University of Tripoli
Faculty of Information Technology

Department of Software Engineering

Software Requirements Analysis
ITSE311 -- F2024

INTRODUCTION TO REQUIREMENTS ENGINEERING

What Is Requirements Engineering?

- It is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems.
- It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families.

Motivation

- There are four kinds of problems that arise when one fails to do adequate requirements analysis:
  - Top-down design is impossible;
  - Testing is impossible;
  - The user is frozen out;
  - Management is not in control.

➡ **Poor Management**

What Are Requirements?

- Requirements can range from high-level, abstract statements to formal (mathematically rigorous) specifications.
- Why?
  - Stakeholders have needs at different levels, hence, depend on different abstraction representations.
  - Stakeholders also have varying abilities to make and read these representations (for example a business customer versus a design engineer), leading to diverse quality in the requirements.

Requirements Versus Goals

- Goals are high-level objectives of a business, organization, or system, but a requirement specifies how a goal should be accomplished by a proposed system.
- Treat a goal as a requirement is to invite TROUBLE because
  - Achievement of the goal will be difficult to prove.
  - Goals evolve as stakeholders change their minds and refine and operationalize goals into behavioral requirements.

Requirements Level Classification

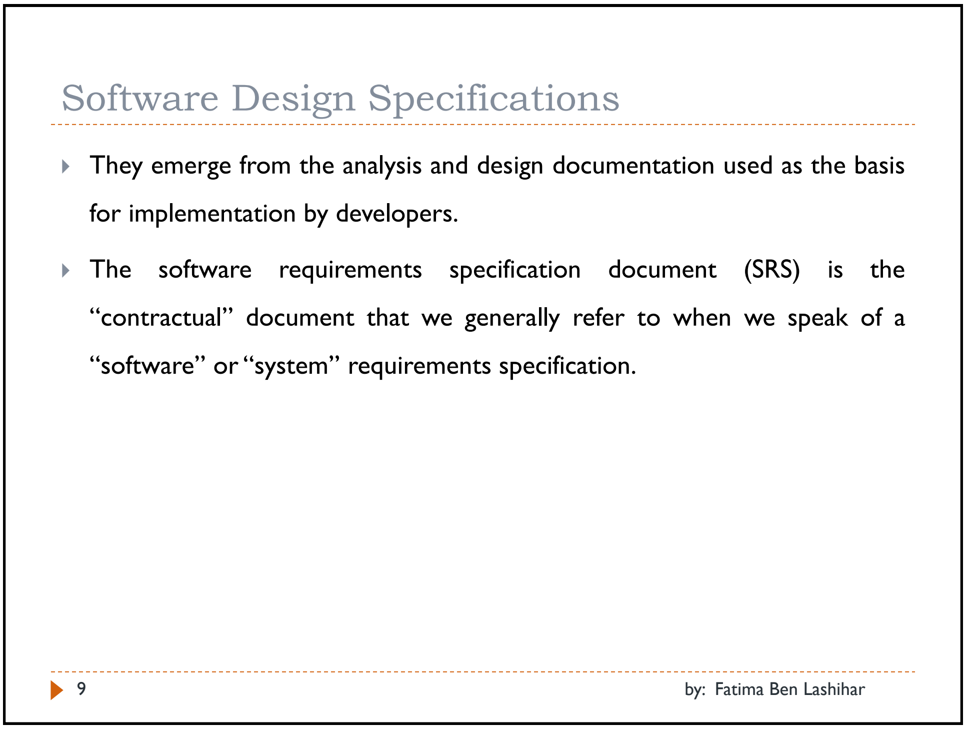
- To deal with the diversity in requirements types, they should be organize into three levels of abstraction:
  - User requirements
  - System requirements
  - Software design specifications

User Requirements

- They are abstract statements written in natural language with accompanying informal diagrams.
- They specify what services (user functionality) the system is expected to provide and any constraints.
- In many situations user stories can play the role of user requirements.

