Volere

Requirements Specification Template

Edition 16—2012

by James Robertson & Suzanne Robertson
principals of the Atlantic Systems Guild

The Volere Requirements Specification Template is intended for use as a basis for your requirements specifications. The template provides sections for each of the requirements types appropriate to today's software systems. You may download the template from the Volere site and adapt it to your requirements gathering process and requirements tool. The template can be used with Requisite, DOORS, Caliber RM, IRqA and other popular tools see http://www.volere.co.uk/tools.htm

The template may not be sold, or used for commercial gain or purposes other than as a basis for a requirements specification without prior written permission. The Template may be modified or copied and used for your requirements work, provided you include the following copyright notice in any document that uses any part of this template:

Contents

Project Drivers

Project Constraints

Functional Requirements

Non-functional Requirements

Project Issues

The Volere Requirements Knowledge Model (included with the download of Version 16 of this template) shows the formal structure and cross references between the components in the above table of contents.

Volere

Volere is the result of many years of practice, consulting, and research in requirements engineering and business analysis. We have packaged our experience in the form of a generic requirements process, requirements training, requirements consultancy, requirements audits, a variety of downloadable guides and articles, a requirements knowledge model and this requirements template. We also provide requirements specification-writing services.

The first edition of the Volere Requirements Specification Template was released in 1995. Since then, organizations from all over the world have saved time and money by using the template as the basis for discovering, organizing, and communicating their requirements.

The Volere web site www.volere.co.uk contains articles about the Volere techniques, experiences of Volere users and case studies, requirements tools, and other information useful to requirements practitioners.

The Volere requirements process is described in the book *Mastering the Requirements Process—Second Edition* by Suzanne Robertson and James Robertson, Addison-Wesley, 2006. ISBN 0-321-41949-9

For more about managing requirements see *Requirements Led Project Management* by Suzanne Robertson and James Robertson, Addison-Wesley, 2005. ISBN 0-321-65904-X

Updates to this template and instructions for downloading are available at
http://www.volere.co.uk

Public seminars on Volere are run on a regular basis in Europe, the United States, Australia, and New Zealand. For a schedule of courses, refer to www.volere.co.uk.

Requirements Types

For ease of use, we have found it convenient to think of requirements as belonging to a type. There are two reasons for the type: as an aid to finding the requirements, to be able to group the requirements that are relevant to a specific expert specialty.

Functional requirements are the fundamental or essential subject matter of the product. They describe what the product has to do or what processing actions it must take.

Non-functional requirements are the properties that the functions must have, such as performance and usability. Do not be deterred by the unfortunate name for this kind of requirements, they are as important as the functional requirements for the product's success.

Project constraints are restrictions on the product due to the budget or the time available to build the product.

Design constraints impose restrictions on how the product must be designed. For example, it might have to be implemented in the hand-held device being given to major customers, or it might have to use the existing servers and desktop computers, or any other hardware, software, or business practice.

Project drivers are the business-related forces. For example, the purpose of the project 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. Our reason for including them as part of the requirements is to present a coherent picture of all factors that contribute to the success or failure of the project and to illustrate how managers can use requirements as input when managing a project.

Testing Requirements

The Volere philosophy is to start testing requirements as soon as you start writing them. You make a requirement testable by adding its *fit criterion*. This fit criterion measures the requirement, making it possible to determine whether a given solution fits the requirement. If a fit criterion cannot be found for a requirement, then the requirement is either ambiguous or poorly understood. All requirements can be measured, and all should carry a fit criterion.

Atomic Requirements Shell

The requirements shell is a guide to writing each atomic requirement. The components of the shell (also called a "snow card") are identified below. You might decide to add some additional attributes to provide traceability necessary for your environment. For example: products that implement this requirement, version of the software that implements this requirement, departments who are interested in this requirement, etc. There are others but do not capriciously add attributes unless they really help you: every attribute you add needs to be maintained.

This requirements shell can, and should, be automated. When you download the template you will also find an Excel spreadsheet implementation of the snow card.

Id of events / use cases that need this requirement

The type from the template

Requirement #: **Unique id**   Requirement Type: Event/BUC/PUC #:

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

Rationale: **A justification of the requirement**

Originator: **The stakeholder 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: Customer Dissatisfaction:

Priority: **The relative importance of the requirement**   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

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

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

The following discusses and provides examples for each of the sections of the Volere Requirements Specification Template. For each section, the Content, Motivation, Considerations, Examples and Form provide the template user with some guidance for writing each type of requirement. When you download the template you will also find a Template Skeleton that you might find convenient to use as the basis for producing a document.

1. The Purpose of the Project

The first section of the template deals with the fundamental reason your client asked you to build a new product. That is, it describes the business problem the client faces and explains how the product is intended to solve the problem.

1a. The User Business or Background of the Project Effort

Content

A short description of the business being done, its context, and the situation that triggered the development effort. It should also describe the work that the user intends to do with the delivered product.

Motivation

Without this statement, the project lacks justification and direction.

Considerations

You should consider whether the business problem is serious, and whether and why it needs to be solved.

Perhaps there are no serious problems, just a significant business opportunity your client wishes to exploit. In this case, describe the opportunity

Alternatively, the project may seek to explore or investigate possibilities. In this case the project deliverable, instead of a new product, would be a document proving that the requirements for a product can (or cannot) be satisfied.

Form

A short text description is often sufficient to provide an understanding of the project. You can choose to support the description with some combination of a current situation model, business process models, samples of current documents, photographs and videos of the current situation, website addresses and organization charts.

1b. Goals of the Project

Content

This part of the specification describes what we want the product to do and what advantage it will bring to the overall goals of the work. Do not be too wordy in this section—a brief explanation of the project's goals is usually more valuable than a long, rambling treatise. A short, sharp goal will be clearer to the stakeholders and improve the chances of reaching a consensus for the goal.

Motivation

There is a danger that this purpose may get lost along the way. As the development effort heats up, and as the customer and developers discover more about what is possible, the system could potentially wander away from the original goals as it undergoes construction. 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 them. *It should be mandatory to acknowledge the goals at every review session.*

Examples

> We want to give immediate and complete response to customers who order our goods online.

> To reduce road accidents by accurately forecasting and scheduling the de-icing of roads.

Measurement

Any reasonable goal must be measurable. This is necessary if you are ever to test whether you have succeeded with the project. The measurement must quantify the *advantage* gained by the business through doing the project. If the project is worthwhile, there must be some solid business reason for doing it. For example, if the goal of the project is

> We want to give immediate and complete response to customers who order our goods online.

you have to ask what advantage meeting that goal brings to the organization. If immediate response will result in more satisfied customers, then the measurement must quantify that satisfaction. For example, you could measure the increase in repeat business (on the basis that a happy customer comes back for more), the increase in customer approval ratings from surveys, the increase in revenue from returning customers, and so on.

Ask whether your goal is a:

- Service goal: This is measured by quantifying what it does for the customer

- Revenue goal: quantify how much revenue or revenue growth over what period of time. Alternatively, a revenue goal could be quantified by market share.

- Legal goal: this is not a quantification, but a way of knowing that the product conforms to a piece of legislation (this could be the law of the land or might be a standard of your industry or organization).

It is crucial to the rest of the development effort that the goal is firmly established, is reasonable, and is measured. It is usually the latter that makes the former possible.

Form

You can use Purpose, Advantage, Measurement (PAM) to structure your goal.

Purpose: one sentence to explain the organisation's reason for investing in the project.

Advantage: One sentence describing the benefit that the organization will realize if the project is successful.

Measurement: One sentence or a graph or diagram that quantifies how you will measure whether or not the benefit has been achieved.

Another form for your goals might be to use some kind of goal model. For example, the Extended Enterprise Modelling Language EEML, includes a goal modelling technique. If your organization is using enterprise modelling then this provides a connection between the enterprise's strategic goals and the goal of an individual project.

2. The Stakeholders

This section describes the stakeholders—the people who have an interest in the product. It is worth your while to spend enough time to accurately determine and describe these people, as the penalty for not knowing who they are can be very high.

2a. The Client

Content

This item gives the name of the client (sometimes referred to as the sponsor) It is permissible to have several names, but having more than three negates the point.

Motivation

The client has the final say on acceptance of the product, and thus must be satisfied with the product as delivered. You can think of the client as the person who makes the investment in the product. Where the product is being developed for in-house consumption, the same person often fills the roles of the client and the customer. 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.

Form

An annotated organization chart showing where the client fits within the organization.

A list of the decisions for which the client will be responsible.

You can also include a chart showing the review checkpoints and itemizing what you will provide for the client as progress indicators for the project.

2b. The Customer

Content

The person intended to buy the product. In the case of in-house development, the client and the customer are probably the same person. The customer might also be the manager who decides whether or not the people for whom he is responsible will adopt a new/changed product.

In the case of development of a mass-market product, this section contains a description of the persona developed as the archetypical customer for the product (See section 2e).

Motivation

The customer is ultimately responsible for deciding whether to buy or recommend the use of the product. The correct requirements can be gathered only if you understand the customer and his aspirations when it comes to using your product.

Form

A list of the decisions for which the customer will be responsible.

You can also include a chart showing the review checkpoints and itemizing what you will provide for the customer as progress indicators for the project.

This might include a list of possible prototypes or simulations that you will provide for the customer during the progress of the project.

