Chapter 15:
Organizing Requirements Information

Objectives

- To understand how to organize requirements information

Introduction

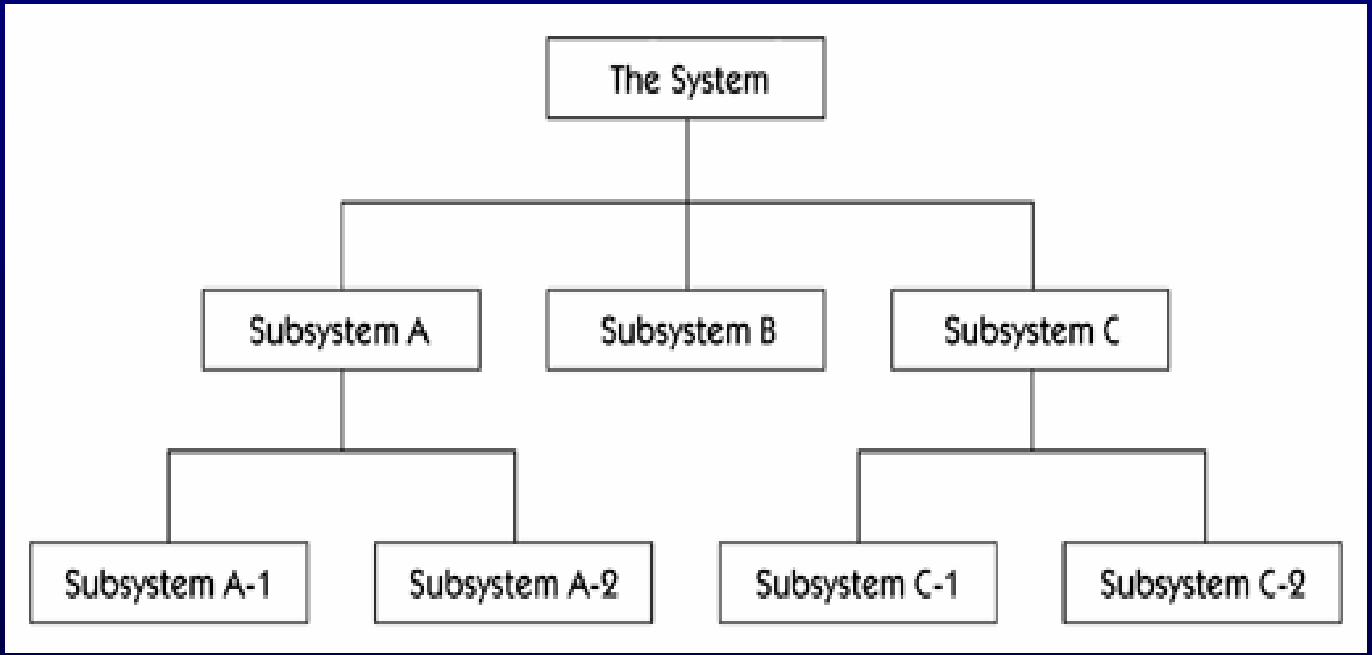
- Inevitable communication problems inherent in a multiple-person effort demand that requirements be captured and documented in a way that they can be reviewed and approved and to which all parties can agree and refer.

- Traditionally large documents, called requirements specifications, have been built to capture and communicate this information.
  - The requirements specification for a system or application describes the external behavior of that system.

Introduction *(Cont'd)*

- Requirements can rarely be defined in a single document or in a single use-case model for that matter. There are a number of reasons.
  - The system may be very complex, and the volume of documentation demands both organizational and interactive access techniques.
  - The system of interest may be a member of a family of related products. No one document can contain all the specifications.
  - The system being constructed may be a subsystem of a larger system and may satisfy only a subset of all the requirements identified.
  - Marketing and business goals need to be separated from the detailed product requirements.
  - Other requirements, perhaps regulatory or legal, may also be imposed on the system, and these requirements may be documented elsewhere.

