Requirements Engineering

Hoan Ng

Agenda

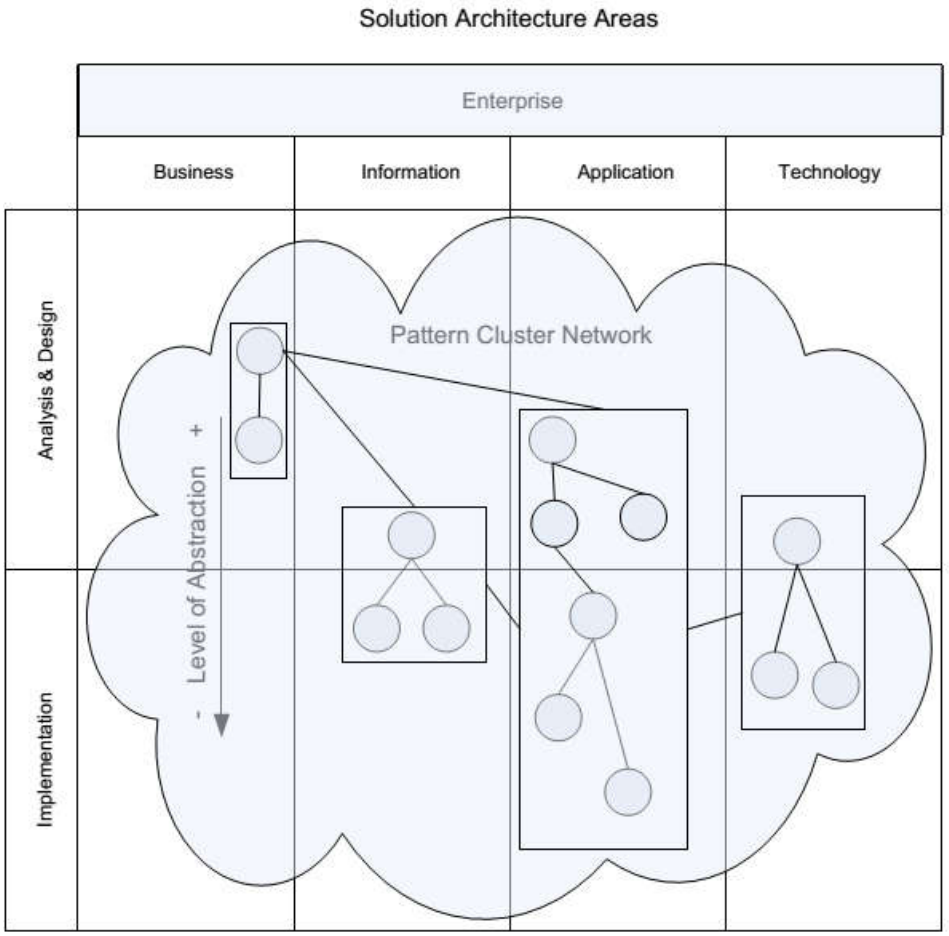
- Requirement Definition
- Requirement Engineering
- Requirement Modelling

What is Requirement?

- Requirement is anything that the business needs to have implemented in the solution.
- 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.

Requirement
The system shall be able to automatically approve or deny credit.
When the credit score is above 750, the system shall automatically approve credit.
The system shall use the following algorithm when automatically determining credit approvals for scores less than 750: [*algorithm would be included here*]
Approvals shall be returned to the user within 30 seconds.

Requirements Examples



Requirements abstraction

Requirements definition/specification

- **Requirements definition**
  - A statement in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers
- **Requirements specification**
  - A structured document setting out detailed descriptions of the system services. Written as a contract between client and contractor
- **Software specification**
  - A detailed software description which can serve as a basis for a design or implementation. Written for developers

