

# Volere

## Requirements Specification Template

*Edition 9*

James & Suzanne Robertson  
Principals of the Atlantic Systems Guild  
London, Aachen & New York  
Email [james@systemsguild.com](mailto:james@systemsguild.com) [suzanne@systemsguild.com](mailto:suzanne@systemsguild.com)

**Copyright © 1995 – 2003 Atlantic Systems Guild**

This document may be used, modified or copied for internal use, provided this copyright is acknowledged. This is intended to form the basis of your requirements specification. It may not be sold, or used for commercial gain or other purposes without prior written permission.

Updates to this template are posted on our web sites  
[www.systemsguild.com](http://www.systemsguild.com) and [www.volere.co.uk](http://www.volere.co.uk)

# The ..... System Requirements Specification Version ...

## Table of Contents

### **PROJECT DRIVERS**

1. The Purpose of the Product
2. Client, Customer and other Stakeholders
3. Users of the Product

### **PROJECT CONSTRAINTS**

4. Mandated Constraints
5. Naming Conventions and Definitions
6. Relevant Facts and Assumptions

### **FUNCTIONAL REQUIREMENTS**

7. The Scope of the Work
8. The Scope of the Product
9. Functional and Data Requirements

### **NON-FUNCTIONAL REQUIREMENTS**

10. Look and Feel Requirements
11. Usability Requirements
12. Performance Requirements
13. Operational Requirements
14. Maintainability and Portability Requirements
15. Security Requirements
16. Cultural and Political Requirements
17. Legal Requirements

### **PROJECT ISSUES**

18. Open Issues
19. Off-the-Shelf Solutions
20. New Problems
21. Tasks
22. Cutover
23. Risks
24. Costs
25. User Documentation and Training
26. Waiting Room
27. Ideas for Solutions

Specification prepared by ..... Date .....

## Preamble

---

*This is a **template** for a requirements specification. Select all the sections that apply to your project, and replace the entries with your text. Delete any sections that are not relevant. Add any applicable new sections, and any facts that are relevant to your product.*

### **Volere**

Volere is the result of many years of practice, consulting and research in requirements engineering. We have packaged our experience in the form of a generic requirements process, requirements training, requirements consultancy, requirements audits and this requirements template.

The Volere requirements process is described in the book:

*Mastering the Requirements Process* by Suzanne and James Robertson, Addison-Wesley, London, 1999. ISBN is 0-201-36046-2

**Public seminars** on Volere are run on a regular basis in Europe, United States and Australia.

**In house seminars and consulting** on Volere can be arranged on demand.

**For further information** contact: The Atlantic Systems Guild, 11 St Mary's Terrace, London, W2 1SU, United Kingdom.

email: [suzanne@systemsguild.com](mailto:suzanne@systemsguild.com) [james@systemsguild.com](mailto:james@systemsguild.com)

web: <http://www.systemsguild.com>

## Requirements Types

*Functional requirements* are the fundamental subject matter of the system and are measured by concrete means like data values, decision making logic and algorithms.

*Non-functional requirements* are the behavioral properties that the specified functions must have, such as performance, usability, etc. Non-functional requirements can be assigned a specific measurement. This template will give examples of quantifying non-functional requirements.

*Project constraints* identify how the eventual product must fit into the world. For example the product might have to interface with or use some existing hardware, software or business practice, or it might have to fit within a defined budget or be ready by a defined date.

*Project drivers* are the business- related forces. For example the purpose of the product is a project driver, as are all of the stakeholders – each for different reasons.

*Project issues* define the conditions under which the project will be done. We include these in the requirements specification to present a coherent picture of all the factors that contribute to the success or failure of the project.

## Testing requirements

You start testing requirements as soon as you start writing them.

Your first test is to determine if you can quantify the requirement by specifying its fit criterion. This *fit criterion* is an objective measure of the requirement's meaning; it is the criterion for evaluating whether or not a given solution fits the requirement. If a *fit criterion* cannot be adequately specified, then the requirement is ambiguous, or ill understood. If there is no *fit criterion*, then there is no way of knowing if a solution matches the requirement.

## Requirement Shell

Use this requirement shell as a guide for writing each requirement.

**Requirement #:** Unique id

**Requirement Type:** The type from the template

**Event/use case #:** List of events / use cases that need this requirement

**Description:** A one sentence statement of the intention of the requirement

**Rationale:** A justification of the requirement

**Source:** Who raised this requirement?

**Fit Criterion:** A measurement of the requirement such that it is possible to test if the solution matches the original requirement

**Customer Satisfaction:** Degree of stakeholder happiness if this requirement is successfully implemented. Scale from 1 = uninterested to 5 = extremely pleased.

**Customer Dissatisfaction:** Measure of stakeholder unhappiness if this requirement is not part of the final product. Scale from 1 = hardly matters to 5 = extremely displeased.

**Dependencies:** A list of other requirements that have some dependency on this one

**Conflicts:** Other requirements that cannot be implemented if this one is

**Supporting Materials:** Pointer to documents that illustrate and explain this requirement

**History:** Creation, changes,

**Volere**  
Copyright © Atlantic Systems Guild

## Requirement Numbering

Give each requirement a unique identifier to make it traceable throughout the development process. The numbering scheme suggested in the requirement shell is:

**Requirement #** is the next unique requirement number

**Requirement Type** is the section number from the template for this type of requirement

The inclusion of the section number is not absolutely necessary because we do have a unique requirement id. However it serves as a reminder of what this requirement relates to and helps to remind why the requirement is considered important. Also the ability to

compare requirements of the same type makes it easier to identify contradictions and duplications.

For example:

A functional requirement is section 9, and the next unique number is 128.

Requirement #: 128 Requirement Type: 9

We shall record the time when we are notified of a truck breakdown

A performance requirement comes from section 12, and the next unique number is 129.

Requirement #: 129 Requirement Type: 12

Truck drivers shall be informed of their schedule 30 minutes before leaving the depot.

### **Event/use case #**

is the identifier of a business event or use case that contains this requirement. There might be several Event/use case #'s for one requirement because the same requirement might relate to a number of events. The terms event and use case are already widely used in the systems development world.

We use the term business event to mean a business related happening that causes an event-response within the work that we are studying.

We use the term event-driven use case (or product use case) to mean a user-defined (or actor defined) piece of activity within the context of the product. Business events and product use cases provide a way of grouping business-related requirements and tracing them through into implementation; they are used throughout the Volere development process.

### **Customer Value**

Customer Value is a measure of how much your client cares about each requirement.

Ask your stakeholders to grade each requirement for Customer Satisfaction on a scale from 1 to 5 where 1 means mild interest if this requirement is satisfactorily implemented, and 5 means they will be very happy if this requirement is satisfactorily implemented

The stakeholders also grade each requirement for Customer Dissatisfaction on a scale from 1 to 5 where 1 means that it hardly matters, and 5 means that they will be extremely displeased if this requirement is not satisfactorily implemented

The point of having a satisfaction and a dissatisfaction rating is that it guides your clients to think of the requirement from two different perspectives, and helps you to uncover what they care about most deeply.

## **Dependencies**

This keeps track of other requirements that have an impact on this requirement.

If the dependency exists because requirements use the same information, then use of standard naming conventions and definitions (see Section 5) will implement this dependency.

Other dependencies exist because a solution to this requirement has a positive or negative effect on solutions to other requirements. Capture these types of dependencies by cross referencing the requirements.

Some requirements, especially project drivers and project constraints, have an impact on all the other requirements.

## **Conflicts**

This keeps track of other requirements that disagree with this one. Conflicts that are caused by mistake are solved simply by bringing them to the surface and resolving them. Other conflicts are because of true differences in opinion/intention. These are the conflicts that might eventually need to be addressed using negotiation or mediation techniques. There is nothing wrong with having conflicting requirements providing you know that you have them. Then you are in a position to address the conflict.

## **History**

We follow the requirement from the date that it was created, through all its changes. We minimize future confusion by recording the rationale for making major changes. When a requirement is deleted we record when and the rationale behind the deletion. The date that the requirement passes its quality checks, and who passed it, is also recorded.

## **Definitions used in this template**

### **Context of the Product**

The boundaries between the product that we intend to build and the people, organizations, other products and pieces of technology that have a direct interface with the product.

## **Context of the Work**

The subject matter, people and organizations that might have an impact on the requirements for the product. The context of study identifies the intersection of all the domains of interest.

## **Client**

The person or organization for which the product is being built, usually responsible for paying for the development of the product.

## **Customer**

The person or organization who will buy the product (note that the same person/organization might play both the client, customer and sometimes user roles)

## **Design or Systems Design**

Crafting a solution to fit the requirements.

## **Developers**

The people who specify and build the product.

## **Domain of Interest**

A subject matter area that has some relevance to the context of study.

## **Non-Functional Requirement**

A property that the eventual product must have.

## **Event**

We use the term business event to mean a business related happening within a system adjacent to the work that we are studying. The happening causes the work to produce an event-response.

## **Fit Criterion**

Objective measure for defining the meaning of a requirement, and eventually testing whether a given solution satisfies the original requirement.

## **Functional Requirement**

An action that the product must be able to take, something that the product must do.

## **Global Constraint**

Constraints that apply to the system as a whole.

## **Product**

This is what we are attempting to deliver. This could be a piece of software, the installation of a package, a set of procedures, a piece



of hardware, a piece of machinery, a new organization, or almost anything.

### **Requirement**

A measurable statement of intent about something that the product must do, or a property that the product must have, or a constraint on the system.

