing your business into focus® Understanding Nonfunctional Requirements - 16

Get Your Users "IN" the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

Overwhelmability?



Scary List?


Access Control	Controllability	Human Engineering	Portability	Space Boundedness
Accessibility	Coordination Cost	Independence	Precision	Space Performance
Accountability	Coordination Time	Informativeness	Predictability	Specificity
Accuracy	Correctness	Integrity	Process Mgmt. Time	Stability
Adaptability	Coupling	Internal Consistency	Productivity	Subjectivity
Additivity	Customer Evaluation	Interoperability	Project Stability	Supportability
Adjustability	Customer Loyalty	Intuitiveness	Project Tracking Cost	Surety
Affordability	Customizability	Learnability	Promptness	Survivability
Agility	Data Space*	Leveragability	Quality	Susceptibility
Auditability	Decomposability	Main Memory*	Reconfigurability	Sustainability
Augmentability	Degradation of Service	Maintainability	Recoverability	Tankness
Autonomy	Dependability	Maintenance Cost	Recovery	Testability
Availability	Development Cost	Maintenance Time	Reliability	Throughput
Buffer Space*	Development Time	Maturity	Repeatability	Timeliness
Capability	Distributivity	Mean Performance	Replaceability	Traceability
Capacity	Diversity	Measurability	Replicability	Trainability
Clarity	Ease of use	Mobility	Response Time	Transferability
Code Space*	Efficiency	Modifiability	Responsiveness	Transparency
Cohesiveness	Elasticity	Modularity	Retirement Cost	Understandability
Commonality	Enhanceability	Naturalness	Reusability	Uniform*
Compatibility	Evolvability	Nomadicity	Risk Analysis Cost	Uniformity
Completeness	Execution Cost	Observability	Risk Analysis Time	Usability
Comprehensibility	Extensibility	Off-Peak Period*	Robustness	User-Friendliness
Conceptuality	External Consistency	Operability	Safety	Validity
Conciseness	Fault Tolerance	Operating Cost	Scalability	Variability
Confidentiality	Feasibility	Peak Period*	Security	Verifiability
Configurability	Flexibility	Performability	Sensitivity	Versatility
Connectivity	Formality	Planning Cost	Similarity	Visibility
Consistency	Generality	Planning Time	Simplicity	*Performance



bringing your business into focus®

Understanding Nonfunctional Requirements - 17

Quality Attributes Tree




1976

```

    graph LR
      GU[General Utility] --> PU[Portability]
      GU --> AU[As-Is Utility]
      GU --> MA[Maintainability]
      AU --> RE[Reliability]
      AU --> EF[Efficiency]
      AU --> HE[Human Engineering]
      MA --> TE[Testability]
      MA --> UN[Understandability]
      MA --> MO[Modifiability]
      PU --> DI[Device-independence]
      PU --> SC[Self-containedness]
      RE --> ACC[Accuracy]
      RE --> COM[Completeness]
      RE --> RI[Robustness / Integrity]
      RE --> CON[Consistency]
      EF --> ACCO[Accountability]
      EF --> DE[Device Efficiency]
      HE --> ACCO
      HE --> DE
      HE --> ACC[Accessibility]
      TE --> COM
      TE --> SD[Self-descriptiveness]
      UN --> COM
      UN --> SD
      UN --> STR[Structuredness]
      UN --> CON
      UN --> LEG[Legibility]
      MO --> AUG[Augmentability]
  