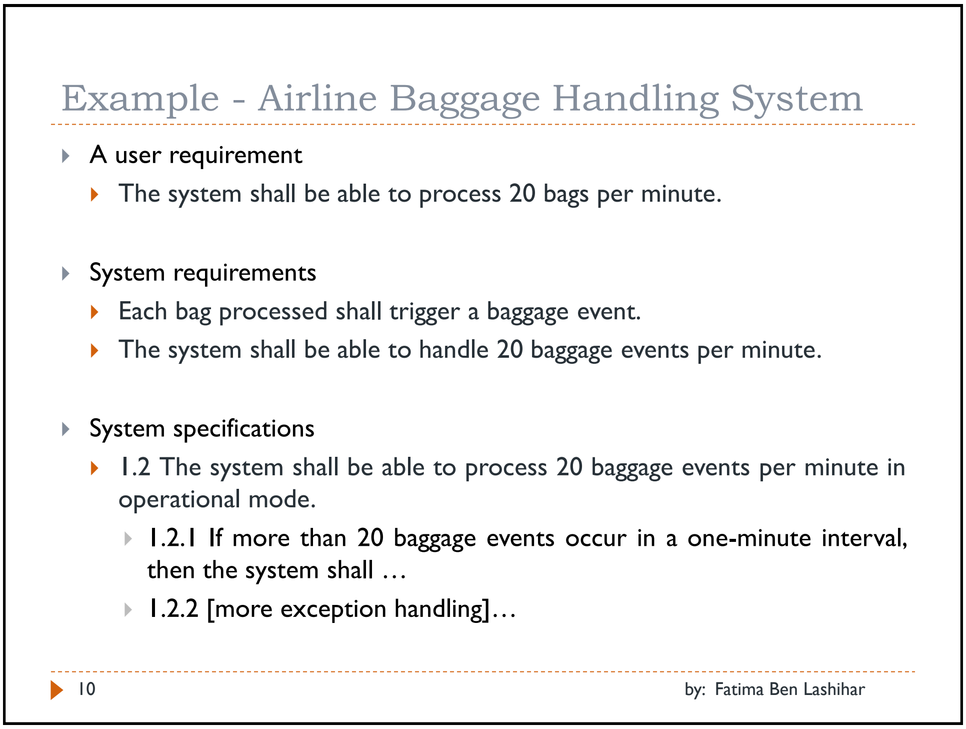
System Requirements

- They are detailed descriptions of the services and constraints.
- They are sometimes referred to as functional specification or technical annex.
- These requirements are derived from analysis of the user requirements.
- They act as a contract between client and contractor, so they should be structured and precise.
- Use cases can play the role of system requirements in many situations.



Software Design Specifications

- They emerge from the analysis and design documentation used as the basis for implementation by developers.
- The software requirements specification document (SRS) is the “contractual” document that we generally refer to when we speak of a “software” or “system” requirements specification.



Example - Airline Baggage Handling System

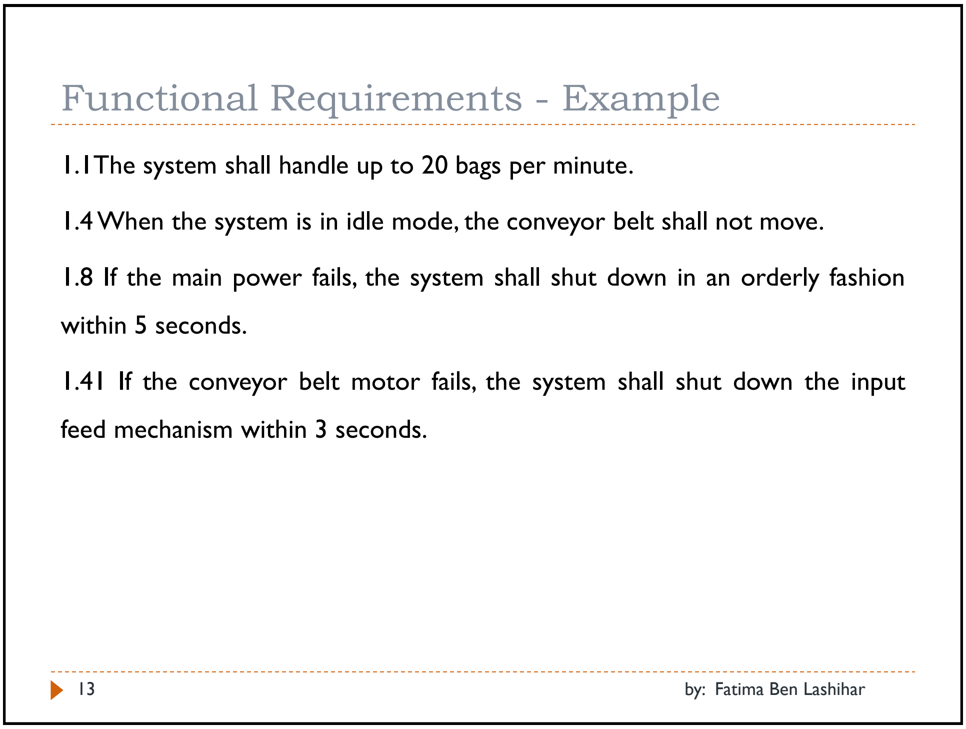
- A user requirement
  - The system shall be able to process 20 bags per minute.
- System requirements
  - Each bag processed shall trigger a baggage event.
  - The system shall be able to handle 20 baggage events per minute.
- System specifications
  - 1.2 The system shall be able to process 20 baggage events per minute in operational mode.
    - 1.2.1 If more than 20 baggage events occur in a one-minute interval, then the system shall …
    - 1.2.2 [more exception handling]…