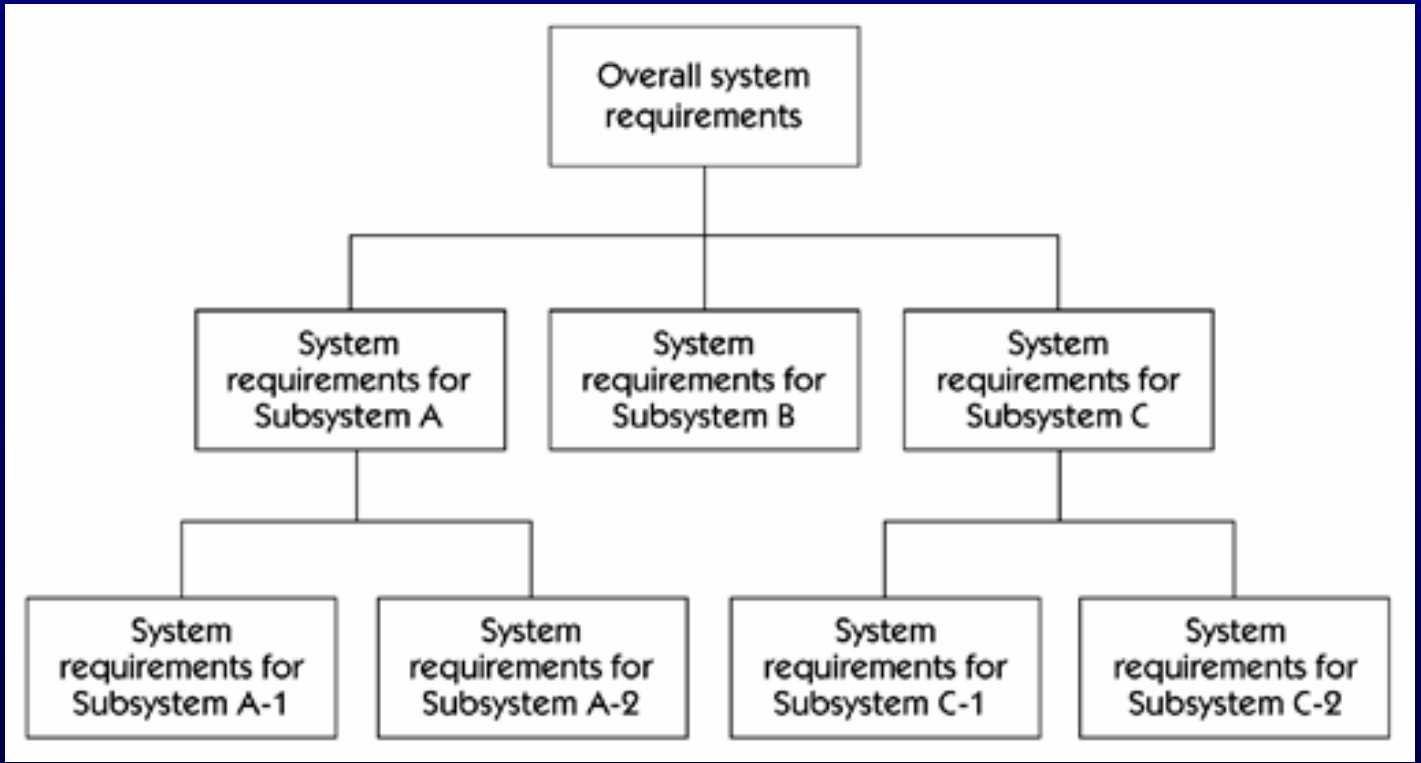
A System of Subsystems



Organizing Requirements of Complex Systems

- A system-level requirements set is created that describes the system as well as the system-level use cases that describe functional behavior, without knowledge of or reference to any of its subsystems.
- Next, requirements are developed for each subsystem.
  - These should describe the external behavior of the subsystem completely, without reference to any of its subsystems. And so on …

