

UR.1 Requirements Gathering & Definition

1 D O M A UR.1.1 Define a standard documentation structure, organization and presentation

1 D O M A UR.1.1.1

The software organization manager (SOM) should have a standard that sets a common structure, organization and presentation for preparing a requirements document (RD).

UR.A provides a structure for the RD and guidelines for arriving at the best structure for the organization.

1 D O M A UR.1.2 Gather request information

1 D O M A UR.1.2.1

The software organization (normally software analysts, development managers, project managers, operations managers or organization manager) receives requests for new development or changes to software from software users, customers and managers at various points in the software life cycle (SLC) by means of direct communication (telephone, electronic mail) or via pre-arranged review meetings with customer and users.

For projects that adopt a Rapid Application Development (RAD) Life Cycle, refer to RAD Guide for details on gathering request information available in ITSD.

1 D O M A UR.1.2.2

The software analyst (SA) should use the requirements document (RD) as the basis to gather request information and define the requirements.

Refer to UR.A for an example of the structure, organization and presentation of the RD

1 D O M A UR.1.3 Evaluate system feasibility

1 D O M A UR.1.3.1

With the assistance of the requestor, the SA should translate the request information into requirements for a new/updated system.

2 D O M A UR.1.3.2

The SA should then carry out a feasibility study to include:

- a) An assessment of whether or not the new/updated system can be implemented and/or integrated given existing technology and working methods.
- b) An evaluation that the new/updated system should

provide adequate returns on investment in development.

2 | D O M A | UR.1.4 Gather & define requirements

2 | D O M A | UR.1.4.1

The SA should ensure accuracy and completeness in gathering requirements.

2 | D O M A | UR.1.4.2

In addition, the SA should obtain a better understanding of the requirements by defining them.

UR.B provides a checklist to follow in gathering and defining requirements. Refer also SSADM V.4.2 Practitioner Manual Vol.1 & 3 (Structured Standards) for detailed guidelines and techniques, available in ITSD.

2 | D O M L | UR.1.5 Prioritize and classify requirements

2 | D O M L | UR.1.5.1

With input from stakeholders, the SA should prioritize and classify the requirements.

UR.H provides guidelines on how to prioritize and classify requirements.

UR.2 Requirements Analysis & Negotiation

1 | D O M A | UR.2.1 Analyze requirements

1 | D O M A | UR.2.1.1

Having gathered and defined an initial set of requirements, the SA should then commence analysis work by defining the boundaries of the computer-based system.

2 | D O M L | UR.2.1.2

The SA should be developing and using checklists/questionnaires of requirement problems as a tool to assess each requirement.

UR.E and UR.F provide guidelines on how the boundaries of a computer-based system can be defined and how the checklist can be developed and used respectively.

2 | D O M A | UR.2.2 Negotiate to agree on a set of system requirements

2 D O M A UR.2.2.1

The SA should negotiate on system requirements by:

- a) planning and holding meetings to discuss the requirements and resolve problems e.g. conflicts, overlaps, omissions etc., uncovered during the analysis with stakeholders e.g. requirements initiators, end-users, customers, software managers etc.
- b) using the negotiation meetings to confirm the classification and prioritization of requirements described in section UR.1.5 above.

3 D O M A UR.2.2.2

The SA should negotiate on system requirements by using interaction matrices to find conflicts and overlaps and also as input to the negotiation meeting.

UR.I provides guidelines on the use of interaction matrices.

2 D O M L UR.2.2.3

The SA should negotiate on system requirements by planning on using electronic systems e.g. electronic mail, bulletin boards for information exchange during negotiation discussion.

UR.G provides a discussion of the technologies that can be used for electronic negotiations.

2 D O M A UR.2.3 Assess requirement risks**2 D O M A** UR.2.3.1

For each requirement or set of requirements, the SA should carry out a risk analysis which:

- a) Suggest possible problems which may arise in the implementation of the requirement;
- b) The probability of these problems arising;
- c) The effects which these problems will have;
- d) The recovery actions that could be taken.

UR.J provides guidelines on how to perform risk analysis for requirements.

UR.3 System Modeling**2 D O M A** UR.3.1 Develop system models**2 D O M A** UR.3.1.1

Having negotiated and agreed on a set of system

requirements, the SA should convert these requirements to system models to better understand and illustrate the system requirements namely:

- a) *Complementary* system models (refer Appendix 6.11) to illustrate different aspects of the system specifications;
- b) *Environmental* system model (refer Appendix 6.12) to show other automated systems which are interfaced to it and business processes which may use the system;
- c) *Architectural* system model (refer Appendix 6.13) to show how the system is decomposed into sub-systems.