```

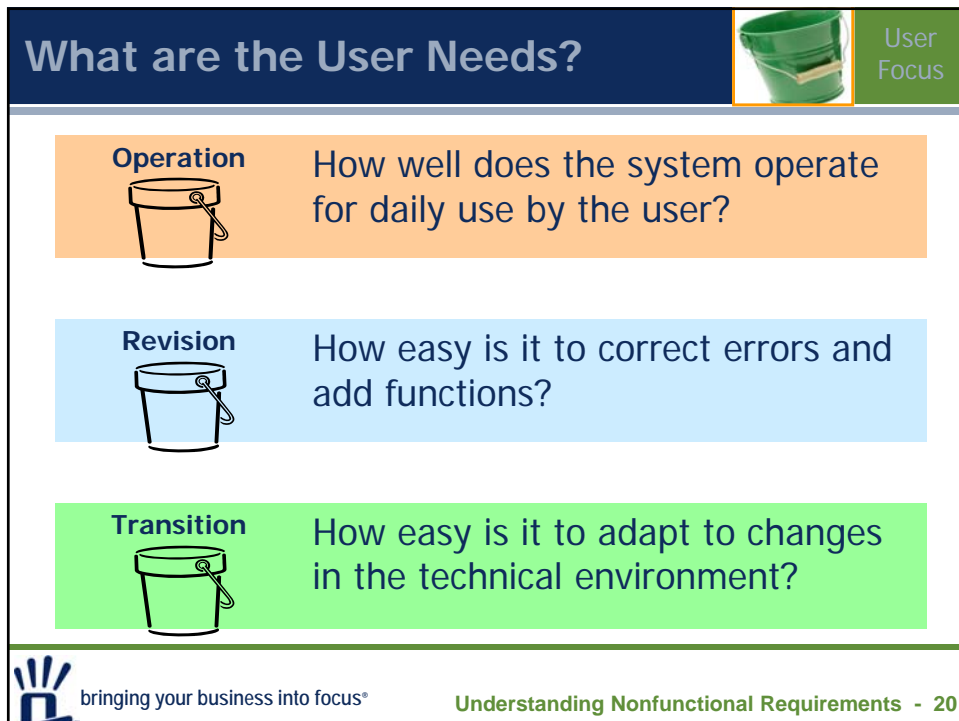
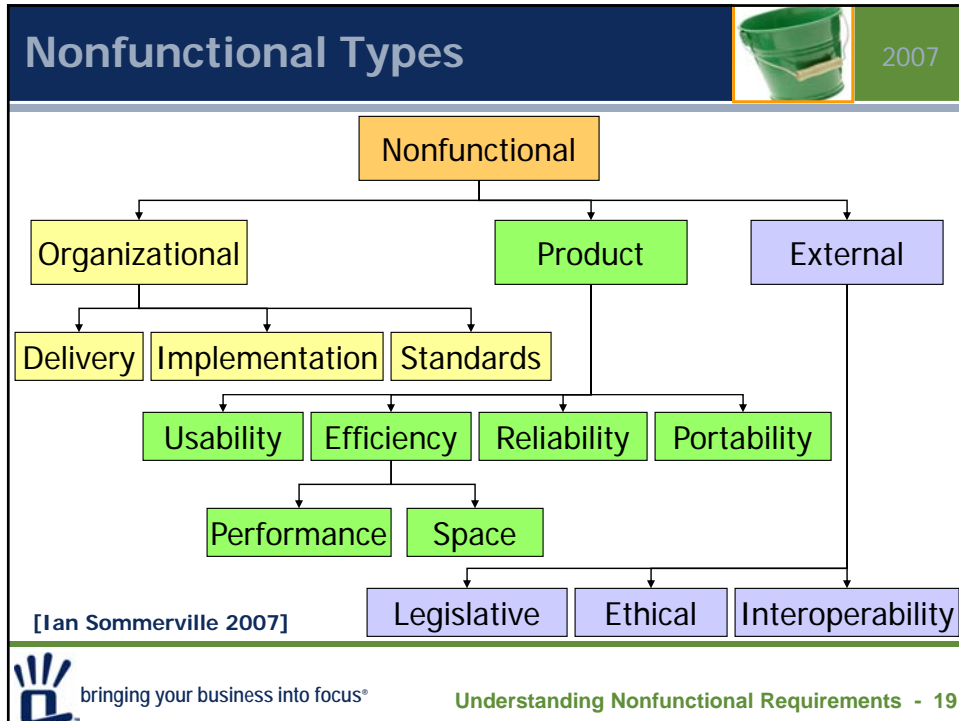
[Barry Boehm 1976]




bringing your business into focus®


Understanding Nonfunctional Requirements - 18

Get Your Users "IN" the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements



Get Your Users “IN” the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

A User-Focused Classification		 User Focus
	User Concern	Nonfunctional
Operation	How well is it safeguarded against unauthorized access?	Access Security
	How dependable is it during normal operating times?	Availability
	How fast, how many, and how well does it respond?	Efficiency
	How accurate and authentic is the data?	Integrity
	How immune is the system to failure?	Reliability
	How resilient is the system from failure?	Survivability
	How easy is it to learn and operate the system?	Usability
Revision	How easy is it to modify to work in different environments?	Flexibility
	How easy is it to upkeep and repair?	Maintainability
	How easy is it to expand or upgrade its capabilities?	Scalability
	How easy is it to show it performs its functions?	Verifiability
Transition	How easy is it to interface with another system?	Interoperability
	How easy is it to transport?	Portability
	How easy is it to convert for use in another system?	Reusability

Operation: Access Security 


Definition — the extent to which the system is safeguarded against deliberate and intrusive faults from internal and external sources.