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 to build the product. These stakeholders might work for your organization but might be external.

Examples of stakeholders:

- Client/Sponsor (refer to 2a)
- Customer (refer to 2b)
- Subject Matter Experts
- Members of the public
- Users of a current system
- Marketing experts
- Legal experts
- Domain experts
- Usability experts
- Representatives of external associations
- Business analysts
- Designers and developers
- Testers
- Systems engineers
- Software engineers
- Technology experts
- System designers

For a complete checklist, download the stakeholder analysis template at http://www.volere.co.uk.

For each type of stakeholder, provide the following information:

- Stakeholder identification (some combination of role/job title, person name, and organization name)
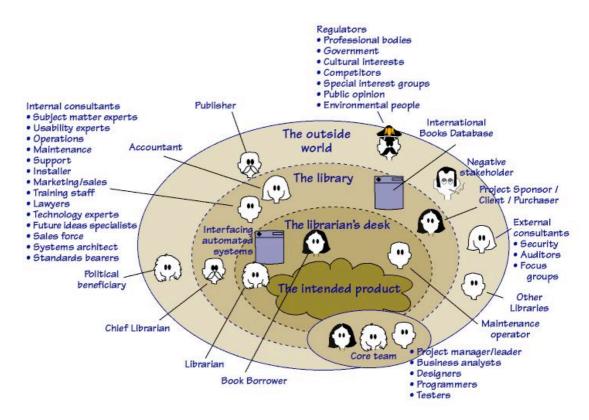- Knowledge that the project needs from that stakeholder

- The degree of involvement necessary for that stakeholder/knowledge combination

- The degree of influence for that stakeholder/knowledge combination

- Agreement on how to address conflicts between stakeholders who have an interest in the same knowledge

Motivation

Failure to recognize stakeholders results in missing requirements.

Form

A stakeholder map supported by the name of the representative of each role together with the knowledge to be supplied by that role. The following diagram is a generic stakeholder map that you can use as a checklist and replace the role names with the specific people/roles/organizations for your project. For more on stakeholder maps refer to http://www.volere.co.uk

Another form you can use to identify the stakeholders is a Stakeholder Analysis Spreadsheet. A sample is downloadable at http://www.volere.co.uk

An annotated organization chart is also a useful form for defining stakeholders.

2d. The Hands-On Users of the Product

Content

A list of a special type of stakeholder—the potential users of the product. For each category of user, provide the following information:

- User name/category: Most likely the name of a user group, such as clerical users, schoolchildren, road engineers, or project managers.

- User role: Summarizes the users' responsibilities.

- Subject matter experience: Summarizes the users' knowledge of the subject matter/business. Rate as novice, journeyman, or master.

- Technological experience: 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. For example:

    Physical abilities/disabilities

    Intellectual abilities/disabilities

    Attitude toward job

    Attitude toward technology

    Physical location

    Education

    Linguistic skills

    Age group

    Gender

    Ethnic group/s

Motivation

Users are human beings who interface with the product in some way. Use the characteristics of the users to define the usability requirements for the product. Users are also known as *actors*.

Examples

Users can come from wide variety of (sometimes unexpected) sources. Consider the possibility of your users being clerical staff, shop workers, managers, highly trained operators, the 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 an Internet connection, emergency workers, and so on.

Form

A simple list or a spreadsheet containing the User Characteristics for each User Role + User Name/Representative

2e. Personas

Content

A story about an invented person that includes:

Persona's name, age, job, family, hobbies, where they live, favourite food, favourite music, likes, dislikes, where they go on holiday, attitude to technology, attitude to money, or any other characteristic that could influence the way that the persona thinks of the product. It helps if you include a photograph or drawing of the imagined person.

Motivation

By having one or more (limit it to 3) personas you can make the requirements specific to the people you are trying to satisfy. This is a particularly effective technique if you are specifying the requirements for a consumer product or a product that will be used by members of the public.

Form

A profile containing the life story of the persona. Include a photograph or drawing of the person. The profile can be in the form of a document that you use to introduce project participants to the persona. You can also put the profile onto a large format, A3, card that you display at meetings to remind participants about whose requirements you are trying to discover. Another idea is to build a storyboard of the persona's life. Also you can make a website for the persona and keep him/her alive by adding more about the persona's everyday life. All of these forms of capturing and communicating the persona are intended to help people think of the persona as a real user with specific, rather than general, real requirements.

2f. Priorities Assigned to Users

Content

Attach a priority to each category of user. This identifies the importance and precedence of the user. Prioritize the users as follows:

- Key users: They 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, as well as people who misuse the product.

The percentage of the type of user is intended to assess the amount of consideration given to each category of user.

Motivation

If some users are considered to be more important to the product or to the organization, then this preference 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 group who has specifically asked for the product, and for which, if they do not get what they want, the results could be a significant loss of business.

Some users may be listed as having no impact on the product. These 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.

Form

Include the user importance rating on your user characteristics spreadsheet, see 2d, for the information of the core project team. Depending on the culture of your organization, you might need to treat this as sensitive information.

2g. User Participation

Content

Where appropriate, attach to the category of user a statement of the participation that you think will be necessary for those users to provide the requirements. Describe the contribution that you expect these users to provide—for example, business knowledge, interface prototyping, or usability requirements. If possible, assess the minimum amount of time that these users must spend for you to be able to determine the complete requirements.

Motivation

Many projects fail through lack of user participation, sometimes 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 usually takes priority. This requirement makes it clear, from the outset, that specified user resources must be allocated to the project.

Form

Include the estimated user participation time, together with the type of knowledge you expect that user to provide, on your user characteristics spreadsheet, see 2d.

2h. Maintenance Users and Service Technicians

Content

Maintenance users are a special type of hands-on users who have requirements that are specific to maintaining and changing the product.

Motivation

Many of these requirements will be discovered by considering the various types of maintenance requirements detailed in section 14. However, if we define the characteristics of the people who maintain the product, it will help to trigger requirements that might otherwise be missed.

Form

Include the maintenance users, on your user characteristics spreadsheet, see 2d.

3. Mandated Constraints

This section describes constraints on the eventual design of the product. Constraints are global—they are factors that apply to the entire product. The product must be built within the stated constraints. Often you know about the constraints, or they are mandated before the project gets under way. They are probably determined by management and are worth considering carefully—they restrict what you can do and so shape the product. Constraints, like other types of requirements have a description, rationale, and fit criterion, and generally are written in the same format as functional and non-functional requirements.

3a. Solution Constraints

Content

This specifies constraints on the way that the problem must be solved. Describe the mandated technology or solution. Include any appropriate version numbers. You should also explain the reason for using the technology.

Motivation

To identify constraints that guide the final product. Your client, customer, or user may have design preferences, or only certain solutions may be acceptable. If these constraints are not met, your solution is not acceptable.

Examples

Constraints are written using the same form as other atomic requirements (refer to the requirements snow card/shell for the attributes). It is important for each constraint to have a rationale and a fit criterion, as they help to expose false constraints (solutions masquerading as constraints). Also, you will usually find that a constraint affects the entire product rather than one or more product use cases.

Description: The product shall use the current two-way radio system to communicate with the drivers in their trucks.

Rationale: The client will not pay for a new radio system, nor are any other means of communication available to the drivers.

Fit criterion: All signals generated by the product shall be audible and understandable by all drivers via their two-way radio system.

Description: The product shall operate using Windows XP.

Rationale: The client uses XP and does not wish to change to a later version.

Fit criterion: The product shall be approved as XP compliant by the MS testing group.

Description: The product shall be a hand-held device.

Rationale: The product is to be marketed to hikers and mountain climbers.

Fit criterion: The product shall weigh no more than 300 grams, no dimension shall be more than 15 centimetres, and there shall be no external power source.

Considerations

We want to define the boundaries within which we can solve the problem. Be careful, because anyone who has experience with or 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, making it very easy for false constraints to creep into a specification. 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.

Form

Include the constraint requirements as a specific type of atomic requirement in your requirements spreadsheet or database. For attributes of an atomic requirement see the Atomic Requirements Shell example at the start of this template. Also refer to article on atomic requirements at
http://www.volere.co.uk

Another form for constraints can be diagram/s of the systems architecture for the new/changed product – see 3b & 3c.

3b. Implementation Environment of the Current System

Content

This describes the technological and physical environment in which the product is to be installed. It includes automated, mechanical, organizational, and other devices, along with the nonhuman 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

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

Considerations

All 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, then include an organization chart.

Form

A diagram that represents each hardware and software component/subcomponent/device/building block that will be used to implement the product. The particular diagrams that you use depend on your organization and your projects' ways of working, the important issue is that the implementation environment is unambiguously understandable by the people who have to make decisions about how the functional and non-functional requirements will be implemented. Types of UML diagrams commonly used are: Class, Component, Component Structure, Deployment, Package diagrams. There are many other home-grown diagrams.

3c. Partner or Collaborative Applications

Content

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

Motivation

To provide information about design constraints caused by using partner applications. By describing or modelling 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 have an effect on the product.

Considerations

Examine the work context model to determine whether 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.

Form

A diagram or table that identifies all the interfaces between the product to be built and other adjacent systems. Bear in mind that the adjacent systems might be software, human or hardware. Some adjacent systems are within your organization and hence potentially more easily understood and perhaps influenced. Other adjacent systems are outside your organization and might be

difficult if not impossible to influence. A product scope diagram (see 8a for an example) is often used to define interfaces with partner or collaborative applications.