Requirements Specifications Types

- Functional requirements
- Nonfunctional requirements (NFRs)
- Domain requirement

Functional Requirements

- They describe the services the system should provide and how the system will react to its inputs.
- They need to explicitly state certain behaviors that the system should not do.
- They can be high level and general (in which case they are user requirements in the sense that was explained previously) or they can be detailed, expressing inputs, outputs, exceptions, and so on (in which case they are the system requirements described before).
- There are many forms of representation for functional requirements, from natural language, visual models, and the more rigorous formal methods.



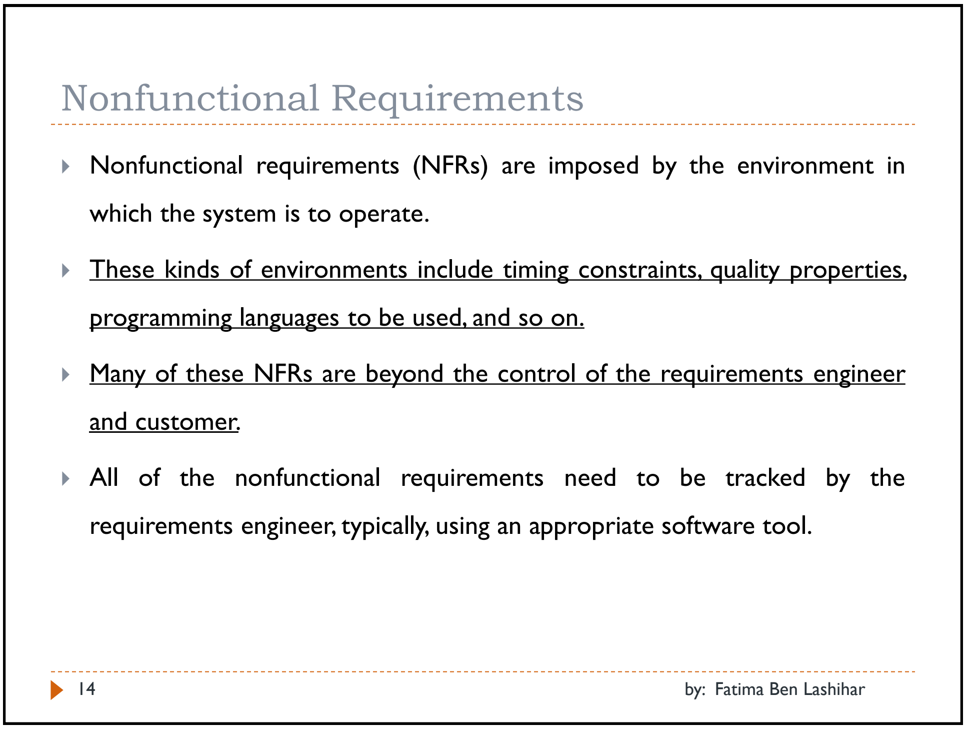
Functional Requirements - Example

1.1 The system shall handle up to 20 bags per minute.

1.4 When the system is in idle mode, the conveyor belt shall not move.

1.8 If the main power fails, the system shall shut down in an orderly fashion within 5 seconds.

1.41 If the conveyor belt motor fails, the system shall shut down the input feed mechanism within 3 seconds.



Nonfunctional Requirements

- Nonfunctional requirements (NFRs) are imposed by the environment in which the system is to operate.
- These kinds of environments include timing constraints, quality properties, programming languages to be used, and so on.
- Many of these NFRs are beyond the control of the requirements engineer and customer.
- All of the nonfunctional requirements need to be tracked by the requirements engineer, typically, using an appropriate software tool.

Nonfunctional Requirements

- Some NFRs are difficult to define precisely, making them difficult to verify.
- It is easy to confuse goals with NFRs. Remember a goal is a general intention of a stakeholder,
  - for example:
    - The system should be easy to use by experienced operators
    - whereas a verifiable NFR is a statement using some objective measure: Experienced operators shall be able to use the following 18 system features after two hours of hands-on instructor-led training with an error rate of no greater than 0.5%.

