Software Engineering

هندسة البرمجيات

CIS 240

Mobile: 0796484613
E-mail: osa_alkhoun82@yahoo.com
تنبيه هام

تم ترجمة هذا المساق لمساعدة الطالب على فهم المواضيع المتعلقة به.

وهذا لا يعني أنك غير بحاجة لفهمها أيضاً باللغة الإنجليزية مع تمنياتي للجميع بالتوافق.
Chapter Four
الفصل الرابع

Software Requirements, Requirements Engineering Processes and Activities
متطلبات البرمجيات، متطلبات هندسة العمليات (المعالجة) والنشاطات
Software Requirements, Requirements Engineering Processes and Activities

(Chapters 6 and 7 from the textbook)

الفصل السادس والسابع من الكتاب

Objectives

- To introduce the concepts of requirements
- To describe functional and non-functional requirements
- To introduce the methods of writing requirements; Native Language (NL), Structured Language, and Formal language.
- To explain how software requirements may be organised in a requirements document

Topics covered

- Requirement sub stages or tasks
- Functional and non-functional requirements
- Interface specification
- The software requirements document
Requirements engineering

- The process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed.

- The requirements themselves are the descriptions of the system services and constraints that are generated during the requirements engineering process.
What is a requirement?

- It may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification.
- This is inevitable as requirements may serve a dual function:
  - May be the basis for a bid for a contract - therefore must be open to interpretation;
  - May be the basis for the contract itself - therefore must be defined in detail;
- Both these statements may be called requirements.

ما هي المطلبات؟

- قد يتراوح من البيان التجريدي (الجمل المجردة من التفاصيل) العالي المستوى للخدمة أو قيد النظام إلى تفاصيل المواصفات الوظيفية الرياضية.
- هذه حتمية كالمطلبات التي قد تخدم وظيفة ثانية:
  - ربما الأساس يعرض العقد - لذا يجب أن تكون مفتوحة إلى التفسير؟
  - ربما الأساس يكون للعقد نفسه - لذا يجب أن تعرف بالتفصيل
  - كلاً من هذه البيانات يمكن أن تدعى المطلبات
Functional and non-functional requirements

المتطلبات الوظيفية وغير وظيفية

• Functional requirements

- Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.

بيانات الخدمات التي يجب على النظام تزويدها، كيف يجب على النظام الرد على المدخلات المعينة وكيف يجب على النظام أن يصرف في الحالات التي تحدث بشكل خاص.

• Non-functional requirements

- Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.

القيود على الخدمات أو الوظائف تم عرضها من خلال النظام مثل توقيت القيود، قيود على عملية التطوير، المعايير، الخ.

• Domain requirements

- Requirements that come from the application domain (such as the operating system) of the system and that reflect characteristics of that domain.

المتطلبات التي تأتي من مجال التطبيق (مثل نظام التشغيل) النظام والتي تعكس خصائص ذلك المجال.
Functional requirements

- Describe functionality or system services. (ex, the software should accept numbers from 0 to 100).

- Depend on the type of software, expected users and the type of system where the software is used.

- Functional user requirements may be high-level statements of what the system should do but functional system requirements should describe the system services in detail.

The LIBSYS system- Example

- A library system that provides a single interface to a number of databases of articles in different libraries. (Usually used for university library systems).

- Users can search for, download and print these articles for personal study.
Examples of functional requirements

- The user shall be able to search either all of the initial set of databases or select a subset from it.

- The system shall provide appropriate viewers for the user to read documents in the document store.

- Every order shall be allocated a unique identifier (ORDER_ID) which the user shall be able to copy to the account’s permanent storage area.
Requirements imprecision or ambiguity

- Problems arise when requirements are not precisely stated. (Ex, the software should accept too many users, considered ambiguous).
- Ambiguous requirements may be interpreted in different ways by developers and users. (I believe 50 is too many, while somebody else think that 50 is too few).
- Consider the term ‘appropriate viewers’
  - User intention - special purpose viewer for each different document type;
  - Developer interpretation - Provide a text viewer that shows the contents of the document.
Requirements completeness and consistency

- In principle, requirements should be both complete and consistent.

- Complete
  - They should include descriptions of all facilities required. If some spec is not specified, it will open the door for assumptions.

- Consistent
  - There should be no conflicts or contradictions in the descriptions of the system facilities. Ex, The software should work only under Windows, and the software should be environment independent.

- In practice, it is impossible to produce a complete and consistent requirements document – that may answer every single question.
Non-functional requirements
المتطلبات غير الوظيفية

- These define system properties and constraints e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.

- Process requirements may also be specified mandating a particular CASE system, programming language or development method.

- Non-functional requirements may be more critical than functional requirements. If these are not met, the system is useless. Ex, The software response time should not exceed 2 seconds. (If we develop the best software with response time of 10 seconds, then that maybe considered as a complete failure).
Non-functional classifications

• Product requirements
  • Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.
  • Requirements which define the product that should be delivered e.g. an exchange that should be made on a particular format, quality, etc.

• Organisational requirements
  • Requirements which are a consequence of organisational policies and procedures e.g. process standards used, implementation requirements, etc. Ex, The software should run under MS SQL 2005 server.

• External requirements
  • Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc. Ex, the software should meet ISO 2008 quality standards.
Non-functional requirement types

 وأنواع المتطلبات غير الوظيفية
Non-functional requirements examples

- Product requirement

8.1 The user interface for LIBSYS shall be implemented as simple HTML without frames or Java applets.

- Organisational requirement

9.3.2 The system development process and deliverable documents shall conform to the process and deliverables defined in XYZCo-SP-STAN-95.

- External requirement

7.6.5 The system shall not disclose any personal information about customers apart from their name and reference number to the operators of the system.

Goals and requirements

- Non-functional requirements may be very difficult to state precisely and imprecise requirements may be difficult to verify.

Goal: A general intention of the user such as ease of use.
- Verifiable non-functional requirement
  - المطلوب غير الوظيفي القابل للإثبات (التحقق منه)
- A statement using some measure that can be objectively tested. For example, instead of saying – the software should be reliable, we say “the software MTBF – mean time between failures, should not be less than 1 year”.
  - i.e. something that can be measured [ or isn't it ?!].
- Goals are helpful to developers as they convey the intentions of the system users. The goal is the reason for having a certain requirement or functionality.
  - الأهداف مساعدة إلى المطّورين بينما يحلمون غرض مستخدمي النظام. إن الهدف السبب لامتلاك المتطلّب الأكيد أو الوظيفة الأكيدة.
Examples

• A system goal
  - The system should be easy to use by experienced controllers and should be organised in such a way that user errors are minimised.

