Tarmem Compnay

RIMDEV Company

System Requirement

Worker Application

TABLE OF CONTENTS

0 Preface – please read first 1

0.1 Purpose of this document 1

0.2 Use of this document 1

0.3 Function of System Requirement document 2

0.4 Related documents 2

0.5 Production of System Requirement document 3

0.6 Definitions, Acronyms and Abbreviations 3

1 Introduction 4

1.1 Purpose and scope of the document 4

1.2 Overview 4

1.3 References 5

1.4 Definitions, Acronyms and Abbreviations 6

2 Background 7

2.1 Function and Purpose 7

2.2 Environmental Considerations 7

2.3 Relation to Other Systems 7

2.4 Model Description 8

2.5 Relationship to Other Projects 8

2.6 General Constraints 8

3 Specific Requirements 10

3.1 Overview 10

3.2 Detailed Requirements 10

3.3 Pre-development Services 13

4 Hardware Requirements 14

5 Telecommunications Services 15

6 System capability 16

6.1 Functional Requirements 16

6.2 Interface Requirements 16

6.3 Operational Requirements 16

6.4 Security Requirements 16

6.5 Safety Requirements 17

6.6 Quality Requirements 17

7 System Management 18

7.1 Installation Support 18

7.2 Diagnostic Tools 18

7.3 Configuration Control, Release Control, Fault Reporting, etc. 18

7.4 Instrumentation, Reporting and Accounting 18

7.5 Tuning 18

7.6 Back-up and Recovery 19

7.7 Operational Control 19

7.8 Other Support Facilities and Tools 19

8 Operational Characteristics 20

8.1 Capacity Requirements 20

8.2 Performance Requirements 20

8.3 Availability Requirements 20

8.4 Reliability Requirements 21

9 System Architecture 22

9.1 Maintainability Requirements 22

9.2 Portability Requirements 22

9.3 Prescribed Components 22

9.4 Software Constitution and Structure 22

10 Documentation 23

11 Other Services 24

11.1 Training 24

11.2 Installation of Hardware and Software 24

11.3 Data Set-up 24

11.4 Parallel Running 25

11.5 Operational Support 25

11.6 Warranty 25

11.7 Maintenance 25

12 Developmental Requirements 26

12.1 Roles and Responsibilities 26

12.2 Phases 26

12.3 Verification and ValWorkertion Requirements 26

A. Requirements Traceability Matrix 28

B. Services to be provided by the Commission 29

Document Control 30

Document Signoff 30

Document Change Record 30

0 Preface – please read first

0.1 Purpose of this document

1. This document is a generic System Requirement document for use by Worker Project. It provides guWorkernce and template material which is intended to assist the relevant management or technical staff, whether client or supplier, in producing a project‑specific System Requirement document. It is also useful background reading for anyone involved in developing or monitoring the Worker Management System (WORKER‑MS).

0.2 Use of this document

1. This Preface is addressed to the users of this generic document and is not meant to be retained in any project‑specific System Requirement documents based on it.
2. The remaining sections (numbered 1, 2, 3,…) constitute a template that should be used to construct the project-specific System Requirement document.

* Text in normal case is in the most part “boilerplate” that can be retained, amended or deleted in the document.
* Text in italics provides instructions on how to complete a section and should be removed once the section is written.

1. The template should be used pragmatically, that is - where a section is not relevant it should be omitted. Conversely, the material contained in this document is not necessarily exhaustive; if there is a subject that is relevant to the WORKER Project, but is not included in this document, it should still be included.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a. “Summary” Properties | | |  |  |
|  | Title | Type of document (i.e. System Requirement) | | |
|  | Author | Author(s) of document | | |
|  | Keywords | Document reference (i.e. WORKER-MS-SR) | | |
| b. “Custom” Properties | | |  |  |
|  | Proj Id | WORKER Project | | |
|  | Project | WORKER Application Project (Client,woker,admin) | | |
|  | Contr Id | RIMDEV | | |
|  | Contract | RIMDEV Company | | |
|  | Version | V 1.0 | | |
|  | Date | Friday, January 17, 2020 | | |

0.3 Function of System Requirement document

1. A System Requirement document is intended to be used to record the complete requirement for a discrete development task within an WORKER Project.[[1]](#footnote-1) As such it may be used as the “technical” component of a procurement specification (e.g. Invitation To Tender), and it may be referenced or included in the subsequent contract. In all cases, including for development work done “in‑house” it is conceived as the primary warrant (or Terms of Reference) for the development team.
2. This particular template is specifically oriented towards development tasks which include a significant software development component. It covers many other things besides, but it is not intended to be applicable when the software development component is absent.