3d. Off-the-Shelf Software

Content

This describes commercial, open source, or any other off-the-shelf software (OTS) that must be used to implement some of the requirements for the product. It could also apply to non-software OTS components such as hardware or any other commercial product that is intended as part of the solution.

Motivation

To identify and describe existing commercial, free, open source, or other products to be incorporated into the eventual product. The characteristics, behaviour, and interfaces of the package are design constraints.

Considerations

When gathering requirements, you may discover requirements that conflict with the behaviour and characteristics of the OTS software. Keep in mind that the use of OTS software was mandated before the full extent of the requirements became known. In light of your discoveries, you must consider whether the OTS product is a viable choice. If the use of the OTS software is not negotiable, then the conflicting requirements must be discarded.

Note that your strategy for discovering requirements is affected by the decision to use OTS software. In this situation you investigate the work context in parallel with making comparisons with the capabilities of the OTS product. Depending on the comprehensibility of the OTS software, you might be able to discover the matches or mismatches without having to write each of the business requirements in atomic detail. The mismatches are the requirements that you will need to specify so that you can decide whether to satisfy them by either modifying the OTS software or satisfying the requirement in another way or modifying the business requirements.

Given the spate of lawsuits in the software arena, you should consider whether any legal implications might arise from your use of OTS. You can cover this in section 17. Legal Requirements.

Form

Models or written documentation that specifies the functional and non-functional requirements that can be implemented using this OTS software product. If the OTS product has a well structured requirements specification

and systems architecture model then that provides you with the basis for identifying which of your requirements can be satisfied by the product. If the product's documentation is not traceable and well organized then you will need to do more detailed work on your own requirements until you find a level at which you can map your requirements to the OTS product.

Another form is a person or people who are experts in the OTS product and can answer your questions without you having to puzzle through cryptic or marketing-oriented documents.

3e. Anticipated Workplace Environment

Content

This describes the workplace in which the users are to work and use the product. It should describe any features of the workplace that could have an effect on the design of the product, and the social and cultural aspects of the workplace.

Motivation

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

Examples

The single office printer is a considerable distance from the user's desk. This constraint suggests that printed output should be deemphasized.

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

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

The product is to be used in a library; it must be extra quiet.

The product is a photocopier to be used by an environmentally conscious organization; it must work with recycled paper.

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.

Form

Written description of the workplace; rich pictures showing all the components in the workplace; photographs of the workplace; videos of the workplace.

3f. Schedule Constraints

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 by giving 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:

- What happens if we don't build the product by the end of the calendar year?
- What is the financial impact of not having the product by the beginning of the Christmas buying season?
- What parts of the product are most critical for the Christmas buying season?

Form

A written statement giving: The date of the deadline, The reason for the deadline, The effect of not meeting the deadline

3g. Budget Constraints

Content

This section shows the budget for the project, expressed in money or available resources.

Motivation

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

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

Considerations

The intention is to restrict the wildest ambitions and to prevent the team from gathering requirements for an Airbus 380 when the budget can buy only a Cessna. 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 the client does not place enough value on the product. In either case you should consider whether it is worthwhile continuing.

Form

A written statement giving the amount of the budget and the source of the funding.

3h. Enterprise Constraints

Content

This section contains requirements that are specific to the enterprise that is making the investment in your project.

Motivation

To understand requirements that sometimes appear irrelevant or irrational because they are not obviously relevant to the goals of the project.

Examples

The product shall be installed using only American-made components.

The product shall make all functionality available to the CEO.

Considerations

Did you intend to develop the product on a Macintosh, when the office manager has laid down an 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 enterprise requirements has little bearing on the outcome. The reality is that the system has to comply with enterprise requirements even if you can find a better, more efficient, or more economical solution. A few probing questions here may save some heartache later.

The enterprise requirements might be purely concerned with the politics inside your organization. However, in other situations you may need to consider the politics inside your customers' organizations or the national politics of the country. Another way to think about the enterprise requirements is that they are constraint requirements that have been defined by strategic decisions that are outside the obvious boundary of your project scope.

4. Naming Conventions and Terminology

It has been our experience that all projects have their own unique vocabulary usually containing a variety of acronyms and abbreviations. Failure to understand this project-specific nomenclature correctly inevitably leads to misunderstandings, hours of lost time, miscommunication between team members, and ultimately poor-quality specifications.

4a. Definitions of All Terms, Including Acronyms, Used by Stakeholders Involved in the Project

Content

A glossary containing the meanings of all names, acronyms, and abbreviations used by the stakeholders. Select names carefully to avoid giving a different, unintended meaning.

If the work that you are studying already has a glossary of terms then use this as your starting point. This glossary should be enlarged and refined as the analysis proceeds, but for the moment, it should introduce the terms that the stakeholders use and the meanings of those terms.This glossary reflects the terminology in current use within the work area. You might also get started by building on the standard names used within your industry.

For each term, write a description. The appropriate stakeholders must agree on this description of the meaning of the term.

We suggest you add *all* acronyms and abbreviations. We often encounter situations where team members use acronyms, but admit they do not know the

meanings of those acronyms. This section gives you a place to register your acronyms.

Motivation

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

As the detailed work progresses the glossary provides input to the more precisely specified business/work data model and data dictionary – see section 7 of the template. As the analysis data dictionary evolves, many of the definitions from the glossary are expanded in the dictionary by adding their data composition.

Examples

Truck: A vehicle used for spreading de-icing material on roads. "Truck" is not used to refer to goods-carrying vehicles.

BIS: Business Intelligence Service. The department run by Steven Peters to supply business intelligence for the rest of the organization.

Thermal Map: A region or other geographical area is surveyed to determine the temperature differences at various parts of the area. The resulting thermal map means the temperature at any part of the area can be determined by knowing the temperature at a reference point.

Considerations

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

From the beginning of the project, emphasize the need to avoid homonyms and synonyms. Explain how they increase the cost of the project.

Form

An existing glossary of terms, or a pointer to industry dictionaries, or a list of terms commonly used in the problem domain along with a sentence describing the meaning and purpose of each term.

5. Relevant Facts and Assumptions

Relevant facts are external factors that have an effect on the product but are not covered by other sections in the requirements template. They are not necessarily translated into requirements but could be. Relevant facts alert the developers to conditions and factors that have a bearing on the requirements.

5a. Relevant Facts

Content

Factors that have an effect on the product, but are not mandated requirements constraints. Facts provide the reader of the specification with more background for understanding the business problem.

Motivation

Relevant facts provide background information to the specification readers, and 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 three miles of single-lane roadway.

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

5b. Business Rules

Content

These are business rules that might have an impact on the work/business/domain that is the source of the requirements. Relevant business rules will be the trigger for requirements.

Motivation

Business rules are mentioned at all stages of the requirements discovery process. It is often difficult to immediately ascertain whether a business rule is or is not relevant to the project that you are doing. This section provides a place to capture the business rules and, as understanding of the work increases, to revisit them and use them as triggers to discover relevant requirements.

Examples

The maximum length of a truck driver's shift is 5 hours.

The engineers maintain the weather stations once a week.

Form

A written statement describing the business rule, the reason for the rule, the authority for the rule.

At the start of a new project have a look to see if there are some relevant business rules that have already been defined. This sets you on the road to requirements reuse. When your project discovers a new or changed business

rule then add this to the *Business Rule Book* for your enterprise. The business rule book then becomes input to and is updated by every project.

You might include business process models (there are many different forms of these) to illustrate how the business rule affects the organization.

5c. Assumptions

Content

A list of the assumptions that the developers are making. These assumptions might be about the intended operational environment, but can be about anything that has an effect on the product. As part of managing expectations, assumptions also contain statements about what the product will ***not*** do.

Motivation

To make people declare the assumptions that they are making. Also, to make everyone on the project aware of assumptions that have already 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, or software components.

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.

The availability and capability of bought-in components.

Dependencies on computer systems or people external to this project

The requirements that will specifically *not* be carried out by the product.

Some specific examples of assumptions from the IceBreaker project are:

Roads that have been treated will not need treating for at least two hours.

Road treatment stops at county boundaries.

 Road Engineering's Apian system will be available for integration testing before November.

The treatment trucks being built will be capable of operating at up to 40 mph. They will have a material capacity of two tons.

The Bureau's forecasts will be transmitted according to its specification 1003-7 issued by its engineering department.

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 such as these:

- 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 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—the assumption should have become either a requirement or a constraint. For example, if the assumption related to the capability of a product that is intended to be a partner product to yours, then the capability should have been proven satisfactory, and it becomes a constraint to use it. Conversely, if the bought-in product is not suitable, then it becomes a requirement for the project team to construct the needed capability.

Form

A written statement describing the assumption along with the effect on the project if the assumption is false. Depending on the complexity of the assumption, it might be necessary to include references to other documents or people

Understanding of assumptions can be explored and shared by using cause and effect diagrams such as Peter Senge's dynamics models.

6. The Scope of the Work

The scope of the work determines the boundaries of the business area to be studied and outlines how it fits into its environment. Once you understand the work and its constraints, you can establish the scope of the product see Section 8 of the template.