• A verifiable non-functional requirement
  - Experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.
## Requirements measures

<table>
<thead>
<tr>
<th><strong>Property</strong></th>
<th><strong>Measure</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Speed</strong></td>
<td>Processed transactions/second</td>
</tr>
<tr>
<td></td>
<td>User/Event response time</td>
</tr>
<tr>
<td></td>
<td>Screen refresh time</td>
</tr>
<tr>
<td><strong>Size</strong></td>
<td>M Bytes</td>
</tr>
<tr>
<td></td>
<td>Number of ROM chips</td>
</tr>
<tr>
<td><strong>Ease of use</strong></td>
<td>Training time</td>
</tr>
<tr>
<td></td>
<td>Number of help frames</td>
</tr>
<tr>
<td><strong>Reliability</strong></td>
<td>Mean time to failure</td>
</tr>
<tr>
<td></td>
<td>Probability of unavailability</td>
</tr>
<tr>
<td></td>
<td>Rate of failure occurrence</td>
</tr>
<tr>
<td><strong>Robustness</strong></td>
<td>Time to restart after failure</td>
</tr>
<tr>
<td></td>
<td>Percentage of events causing failure</td>
</tr>
<tr>
<td></td>
<td>Probability of data corruption on failure</td>
</tr>
<tr>
<td>Portability</td>
<td>Percentage of target dependent statements</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>قابلية النقل</td>
<td>النسبة المئوية من بيانات الهدف التابع</td>
</tr>
<tr>
<td>اﻟﻨﺴﺒﺔ ﻣﻨﺒﺎء ﻣﻦ ﺑﻴﺎﻧﺎت ﺍﻟﻬﺪف ﺛﺎﺑﻊ</td>
<td>Number of target systems</td>
</tr>
<tr>
<td>ﺛﺪف</td>
<td>عدد أنظمة الهدف</td>
</tr>
<tr>
<td>أﻧﻈﻤﺔ</td>
<td>19</td>
</tr>
</tbody>
</table>
Requirements interaction

- Conflicts between different non-functional requirements are common in complex systems.
- Spacecraft system
  - To minimise weight, the number of separate chips in the system should be minimised.
  - To minimise power consumption, lower power chips should be used.
  - However, using low power chips may mean that more chips have to be used.
  - Which is the most critical requirement? i.e. Requirements prioritization.

Domain requirements

- Derived from the application domain, by domain or business experts, and describe system characteristics and features that reflect the domain.
- Domain requirements can be new functional requirements, constraints on existing requirements or define specific computations.
• If domain requirements are not satisfied, the system may be unworkable.

• Because of copyright restrictions, some documents must be deleted immediately on arrival. Depending on the user’s requirements, these documents will either be printed locally on the system server for manually forwarding to the user or routed to a network printer.

• There shall be a standard user interface to all databases which shall be based on the Z39.50 standard.

Library system domain requirements

• If domain requirements are not satisfied, the system may be unworkable.

• Because of copyright restrictions, some documents must be deleted immediately on arrival. Depending on the user’s requirements, these documents will either be printed locally on the system server for manually forwarding to the user or routed to a network printer.

• There shall be a standard user interface to all databases which shall be based on the Z39.50 standard.

• If domain requirements are not satisfied, the system may be unworkable.

• Because of copyright restrictions, some documents must be deleted immediately on arrival. Depending on the user’s requirements, these documents will either be printed locally on the system server for manually forwarding to the user or routed to a network printer.
Domain requirements problems

• Understandability

- Requirements are expressed in the language of the application domain i.e. domain experts;
- This is often not understood by software engineers developing the system. Ex, The software should optimize the nutrition formula to the least solution cost while observing all nutrients constraints.

• Implicitness

- Domain specialists understand the area so well that they do not think of making the domain requirements explicit.
Problems with natural language

- Lack of clarity
  - الافتقار (القلة) إلى الوضوح
  - Precision is difficult without making the document difficult to read.
  - الدقة صعبة بدون جعل الوثيقة صعبة القراءة
- Requirements confusion
  - التشويش على المتطلبات
  - Functional and non-functional requirements tend to be mixed-up.
  - المتطلبات الوظيفية وغير الوظيفية تميل إلى أن تكون مخلوطة (مشتركة)
- Requirements amalgamation [ combination]
  - دمج المتطلبات [الجمع، الاتحاد، الدمج]
  - Several different requirements may be expressed together.
  - عدة متطلبات مختلفة قد ترمز إليه سوياً
Problems with NL specification

- Ambiguity
  - The readers and writers of the requirement must interpret the same words in the same way. NL is naturally ambiguous so this is very difficult. Example of ambiguous words: Good, complete, perfect, full, all, whole, thing, etc. Those words does not give a quantitative value of the property.

- Over-flexibility [too much flexible]
  - The same thing may be said in a number of different ways in the specification. Usually you should give one or two options only.

- Lack of modularisation
  - NL structures are inadequate to structure system requirements [ in modules format].
LIBSYS requirement

4.5 LIBSYS shall provide a financial accounting system that maintains records of all payments made by users of the system. System managers may configure this system so that regular users may receive discounted rates.

Editor grid requirement

2.6 Grid facilities To assist in the positioning of entities on a diagram, the user may turn on a grid in either centimetres or inches, via an option on the control panel. Initially, the grid is off. The grid may be turned on and off at any time during an editing session and can be toggled between inches and centimetres at any time. A grid option will be provided on the reduce-to-fit view but the number of grid lines shown will be reduced to avoid filling the smaller diagram with grid lines.

Requirement problems

- Database requirements includes both conceptual and detailed information
- Describes the concept of a financial accounting system that is to be included in LIBSYS;
• However, it also includes the detail that managers can configure this system - this is unnecessary at this level.

• Grid requirement mixes three different kinds of requirement
  يخلط متطلبات الشبكة ثلاثة أنواع مختلفة من المتطلبات
• Conceptual functional requirement (the need for a grid);
  مفاهيم المتطلب الوظيفي (الحاجة للشبكة);
• Non-functional requirement (grid units);
  المتطلب غير الوظيفي (وحدات الشبكة);
• Non-functional UI requirement (grid switching).
  المتطلب غير الوظيفي ومعطيات التبديل (تحويل الشبكة).