Requirements definition

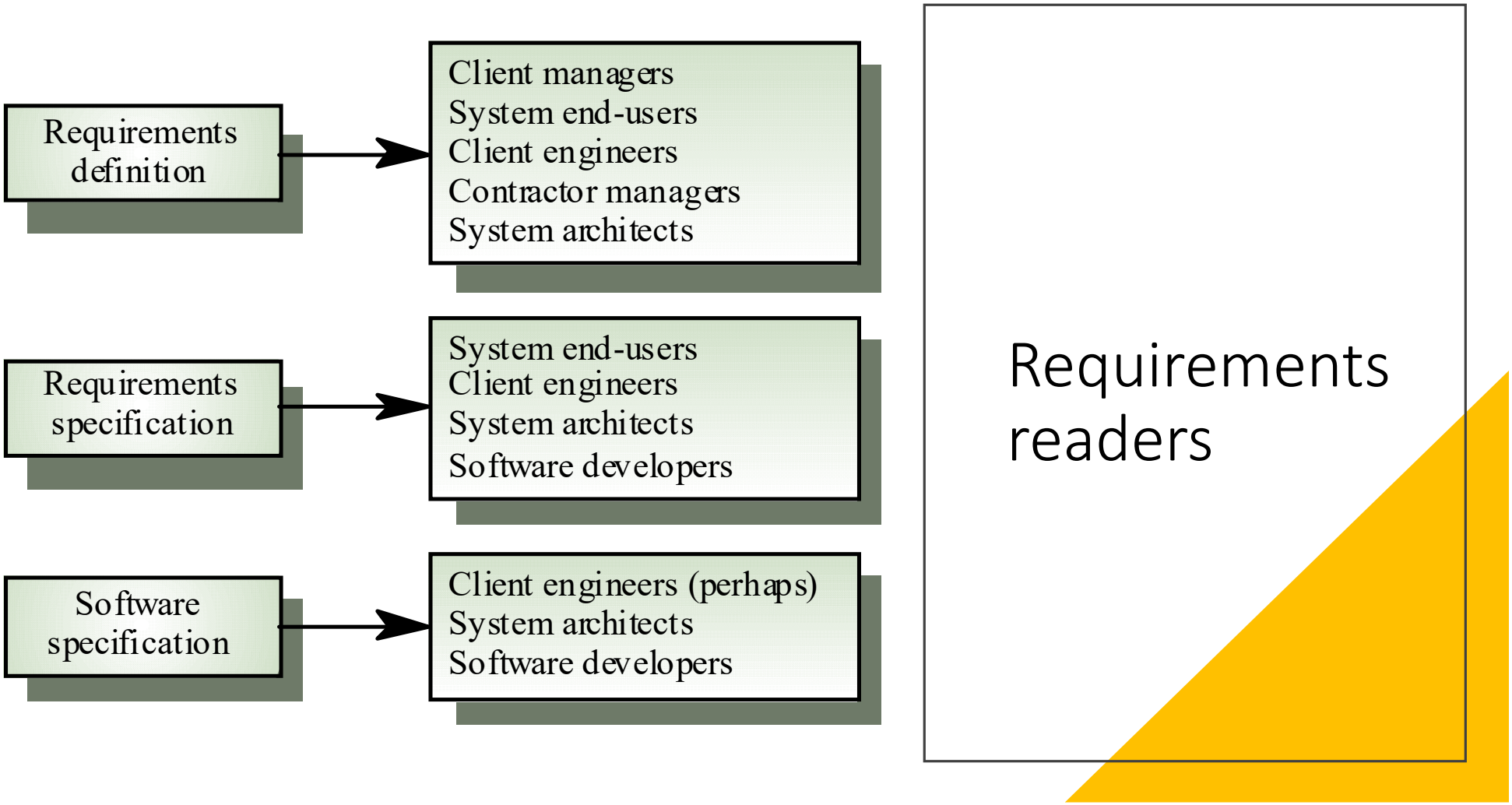
1. The software must provide a means of representing and accessing external files created by other tools.

Requirements specification

1.1 The user should be provided with facilities to define the type of external files.
1.2 Each external file type may have an associated tool which may be applied to the file.
1.3 Each external file type may be represented as a specific icon on the user's display.
1.4 Facilities should be provided for the icon representing an external file type to be defined by the user.
1.5 When a user selects an icon representing an external file, the effect of that selection is to apply the tool associated with the type of the external file to the file represented by the selected icon.

Definitions and specifications

Requirements readers



Wicked problems

- Most large software systems address wicked problems
- Problems which are so complex that they can never be fully understood and where understanding develops during the system development
- Therefore, requirements are normally both incomplete and inconsistent

Reasons for inconsistency

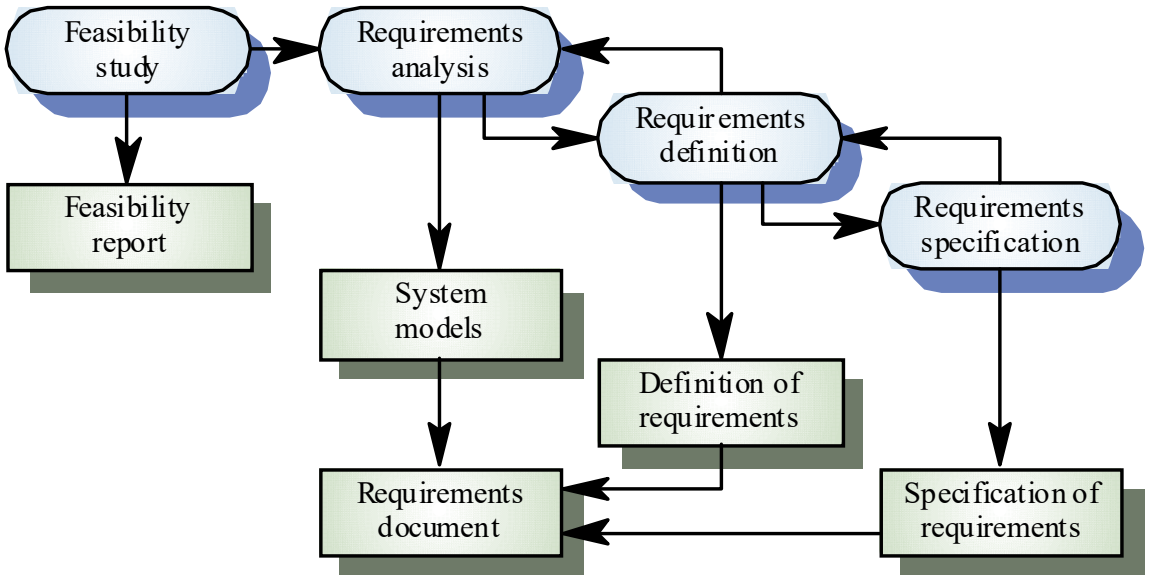
- Large software systems must improve the current situation. It is hard to anticipate the effects that the new system will have on the organisation
- Different users have different requirements and priorities. There is a constantly shifting compromise in the requirements
- System end-users and organisations who pay for the system have different requirements
- Prototyping is often required to clarify requirements

Requirements Engineering

- The process of establishing the services that a customer requires from a system and the constraints under which it operates and is developed.
- The system requirements are the descriptions of the system services and constraints that are generated during the requirements engineering process.