6a. The Current Situation

Content

This is an analysis of the existing business processes, including the manual and automated processes that might be replaced or changed by the new product. In terms of the Volere Brown Cow model you refer to this view as the "How Now" view. Business analysts might already have done this investigation as part of the business case analysis for the project. This is where it might be appropriate to build some Business process models. These are models of the processes that the business uses to carry out the work of the organization. The models include roles, individuals, departments, technology and procedures. They illustrate the workflow and the dependencies between the components of the process.

Motivation

If your project intends to make changes to an existing manual or automated system, you need to understand the effect of proposed changes. The study of the current situation provides the basis for understanding the effects of proposed changes and choosing the best alternatives. Business process modelling does not always lead to building software. Instead, some changes in procedures and the way roles are allocated might be the best way of making a necessary improvement.

Form

There are many different notations suitable for building business process models, for example: activity diagrams, business process diagrams, swimlane diagrams, dataflow diagrams.
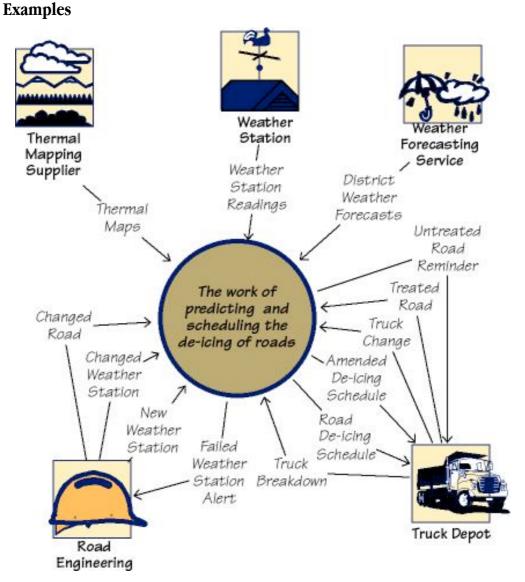
6b. The Context of the Work

Content

The work context diagram identifies the boundaries of the work that you need to investigate to be able to build the product. Note that it includes more than the intended product. Unless you understand the work that the product will support, you have 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 Service) 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 Service, we can say that we

are interested in the details of when, how, where, who, what, and why it produces the District Weather Forecasts information.

Motivation

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

Examples

This work context model defines the connections between the part of the world that is under investigation and other people, organizations, hardware and software (referred to as adjacent systems). The inputs and outputs represent the data and material that travels between the work and other parts

of the world. The work context is the basis for partitioning the investigation and discovering the requirements.

Considerations

The names used on the context diagram should be consistent with the naming conventions (section 4) and should eventually be defined in the data dictionary (section 7). Without these definitions, the context model lacks the required rigor, and it may be misunderstood. Relevant stakeholders must agree to the definitions of the interfaces shown on the context model.

Form

A diagram showing the inputs and outputs that flow between the work and the adjacent systems.

or

A table that identifies all the inputs and outputs that flow between the work and the adjacent systems

The names of the inputs and outputs are eventually defined in the data dictionary – see section 7b.

6c. Work Partitioning

Content

A list showing all business events to which the work responds. Business events are happenings in the real world that affect the work. They also happen because it is time for the work to do something—for example, produce weekly reports, remind non-paying customers, check the status of a device, and so on. The response to each event is called a business use case (known as a BUC); it represents a discrete partition of work that contributes to the total functionality of the work.

The event list includes the following elements:

- Event name
- Input or triggering data flow from adjacent systems (identical with name on context diagram)
- Output/s to adjacent systems (identical with name/s on context diagram)
- Brief summary of the business use case (This is optional, but we have found it is a very useful first step in defining the requirements for the business use case—you can think of it as a mini-scenario.)

- Classes of business data relevant to this event (you won't know this early in the study of the event, as you go into detail you will start to understand the essential data and you can add it to the event list.)

Motivation

To identify logical chunks of the work 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. Each business event has a business use case (BUC) whose details can be studied independently. However all BUCs connect to each other through the stored business data (see section 7).

Example

Business Event List

Event Name	Input and Output	Summary of BUC
1. Weather Station transmits reading	Weather Station Readings (in)	Record the readings as belonging to the weather station.
2. Weather Service forecasts weather	District Weather Forecast (in)	Record the forecast.
3. Road engineers advise changed roads	Changed Road (in)	Record the new or changed road. Check that all appropriate weather stations are attached.
4. Road Engineering installs new Weather Station	New Weather Station (in)	Record the weather station and attach it to the appropriate roads.
5. Road Engineering changes Weather Station	Changed Weather Station (in)	Record the changes to the weather station.
6. Time to test Weather Stations	Failed Weather Station Alert (out)	Determine if any weather stations have not transmitted for two hours, and inform Road Engineering of any failures.
7. Truck Depot changes a truck	Truck Change (in)	Record the changes to the truck.
8. Time to detect icy roads	Road De-icing Schedule (out)	Predict the ice situation for the next two hours. Assign a truck to any roads that will freeze. Issue the schedule.
9. Truck treats a road	Treated Road (in)	Record the road as being in a safe condition for the next three hours.
10 Truck Depot reports problem with truck	Truck Breakdown (in) Amended Gritting Schedule (out)	Reassign available trucks to the previously assigned roads.
11. Time to monitor road treatment	Untreated Road Reminder (out)	Check that all scheduled roads have been treated in the assigned time, and issue reminders for any untreated roads.

Considerations

Attempting to list the business events and do a one-sentence summary of each of the BUCs is a way of testing the work context. This activity uncovers uncertainties and misunderstandings about the project and facilitates precise

communications. When you do an event analysis, it will usually prompt you to make some changes to your work context diagram.

We suggest you gather requirements for discrete sections of the work. This requires you to partition the work, and we have found business events to be the most convenient, consistent, and natural way to break the work into manageable units and to be able to trace the details back to the scope of the work.

Form

Business event list/table containing for each event: Event number, Event name, Name of input, Name of output/s, Summary of the business event response. The names on the business event list must match the names on the work context model/table ref. 6.b

6d. Specifying a Business Use Case (BUC)

Content

A specification of the details of how a Business Use Case (BUC) responds to a Business Event.

Motivation

To understand the detailed business response that must be carried out when a business event takes place and provide a basis for discovering the detailed requirements. The understanding of the BUC also provides the basis for discussing which parts of the BUC should be carried out by the product that will be built.

Example

In the sample specifications included with the download of this template you will find examples of BUC scenarios.

Considerations

Whatever approach you use to specify the details of a BUC, you should stay within the boundary of the input and output/s for that business event. If you discover additional input or output data then it is an indication that you need to make changes to the input/output data on the event list and also on the work context diagram.

Form

A BUC can be specified using any combination of models that suits the analyst. The most common approaches are: activity diagrams, BUC scenarios, process flow diagrams, sequence diagrams, mind maps, interview notes.... The only caveat is that the inputs and outputs on your BUC are precisely the same and

hence traceable to the inputs and outputs on the corresponding Business Event.

7. Business Data Model and Data Dictionary
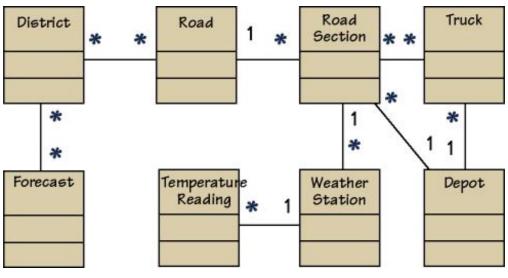
7a. Business Data Model

Content

A specification of the essential subject matter, business objects, entities, and classes that are germane to the product. It might take the form of a first-cut class model, an entity-relationship model, or any other kind of data model.

Motivation

To clarify the system's subject matter, thereby triggering recognition of requirements not yet considered. To discover missing requirements you can cross check the data model and the events using a Create, Reference, Update, Delete (CRUD) table. The data model is a specification for all of the business data that is relevant to the scope of the work.

Example

This is a model of the business system's business subject matter using the Unified Modelling Language (UML) class model notation. This is all the data that is Created, Referenced, Updated and Deleted by processes within the scope of the work being studied. See section 6 for more about the scope of the work.

Each of the rectangles represents a class of business data. The attributes of that class are defined in the data dictionary.

e.g.: District = *A geographical area defined by the council*

District Name + District Size + District Co-ordinates

Similarly, each attribute is also defined in the dictionary

e.g. District Name = *The name used by the engineers to identify a district*

You can use any type of data or class 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 to show that you are consistent within your project. If you have an established company standard notation, use that, as it will help you to reuse knowledge between projects.

For more examples of data models look at the example specifications that are packaged with the download of this requirements template.

Considerations

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

Form

There are many different types of data models that you can use to model the business data. The ones you are most likely to come across are:

- UML class model

- Crow's foot diagrams

- Entity Relationship diagrams

- A table showing: Class Name, Relationships between classes, Attributes for each class.

If your organization prefers a particular model then you must use that one. The important thing is that the data model that you build is a *business data model,* not a design for a database. Your model is concerned with identifying business classes by making a logical partitioning of all the data within the work context and the necessary business relationships between those business classes. Your model is used as input to designing how the data will be implemented. The definitions of the attributes in each business class is in the data dictionary (see section 7b).