### **Stakeholder**

A stakeholder is a person who can affect the outcome/success of the project and/or is affected by its outcome/success.

### **System**

The business system whose requirements are being studied.

### **Systems Analysis**

Detailed study of the requirements, intended to prove their workability as input to systems design.

### **Use case**

We use the term event-driven use case (or product use case) to mean a user-defined (or actor defined) piece of activity within the context of the product.

### **User or End User**

Someone who has some kind of direct interface with the product.

# 1 The Purpose of the Product

---

## 1a. The user problem or background to the project effort.

### Content

A short description of the work context and the situation that triggered the development effort. It should also describe the work that the user wants to do with the delivered product.

### Motivation

Without this statement, the project lacks justification and direction.

### Considerations

You should consider whether or not the user problem is serious, and whether and why it needs to be solved.

## 1b. Goals of the project.

### Content

This boils down to one, or at most a few, sentences that say “What do we want this product for?” In other words, the real reason that the product is being developed.

### Motivation

There is a real danger of this purpose getting lost along the way. As the development effort heats up, and the customer and developers discover more and more what is possible, it may well be that the system as it is being constructed wanders away from the original goals. This is a bad thing unless there is some deliberate act by the client to change the goals. It may be necessary to appoint a person to be “custodian of the goals”, but it is probably sufficient to make the goals public, and periodically remind the developers of it. It should be mandatory to acknowledge the goals at every review session.

### Examples

“We want to give immediate and complete response to customers ordering our goods over the telephone.”

“We want to be able to forecast the weather.”

### Fit Criterion

An objective measure that will enable testing to determine if the goal has been met by the product.

Some guideline for making goals measurable are:

- Specify each adverb and adjective so that everyone on the project understands the same meaning.
- Replace pronouns with the names of specific people or organisations.
- Ensure that the meaning of every noun is defined in one place in the specification

For instance the above example could be analysed and made less ambiguous as follows:

*We* - Employees of XYZ Corporation

want to give

*immediate* - during the course of a telephone call  
and

*complete* - product availability and price

*response* - verbal information

to

*customers* - anyone who enquires about our products

*our* - supplied by XYZ Corporation

*goods* - products that we manufacture

over the telephone

Following the analysis, the goal could be restated as:

Employees of XYZ Corporation want to tell enquirers, during the course of a telephone call, the availability and price for any product manufactured by XYZ.

Whenever you analyze a goal using this technique you will find yourself going through several iterations. The discipline imposed by making the goal measurable guides you into asking more relevant questions about the meaning.

You can use the Volere Purpose, Advantage, Measurement questioning technique to help you make goals measurable.

## 2 Client, Customer and other Stakeholders

---

### **2a. The client is the person/s paying for the development, and owner of the delivered system.**

#### **Content**

This item must give the name of the client. It is permissible to have several names, but more than three negates the point.

#### **Motivation**

The client has the final acceptance of the system, and thus must be satisfied with the system as delivered. Where the system is being developed for in-house consumption, the roles of the client and the customer may be filled by the same person. If you cannot find a name for your client, then perhaps you should not be building the product.

#### **Considerations**

Sometimes, when building a package or a product for external users, the client is the marketing department. In this case, a person from the marketing department must be named as the client.

### **2b. The customer is the person/s who will buy the product from the client.**

#### **Content**

The name of the person who plays the role of the customer for the product. In the case of in house development the roles of the client and the customer are often played by the same person. In the case of the development of a mass market product there may be several people playing the role of customer. In the case of a product that is being developed for an international market, there might be a different customer (or customer profile) in each country.

#### **Motivation**

The customer role is ultimately responsible for deciding whether or not to buy the product from the client. The product must be built to satisfy the aims of the customer/s whilst conforming to the constraints of the client. Even if the customer/s are people who work for another part of the client's organization, they might still have the authority to decide whether or not to invest budget in the new product.

## 2c. Other stakeholders

### Content

The roles and (if possible) names of other people and organizations who are affected by the product or whose input is needed in order to build the product.

Examples of stakeholders include:

- Users (detailed in section 3)
- Sponsor
- Testers
- Business Analysts
- Technology Experts
- System Designers
- Marketing Experts
- Legal Experts
- Domain Experts
- Usability Experts
- Representatives of external associations

For each type of stakeholder identify:

- Stakeholder Identification (some combination of role/job title, person name, organization name),
- Knowledge needed by the project,
- Necessary degree of involvement for that stakeholder/knowledge combination,
- Degree of influence for that stakeholder/knowledge combination,
- Agreement on how to address conflict between stakeholders who have an interest in the same knowledge

### Motivation

Failure to recognize stakeholders results in missing requirements.

## 3 Users of the Product

---

### 3a. The users of the product

#### Content

A list of the potential users of the product. For each category of user, provide the following information:

- User name – This is most likely to be the name of a user group like: schoolchildren, road engineers, project managers.
- User role – Summarizes the users' responsibilities.
- Subject matter experience – Summarizes the users' knowledge of the business. Rate as novice, journeyman or master.
- Technological experience – this describes the users' experience with relevant technology. Rate as novice, journeyman or master.
- Other user characteristics – Describe any characteristics of the users that have an effect on the requirements and eventual design of the product. Describe things like:
  - Physical abilities/disabilities
  - Intellectual abilities/disabilities
  - Attitude to job
  - Attitude to technology
  - Education
  - Linguistic skills
  - Age group
  - Gender

#### Motivation

Users are human beings who interface with the product in some way. The role of the client is to pay for the development of the product and the role of the customer is to buy the product. The role of the user is to use the product to do work. You use the characteristics of the users to define the usability requirements for the product.

#### Examples

Users can come from wide, and sometimes unexpected, sources. Consider the possibility of your users being clerical staff, shop workers, managers, highly-trained operators, general public, casual

users, passers-by, illiterate people, tradesmen, students, test engineers, foreigners, children, lawyers, remote users, people using the system over the telephone or Internet connection, emergency workers, and so on.

### **3b. The priorities assigned to users**

#### **Content**

Attach to each category of user a priority rating. This gives the importance and precedence of the user. Prioritize the users into:

**Key users.** These are critical to the continued success of the product. Give greater importance to requirements generated by this category of user.

**Secondary users.** They will use the product, but their opinion of it has no effect on its long-term success. Where there is a conflict between secondary users' requirements and those of key users the key users take precedence.

**Unimportant users.** This category of user is given the lowest priority. It includes infrequent, unauthorized and unskilled users, and people who misuse the product.

Percentage of this type of user – this is intended to assess the amount of consideration given to this category of user.

#### **Motivation**

If some users are considered to be more important to the product, or the organization, then this should be stated because it should affect the way that you design the product. For instance, you need to know if there is a large customer who has specifically asked for the product, and if they do not get what they want then the results could be a significant loss of business.

Some users may be listed as having no impact on the product. This means that the users will make use of the product, but have no vested interest in it. In other words, these users will not complain, nor will they contribute. Any special requirements from these users will have a lower design priority.

### **3c. User participation**

#### **Content**

Where appropriate attach to the category of user, a statement of the participation that you think will be necessary to provide the requirements. Describe the contribution that you expect this user to provide – business knowledge, interface prototyping, usability requirements etc. If possible, assess the minimum amount of time

that this user must spend for you to be able to determine the complete requirements.

### **Motivation**

Many projects fail through lack of user participation, sometimes this is because the required degree of participation was not made clear. When people have to make a choice between getting their everyday work done and working on a new project, the everyday work takes priority. This requirement makes it clear, from the outset, that specified user resources must be allocated to the project.

## **4 Mandated Constraints**

---

**This section describes constraints on the requirements and the eventual design of the product.**

### **4a. Solution constraints**

#### **Content**

This specifies constraints on the way that the problem must be solved. You can think of these as mandated solutions. Carefully describe the mandated technology, include the appropriate version numbers, and a measurement of how you will test compliance. If possible, you should also explain the reason for using the technology.

#### **Motivation**

To identify constraints that must be part of the final product. Your client, customer or user may have design preferences. If these are not met then your solution is not acceptable.

#### **Examples**

The product must use the current 2-way radio system to communicate with the drivers in their trucks.

The product must use the Windows NT operating system.

The product must be a hand-held device.

#### **Considerations**

We want to define the boundaries within which we can solve the problem. Be careful because anyone who has experience/exposure to a piece of technology tends to see requirements in terms of that technology. This tendency leads people to impose solution constraints for the wrong reason and it's very easy for untrue constraints to creep into a specification. If you impose untrue



constraints the danger is that you do not have the creative freedom to come up with the best solution to the problem. The solution constraints should only be those that are absolutely non-negotiable. In other words, however you solve this problem you must use this particular technology. Any other solution would be unacceptable.

#### **4b. Implementation environment of the current system**

##### **Content**

This describes the technological and physical environment in which the product will be installed. This includes automated, mechanical, organizational and other devices. These include the non-human adjacent systems.

##### **Motivation**

To describe the technological environment into which the product must fit. The environment places design constraints on the product. This part of the specification provides enough information about the environment for the designers to make the product successfully interact with its surrounding technology.

The operational requirements are derived from this description.

##### **Examples**

This can be shown as a diagram, with some kind of icon to represent each separate device or person (processor). Draw arrows to identify the interfaces between the processors and annotate them with their form and content .

##### **Considerations**

All the component parts of the current system, regardless of their type, should be included in the description of the implementation environment.

If the product is to affect, or be important to the current organization, include an organization chart.

