Template for IDA Project (Project Id)

Template for specific development (Contract ID)

User Requirement

Issue 1

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 The User Requirements document 2

0.4 Definitions, acronyms and abbreviations 3

0.5 Related documents 3

1 Introduction 5

1.1 Purpose of the document 5

1.2 Overview 5

1.3 References 6

1.4 Definitions, acronyms and abbreviations 6

2 General Description 7

2.1 Objective 7

2.2 System concept 7

2.3 Concept of operations 8

2.4 Users and user interface 8

2.5 Operational characteristics 9

2.6 Control, support and maintenance 9

2.7 System architecture and construction 9

2.8 Assumptions and dependencies 9

3 Specific Requirements 11

3.1 General 13

3.2 User facilities 13

3.3 System interfaces 14

3.4 Communications requirements 14

3.5 Hardware requirements 14

3.6 Capability requirements 14

3.7 System management functions 15

3.8 Other products 15

3.9 Constraint requirements 15

Document Control 18

Document Signoff 18

Document Change Record 18

0. Preface – please read first

0.1 Purpose of this document

1. This document is a generic User Requirement document for use by IDA projects. It provides guidance and template material which is intended to assist the relevant management or technical staff, whether client or supplier, in producing a project‑specific User Requirement document. It is also useful background reading for anyone involved in developing or monitoring the IDA Management System (IDA‑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 User 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 User 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 project, but is not included in this document, it should still be included.
2. This document has been prepared using MS Word 97. The following variables are currently recorded as File “Properties” under MS Word. They may be modified by that means or overwritten directly at each occurrence in the document, at the discretion of the user.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a. “Summary” Properties | | |  |  |
|  | Title | Type of document (i.e. User Requirement) | | |
|  | Author | Author(s) of document | | |
|  | Keywords | Document reference (i.e. IDA-MS-UR) | | |
| b. “Custom” Properties | | |  |  |
|  | Proj Id | Short mnemonic of IDA Project (set, in this document, to “**Project Id**”) | | |
|  | Project | Full name of IDA Project (set, in this document, to “**Template for IDA Project**”) | | |
|  | Contr Id | Short identifier of contract (set, in this document, to “**Contract ID**”) | | |
|  | Contract | Full name of contract (set, in this document, to “**Template for specific development**”) | | |
|  | Version | Issue number (currently Issue 1) | | |
|  | Date | Date of document (currently 17 January 2001) | | |

0.3 The User Requirement document

1. The function of a User Requirement document is to serve as the mandate or terms of reference for the design, development and realisation of the technical component of a system or subsystem within an IDA Project. Usually there is only one User Requirement document for each IDA Project, even when the required development encompasses more than one development contract.
2. A User Requirement document is produced as a result of appropriate Requirements Analysis activity, based on the stipulations of the Project Definition[[1]](#footnote-1) document and the Global Implementation Plan. All specific requirements in the User Requirement document must be consistent with similar statements in higher-level specifications, if they exist.
3. The User Requirement document is the primary input to subsequent System Design work and to the procurement specifications for pertinent system development contracts.
4. Requirements Analysis will generally include

* review of all relevant documents (including the Project Definition document and the Global Implementation Plan),
* establishment of an initial system concept, if not already done[[2]](#footnote-2)
* consultation of users
* consolidation of inputs
* negotiation of a definitive statement of User Requirements.

1. All users, both end-users and operators/maintainers, should be consulted. Consultation may be by any expedient means, including interviews, workshops, questionnaires, and e‑mail.
2. The “user” in the title of this document is a deliberately vague term intended to encompass all parties who have a legitimate interest in the system envisaged, including even customers who will never see it. They are thus sometimes called “stakeholders”. They may include, for example

* those who will actually operate the telematics system envisaged
* those who will operate or manage the processes which the telematics system will support
* the external “customers” of those processes
* those who will support the system or its users, or provide the platforms on which it runs.