7b. Data Dictionary

The glossary described earlier in section 4 of the template is the starting point for establishing common understanding of terminology. As you start to define the scope of the investigation you define the data inputs and outputs in a formal data dictionary. The terms that you define in this dictionary, right down to elemental level, are the same terms that you use when defining detailed atomic requirements.

Content

The data dictionary specifies the content of:

- Classes on the data model
- Attributes of the classes
- Relationships between the classes
- Inputs and Outputs on all models
- Elements of data within the Inputs and Outputs

When implementation decisions are made the technical specifications for the interfaces should be added to the dictionary.

Motivation

The work context diagram provides an accurate definition of the scope of the work being studied and the product scope diagram (See Sections 8a & 8b) defines the boundary of the product to be built. These definition can be completely accurate only if the information flows bordering the scope have their attributes defined.

Examples

The following is a partial data dictionary for the road de-icing project we have been using as an example in this template. Note that this version of the dictionary is sorted alphabetically within Type.

When implementation decisions are eventually made the format of the data is added to the dictionary by the designers/implementors.

Name	Content	Type
Depot	Depot Identifier	Class
District	District Name + District Size	Class
Forecast	Forecast Temperature + Forecast Time	Class
Road	Road Name + Road Number	Class

Road Section	Road Section Identifier + Road Section Coordinates	Class
Temperature Reading	Reading Time + Temperature Measurement	Class
Truck	Truck Identifier	Class
Road De-Icing Schedule	{Road Section Identifier + Treatment Scheduled Date + Treatment Scheduled Start Time + Critical Start Time + Truck Identifier}	Dataflow
District Name	*Listed in District Catalogue*	Attribute/Element
District Size	*Measured in Kilometres*	Attribute/Element
Forecast Temperature	*Measured in Celsius*	Attribute/Element
Forecast Time	*HH/MM/SS 24 hour clock*	Attribute/Element
Reading Time	*HH/MM/SS 24 hour clock*	Attribute/Element
Road Name	*See Road Database*	Attribute/Element
Road Number	*See Road Database*	Attribute/Element
Road Section Coordinates	*See Road Database*	Attribute/Element
Road Section Identifier	*See Road Database*	Attribute/Element
Temperature Measurement	*Measured in Celsius*	Attribute/Element
Treatment Scheduled Date	*YY/MM/DD*	Attribute/Element
Treatment Scheduled Start Time	*HH/MM/SS 24 hour clock*	Attribute/Element
Truck Identifier	*Number between 1 and 1000. Refer to Chief Engineer for list*	Attribute/Element

Critical Start Time	*Treatment started after this time is not guaranteed to be effective*	Attribute/Element
Treatment Scheduled Date	*YY/MM/DD*	Attribute/Element
Treatment Scheduled Start Time	*HH/MM/SS 24 hour clock*	Attribute/Element

Considerations

The dictionary provides a link between the requirements/business analysts and the designers/developers/implementers. The implementers add implementation details to the terms in the dictionary, defining how the data will be implemented. Also, implementers add terms that are present because of the chosen technology and that are independent of the business requirements.

As you study the work you often discover that an entry that you have put in the Naming Conventions and Terminology (section 4) turns out to be a specific flow of data or an attribute data. When this happens you should transfer the entry to the data dictionary.

Form

The data dictionary is maintained in a variety of forms, depending on the tools that you have at your disposal. The important issue is that you make it as easy as possible to cross reference the use of terms in requirements, documents and models with their definitions in the dictionary. Common forms for maintaining the data dictionary are spreadsheets, databases and automated requirements tools.

8. The Scope of the Product

8a. Product Boundary

A use case diagram identifies the boundaries between the users (actors) and the product. You arrive at the product boundary by inspecting each business use case and determining, in conjunction with the appropriate stakeholders, which part of the business use case should be automated (or satisfied by some sort of product) and what part should be done by the user or some other product. This task must take into account the abilities of the users/actors

(section 2), the constraints (section 3), the goals of the project (section 1), and your knowledge of both the work and the technology that can make the best contribution to the work.

The example use case diagram shows the users/actors outside the product boundary (the rectangle). The product use cases (PUCs) are the ellipses inside the boundary. The numbers link each PUC back to the BUC that it came from (see section 7). The arrows denote usage. In this version of a PUC diagram we put names on the arrows to make it more precise and traceable. Note that actors can be either automated or human.

You derive the PUCs by deciding where the product boundary should be for each business use case (BUC). These decisions are based on your and appropriate stakeholders' knowledge of the work and the requirements constraints. Note that the PUCs that you come up with must be traceable back to the BUCs. When you implement a PUC your internal design might, and probably will, mean that you will implement a PUC with several system use cases (SUCs).
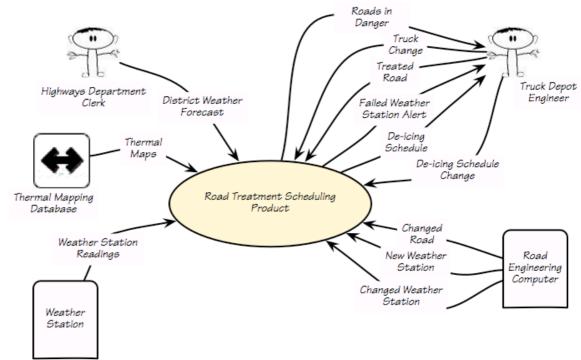
Example

The numbers on the PUC diagram above correspond to the BUC numbers on the Business Event List (see section 7).

You can see that the PUC diagram is an effective summary for a small number (up to around 20) PUCs. If you have a larger number of PUCs then a Product Scope Diagram together with a PUC Summary Table (see 8b) is a better approach.

Below is an example of a Product Scope Diagram for the Road De-icing project.

The content of each one of the interfaces on the product scope diagram is defined in the data dictionary and can also be supported by a prototype or simulation of some kind. Some projects publish a separate document called the Interface Specification Document. The content of that document would be the definitions of the interfaces on the Product Scope Diagram often supported by prototypes and models.

The above Product Scope diagram summarises the interfaces between the product and the actors/users.

Form

A product use case diagram.

A product scope diagram supported by individual use case specification.

Specifications and prototypes of the interfaces on the scope diagram.

8b. Product Use Case Table

The product scope diagram is a useful summary of all the interfaces between the product and other automated systems, organizations and users. If there are

a manageable number of PUC's – say less than twenty – then the PUC diagram is useful as a graphical way of summarizing the PUC's relevant to the product, But in practice we have found that a Product Use Case Table is more useful because it can handle larger numbers of PUCs and it precisely identifies the input and output data that defines the boundary of each PUC.

Product Use Case (PUC) Summary Table

PUC No	PUC Name	Actor/s	Input & Output
1	Record Weather Station Readings	Weather Station	Weather Station Readings (in)
2	Record Weather Forecast	Highways Department Clerk	District Weather Forecast (in)
3	Record Roads	Road Engineering Computer	Changed Road (in)
4	Record New Weather Station	Road Engineering Computer	New Weather Station (in)
5	Record Changed Weather Station	Road Engineering Computer	Changed Weather Station (in)
6	Identify Faulty Weather Stations	Truck Depot Engineer	Failed Weather Station Alert (out)
7	Record Truck Changes	Truck Depot Engineer	Truck Change (in)
8	Produce De-icing Schedule	Truck Depot Engineer / Thermal Mapping Database	De-icing Schedule (out) / Thermal Maps (in)
9	Record Treated Roads	Truck Depot Engineer	Treated Road (in)
10	Amend De-icing Schedule	Truck Depot Engineer	De-icing Schedule Change (in) / Amended De-icing Schedule (out)
11	Monitor Untreated Roads	Truck Depot	Roads in Danger

		Engineer	(out)

8c. Individual Product Use Cases

This is where you define the details about the individual product use cases PUCs listed on your PUC table. You can include a scenario or model, for each product use case on your list.

Form
- A text scenario
- A storyboard
- A low fi prototype
- A hi fi prototype
- A formal use case specification including exceptions and alternatives
- A sequence diagram, activity diagram, dataflow diagram, or any other type of model that is familiar to your project group

9. Functional Requirements

9a. Functional Requirements

Content

This is a specification for each atomic functional requirement. As for all types of atomic requirements(functional, non-functional, constraint), use the requirements shell as a guide for which attributes should be specified. A full explanation of the atomic requirement and its attributes is included in this template's introductory material.

Motivation

To specify the detailed functional requirements to be carried out by the product.

Examples

```
  Requirement #: 75          Requirement Type: 9        Event/BUC/PUC #: 7, 9

  Description: The product shall record all the roads that have been treated


  Rationale: To be able to schedule untreated roads and highlight potential danger

  Originator: Arnold Snow - Chief Engineer

  Fit Criterion: 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

  Customer Satisfaction: 3          Customer Dissatisfaction: 5

  Dependencies: All requirements using road and        Conflicts: 105
                scheduling data
  Supporting Materials:    Work context diagram, terms
                           definitions in section 5                    Volere
  History: Created February 29, 2010               Copyright © Atlantic Systems Guild
```

Fit Criterion

Each functional requirement should have a fit criterion or a test case. In any
event, the fit criterion is the benchmark so that the tester can objectively
determine whether or not the implemented product has met the requirement.

Considerations

If you have produced an event/use case list (see sections 6c and 8a/b), then you
can use it 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, partition these
requirements into event/use case–related groups later in the development
process.