The requirements engineering process

- Feasibility study
  - Find out if the current user needs be satisfied given the available technology and budget?
- Requirements analysis
  - Find out what system stakeholders require from the system
- Requirements definition
  - Define the requirements in a form understandable to the customer
- Requirements specification
  - Define the requirements in detail



The RE process

The requirements document

- The requirements document is the official statement of what is required of the system developers
- Should include both a definition and a specification of requirements
- 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

Requirements document requirements

- Specify external system behaviour
- Specify implementation constraints
- Easy to change
- Serve as reference tool for maintenance
- Record forethought about the life cycle of the system i.e. predict changes
- Characterise responses to unexpected events

Requirements document structure

- Introduction
  - Describe need for the system and how it fits with business objectives
- Glossary
  - Define technical terms used
- System models
  - Define models showing system components and relationships
- Functional requirements definition
  - Describe the services to be provided

Requirements document structure

- Non-functional requirements definition
  - Define constraints on the system and the development process
- System evolution
  - Define fundamental assumptions on which the system is based and anticipated changes
- Requirements specification
  - Detailed specification of functional requirements
- Appendices
  - System hardware platform description
  - Database requirements (as an ER model perhaps)
- Index

Requirements validation

- Concerned with demonstrating that the requirements define the system that the customer really wants
- Requirements error costs 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
- Prototyping (discussed in Chapter 8) is an important technique of requirements validation

Requirements checking

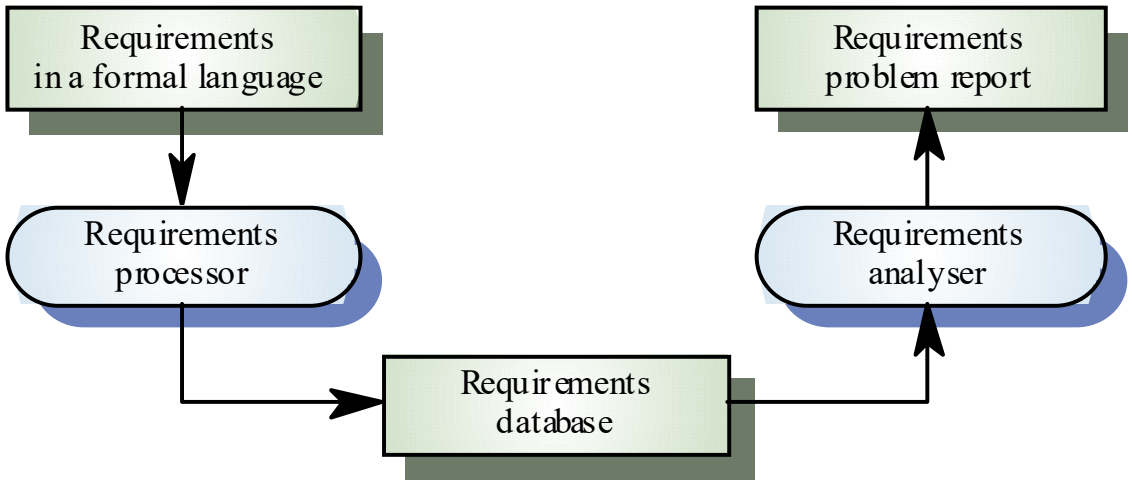
- Validity. Does the system provide the functions which best support the customer's needs?
- Consistency. Are there any requirements conflicts?
- Completeness. Are all functions required by the customer included?
- Realism. Can the requirements be implemented given available budget and technology

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

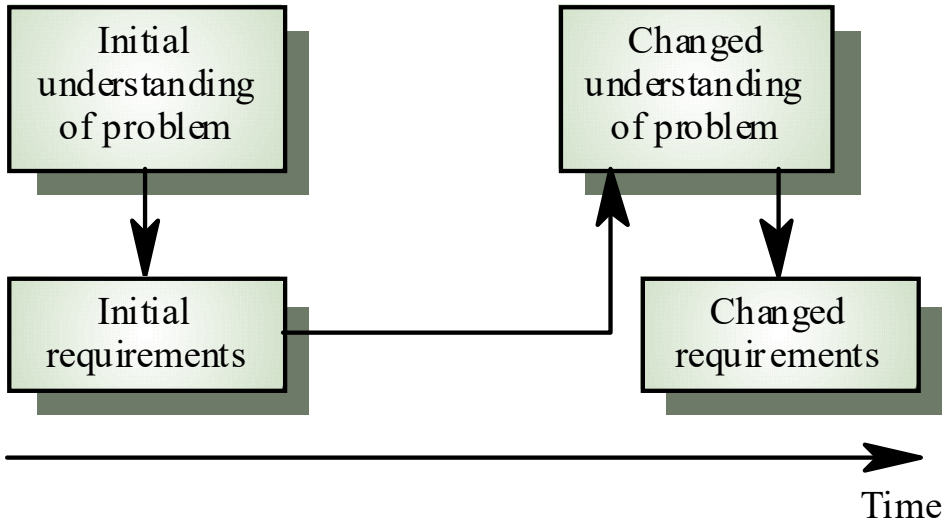
- Verifiability. Is the requirement realistically testable?
- Comprehensibility. Is the requirement properly understood?
- Traceability. Is the origin of the requirement clearly stated?
- Adaptability. Can the requirement be changed without a large impact on other requirements?