Structured presentation

2.6.1 Grid facilities

The editor shall provide a grid facility where a matrix of horizontal and vertical lines provide a background to the editor window. This grid shall be a passive grid where the alignment of entities is the user's responsibility.

المحرر سيود وسيلة الشبكة حيث أن مصفوفة الخطوط الأفقية والعمودية تزود خلفية إلى نافذة المحرر. هذه الشبكة ستكون شبكة قائمة حيث أن محاذاة الكيانات تكون من مسؤولية المستخدم.

Rationale: A grid helps the user to create a tidy diagram with well-spaced entities. Although an active grid, where entities 'snap-to' grid lines can be useful, the positioning is imprecise. The user is the best person to decide where entities should be positioned.

السبب الجوهري: شبكة تساعد المستخدم إنشاء ترتيب تخطيطي بالكيانات المباعد بشكل جيد. بالرغم من أن الشبكة نشطة، عندما الكيانات 'تتقاطع' خطوط الشبكة
Guidelines for writing requirements

- Invent – or use- a standard format and use it for all requirements [see requirement templates such as IEEE template].
- Use language in a consistent way. Use shall for mandatory requirements, should for desirable requirements.
- Use text highlighting to identify key parts of the requirement.
- Avoid the use of computer jargon or terminology.

System requirements

- More detailed specifications of system functions, services and constraints than user requirements.
- They are intended to be a basis for designing the system.
- They may be incorporated into the system contract.
Requirements and design

- In principle, requirements should state what the system should do and the design should describe how it does this.
- In practice, requirements and design are inseparable.
- A system architecture may be designed to structure the requirements;
- The system may inter-operate with other systems that generate design requirements;
- The use of a specific design may be a domain requirement.

**استخدام تصميم معين قد يكون متطلب المجال**
### Alternatives to NL specification

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured natural language</td>
<td>This approach depends on defining standard forms or templates to express the requirements specification.</td>
</tr>
<tr>
<td>Design description languages</td>
<td>This approach uses a language like a programming language but with more abstract features to specify the requirements by defining an operational model of the system. This approach is not now widely used although it can be useful for interface specifications.</td>
</tr>
<tr>
<td>Graphical notations</td>
<td>A graphical language, supplemented by text annotations is used to define the functional requirements for the system. An early example of such a graphical language was SADT(System Analysis Design Technic). Now, use-case descriptions and sequence diagrams are commonly used. The language, designed for the system, is intended to be interpreted in the technical and procedural systems. An example of this language, used in the technical and procedural systems, is a technical system designed for the technical and procedural systems.</td>
</tr>
</tbody>
</table>
These are notations based on mathematical concepts such as finite-state machines or sets. These unambiguous specifications reduce the arguments between customer and contractor about system functionality. However, most customers don’t understand formal specifications and are reluctant to accept it as a system contract.
Structured language specifications

- The freedom of the requirements writer is limited by a predefined template for requirements. [But it acts as a common ground for understanding].
- All requirements are written in a standard way.
- The terminology used in the description may be limited.
- The advantage is that the most of the expressiveness of natural language is maintained but a degree of uniformity is imposed on the specification.

مواصفات اللغة المنظمة

- إن حرية كاتب المتطلبات تحدد من قبل طبعة معرفة للمتطلبات. [لكن يحدث أثرا أساسيا يكون مشترك للفهم].
- كل المتطلبات مكتوبة على نحو قياسي.
- المصطلح المستخدم في الوصف الذي يحدد.
- الفائدة في الأغلب هي تعبير اللغة الطبيعية التي تحافظ على درجة الاتفاق المفروضة على المواصفات.
Form-based specifications

- Definition of the function or entity.
- Description of inputs and where they come from.
- Description of outputs and where they go to.
- Indication of other entities required.
- Pre and post conditions (if appropriate). Each event needs conditions in order to occur [i.e. pre] and has results [i.e. post].
- The side effects (if any) of the function.
Form-based node specification

Insulin Pump/Control Software/SRS/3.3.2

Function: Compute insulin dose: Safe sugar level
**Description**  Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units.

Inputs  Current sugar reading (r2), the previous two readings (r0 and r1)

Source  Current sugar reading from sensor. Other readings from memory.

Outputs  Comp Dose – the dose in insulin to be delivered

Action: CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered.

Requires  Two previous readings so that the rate of change of sugar level can be computed.

Pre-condition  The insulin reservoir contains at least the maximum allowed single dose of insulin.

Post-condition  r0 is replaced by r1 then r1 is replaced by r2

Side-effects  None
Tabular specification [In Table-like format]

<table>
<thead>
<tr>
<th>Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sugar level falling ((r_2 &lt; r_1))</td>
<td>(\text{CompDose} = 0)</td>
</tr>
<tr>
<td>Sugar level stable ((r_2 = r_1))</td>
<td>(\text{CompDose} = 0)</td>
</tr>
<tr>
<td>Sugar level increasing and rate of increase decreasing ((\Delta r_2 &lt; \Delta r_1))</td>
<td>(\text{CompDose} = 0)</td>
</tr>
<tr>
<td>Sugar level increasing and rate of increase stable or increasing. ((\Delta r_2 \geq \Delta r_1) ) ((r_2-r_1) \geq (r_1-r_0))</td>
<td>(\text{CompDose} = \text{round} \left(\frac{(r_2-r_1)}{4}\right)) If rounded result = 0 then (\text{CompDose} = \text{MinimumDose})</td>
</tr>
</tbody>
</table>

Graphical models

- **Graphical models** are most useful when you need to show how state changes or where you need to describe a sequence of actions.
- The most used graphical models is **UML (Unified Modeling Language)** diagrams.

"Graphical models" are particularly useful when you have to define a number of possible alternative courses of action.
Sequence diagrams

- These show the sequence of events that take place during some user interaction with a system.
- You read them from top to bottom to see the order of the actions that take place.
- Cash withdrawal from an ATM
  - Validate card;
  - Handle request;
  - Complete transaction.

Sequence diagram of ATM withdrawal

- انسحب النقد من آلات سحب النقود
- تفتيش البطاقة؛
- طلب المقبض؛
- اكتمال الحركة.

- اتخاذهم تتم من الأعلى إلى الأسفل لرؤية طلب الأعمال (الأحداث) التي تحدث.
- قراءتهن تتم من الأسفل إلى الأعلى لرؤية طلب الأعمال (الأحداث) التي تحدث.

- تفتيش الأحداث التي تحدث أثناء تفاعل بعض المستخدمين مع النظام.