Note that if you have not identified the product boundary and are not in a
position to determine the product use cases (PUC's), then write the functional
and non-functional requirements for the business use cases (BUC's). This is an
especially good strategy if you are writing business requirements and asking
suppliers to tell you which of your business requirements can be satisfied by
their product/s.

Form

The form that you use to capture and maintain your atomic requirements (functional, non-functional and constraint) depends on the tools that you have available to you. Volere snow cards are often a useful aid to help you in discovering requirements but, due to volume and need to be able to make changes, some kind of automated form is the best way to manage and maintain your atomic requirements. Common forms for atomic requirements are:

- A spreadsheet (a sample is included with the download of this template)
- A database provided with whatever requirements tool/s you have available. There is a wide variety of tools on the market, refer to http://www.volere.co.uk/tools for a list
- An intranet set up by you to maintain and make accessible the atomic requirements and their attributes
- A custom built database

Whatever form you use to record and maintain your requirements, it is important to be consistent with your numbering and terminology so that you can check for completeness and respond to change.

Non-functional Requirements

The following sections 10-17 describe the non-functional requirements. The form of these requirements is the same as for the functional requirements as described above.

10. Look and Feel Requirements

10a. Appearance Requirements

Content

The section contains requirements relating to the spirit of the product. Your client may have made particular demands for the product, such as corporate branding, colours to be used, and so on. This section captures the requirements for the appearance. Do not attempt to design it until the appearance requirements are known.

Motivation

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

Examples

The product shall be attractive to a teenage audience.

The product shall comply with corporate branding standards.

Fit Criterion

A sampling of representative teenagers shall, without prompting or enticement, start using the product within four minutes of their first encounter with it.

The office of branding shall certify that the product complies with the current standards.

Considerations

Even if you are using prototypes, it is important to understand the requirements for the appearance. The prototype is used to help elicit requirements; it should not be thought of as a substitute for the requirements.

10b. Style Requirements

Content

Requirements that specify the mood, style, or feeling of the product, which influences the way a potential customer will see the product. Also, the stakeholders' intentions for the amount of interaction the user is to have with the product.

In this section, you would also describe the appearance of the package if this is 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. Keep in mind the European laws on packaging, which require that the package not be significantly larger than the product it encloses.

The style requirements that you record here will guide the designers to create 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 the wrong style. Once the functional requirements are satisfied, it is often the appearance and style of products that determine whether they are successful. Your task in this section is to determine precisely how the product shall appear to its intended consumer.

Example

The product shall appear authoritative.

Fit Criterion

After their first encounter with the product, 70 percent of representative potential customers shall agree they feel they can trust the product.

Considerations

The look and feel requirements specify your client's vision of the product's appearance. The requirements may at first seem to be rather vague (e.g., "conservative and professional appearance"), but these will be quantified by their fit criteria. The fit criteria give you the opportunity to extract from your client precisely what is meant, and give the designer precise instructions on what he is to accomplish.

11. Usability and Humanity Requirements

This section is concerned with requirements that make the product usable and ergonomically acceptable to its hands-on users.

11a. Ease of Use Requirements

Content

This section describes your client's aspirations for how easy it is 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 properties such as these:

- Efficiency of use: How quickly or accurately the user can use the product.

- Ease of remembering: How much the casual user is 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 that face a lot of competition. Web sites are a good example.

- Feedback: How much feedback the user needs to feel confident that the product is actually accurately doing what the user expects. The necessary degree of feedback will be higher for some products (e.g., safety-critical products) than for others.

Motivation

To guide the product's designers toward building a product that meets 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, you must add a measurement against which it can be tested—that is, a fit criterion. Here are the fit criteria for the preceding examples:

Eighty percent of a test panel of 11-year-old children 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 1 percent.

An anonymous survey shall show that 75 percent of the intended users are regularly using the product after a three-week familiarization period.

Considerations

Refer to section 3, Users of the Product, 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 any special usability considerations must be built into the product.

You could also consider consulting a usability laboratory experienced in testing the usability of products that have a project situation (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 issues such as the following:

- Languages, spelling preferences, and language idioms
- Currencies, including the symbols and decimal conventions
- Personal configuration options

Motivation

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

Examples

The product shall retain the buyer's buying preferences.

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

Considerations

Consider the country and culture 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 give them the opportunity to participate more closely with your organization as well as enjoy their own personal user experience.

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

11c. Learning Requirements

Content

Requirements specifying how easy it should be to learn to use the product. This learning curve ranges from zero time for products intended for placement in the public domain (e.g., a parking meter or a web site) to a considerable amount of time for complex, highly technical products. (We know of one product where it was necessary for graduate engineers to spend 18 months in a training program 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 guides designers to understand how users will learn the product. For example, 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 five weeks of training before using the product.

Fit Criterion

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 to section 2d, Hands-On Users of the Product, to ensure that you have considered the ease of learning requirements from the perspective of all the different types of users.

11d. Understandability and Politeness Requirements

This section is concerned with discovering requirements related to concepts and metaphors that are familiar to the intended end users.

Content

This specifies the requirement for the product to be understood by its users. While "usability" refers to ease of use, efficiency, and similar characteristics, "understandability" determines whether the users instinctively know what the product will do for them and how it fits into their view of the world. You can think of understandability as the product being polite to its users and not expecting them to know or learn things that have nothing to do with their business problem. Another aspect of politeness is that the product should not expect the user to input any information to which the product already has access.

Motivation

To avoid forcing users to learn terms and concepts that are part of the product's internal construction and are not relevant to the users' world. To make the product more comprehensible and thus more likely to be adopted by its intended users.

Examples

The product shall use symbols and words that are naturally understandable by the user community.

The product shall hide the details of its construction from the user.

Considerations

Refer to section 2d, Hands-On Users of the Product, and consider the world from the point of view of each of the different types of users.

11e. Accessibility Requirements

Content

The requirements for how easy it should be for people with common disabilities to access the product. These disabilities might be related to physical disability or visual, hearing, cognitive, or other abilities.

Motivation

In many countries it is required that some products be made available to the disabled. In any event, it is 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

Some users have disabilities other than the commonly described ones. In addition, some partial disabilities are fairly common. A simple, and not very consequential, example is that approximately 20 percent of males are red-green colour-blind.

12. Performance Requirements

12a. Speed and Latency Requirements

Content

Specifies the amount of time available to complete specified tasks. These requirements often refer to response times. They can also refer to the product's ability to operate at a speed suitable for 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 (e.g., 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 (e.g., 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

Fit criteria are needed when the description of the requirement is not quantified. However, we find that most performance requirements are stated in quantified terms. The exception is the second requirement shown above, for which the suggested fit criterion is

> The product shall respond in less than 1 second for 90 percent of the interrogations. No response shall take longer than 2.5 seconds.

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. By contrast, an inventory control report that is run once every six months has very little need for a lightning-fast response time.

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 the perceived risk of damage to people, property, and environment. Different countries have different standards, so the fit criteria must specify precisely which standards the product must meet.

Motivation

To understand and highlight the damage that could potentially 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

> The product shall be certified to comply with the Health Department's standard E110-98. It 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 example requirements given here 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.

Also, be aware that different countries have different safety standards and laws relating to safety. If you plan to sell your product internationally, you must be aware of these laws. A colleague has suggested that for electrical products, if you follow the German standards, the largest number of countries will be supported.

 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 experts are the best source of the relevant safety-critical requirements for your type of product. They will almost certainly have copious information that you can use.

Consult your legal department. Members of this department 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 or Accuracy Requirements

Content

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

Motivation

To set the client's and users' expectations for the precision of the product.

Examples

All monetary amounts shall be accurate to two decimal places.

Accuracy of road temperature readings shall be within ±2°C.

Considerations

If you have done any detailed work on definitions, then some precision requirements might be adequately defined by definitions in the dictionary in section 7.

You might consider which units the product is intended to use. Readers will recall the spacecraft that crashed on Mars when coordinates were sent as metric data rather than imperial data.

The product might also need to keep accurate time, be synchronized with a time server, or work in UTC.

Also, be aware that some currencies have no decimal places, such as the Japanese yen.

12d. Reliability and Availability Requirements

Content

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

This section 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 the client's and users' 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 8:00 A.M. and 5:30 P.M.

The escalator shall run from 6 A.M. until 10 P.M. or until the last flight arrives.

The product shall achieve 99 percent uptime.

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 or Fault-Tolerance 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 after or during 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. Today's 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 the product.

You could also consider disaster recovery in this section. This plan describes the ability of the product to re-establish acceptable performance after faults or abnormal happenings.

12f. Capacity Requirements

Content

This section specifies the volumes that the product must be able to deal with and the amount 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 9:00 A.M. to 11:00 A.M. Maximum loading at other periods will be 150 simultaneous users.

During a launch period, the product shall cater for a maximum of 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 a business grows (or is expected to grow), our software products must increase their capacities to cope with the new volumes.

Motivation

To ensure that the designers allow capacity for future growth.

Examples

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

> The product shall be able to process 50,000 transactions per hour within two years of its launch.

12h. Longevity Requirements

Content

This specifies the expected lifetime of the product.

Motivation

To ensure that the product is built based on an understanding of expected return on investment.

Examples

> The product shall be expected to operate within the defined maximum maintenance budget for a minimum of five years.