0.4 Related documents

1. The following documents for the WORKER Project concerned are specifically related to each System Requirement document.

|  |  |
| --- | --- |
| Project Definition document | Top level specification of the WORKER Project as a whole. |
| Global Implementation Plan | WORKER Project development plan, including details of funding, phasing and roll-out plans. |
| User Requirement document | The overall user requirements on which the specific requirements in the System Requirement document are based. |
| User Guide | Preliminary draft description of system operation from the user point of view. |

1. Note that all of these are likely to be at least “Reference Documents” within the System Requirement document (See 1.3.2), and that the User Requirement document in particular may possibly be an “Applicable Document” (See 1.3.1). It may also be appropriate to reference a draft Service Level Agreement (SLA) which specifies requirements for services to be delivered as part of the project. In any event, the WORKER-MS SLA template (WORKER-MS-SLA) provides useful guWorkernce on service criteria which should be considered when specifying requirements.
2. The document “WORKER Lifecycles” presents an overview of the documentation of the WORKER Programme and of the WORKER Projects and development contracts executed under its auspices.

0.5 Production of System Requirement document

1. Development within a given WORKER Project is usually contracted out; but even where it is done internally explicit terms of reference are needed. These are provided by the System Requirement document.
2. In some cases there is only one development contract within a given WORKER Project, but more usually development is in distinct phases, and sometimes there are parallel developments within a phase. Each distinct development contract within an WORKER Project should be the subject of a separate System Requirement document.
3. It is important that each distinct contract (or internal development activity) have a clearly defined scope: the work to be done must be coherent and self-contained if it is to be managed effectively and reliably.
4. Note, moreover, that the specification of the System Requirements for each development has to be done with careful attention to the need to ensure that the planned developments collectively, in conjunction with the non‑developmental components of the target system, can be expected to satisfy the defined User Requirement for the WORKER Project as a whole.
5. Because the production of a System Requirement document is therefore part of the overall system design process,

it is often found appropriate to consult (or at least notify) the users concerning the content of each System Requirement document

it is essential to perform appropriate design reviews and other quality control checks.

0.6 Definitions, Acronyms and Abbreviations

1. The following is a list of definitions for this template:

|  |  |
| --- | --- |
| *(System) Block diagram* | *a diagram showing the major components of the system with its interconnections and external interfaces.* |
| *Data flow diagram* | *a diagram showing how data flows around the system. Generally the components are shown as black boxes with only the data flows into and out of the component.* |
| *Verification* | *checks that each component meets its specific requirement, usually as defined in the design document* |
| *ValWorkertion* | *confirms that the system meets the requirements defined in the User requirement document* |

**The remainder of this document comprises a skeleton of a System Requirement document. Throughout there are *embedded instructions and guidelines in italics*. All such material should be omitted from the System Requirement document itself, as should the whole of this Preface.**

# Introduction

1. Tarmem Services brings Home Services at Your Doorstep. All you have to do is, Place Request and relax. Tarmem Services is for Home and Professional Services.  
     
   Repair and Maintenance - Plumbers, Electricians, Carpenters, Painters, Computer and Laptop Repair, Mobile Repair, Interior designer, Air Conditioner Repair, Washing Machine Repair, TV Repair and Other appliances Repair

## Purpose and scope of the document

1. d

## Overview

1. Present a complete overview of the structure and content of the rest of the document. In the simplest case:

##### The remaining sections of this documents are as follows:

|  |  |
| --- | --- |
| Section 2 | a general description of the background to the development and of factors that affect the product(s) and their requirements. |
| Section 3 | the specific requirements for the product(s). |

*OR, for purposes of illustration only, in a wider ranging and more complex development:*

|  |  |
| --- | --- |
| Section 2 | a general description of the background to the development and of factors that affect the product(s) and their requirements. |
| Section 3 | a top-level statement of the specific requirements for the product(s), elaborated as below |
| Section 4 | hardware requirements |
| Section 5 | telecommunications services |
| Section 6 | system capability |
| Section 7 | system management |
| Section 8 | operational characteristics |
| Section 9 | system architecture |
| Section 10 | documentation |
| Section 11 | other products and services |
| Section 12 | developmental requirements |

1. See Section 3 of this template for a discussion of options.

## References

1. This section should provide a complete list of all the applicable and reference documents. Each document should be identified by its title, author and date. Each document should be marked as applicable or reference. If appropriate, report number, journal name and publishing organisation should be included.
2. Alternatively, the list of documents may be put into an appendix. However, even in that case, a clear indication of the existence of applicable documents, and a pointer to where the list of them may be found, must as a minimum be included in this section.
3. Applicable documents would normally include, as a minimum, the Project Definition document or Preparatory Report, the Global Implementation Plan, and the User Requirement document, for the WORKER Project in question.
4. There should also be a reference to the document which states how changes to this document are controlled.

