Piecing Together the Requirements Jigsaw-Puzzle

Ian Alexander
www.scenarioplus.org.uk

REFSQ
Essen, July 2010
About Keynote Talks

• Novelty (for Researchers)
  – impossible

• Specific Advice (for Practitioners)
  – unwise

• Interest (for Everyone)
  – required

Propositions

1. Requirements are **not put together well**.
2. Researchers, Authors, Trainers behave (and write) **as if all projects are alike**.
3. Projects are **NOT all alike**.
4. But certain **patterns constantly recur**.
5. **Quite enough solutions** have been suggested already.

pieces of the puzzle
Questions

1. What are these basic pieces?

2. How do the pieces fit together, typically?

3. How can different projects re-assemble the pieces of their puzzles?

An Industry Observation

You mean there are PIECES?!!

What would we DO with them all?

Let’s just get on and write REQUIREMENTS!
A Research Observation

There AREN'T ENOUGH pieces …

Why don't we just create SOME MORE formal meta-systolic quasi-temporal logics?

A Fashion Observation

Requirements are so passé (…so 1990s)

Now we're all agile

New Agile Requirements

Old Maladroit Requirements
Challenges

I've got to...

• model my Requirements in UML
• create Use Cases
• do Agile instead of requirements
• use my company’s Standard SRS* Template
• upgrade a highly-constrained existing system

• research aspect-oriented cultural hermeneutics
  as commonly used in practical industrial applications

Mmhmm. Let's see if we can help a little.........

* Software Requirements Specification

The Requirements Jigsaw-Puzzle

1. **What are the basic pieces?**
2. How do the pieces fit together, typically?
3. How can different projects re-assemble the pieces of their puzzles?

We'll take a look at the Challenges here
### Pieces of the Puzzle

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Given</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Vision</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Discovery Contexts**

- A. From Individuals
- B. From Groups
- C. From Things
- D. Trade-Offs

*Not talking much about Discovery Contexts today*

---

**1. Vision**

- **What is this project for?**
Business Vision

• “To become market leader in small-household burglar alarms”

• “To make steadily growing annual income from alarm sales, maintenance, and monitoring”

2. Stakeholders

• *Who has a valid interest in this product?*
Space Telescope Stakeholders

(From Writing Better Requirements, by Ian Alexander & Richard Stevens, Addison-Wesley 2002)

© Ian Alexander 2010

Typical Stakeholder Roles

- Beneficiary
- Negative
- Regulatory
- Operational
- Expert

- Functional
- Financial
- Political
- Purchasing
- Voluntary (Standardising)
- Enforcing
- Human Operator
- Neighbouring System
- Safety Opinion
- Usability Opinion
- Domain Knowledge
- Implementability

© Ian Alexander 2010
3. Goals

- What do different stakeholders want?
- What conflicts exist?
4. Context

- *Where is the boundary of this system?*
## Context, Events, Scope

<table>
<thead>
<tr>
<th>Event</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notified Intrusion</td>
<td>In scope</td>
</tr>
<tr>
<td>Guard Callout</td>
<td>In scope</td>
</tr>
<tr>
<td>Maintenance Appointment</td>
<td>In scope</td>
</tr>
<tr>
<td>Repair</td>
<td>In scope</td>
</tr>
<tr>
<td>Disconfirmation</td>
<td>In scope</td>
</tr>
<tr>
<td>Burglary Report</td>
<td>In scope</td>
</tr>
<tr>
<td>Subscription</td>
<td>In scope</td>
</tr>
<tr>
<td>Change Details</td>
<td>In scope</td>
</tr>
</tbody>
</table>
5. Scenarios

- How will people use this product?

### Scenarios

<table>
<thead>
<tr>
<th>Role</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Householder</td>
<td>Arms the alarm</td>
</tr>
<tr>
<td>Alarm</td>
<td>Indicates 'arming' (e.g. buzzer, lamp), starts countdown timer for arming_period</td>
</tr>
<tr>
<td>Householder</td>
<td>Leaves house by main_door, closes main_door (and probably locks it)</td>
</tr>
<tr>
<td>Alarm</td>
<td>Stops countdown timer, stops indicting 'arming', begins watching</td>
</tr>
<tr>
<td>Burglar</td>
<td>Breaks into protected house</td>
</tr>
<tr>
<td>Alarm</td>
<td>...</td>
</tr>
</tbody>
</table>

Notations include Role/Action, Swimlanes, Use Cases, Storyboards, …
6. Qualities & Constraints

- How do people want this product to be?
- What limits exist?

Qualities and Constraints

Local: One Function
- set the alarm
  - quickly
  - reliably

Global: Whole Product
- complying with EU Low Voltage Directive (LVD) for 'CE' marking
7. Rationale

- Why is this requirement here?
Rationale
e.g. as Assumptions

- Goal: Alarm sounds locally
  - Examples: bell, siren
  - Assumption (Warrant):
    - Customers expect audible alarm
    - Audible alarm acts as deterrent
  - Assumption (Rebuttal):
    - Silent alarm gives more chance to arrest intruders

Ways to Document Rationale

- What do practitioners think when they find there are (at least) 8 ways to achieve a task and (at least) 8 graphical notations to choose from?

Notations include Toulmin, GSN, CAE, IBIS, DRied, GOC, DRSS, DRL, ...
8. Definitions

- *What does this term mean?*

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suspected Intrusion</td>
<td>Message from <em>Household Alarm</em> to <em>Control Centre</em>, indicating a possible <em>Intrusion</em> for <em>Confirmation</em></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

© Ian Alexander 2010
9. Measurements

How to know we have what we asked for?