#### **4c. Partner applications**

##### **Content**

This describes applications that are not part of the product but with which the product will collaborate. These can be external applications, commercial packages or pre-existing in-house applications.

##### **Motivation**

To provide information about design constraints that are caused by using partner applications. By describing or modeling these partner

applications, you discover and highlight potential problems of integration.

### **Examples**

This section can be completed by including written descriptions, models or references to other specifications. The descriptions must include a full specification of all interfaces that will have an effect on the product.

### **Considerations**

Examine the work context model to determine if any of the adjacent systems should be treated as partner applications. It might also be necessary to examine some of the details of the work to discover relevant partner applications.

## **4d. Off-the-shelf software**

### **Content**

This describes applications that must be used to implement some of the requirements for the product.

### **Motivation**

To identify and describe existing commercial, free, open source, etc. products that must be incorporated into the eventual product. The characteristics, behavior and interfaces of the package are design constraints.

### **Examples**

This section can be completed by including written descriptions, models or references to supplier's specifications.

### **Considerations**

The use of a specific package has been mandated. When gathering requirements you may discover requirements that are in serious conflict with the behavior and characteristics of the package. Keep in mind that the use of the package was mandated before the full extent of the requirements was known. In light of your discoveries you must consider whether the package is a viable choice when all the requirements are known. If the use of the package is not negotiable, then the conflicting requirements will have to be discarded.

You should also consider if there are any legal implications arising from your use of the software. You might also cover this in section 17 – Legal Requirements.

## **4e. Anticipated workplace environment**

### **Content**

This describes the workplace in which the users will work and use the product. This should describe any features of the workplace that could have an effect on the design of the product.

### **Motivation**

To identify characteristics of the physical workplace so that the product is designed to compensate for any difficulties.

### **Examples**

The printer is a considerable distance from the user's desk. This constraint suggests that printed output should be de-emphasized.

The workplace is noisy, so audible signals might not work.

The workplace is outside so the product must be waterproof, have displays that are visible in sunlight and allow for the effect of wind on any paper output.

The user will be standing up or working in positions where he must hold the product. This suggests a hand-held product but only a careful study of the users' work and workplace will provide the necessary input to identifying the operational requirements.

### **Considerations**

The physical work environment constrains the way that work is done. The product should overcome whatever difficulties exist, however you might consider a redesign of the workplace as an alternative to having the product compensate for it.

## **4f. How long do the developers have to build the system?**

### **Content**

Any known deadlines, or windows of opportunity, should be stated here.

### **Motivation**

To identify critical times and dates that have an effect on product requirements. If the deadline is short, then the requirements must be kept to whatever can be built within the time allowed.

### **Examples**

To meet scheduled software releases.

There may be other parts of the business or other software products that are dependent on this product.

Windows of marketing opportunity.

Scheduled changes to the business that will use your product. For example the organization may be starting up a new factory and your product is needed before production can commence.

### **Considerations**

State deadline limitations that exist by stating the date and describing why it is critical. Also identify prior dates where parts of your product need to be available for testing.

You should also ask questions about the impact of not meeting the deadline like:

What happens if we don't build the system by .....?

What is the financial impact of not having, the system by...?

## **4g. What is the financial budget for the system?**

### **Content**

The budget for the system, expressed in money or available resources.

### **Motivation**

The requirements must not exceed the budget. This may constrain the number of requirements that can be included in the product.

The intention of this question is to determine if the product is really wanted.

### **Considerations**

Is it realistic to build a product within this budget? If the answer to this question is no, then either the client is not really committed to building the product or does not place enough value on the product. In either case you should consider whether it is worthwhile continuing.

## **5 Naming Conventions and Definitions**

---

**This section gives definitions of all terms, including acronyms, used in the project.**

### **Content**

A dictionary containing the meaning of all the names used within the requirements specification. Select names carefully to avoid giving a different, unintended meaning.

This dictionary should build on the standard names that your organization, or industry, uses. The names should also reflect the terminology in current use within the work area.

The dictionary contains all important names that are used by the project. For each name write a succinct definition. This definition must be agreed by the appropriate stakeholders.

### **Motivation**

Names are very important. They invoke meanings that, if carefully defined, can save hours of explanations. Attention to names at this stage of the project helps to highlight misunderstandings.

The dictionary produced during requirements is used and added to throughout the project.

### **Examples**

Gritter Truck -- a truck used for spreading de-icing substances on roads in winter.

### **Considerations**

Make use of existing references and data dictionaries. Obviously it is best to avoid renaming existing items unless they are so ambiguous that they cause confusion.

From the start of the project emphasize the need to avoid homonyms and synonyms and explain how they increase the cost of the project.

Later on, as the analysis progresses, this description will be expanded to define all the elementary terms that describe a truck.

Truck = Truck Registration Number, Truck Capacity,  
Truck Service Status

As we progress through the requirements specification each of the elementary terms will be defined in detail

Truck Capacity - the number of tonnes of deicing material that can be carried by a truck. From 0.5 to 2 tonnes

## 6 Relevant Facts and Assumptions

---

### **6a. External factors that have an effect on the product, but are not mandated requirements constraints.**

#### **Content**

Statements describing business rules, systems, activities in the world that have an effect on this product.

#### **Motivation**

Relevant facts might contribute to requirements. They will have an effect on the eventual design of the product.

#### **Examples**

One ton of de-icing material will treat 3 miles of single lane roadway.

The existing application is 10,000 lines of C code.

### **6b. Assumptions that the team are making about the project**

#### **Content**

A list of the assumptions that the developers are making. These might be about the intended operational environment, but can be about anything that has an effect on the product.

#### **Motivation**

To make people declare the assumptions that they are making. Also to make everyone on the project aware of assumptions that have been made.

#### **Examples**

Assumptions about new laws or political decisions.

Assumptions about what your developers expect to be ready in time for them to use. For example, other parts of your products, the completion of other projects, software tools, software components, etc.

Assumptions about the technological environment in which the product will operate. These assumptions should highlight areas of expected compatibility.

The software components that will be available to the developers.

Other products being developed at the same time as this one.

Availability and capability of bought-in components.

Dependencies on computer systems or people external to this project

## Considerations

We often make unconscious assumptions. It is necessary to talk to the members of the project team to discover any unconscious assumptions that they have made. Ask stakeholders (both technical and business-related) questions like “What software tools are you expecting to be available, will there be any new software products, are you expecting to use a current product in a new way, are there any business changes you are assuming we will be able to deal with....?” It is important to state these assumptions up front. You might also consider the probability of whether or not the assumption is correct, and where relevant, a list of alternatives if something that is assumed does not happen.

The assumptions are intended to be transient. That is, they should all be cleared by the time the specification is released. In other words, the assumption should have become either a requirement or a constraint. For example, if the assumption was about the capability of a product that is intended to be a partner product to yours, then the capability should have been proven satisfactory, and thus it becomes a constraint to use it. On the other hand, if the bought-in product is not suitable, then it becomes a retirement for the project team to construct the needed capability.

## 7 The Scope of the Work

---

### 7a. The context of the work.

#### Content

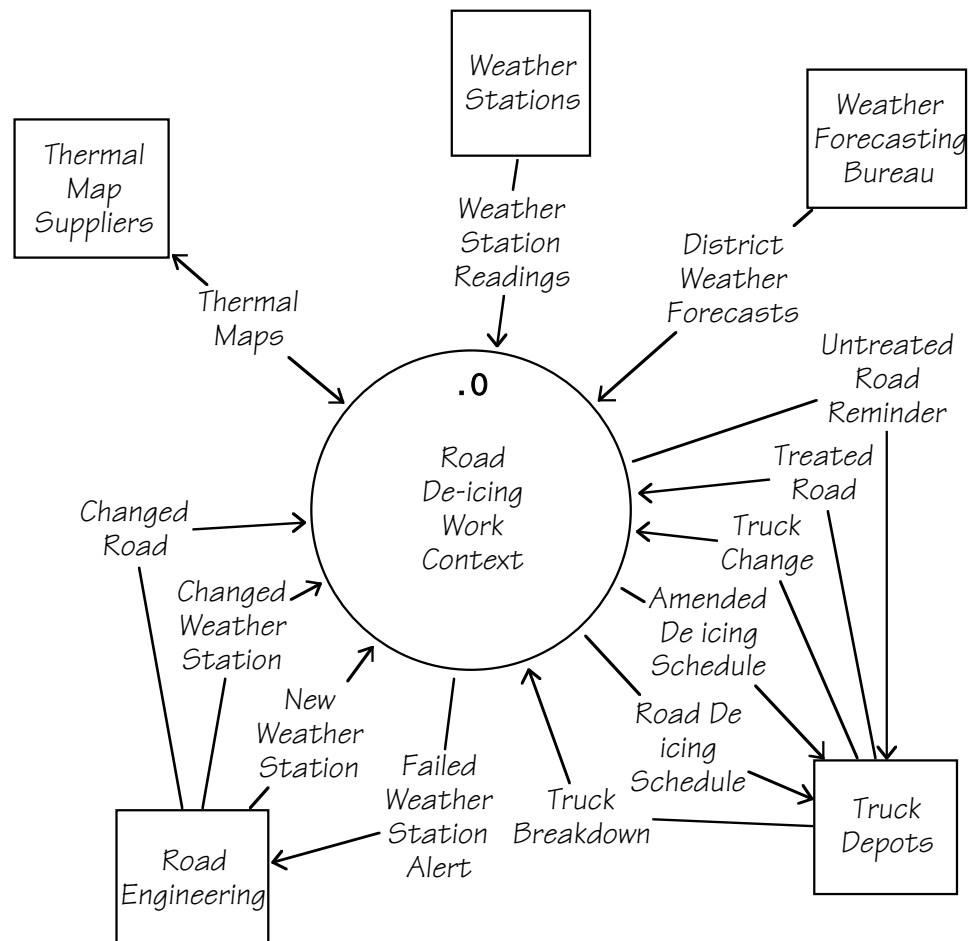
The work context diagram identifies the work that we need to investigate in order to be able to build the product. Note that this includes more than the intended product. Unless we understand the work that the product will support, there is little chance of building a product that will fit cleanly into its environment.