13. Operational and Environmental 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.

> The product shall not be louder than the existing noise level in the environment.

Considerations

The work environment: Is the product to operate in some unusual environment? Does this lead to special requirements? Also see section 11, Usability and Humanity Requirements.

13b. Requirements for Interfacing with Adjacent Systems

Content

This section describes the requirements to interface with partner applications and/or devices that the product needs in order to successfully operate.

Motivation

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

Examples

> The products shall work on the last four releases of the five most popular browsers.
>
> The new version of the spreadsheet must be able to access data from the previous two versions.
>
> Our product must interface with the applications that run on the remote weather stations.

Fit Criterion

For each inter-application interface, specify the following elements:

- The data content
- The physical material content
- The medium that carries the interface
- The frequency
- The volume
- The trigger
- The standards/protocols that apply to the interface

13c. 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 needed to install a software product successfully.

Motivation

To ensure that if work must be done to get the product out the door, then that work becomes part of the requirements. Also, to quantify the client's and users' expectations about the amount of time, money, and resources they will need to allocate to install the product.

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 such that it can fit on 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.

Ask questions of your marketing department to discover unstated assumptions that have been made about the specified environment and the customers' expectations of how long installation will take and how much it will cost.

Most commercial products have needs in this area.

13d. Release Requirements

Content

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

Motivation

To make everyone aware of how often you intend to produce new releases of the product.

Examples

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

Each release shall not cause previous features to fail.

Fit Criterion

Description of the type of maintenance plus the amount of effort budgeted for it.

Considerations

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

14. Maintainability and Support Requirements

14a. Maintenance Requirements

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 when the requirements are agreed upon.

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

Considerations

There may be special requirements for maintainability, such as that the product must be able to be maintained by its end users or by developers who are not the original developers. These requirements have an effect on the way that the product is developed. In addition, there may be requirements for documentation or training.

You might also consider writing testability requirements in this section.

14b. Supportability Requirements

Content

This specifies the level of support that the product requires. Support is often provided via a help desk. If people will provide support for the product, that service is considered part of the product: Are there any requirements for that support? You might also build support into the product itself, in which case this section 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, a constraint might state that there is to be no printed manual. Alternatively, the product might need to be entirely self-supporting.

14c. Adaptability Requirements

Content

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

Motivation

To publicise the client's and users' expectations about the platforms on which the product will be able to run.

Examples

> The product is expected to run under Windows 7 and Linux.
>
> The product might eventually be sold in the Japanese market.
>
> The product is designed to run in offices, but we intend to have a version running 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

Question 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 is authorized to access to the product (both functionality and data), under what circumstances that access is granted, and to which 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 a current security clearance can enter the building.

Fit Criterion

System function name or system data name.

User roles and/or names of people who have clearance to access specified data.

User roles and/or names of people who have clearance to add, change, delete specified data.

Considerations

Is there any data that management considers to be sensitive? 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 stating how you would design a solution to the security requirements. For instance, don't specify a password system. Your aim here is to identify the security requirement; the design will come from the requirement.

Consider asking for help. Computer security is a highly specialized field, and one where improperly qualified people have no business. If your product has need of more than average security, we advise you to make use of a security consultant. Such consultants 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 product's data. To specify what the product will do to ensure its integrity in the case of an unwanted happening such as attack from the outside or unintentional misuse by an authorized user.

Examples

The product shall prevent incorrect data from being introduced.

The product shall protect itself from intentional abuse.

Considerations

Organizations are relying more and more on their stored data. If this data should be come corrupt or incorrect—or disappear—then it could be a fatal blow to the organization. For example, 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 ensure the privacy of individuals about whom it stores information. The product must also ensure that all laws related to privacy of an 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 users aware of its information practices before collecting 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 the relevant privacy laws and the organization's information policy.

Considerations

Privacy issues 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 you must issue to your customers before collecting their personal information. Also, do you have to do anything to keep customers aware that you hold their personal information?

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

Also consider the integrity and security of private data—for example, when you are storing 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 regarding 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 such 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, malware, spyware and any other undesirable interference.

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 Requirements

16a. Cultural Requirements

Content

This section contains requirements that are specific to the sociological 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.

Examples

The product shall not be offensive to religious or 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, or cultural norms that do not apply to your own culture? Are there colours, icons, measurement units or words that have different meanings in another cultural environment? If your reaction to a requirement is – that's rather odd/unusual/weird then likely you have a cultural requirement.

17. Legal Requirements

17a. Compliance Requirements

Content

A statement specifying the legal requirements for this system.

Motivation

To comply with the law so as to avoid later delays, lawsuits, 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 or other intellectual property that must be protected? Conversely, do any competitors have copyrights on which you might be in danger of infringing?

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

The Sarbanes-Oxley (SOX) Act, the Health Insurance Portability and Accountability Act (HIPAA) and the Gramm-Leach-Bliley Act may have implications for you. Check with your company lawyer.

Might any pending legislation affect the development of this system?

Are there any aspects of criminal law you should consider?

Have you considered the tax laws that affect your product?

Are there any labour laws (e.g., working hours) relevant to your product?

17b. Standards Requirements

Content

A statement specifying applicable standards and referencing detailed standards descriptions. This does not refer to the law of the land—think of it as an internal law imposed by your company or by your industry.

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 certifies 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:

- Do any industry bodies have applicable standards?

- Does the industry have a code of practice, watchdog, or ombudsman?

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

Project Issues

The following *sections 18-27* contain issues that must be faced if the requirements are to be met and the product to become a reality. These sections also connect the requirements with the project activities that discover and progress the requirements. If you are using a consistent language for communicating requirements then project managers can use the requirements as input to steering the project. The Volere Requirements Knowledge Model (included with the download of Version 16 of the template) provides the basis for a requirements common language by identifying classes of requirements knowledge and the relationships between them. Each of the classes of knowledge is cross-referenced to sections in this template.

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 the new version of the processor will be suitable for our application is not yet complete.

The government is planning to change the rules about who is responsible for gritting the motorways, but we do not know what those changes might be.

The feasibility study to determine whether to use the Regional Weather Center's online database is not yet complete. This issue affects how we should handle the weather data.

Planned changes to working hours for drivers may affect the way that trucks are scheduled and the length of the routes that drivers are permitted to travel. The changes are still in the proposal stage; details will be available by the end of the year

Considerations

When you are probing around the user's business, questions often come to the surface, and they cannot for the moment be answered. Similarly, as you are gathering the requirements for a future product, it may well be that your stakeholders are unsure of how the work should be done in the future. 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 or systems on your context diagram? Are there any legislative changes that might affect your system? Are there any rumours about your hardware or software suppliers that might have an impact?

Form

A list of open issues containing:
- Issue Number
- Cross reference to affected requirements (Business Events, BUCs, PUCs, Atomic Requirements, Dictionary Definitions)
- Summary of the issue
- Stakeholders involved
- Action
- Resolution

19. Off-the-Shelf Solutions

This section looks at available solutions and summarizes their applicability to the requirements. This discussion is not intended to be a full feasibility study of the alternatives, but it should tell your client that you have considered some alternatives and determined how closely they match the requirements for the product.

19a. Ready-Made Products

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 a solution can be bought.

Considerations

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

Also consider whether some products must not be used.

19b. Reusable Components

Content

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

Motivation

Reuse rather than reinvention.

19c. Products That Can Be Copied

Content

List of other similar products or parts of products that you can legally copy or easily modify.

Motivation

Reuse rather than reinvention.

Examples

Another electricity company has built a customer service system. Its hardware is different from ours, but we could buy its specification and cut our analysis effort by approximately 60 percent.

Considerations

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

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

Form

For each of 19a, 19b, and 19c, set our the alternatives that you think are suitable. If your findings are preliminary, then say so. It is useful to add approximate costs, availability, time to implement, and other factors that may have a bearing on the decision.

20. New Problems

20a. Effects on the Current Environment

Content

A description of how the new product 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 work of the truck drivers.

Considerations

Is it possible that the new system might damage some existing system? Can people be displaced or otherwise affected by the new system?

Form

These issues require 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. Effects on the Installed Systems

Content

Specification of the interfaces between new and existing systems.

Motivation

Very rarely is a new development intended to stand completely alone. Usually the new system must coexist with some older system. This question forces you to look carefully at the existing system, examining it for potential conflicts with the new development.

Form

A model identifying the interfaces between the new and existing systems supported by data dictionary definitions of the interfaces. The interfaces might also be supported by prototypes or sketches of format.

20c. Potential User Problems

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 or feature. Identify any likely adverse user reactions, and determine whether we care about those reactions and what precautions we will take.

20d. Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

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.

The size and weight of the new product do not fit into the physical environment.

The power capabilities will not satisfy the new product's projected consumption.

Considerations

This requires a study of the intended implementation environment.

20e. Follow-Up 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 run afoul 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

What steps have to be taken to deliver the product? This section highlights the effort required to build the product, the steps needed to buy a solution, the amount of effort to modify and install a ready-made solution, and so on.

21a. Project Planning

Content

Details of the life cycle and approach that will be used to deliver the product.

Motivation

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

Considerations

Depending on the maturity level of your process, the new product will be developed using your standard approach. However, some circumstances are unique to a particular product and will necessitate changes to your life cycle. While these considerations are not product requirements, they are needed if the product is to be successfully developed.