1. It must be recognised, however, that users are generally not directly consulted in the course of Requirements Analysis, but only through designated representatives. Some representatives are more closely linked to the population they represent than others are, of course. Nor are all designated representatives fully enfranchised within the development process. In particular, only a limited number have the power to accept or reject the specifications which are produced.
2. It is important that the nature of the entire user population, the manner in which they were represented during consultation and negotiation, and the way in which specification of User Requirements was (or will be) approved, be spelt out as completely as possible in the User Requirement document.
3. There may be a conflict of requirements, where there are multiple stakeholders and users. These must be resolved before this document is finalised.
4. The final decision on approval of the specification of User Requirements should be with the pertinent Sectoral Committee. Therefore:

* the User Requirement document should not be circulated as other than an unauthorised draft until that approval has been given;
* the User Requirement document should be in all respects presented at a level and in a manner suitable for evaluation and approval by the Sectoral Committee.

0.4 Definitions, acronyms and abbreviations

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

|  |  |
| --- | --- |
| Context diagram | a diagram showing how this system fits with its external interfaces |
| System Block diagram | a diagram showing the major components of the system with its interconnections and external interfaces (i.e. a first-level elaboration of the Context diagram) |
| Capability requirements | what the system has to do |
| Constraint requirements | what other requirements the system has to satisfy. |

0.5 Related documents

1. The document “IDA Lifecycles” presents an overview of the documentation of the IDA Programme and of the IDA Projects and development contracts executed under its auspices.
2. Within that, the following documents are specifically related to the User Requirement document.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a. For the IDA Project | | |  |  |
|  | Project Definition document |  | | |
|  | Global Implementation Plan |  | | |
|  | User Guide |  | | |
| b. For a development contract within an IDA Project | | |  |  |
|  | System Requirement document |  | | |

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

# Introduction

## Purpose of the document

1. This section should:

define the purpose of this document;

specify the intended readership of this document.

*For the User Requirement these would normally (or at least typically) be as follows:*

##### This document is the definitive specification of the user requirements for telematics facilities to be developed under the IDA Project “*Template for IDA Project*” (*Project Id*). It is a primary input to the technical development of those facilities, and the primary specification of the criteria against which the acceptability of those facilities will be evaluated after they have been developed.

1. <It is sometimes found appropriate to specify the provenance of the document at this point, and how it fits into the scheme of things.>

##### This document is intended to be read by

###### all responsible for the management of the developments in question, including participants in the *Project Id* Sectoral Committee and members of the IDA Unit

###### users, user representatives, and other interested parties, in Member States and in European Institutions

###### contractors who undertake all or parts of the development, as optional illumination of the source of and background to the specific requirements to which they are working.

## Overview

1. This section should provide an outline of the scope of the IDA Project and an overview of the entire User Requirement document.
2. The IDA Project may consist of the production of a system, i.e. both hardware and software, or of software only. There are also likely to be requirements for other things, such as technical documentation, user documentation, user training, system installation, data load, initial operation, etc. These will be the product(s).

Identify all product(s) to be provided;

explain what the product(s) will do (and will not do, if necessary);

1. Note, however, that this subsection is intended to be just an identification of the scope of the IDA Project, not a specification of the products.
2. Then go on to give a guide to the structure and content of the document. Typically:

##### Section 2 provides a general description of the product(s) and of the factors that affect their requirements.

##### Section 3 contains the specific requirements for the product(s). <Or, if the volume and nature of requirements makes it appropriate to arrange them in multiple sections, list them all here, and explain the scope of each.>

##### Section 1.3 below contains a list of all documents referenced in this document.

##### Section 1.4 below *<or a designated appendix>* contains a glossary of pertinent terms and abbreviations.

## References

1. This section should provide a complete list of all the applicable and reference documents, identified by 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. There should be references to European standards and publicly available specifications, such as open Internet Standards, as appropriate, and also to the document which states how changes to this document are controlled.
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. Because the User Requirement document is a key component of the body of documentation for the IDA project, it must necessarily contain references. But it also needs to avoid excessive dependence on the content of those other documents. Remember that it is a user document. **It should be entirely comprehensible to a reader who does not have access to any of the referenced documents.**

### 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 the definitions of all terms, acronyms, and abbreviations, or refer to other documents where the definitions can be found.