- يبين (ترى من خلالها) سلسلة الأحداث التي تحدث أثناء تفاعل بعض المستخدمين مع النظام.
Most systems must operate with other systems and the operating interfaces must be specified as part of the requirements.

Three types of interface may have to be defined:

- Procedural interfaces;
- Data structures that are exchanged;
- The operating interfaces of the system and other operating interfaces;
• Data representations.
  • تمثيل البيانات.
• Formal notations are an effective technique for interface specification.
  • الترميم الرسمي تقنية فعالة لمواصفات الواجهة.
• We may use prototypes for GUI specs.
  • يمكن استخدام النموذج الأولي لمواصفات واجهة المستخدم الرسومية.

PDL (Programming Description Language) interface description

```java
interface PrintServer {
  // defines an abstract printer server
  // requires: interface Printer, interface PrintDoc
  // provides: initialize, print, displayPrintQueue, cancelPrintJob, switchPrinter
  void initialize ( Printer p ) ;
  void print ( Printer p, PrintDoc d ) ;
  void displayPrintQueue ( Printer p ) ;
  void cancelPrintJob ( Printer p, PrintDoc d ) ;
  void switchPrinter ( Printer p1, Printer p2, PrintDoc d ) ;
}
```

The requirements document

• The requirements document is the official statement of what is required of the system developers.
  • إن وثيقة المتطلبات هي البيان الرسمي المطلوب من مطور النظام.
• Should include domain, user, and system requirements. It should also include functional and non functional requirements.
  • يجب أن يتضمن المجال المستخدم، ومتطلبات النظام. يجب أن يتضمن متطلبات وظيفية وغير الوظيفية أيضاً.
• It is NOT a design document. As far as possible, it should set of WHAT the system should do rather than HOW it should do it.
  • إنها ليست وثيقة تصميم. من الممكن أقصى ما هو أن يحدد ما يجب على النظام أن يفعله بدلاً من كيفية القيام به.
هو ليس وثيقة تصميم. بقدر الإمكان، يجب وضع ماذا يعمل النظام بدلاً من كيف يجب على النظام أن يعمل.
Users of a requirements document

IEEE requirements standard

- Defines a generic structure for a requirements document that must be instantiated for each specific system.

  يعرف الترميح العام لتوثيق المتطلبات لذلك يجب أن يكون جاهز لكل نظام معين.

- Introduction.

  المقدمة

- General description.

  الوصف العام

- Specific requirements.

  تخصيص المتطلبات

- Appendices.
• Index.
Requirements document structure

- Preface
- Introduction
- Glossary
- User requirements definition
- System architecture
- System requirements specification
- System models
- System evolution
- Appendices
- Index
Requirements Engineering Processes and Activities (Chapter 7 from the textbook)

Objectives

- To describe the principal requirements engineering activities and their relationships
- To introduce techniques for requirements elicitation and analysis
- To describe requirements validation and the role of requirements reviews
- To discuss the role of requirements management in support of other requirements engineering processes

Topics covered

- Feasibility studies
- Requirements elicitation and analysis
- Requirements validation
- Requirements management

Requirements engineering processes
The processes used for RE vary widely depending on the application domain, the people involved, and the organisation developing the requirements.

However, there are a number of generic activities common to all processes:

• Requirements elicitation, inception, elaboration, and negotiation;
• Requirements analysis or specification;
• Requirements validation or verification;
• Requirements management.

The requirements engineering activities
Feasibility studies

- A feasibility study decides whether or not the proposed system is worthwhile or if the project is doable given the current resources and constraints.

- A short focused study that checks
  - If the system contributes to organisational objectives;
  - If the system can be engineered using current technology and within budget – project triple constraints.
  - If the system can be integrated with other systems that are used.
This is all based on information assessment (what is required), information collection and report writing.

Feasibility study implementation

- Questions for people in the organisation
  - What if the system wasn’t implemented?
  - What are current process problems?
  - How will the proposed system help?
  - What will be the integration problems?
  - Is new technology needed? What skills?
  - What facilities must be supported by the proposed system?

Elicitation and analysis

- Sometimes called requirements elicitation or requirements discovery. Some requirements maybe embedded in the domain users’ minds and/or in the domain environment. So it needs to be discovered.

- أحيانًا تُدعى استنباط المتطلبات أو اكتشاف المتطلبات. بعض المتطلبات تضمنت ربما في عقول مستخدمي المجال أو في بيئته المجال. لذا من الضروري أن نكتشف.
It involves technical staff working with customers to find out about the application domain, the services that the system should provide and the system’s operational constraints.

It may involve end-users, managers, engineers involved in maintenance, domain experts, trade unions, etc. These are called stakeholders.

Inception or discovery and Elicitation—

• Interacting with stakeholders to discover their requirements. Domain requirements are also discovered at this stage.

• ask a set of questions that establish …

  • basic understanding of the problem
  • the people who want a solution, their problems that needs to be solved.
  • the nature of the solution that is desired.
  • Preliminary communication and collaboration between the customers and the developers at this stage is very important and effective.

• التفاعل مع المالكين لاكتشاف متطلباتهم. متطلبات المجال تكتشف أيضا في هذه المرحلة.

• الاستبلاو أو الاكتشاف والاستبناط

• استماع أسئلة تأسس مجموعة الأسئلة التي تؤسس ...

• الفهم الأساسي للمشكلة

• الأشخاص الذين يريدون الحل، مشاكلهم التي من الضروري أن تحل.

• طبيعة الحل المطلوبة.

• الانصال التمهيدي والتعاون بين الزبائن والمطورين في هذه المرحلة مهم جداً وفعال.
• Elicit requirements from all stakeholders as requirements are not only related to users.
• Elaboration, classification and organization
• Create an analyze the solution model that identifies data, function and behavioral requirements.
• Group related requirements and organize them into coherent clusters.

Process activities

• Prioritisation and negotiation

• Prioritising requirements and resolving requirements conflicts. Conflicts may arise as different stakeholders have different interests in the project.

• Agree on a deliverable system that is realistic for developers and customers. [A compromised not perfect solution].

• Requirements documentation

• Requirements are documented and input into the next round of the software stage.

• Specification—can be any one (or more) of the following:

- A written document
- A set of models
- A formal mathematical
- A collection of user scenarios (use-cases)
- A prototype
- Validation or verification—a review mechanism that looks for
  - errors in content or interpretation
  - areas where clarification may be required
  - missing information
  - inconsistencies (a major problem when large products or systems are engineered)
- Requirements management

Inception
Identify stakeholders

"Who else do you think I should talk to?"