Automated consistency checking

Requirements evolution

- Requirements always evolve as a better understanding of user needs is developed and as the organization's objectives change
- It is essential to plan for change in the requirements as the system is being developed and used



Requirements evolution

Requirements classes

- 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

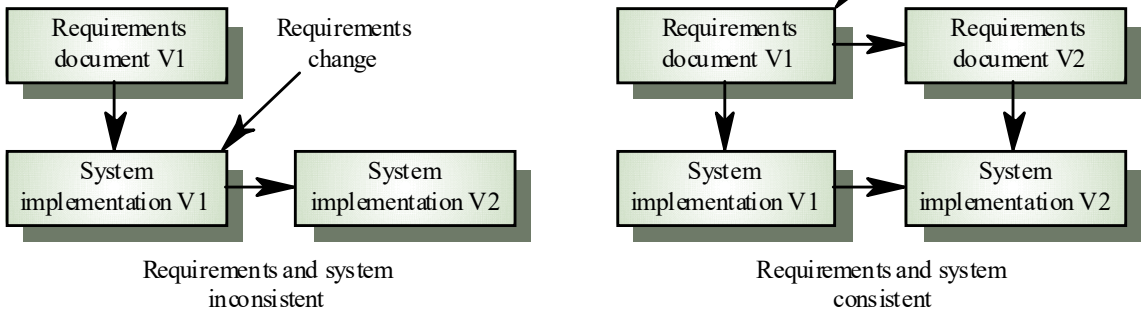
Classification of requirements

- Mutable requirements
  - Requirements that change due to the system's environment
- Emergent requirements
  - Requirements that emerge as understanding of the system develops
- Consequential requirements
  - Requirements that result from the introduction of the computer system
- Compatibility requirements
  - Requirements that depend on other systems or organisational processes

Requirements document changes

- The requirements document should be organised so that requirements changes can be made without extensive rewriting
- External references should be minimised and the document sections should be as modular as possible
- Changes are easiest when the document is electronic. Lack of standards for electronic documents make this difficult

Controlled evolution



Key points

- It is very difficult to formulate a complete and consistent requirements specification
- A requirements definition, a requirements specification and a software specification are ways of specifying software for different types of reader
- The requirements document is a description for customers and developers

Key points

- Requirements errors are usually very expensive to correct after system delivery
- Reviews involving client and contractor staff are used to validate the system requirements
- Stable requirements are related to core activities of the customer for the software
- Volatile requirements are dependent on the context of use of the system

