Contents

1 Introduction .................................................................................................................................. 2
  1.1 References .......................................................................................................................... 2
  1.2 Abbreviations used in this Document ......................................................................................... 2
2 Red Light ...................................................................................................................................... 3
3 Amber Light .................................................................................................................................. 4
4 Green Light .................................................................................................................................. 5
5 Observations ................................................................................................................................. 8
1 Introduction

This document gives some guidance notes on using Scarecrow’s Approach to Context-based Requirements Engineering (ACRE). It is not intended to provide a requirements engineering process for the application of ACRE, nor does it describe ACRE. For such a process and description see Appendix F and Chapter 9 of [Holt & Perry 2013]. The guidance notes in this document are intended to aid an engineer in gaining familiarity with ACRE.

The guidance notes use a traffic light analogy that is intended to ensure engineers do not start requirements analysis before requirements elicitation is complete, an all too common problem with in-depth analysis of requirements often taking place leading to proposed solutions before the full requirements set is even complete.

The colours adopted equate to the amount of thinking (in reality, the amount of analysis) that should be undertaken:

- Red light - Don’t think too much. This is about elicitation, not analysis.
- Amber light - Start thinking now, but not too deeply; still not into analysis.
- Green light - Now start thinking hard. This is not easy and requires effort.

You will see throughout this document that some of the ACRE Views are referred to in two ways, for example the Requirement Description View (RDV) is also called the Need Description View (NDV). Similarly, the Requirement Context View (RCV) is also called the Need Context View (NCV). This reflects the fact that ACRE continues to evolve and has been implemented in many organisations. In some the opportunity has been taken to rename the slightly misnamed (for historical reasons beyond the scope of this document) RDV and RCV to NDV and NCV. Nevertheless, these Views are the same, albeit with slightly different names.

1.1 References


1.2 Abbreviations used in this Document

The following abbreviations are used in this document:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACRE</td>
<td>Approach to Context-Based Requirements Engineering</td>
</tr>
<tr>
<td>CDV</td>
<td>Context Definition View</td>
</tr>
<tr>
<td>NCV</td>
<td>Need Context View</td>
</tr>
<tr>
<td>NDV</td>
<td>Need Description View</td>
</tr>
<tr>
<td>RCV</td>
<td>Requirement Context View</td>
</tr>
<tr>
<td>RDV</td>
<td>Requirement Description View</td>
</tr>
<tr>
<td>SEMP</td>
<td>Systems Engineering Management Plan</td>
</tr>
<tr>
<td>SEV</td>
<td>Source Element View</td>
</tr>
<tr>
<td>SysML</td>
<td>Systems Modelling Language</td>
</tr>
<tr>
<td>VV</td>
<td>Validation View</td>
</tr>
</tbody>
</table>
2 Red Light

Don’t think too much. This is about elicitation, not analysis.

- What is your System?
  - What is the ‘mission statement’ or ‘root definition’ (see http://www.incoseonline.org.uk/Program_Files/Publications/zGuides_4.aspx)
  - This should be short and snappy. You should be able to tell it to your mother/father/significant other and have them understand it.
  - Document this in the SEMP

- Who are your Stakeholders?
  - (In full Stakeholder Roles)
  - Remember: It is not sufficient to think just about ‘customers’/ ‘users’/‘suppliers’ etc.
    - There are lots of different about ‘customers’/ ‘users’/‘suppliers’ etc.
    - For example, in an automotive project there are lots of different ‘users’: fleet owner, weekend driver, “boy racer”, mother on school run, commuter, salesman etc., along with other Stakeholder Roles such as safety engineers, testers, car thieves, garage mechanics, marketing, manufacturing (and this breaks down too), emergency services, legislation etc. etc. etc.
    - They all think about the System in different ways.
      - Each has their own point of view = Context
  - This all means that there is no such thing as the Customer/User Requirements Specification.
  - Capture on CDVs.

- Gather the Needs
  - No thinking, no interpretation, just capture
    - Make no assumptions about what Stakeholders want
      - If in doubt, ask them
    - Use the language of the Stakeholder
  - If want, classify as Goal, Capability (Feature) or Requirement and trace between them
    - On first use of ACRE, most Needs should be Goals or Capabilities (Features)
    - Those Requirements you have will typically be acting as constraints
  - Look for Needs in Source Elements – documents, existing systems, emails, talking to Stakeholder Roles etc.
  - Trace all Needs to Source Elements
  - Capture on SEVs and RDVs (NDVs)
3 Amber Light
Start thinking now, but not too deeply; still not into analysis.