The adjacent systems on the example context diagram e.g. **Weather Forecasting Bureau**, indicate other subject matter domains (systems, people and organizations) that need to be understood. The interfaces between the adjacent systems and the work context indicate why we are interested in the adjacent system. In the case of **Weather Forecasting Bureau**, we can say that we are interested in the details of when, how, where, who and why they produce the **District Weather Forecast** information.

## Motivation

To clearly define the boundaries for the work study and requirements effort. Without this definition, there is little chance of building a product that will fit seamlessly into its environment.

## Examples



## Considerations

The names used on the context diagram should be consistent with the naming conventions discussed in section 5.

## 7b. Work partitioning

### Content

An event list, identifying all the business events to which the work responds. The business events are user-defined. The response to each event represents a portion of work that contributes to the total functionality of the work.

The event list includes:



Event Name

Input from other systems (identical with name on context diagram)

Output from other systems (identical with name on context diagram)

Internal objects/entities that are connected to this business event. For example, both events 8 and 9 would be connected to an internal object called road. In other words there is a need within the context to remember information about roads and that information is relevant to events 8 and 9 (and many other events as well). It is this identification of common internal objects that provides a link between events.

### **Motivation**

To identify logical chunks of the system that can be used as the basis for discovering detailed requirements. These business events also provide the subsystems that can be used as the basis for managing detailed analysis and design.

### **Example**

#### **Event List**

<b>Event Name</b>	<b>Input &amp; Output</b>
1. Weather Station transmits reading	Weather Station Readings (in)
2. Weather Bureau forecasts weather	District weather Forecast (in)
3. Road engineers advise changed roads	Changed Road (in)
4. Road Engineering installs new weather station	New Weather Station (in)
5. Road Engineering changes weather station	Changed Weather Station (in)
6. Time to test Weather Stations	Failed Weather Station Alert (out)
7. Truck Depot changes a truck	Truck Change (in) Amended De-icing Schedule (out)
8. Time to detect icy roads	Road De-icing Schedule (out)
9. Truck treats a road	Treated Road (in)
10. Truck Depot reports problem with truck	Truck Breakdown (in) Amended Gritting Schedule (out)
11. Time to monitor road gritting	Untreated Road Reminder (out)

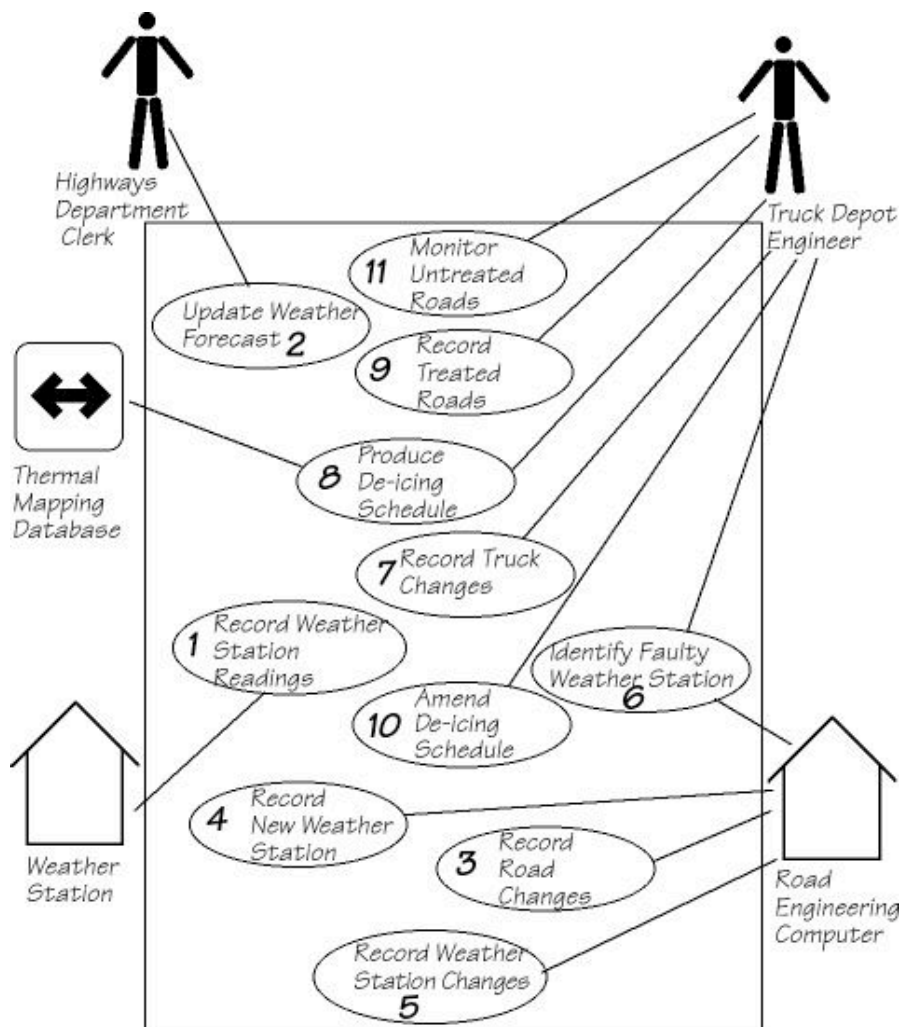
## Considerations

Attempting to list the business events is a way of testing the work context. This activity uncovers uncertainty and misunderstanding about the project and helps with precise communications.

## 8 The Scope of the Product

### 8a Product Boundary

Use case diagram identifies boundaries between the users and product.



You derive the product use cases by deciding where the product boundary should be for each one of the business events. These decisions are based on your knowledge of the work and the requirements constraints.

## 8b Use case list

The use case diagram is a graphical way of summarizing all the use cases relevant to the product. If you have a large number of use cases, we find 15-20, is around the limit, then it is better to list the use cases and model each one individually. For each use case on the list you should have: use case number, user/actor name, use case description and use case fit criterion. Also if you have built a use case description and/or any scenario models for this use case then this list can point to them.

### Use Case 8

#### User/actor name

Truck Depot Engineer

#### Description

Produce road de-icing schedule

#### Fit Criterion

Sensor readings shall be used to prepare a schedule for the de-icing trucks.

#### Use Case Scenarios

The description for this use case describes the normal way that it operates. Scenario models 8.1, 8.2, 8.3 illustrate exception cases for this use case.

Each of the individual requirements that relates to this use case will contribute to meeting the fit criterion of the use case. Each individual requirement will also have its own detailed fit criterion.

## 9 Functional and Data Requirements

---

### 9a. Functional Requirements.

#### Content

A specification for each individual functional requirement. As with all types of requirements, use the *Requirements Shell*. A full explanation is included in this template's introductory material.

#### Motivation

To specify the detailed functional requirements that must be supported by the system.

## Examples

Requirement #:	<b>75</b>	Requirement Type:	<b>9</b>	Event/use case #:	<b>7, 9</b>
Description:	<b>The product shall record all the roads that have been treated</b>				
Rationale:	<b>To be able to schedule untreated roads and highlight potential danger</b>				
Source:	<b>Arnold Snow - Chief Engineer</b>				
Fit Criterion:	<b>The recorded treated roads shall agree with the drivers' road treatment logs and shall be up to date within 30 minutes of the completion of the road's treatment</b>				
Customer Satisfaction:	<b>3</b>	Customer Dissatisfaction:	<b>5</b>		
Dependencies:	<b>All requirements using road and scheduling data</b>			Conflicts:	<b>105</b>
Supporting Materials:	<b>Work context diagram, terms definitions in section 5</b>				
History:	<b>Created February 29, 2000</b>				<small>Copyright © Atlantic Systems Guild</small>

### Fit Criterion

Each functional requirement must have a fit criterion. The fit criterion depends on the required action. For example, if the requirement is to record some data, then the fit criterion would say that the data must be able to be retrieved and must match certain standards. For calculations, the resulting data must conform to predicted results.

### Considerations

If you have produced an event/use case list (see 7b & 8a) you can use them to help you trigger the functional requirements for each event/use case. If you have not produced an event/use case list, give each functional requirement a unique number and, to help with traceability, they can be partitioned into event/use case-related groups later in the development process.