Nonfunctional Requirements - Example

- Product requirements
  - Efficiency
    - □ Performance (e.g., number of bags per minute)
    - □ Space (e.g., physical size of system, amount of memory, power consumption)
  - Reliability (e.g., Mean time before failure or Mean time before first failure)
  - Portability (e.g., can it be used with other hardware?)
  - Usability (amount of training required)

Nonfunctional Requirements - Example

- Organizational requirements
  - Delivery (e.g., date of delivery, date when fully operational, training sessions, updates)
  - Implementation (e.g., full capability in first roll-out or phased capability)
  - Standards (if there are industry standards for baggage handling systems)
- External requirements
  - Interoperability (e.g., with other equipment, communications standards)
  - Ethical (e.g., security clearance for REs, professional certification)
  - Legislative
    - □ Privacy
    - □ Safety

Domain Requirements

- Domain requirements are derived from the application domain.
- These types of requirements may consist of:
  - New functional requirements
  - Constraints on existing functional requirements
  - They may specify how particular computations must be performed.

Domain Requirements - Example

- There are industry standards (we wouldn't want the new system to under-perform versus other airlines' systems).
- There are constraints imposed by existing hardware available (e.g., conveyor systems).
- And there may be constraints on performance mandated by collective bargaining agreements with the baggage handlers union.

Requirements Engineering Activities

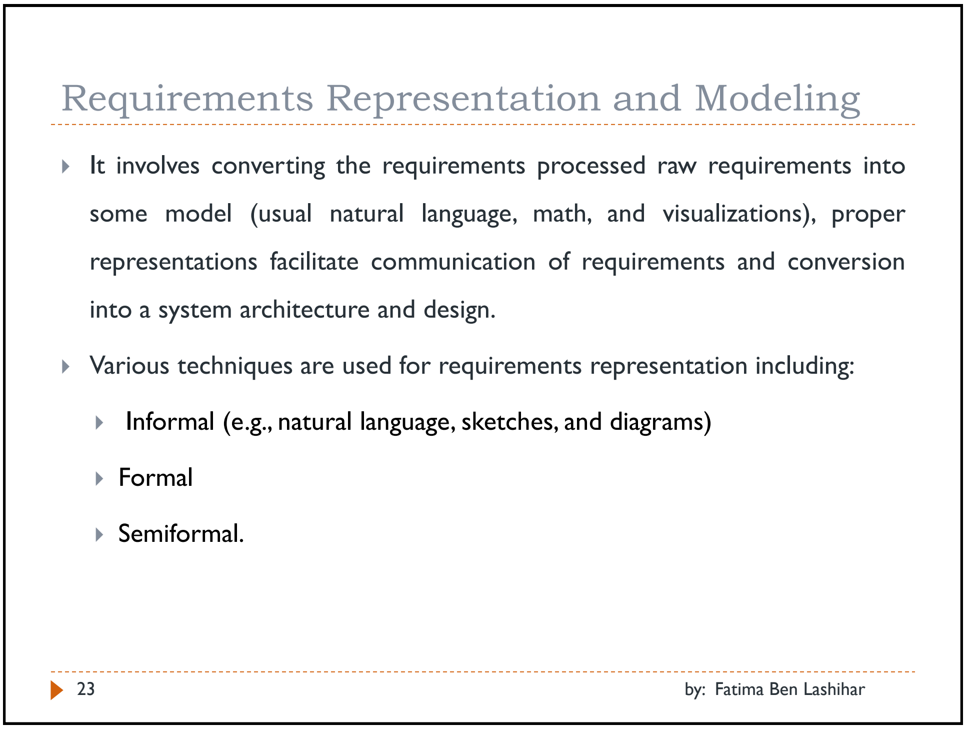
- The requirements engineer is responsible for a number of activities. These include:
  - Requirements Elicitation
  - Requirements Analysis
  - Requirements Representation/Modeling
  - Requirements Verification And Validation
  - Requirements Management

Requirements Elicitation

- It involves uncovering what the customer needs and wants, while some requirements will be obvious, many requirements will need to be discovered from the customer through well-defined approaches.
- Requirements engineering also involves discovering who the stakeholders.
- Elicitation also involves determining the nonfunctional requirements, which are often overlooked.