### Applicable Documents

1. List all the documents with which this system must comply.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Title** | **Author** | **Date** | **Issue** |
|  |  |  |  |  |
|  |  |  |  |  |

### Reference Documents

1. List all the documents referred to in this document, other than applicable documents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Title** | **Author** | **Date** | **Issue** |
|  |  |  |  |  |
|  |  |  |  |  |

## Definitions, Acronyms and Abbreviations

1. This section should provide definitions of all terms, acronyms, and abbreviations needed for this document, or refer to other documents where the definitions can be found.
2. Alternatively, this could be an appendix.

# Background

1. This chapter should describe the general factors that affect the product(s) and their requirements. It does not state specific requirements; its purpose is only to make those requirements easier to understand.

## Function and Purpose

1. This section should discuss the purpose of this product. It should expand upon the points made in Section 1.1.

## Environmental Considerations

1. This section should summarise:

* the physical environment of the target system, i.e. where is the system going to be used and by whom? The User Requirement Document section called 'User Characteristics' may provide useful material on this topic.
* the hardware environment in the target system, i.e. what computer(s) does the software have to run on?, what networks will be used?, will the different infrastructures in the different countries require extra software to run satisfactorily?
* the operating environment in the target system, i.e. what operating systems are used?, does it need to run on more than one systems or more than one version of a single system?
* the hardware environment in the development system, i.e. what computer(s) does the software have to be developed on?, will it be built on several systems and then assembled on another?
* the operating environment in the development system, i.e. what software development environment is to be used? or what software tools are to be used?

## Relation to Other Systems

1. This section should describe in detail the product's relationship to other systems. For example is the product:

* an independent system?
* a subsystem of a larger system?
* replacing another system?

1. If the product is a subsystem then this section should:

* summarise the essential characteristics of the larger system;
* identify the other subsystems this subsystem will interface with;
* summarise the computer hardware and peripheral equipment to be used.

1. A block diagram may be presented showing the major components of the larger system or project, interconnections, and external interfaces.
2. The User Requirement Document contains a section ‘System Perspective' that may provide relevant material for this section.

## Model Description

1. This section should include a top-down description of the logical model. Diagrams, tables and explanatory text may be included.
2. The functionality at each level should be described, to enable the reader to walkthrough the model level-by-level, function-by-function, flow-by-flow. A bare-bones description of a system in terms of data flow diagrams and low-level functional specifications needs supplementary commentary. Natural language is recommended.

## Relationship to Other Projects[[2]](#footnote-2)

1. This section should describe the context of this project in relation to other projects, past, current and future.
2. The project may be independent of other projects or part of a larger project. Any parent projects should be identified. A detailed description of the interface of the product to the larger system should, however, be deferred until Section 2.5.
3. If the product is to replace an existing product, the reasons for replacing that system should be summarised in this section.
4. Previous WORKER and OSN projects may have used applications, tools and techniques that are relevant for this project.
5. This section may also usefully identify any other relevant projects the developer is carrying out, or has carried out in the past, and also any projects the developer may be expected to carry out in the future, if known.

## General Constraints

1. This section should describe any items that will limit the developer's options for building the software. It should provide background information and seek to justify the constraints.
2. The use of applications, tools and techniques from other WORKER and OSN projects should be considered here. Consider Horizontal Actions and Measures and similar projects in other sectors. Though the requirement to use these could be classed as a constraint they also provide reduced costs as some of the work has already been done.
3. The User Requirement Document contains a section called 'General Constraints' that may provide relevant material for this section.

# Specific Requirements

## Overview

1. **Deliverables**. Always provide a complete top-level statement of what products and services the contract is to deliver.
2. These might encompass, for example,

* pre-developments services (such as Requirements Analysis)
* hardware
* software
* documentation
* other services such as training, system set‑up, operational running, and warranty.

1. **Requirements**. Provide an overview of the specific requirements pertaining to these deliverables.
2. Note, however, that this should generally not be treated as a list of identified requirements but only as a top-level index to the requirements which follow in later sections.
3. It might be presented, for example, as requirements for

|  |  |  |
| --- | --- | --- |
|  | *See Section . . .* | |
| *Pre-development services* | | *3.3* |
| *Hardware* | | *4* |
| *System capability* | | *5* |
| *System management functions* | | *6* |
| *Operational characteristics* | | *7* |
| *System architecture* | | *8* |
| *Documentation* | | *9* |
| *Other services* | | *10* |
| *The manner of conduct of the contract.* | | *11* |