- Create a Stakeholder to Need table
  - (This will help with Use Cases later)
  - Think about which Needs are important to which Stakeholder Roles
    - But don’t think too much
    - “Gut reaction” is the level of thought here
  - Put a ✓ or ✗ in box
  - If in doubt, put in a ✗

<table>
<thead>
<tr>
<th>SH1</th>
<th>SH2</th>
<th>SH3 ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>N2</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>N3</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

- Might have to try to think like a particular Stakeholder Role
  - Talk to the Stakeholder Role(s) if possible
- Sum the rows and the columns
  - Big numbers imply importance of Stakeholder Role and Need
- Review with "important" Stakeholder Roles
  - Think more here – don’t blindly use the scores
  - Care is needed. A Stakeholder Role (e.g. ‘Safety Engineer’) might score low (i.e. not many ✓s in their column) but be a very important Stakeholder Role
4 Green Light

Now start thinking hard. This is not easy and requires effort.

- Create Requirement (Need) Context Views
  - Take each Stakeholder Role column from the Stakeholder to Need Table and create one Requirement (Need) Context View per Stakeholder Role.
  - Each ✔ Need becomes a Use Case
    - Trace the Use Case to the Need with a «refine» dependency
    - Think what the Need means to that Stakeholder Role
      - Write this down!
      - This is the Use Case's description
      - A Need interpreted in a Context
    - Use the table to link Use Cases to other Stakeholder Roles
      - Look across the rows for ✔s
      - Relate Use Cases to each other using generalisation, «include», «extend» & «constrain» dependencies
        - It is useful to include the ‘mission statement’ or ‘root definition’ on all RCVs (NCVs)

- Does a Use Case need breaking down into more detail (in the current Context)?
  - (Possibly) do on a separate RCV with no boundary but showing the Stakeholder Role whose Context it is from as an actor
  - E.g. for Use Case ‘SH1-N1’ above:

- Remember the 7 ± 2 rule when drawing diagrams
• Validate the Use Cases
  • Again, in Context
    • This is not easy and requires effort
  • Take a RCV (NCV)
  • Look at each Use Case & think about Scenarios – stories
    • (In practice, a Use Case & “immediate” connected/related Use Cases E.g.

  • How does the System, as a “black box”, behave for that Uses Case?
  • Don’t forget that the Use Case is in a particular Context
  • Write the Scenario down – include other Stakeholder Roles related to the Use Case on the RCV (NCV) and don’t forget to include the Stakeholder Role whose Context is being considered
  • Capture on a VV
    • In SysML, typically a sequence diagram

  • There will be > 1 Scenario for each Use Case
    • Ideal behaviour – “sunny day”
    • Fault behaviour – “rainy day”
    • Etc.
  • If can think of only one or two Scenarios
    • Probably not a Use Case
  • If can think of more than ten
    • Use Case probably needs to be broken down
  • Trace the VV to the Use Case that it validates with a «validate» dependency
• Capture new **Needs**
  • New **Needs** may be discovered while creating **Scenarios**
  • Capture these on a **RDV (NDV)**
  • These new needs must be discussed with **Stakeholders** to ensure that they are valid
  • Trace the new **Needs** – the **Scenarios** will act as **Source Elements**
  • These new **Needs** may be:
    • At the same level
    • At the next level down in the **System** hierarchy
5 Observations

The above description is typical of the first pass through ACRE, when the emphasis is outward-facing and the System is treated as a “black box”. Needs (and hence Use Cases) are from the point of view of Stakeholder Roles and are typically written in the form “The Stakeholder wants...”.

In subsequent passes through the approach at lower levels of the System hierarchy, the emphasis becomes inward-facing. Needs (and hence Use Cases) are from the point of view of the System & as the approach is applied at lower levels, from the point of view of elements of the System (e.g. Sub-systems, Components, Items etc.) and are typically written in the form “The System/Sub-system/Component/Item shall...”. Scenarios at these levels typically do not explicitly include Stakeholder Roles.