Non-Functional Requirements
How to Get Them in Shape

Dan Bergh Johnsson
Omegapoint Consulting AB, Sweden
CSS 2006
Ambition

- Share some ideas on requirements
- Insights gained
- Tools to use
- Courage to question the next req spec

- Disclaimer: No code
Agenda

- What are NFRs
- Requirement Document as Contract
- SMART Requirements
- Sharpen up requirements
- NFR vs FR – Who’s Who?
- Architecture and NFR trade-offs
- Making Priorities
Agenda

- What are NFRs
- Requirement Document as Contract
- SMART Requirements
- Sharpen up requirements
- NFR vs FR – Who’s Who?
- Architecture and NFR trade-offs
- Making Priorities
Functional Requirements

- A customer can order drinks, either paid in cash or put on tab for later payment
- Price of drinks are computed from a price list
- A customer can only have one open tab at a time
- When opening a tab, a credit card must be presented
- Each tab has a credit limit, computed from the payment history of the customer

Will break down into lots of system functionality
# The Cube – 2D

<table>
<thead>
<tr>
<th>check-qty.js</th>
<th>orders.jsp</th>
<th>Tab</th>
<th>SELECT ... UPDATE ...</th>
<th>payments</th>
</tr>
</thead>
<tbody>
<tr>
<td>JavaScript</td>
<td>Servlet 2.3</td>
<td>EJB 2.1</td>
<td>JDBC 2.0</td>
<td>SQL 92</td>
</tr>
<tr>
<td>Mozilla</td>
<td>Tomcat 4</td>
<td>Geronimo</td>
<td>Connector/J</td>
<td>MySQL</td>
</tr>
<tr>
<td>OpenBSD</td>
<td>BeOS</td>
<td>Linux</td>
<td>Linux</td>
<td>Linux</td>
</tr>
</tbody>
</table>
The Cube – Capabilities

- Emergent properties
  - Performance
  - Availability
  - Security
  - Extensibility

- Not located in single component
- Discussing architecture – must consider entire system
NFRs and Capabilities

- Requirements that are non-functional?
- Non-functional requirement is often requirement on capabilities
- Can be other constraints
  - Implemented in Java
  - Run on IBM
Technical System Structure

- Given functionality requirements
- How to structure them?
  - Lots of ways (≈ O(2^2^n))
  - Infeasible to try them all
- Creative task
  - Takes knowledge of technology
NFRs – What For?

- FR drives development
  - Development cycle
  - Increments

- NFR?
Architecture

- Technical structure of system

- *Structure functionality into components in such a way that all NFRs are fulfilled*

- Requires deep technical knowledge of component technologies

- BTW: Questionable metaphor
Requirement Gathering

- “Gathering” misleading
- Will not give what is really wanted
  - Asking is not enough
  - Takes effort
- Requirement Digging/Mining
  - Apply pressure
  - Press ‘til they squeak
- Requirement Hunting?

Push customer out of comfort zone
Requirement Gardening

- Revise after each release/increment
  - System context change
  - Follow change or become obsolete
- After deploy/production
  - Keep revising
- More like gardening than gathering/mining
  - Small garden, but neat
  - Keep most important, and culture
Agenda

- What are NFRs
- Requirement Document as Contract
- SMART Requirements
- Sharpen up requirements
- NFR vs FR – Who’s Who?
- Architecture and NFR trade-offs
- Making Priorities
Requirements Document – A Contract

- Binding
- Legally
- Morally

Your professional honour!

What would you sign?
What Would I Sign?

- Realistic
  - Can be done
  - By me

- Verifiable
  - Success criteria
  - Measure

- Mutually Agreed
  - Best of interest for both parties
What About the Rest

- Wish lists are valuable
  - But not in a contract
- Out of contract... into
  - Erasure
  - “Design considerations”
  - Vision document
  - Change Cases / Request Proposals
Contract, Is That Agile?

Agile Manifesto:
- Customer Collaboration over Contract Negotiations

Agile Way (cfr how FR are handled)
- Specify as you go
  - Don’t spend entire salary first Friday
- Avoid premature exactness (specific ≠ exact)
  - “750 ms” or “three-quarter of a second”
  - Clarify when needed
Agenda

- What are NFRs
- Requirement Document as Contract
- *SMART Requirements*
- Sharpen up requirements
- NFR vs FR – Who’s Who?
- Architecture and NFR trade-offs
- Making Priorities
SMART

- **Specific**
- **Measurable**
- **Attainable**
- **Realisable**
- **Traceable**

ACM SIGSOFT 1995 Mannion Keepence, “SMART requirements”

[http://doi.acm.org/10.1145/224155.224157](http://doi.acm.org/10.1145/224155.224157)
SMART – Variations

- **Specific**
- **Measurable**
- **Attainable / Agreed** / **Appropriate**
- **Realisable / Relevant** / **Realistic** / **Results-oriented**
- **Traceable** / **Testable** / **Time-bound** / **Timely**

- (Sustainable Model for Arctic Regional Tourism)
Agenda

- What are NFRs
- Requirement Document as Contract
- SMART Requirements
- *Sharpen up requirements*
- NFR vs FR – Who’s Who?
- Architecture and NFR trade-offs
- Making Priorities
### Some Common NFRs