## 9b. Data requirements.

### Content

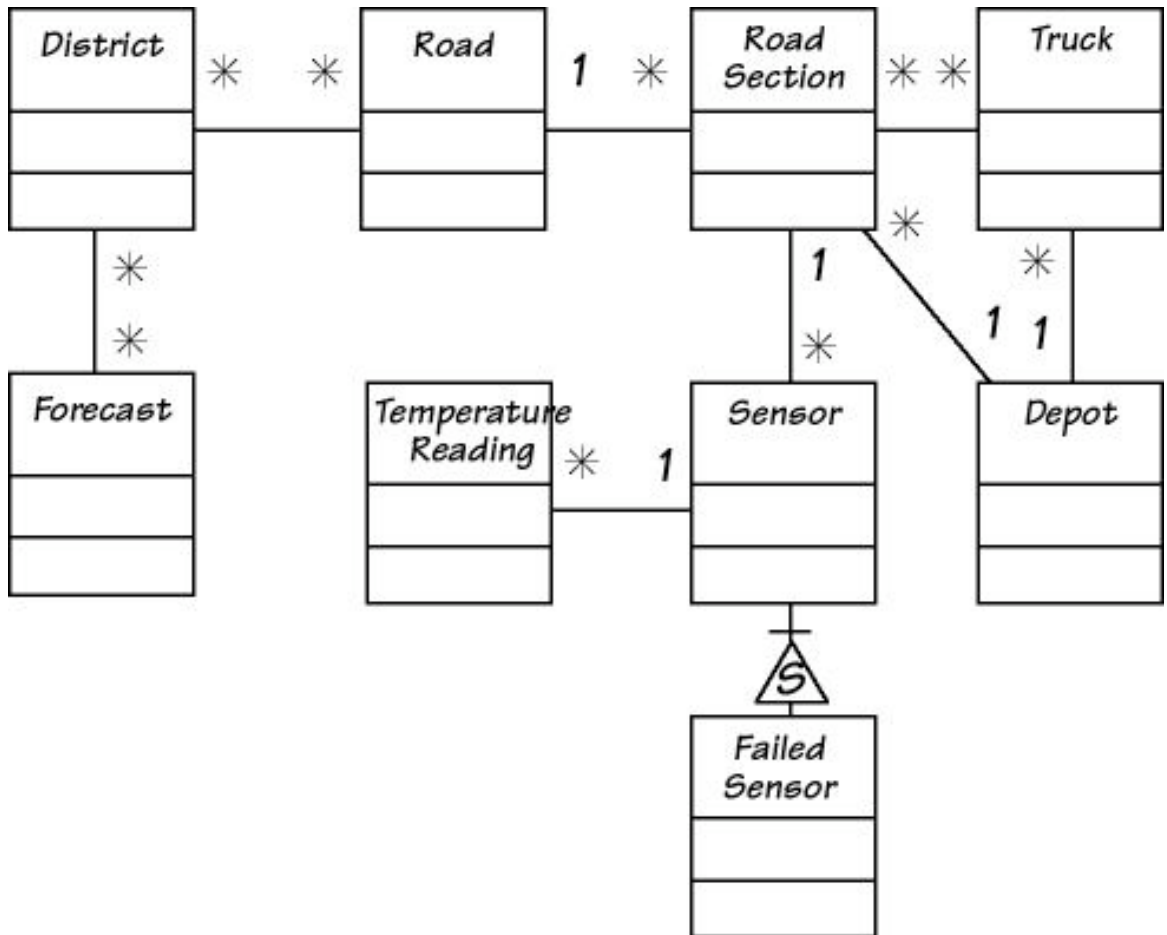
A specification of the essential subject matter/business objects/entities/classes that are germane to the system. This might take the form of a first-cut data model, an object model or a domain model. Or it might be adequately dealt with by defining the terms in the dictionary described in section 5. We have included examples of 2 notations for modeling business data, there are many others.

## Motivation

To clarify the system's subject matter and thereby trigger requirements that have not yet been thought of.

## Example 1

The following is a model of the system's business subject matter using the Unified Modeling Language (UML) class model notation.



## Fit Criterion

You can use any type of data or object model to capture this knowledge. The issue is to capture the meaning of the business subject matter and the connections between the individual parts and that you are consistent within your project. If you have an established company standard notation, then use that as it will help you to reuse knowledge between projects.

To support your data model you would also define:

Name of business object/entity (use naming convention from 5)

Statement of the purpose of the class/entity

Description of relationships between classes/entities  
Attributes of the object/entity (use conventions from 5)

### **Considerations**

Are there any data/object models for similar/overlapping systems that might be a useful starting point? Is there a domain model for the subject matter dealt with by this system?

## **10 Look and Feel Requirements**

---

### **10a. The interface**

#### **Content**

The section contains requirements relating to spirit of the interface. Your client may have given you particular demands such as corporate branding, style, colors to be used, degree of interaction and so on. This section captures the requirements for the interface rather than the design for the interface.

#### **Motivation**

To ensure that the appearance of the product conforms to the organization's expectations.

#### **Examples**

“The product shall comply with corporate branding standards.”

“The product shall be attractive to a teenage audience.”

“The product shall appear authoritative.”

#### **Considerations**

Interface design may overlap the requirements gathering process. This particularly true if you are using prototyping as part of your requirements process. As prototypes develop it is important to capture the requirements that relate to the look and feel. In other words, be sure that you understand your client's intentions for the product's look and feel. Record these as requirements instead of merely having a prototype to which the client has nodded his approval.

### **10b. The style of the product**

#### **Content**

A description of salient features of the product that are related to the way a potential customer will see the product. For example, if

your client wants the product to appeal to the business executive, then a look and feel requirement is that the product has a conservative and professional appearance. Similarly if the product is for sale to children, then the look and feel requirement is that it be colorful and look like it's intended for children.

You would also consider here the design of the package if this were to be a manufactured product. The package may have some requirements as to its size, style, and consistency with other packages put out by your organization, etc. Keep in mind the European laws on packaging. There is a requirement that the package not be significantly larger than the product it encloses.

The requirements that you record here will guide the designers to produce a product as envisioned by your client.

### **Motivation**

Given the state of today's market and people's expectations, we cannot afford to build products that have an inadequate appearance. Once the functional requirements are satisfied, it is often the appearance of products that determines whether they are successful or not. Your task in this section is to determine precisely how the product shall appear to its intended consumer.

### **Considerations**

The look and feel requirements specify the your client's vision of the product's appearance. The requirements may at first seem to be rather vague – “conservative and professional appearance” – but these will be quantified by their fit criterion. The fit criterion in this case give you the opportunity to extract from your client precisely what is meant, and gives the designer precise instructions on what he is to accomplish.

## **11 Usability Requirements**

---

### **11a. Ease of use.**

#### **Content**

This section describes your client's aspirations for how easy it will be for the intended users of the product to operate it. The product's usability is derived from the abilities of the expected users of the product and the complexity of its functionality.

The usability requirements should cover such things as:

- Efficiency of use – how quickly or accurately the user can use the product.

- Ease of remembering – how much is the casual user expected to remember about using the product
- Error rates – for some products it is crucial that the user commits very few, or no, errors.
- Overall satisfaction in using the product – this is especially important for commercial, interactive products where there is a lot of competition. Web sites are good example of this.

### **Motivation**

To guide the product’s designers into building a product that will meet the expectations of its eventual users.

### **Examples**

“The product shall be easy for 11 year-old children to use.”

“The product shall help the user to avoid making mistakes.”

“The product shall make the users want to use it.”

“The product shall be used by people with no training, and possibly no understanding of English.”

### **Fit Criterion**

These examples may seem simplistic, but they do express the intention of the client. To completely specify what is meant by the requirement it is necessary to add a measurement of acceptance. We call this a fit criterion. The fit criterion for the above examples would be:

[An agreed percentage, say 90%] of a test panel of 11 year olds shall be able to successfully complete [list of tasks] within [specified time]

One month’s use of the product shall result in a total error rate of less than [an agreed percentage, say 2%]

An anonymous survey shall show that [an agreed percentage, say 75%] of the users are regularly using the product after [an agreed time] familiarization period.

### **Considerations**

Refer back to Section 3, the Users of the System, to ensure that you have considered the usability requirements from the perspective of all the different types of users.

It may be necessary to have special consulting sessions with your users and your client to determine whether there are any special usability considerations that must be built into the product.



You could also consider consulting a usability laboratory that has experience with testing the usability of products that have constraints (sections 1-7 of this template) similar to yours.

## **11b. Personalization and internationalization requirements**

### **Content**

This section describes the way in which the product can be altered or configured to take into account the user's personal preferences or choice of language.

The personalization requirements should cover such things as:

- Languages, spelling preferences, language idioms
- Currencies including the symbols and decimal conventions
- Personal configuration options – there are a myriad of these

### **Motivation**

To ensure that the product's users do not have to struggle with, or meekly accept, the cultural conventions of the builder.

### **Examples**

“The product shall retain the buyer's buying preferences.”

“The product shall allow the user to select a chosen language.”

### **Considerations**

Consider the locations of the potential customers and users of your product. Any out of country users will welcome the opportunity to convert to their home spelling and expressions.

By allowing users to customize the way in which they use the product, you are giving them the opportunity to participate more closely with your organization, as well as give them their own personal user experience.

You might also consider the configurability of the product. This allows different users to have different functional variations of the product.

## **11c. Ease of learning.**

### **Content**

A statement of how easy it should be to learn to use the product. This will range from zero time for products intended for placement in the public domain (for example a parking meter or a web site) to

a considerable time for complex, highly technical products. (We know of one product where it was necessary for graduate engineers to spend 18 months in training before being qualified to use the product.)

### **Motivation**

To quantify the amount of time that your client feels is allowable before a user can successfully use the product. This requirement will guide designers in how users will learn the product. For example, the designers may build elaborate interactive help facilities into the product, or the product may be packaged with a tutorial. Alternatively the product may have to be constructed so that all of its functionality is apparent upon first encountering it.

### **Examples**

“The product shall be easy for an engineer to learn.”

“A clerk shall be able to be productive within a short time.”

“The product shall be able to be used by members of the public who will receive no training before using it.”

“The product shall be used by engineers who will attend 5 weeks of training before using the product.”