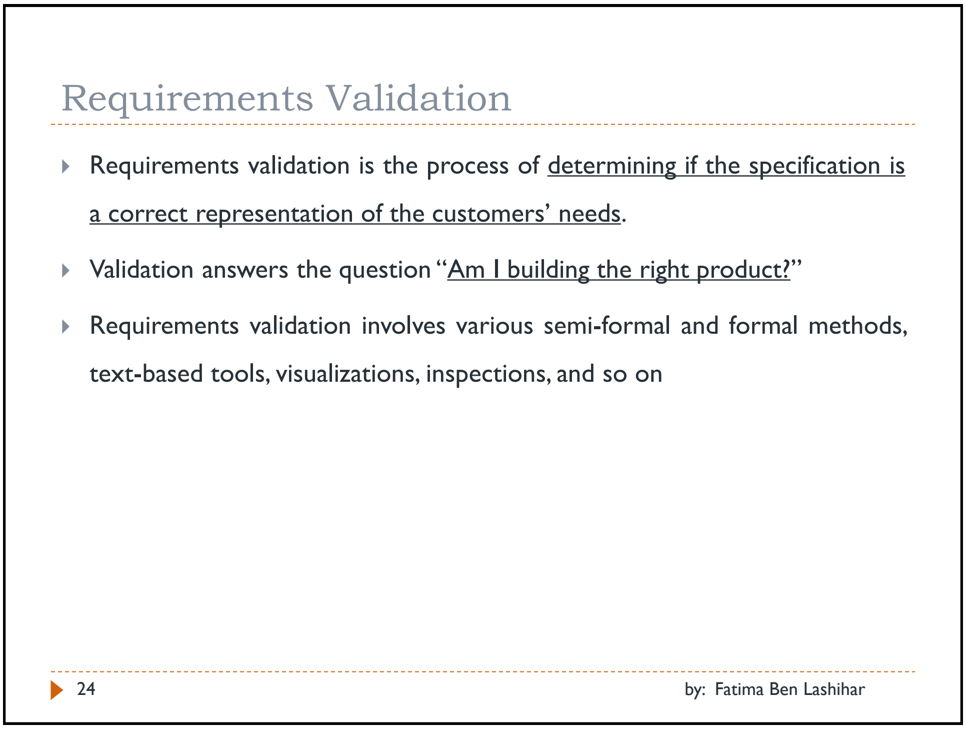
Requirements Analysis

- Requirements analysis involves techniques to deal with a number of problems with requirements in their "raw" form, that is, after they have been collected from the customers. Problems with raw requirements include:
  - They don't always make sense.
  - They often contradict one another (and not always obviously so).
  - They may be inconsistent.
  - They may be incomplete.
  - They may be vague or just wrong.
  - They may interact and are dependent on each other.



Requirements Representation and Modeling

- It involves converting the requirements processed raw requirements into some model (usual natural language, math, and visualizations), proper representations facilitate communication of requirements and conversion into a system architecture and design.
- Various techniques are used for requirements representation including:
  - Informal (e.g., natural language, sketches, and diagrams)
  - Formal
  - Semiformal.



Requirements Validation

- Requirements validation is the process of determining if the specification is a correct representation of the customers' needs.
- Validation answers the question "Am I building the right product?"
- Requirements validation involves various semi-formal and formal methods, text-based tools, visualizations, inspections, and so on

Requirements Management

- One of the most overlooked aspects of requirements engineering.
- Requirements management involves managing the realities of changing requirements over time.
- It also involves communicating changes in requirements to those who need to know.

The Requirements Engineer Skills
مهارات مهندس المتطلبات

- What skills should a requirements engineer have?
- the requirements engineer should:
  - be organized منظم
  - has experience throughout the (software) engineering lifecycle
    لديه خبرة طوال دورة حياة هندسة البرمجيات
  - has the maturity to know when to be general and when to be specific
    لديه النضج لمعرفة متى يكون عاما أو محددا
  - be able to stand up to the customer when necessary
    يكون قادر على الوقوف في وجه العميل عند الضرورة
  - be a good manager مدير جيد
  - Be a good listener, fair, a good negotiator, multidisciplinary
    مستمع جيد، عادل، مفاوض، ومتعدد التخصصات
  - understand the problem domain فهم مجال المشكلة

The Requirements Engineer Skills (cont’d)

- Be thinking: requirements engineers are structured and logical

مفكرا: مهندسو المتطلبات منظمون ومنطقيون

- Be sensing: focus on information gathered and do not try to interpret it

مستشعرا: ركز على المعلومات التي تم جمعها ولا تحاول تفسيرها

- Be judging: seek closure rather than leaving things open

حَكَمًا: ابحث عن الإغلاق بدلا من ترك الأمور مفتوحة

Requirements Engineering Paradigms
نماذج هندسة المتطلبات

- Another way to understand the nature of requirements engineering is to look at various models for the role of the requirements engineer:

طريقة أخرى لفهم طبيعة هندسة المتطلبات هي بالنظر إلى نماذج مختلفة لدور مهندس المتطلبات:

  - requirements engineer as software systems engineer
  - requirements engineer as subject matter expert
  - requirements engineer as architect
  - the ignorant requirements engineer

Requirements Engineer as Software Systems Engineer مهندس المتطلبات كمهندس أنظمة البرمجيات

- It is likely that many requirements engineers are probably former software systems designers or developers.