<table>
<thead>
<tr>
<th>Performance</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td></td>
</tr>
<tr>
<td>Scalability</td>
<td></td>
</tr>
<tr>
<td>Availability</td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td></td>
</tr>
<tr>
<td>Maintainability</td>
<td></td>
</tr>
<tr>
<td>Extensibility</td>
<td></td>
</tr>
<tr>
<td>Manageability</td>
<td></td>
</tr>
<tr>
<td>Flexibility</td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td></td>
</tr>
<tr>
<td>Testability</td>
<td></td>
</tr>
<tr>
<td>Usability</td>
<td></td>
</tr>
</tbody>
</table>
Dealing with Vague Req

- System should be fast
  - What do you mean “fast”? Ahh, response time
  - How fast? 3 s
- Must handle 10,000 simultaneous users
  - 10,000 registered? Logged on? Concurrent requests?
- Easy to switch to Linux
  - Invent measure
- Remember SpecificMeasurableART
  - Few specific and incomplete...
  - ... rather than...
  - ... many vague and complete
# NFRs – A List

<table>
<thead>
<tr>
<th>Category</th>
<th>Requirement</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>response time</td>
<td>3s</td>
</tr>
<tr>
<td>Capacity</td>
<td>Broadness/throughput</td>
<td>100 simultaneous requests</td>
</tr>
<tr>
<td>Scalability</td>
<td>potential capacity</td>
<td>?</td>
</tr>
<tr>
<td>Availability</td>
<td>responding</td>
<td>99 %, 20 min / day</td>
</tr>
<tr>
<td>Reliability</td>
<td>behave well</td>
<td>1 fail session per...</td>
</tr>
<tr>
<td>Maintainability</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>Extensibility</td>
<td>can be changed</td>
<td>change cases, unit tests</td>
</tr>
<tr>
<td>Manageability</td>
<td>keep running / CoO</td>
<td>10 admin-h/month</td>
</tr>
<tr>
<td>Flexibility</td>
<td>modify deploy</td>
<td>change cases</td>
</tr>
<tr>
<td>Security</td>
<td>privacy, integrity, audit</td>
<td>can only access...</td>
</tr>
<tr>
<td>Testability</td>
<td>possible to test</td>
<td>automated acceptance tests</td>
</tr>
<tr>
<td>Usability</td>
<td>ease of use</td>
<td>70 % do w/o help</td>
</tr>
</tbody>
</table>
Agenda

- What are NFRs
- Requirement Document as Contract
- SMART Requirements
- Sharpen up requirements
- *NFR vs FR – Who’s Who?*
- Architecture and NFR trade-offs
- Making Priorities
Req in Fake Format

- Things are not always what they seem!
- "Ease of use" – NFR?
  - 3-digit codes
  - Cottage booking not more than two clicks away
  - NFR -> FR
- "Should be login screen"
  - Strictly FR – but not intent
  - Intent "should not be possible to access…"
  - NFR Security
  - FR -> NFR

Press until they squeak – then listen to that squeak
Agenda

- What are NFRs
- Requirement Document as Contract
- SMART Requirements
- Sharpen up requirements
- NFR vs FR – Who’s Who?
- Architecture and NFR trade-offs
- Making Priorities
“Improving” Capabilities

- Increase performance
  - Optimize code
- Better security
  - Encrypted communication
- Better availability
  - Clustered application servers
- Higher flexibility
  - Only use standard JBDC
Architectural Transformations

- Increase performance – Optimize code
  - Hard-to-read code: lower extensibility
- Better security – Encrypted communication
  - Takes CPU: lower performance
  - HTTPS not poolable: lower capacity
- Better availability – Clustered application servers
  - More machines: lower manageability
- Higher flexibility – Only use standard JBDC
  - Restricted use of features: lower performance, extensibility
Trade-Offs

- Knowing/explaining trade-offs is architect’s job
- Trade-offs depend on technology
- Takes deep technical knowledge

Architectural transformations are trade-offs
Agenda

- What are NFRs
- Requirement Document as Contract
- SMART Requirements
- Sharpen up requirements
- NFR vs FR – Who’s Who?
- Architecture and NFR trade-offs
- Making Priorities
Good Transformations

- Should I use transactions? Are they Good?
- Trade-off
  - Reliability
  - Capacity
- Which is better?
- Which have you in surplus?
- Trade surplus against shortage
  - Pay expensive currency using cheap?
Prioritise Capabilities

- Not a technical decision
- Impossible to rank “as such”
  - capability reliability vs capability capacity
- Must be quantified
  - NFR vs NFR
- Lose 1 session of 1000 vs max 500 users

- You must assist in decision!
Understanding NFR Origin

- NFRs drive architecture
- What drives NFRs?
Ebay vs Bank of England

- Ebay auctions
- Do not use transactions
- Lose a house a day
- Why not use transactions?

- Bank
- Do use transactions
- Might have capacity problems
- Why not skip transactions?
Business Risks

E-bay
- Marketbuzz
- What if nobody is there?
- What if people are unable to get in?

Bank of England
- Trustworthiness
- What if something goes wrong?
- What if we lose a retirement fund?

Squeak, squeak
Leaving comfort zone!
Origin of NFRs

Question to ask:

*What are the most severe business risks?*

Functional requirements address *business value*

Non-functional requirements address *business risk*
Agenda

- What are NFRs
- Requirement Document as Contract
- SMART Requirements
- Sharpen up requirements
- NFR vs FR – Who’s Who?
- Architecture and NFR trade-offs
- Making Priorities
Remember

- Architecture is a Technical Trait
- Requirements take more than Gathering
  - Requirement Digging
  - Requirement Gardening
- Req Doc is a Contract, but do it Agile
- SMART
- All transformations are trade-offs
- NFR priorities reflect business risk
Comments? Reflections?