Long list of requirement

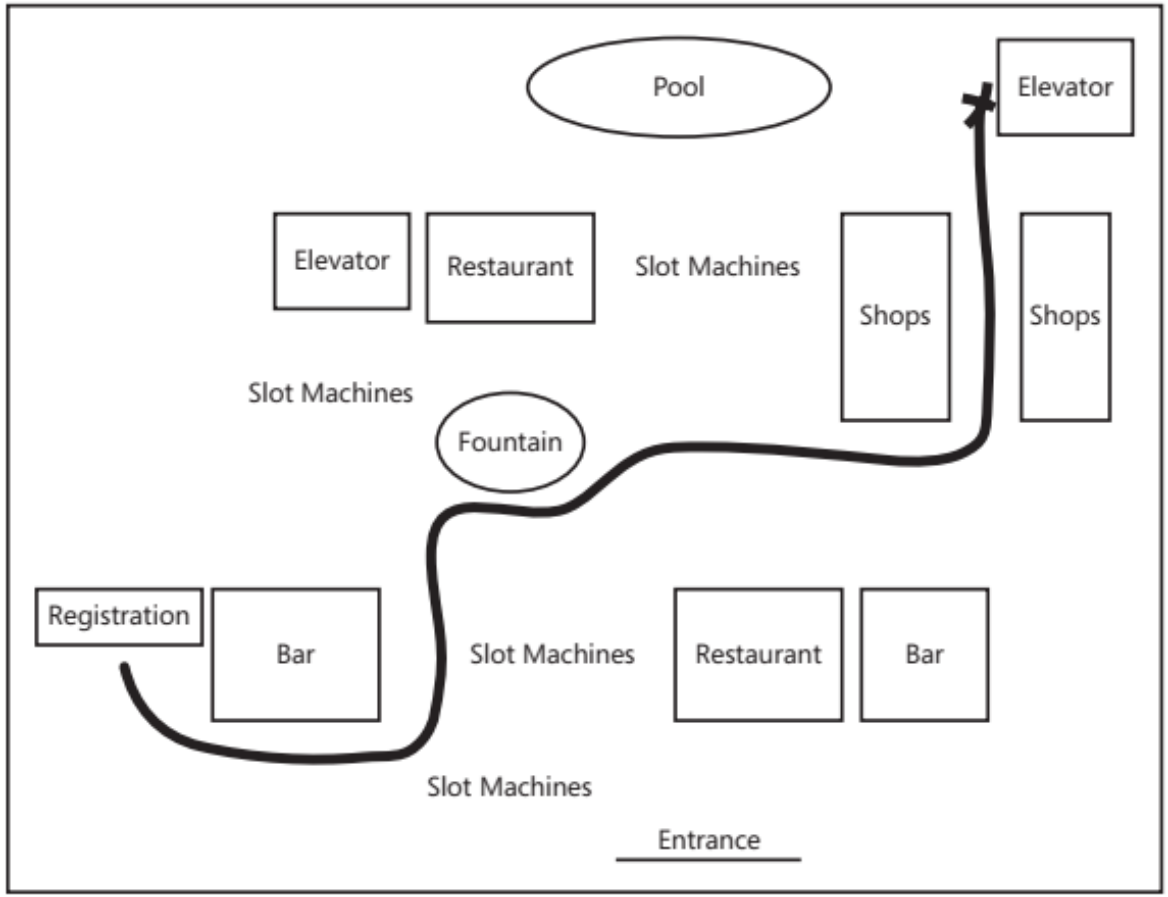
Requirements Document

REQ001	System shall have fields for firstname, middle initial and last name.
REQ002	System shall display a name if there is one in the stored profile.
REQ003	System shall require name is completed.
REQ004	System shall have a field for position or title.
REQ005	System shall require title is completed.
REQ006	System shall display a position or title if there is one in the stored profile.
REQ007	System shall have a field for email address.
REQ008	System shall have a field for alternate email address.
REQ009	System shall display an email address if there is one in the stored profile.
REQ010	System shall display an alternate email address if there is one in the stored profile.
REQ011	System shall require email address is completed.
REQ012	System shall require alternated email address is completed.
REQ013	System shall have a field for daytime phone number.
REQ014	System shall display a phone number if there is one in the stored profile.
REQ015	System shall require phone number is completed.
REQ016	System shall validate all characters in the phone number field are digits when user exits the field.
REQ017	System shall display an error message if not all characters in the phone number field were digits.
REQ018	System shall have a field for a fax number.
REQ019	System shall require fax is completed.
REQ020	System shall display a fax number if there is one in the stored profile.
REQ021	System shall validate all characters in the fax number field are digits when user exits the field.
REQ022	System shall display an error message if not all characters in the fax number field were digits.
REQ023	System shall have two fields for a street address.
REQ024	System shall require the first street address field is completed.
REQ025	System shall display an address if there is one in the stored profile.
REQ026	System shall have a field for city.
REQ027	System shall require the city field is completed.
REQ028	System shall display a city if there is one in the stored profile.
REQ029	System shall have a field for state.
REQ030	System shall display a state if there is one in the stored profile.
REQ031	System shall require the state field is completed.
REQ032	System shall have a field for zip code.
REQ033	System shall display a zip code if there is one in the stored profile.
REQ034	System shall require the zip code field is completed.

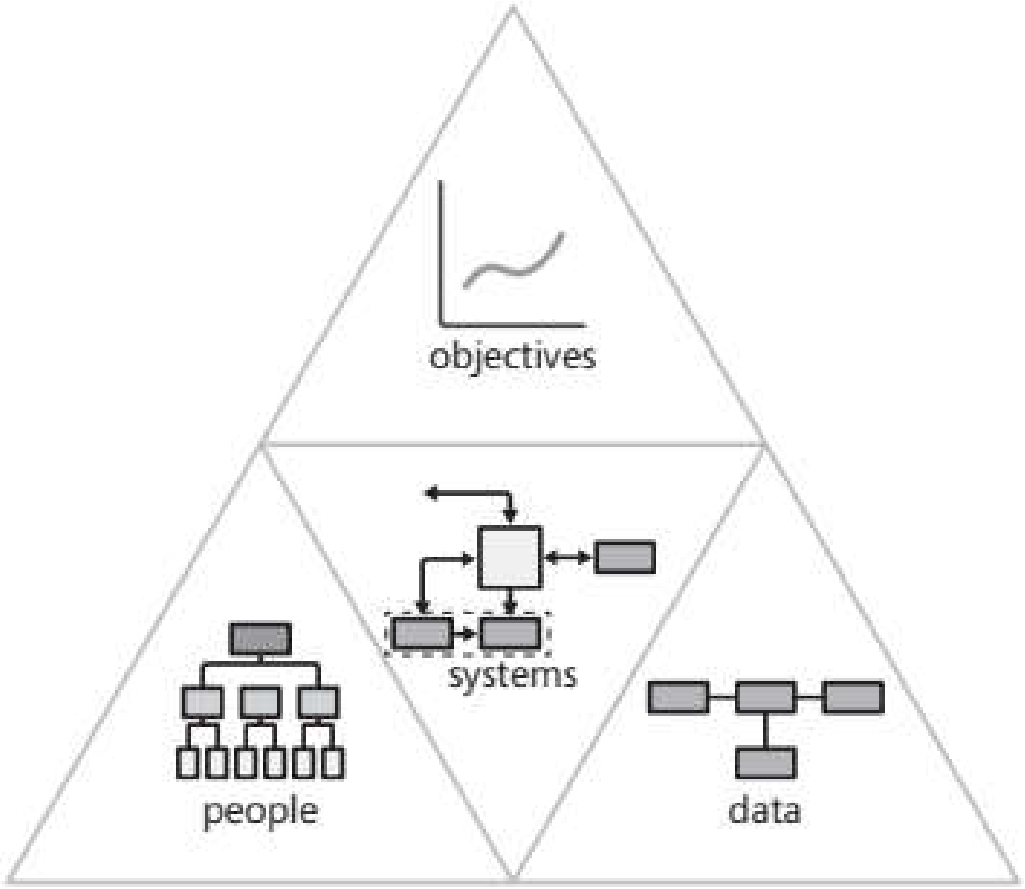
Requirement Modelling

- **Models** are visual representations (pictures) of information related to the processes, data, and interactions within and surrounding the solution being developed