من المحتمل أن يكون العديد من مهندسي المتطلبات من مصممي أو مطوري أنظمة البرمجيات السابقين.

- The positive point, the requirements engineer can influence downstream development of models (e.g., the software design).

لنقطة الإيجابية ، يمكن لمهندس المتطلبات التأثير على تطوير النماذج النهائية (على سبيل المثال ، تصميم البرنامج).

- The danger in this case is that the requirements engineer may begin to create a design when he should be developing requirements specifications.

الخطر في هذه الحالة هو أن مهندس المتطلبات قد يبدأ في إنشاء تصميم عندما يجب أن يقوم بتطوير مواصفات المتطلبات.

Requirements Engineer as Subject Matter Expert مهندس المتطلبات كخبير في الموضوع

- In many cases the customer is looking to the requirements engineer to be a subject matter expert (SME) for expertise either in helping to understand the problem domain or in understanding the customers' own wants and desires.

في كثير من الحالات، يتطلع العميل إلى مهندس المتطلبات ليكون SME للحصول على الخبرة إما في المساعدة على فهم مجال المشكلة أو في فهم رغبات العملاء ورغباتهم الخاصة.

- Sometimes the requirements engineer isn't an SME—they are an expert in requirements engineering. In those cases where the requirements engineer is not an SME, consider joining forces with an SME.

في بعض الأحيان لا يكون مهندس المتطلبات خبير في موضوع محدد - فهو خبير في هندسة المتطلبات. في الحالات التي لا يكون فيها مهندس المتطلبات خبير بالموضوع ، فكر في توحيد الجهود مع خبراء.

SME : is a person who has a deep understanding of a particular subject, which can be anything1. They may have gained their knowledge through education, experience, or both

Requirements Engineer as Architect
مهندس المتطلبات كمهندس معماري

Home Building	*Software/System Building*
Architect meets with and interviews clients. Tours property. Takes notes and pictures.	Requirements engineer meets with customers and uses interviews and other elicitation techniques.

المهندس المعماري يلتقي مع العملاء ويجرون مقابلات معهم. جولات عقارية. يدون الملاحظات والصور.	مهندس المتطلبات يجتمع مع العملاء ويستخدم المقابلات و تقنيات الاستنباط الأخرى

Architect makes rough sketches (shows to clients, receives feedback).	Requirements engineer makes models of requirements to show to customers (for example, prototypes, draft SRS)

مهندس معماري يصنع الرسومات التقريبية (يعرضها للعملاء ، يتلقى الملاحظات).	مهندس المتطلبات يصنع النماذج من المتطلبات لإظهارها للعملاء (على سبيل المثال ، النماذج الأولية ، مسودة SRS).

Requirements Engineer as Architect

Home Building	*Software/System Building*
Architect prepares models with additional detail (floor plans).	Requirements engineer uses information determined above to develop complete SRS

مهندس معماري يعد النماذج مع تفاصيل إضافية (مخططات الطوابق)	يستخدم مهندس المتطلبات المعلومات المحددة أعلاه إلى تطوير SRS كاملة

Future models (for example, construction drawings) are for contractors' use.	Future models (for example, software design documents) are for developers' use.

النماذج المستقبلية (على سبيل المثال ، رسومات البناء) هي لاستخدام المقاولين.	النماذج المستقبلية (على سبيل المثال ، البرامج وثائق التصميم) هي لاستخدام المطورين.

Ignorance as Virtue
الجهل كفضيلة

- The "ignorant" requirements engineer is completely in opposition to the role of subject matter expert.

  مهندس المتطلبات "الجاهل" يتعارض تمامًا مع دور الخبير في الموضوع.

- The "ignorant" people ask the "dumb" questions, and the experts answer these questions.

  "الجاهل" يطرح الأسئلة "الغبية"، والخبراء يجيبون على هذه الأسئلة.

Role of the Customer?
دور العميل؟

- Helping the requirements engineer understand what they need and want (elicitation and validation)

  مساعدة مهندس المتطلبات على فهم ما يحتاجونه ويريدونه (الاستنباط والتحقق من الصحة)

- Helping the requirements engineer understand what they don't want (elicitation and validation)