## Detailed Requirements

1. **Structure.** The functional requirements should be structured top-down in this and any subsequent sections. The number of sections and the sequence of presentation throughout is entirely discretionary, depending upon the peculiarities of the specific development envisaged. The structure and sequence presented herein is illustrative rather than prescriptive.
2. Non-functional requirements can appear at all levels of the hierarchy of functions, and, by the inheritance principle, apply to all the functional requirements below them. Non-functional requirements may be attached to functional requirements by cross‑references or by physically grouping them together in the document.
3. A non-functional requirement which appears at a lower level, it overrides any requirement of that type that appears at a higher level.[[3]](#footnote-3) Critical functions, for example, may have more stringent reliability and safety requirements than those of non-critical functions.
4. **Language.** Specific requirements may be written in natural language. This makes them understandable to non-specialists, but permits inconsistencies, ambiguity and imprecision to arise. These undesirable properties can be avoided by using requirements specification languages, provided that specific precautions are taken to ensure that

* the System Requirement specification produced is consistent with the defined User Requirement
* no harm is caused by the System Requirement document‘s possible lack of comprehensibility to the users
* the language used is one that the prospective developers can reasonably be expected to understand
* it is undertood how the delivered system will be verified to conform with its specification.

1. **Identification.** Each statement should contain one and only one requirement
2. Each system requirement must have a unique identifier. Forward traceability to subsequent phases in the life cycle depends upon each requirement having a unique identifier.
3. The source of each system requirement must be stated, using the identifier of a user requirement, a document cross-reference, or even the name of a person or group. Backwards traceability depends upon each requirement explicitly referencing its source in the User Requirement Document or elsewhere.
4. **Verification.** Each requirement must be verifiable. A requirement is verifiable if some method can be devised for demonstrating that the system implements it correctly.
5. Verification should always be as objective as possible. For example statements such as:

* 'the software will work well';
* 'the product shall be user friendly';
* 'the output of the program shall usually be given within 10 seconds';

*are not readily verifiable because the terms 'well', 'user friendly' and 'usually' have no obvious objective interpretation.*

1. A statement such as: 'the output of the program shall be given within 20 s of event X, 60% of the time; and shall be given within 30 s of event X, 99% of the time', is more obviously verifiable because it uses concrete terms and measurable quantities. Even here, however, there is scope for discussion about what actual set of “event X” will be used in verification.
2. For these reasons, requirements (and particularly performance requirements) will potentially be stated at three levels:

* what “the user” actually wants (e.g. good performance)
* what “the user” will accept as a practical demonstration that this requirement is satisfied
* the means whereby the requirement will in practice be verified, during construction of the system, at the time of delivery (“Acceptance Tests”), and during operation.

1. Note, however, that this third level, the actual verification process, is typically left to be agreed between the developers and a customer representative in the course of development. And sometimes, of necessity, the second level too is postponed for later elaboration and negotiation.
2. Where verification depends on services provided outside the scope of the contract in question, it is desirable, and arguably essential, to specify permissible assumptions for the purposes of verification. If, for example, the contractor is not supplying the equipment, the operating system, the DBMS, the communications facilities, and the mains power, he cannot reasonably be expected to give absolute, unqualified, guarantees on performance, availability, etc.
3. It should be clearly indicated at what point verification will take place. The norm is that it will be at the time of completion of development (e.g. in the course of “Acceptance Testing”). However, it may also be, for example

* in the course of bid evaluation and contract negotiation (“The bidder shall …”)
* in the course of development
* during or at the end of a post-delivery warranty period.

1. **Provisional requirements.** Some requirements may be dependent on feedback from other stages in the lifecycle, like the technical design phase. These requirements should be marked 'TBC' (to be confirmed).
2. It should be clearly indicated on what factors confirmation will depend, and at what time and in what way it will take place.
3. **Non-essential requirements.** Each requirement should be marked as essential or non-essential. Essential requirements have to be met for the system to be acceptable. Non-essential requirements should be marked with a measure of desirability (e.g. scale of 1, 2, 3).
4. Note that it is permissible, in appropriate circumstances, to specify requirements concerning the number of non-essential requirements which must be satisfied.
5. Note also that non-essential requirements may or may not become contractual requirements. If they are promised in the bid then the contractor may become committed to satisfying them, and they may thus become mandatory requirements, even though not originally deemed essential. If they are not promised in the bid then they may remain as aspirations (“The contractor will endeavour …”), but in that case there is no guarantee that they will be met.
6. **Priority of requirements.** The priority of a requirement measures the order, or the timing, of the related functionality becoming available. If the delivery is to be phased, so that some parts come into operation before others, each requirement must be marked with a measure of priority.

## Pre-development Services

1. The norm is to have a contract for the development of defined products in accordance with the exact specifications of a System Requirement document. In some cases, however, it may be found expedient to contract at the same time for some pre‑development services such as

* Feasibility study
* Benchmarking study
* Requirements Analysis
* Project Planning
* Evaluation of available products
* Technological trials
* Development of demonstrator (system mock-up).