Recognize multiple points of view

Work toward collaboration

The first questions

Who is behind the request for this work?

Who will use the solution?

What will be the economic benefit of a successful solution?

Is there another source for the solution that you need?

Eliciting Requirements

meetings are conducted and attended by both software engineers and customers

rules for preparation and participation are established

an agenda is suggested

a "facilitator" (can be a customer, a developer, or an outsider) controls the meeting
Eliciting Requirements

• to identify the problem
• propose elements of the solution
• negotiate different approaches, and
• specify a preliminary set of solution requirements

The goal is

• to identify the problem
• to define the problem
• to propose elements of the solution
• to negotiate different approaches, and
• to specify a preliminary set of solution requirements

A "definition mechanism" (can be work sheets, flip charts, or wall stickers or an electronic bulletin board, chat room or virtual forum) is used
Elicitation Work Products

- a statement of need and feasibility.
- a bounded statement of scope for the system or product.
- a list of customers, users, and other stakeholders who participated in requirements elicitation.
- a description of the system’s technical environment.
- a list of requirements (preferably organized by function) and the domain constraints that apply to each.

استنباط منتجات العمل

- بيان الحاجة والعملية.
- إحاطة بيان المجال للنظام أو المنتج.
- قائمة الزبائن المستخدمين، والمالكين الآخرين الذين شاركوا في استنباط المتطلبات.
- وصف بيئة النظام التقنية.
- قائمة المتطلبات (منظمة بأفضلية الوظيفة) وقيود المجال التي تقدم إلى الكل.
Features or Functions

Requirements documents may include terminology in both
the technical and the domain sides. For example, to users
functions or requirements are listed as features. Those two
can be stated in different ways although they are similar in
concept.

Each scenario is described from the point-of-view of an
"actor"—a person or device that interacts with the
software in some way

Each scenario answers the following questions:

- Who is the primary actor, the secondary actor (s)?
  - ما هو الممثل الأساسي، الممثل أو الممثلين الثاني؟
- What are the actor’s goals?
  - إلى ماذا يهدف الممثل؟

Use-Cases

A collection of user scenarios that describe the thread of
usage of a system

Each scenario is described from the point-of-view of an
"actor"—a person or device that interacts with the
software in some way

Each scenario answers the following questions:

- Who is the primary actor, the secondary actor (s)?
  - ما هو الممثل الأساسي، الممثل أو الممثلين الثاني؟
- What are the actor’s goals?
  - إلى ماذا يهدف الممثل؟

- A set of usage scenarios that provide insight into the use of
the system or product under different operating conditions.

- Any prototypes developed to better define requirements.

- Features or Functions

- Requirement documents may include terminology in both
the technical and the domain sides. For example, to users
functions or requirements are listed as features. Those two
can be stated in different ways although they are similar in
concept.

- Use-Cases

A collection of user scenarios that describe the thread of
usage of a system
• What preconditions should exist before the story begins?
  • ماذا يجب أن يحد قبل أن تبدأ القصة؟
• What main tasks or functions are performed by the actor?
  • ما هي المهام أو الوظائف الرئيسية المؤدية من قبل الممثل؟
• What extensions might be considered as the story is described?
  • ما هي الامتدادات التي يمكن أن تعتبرها كالقصة الموصوفة؟
• What variations in the actor’s interaction are possible?
  • ما هي الاختلافات المحتملة في التفاعل الممثل؟
• What system information will the actor acquire, produce, or change?
  • ما هي معلومات النظام التي ستكسب الممثل، المنتج، أو التغيير؟
• Will the actor have to inform the system about changes in the external environment?
  • هل الممثل يجب عليه أن يعلم النظام حول التغييرات في البيئة الخارجية؟
• What information does the actor desire from the system?
  • ما هي المعلومات المماثلة المرغوبة من النظام؟
• Does the actor wish to be informed about unexpected changes?
  • هل أمنية الممثل التي ستعل حول التغيرات غير متوقعة؟

Use cases

• Use-cases are a scenario based technique in the UML which identify the actors in an interaction and which describe the interaction itself.

استخدام حالات塞納里奥 معممة على تقنية في لغة النمذجة الموحدة التي تميز الممثلين في التفاعل والتي تعزز التفاعل بنفسه.
A set of use cases should describe all possible interactions with the system.

Sequence diagrams may be used to add detail to use-cases by showing the sequence of event processing in the system.

Article printing use-case
LIBSYS use cases

Article printing
Print article sequence

سلسلة طباعة
Use-Case Diagram

Building the Analysis Model

(Usually found in the architectural design document)

• Elements of the analysis model
  • Scenario-based elements
  • Functional—processing narratives for software functions or features.

العناصر المعتمدة على السيناريو
الوظيفية - قصص المعالجة للوظائف أو ميزات البرمجيات.
- Use-case—descriptions of the interaction between an “actor” and the system

- Class-based elements

- Implied by scenarios

- Behavioral elements

- State diagram

- Flow-oriented elements

- Data flow diagram

Class Diagram

From the SafeHome system ...

<table>
<thead>
<tr>
<th>Sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>name/id</td>
</tr>
<tr>
<td>location</td>
</tr>
<tr>
<td>characteristics</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>identify()</td>
</tr>
<tr>
<td>enable()</td>
</tr>
<tr>
<td>disable()</td>
</tr>
<tr>
<td>reconfigure()</td>
</tr>
</tbody>
</table>
State Diagram

Analysis Patterns

Pattern name: A descriptor that captures the essence of the pattern.

Pattern name: نام النمط: النص المكتوب الذي يأسر جوهر النمط.

Intent: Describes what the pattern accomplishes or represents

Intent: يصف ما هو النمط المنجز أو الممثل

Motivation: A scenario that illustrates how the pattern can be used to address the problem.

Motivation: نية: يصف كيف يمكن استخدام النمط لمعالجة المشكلة.
Forces and context: A description of external issues (forces) that can affect how the pattern is used and also the external issues that will be resolved when the pattern is applied.

Solution: A description of how the pattern is applied to solve the problem with an emphasis on structural and behavioral issues.

Consequences: Addresses what happens when the pattern is applied and what trade-offs exist during its application.

Design: Discusses how the analysis pattern can be achieved through the use of known design patterns.

Known uses: Examples of uses within actual systems.

Related patterns: One or more analysis patterns that are related to the named pattern because (1) it is commonly used with the named pattern; (2) it is structurally similar to the named pattern; (3) it is a variation of the named pattern.

