

Key Skills For Requirements Management
An Analyst/Requirements Manager's Cheat-Sheet for New Projects

**The Goal: To deliver *quality* products
on time and on budget
which meet the customer's *real* needs.**

1. Build the *Right System Right*

Adopt a **process** (*building the system right*) that will provide a systematic approach to establishing and maintaining agreement with the customer on a set of **requirements** that will drive the design, implementation and testing of the system (*building the right system*). Engage the test department in the requirements management challenge now. Have them involved in test planning from day 1. Have QA assume a role in the Vision Document and SRS reviews. Have the testers build test procedures and test cases that trace back to the functional and nonfunctional requirements. Hold them responsible for maintaining the traceability links and for updating status of the test activities.

2. Get Agreement on the Problem Being Solved

Execute the four steps in problem analysis:

- a) Gain agreement on the **problem** being solved.
- b) Identify the **stakeholders**. (Which are actors?)
- c) Define the **system boundaries**. (Use identified actors.)
- d) Identify **constraints** imposed on the solution

Circulate the problem statement and insist that you gain agreement on the problem statement before moving forward (to avoid the "Yes..., but ...").

3. Understand Stakeholder Needs

- a) Create a structured **interview** (use **generic template**) pertinent to your application. Interview 5-15 users/stakeholders (identified in Step 1 above). Summarize the interviews by aggregating 10-15 user needs or requested features for each. Start **requirements traceability** now.
- b) Facilitate a **requirements workshop** for your project. Use "out-of-box" and "in-box" warm-up papers. (in-box data taken from Step 2A). Run a **brainstorming session** to identify/refine **features** and **use cases** identified in your interviews. Do **idea reduction and prioritization**. Rerun the workshop periodically, (annually or twice annually) to provide ongoing top-down customer input.
- c) Create **storyboards** for all innovative concepts. Use storyboards to develop scenarios with your users to make sure you get it right.

4. Define the System.

- a) Adapt and adopt the **Vision Document** template to suit your needs. Create a **Product Position Statement**. Circulate it WIDELY (everybody.com) and make sure you have agreement. If you don't have it, stop and get it.
- b) Enter all **features** identified in step 2 and through other inputs (development, help desk, marketing, etc) in the Vision Document. Trace them back to stakeholder needs. Use **attributes** of Priority (Critical, Important, Useful), Risk (H, M, L), Effort (team-months) Stability (H, M, L), Release (e.g., v1.0). Identify

sources for non-functional requirements (licensing, documentation, legal and regulatory, etc) in Vision Document. Make the Vision Document a living document, the official channel. Find a **Product Champion**. Use **Delta Vision Document** concept going forward. Commit your company to having a current Vision Document at all times.

- c) Develop the initial **Use-Case Model**: Identify **actors** and **use cases** with brief descriptions and communication-associations between them.

5. Manage Scope.

Set the **baseline** for each release in the Vision Document by adjusting attributes and setting the version number. Help the team make the hard decisions and get the decisions behind you. Preach and teach **iterative development**. Communicate and **manage expectations** everywhere. Use the Delta Vision Document to capture all new features that arise through normal course of events. Make sure that all suggested features are recorded; none are lost.

6. Refine the System Definition.

There shall be a **software requirements specification** for your project which defines the complete set of functional and nonfunctional behaviors of the product. Engage the development and test team to adopt and manage this workload. Assist them with training and find them help if they need it. Use **formal analysis methods** only where necessary.

Adopt **use-case methodology** to document all functional behaviors. Trace non-functional requirements to use cases, where appropriate. Create Supplementary Specifications for all system-wide nonfunctional requirements. Trace requirements back to their source.

[And /or]

Adopt the **IEEE SRS** standard and modify it to suit your needs.

7. Manage Changing Requirements

Personally maintain responsibility for the Vision Document. Have weekly reviews with the team to assess status. Set up default reports and queries to assist this effort. Let QA help monitor the SRS and test process. Keep the Vision Document alive as your primary management document.

8. Next Iteration.

Go back to Step 1 and repeat and refine as necessary!

Keep the goal in mind:

<p style="text-align: center;">The Goal: To deliver <i>quality</i> products <i>on time</i> and <i>on budget</i> which meet the customer's <i>real needs</i>.</p>