User concern

- ☞ How well is the system safeguarded against unauthorized access?

Example:

Employees shall be forced to change their password the next time they log in if they have not changed it within the length of time established as “password expiration duration.”




bringing your business into focus®
Understanding Nonfunctional Requirements - 22

Get Your Users “IN” the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

Operation: Availability



Definition — the degree to which users can depend on the system to be up (able to function) during “normal operating times.”

User concern

- How dependable is the system during normal operating times?



Example:

The Automated Teller Machine shall be at least 99.5 percent available on weekdays between 6:00 a.m. and midnight local time. The machine shall be at least 99.95 percent available on weekdays between 3:00 p.m. and 7:00 p.m. local time.



bringing your business into focus®

Understanding Nonfunctional Requirements - 23

Operation: Efficiency



Definition — the extent to which the software system handles capacity, throughput, and response time.

User concern

- How fast does it process (capacity)?
- How many at a time (throughput)?
- Response time?



Example:

The system must download the new rate parameters within ten minutes of a non-scheduled rate change.



bringing your business into focus®

Understanding Nonfunctional Requirements - 24

Get Your Users “IN” the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

Operation: Integrity



Definition — the degree to which the data maintained by the software system is accurate, authentic, and without corruption.

User concern

How accurate and authentic is the data?

Example:

The integrity of the system data area must be checked by the internal audit system twice per second; if inconsistencies in the data are detected, the system operation should be disabled.



bringing your business into focus®

Understanding Nonfunctional Requirements - 25

Operation: Reliability



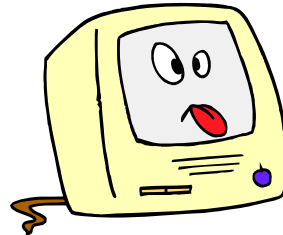
Definition — the extent to which the software system consistently performs the specified functions without failure.

User concern

How immune is the system to failure?

Example:

No more than 10 claim assignments out of 5000 can be “unassigned” because of system failures.



bringing your business into focus®

Understanding Nonfunctional Requirements - 26

Get Your Users “IN” the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

Operation: Survivability



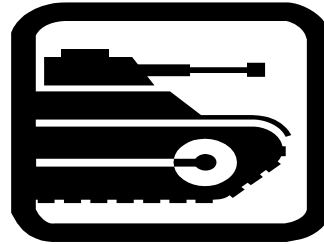
Definition — the extent to which the software system continues to function and recovers in the presence of a system failure.

User concern

How resilient is the system from failure?

Example:

All policy statement parameters shall have default values specified, which the Report Writer system shall use if a parameter's input data is missing or invalid.



bringing your business into focus®

Understanding Nonfunctional Requirements - 27

Operation: Usability



Definition — the ease in which the user is able to learn, operate, prepare inputs and interpret outputs through interaction with a system.

User concern

How easy is it to learn and operate the system?

Example:



A trained order-entry clerk shall be able to submit a complete order for a product chosen from a supplier catalog in a maximum of 7 minutes, and an average order entry time of 4 minutes.



bringing your business into focus®

Understanding Nonfunctional Requirements - 28

Get Your Users “IN” the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

Revision: Flexibility



Definition — the ease in which the software can be modified to adapt to different environments.

User concern

How easy is it to change and add new features?

Example:

It shall be possible to add a new delivery option for customer mailing method by developing and “plugging in” the functionality necessary to support that delivery option.



bringing your business into focus®

Understanding Nonfunctional Requirements - 29

Revision: Maintainability



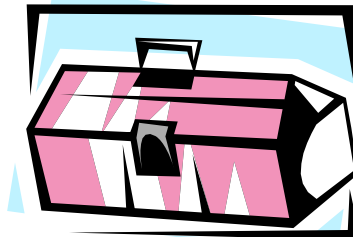
Definition — the ease in finding and fixing faults in the software system.

User concern

How easy is it to upkeep and repair the system?

Example:

The application development process must have a regression test procedure that allows complete re-testing within 2 business days.



bringing your business into focus®

Understanding Nonfunctional Requirements - 30

Get Your Users “IN” the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

Revision: Scalability



Definition — the degree in which the software system is able to expand its processing capabilities upward and outward to support business growth.

User concern

- How easy is it to expand or upgrade the capabilities of the system?



Example:

Any increase in the number of customers shall not degrade system availability to an extent noticeable by any users.



bringing your business into focus®

Understanding Nonfunctional Requirements - 31

Revision: Verifiability



Definition — the extent to which tests, analysis, and demonstrations are needed to prove that the software system will function as intended.