Related patterns: One or more analysis patterns that are related to the named pattern because (1) it is commonly used with the named pattern; (2) it is structurally similar to the named pattern; (3) it is a variation of the named pattern.
Identify the key stakeholders

• These are the people who will be involved in the negotiation

Determine each of the stakeholders “win conditions”

• Win conditions are not always obvious

Negotiate

• Work toward a set of requirements that lead to “win-win” between the 2 sides of the contract.

Problems of requirements analysis

• Stakeholders don’t know what they really want.

• Stakeholders express requirements in their own terms.

• Different stakeholders may have conflicting requirements.

• Organisational and political factors may influence the system requirements.

• The requirements change during the analysis process. New stakeholders may emerge and the business environment change.

• Organisational and political factors may influence the system requirements.

• The requirements change during the analysis process. New stakeholders may emerge and the business environment change.
Requirements discovery

- The process of gathering information about the proposed and existing systems and distilling or extracting the user and system requirements from this information.

- Sources of information include documentation, system stakeholders and the specifications of similar or earlier systems.

ATM stakeholders

- Bank customers
- Representatives of other banks
- Bank managers
- Counter staff
- Database administrators
- Security managers
- Marketing department
- Hardware and software maintenance engineers
- Banking regulators
المنظمات المصرفية
Viewpoints

- Viewpoints are a way of structuring the requirements to represent the perspectives of different stakeholders. Stakeholders may be classified under different viewpoints.

  - Stakeholders who do not use the system themselves but who influence the requirements. In an ATM, management and security staff are indirect viewpoints.
  - People or other systems that interact directly with the system. In an ATM, the customer’s and the account database are interactor VPs.
  - Stakeholders that are affected by the system. In an ATM, the customer and the account database are interactor VPs.

Types of viewpoint

- Interactor viewpoints
  - Stakeholders who do not use the system themselves but who influence the requirements. In an ATM, management and security staff are indirect viewpoints.
  - People or other systems that interact directly with the system. In an ATM, the customer’s and the account database are interactor VPs.

- Indirect viewpoints

- Domain viewpoints
• Domain characteristics and constraints that influence the requirements. In an ATM, an example would be standards for inter-bank communications.

• خصائص وقيود المجال التي تؤثر على المتطلبات. في آلات سحب النقود، مثال سيكون هناك معايير للاتصالات بين المصرف بالشكل الداخلي.
Viewpoint identification

- Identify viewpoints using
  - Providers and receivers of system services;
  - Systems that interact directly with the system being specified;
  - Regulations and standards;
  - Sources of business and non-functional requirements.
  - Engineers who have to develop and maintain the system;
  - Marketing and other business viewpoints.

LIBSYS viewpoint hierarchy

[Diagram of viewpoint hierarchy]
Interviewing

- In formal or informal interviewing, the RE team puts questions to stakeholders about the system that they use and the system to be developed.
- There are two types of interview:
  - Closed interviews where a pre-defined set of questions are answered.
  - Open interviews where there is no pre-defined agenda and a range of issues are explored with stakeholders.

Interviews in practice

- Normally a mix of closed and open-ended interviewing.
- Interviews are good for getting an overall understanding of what stakeholders do and how they might interact with the system.
- Interviews are not good for understanding domain requirements.

- Requirements engineers cannot understand specific domain terminology;
• Some domain knowledge is so familiar that people find it hard to articulate or think that it isn’t worth articulating.

• Usually, software companies hire domain experts for large projects.

• بعض معرفة المجال مألوف جداً بحيث الأشخاص يجد صعوبة بالتبيين أو يعتقد بأنه ليس جدير بالتبيين.

• عادةً، شركات البرمجيات توظف خبراء مجال للمشاريع الكبيرة.
Effective interviewers

المقابلون الفعالون (النشطون)

• Interviewers should be open-minded, willing to listen to stakeholders and should not have pre-conceived ideas about the requirements.

المقابلون يجب أن يكونوا راغبين منفتحين للاستماع إلى المالكين ولا يجب أن يكون عندهم أفكار محمولة من قبل حول المتطلبات.

• They should prompt the interviewee with a question or a proposal and should not simply expect them to respond to a question such as ‘what do you want’.

يجب أن يحتوى المقابل مع السؤال أو الاقتراح ويجب أن لا يتوقعهم أن يردوا ببساطة على السؤال مثل ‘ما الذي تريده أنت’.

Scenarios

السيناريوهات

• Scenarios are real-life examples of how a system can be used.

السيناريوهات أمثلة واقعية للنظام الذي يمكن أن يستخدم.

• They should include

يجب أن يتضمنوا

• A description of the starting situation;

وصف الحالة (الوضع) الابتدائية.

• A description of the normal flow of events;

وصف التدفق الطبيعي للأحداث.

• A description of what can go wrong;

وصف ما الفشل.

• Information about other concurrent activities;

معلومات حول النشاط المتلاقي الأخر.

• A description of the state when the scenario finishes.

وصف الحالة عندما ينتهي السيناريو.

LIBSYS scenario (1)

سيناريو (1) نظام المكتبة
**Initial assumption:** The user has logged on to the LIBSYS system and has located the journal containing the copy of the article.

**Normal:** The user selects the article to be copied. He or she is then prompted by the system to either provide subscriber information for the journal or to indicate how they will pay for the article. Alternative payment methods are by credit card or by quoting an organisational account number.

The user is then asked to fill in a copyright form that maintains details of the transaction and they then submit this to the LIBSYS system.

**What can go wrong:** The user may fail to fill in the copyright form correctly. In this case, the form should be re-presented to the user for correction. If the resubmitted form is still incorrect then the user’s request for the article is rejected.

The payment may be rejected by the system. The user’s request for the article is rejected.

The article download may fail. Retry until successful or the user terminates the session.

It may not be possible to print the article. If the article is not flagged as ‘print-only’ then it is held in the LIBSYS workspace. Otherwise, the article is deleted and the user’s account credited with the cost of the article.

**Other activities:** Simultaneous downloads of other articles.
Social and organisational factors

- Software systems are used in a social and organisational context. This can influence or even dominate the system requirements.
- Social and organisational factors are not a single viewpoint but are influences on all viewpoints.
- Good analysts must be sensitive to these factors but currently no systematic way to tackle their analysis.
Ethnography

A social scientists spends a considerable time observing and analysing how people actually work.

People do not have to explain or articulate their work.

Social and organisational factors of importance may be observed.

Ethnographic studies have shown that work is usually richer and more complex than suggested by simple system models.