Goal: Alarm sounds locally (e.g. bell, siren)

- Acceptance Criteria:
  - audible at **100** metres
  - stops after `alarm_sound_period` minutes

Approaches include:
- Acceptance / Test / Fit Criteria
- Postconditions
- Success/Minimal Guarantees
- MoP
- MoE
- QoS …

... not to mention “traditional Shall-Statement requirements” …
10. Priorities

- **Input:** What do people want?
- **Output:** What should be done (first)?

**Goal:** Alarm sounds locally (e.g. bell, siren)
- Priority: **Very Desirable**
  - Rationale: Alarm can work without it, but customers want it

**Goal:** Alarm notifies Control Centre
- Priority: **Essential**
  - Rationale: Our monitoring business depends on it (see Business Case)
The Requirements Jigsaw-Puzzle

1. What are the basic pieces?

2. How do the pieces fit together, typically?

3. How can different projects re-assemble the pieces of their puzzles?

How do the pieces fit together?

1. **Embryology**: as sequences in development life-cycle

2. **Traceability**: by a rich web of connections

3. **Validation**: making use of connections to cross-check

4. **Teamwork**: having all specialists pulling together

5. **Trade-offs**: choosing "least-worst" option(s)

6. **Dialogue**: translating back to stakeholder language

… no doubt you can think of more …
1. Embryology

- Embryology
- Requirements Completeness
- Urgent to High
- Stakeholder-directed search
- Goal-directed search
- Context-directed search
- Event-directed search
- Scenario-directed search
- Definition-directed search
- Exception-directed search
- Prototyping-directed search
- NFR-directed search
- Negative Stakeholder-directed search
- Archaeology-directed search

2. Traceability

- Functional Goal
- Quality Goal
- Measurable Quality
- Use Case
- Functional Requirement

- This looks like software design modelling…
- Maybe, but we’re modelling what is REQUIRED – many design decisions can wait till later

A rich web of interconnections everywhere
3. Validation

Named State contains Statechart contains Action

Named Value defines Dictionary
implies

Functional Requirement

refers to

Makes use of rich web of interconnections

This state isn’t defined yet

This action isn’t implemented anywhere...

This value isn’t in use

… etc …

4. Teamwork

Sales  Planning  Documentation  Software  Systems  Security  Usability  Testing

I met a guy from Testing the other week

Bet he didn’t know anything about what’s going on

© Ian Alexander 2010
5. Trade-offs

Here are "The Definitive Requirements" that will be too slow. That one is very risky. This would be much cheaper. With this solution we can get 85%. That way isn't as safe.

- Not much on this in RE books
- Too dirty?
- Not "proper RE"?

6a. A traditional dialogue

Discover Document Validate

Stakeholders

Think

Act

Develop
The Requirements Jigsaw-Puzzle

1. What are the basic pieces?
2. How do the pieces fit together, typically?
3. How can different projects re-assemble the pieces of their puzzles?
Example: Meta-Model for a Retail Project

Example: Matrix for a Retail Project

<table>
<thead>
<tr>
<th>Requirement Elements</th>
<th>Stakeholders</th>
<th>Context, Interfaces, Scope</th>
<th>Scenarios</th>
<th>Qualities and Constraints</th>
<th>Rationale</th>
<th>Definitions</th>
<th>Measurements</th>
<th>Priorities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery Contexts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>From Individuals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>From Groups</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>From Things</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trade-Offs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Example Situations: "I've got to…"

- model my requirements in UML
- create Use Cases
- do Agile instead of requirements
- use my company's standard SRS template
- upgrade a highly-constrained existing system
- research …
I've got to **model my requirements in UML**

- **Goals:**
  - add informal diagrams
    (abuse the Use Case notation...)
  - or just make a list

- **Rationale:**
  - annotate with notes for assumptions, etc
  - add informal diagrams

---

I've got to **create Use Cases**

- **Annotate your Use Cases**
  - add subsections for
    - **Stakeholders**
    - **Rationale**
    - **Qualities & Constraints**
    - **Priority**

- **Add Misuse Cases**
  - to identify & justify
    - **Safety, Security, Reliability Requirements**
    - **Trade-offs**
I've got to do Agile *instead of* Requirements

- make use of User Stories to define both Functions (scenarios) and product Qualities

- write Usability tests, Performance tests, Security tests

- list Issues, Risks

- *when all else fails*, draw an Architecture

I've got to use my company's standard SRS template

- well, fill it in then!
  - as briefly as possible

- then add sections for "Goals", "Scenarios", "Qualities" *, ...  
  - until you can say what you need to, clearly

* why not borrow the NFR template from www.scenarioplus.org.uk ?
I've got to **upgrade** a highly-constrained existing system

0. Throw away your "green-field" requirements textbooks

1. Model your **goals** for the upgrade
2. Define **context** of existing and new systems
3. Identify interfaces that **cannot be changed**
4. Explore **options** where scope can be changed
5. Identify **stakeholders**, esp. where scope has changed
6. Discover stakeholder **goals, conflicts, input priorities**
7. **Trade-off** alternative solutions against goals
8. Set project's **output priorities**

---

I've got to **research** …

- **remember only specially tame** industrialists are allowed into research conferences

- **go and visit** an industrial project
  - **see** what they are doing
  - **find out** what they need
  - **prototype a simple front end** that everyone can use

Whatever you do, please keep it simple
Challenges for the Future

- Industry-wide notations for Goals, Rationale
- Incorporated into UML?
- Managers, Analysts know to do Stakeholders, Goals, NFRs ...
- Education
- Standardisation
- Case Studies
- Dissemination
- Researchers, Teachers, Authors understand diversity of projects in industry ...
- Action Research

Thank you for Listening