### **Fit Criterion**

Fit criterion for the above example requirements are:

An engineer shall produce a [specified result] within [specified time] of beginning to use the product, without needing to use the manual.

After receiving [number of hours] training a clerk shall be able to produce [quantity of specified outputs] per [unit of time].

[Agreed percentage] of a test panel shall successfully complete [specified task] within [specified time limit].

The engineers shall achieve [agreed percentage] pass rate from the final examination of the training.

### **Considerations**

Refer back to Section 3, the Users of the System, to ensure that you have considered the ease of learning requirements from the perspective of all the different types of users.

## **11c. Accessibility requirements.**

### **Content**

The requirements for how easy it should be for people with common disabilities to access the product. These disabilities might

be to do with sight, physical disablement, hearing, cognitive, or others.

### **Motivation**

In many countries it is required that some products are made available to the disabled. In any event, it seems self-defeating to exclude this sizable community of potential customers.

### **Examples**

“The product shall be usable by partially-sighted users.”

“The product shall conform to the Americans with Disabilities Act.”

### **Considerations**

There are users with disabilities other than the commonly-described ones. Similarly, there are partial disabilities that are fairly common. A simple, and not very consequential example, is that approximately 20% of males are red-green color blind.

## **12 Performance Requirements**

---

### **12a. Speed and latency requirements**

#### **Content**

Specifies the amount of time available to complete specified tasks. These often refer to response times. They can also refer to the product's ability to fit into the intended environment.

#### **Motivation**

Some products, usually real-time products, must be able to perform some of their functionality within a given time slot. Failure to do so may mean catastrophic failure (for example a ground-sensing radar in an airplane fails to detect an upcoming mountain) or the product will not cope with the required volume of use (an automated ticket selling machine).

#### **Examples**

“Any interface between a user and the automated system shall have a maximum response time of 2 seconds”

“The response shall be fast enough to avoid interrupting the user's flow of thought”

“The product shall poll the sensor every 10 seconds”

“The product shall download the new status parameters within 5 minutes of a change”

### **Fit Criterion**

Unit of measurement

Required range of values

### **Considerations**

There is a wide variation in the importance of different types of speed requirements. If you are working on a missile guidance system then speed is extremely important. On the other hand, an inventory control report that is run once every 6 months has very little need for split second speed.

Customize this section of the template to give examples of the speed requirements that are important within your environment.

## **12b. Safety critical requirements**

### **Content**

Quantification of perceived risk of possible damage to people, property and environment.

### **Motivation**

To understand and highlight the potential damage that could occur when using the product within the expected operational environment.

### **Examples**

“The product shall not emit noxious gases that damage people’s health.”

“The heat exchanger shall be shielded from human contact.”

### **Fit Criterion**

Description of the perceived risk

Factors that could cause the damage

Unit for measuring the factors that could cause the damage

“The product shall be certified to comply with the Health Department’s standard E110-98. This is to be certified by qualified testing engineers.”

“No member of a test panel of [specified size] shall be able to touch the heat exchanger. The heat exchanger must also comply with safety standard [specify which one].”

## **Considerations**

The sample requirements given above apply to some, but not all, products. It is not possible to give examples of every variation of safety critical requirement. To make the template work in your environment, you should customize it by adding examples that are specific to your products.

If you are building safety critical systems then the relevant safety critical standards are already well specified. You will likely have safety experts on your staff. These safety experts are the best source of the relevant safety critical requirements for your type of product. The safety experts will almost certainly have copious information that you can use.

Consult your legal department. They will be aware of the kinds of lawsuits that have resulted from product safety failure. This is probably the best starting place for generating relevant safety requirements.

## **12c. Precision requirements**

### **Content**

Quantification of the desired accuracy of the results produced by the product.

### **Motivation**

To set the client and user expectations for the precision of the product.

### **Examples**

“All monetary amounts shall be accurate to 2 decimal places.”

“Accuracy of road temperature readings shall be within + or - 2 degrees centigrade.”

### **Fit Criterion**

Unit of measure plus degree of precision

### **Considerations**

If you have done any detailed work on definitions, then some precision requirements might be adequately defined by definitions in section 5.

## **12d. Reliability and Availability requirements**

### **Content**

This section quantifies the necessary reliability of the product. This is usually expressed as the allowable time between failures, or the total allowable failure rate.

It also quantifies the expected availability of the product.

### **Motivation**

It is critical for some products not to fail too often. This section allows you to explore the possibility of failure and to specify realistic levels of service. It also gives you the opportunity to set client and user expectations about the amount of time that the product will be available for use.

### **Examples**

“The product shall be available for use 24 hours per day, 365 days per year.”

“The product shall be available for use between the hours of 8:00am and 5:30pm.”

“The escalator shall run from 6am until the last flight arrives at 10pm.”

“The product shall achieve 99% up time.”

### **Considerations**

Consider carefully whether the real requirement for your product is that it is available for use, or that it does not fail at any time.

Consider also the cost of reliability and availability, and whether it is justified for your product.

## **12e. Robustness requirements**

### **Content**

Robustness specifies the ability of the product to continue to function under abnormal circumstances.

### **Motivation**

To ensure that the product is able to provide some or all of its services when there is some abnormal happening in its environment.

### **Examples**

“The product shall continue to operate in local mode whenever it loses its link to the central server.”

“The product shall provide 10 minutes of emergency operation should it become disconnected from the electricity source.”

### **Considerations**

Abnormal happenings can almost be considered normal. Our products are so large and complex that there is a good chance that at any given time, one component will not be functioning correctly. Robustness requirements are intended to prevent total failure of your product.

You could also consider disaster recovery in this section.

## **12f. Capacity requirements**

### **Content**

This section specifies the volumes that the product must be able to deal with and the numbers of data stored by the product.

### **Motivation**

To ensure that the product is capable of processing the expected volumes.

### **Examples**

“The product shall cater for 300 simultaneous users within the period from 9:00am to 11:am. Maximum loading at other periods will be 150.”

“During a launch period the product shall cater for up to 20 people to be in the inner chamber.”

### **Fit Criterion**

In this case, the requirement description is quantified, and thus can be tested.

## **12g. Scalability or extensibility requirements**

### **Content**

This specifies the expected increases in size that the product must be able to handle. As business grow (or are expected to grow) our software products must increase their capacities to cope with the new volumes.

### **Motivation**

To ensure that the designers allow for future capacities.

### **Examples**

“The product shall be capable of processing the existing 100,000 customers. This number is expected to grow to 500,000 within three years.”

“The product shall be able to process 50,000 transactions an hour within two years of its launch.”

## 13 Operational Requirements

---

### 13a. Expected physical environment

#### Content

This section specifies the physical environment in which the product will operate.

#### Motivation

To highlight conditions that might need special requirements, preparations or training. These requirements ensure that the product is fit to be used in its intended environment.

#### Examples

“The product shall be used by a worker, standing up, outside in cold, rainy conditions.”

“The product shall be used in noisy conditions with a lot of dust.”

“The product shall be able to fit in a pocket or purse.”

“The product shall be usable in dim light.”

#### Considerations

The work environment Is the product to operate in some unusual environment? Does this lead to special requirements? Also see section 11 - Usability.

### 13b. Expected technological environment

#### Content

Specification of the hardware and other devices that make up the operating environment for the new system.

#### Motivation

To identify all the components of the new system so that the acquisition, installation and testing can be effectively managed.

#### Considerations

Describe the hardware and other devices that make up the operating environment for the new system. This may not be known at the



time of the requirements process, as these devices may be decided at design time.

It may be that the operating environment is complex, and becomes a subject of requirements study itself.

Special considerations should also be given if the product is to be embedded in a device.

If the expected operating environment is the same or similar to the current one, then this might be adequately covered in section 4b - Implementation Environment of the Current System.

### **13c. Partner applications**

#### **Content**

Description of other applications that the product must interface with.

#### **Motivation**

Requirements for interfacing to other applications often remain undiscovered until implementation time. Avoid a high degree of rework by discovering these requirements early.

#### **Examples**

“We must be able to interface with any html browser.”

“The new version of the spreadsheet must be able to access data from the previous 2 versions”

“Our product must interface with the applications that run on the remote weather stations”

#### **Fit Criterion**

For each inter-application interface specify:

- The data content
- The physical material content
- The medium that carries the interface
- The frequency
- The volume

### **13d. Productization Requirements**

#### **Content**

Any requirements that are necessary to make the product into a distributable or saleable item. It is also appropriate to describe here the operations to be performed to have a software product successfully installed.

### **Motivation**

To ensure that if work has to be done to get the product out the door, then it becomes part of the requirements.

### **Examples**

“The product shall be distributed as a ZIP file.”

“The product shall be able to be installed by an untrained user without recourse to separately-printed instructions.”

“The product shall be of a size that it can fit onto one CD.”

### **Considerations**

Some products have special needs to turn them into a saleable, or usable product. You might consider that the product has to be protected such that only paid-up customers can access it. This might be implemented as a dongle, a daily keyword, a check that no other copy of the product is running on the network at the same time.

Most commercial products have some needs in this area.

## **14 Maintainability and Support Requirements**

---

### **14a. How easy must it be to maintain this product?**

#### **Content**

A quantification of the time necessary to make specified changes to the product.

#### **Motivation**

To make everyone aware of the maintenance needs of the product.

#### **Examples**

“New MIS reports must be available within one working week of the date the requirements are agreed”

“A new weather station must be able to be added to the system overnight”

#### **Considerations**

There may be special requirements for maintainability, such as this product must be able to be maintained by its end-users, or developers who are not the original developers. This has an effect on the way that the product is developed, and there may be additional requirements for documentation or training.