Pictures Are Easy, Words Are Hard



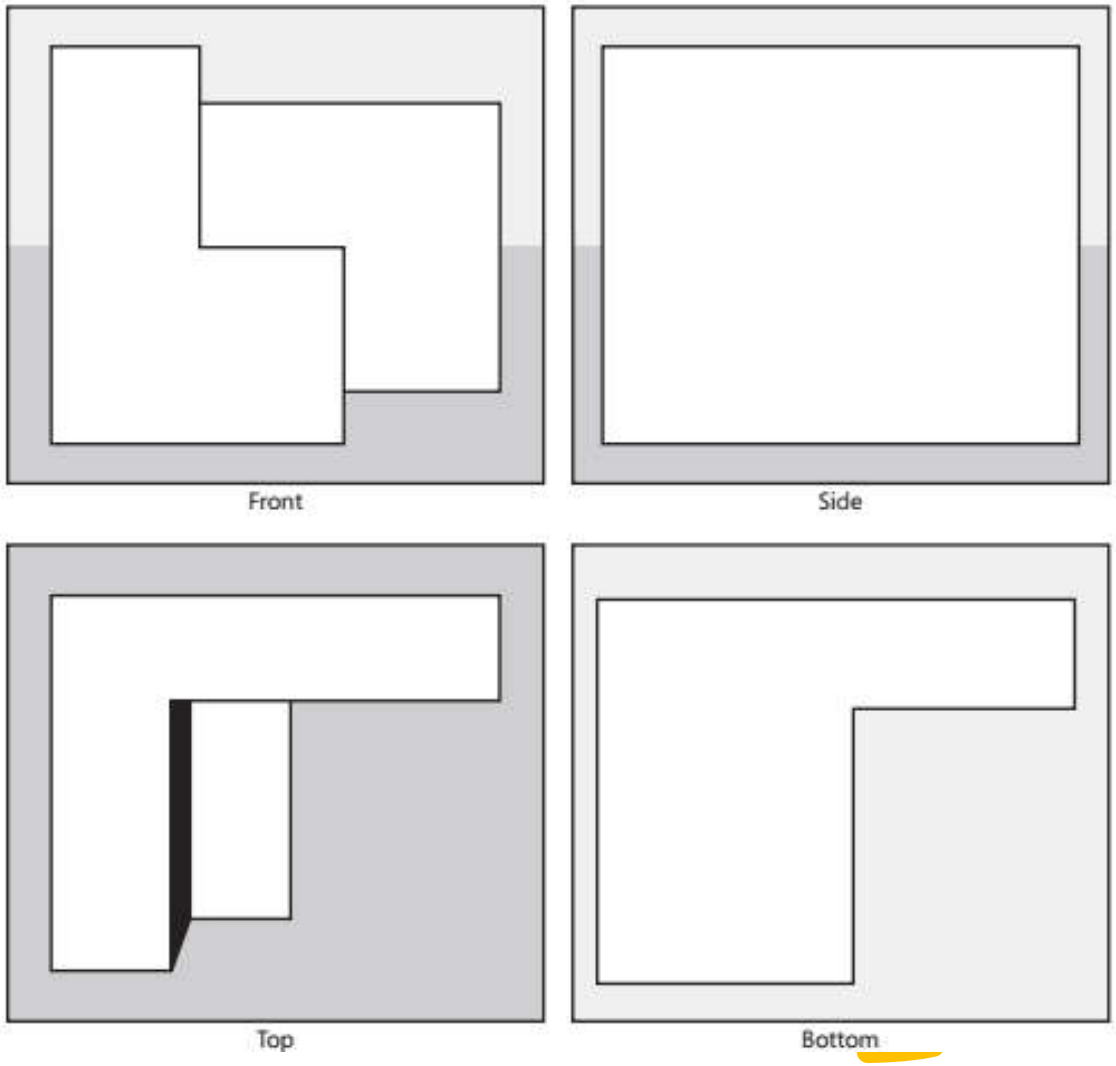
Model Categorization



RML Model Categorization

	Description	Models	Bounding Model
Objectives	Describe the business value of the system and help you prioritize features and requirements based on their value	Business Objectives Model Objective Chain Key Performance Indicator Model Feature Tree Requirements Mapping Matrix	A Business Objectives Model bounds the objectives space
People	Describe who is using the system, along with their business processes and goals	Org Chart Process Flow Use Case Roles and Permissions Matrix	An Org Chart bounds the people space
Systems	Describe what systems exist, what the user interface looks like, how the systems interact, and how they behave	Ecosystem Map System Flow User Interface Flow Display-Action-Response Decision Table Decision Tree System Interface Table	An Ecosystem Map bounds the systems space
Data	Describe the relationships between business data objects from an end-user perspective, the life cycle of the data, and how that data is used in reports to make decisions	Business Data Diagram Data Flow Diagram Data Dictionary State Table State Diagram Report Table	A Business Data Diagram bounds the data space