1. In such circumstances it is highly desirable, and arguably essential, to

* include delivery of a revised (expanded) System Requirement document as an additional pre‑development service
* specify the procedure for the review of that document and the authorisation of the continuation of work on the basis of it.

# Hardware Requirements

1. Either the contract delivers hardware or it does not.[[4]](#footnote-4) If it does, specify the requirements here. If it does not then omit this section. (Constraints on platform, operating system, etc. appear further down.)
2. Specify what types and numbers of items of equipment are to be delivered. Specify for each of them

* functionality
* standards
* interfaces
* performance
* capacity
* expansibility
* reliability and availability
* durability
* maintainability
* running cost limitations
* operational requirements
* etc.

1. For processors, specify any operating system and other pre‑loaded software required as part of the hardware package, or, if more appropriate, include such things as part of the specification of software requirements or system architecture (See below).

# Telecommunications Services

1. Either the contract undertakes to deliver new telecommunications facilities or it does not. If it does, specify the requirements here. If, as is more likely, it does not, but makes use of existing telecommunications services, then omit this section. Or, if the volume of material justifies it, transfer the telecommunications interface requirements specifications from 6.2 below.

# System capability

## Functional Requirements

1. These define the purpose of the system. The Functional requirements are derived from the logical model, which is in turn derived from the user’s capability requirements. In order that they may be stated quantitatively, the functional requirements may include performance attributes.

## Interface Requirements

1. These specify hardware, system or database elements with which the system must interact or communicate (as described in Section 2.3).
2. Interface requirements should be classified into software, hardware and communication interfaces.
3. Software interfaces could include operating systems, software environments, file formats, database management systems and other software applications.
4. Hardware interface requirements may specify the hardware configuration.
5. Communications interface requirements constrain the nature of the interface to other hardware and software. They may demand the use of a particular network protocol, for example.
6. External interface requirements should be described or referenced in Interface documents. User interface requirements should be specified under ‘Operational requirements’ (see below). Interface requirements can be illustrated with system block diagrams.

## Operational Requirements

1. These specify how the system will run and how it will communicate with the human operators. Operational requirements include all user interface, usability and human-computer interaction requirements as well as the logistical and organisational requirements. Examples are: the screen layouts, the content of error message, help systems, etc. It is often useful to define the semantics and syntax of commands.

## Security Requirements

1. These specify the requirements for securing the system against threats to confidentiality, integrity and availability.
2. Examples of security requirements are interlocking operator commands, inhibiting of commands, read-only access, password system and computer virus protection.
3. Pay particular attention to telecommunications‑related security requirements, including requirements for

* data encryption
* authentication of users and equipment
* protection against viruses and other malevolent agents.

1. The level of physical protection needed of the computer facilities may also be stated e.g. backups are to be kept in fire-proof safe off-site.

## Safety Requirements

1. These specify any requirements to reduce the possibility of damage that can follow from system failure. Safety requirements may identify critical functions whose failure may be hazardous to people or property.

## Quality Requirements

1. These specify attributes of the system that ensure that it will be fit for its purpose (other than the major quality attributes of reliability, maintainability and safety, which should always be specified). Where appropriate system quality attributes should be specified in measurable terms, i.e. with the use of metrics.

# System Management

1. Over and above the system functionality required by the end users of the system in question, there are likely to be numerous functional requirements for facilities to support the operational management of the system itself. The following subsections provide examples of such facilities, and should provide a useful checklist for consideration.

## Installation Support

1. Specify any requirements for mechanisms to enable the installation and configuration of the whole system, or of any replacement components of it.

## Diagnostic Tools

1. Specify any requirements for tools to verify the correctness of system operation.
2. Specify any requirements for tools to help diagnose the cause of any system faults encountered.

## Configuration Control, Release Control, Fault Reporting, etc.

1. Specify any requirements for facilities to register and manage the specific collection of software components which make up the operational system, and to regulate changes to that collection.
2. Specify any requirements for facilities to assist the registration and management of reports of system malfunction.

## Instrumentation, Reporting and Accounting

1. Specify any requirements for mechanisms to obtain, display and collect measures of system behaviour and performance.
2. Specify any requirements for tools to assist the analysis and reporting of the information thus collected.
3. Specify any requirements for facilities to support system usage accounting, such as the apportionment of system usage costs between different user groups.

## Tuning

1. Specify any requirements for mechanisms to adjust those operational characteristics of the system which affect its performance.