You might also consider writing testability requirements in this section.

#### **14b. Are there special conditions that apply to the maintenance of this product?**

##### **Content**

Specification of the intended release cycle for the product and the form that the release will take.

##### **Motivation**

To make everyone aware of how often it is intended to produce new releases of the product.

##### **Examples**

“The maintenance releases will be offered to end-users once a year.”

“Every registered user will have access to our help site via the Internet.”

##### **Fit Criterion**

Description of type of maintenance + amount of effort budgeted

##### **Considerations**

Do you have any existing contractual commitments or maintenance agreements that might be affected by the new system?

#### **14c. Supportability**

##### **Content**

This specifies the level of support that the product requires. This is often done using a help desk. If there are to be people who provide support for the product, is that considered part of the product and are there any requirements for that support. You might also build support into the product itself, in which case this is the place to write those requirements.

##### **Motivation**

To ensure that the support aspect of the product is adequately specified.

##### **Considerations**

Consider the anticipated level of support, and what forms it might take. For example, there may be a constraint that there is to be no printed manual. Or you might consider that the product is to be entirely self-supporting.

## 14d. Portability requirements

### Content

Description of other platforms or environments to which the product must be ported.

### Motivation

To quantify client and user expectations about the platforms on which the product will be able to run.

### Examples

“The product is expected to run under Windows XP and Linux”

“The product might eventually be sold to the Japanese market”

“The product is designed to run in offices, but we intend to have a version which will run in restaurant kitchens”

### Fit Criterion

Specification of system software on which the product must operate.

Specification of future environments in which the product is expected to operate.

Time allowed to make the transition.

### Considerations

Ask questions from your marketing department to discover unstated assumptions that have been made about the portability of the product.

## 15 Security Requirements

---

### 15a. Access requirements

#### Content

Specification of who has authorized access to the product, and under what circumstances that access is granted, and to what parts of the product access is allowed.

#### Motivation

To understand the expectations for confidentiality aspects of the system.

#### Examples

“Only direct managers can see the personnel records of their staff.”

“Only holders of current security clearance can enter the building.”

### **Fit Criterion**

System function name or system data name

User role/s and/or names of people who have clearance

### **Considerations**

Is there any data that is sensitive to the management? Is there any data that low-level users do not want management to have access to? Are there any processes that might cause damage or might be used for personal gain? Are there any people who should not have access to the system?

Avoid solving how you will design a solution to the security requirements. For instance, don't design a password system. Your aim here is to identify what the security requirement is. The design will come from this description.

Consider asking for help. Computer security is a highly-specialized field, and one where improperly-qualified people have no business being. If your product has need of more than average security, we advise that you make use of a security consultant. They are not cheap, but the results of inadequate security can be even more expensive.

## **15b. Integrity requirements**

### **Content**

Specification of the required integrity of databases and other files, and of the product itself.

### **Motivation**

To understand the expectations for the integrity of the system's data. To specify what the product will do to insure its integrity in the case of an unwanted happening such as attack from the outside of unintentional misuse by an authorized user.

### **Examples**

“The product shall prevent its data from incorrect data being introduced.”

“The product shall protect itself from intentional abuse.”

### **Considerations**

Organizations rely more and more on their stored data. If this data should be come corrupt or incorrect, or indeed disappear, then it could be fatal. For example, it is true that almost half of small businesses go bankrupt after a fire destroys their computer systems.

Integrity requirements are aimed at preventing complete loss, as well as corruption, of data and processes.

## **15c. Privacy requirements**

### **Content**

Specification of what the product has to do to insure the privacy of individuals that it stores information about. The product must also ensure that all laws about privacy of individual's data are observed.

### **Motivation**

To ensure that the product complies with the law, and to protect the individual privacy of your customers. Few people today look kindly on organizations that do not observe their privacy.

### **Examples**

“The product shall make its user aware of its information practices before collection data from them.”

“The product shall notify customers of changes to its information policy.”

“The product shall reveal private information only in compliance with the organization's information policy.”

“The product shall protect private information in accordance with relevant privacy laws / the organization's information policy.”

### **Considerations**

Privacy may well have legal implications, and you are advised to consult with your organization's legal department about the requirements to be written in this section.

Consider what notices are required to be issued to your customers before collecting personal information. This can go so far as to warn them if you intend to put a cookie in their computer. Also, do you have to do anything to keep the customer aware that you hold personal information?

The customer must always be in a position to give or withhold consent when private data is collected or stored. Similarly, the customer should be able to view any private data, and where appropriate, ask for correction of the data.

Also consider the integrity and security of private data. A common example of this is the storage of credit card information.

## **15d. Audit requirements**

### **Content**

Specification of what the product has to do (usually retain records) to permit the required audit checks.

### **Motivation**

To build a system that complies with the appropriate audit rules.

### **Considerations**

This section may have legal implications. You are advised to seek the approval of your organization's auditors for what you write here.

You should also consider whether the product should retain information on who has used it. The intention is to provide security in the form that a user may not later deny having used the product, or participated in some form of transaction using the product.

## **15e. Immunity requirements**

### **Content**

The requirements for what the product has to do to protect itself from infection by unauthorized or undesirable software programs, such as viruses, worms, Trojan horses and others.

### **Motivation**

To build a product that is as secure as possible from malicious interference.

### **Considerations**

Each day brings more malevolence from the unknown, outside world. People buying software, or any other kind of product, expect that it can protect itself from outside interference,

## **16 Cultural and Political Requirements**

---

### **Are there any special factors about the product that would make it unacceptable for some political reason?**

#### **Content**

This section contains requirements that are specific to the sociological and political factors that affect the acceptability of the product. If you are developing a product for foreign markets then these requirements are particularly relevant.

## **Motivation**

To bring out in the open requirements that are difficult to discover because they are outside the cultural experience of the developers.

In the case of political requirements the requirements sometimes appear irrational.

## **Examples**

“The product shall not be offensive to religious and ethnic groups.”

“The product shall be able to distinguish between French, Italian and British road numbering systems.”

“The product shall keep a record of public holidays for all countries in the European Union and for all states in the United States.”

## **Considerations**

Question whether the product is intended for a culture other than the one with which you are familiar. Ask whether people in other countries or in other types of organizations will use the product. Do these people have different habits, holidays, superstitions, cultural norms that do not apply to your own culture?

Did you intend to develop the product on a Macintosh, when the office manager has laid down a edict that only Windows machines are permitted?

Is a director also on the board of a company that manufactures products similar to the one that you intend to build?

Whether you agree with these political requirements has little bearing on the outcome. The reality is that the system has to comply with political requirements even if you can find a better/more efficient/more economical solution. A few probing questions here may save some heartache later.

# **17 Legal Requirements**

---

## **17a. Does the system fall under the jurisdiction of any law ?**

### **Content**

A statement specifying the legal requirements for this system..

### **Motivation**

To comply with the law so as to avoid later delays, law suits and legal fees.



## **Examples**

“Personal information shall be implemented so as to comply with the data protection act.”

## **Fit Criterion**

Lawyers’ opinion that the product does not break any laws.

## **Considerations**

Consider consulting lawyers to help identify the legal requirements?

Are there any copyrights that must be protected? Alternatively, do any competitors have copyrights that you might be in danger of infringing?

Is it a requirement that developers have not seen competitors’ code or even have worked for competitors?

Is there any pending legislation that might affect the development of this system?

## **17b. Are there any standards with which we must comply ?**

### **Content**

A statement specifying applicable standards and referencing detailed standards descriptions.

### **Motivation**

To comply with standards so as to avoid later delays.

### **Example**

“The product shall comply with MilSpec standards.”

“The product shall comply with insurance industry standards”.

“The product shall be developed according to SSADM standard development steps.”

### **Fit Criterion**

The appropriate standard-keeper that the standard has been adhered to.

### **Considerations**

It is not always apparent that there are applicable standards because their existence is often taken for granted. Consider the following:

Are there any industry bodies that have applicable standards?

Has the industry a code of practice, watchdog or ombudsman?

Are there any special development steps for this type of product?

## 18 Open Issues

---

### **Issues that have been raised and do not yet have a conclusion.**

#### **Content**

A statement of factors that are uncertain and might make significant difference to the product.

#### **Motivation**

To bring uncertainty out in the open and provide objective input to risk analysis.

#### **Examples**

“Our investigation into whether or not the new version of the processor will be suitable for our application is not yet complete.”

“The government are planning to change the rules about who is responsible for gritting the motor ways, but we do not know what the changes might be.”

#### **Considerations**

Are there any issues that have come up from the requirements gathering that have not yet been resolved? Have you heard of any changes that might occur in the other organizations/systems on your context diagram? Are there any legislative changes that might affect your system? Any rumors about your hardware/software suppliers that might have an impact?

## 19 Off-the-Shelf Solutions

---

### **19a. Is there a ready-made system that could be bought?**

#### **Content**

List of existing products that should be investigated as potential solutions. Reference any surveys that have been done on these products.

#### **Motivation**

To give consideration to whether or not a solution can be bought.

#### **Considerations**

Is it possible to buy something that already exists or is about to become available? It may not be possible at this stage to say with a lot of confidence, but any likely products should be listed here.

Also consider whether there are products that must not be used.

### **19b. Can ready-made components be used for this product?**

#### **Content**

Description of the candidate components, either bought-in or built by your company, that could be used by this project. List libraries that could be a source of components.

#### **Motivation**

Reuse rather than reinvention.