2 | D O M A | UR.3.1.2

In developing the system models, the SA should use systematic approaches to cover notations, guidelines and rules to define system models.

3 | D O M A | UR.3.1.3

The SA should use structured methods to cover

- a) process to develop and validate system models;
- b) usage of Data Dictionary to document all names used in the system model;
- c) links between requirement descriptions to the system models.

UR.N provides guidelines on how to apply Structured Methods and the use of the Data Dictionary in System modeling

2 | D O M A | UR.3.1.4

The SA should ensure that the system models are included as part of the requirements documentation.

UR.4 Requirements Validation

1 | D O M A | UR.4.1 Validate requirements

1 | D O M A | UR.4.1.1

After the systems requirements have been modeled and the requirements document (RD) updated, the SA should circulate the RD for review by stakeholders.

1 D O M A UR.4.1.2

The SA should validate the requirements for any omissions, conflicts and ambiguities by checking that the requirements meet organizational standards.

Please refer to UR.O .

2 D O M A UR.4.1.3

The SA should organize formal requirements inspection meetings to systematically check the requirements, discuss problems and how to fix them.

Please refer to UR.P.

2 D O M L UR.4.1.4

The SA should:

- a) Review requirements by a multi-disciplinary team consisting of people with different backgrounds e.g. a combination of end-user, customer and software designers;
- b) Paraphrase system models to enable end-users, senior management and regulators to understand and comment on system specification.

Please refer to UR.T.

3 D O M A UR.4.1.5

The SA should:

- a) Define and use checklists/questionnaires to validate critical attributes of requirements document;
- b) Prototype system to demonstrate, validate and improve requirements;
- c) Draft user manual to validate system functionality and interfaces;
- d) Develop requirements *test cases* to validate completeness and ambiguity.

Please refer to UR.Q

Please refer to UR.R.

Please refer to UR.S

1 D O M A UR.4.2 Update requirements document (RD)**1 D O M A** UR.4.2.1

The SA should ensure that the RD is kept updated after all validation activities have been completed for use by all stakeholders including software developers.

UR.5 Requirements Management

1 | D O M A | UR.5.1 Define policies, procedures and standards for requirements management

1 | D O M A | UR.5.1.1

After the requirements have been validated and the requirements document (RD) kept up to date, software developers e.g. project manager, system analyst etc. should continue with development activities to create the new or updated system referencing the RD.

1 | D O M A | UR.5.1.2

The software organization (normally via the software organization manager, project manager, operations manager or systems analyst), should be alerted to the emergence of new requirements during the software life cycle as a result of:

- a) misunderstandings during the requirements gathering, definition and validation stages;
- b) software design, development, implementation, operations and maintenance, problems and changes;
- c) stakeholders acquire a deeper understanding of the system;
- d) changes to business strategy and priorities.

1 | D O M A | UR.5.1.3

The software organization should have in place policies, procedures and standards for requirements management to ensure that:

- a) the quality of the requirements is maintained;
- b) the requirements are kept in step with the development and operational system i.e. systems changes must be reflected as requirement changes and vice versa.

UR.U and UR.V provide guidelines for defining policies on requirements management and the use of a database for storing requirements information.

2 | D O M A | UR.5.2 Define traceability policies

2 | D O M A | UR.5.2.1

The software organization should, as a policy, define what traceability information should be maintained and how these should be represented.

2 D O M A UR.5.2.2

The software organization should include traceability information that allows it to find dependencies between requirements and between the requirements and the system design, components and documentation.

UR.W provides guidelines on how to write traceability policies.

2 D O M A UR.5.3 Maintain a traceability manual

2 D O M A UR.5.3.1

The software organization should maintain a traceability manual for each and every project which keeps a central record of all the applicable project traceability policies and information.

UR.X provides guidelines on how to maintain a traceability manual.

2 D O M A UR.5.4 Define change management policies

2 D O M A UR.5.4.1

The software organization should define policies which set out how:

- a) changes are formally proposed, analyzed and reviewed;
- b) accepted changes are implemented to create a new version of the requirements document.

Refer also to Section CC.3 of Change Control practice of how changes are proposed, analyzed and implemented.

3 D O M L UR.5.5 Maintaining other requirements

3 D O M L UR.5.5.1

The software organization should also consider maintaining other related requirements including:

- a) Identifying and documenting global system requirements which sets out desirable or essential properties of the system as a whole;
- b) Maintaining a list of requirements that are most likely to change;
- c) Keeping a record of requirements which have been proposed and subsequently rejected after analysis or

Please refer to UR.Y.

Please refer to UR.Z.

negotiation.