## Back-up and Recovery

1. Specify any requirements for mechanisms to enable the creation of appropriate back‑up copies of key data, and its restoration after system failure or malfunction.
2. Note that it is may be found prudent to include an explicit requirement that it should always be possible to recover from system failure during recovery.

## Operational Control

1. Specify any requirements for facilities to support control of system operation.
2. These may include such things as registration of users, granting of usage permissions, imposition of access controls, allocation of resources, adjudication on conflicts, correction of data corruption, and monitoring of activity levels.
3. Pay particular attention to requirements for facilities related to the telecommunications aspects of operational control, including

* establishment and verification of links to new players
* supervision and control of network operation
* diagnosis of malfunctions
* status reporting
* system accounting and charging.

## Other Support Facilities and Tools

1. Do not assume that the above list of possible system management requirements is necessarily exhaustive. Specify any other system management requirements for the specific development in question.

# Operational Characteristics

## Capacity Requirements

1. Specify requirements pertaining to physical resources such as processing power, main memory, disk space, etc.
2. Note that, depending on circumstances, it may be appropriate to specify these

* in application terms (e.g. population sizes and transaction rates)
* in physical terms (e.g. MIPS and megabytes).

1. It may also be found expedient to accompany these specification with expansibility requirements, i.e. by how much the system architecture should make provision for the stipulated capacity to increase.[[5]](#footnote-5)

## Performance Requirements

1. These specify numerical values for measurable variables, e.g. rate, frequency, capacity and speed. Performance requirements may be incorporated in the quantitative specification of each function, or stated as separate requirements. Qualitative statements are unacceptable, e.g. replace ‘quick response’ with ‘response time must be less x seconds for y% of the cases with an average response time of less than z seconds’. The performance attributes may be presented as a range of values, for example, the

* worse case that is acceptable;
* nominal value, to be used for planning
* best case value, to indicate where growth potential is needed.

## Availability Requirements

1. The system is likely to be intended for operation during specific periods (times of day, days of year). Specify for what periods what system functions should be designed to be available.[[6]](#footnote-6)
2. Specify what breaks in service are tolerable (frequency, duration).
3. Specify how users may come to know that some or all facilities are (temporarily) unavailable. Specify how they will come to know when they again become available.
4. Specify any requirements for facilities to enable users to work through a period of unavailability, and to return to full system usage when it is again available.
5. Specify any requirements for facilities to enable availability levels to be monitored.

## Reliability Requirements

1. In addition to availability requirements, it is usually found expedient to specify reliability requirements.
2. These specify the acceptable mean time interval between failures of the system or of specific components of it, averaged over a significant period (MTBF). They may also specify the minimum time between failures that is ever acceptable. Reliability requirements may have to be derived from the user’s availability requirements.
3. Note that reliability requirements may be verified at different points in the course of the contract. For example:

* in the course of bid evaluation it may be verified, on the basis of available data on the reliability of proposed components, that the system could be capable of offering the required reliability;
* the bidder might even be asked to perform reliability demonstrations in the course of bid evaluation;
* the may be extensive load tests in the course of Acceptance Testing, designed to demonstrate reliability among other things
* operational statistics may be collected during a system warranty period, to be analysed prior to final conclusion of the contract (and preferably before final payment).

# System Architecture

1. It is usually necessary to specify requirements which go beyond the functions and operational characteristics of the system to be delivered and dictate or constrain the technical architecture of the system itself.
2. This section should describe any items that will limit the developer's options for building the software. It should provide background information and seek to justify the constraints.
3. The User Requirement Document contains a section called 'General Constraints' that may provide relevant material for this section.

## Maintainability Requirements

1. These specify how easy it is to repair faults and adapt the software to new requirements. The ease of performing these tasks should be stated in quantitative terms, such as mean time to repair a fault (MTTR). They may include constraints imposed by the potential maintenance organisation. Maintainability requirements may be derived from the user’s availability and adaptability requirements.

## Portability Requirements

1. These specify the ease of modifying the software to execute on other computers and operating systems. Possible computers and operating systems, other than those of the target system should be stated.

## Prescribed Components

1. The use of applications, tools and techniques from other WORKER and OSN projects should be considered here. Consider Horizontal Actions and Measures and similar projects in other sectors. Though the requirement to use these could be classed as a constraint they also provide reduced costs as some of the work has already been done.

## Software Constitution and Structure

1. In some cases it may be found expedient to make explicit stipulations concerning software constitution and structure.[[7]](#footnote-7) Such constraints would generally be in the interests of maintainability or portability, but there may be other reasons for them, and it may be deemed prudent to make them explicit requirements rather than leaving them to the discretion of the developer.

# Documentation

1. Specify the project-specific requirements for documentation , including those contained in referenced standards, for example, the content and layout of the User Guide.
2. Required documentation may include such things as:

* user training documentation
* user reference documentation
* documentation for system management and operational support
* release notes
* configuration control files
* documentation for system maintenance
* supplier reference information
* warranties.

