Requirements Analysis

Angela Dappert
Digital Preservation Coalition
DPC Briefing Day, 13 December 2013
I'll need to know your requirements before I start to design the software.

First of all, what are you trying to accomplish?

I'm trying to make you design my software.

I mean what are you trying to accomplish with the software?

I won't know what I can accomplish until you tell me what the software can do.

Try to get this concept through your thick skull: the software can do whatever I design it to do!

Can you design it to tell you my requirements?
AQR16
Acquire [Highly Desirable] Must handle ingest of digital material where the smallest level of granularity in the submission is larger than the DOM System level of ingest granularity (article). e.g. a journal issue is one PDF with several articles in it

AQR17
Acquire [Highly Desirable] The system must keep and generate reports on failed and unsolicited submissions.

AQR18
Acquire [Highly Desirable] Must notify information provider of satisfactory transfer of data, if this is required in the information provider profile.
In FTP pull scenarios providers may require return notification of satisfactory transfer of data to allow them to manage disk space on their ftp site.

4.12 Pre-Ingest Module Requirements

<table>
<thead>
<tr>
<th>Num</th>
<th>Category</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIR1</td>
<td>Pre-Ingest</td>
<td>Must support batch submission to the pre-ingest module.</td>
</tr>
<tr>
<td></td>
<td>[Highly Desirable]</td>
<td></td>
</tr>
<tr>
<td>PIR2</td>
<td>Pre-Ingest</td>
<td>Must support manual resubmission to the pre-ingest module in case of manual intervention.</td>
</tr>
<tr>
<td></td>
<td>[Highly Desirable]</td>
<td></td>
</tr>
<tr>
<td>PIR3</td>
<td>Pre-Ingest</td>
<td>Must construct ingest list of successfully acquired submissions and schedule batch processing.</td>
</tr>
<tr>
<td></td>
<td>[Highly Desirable]</td>
<td></td>
</tr>
<tr>
<td>PIR4</td>
<td>Pre-Ingest</td>
<td>Must create new object instances for every physical and logical object in the traversal of the submission.</td>
</tr>
<tr>
<td></td>
<td>[Highly Desirable]</td>
<td></td>
</tr>
</tbody>
</table>
When

• For software system specification
• For any specification
• For Digital Preservation or any other domain
• For in-house development, tender, COTS
  • Separate but related decision – what vs. how
What for

- Identify needed attributes, functions, characteristics, quality
- To achieve value for stakeholders
- Based on business needs
- Guided by policies
Why

- Understand the system
- Communicate the function of the system
- Identify conflicting interests

Reduce the development effort
Why

• Understand the system
• Communicate the function of the system
• Identify conflicting interests
• **Measure tendering proposals against it**

*Customer-supplier agreement: what the software product is to do*

*Provide a basis for estimating costs and schedules.*
Why

• Understand the system
• Communicate the function of the system
• Identify conflicting interests
• Measure tendering proposals against it
• Feed into the design stage of product development

Also:
• Facilitate transfer.
• Serve as a basis for enhancement.
Why

• Understand the system
• Communicate the function of the system
• Identify conflicting interests
• Measure tendering proposals against it
• Feed into the design stage of product development

• Test software against it

Provide a baseline for validation and verification
Use determines form

Requirement as **basis for tendering**
must not exclude credible options
must be open
-> **high-level** abstract statement

Requirement as **basis for the software contract**
and for testing
must define detail
-> **detailed** specification
## What

<table>
<thead>
<tr>
<th>Business Requirements</th>
<th>Goals, objectives, needs, opportunities, problems</th>
<th>High level</th>
<th>Business perspective: what must be accomplished</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Requirements</td>
<td>Functionality provided to the user; user interaction with the system</td>
<td>Mid level</td>
<td></td>
</tr>
<tr>
<td>System Requirements</td>
<td>How to integrate with existing system components, platforms, interfaces</td>
<td>Higher level</td>
<td>Solution perspective: what the solution must be able to do</td>
</tr>
<tr>
<td>Functional Requirements</td>
<td>system services or functions; capabilities; behaviour;</td>
<td>Lower level</td>
<td>how well it must perform</td>
</tr>
<tr>
<td>Non-functional requirements</td>
<td>constraints on the system or on the development process: quality of service, performance, reliability, testability</td>
<td>Lower level</td>
<td></td>
</tr>
</tbody>
</table>
Requirement properties