User concern

- How easy is it to show it performs its functions?

Example:

The maximum number of test cases to cover testing of any particular source code module shall be 20.



bringing your business into focus®

Understanding Nonfunctional Requirements - 32

Get Your Users “IN” the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

Transition: Interoperability



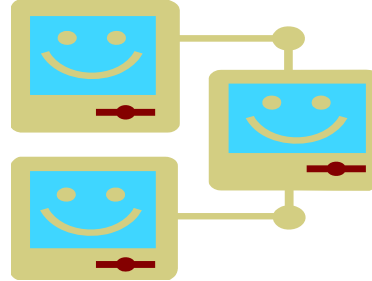
Definition — the extent to which the software system is able to couple or facilitate the interface with other systems.

User concern

How easy is it to interface with another system?

Example:

The system must be able to interface with any HTML (HyperText Markup Language) browser.



bringing your business into focus®

Understanding Nonfunctional Requirements - 33

Transition: Portability



Definition — the ease in which a software system from its current hardware or software environment can be transferred to another environment.

User concern

How easy is it to transport?

Example:

The system is designed to run in business offices, but the intent is to have a version which will run in manufacturing assembly plants.



bringing your business into focus®

Understanding Nonfunctional Requirements - 34

Get Your Users “IN” the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

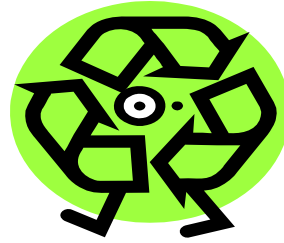
Transition: Reusability



Definition — the extent to which a portion of the software system can be converted for use in another.

User concern

How easy is it to convert for use in another system?



Example:

The payment subsystem design is based on the payment module from the ALPHA product line. The ePAYZ system should not be modified unless absolutely necessary.



bringing your business into focus®

Understanding Nonfunctional Requirements - 35

Session Learning Objectives

Recap



Define nonfunctional requirements



Understand these vital, yet overlooked requirements



Classify common nonfunctional requirements



bringing your business into focus®

Understanding Nonfunctional Requirements - 36


Get Your Users “IN” the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

A Requirement “Pattern”

Useful Tool

- Basic details
- Applicability
- Discussion
- Content
- Template(s)
- Example(s)
- Extra requirements
- Development considerations
- Testing considerations

**37 patterns!
Over 400
real-life
examples!**



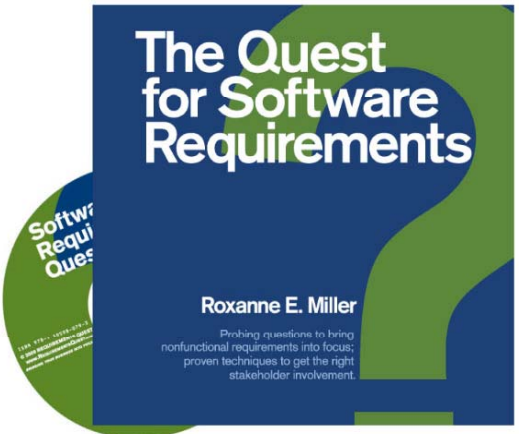
Stephen Withall, Microsoft Press, 2007.

bringing your business into focus® **Understanding Nonfunctional Requirements - 37**

Over 2,000 Suggested Questions

Useful Tool

- Book:
The Quest for Software Requirements
- CD (MS Excel):
Software Requirements Questions



Order Yours Today! www.RequirementsQuest.com

bringing your business into focus® **Understanding Nonfunctional Requirements - 38**


Get Your Users "IN" the Doghouse and Keep Yourself Out: Eliciting Nonfunctional Requirements

JOIN THE WISCONSIN CHAPTERS OF THE IIBA® FOR OUR
BUSINESS ANALYST DEVELOPMENT DAY 2010!

WHEN
Tuesday – October 5th

WHERE
Alliant Energy Center
Madison, Wisconsin


WHO
Business Analysts, Systems
Analysts, Quality Assurance,
Business Analyst Managers,
and Project Managers



WHY

2 Keynote Speakers	30+ Sponsors
4 Symposium Tracks	500+ Attendees
1 Technology Track	Great Food & Prizes

<http://www.wibadd.org>



Questions Thank You



IIBA™ INDUSTRY Sponsor

IIBA Endorsed Education Providers
Charter Member

Presented by: Roxanne E. Miller, CBAP
www.RequirementsQuest.com