Focused ethnography

Developed in a project studying the air traffic control process.

Combines ethnography with prototyping.

Prototype development results in unanswered questions which focus the ethnographic analysis.

The problem with ethnography is that it studies existing practices which may have some historical basis which is no longer relevant.

The problem with ethnography is that it studies existing practices which may have some historical basis which is no longer relevant.
Ethnography and prototyping

Scope of ethnography

- Requirements that are derived from the way that people actually work rather than the way which process definitions suggest that they ought to work.
  - المتطلبات التي تكون مشتقة من ذلك الطريق الذي يعمل فيه الأشخاص في الحقيقة بدلاً من ذلك الطريق الذي يتم تعريفه من خلال عملية يتم اقتراحها ومن ثم يجب عليهم العمل بها.
- Requirements that are derived from cooperation and awareness of other people’s activities.
  - المتطلبات التي تكون مشتقة من التعاون ووعي نشاطات الأشخاص الآخرين.
- Sometimes, companies send developers for a month or more to the client to see how they do their work.
  - أحياناً، الشركات ترسل المطورين لمدة شهر أو أكثر إلى الزبون لرؤية كيف يعملون.

Requirements validation

- Concerned with demonstrating that the requirements define the system that the customer really wants.
  - معني بتوضيح المتطلبات المعرفة بالنظام الذي يريده الزبون حقاً.
- Requirements error costs- if discovered later, are high so validation is very important
• Fixing a requirements error after delivery may cost up to 100 times the cost of fixing an implementation error.

• The draw back – other than its actual cost is that It may delay the development process and requires extra knowledge [such as formal methods].

• Is each requirement consistent with the overall objective for the system/product?

• Is each requirement consistent with the overall objective for the system/product?

• Have all requirements been specified at the proper or the same level of abstraction? That is, do some requirements provide a level of technical detail that is inappropriate at this stage?

• Is the requirement really necessary or does it represent an add-on feature that may not be essential to the objective of the system?

• Is each requirement bounded and unambiguous?

• Does each requirement have attribution? That is, is a source (generally, a specific individual) noted for each requirement?
- Do any requirements conflict with other requirements?
  هل هناك أي متطلبات تتضارب بالمتطلبات الأخرى؟

- Is each requirement achievable in the technical environment that will house the system or product?
  هل كل متطلب قابل للإنجاز في البيئة التقنية التي سيفعال فيه النظام أو المنتج؟

- Is each requirement testable, once implemented?
  كل متطلب قابل للفحص، يطبق مرة واحدة؟

- Does the requirements model properly reflect the information, function and behavior of the system to be built.
  عمل نموذج المتطلبات يعكس المعلومات بشكل صحيح (دقيق)، وظيفة وسلوك النظام سيتم بناءه.

- Has the requirements model been “partitioned” in a way that exposes progressively more detailed information about the system.
  نموذج المتطلبات "تقسيم" بطريقة ما يعرض بتقدم تدريجي معلومات أكثر تفصيلا حول النظام.

- Have requirements patterns been used to simplify the requirements model. Have all patterns been properly validated? Are all patterns consistent with customer requirements?
  أنماط المتطلبات تستخدم لتسييس نموذج المتطلبات. هل كل الأنماط صدقت بشكل صحيح (دقيق)؟ هل كل الأنماط مناسبة (متساقطة) مع متطلبات الزبون؟

Requirements checking

- Validity. Does the system provide the functions which best support the customer’s needs?
  الصلاحية (التصديق). هل النظام يزود الوظائف الأفضل التي تدعم حاجات الزبائن؟

- Consistency. Are there any requirements conflicts?
  الاتساق (التناسق). هل هناك أي متطلبات متضاربة؟

- Completeness. Are all functions required by the customer included?
  التدقيق المتطلبات
• Realism or Feasibility. Can the requirements be implemented given available budget and technology?
• Verifiability. Can the requirements be checked?

Requirements validation techniques

• Requirements reviews

  • Systematic manual analysis of the requirements.
  • Prototyping

  • Using an executable model of the system to check requirements. Example, using Visual basic to design the GUI of the application.
  • Test-case generation

• Developing tests for requirements to check testability. [White box testing].

Requirements reviews

• Regular reviews should be held while the requirements definition is being formulated.
Both client and contractor staff should be involved in reviews.

Reviews may be formal (with completed documents) or informal. Good communications between developers, customers and users can resolve problems at an early stage.

Review checks

- Verifiability. Is the requirement realistically testable?
- Comprehensibility. Is the requirement properly understood?
- Traceability. Is the origin of the requirement clearly stated?
- Adaptability. Can the requirement be changed without a large impact on other requirements?

Requirements management

- Requirements management is the process of managing changing requirements during the requirements engineering process and system development.
- Requirements are inevitably incomplete and inconsistent.
• New requirements emerge during the process as business needs change and a better understanding of the system is developed;

• تظهر المتطلبات الجديدة أثناء المعالجة بينما يحتاج العمل تغييرا وفهما أفضل من النظام المطور.

• Different viewpoints have different requirements and these are often contradictory.

• وجهات النظر المختلفة لها متطلبات مختلفة وهذه متناقضة في أغلب الأحيان.

Requirements change

• The priority of requirements from different viewpoints changes during the development process.

• أولوية المتطلبات من تغييرات وجهات النظر المختلفة أثناء عملية التطوير.

• System customers may specify requirements from a business perspective that conflict with end-user requirements.

• زبائن النظام قد يحددون المتطلبات من منظور عمل ذلك التعارض بمتطلبات المستخدم.

• The business and technical environment of the system changes during its development.

• العمل والبيئة التقنية تتغير من النظام أثناء تطويره.

• Customers may ask for requirements that are not feasible within the given constraints.

• الزبائن قد يسألون عن المتطلبات التي من المحتمل أن تكون ليست ضمن القيود المعطية.
Requirements evolution

Enduring and volatile requirements

- **Enduring requirements.** Stable requirements derived from the core activity of the customer organisation. E.g. a hospital will always have doctors, nurses, etc. May be derived from domain models.