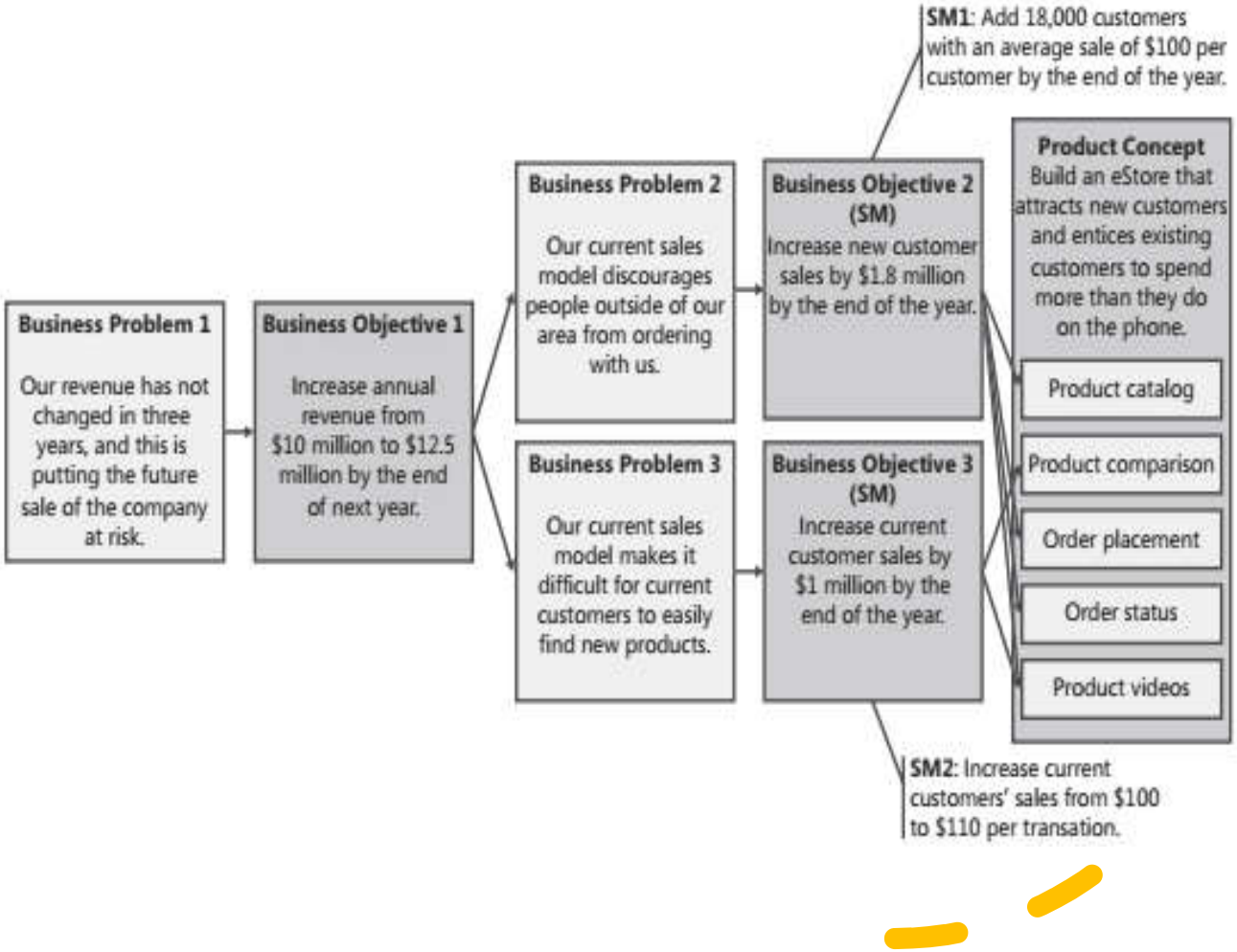
All Four Categories Are Needed



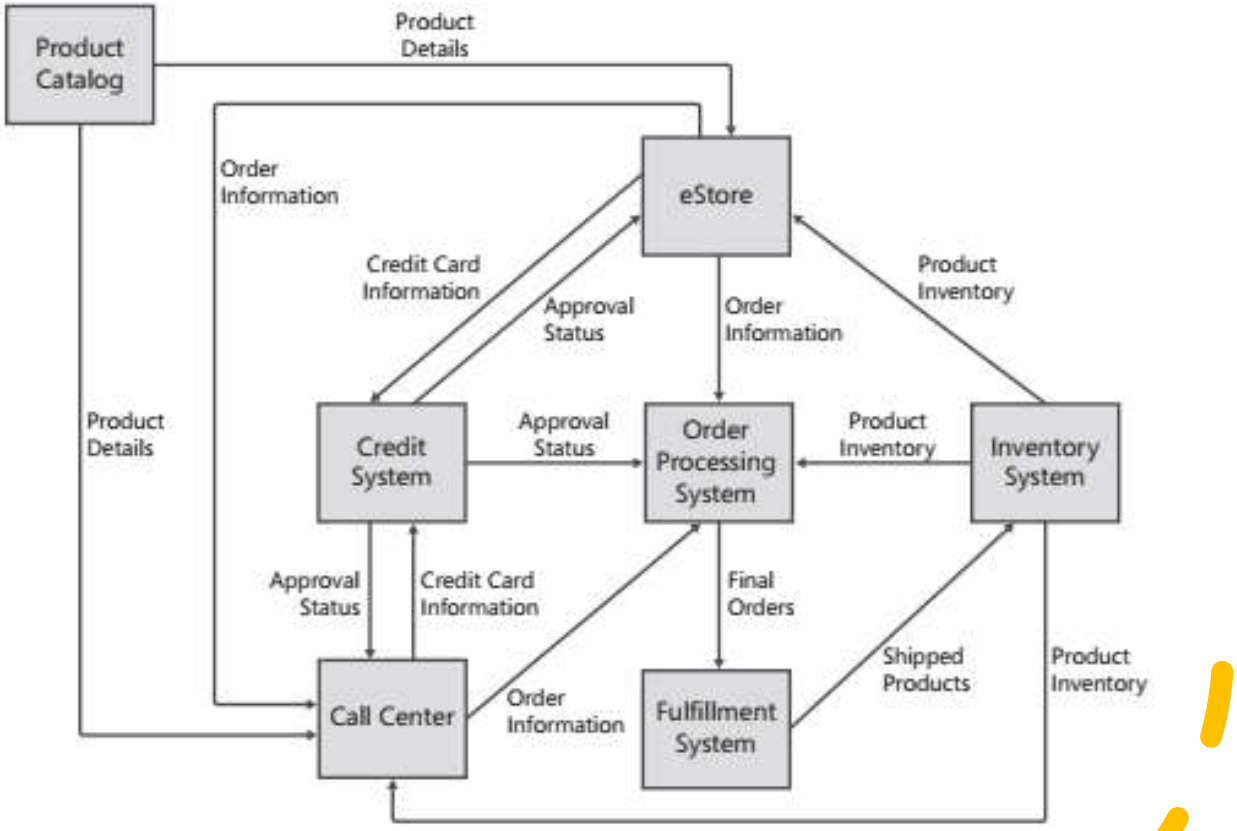
Four views of a product

- **Objectives models** describe the business value of the system and help you prioritize features and requirements based on their value
- **People models** describe the stakeholders of the system, their business processes, and their goals
- **Systems models** describe what systems exist, what the user interface looks like, how the systems interact, and how they behave
- **Data models** describe the relationships of business data objects from an end-user perspective, the life cycle of the data, and how that data is used in reports to make decisions