1. Specify numbers of copies required and medium of documentation. Note that this may include

* paper documents
* CD-ROMs or magnetic disks
* installed “Help” files.

1. Specify, where appropriate, the form in which digital documentation may be presented. Note that this may include, for example

* text files for a specific Word processor, possibly in accordance with specific documentation standards
* data files for a specific DBMS or spreadsheet package
* PDF files or other image files.

# Other Services

1. There are numerous other services which the developer may be commissioned to provide during or after development. The checklist below is by no means guaranteed to be exhaustive. (See also the WORKER-MS Service Level Agreement (SLA) template (WORKER-MS-SLA) which lists other service elements to be considered, and also provides guWorkernce on how they should be specified).
2. Under each head there is a potential need to specify

* when the contractor will produce a detailed specification of what is to be delivered
* the procedure for approval of that detailed specification
* the reponsiveness and level of service to be provided by the contractor
* the procedure for complaints and dispute resolution.

## Training

1. Will the developer provide any training courses? Will he deliver any material for use in training courses?
2. Note that training may be required, for example, for

* users
* user support staff (Help Desk)
* system operators
* maintenance programmers
* trainers.

## Installation of Hardware and Software

1. What will actually be delivered and/or handed over by the developer? Will he himself install all required hardware and software, and hand it over in a demonstrably working state? Or, at the other extreme, will just send a disk containing software and documentation?

## Data Set-up

1. Will the developer set-up all required initial data in system files? Will he do any of this? Will he support or assist the users in doing it?

## Parallel Running

1. Is there any proposal to operate the old and new systems in parallel for a period? Will the developer have any involvement in such activity?

## Operational Support

1. Will the developer provide any assistance during initial operation of the system? Will he, for example, provide staff for the Help Desk? Or will he provide a second‑level Help facility, for use only by the front‑line Help Desk?

## Warranty

1. For how long after the designated completion date of the development will the developer provide any warranty of the correctness of what he has delivered? What will be the procedures for calling on this warranty? How will it be judged whether a warranty claim is valid? What levels of service are required for corrections made under warranty? What warranty is required on such corrections?

## Maintenance

1. Will the developer be required to perform any system maintenance activities other than those performed under warranty?
2. Will these include upgrades to bought‑in components? If so, will there be a charge?
3. Can these include changes to the system specification? If so, how will the developer’s effort be charged?
4. Will there possibly be a requirement for user‑initiated changes to the system specification during warranty? If so, how will these be handled?

# Developmental Requirements

1. Specify any requirements relating to the conduct of the contract. In other words, having specified in detail the products and services the contractor is to deliver, including both what they will be and (to the extent that this is of concern to the users) what designs, components, methods and tools will be used in their construction, we now turn to any pertinent stipulations relating to the way in which the work of the contract will be organised and regulated.

## Roles and Responsibilities

1. It is often appropriate to stipulate requirements concerning the developer’s team structures and roles. In particular, it may be desirable to ensure that

* appropriate points of contact are defined, for liaison on contractual, technical and other matters
* designated people have authority to make decisions which may be necessary in the course of development
* adequate quality control mechanisms will be in place within the developer’s organisation
* all other necessary tasks (such as the continuing role of system designer/architect) will be performed in the course of development.

## Phases

1. If products of development are to be delivered in phases, specify as precisely as expedient what these shall be. If some flexibility is permitted, specify when and how their content will be fully defined.

## Verification and ValWorkertion Requirements

1. Specify any constraints on how the system is to be verified (i.e. shown to conform with the defined System Requirement). The verification requirements constrain the Test Management Plan. They might include

* rights of access to and inspection of documentation and work in progress
* requirements for simulation, emulation, live tests with simulated inputs, live tests with real input, and interfacing with the testing environment.

1. Such tests may be performed

* on the developer’s premises prior to delivery
* on the customer’s premises on test equipment
* on operational equipment.

1. Specify

* who will define the precise test plans and test data and when, how they will be approved, and to whom they will be available prior to the tests[[8]](#footnote-8)
* the criteria for the grading, if any, which will be applied to faults revealed during testing, and by whom the grading will be done
* the precise criteria by which it will be judged that a test has been passed[[9]](#footnote-9) or failed
* which tests, if any, may result in an intermediate sign-off of work done (i.e. a certified partial completion of the developer’s contractual obligation)
* time allowed for customer comments on delivered products and the results of tests
* procedures for dispute resolution
* procedures for the management and correction of faults revealed during verification.