- **Volatile requirements.** Requirements which change during development or when the system is in use. In a hospital, requirements derived from health-care policy (that may change).
Requirements classification

<table>
<thead>
<tr>
<th>Requirement Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutable requirements</td>
<td>Requirements that change because of changes to the environment in which the organisation is operating. For example, in hospital systems, the funding of patient care may change and thus require different treatment information to be collected.</td>
</tr>
<tr>
<td>Emergent requirements</td>
<td>Requirements that emerge as the customer's understanding of the system develops during the system development. The design process may reveal new emergent requirements.</td>
</tr>
<tr>
<td>Consequential requirements</td>
<td>Requirements that result from the introduction of the computer system. Introducing the computer system may change the organisation's processes and open up new ways of working which generate new system requirements.</td>
</tr>
<tr>
<td>Compatibility requirements</td>
<td>Requirements that depend on the particular systems or business processes within an organisation. As these change, the compatibility requirements on the commissioned or delivered system may also have to evolve.</td>
</tr>
</tbody>
</table>

Requirements management planning

During the requirements engineering process, you have to plan:

- Requirements identification

  - How requirements are individually identified;

  - Requirements that change because of changes to the environment in which the organisation is operating. For example, in hospital systems, the funding of patient care may change and thus require different treatment information to be collected.

  - Requirements that emerge as the customer's understanding of the system develops during the system development. The design process may reveal new emergent requirements.

  - Requirements that result from the introduction of the computer system. Introducing the computer system may change the organisation's processes and open up new ways of working which generate new system requirements.

  - Requirements that depend on the particular systems or business processes within an organisation. As these change, the compatibility requirements on the commissioned or delivered system may also have to evolve.
• A change management process
  • عملية إدارة التغيير
• The process followed when analysing a requirements change;
  • تتبع العملية متى تم تحليل تغيير المتطلبات;
• Traceability policies
  • سياسات قابلية التتبع
• The amount of information about requirements
relationships that is maintained;
• CASE tool support
  أدوات دعم هندسة البرمجيات المعتمدة على الحاسوب
• The tool support required to help manage
requirements change;
  أداة الدعم مطلوبة للمساعدة على إدارة تغيير المتطلبات؛

Traceability

• Traceability is concerned with the relationships between
requirements, their sources and the system design
  قابلية التتبع مهتمة بالعلاقات بين المتطلبات، ومصادرها وتصميم النظام.
• Source traceability
  مصدر قابلية التتبع
  • Links from requirements to stakeholders who proposed
these requirements;
  الربط من المتطلبات إلى المالكين الذين اقترحوا هذه المتطلبات؛
• Requirements traceability
  قابلية التتبع للمتطلبات
  • Links between dependent requirements;
  البربط بين المتطلبات التابعة؛
• Design traceability
  قابلية التتبع للتصميم
  • Links from the requirements to the design;
  الربط من المتطلبات إلى التصميم؛

A traceability matrix

<table>
<thead>
<tr>
<th>Req. id</th>
<th>1.1</th>
<th>1.2</th>
<th>1.3</th>
<th>2.1</th>
<th>2.2</th>
<th>2.3</th>
<th>3.1</th>
<th>3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td></td>
<td>D</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>R</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>R</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td></td>
<td></td>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.3</td>
<td>R</td>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td></td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td></td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CASE tool support**

- **Requirements storage**
  - Requirements should be managed in a secure, managed data store.
  - *المتطلبات يجب أن يتم إدارتها بشكل آمن، إدارة تخزين البيانات.*

- **Change management**
  - The process of change management is a workflow process whose stages can be defined and information flow between these stages partially automated.
  - *إن عملية تغيير الإدارة لعملية تدفق العمل التي مراحلها يمكن أن تعرف وتدفق المعلومات بين هذه مراحل آلية بشكل جزئي.*

- **Traceability management**
  - Automated retrieval of the links between requirements.
  - * الاسترجاع الآلي للروابط بين المتطلبات.*

**Requirements change management**

- Should apply to all proposed changes to the requirements.
  - *يجب أن يقدم إلى كل التغييرات المقترحة إلى المتطلبات.*
Principal stages

- Problem analysis. Discuss requirements problem and propose change;
- Change analysis and costing. Assess effects of change on other requirements;
- Change implementation. Modify requirements document and other documents to reflect change.

Change management

![Diagram of change management process]

Identified problem → Problem analysis and change specification → Change analysis and costing → Change implementation → Revised requirements
Key points
النقاط الرئيسية

- The requirements engineering process includes a feasibility study, requirements elicitation and analysis, requirements specification and requirements management.
- Requirements elicitation and analysis is iterative involving domain understanding, requirements collection, classification, structuring, prioritisation and validation.
- Systems have multiple stakeholders with different requirements.
- Social and organisation factors influence system requirements.
- Requirements validation is concerned with checks for validity, consistency, completeness, realism and verifiability.
- Business changes inevitably lead to changing requirements.
- Requirements management includes planning and change management.
- Requirements set out what the system should do and define constraints on its operation and implementation.

- هندسة المتطلبات عملية تتضمن دراسة الجدو، استنباط وتحليل المتطلبات.
- استنباط المتطلبات تضمن تكراري لتحليل وفهم المجال، مجموعة المتطلبات، التصنيف، التنظيم الفضيل (إعطاء الأولوية) والتصديق.
- الأنظمة لها مالكين متعددين بالمتطلبات المختلفة.
- العوامل الاجتماعية والتنظيمية تؤثر على متطلبات النظام.
- تصديق المتطلبات مهتمة بعمليات المراقبة للصلاحية، الاتساق (التناسق)، الاكتمال، الواقعية وإمكانية التحقق منه.
- تؤدي تغييرات العمل إلى تغيير المتطلبات حتما.
- إدارة المتطلبات تتضمن إدارة التغيير والتخطيط.
- تعرض المتطلبات ماهية النظام وكيف يجب أن يعمل وتعرض تعريف الفيود على عملياتها وتطبيقاتها.
• Functional requirements set out services the system should provide.
  • تعرض المتطلبات الوظيفية الخدمات للنظام الذي يجب أن يزود.
• Non-functional requirements constrain the system being developed or the development process.
  • معوقات المتطلبات غير الوظيفية النظام يكون متطور أو يكون في عملية التطوير.
• There are several ways – other than NL- for writing the requirements. Usually people combine one or more of those types when writing a requirement document.
  • هناك عدة طرق - ما عدا اللغة الطبيعية - لكتابة المتطلبات. عادة الأشخاص يجمعون واحد أو أكثر من تلك الأنواع عندما يقومون بكتابة وثيقة المتطلب.
A software requirements document is an agreed statement of the system requirements. It is the reference throughout the project for what should be developed. It is not only useful for the developers of the software. It acts as the basis for the contract and the project management. It is also acts as a base for writing test cases.

The IEEE standard is a useful starting point for defining more detailed specific requirements standards.