• Specific
  • Cohesive: one issue

Too much:  ✗

The system must generate reports and performance metrics.
Requirement properties

- Specific
  - Cohesive: one issue
  - Complete: fully stated

Too little:
The system must generate reports. ✗
Requirement properties

- Specific
  - Cohesive: one issue
  - Complete: fully stated
  - Correct

- Measurable
  - Testable

The system must generate *attractive* reports in *a timely manner* ❌
Requirement properties

- Specific
  - Cohesive: one issue
  - Complete: fully stated
  - Correct
- Measurable
  - Testable
  - Defined terms

The system must generate HSTQ-style reports ✗
Requirement properties

• Specific
  • Cohesive: one issue
  • Complete: fully stated
  • Correct

• Measurable
  • Testable
  • Defined terms

• Attainable

Reports on unsolicited submissions must be available at all sites as soon as they occur.
Requirement properties

- Specific
  - Cohesive: one issue
  - Complete: fully stated
  - Correct
- Measurable
  - Testable
  - Defined terms
- Attainable

- Relevant
  - Traceable to a business need

The system must generate reports on the publishers’ business growth
Requirement properties

• Specific
  • Cohesive: one issue
  • Complete: fully stated
  • Correct

• Measurable
  • Testable
  • Defined terms

• Attainable

• Relevant
  • Traceable to a business need

• Time Bound

The system must generate daily reports on unsolicited submissions.
Requirement properties

• Specific
  • Cohesive: one issue
  • Complete: fully stated
  • Correct

• Measurable
  • Testable
  • Defined terms

• Attainable

• Relevant
  • Traceable to a business need

• Time Bound

• Implementation neutral

The system must generate reports on unsolicited submissions as relational database tables. The system must email reports on unsolicited submissions to the head of department.
Requirement properties

- Specific
  - Cohesive: one issue
  - Complete: fully stated
  - Correct
- Measurable
  - Testable
  - Defined terms
- Attainable

- Relevant
  - Traceable to a business need
- Time Bound
- Implementation neutral

The system must generate reports on unsolicited submissions as relational database tables. / The system must email reports on unsolicited submissions to the head of department.
Identify stakeholders

- People affected by the system, who
  - Operate the system
  - Benefit from functionality, politically, financially, socially
  - Involved in procuring the system
  - Regulators (legal, health & safety)
  - Responsible for the system
  - Outside the organization, who are affected
  - Who oppose it
Elicit requirements: stakeholders

- Identify needs
  - Interviews with individuals
    - Detailed specifications
    - Uninfluenced perspectives
  - Focus groups, requirements workshops
  - Document analysis, ...
- Later:
  - Requirements prioritization
  - Requirements review
Elicit requirements: stakeholders

• Identify needs
  • Interviews with individuals
    • Detailed specifications
    • Uninfluenced perspectives
  • Focus groups, requirements workshops
  • Document analysis, ...
  • Later:
    • Requirements prioritization
    • Requirements review

• Prototype
  • Involve stakeholders early
  • Improve ones understanding
  • Manage expectations

• Storyboards
  • Screen sequences illustrate steps in user experience

© Scott Adams, Inc./Dist. by UFS, Inc.
Elicit requirements: stakeholders

- Change management to ensure system acceptance and fit-for-purpose
  - Stakeholders must see the benefit of the change
  - Stakeholders must own the system
  - Stakeholders must be trained effectively
Elicit requirements: processes

- Identify business processes
  - Work place observation
  - How stakeholders interact with the system
  - Technological change leads to business change
Elicit req’s: use cases & scenarios

1. Determine Descriptive Metadata (ILS, DLS, other stores)
2. Map Descriptive Metadata between catalogues (ILS, DLS, other stores)
3. Determine content, format and schedule of export files
4. Determine interfaces for modular formats and structures
5. Determine normalised forms
6. Determine workflow processes
7. Determine Acquisition Metadata
Elicit requirements: frameworks

- Other organizations' requirements documents
- Software vendor’s specifications
- Strategy and policy documents
- OAIS – Open Archival Information System
- TRAC – Trusted Digital Repository Audit and Certification
- DoD 5015.2 – Baseline requirements for records management applications
- MOREQ – Model Requirements for the Management of Electronic Records
- GARP – General Accepted Recordkeeping Principles
- SAA Glossary of Terms (http://www.archivists.org/glossary)
Requirements sets