### **19.3 Is there something that we could copy?**

#### **Content**

List of other similar systems.

#### **Motivation**

Reuse rather than reinvention.

#### **Examples**

“Another electricity company has built a customer service system. Their hardware is different from ours but we could buy their specification and cut our analysis effort by approximately 60%.”

#### **Considerations**

While a ready-made solution may not exist, there may well be something that, in its essence, is similar enough that you could copy, and possibly modify, to better effect than starting from scratch. This is dangerous because it relies on the base system being of good quality.

This question should always be answered. The act of answering will force you to look at other existing solutions to similar problems.

## 20 New Problems

---

### **20a. What problems could the new system cause in the current environment?**

#### **Content**

A description of how the new system will affect the current implementation environment. This section should also cover things that the new product should *not* do.

#### **Motivation**

The intention is to discover early any potential conflicts that might otherwise not be realized until implementation time.

#### **Examples**

Any change to the scheduling system will affect the work of the engineers in the divisions and the truck drivers.

#### **Considerations**

Is it possible that the new system will damage some already existing system? Can people be displaced, or affected by the new system?

This requires a study of the current environment. A model highlighting the effects of the change is a good way to make this information widely understandable.

### **20b. Will the new development affect any of the installed system?**

#### **Content**

Specification of the interfaces between new and existing systems.

#### **Motivation**

Very rarely is a new development intended to stand completely alone. Usually there is some existing system that the new one must coexist with. This question forces you to look carefully at the existing system and examine it for potential conflicts with the new development.

### **20c. Will any of our existing users be adversely affected by the new development?**

#### **Content**

Details of any adverse reaction that might be suffered by existing users

### **Motivation**

Sometimes existing users are using a product in such a way that they will suffer ill effects from the new system/feature. Identify any likely adverse user reaction, determine whether we care and what precautions we will take.

### **20d. What limitations exist in the anticipated implementation environment that may inhibit the new system?**

#### **Content**

Statement of any potential problems with the new automated technology or new ways of structuring the organization.

#### **Motivation**

The intention is to make early discovery of any potential conflicts that might otherwise not be realized until implementation time.

#### **Examples**

The planned new server is not powerful enough to cope with our projected growth pattern.

#### **Considerations**

This requires a study of the intended implementation environment.

### **20e. Will the new system create other problems?**

#### **Content**

Identification of situations that we might not be able to cope with.

#### **Motivation**

To guard against situations where the product might fail.

#### **Considerations**

Will we create a demand for our product that we are not able to service? Will the new system cause us to fall foul of laws that do not currently apply? Will the existing hardware cope?

There are potentially hundreds of unwanted effects. It pays to answer this question very carefully.

## 21 Tasks

---

### 21a. What steps have to be taken to deliver the system?

#### Content

Details of the life cycle and approach that will be used to deliver the product. A high level process diagram showing the tasks and interfaces between them is a good way to communicate this information.

#### Motivation

To specify the approach that will be taken to deliver the product so that everyone has the same expectations.

#### Considerations

Depending on the level of maturity of your process, the new product will be developed using your standard approach. However, there are some circumstances that are special to a particular product and will necessitate changes to your lifecycle. While these are not a product requirement, they are needed if the product is to be successfully developed.

If possible, attach an estimate of the time and resources need for each task based on the requirements that you have specified. Tag your estimates to the events/use cases/functions that you specified in sections 8 and 9.

Do not forget data conversion, user training and cutover. We have listed these because they are usually ignored when projects set implementation dates.

### 21b. Development phases

#### Content

Specification of each phase of development and the components in the operating environment.

#### Motivation

To identify the phases necessary to implement the operating environment for the new system so that the implementation can be managed.

#### Fit Criterion

Name of the phase

Required operational date

Operating environment components included

Functional requirements included

Non-functional requirements included

### **Considerations**

Identify which hardware and other devices are necessary for each phase of the new system. This may not be known at the time of the requirements process, as these devices may be decided at design time.

## **22 Cutover**

---

### **22a. What special requirements do we have to get the existing data, and procedures to work for the new system?**

#### **Content**

A list of the Cutover activities. Timetable for implementation.

#### **Motivation**

To identify cutover tasks as input to the project planning process.

#### **Considerations**

Will you be using phased implementation to install the new system? If so, describe the requirements that will be implemented by each of the major phases.

What data conversion has to be done? Are there special programs to be written to transport data from an existing system to the new one? If so, the requirements for this program(s) are to be described here.

What manual backup is needed while the new system is installed?

When are each of the major components to be put in place, when are phases of the implementation to be released?

This section is the timetable for implementation of the new system.

### **22b. What data has to be modified/translated for the new system?**

#### **Content**

List of data translation tasks.

#### **Motivation**

To discover missing tasks that will affect the size and boundaries of the project.

## Fit Criterion

Description of the current technology that holds the data

Description of the new technology that will hold the data

Description of the data translation task/s

Foreseeable problems

## Considerations

Every time you make an addition to your dictionary (see section 5) ask the question “What are all the places that this data is currently held and will the new system affect that implementation?”.

## 23 Risks

---

All projects involve risk. By this we mean the risk that something will go wrong. Risk is not necessarily a bad thing, as no progress is made without taking some risk. However, there is a difference between unmanaged risk – say shooting dice at a craps table – and managed risk where the probabilities are well understood, and contingencies made. Risk is only a bad thing if the risks are ignored and they become problems. Risk management is assessing which risks are most likely to apply to the project, deciding a course of action if they become problems, and monitoring projects to give early warnings of risks becoming problems.

This section of your specification should contain a list of the most likely and the most serious risks for your project. Against each risk include the probability of that risk becoming a problem. Capers Jones’ book *Assessment and Control of Software Risks*. Prentice-Hall, Englewood Cliffs, NJ. 1994 gives comprehensive lists of risks and their probabilities, you can use these as a starting point. For example, Jones cites the following risks as being the most serious:

- Inaccurate metrics
- Inadequate measurement
- Excessive schedule pressure
- Management malpractice
- Inaccurate cost estimating
- Silver bullet syndrome
- Creeping user requirements
- Low quality
- Low productivity



- Cancelled projects

It is also useful input to project management if you include the impact on the schedule, or the cost, if the risk does become a problem.

## 24 Costs

---

The other cost of requirements is the amount of money or effort that you have to spend building them into a product. Once the requirements specification is complete, you can use one of the estimating methods to assess the cost, and express this in a monetary amount or time to build.

There is no best method to use when estimating. However your estimates should be based on some tangible, countable, artifact. If you are using this template then, as a result of doing the work of requirements specification, you are producing many measurable deliverables. For example:

Number of input and output flows on the work context

Number of business events

Number of product use cases

Number of functional requirements

Number of non-functional requirements

Number of requirements constraints

Number of function points

The more detailed work you do on your requirements the more accurate will be your deliverables. Your cost estimate is the amount of resources you estimate each type of deliverable will take to produce within your environment. You can do some very early cost estimates based on the work context. At that stage, your knowledge of the work will be general and you should reflect this by making the cost estimate a range rather than one figure.

As you get more knowledge about the requirements we suggest you try using function point counting – not because it is an inherently superior method - but because it is so commonly accepted. So much is known about it, that it is possible to make easy comparisons with other products, and other installations' productivity.

It is important that your client knows at this stage what the product is likely to cost. You usually express this as a total cost to complete

the product, but you may also find it advantageous to be able to point out the cost of individual requirements.

Whatever you do, do not leave the costs in the lap of hysterical optimism. Make sure that this section includes meaningful numbers based on tangible deliverables.

## 25 User Documentation and Training

---

### 25a. The plan for building the user documentation.

#### **Content**

List of the user documentation that will be supplied as part of the system and to describe the training that will be available.

#### **Motivation**

To set expectations for the documentation and training and to identify who will be responsible for creating it.

#### **Considerations**

What level of documentation is expected? Will the users be involved in the production of the documentation? Who will be responsible for keeping the documentation up to date? What form will the documentation take? What training will be necessary? Who will design the training? Who will provide the training?

## 26 Waiting Room

---

Requirements that will not be part of the agreed product. These requirements might be included in future versions of the product.

#### **Content**

Any type of requirement.

#### **Motivation**

To allow requirements to be gathered, even though they cannot be part of the current development. To ensure that good ideas are not lost.

#### **Considerations**

The requirements gathering process often throws up requirements that are beyond the sophistication of, or time allowed for, the current release of the product. This section is a hold-all for requirements in waiting. The intention is to avoid stifling your users

and clients by having a repository of future requirements. You are also managing expectations by making it clear that you take these requirements seriously but they will not be part of the agreed product.

Many people use the waiting room as a way of planning future versions of the product. Each requirement in the waiting room is tagged with its intended version number.

## 27 Ideas for Solutions

---

When you are gathering requirements you are focusing on finding out what the real requirements are, you are trying to avoid coming up with solutions. However when creative people start to think about a problem they always have ideas. This section of the template is a place to put those ideas so that you do not forget them and so that you can separate them from the real business requirements.

### **Content**

Any idea for a solution that you think is worth keeping for future consideration. This can take the form of rough notes, sketches, pointers to other documents, pointers to people, pointers to existing products....the aim is to capture, with the least amount of effort, an idea that you can come back to later.

### **Motivation**

To make sure that good ideas are not lost and to help you separate requirements and solutions.

### **Considerations**

Whilst you are gathering requirements you will have solution ideas, this is a way of capturing them. Bear in mind that this section will not necessarily be published as part of the specification that you publish.