# General Description

1. This chapter should describe the general factors that affect the product(s) and their requirements. This chapter does not state specific requirements but should make those requirements easier to understand.
2. The sequence of presentation may vary, depending on the nature and circumstances of the IDA Project. However, all topics identified should be covered in one way or another: it is generally preferable to say “None” or “Not applicable” than to allow it to be thought that some aspects of requirements might have been overlooked.

## Objective

1. Specify what the IDA Project is intended to achieve. Describe relevant benefits, objectives, and goals as precisely as possible. This covers much the same ground as corresponding material in the Project Definition document and the Global Implementation Plan, but it is addressed to the user community rather than to those responsible for the development.

## System concept

1. Give a preliminary idea of what components the system is expected to have, and how they will be used; for example, location and nature of networked equipment, location of data and processing, type of user terminals, software infrastructure, etc.
2. Describe the environment in which the system is to operate. This description may be supported by context diagrams, which summarise the external interfaces, and system block diagrams, which show how the activity fits within the larger system. The nature of the exchanges with external systems should be specified. If the system is to be a component of a parent system or project then this section should:

* outline the activities that will be supported by external systems;
* reference the interface documents that define the external interfaces with the other systems;
* describe the telematics infrastructure to be used.

1. If the system is ‘standalone’, it should be stated here.
2. And what is the starting point for the IDA Project? How much of the system already exists or is being developed elsewhere? How much can be procured on the market? Most particularly, how much of it needs to be developed by the IDA Project? Note that the defined User Requirements should in general be assumed to be requirements for the overall system when it is completed. On the other hand, the defined requirements for subsequent procurement contracts, even if there is only one of them, will only be for the new things to be added. Consequently, there is often a real risk that the presumed (or imposed) foundations for the new system will actually make it impossible to satisfy the defined User Requirement[[3]](#footnote-3). Any such risk should be identified during Requirements Analysis and documented in this section of the User Requirement document.
3. This section should also put the system into perspective with other related systems. If the system is to replace an existing system, the existing system should be described and referenced. Ancestors of the system that are no longer in use might be mentioned.

## Concept of operations

1. Describe the main capabilities and why they are needed.
2. Describe the processes to be supported by the system, indicating those parts of the processes where it is used, and how, and to what effect.
3. If the processes are assumed to be different from those currently in operation, specify what the changes are, and how it is envisaged that they will be brought about. It is very important that such things not be left to chance. Practical experience makes clear what should be obvious, that a new system is likely to fail unless the organisational changes it assumes or requires are actively managed to achieve the desired benefits. Projects tend to stumble more over getting the people and their processes to change than over the technology.

## Users and user interface

1. Describe those general characteristics of the users affecting the specific requirements.
2. Many people may interact with the system during its operations and maintenance, e.g. users, operators and maintenance personnel. Certain characteristics of these people, such as educational level, language, experience and technical expertise impose important constraints on the software.
3. The system may be frequently used, but some individuals may use it only occasionally. Frequent users will become experts whereas infrequent users may remain relative novices. It is important to classify the users and estimate the likely numbers in each category. If absolute numbers cannot be stated, relative numbers can still be useful.
4. Environmental characteristics may also need to be described. It should not be assumed that users are always in quiet private offices. For example, the required system may include terminals on service counters, kiosks in public places, call centres, or other forms of user interface which potentially have a significant impact on requirements.

## Operational characteristics

1. Present the background to, and rationale for, requirements for such things as

* Performance
* Reliability, availability, recovery
* Confidentiality, security
* Running costs (e.g. consumables)

## Control, support and maintenance

1. Specify the presumed mode of organisation and execution of system operation, support, and maintenance, as background to, and rationale for, requirements for such things as

* System set-up
* Instrumentation[[4]](#footnote-4), accounting[[5]](#footnote-5), tuning
* Change management, configuration management, release control
* Etc.

## System architecture and construction

1. Describe any items that will limit the developers' options for building the system.
2. This section should not be used to impose specific requirements or specific design constraints, but should state the reasons why certain requirements or constraints exist. Requirements and constraints may be geographical or political.
3. The use of applications, tools and techniques from other IDA and OSN projects should be considered here. Consider Horizontal Actions and Measures and similar projects in other sectors. Make use of the IDA Architecture Guidelines[[6]](#footnote-6).