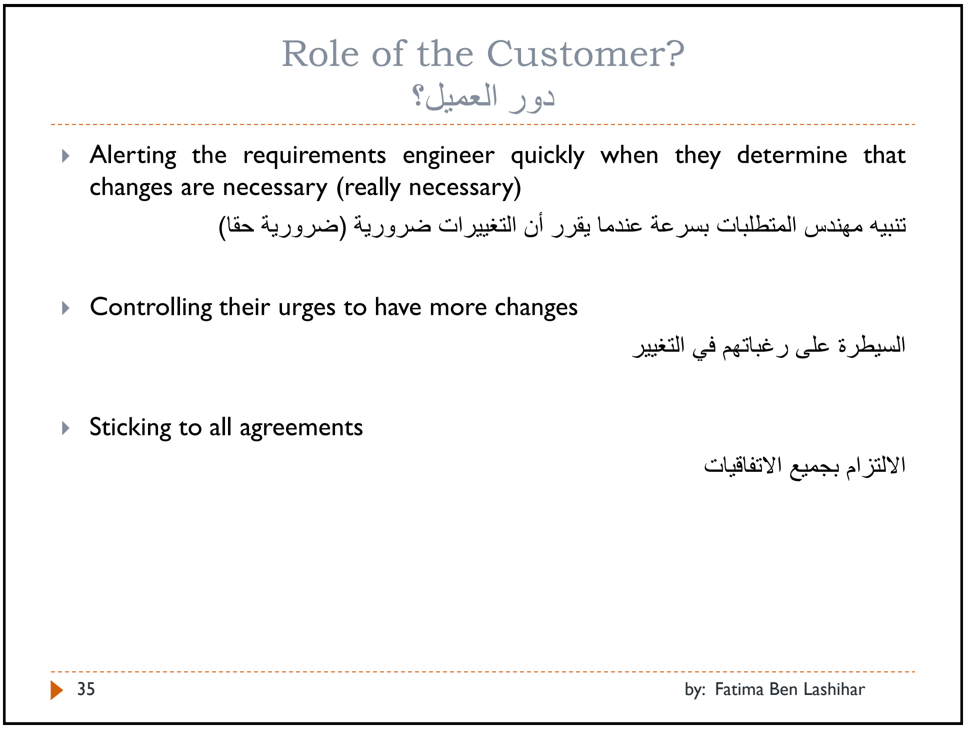
  مساعدة مهندس المتطلبات على فهم ما لا يريدونه (الاستنباط والتحقق من الصحة)

- Providing domain knowledge when necessary and possible

  توفير المعرفة بالمجال عند الضرورة والإمكان

- Alerting the requirements engineer quickly and clearly when they discover that they or others have made mistakes

  تنبيه مهندس المتطلبات بسرعة ووضوح عندما يكتشفون أنهم أو غيرهم قد ارتكبوا أخطاء



Role of the Customer?
دور العميل؟

- Alerting the requirements engineer quickly when they determine that changes are necessary (really necessary)

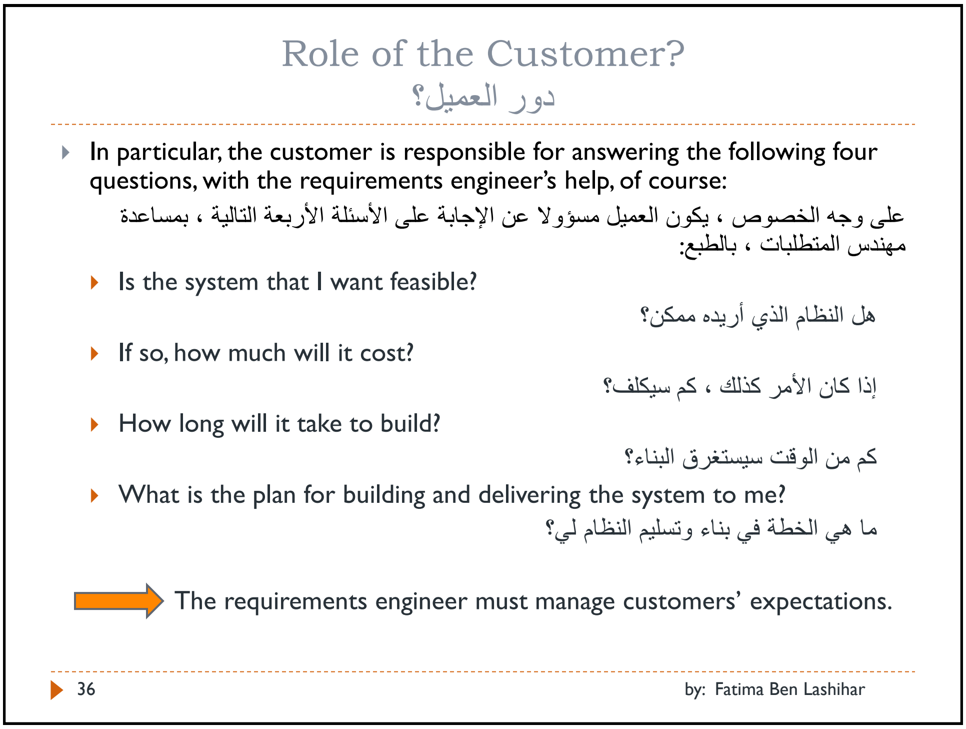
  تنبيه مهندس المتطلبات بسرعة عندما يقرر أن التغييرات ضرورية (ضرورية حقا)