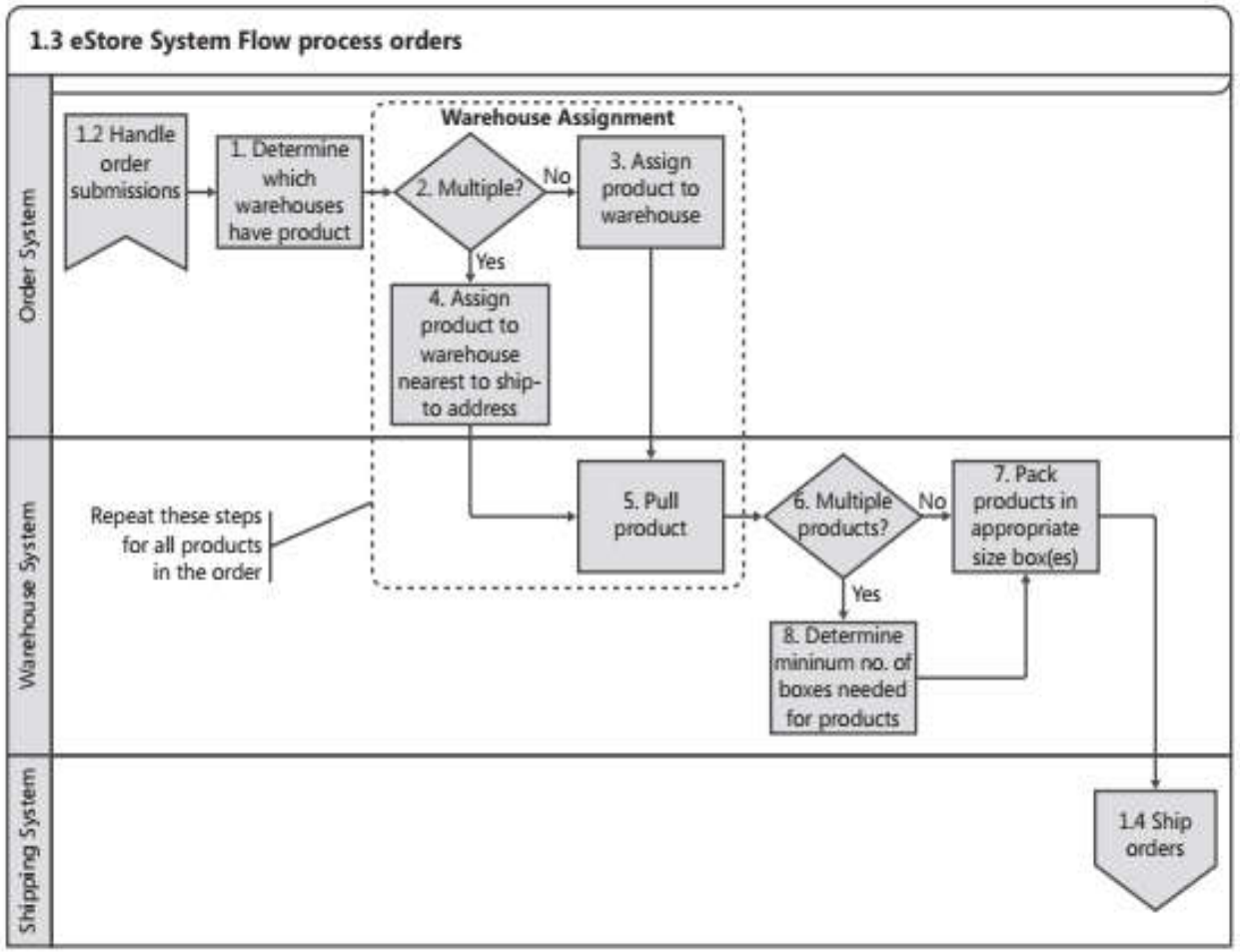
The eStore Business Objectives Model



Ecosystem Map for the eStore



System Flow


1.3 eStore System Flow process orders
Order System
Warehouse System
Shipping System
1.2 Handle order submissions
1. Determine which warehouses have product
Warehouse Assignment
2. Multiple?
No
Yes
3. Assign product to warehouse
4. Assign product to warehouse nearest to ship-to address
Repeat these steps for all products in the order
5. Pull product
6. Multiple products?
No
Yes
7. Pack products in appropriate size box(es)
8. Determine mininum no. of boxes needed for products
1.4 Ship orders