## Assumptions and dependencies

1. List the assumptions that the specific requirements are based on.
2. Risk analysis should be used to identify assumptions that may not prove to be valid. For example, an interface may be specified with a system that does not yet exist. If the production of the system does not occur when expected, this document may have to change.
3. This subsection should specify (or reference specifications of) any significant capacity limitations which may be assumed. For example, such things as the number of Member States, the frequency of user enquiries, or even the reliability of the public electricity supply, may be critical factors in the specification of detailed requirements.

# Specific Requirements

1. Specific requirements should be described in this section, which is the core of the document.
2. **Sequence and structure.** Although the scope of the material to be included in this section is fairly clear, there is no “right” form of organisation for its presentation. The author should aim to make it as comprehensible as possible (particularly to the non‑technical reader), with a natural, logical flow, and a minimum of cross‑references. And that does imply a sequence which may be very different for different kinds of IDA Project. The structure presented in this section is offered as one plausible option, but it is by no means the only way of doing it, and is certainly not prescriptive.
3. **Technical requirements.** The User Requirement document is a specification of requirements from the user point of view, and its contents are thus essentially non‑technical.
4. It is not mandatory for the specification to include any technical elements. However, the users often do have technical requirements, and when they do such requirements have to be included in the User Requirement document. But even then they must be presented so as to be capable of being understood by the non-technical reader.
5. The users will usually rely upon the services of appropriate technical advisors to help in the specification of such requirements.
6. The IDA Unit may also be able to provide or recommend some pre‑existing re‑usable specifications of the more technical aspects of user requirements. Examples include statements of requirements for telecommunications, security, reliability, recovery, and system management, and specifications for performance and support requirements.
7. **Identification of requirements**. Each statement should contain one and only one requirement.
8. Each requirement must be uniquely identified. Forward traceability to subsequent phases in the life cycle depends upon each requirement having a unique identifier.
9. The source of each user requirement must be stated. The source may be defined using a document cross-reference or even the name of a person or group.[[7]](#footnote-7) Backwards traceability depends upon each requirement explicitly referencing its source.
10. **Acceptance**. The acceptability of the system will be assessed against these specific requirements.
11. Note that this applies not just to the conduct of Acceptance Tests at the end of system development, but also to

* QA and other verification processes executed during system construction
* ex post facto evaluation of the system.

1. Note, however, that systems are often built from the products of multiple development contracts, and in addition nearly always make use of pre‑existing foundations. In these circumstances it is possible for the final system to fail to satisfy the defined User Requirement even though all contributory developments satisfy the requirements defined for them[[8]](#footnote-8). The consequent scope for user disappointment must be minimised by appropriate attention to the risks during Requirements Analysis and subsequent procurement specification, and by system/architecture design and Quality Assurance activities throughout the IDA Project.
2. **Verifiability**. Requirements need to be verifiable. A user requirement is verifiable if some method can be devised for demonstrating that the system implements it.
3. 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 not usually covered in the User Requirement document. And sometimes, of necessity, the second level too is postponed for later elaboration and negotiation.
2. **Qualification of 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).
3. Describe the consequences of losses of availability and breaches of security, so that the developers can fully appreciate the criticality of each function.
4. Some user requirements may be 'suspended' pending resources becoming available. Such non-applicable user requirements must be clearly flagged.
5. The priority of a requirement measures the order, or the timing, of the related system becoming available. If the delivery is to be phased, so that some parts of the system come into operation before others, then each requirement must be marked with a measure of priority.
6. Some requirements may be dependent on feedback from later stages in the lifecycle, like the system/software requirement phase or technical design phase. These requirements should be marked 'TBC' (to be confirmed), and preferably with an indication of what the confirmation will depend on.

## General

1. IDA Projects are concerned with the implementation of trans‑European telematics networks. The user requirements for such networks may encompass a variety of facilities and services, and may entail the provision of a wide range of technological and other products in support of such facilities and services.
2. For example, in addition to the provision of specified user facilities and functional capability, the IDA Project may require the supply of telecommunications facilities and computing equipment, not to mention a mixture of other potential services such as web-site hosting, system installation, data loading, parallel running and/or phased roll‑out, User Guides and training material, training courses, initial support for users, help desks, system maintenance under warranty, etc.
3. It is accordingly very often expedient, and arguably necessary, to begin the statement of specific requirements with general requirements relating to the scope, constitution and structure of the overall telematics system to be created.