If possible, attach an estimate of the time and resources needed for each task based on the requirements that you have specified. Attach your estimates to the events, use cases, and/or functional requirements that you specified in sections 6, 8 and 9.

Do not forget issues related to data conversion, user training, and cutover. These needs are usually ignored when projects set implementation dates.

Form

A high-level process diagram or a task list showing the tasks and the interfaces between them is a good way to communicate this information. Here you can also identify the strategy that you intend to use to maximise your potential for agility.

21b. Planning of the 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.

Considerations

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

Form

This is usually a mixture of diagrams and text. For each phase of the project:

Name of the phase.

Value/benefit of delivery to user.

Required operational date.

Operating environment components included.

Functional requirements included.

Non-functional requirements included.

22. Migration to the New Product

When you install a new product, some things always have to be done before it can work successfully. For example, databases often have to be converted. There is usually new data to be collected, procedures to be converted, and many other steps to be taken to ensure the successful transition to the new product.

Often there are periods where the organization will run both the old product and the new product in parallel until the new one has proven that it is functioning correctly. This section of the specification is where you identify the tasks necessary for the period of transition to the new product. This section is input to the project planning process.

22a. Requirements for Migration to the New Product

Content

A list of the conversion activities. Timetable for implementation.

Motivation

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

Considerations

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

What kind of data conversion is necessary? Must special programs be written to transport data from an existing system to the new one? If so, describe the requirements for these programs here.

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

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

Is there a need to run the new product in parallel with the existing product?

Will we need additional or different staff?

Is any special effort needed to decommission the old product?

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

Form

A cross-reference between the development tasks, your project phases and the Product Use Cases and Atomic Requirements.

22b. Data That Has to Be Modified or 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.

Considerations

Every time you make an addition to your dictionary (see section 7), ask this question: Where is this data currently held, and will the new system affect that implementation?

Form

Description of the current technology that holds the data.

Description of the new technology that will hold the data.

Description of the data translation tasks.

Foreseeable problems.

23. Risks

All projects involve risk—namely, the risk that something will go wrong. Risk is not necessarily a bad thing, as no progress is made without taking some risk. Risk is only a bad thing if the risks are ignored and they become problems. Risk management entails 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.

Content

This section of the specification contains a list of the most likely and the most serious risks for your project. For each risk, include the probability of it becoming a problem and any contingency plans.

Motivation

To discover and manage the risks.

Considerations

Risks will undoubtedly change during the lifetime of a project. The better you understand the requirements the better you can identify which risks are most serious for your project. The project manager determines how to manage the risks but the requirements specialists and developers provide input on new risks and which risks are turning into problems.

Use your knowledge of the requirements as input to discover which risks are most relevant to your project. Use the Volere Requirements Knowledge Model (included with the download of Version 16 of the template) as a trigger for identifying relevant risks.

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

As an alternative, you may prefer to identify the single largest risk—the showstopper. If this risk becomes a problem, then the project will definitely fail. Identifying a single risk in this way focuses attention on the single most

critical area. Project efforts are then concentrated on not letting this risk become a problem.

This section of the book is not intended to be a thorough treatise on risk management. Nor is this section of the requirements specification meant to be a substitute for proper risk management. The intention here is to assign risks to requirements and show clearly that requirements are not free—they carry a cost that can be expressed as an amount of money or time, and as a risk. Later, you can use this information if you need to make choices about which requirements should be given a higher priority.

Form

A risk list or log. Risk models as defined in: DeMarco, Tom, and Timothy Lister. *Waltzing with Bears: Managing Risk on Software Projects.* Dorset House, 2003.

For each risk, include the probability of that risk becoming a problem. Capers Jones's *Assessment and Control of Software Risks* (Prentice-Hall, Englewood Cliffs, N.J., 1994) gives comprehensive lists of risks and their probabilities; you can use these lists 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

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, expressing the result as a monetary amount or time to build.

There is no best method to use when estimating. The important thing is to create your estimates using metrics directly related to the requirements. If you have specified the requirements in the way we have described, you will have the following metrics:

- 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 the work you do on your requirements, the more accurate your estimates will be. Your cost estimate is the amount of resources you estimate each type of deliverable will take to produce within your environment. You can create 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 vagueness by making the cost estimate a range rather than a single figure. You can use these metrics as the basis for estimating the time, effort, and cost of building the product. First you need to determine what each of these metrics means within the environment in which you are building the product. For example, do you know how long it will take you to do all the work necessary to implement a product use case? If you do not, then you can take one of the use cases and benchmark it.

As you increase your knowledge of the requirements, we suggest you try using function point counting—not because it is an inherently superior method, but because it is so widely accepted. So much is known about function point counting that it is possible to make easy comparisons with other products and other installations' productivity. For details on how to estimate requirements effort and costs, refer to *Mastering the Requirements Process,* Addison Wesley, 2011. Appendix C Function Point Counting: A Simplified Introduction

It is important that your client be told at this stage what the product is likely to cost. You usually express this amount as the total cost to complete the product, but you may also find it advantageous to point out the cost of the requirements effort, or the costs 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

This section specifies the user documentation that will be produced as part of the product-building effort. This is not the documentation itself, but a description of what must be produced. The reason for including this description is to establish your client's expectations, and to give your usability people and your users the chance to assess whether the proposed documentation will be sufficient.

25a. User Documentation Requirements

Content

List of the user documentation to be supplied as part of the product. Be careful not to waste time defining anything that has already been defined. Bear in mind that the requirements, especially the Product Use Cases, Atomic Requirements and definitions of data provide the input for the user documentation.

Motivation

To set expectations for the user manuals and to identify who will be responsible for creating it.

Examples

Technical specifications to accompany the product.

User manuals.

Service manuals (if not covered by the technical specification).

Emergency procedure manuals (e.g., the card found in airplanes).

Installation manuals.

Considerations

Which documents do you need to deliver, and to whom? Bear in mind that the answer to this question depends on your organizational procedures and roles.

For each document, consider these issues:

- The purpose of the document
- The people who will use the document
- Maintenance of the document

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?

Use the requirements that you have already specified as input to writing the user documentation. For example you have defined all the terms used in the requirements, then take advantage of this and use the same definitions and dictionary in the user documentation. Use the product use case (PUC) scenarios as the core of describing how a user can do a particular task. If someone else is writing the user documentation then show them how they can use a lot of the work that has already been done as the basis for user manuals.

25b. Training Requirements

Content

A description of the training needed by users of the product. Be careful not to waste time defining anything that has already been defined. Bear in mind that the requirements, especially the Product Use Cases, Atomic Requirements and definitions of data provide the input for the user training.

Motivation

To set expectations for the training. To identify who is responsible for creating and providing that training.

Considerations

What training will be necessary? Who will design the training? Who will provide the training? What are the plans for holding the training sessions?

Use the requirements that you have already specified as input to designing the training. For example you have specified what a product use case (PUC) must do. Use that PUC scenario or model as the basis for building training for the users who will be carrying out that task. You have defined all the terms used in the requirements, so take advantage of this and use the same definitions and dictionary in the user training.

26. Waiting Room

The waiting room holds requirements that will not, for one reason or another, be part of the initial release of the product. If you are competent at gathering requirements, your users may often be inspired to think of more requirements than you can fit within the constraints of the project. While you may not want to include all of these requirements in the initial version of the product, neither do you want to lose them. If you are doing iterative development then the waiting room is your backlog.

Content

Any type of requirement at any level of detail.

Motivation

To allow requirements to be captured, even though they will not be part of the current development. To ensure that good ideas are not lost. To manage a backlog.

Considerations

The requirements-discovery process often throws up requirements that are beyond the sophistication of, or time allowed for, the current release of the product. This section holds these requirements in waiting. The intention is to avoid stifling the creativity of your users and clients, by using a repository to retain future requirements. You are also managing expectations by making it clear that you take these requirements seriously, although they will not be part of the agreed-upon 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. As a requirement progresses closer to implementation, then you can spend more time on it and add details such as the cost and benefit attached to that requirement.

You might also prioritize the contents of your waiting room. "Low-hanging fruit"—requirements that provide a high benefit at a low cost of implementation—are the highest-ranking candidates for the next release. You would also give a high waiting room rank to requirements for which there is a pent-up demand. You can think of the waiting room as a way of managing your backlog.

The waiting room has a calming effect on everyone because it shows their ideas are being taken seriously. Your users and client know the requirements are not forgotten, merely parked until it is time to review them and make decisions about whether they will be incorporated in the product.

27. Ideas for Solutions

Ideas for solutions are obviously not requirements, but it is impossible when gathering requirements not to get ideas for how they might be implemented. Rather than discarding the ideas or—even worse—writing them as if they are requirements, a practical idea is to simply set aside a section of your specification for these ideas. Record each one faithfully in this area, and then hand them over, along with the requirements, to your designer.

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, prototypes and so on. The aim is to capture, with the least amount of effort, an idea that you can return to later.

Motivation

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

Considerations

While you are gathering requirements, you will inevitably have solution ideas; this section offers a way to capture them. Bear in mind that this section will not necessarily be included in every document that you publish.

Form

The ideas for solutions are an ideal subject for a blog. This is, after all, the place for ideas, and blogs are a great place for creativity and building on one another's ideas.