- Controlling their urges to have more changes

  السيطرة على رغباتهم في التغيير

- Sticking to all agreements

  الالتزام بجميع الاتفاقيات



Role of the Customer?
دور العميل؟

- In particular, the customer is responsible for answering the following four questions, with the requirements engineer's help, of course:

  على وجه الخصوص ، يكون العميل مسؤولا عن الإجابة على الأسئلة الأربعة التالية ، بمساعدة مهندس المتطلبات ، بالطبع:

  - Is the system that I want feasible?

    هل النظام الذي أريده ممكن؟

  - If so, how much will it cost?

    إذا كان الأمر كذلك ، كم سيكلف؟

  - How long will it take to build?

    كم من الوقت سيستغرق البناء؟

  - What is the plan for building and delivering the system to me?

    ما هي الخطة في بناء وتسليم النظام لي؟

➡ The requirements engineer must manage customers' expectations.

Problems with Traditional Requirements Engineering مشاكل في هندسة المتطلبات التقليدية

- Traditional requirements engineering approaches suffer from a number of problems, include:

  تعاني مناهج هندسة المتطلبات التقليدية من عدد من المشاكل، منها:

  - Natural language problems (e.g., Ambiguity, imprecision)

    مشاكل اللغة الطبيعية (مثل الغموض وعدم الدقة)
  - Domain understanding

    فهم المجال
  - Dealing with complexity

    التعامل مع التعقيد
  - Incompleteness (missing functionality)

    عدم اكتمال (وظائف مفقودة)

Problems with Traditional Requirements Engineering مشاكل في هندسة المتطلبات التقليدية

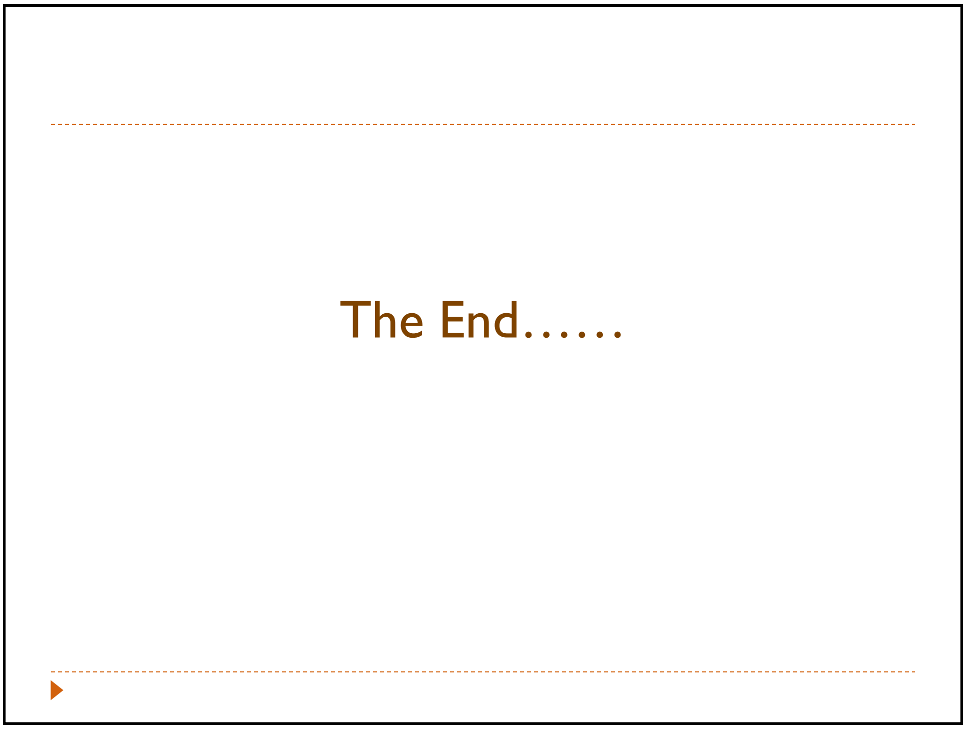
- Over-completeness

  الإفراط في الاكتمال
- Overextension

  الإفراط في التمديد
- Inconsistency

  عدم تناسق
- Incorrectness

  خطأ



The End……