1. Specify also any “valWorkertion” requirements, i.e. requirements for the developer to check that the system is what the users wanted. Note, however, that since these would implicitly be requirements to check on the validity of the System Requirement to which the developer is working, it is unlikely that the developer would be held responsible for any valWorkertion failure.
2. Specify when, by what means, and on what criteria, the development will achieve its Final Acceptance, i.e. the final certification that the developer has no further obligations under the contract in question.

A. Requirements Traceability Matrix

1. This section should contain a table that summarises how each user requirement from the User Requirement document has been met in this document.
2. The tabular format permits one-to-one and one-to-many relationships to be shown (I.e. “This User Requirement is satisfied by means of the following System Requirement(s):”.
3. Many‑to‑one relationships (“This System Requirement supports the following User Requirement(s):”) may potentially be exhibited by presenting the table in sequence of System Requirement rather than of User Requirement.
4. Many‑to‑many relationships will never be visible as such; and if there were a strong requirement for tools to assist their tracking and analysis then the “traceability matrix” would have to be stored in an appropriate database rather than as a simple table.

|  |  |  |  |
| --- | --- | --- | --- |
| User Req Number | User Req Item | System Req Number | System Ref Item |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

B. Services to be provided by the Commission

1. Specify any tools, services or facilities which the contractor can assume will be provided by or on behalf of the European Commission.
2. For example, this may possibly include, in particular

* equipment (for development, testing and operation)
* software licences (for tools and components)
* telecommunications facilities
* software from earlier systems
* office facilities and services
* staff time (for reviews, tests, meetings, etc.).

1. Note that these are NOT “requirements”, and should not be identified as such, which is why they are, in this template, consigned to an appendix. But there are other ways in which they may be presented.

Document Control

|  |  |
| --- | --- |
| **Title:** | System Requirement |
| **Issue:** | Issue 1 |
| **Date:** | 17 January 2001 |
| **Author:** | John Brinkworth, Bill Waghorn |
| **Distribution:** | Project Sponsor Project Team |
| **Reference:** | WORKER-MS-SR |
| **Filename:** | WORKER-MS-SR-i1 |
| **Control:** | Reissue as complete document only |

Document Signoff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nature of Signoff** | **Person** | **Signature** | **Date** | **Role** |
| Author | John Brinkworth, Bill Waghorn |  |  | Project Manager, Consultant |
| Review | Mark Pillatt |  |  | Consultant |

Document Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Change Details** |
| 16 November 2000 | Issue 1 Draft C | John Brinkworth, Bill Waghorn | First issue |
| 17 January 2001 |  | Bill Waghorn | Update in the light of review comments, plus further review by Mark Pillatt and reformatting by Sue Turner |

1. In this document we always to refer to IDA Projects as “IDA Projects”, with the “IDA” prefix and the capital “P”. This is to avoid any avoidable confusion between IDA Projects and any other activities, including contracts executed within IDA Projects, which may be referred to as “projects” by those working on them. [↑](#footnote-ref-1)
2. Note that “project” is used in this section in the normal English sense, and does not necessarily refer to an IDA Project. It might possibly have been replaced by “contract”, except that the projects in question will not necessarily all be contracts. [↑](#footnote-ref-2)
3. Note that this applies only within the specific System Requirement document. It does not imply, for example, that requirements in a System Requirement document override or supersede those in the User Requirements document to which it is subordinate. [↑](#footnote-ref-3)
4. Note that, as a principle, IDA itself will not supply hardware. This is considered part of (sectoral) infrastructure, and hence any hardware required for an IDA Project has to be paid for from sectoral budgets or other sources. [↑](#footnote-ref-4)
5. It should not be assumed that such increases would be free of charge. However, it is unlikely that the development would be deemed to have made provision for them if their cost were more than a proportionate increase in the price of the hardware components affected. [↑](#footnote-ref-5)
6. For example, a system designed for 24x7 usage may differ substantially from one suitable only for use during “normal working hours”. And the facilities offered to the user in a 24x7 system may be different if the help desk or the system operators are not always available. [↑](#footnote-ref-6)
7. For example: “The system must use Oracle for …”; “The following parameters must be stored as data in the database or other files, rather than embedded in the source code of the software …”; “All source code shall conform with the EC’s C++ coding standards”; “The following functions must be implemented as Jaffa Biscuits …”; “There shall be a separate software package to perform the xyz function, with an spqr interface”; etc. [↑](#footnote-ref-7)
8. Be wary, for example, of letting programmers have prior access to Acceptance Test data. It is not unknown for them to know two years in advance precisely what the Acceptance Tests will be, in which case the Acceptance Test data becomes the *de facto* functional specification. [↑](#footnote-ref-8)
9. For example “No Grade A faults and not more than 12 Grade B faults”. [↑](#footnote-ref-9)