## User facilities

1. Specify all requirements relating to human-computer interactions (user interfaces).
2. Note that, from the user point of view, the user interface is the system.
3. Following the description of user characteristics (See 2.4), specify all specific requirements for

types of user session (in accordance with 2.5)

styles and modes of interaction

user conceptual models of data and processes

user interface hardware, including access mechanisms (smart cards, etc)

interactive performance

on-line help

training.

## System interfaces

1. Constraint requirements that relate to system interfaces may be grouped around the headings:

* communications interfaces;
* hardware interfaces;
* software interfaces.

1. However, although these topics should undoubtedly be covered, it is not obvious that this grouping is necessarily best. For example, there could be some merit in grouping together all specifications for a particular communications channel, including protocols, hardware and software.
2. If the system described in this document is part of a larger system then any documents that describe the interfaces should be identified. These should include documents already written, which state interfaces with which this system must comply and also documents that must be written as part of the project which will provide interfaces which other later projects must comply.

## Communications requirements

1. If a telecommunication capability is to be supplied, it warrants its own detailed requirements section. On the other hand, if (as will normally be the case) pre‑existing telecommunications facilities are to be used, telecommunications requirements are probably better specified elsewhere (e.g. in relation to Interfaces or Performance).

## Hardware requirements

1. If hardware is to be supplied, it warrants its own detailed requirements section.
2. This should specify requirements in a little or as much detail as the users care about the matter. A minimal specification might be concerned just with the general nature, capacity and performance of the equipment to be provided. But the defined requirements might even, for reasons of compatibility or standardisation, go so far as to specify particular makes and models of equipment, if that is what the user community wants.

## Capability requirements

1. This section is centred on functional requirements to be satisfied by software.
2. The capability requirements can be structured around a processing sequence, for example (in an message switch system):

reception of a message

processing of a message

delivery of a message (perhaps followed by deviations from the baseline operation):

handling incorrectly addressed messages.

1. Each capability requirement should be checked to see whether the inclusion of capacity, speed and accuracy attributes is appropriate.

## System management functions

1. On the basis of the defined concept of system operation and maintenance (See 2.8 above), there are generally significant User Requirement for such things as instrumentation, support tools, etc.

* system set-up
* operational monitoring and control
* system instrumentation, accounting, tuning
* mechanisms for back-up and recovery
* tools for system verification and fault diagnosis
* software tools for system maintenance and development
* systems for change management, configuration management, release control
* etc.

## Other products

1. The users may also have explicit requirements for many other things, such as system installation, data loading, parallel running and/or phased roll‑out, User Guides and training material, training courses, initial support for users, system maintenance under warranty, etc. The IDA-MS generic Service Level Agreement (SLA)[[9]](#footnote-9) includes additional topics that could be considered.
2. Any such requirements should be included in the User Requirement document. And if none are known then it is better to say “None identified” rather than to say “None” (because there always are some!) or to omit the section (which might then be forgotten).

## Constraint requirements

1. Constraint requirements may cover any topic that is not included in the preceding sections.
2. Requirements that ensure the system will be fit for its purpose should be stated, for example:

* performance
* adaptability;
* reliability;
* availability;
* capacity and expansibility;
* portability;
* security;
* safety;
* standards
* system construction.
* timetable.