• Correct
• Consistent
• Complete
• Non-redundant
Requirements sets

• Correct
• Consistent
• Complete
• Non-redundant
• Structured
Requirements structure

4 REQUIREMENTS

4.1 REQUIREMENTS FOR INGEST FROM PREVIOUS RELEASES
4.2 HIGH LEVEL INGEST REQUIREMENTS
4.3 BUSINESS REQUIREMENTS
4.4 SYSTEMS REQUIREMENTS
4.5 SECURITY REQUIREMENTS
4.6 USABILITY REQUIREMENTS
4.7 OPERATIONS REQUIREMENTS
4.8 USER INTERFACE REQUIREMENTS
4.9 DATA AND METADATA REQUIREMENTS
4.10 SET-UP REQUIREMENTS
4.11 ACQUISITION MODULE REQUIREMENTS
4.12 PRE-INGEST MODULE REQUIREMENTS
4.13 INFORMATION PROVIDER ADAPTOR REQUIREMENTS
Requirements sets

• Correct
• Consistent
• Complete
• Non-redundant
• Structured
• Prioritised
Matching requirements to resources

• Resources are limited
• Assign a priority to each requirement
  • MoSCoW: Must Should, Could, Would be nice
• Specify what is out of scope

• Adopt an incremental approach
  • Start w/ core functionality
  • Add optional functionality over time
  • Learn from each increment
    • Functionality
    • Implementation
  • Adapt as necessary
Requirements sets

- Correct
- Consistent
- Complete
- Non-redundant
- Structured
- Prioritised

- Traceable
  - Forward: unique reference
  - Backward: source reference
Requirements sets

- Correct
- Consistent
- Complete
- Non-redundant
- Structured
- Prioritised
- Traceable
  - Forward: unique reference
  - Backward: source reference
- Modifiable
  - organization
  - table of contents
  - index
  - cross-referencing
Parts of an SRS

Introduction

Purpose

Scope: positive and negative

Definitions, acronyms, and abbreviations: for unambiguous requirements

References: for traceable requirements

Overview
## Glossary

**Acquisition**

Acquisition is the first step in the ingest process which involves the (push or pull) submission, virus check, and unpacking of archival and compressed files. In library terms it corresponds to the acquisition and accession tasks.

**Acquisition Method**

The Acquisition Method specifies the technical details needed to perform a successful acquisition. Considerations include the transport mechanism (e.g., FTP, SSH), control (pull or push), authentication (certificates, user names, passwords).

**Ingest**

The term Ingest is currently used in three ways in this and related documents:

a. The overarching term for the whole process of ingesting digital materials, comprising submission, acquisition, and so on.

b. The step in the ingest process which follows successful ‘acquisition’.

c. The core ‘ingest’ tasks such as validation, normalisation, and creating DSIPs for digital objects in a submission. In this usage, it may be preceded by ‘pre-ingest’ and followed by ‘finalise ingest’ tasks.
Parts of an SRS

Overall description

• Objectives
• Model
• Business Process
• Product functions
• Constraints
• Assumptions and dependencies

• User Roles and Responsibilities
• User characteristics
• Interactions with Other Systems
• Replacement of Legacy Systems
• Production Rollout Considerations
Parts of an SRS

Specific Requirements

Functionality
- Security
- Auditing
- Administration / Customization of the Application
- Reporting

Performance

Usability

External interface requirements
- User interfaces
- Hardware interfaces
- Software interfaces
- Communications interfaces

Concurrency

Design

Software system attributes

Our digital memory accessible tomorrow www.dpconline.org
Parts of an SRS

Supporting Information

- Table of contents
- Index
- Appendices
Requirements sets

- Correct
- Consistent
- Complete
- Non-redundant
- Structured
- Prioritised

- Traceable
  - Forward: unique reference
  - Backward: source reference

- Modifiable
  - organization
  - table of contents
  - index
  - cross-referencing
Customer requirement

1. Have one trunk
2. Have four legs
3. Should carry load both passenger & cargo
4. Black in color
5. Should be herbivorous

Our Solution

1. Have one trunk ✔
2. Have four legs ✔
3. Should carry load both passenger & cargo ✔
4. Black in color ✔
5. Should be herbivorous ✔

Our Value add:
Also gives milk 😊