1. Note that in most cases what starts as a “constraint” can lead directly to specific functional requirements, though admittedly “secondary” ones.
2. **Performance**. Provision is made for specific performance requirements to be specified in conjunction with associated functional requirements, but not all performance requirements can necessarily be specified conveniently in that way. (Note that there may also be specific functional requirements for performance monitoring and tuning facilities: see Section 3.7 above.)
3. **Adaptability**. Leaving aside adaptability to different platforms, which comes under “Portability” below, this head essentially covers the ability to change functionality. Clearly functionality can be changed by rewriting all the software, so the “Adaptability” head only makes sense as a specification of a range of functionality which the system should be able to provide (which is merely a generic functional requirement), and perhaps a specification of (functional) requirements for user selection of specific functions within that range (i.e. user modifiable parameters). In any case, the ease of adaptation is something on which the users are likely to have requirements.
4. **Reliability**. Specify tolerable system failure rates (hardware plus software). Identify requirements for the reliability of processes performed and calculations done. Specify limits on how long it is tolerable for different types of fault to remain undetected.[[10]](#footnote-10)
5. **Availability**. When should the system be scheduled to be available to its users (times of day, days of year)? What loss of availability during those times is tolerable? How will the users learn of non-availability? Are fallback facilities needed in the event of non‑availability? Is any special provision needed for bringing the system back into safe, productive operation after a period of non‑availability?
6. Note that availability requirements can lead on to architectural constraints, to requirements for high availability hardware and duplication of communications channels, and to specific functional requirements for availability monitoring and for backup and recovery mechanisms.
7. **Capacity and expansibility**. What volumes of transactions and of data to be stored should the system be designed to accommodate? What levels of growth in these volumes should the system be adaptable to accommodate?
8. **Portability**. Specify the range of platforms on which the system should be able to operate, and the required ease of transfer between them.
9. **Security**. Specify any requirements for the control of access to or dissemination of sensitive information. Identify any specific functional requirements for security administration.
10. **Safety**. Specify any pertinent safety requirements.
11. **Standards**. Specify any predefined standards which are applicable to the IDA Project.
12. **System construction**. Identify requirements concerning system architecture, tools and methods used in system construction and verification.
13. **Timetable**. When is the system needed (perhaps with reference to the cost of not having it)? What is its required durability (i.e. how long should it be capable of continuing in operation)?

Document Control

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

Document Signoff

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

Document Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Change Details** |
|  |  | John Brinkworth, Bill Waghorn | Updates following EC review |
| 22 December 2000 | Issue 1 Draft F | Mark Pillatt | Internal review |
| 10 January 2001 | Issue 1 Draft 7 | Sue Turner | Updated formatting |
| 17 January 2001 | Issue 1 | Mark Pillatt | Apply review comments and issue |

1. Note that the initial version of the Project Definition document corresponds with the “Preparatory Report” identified in the IDA Decisions (Projects of common interest – Article 5 Clause 4): “*IDA projects shall include a preparatory phase, a feasibility phase, a development and validation phase, and an implementation phase. The preparatory phase shall lead to the establishment of a preparatory report comprising the objectives, scope and rationale of the project and in particular the anticipated costs and benefits, as well as the achievement of the necessary commitment and understanding among the participants through appropriate consultation.*” [↑](#footnote-ref-1)
2. Requirements are negotiated in the light of the expected feasibility (including cost) of satisfying them. Hence a preliminary/provisional “system concept” is needed before consultation of users. [↑](#footnote-ref-2)
3. Possible examples are numerous, and too often encountered in practice.

   Watertight security in a developed system can be completely undermined by lack of security in the operating system on which it runs.

   Impeccable performance in a developed system can be impaired by use of inadequate hardware, sharing of computers, inappropriate scheduling of tasks, contention in communications, volume dependencies, and many other factors.

   Superb reliability and availability in a developed system can be rendered irrelevant by very poor mains power supply, or by siting equipment in basements subject to flooding.

   Even the apparently perfect functional correctness of a developed system can be spoiled by the behaviour of the other systems with which it operates, such as the collapse of the infrastructure on defective inputs, or the interception or mutilation of some data sequences by the communications system. [↑](#footnote-ref-3)
4. I.e. the inclusion of mechanisms to obtain, display and collect measures of system behaviour and performance. [↑](#footnote-ref-4)
5. E.g. the apportionment of system usage costs between different user groups. [↑](#footnote-ref-5)
6. IDA Architecture Guidelines v4.1(01 March 1999). An update is planned for early 2001. [↑](#footnote-ref-6)
7. A list of sources should therefore appear either in an Appendix or in a referenced document. [↑](#footnote-ref-7)
8. As discussed in 2.2 above. [↑](#footnote-ref-8)
9. Ref: IDA-MS-SLA [↑](#footnote-ref-9)
10. Any such limits are likely to generate specific system requirements for additional fault detection and fault prevention provisions in the system. [↑](#footnote-ref-10)