Hierarchy of requirements resulting from system design



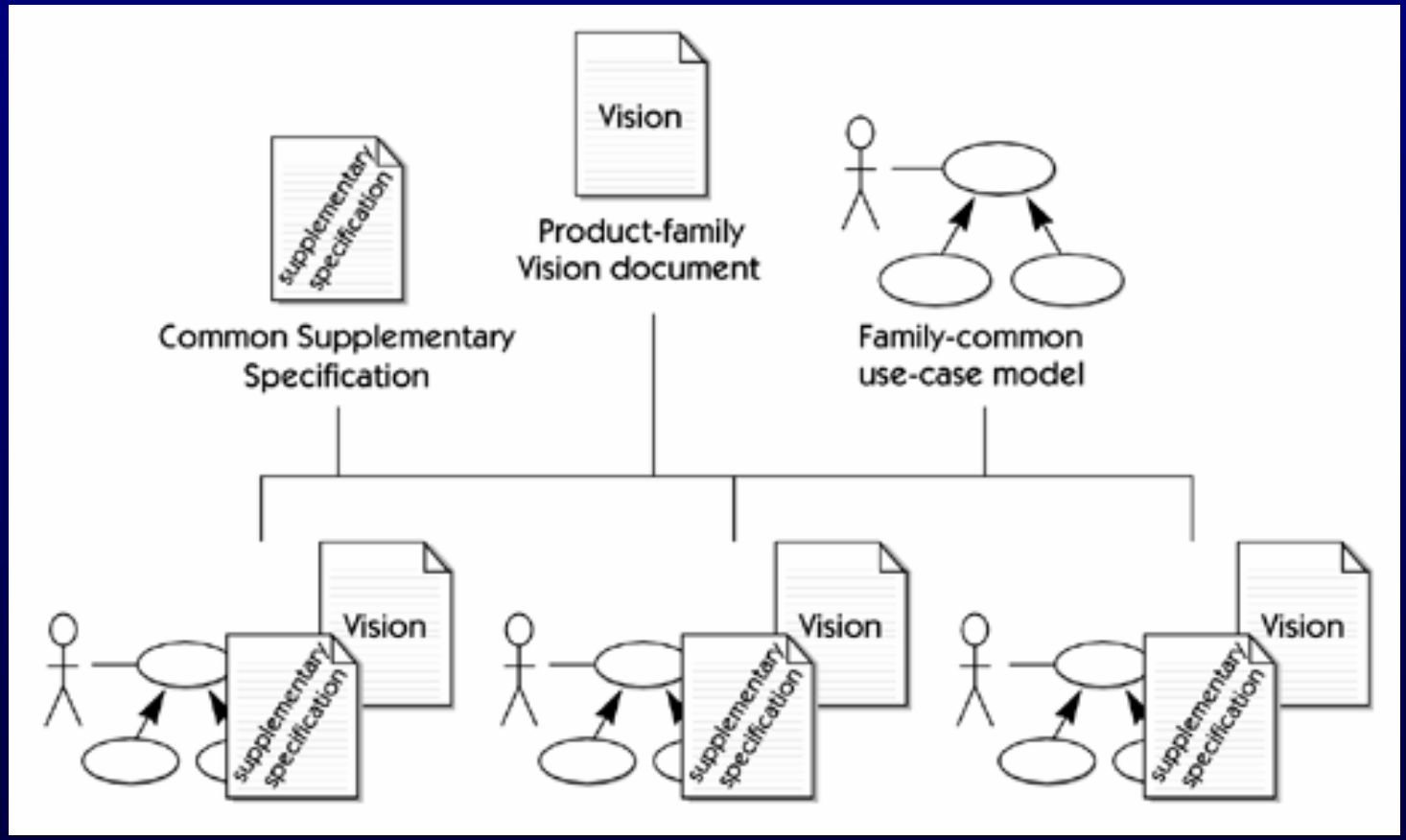
Organizing Requirements for Product Families

- Many industries build sets of closely related products that have much functionality in common, yet each product contains some unique features.
  - Examples: inventory control systems, telephone answering machines, application development tools.

Organizing Requirements for Product Families: Approach

- Develop a product-family Vision document that describes the ways in which the products are intended to work together and the other features that could be shared.
- To better understand the shared-usage model, you might also develop a set of use cases showing how the users will interact with various applications running together.
- Develop a common software requirements set that defines the specific requirements for shared functionality, such as menu structures, common GUIs, and communication protocols.
- For each product in the family, develop a Vision document, supplementary specification, and a use-case model that defines its specific functionality.

Requirements Organization for a Software Product Family



On "Future" Requirements

- During any process of requirements elicitation, requirements will arise that are deemed appropriate for the next release of the product.
- It makes sense to record both current and future requirements but to clearly identify those requirements that are planned for the current release. Reasons:
  - Future requirements represent value-added work products.
  - We will want to produce requirements from them for future releases.
  - The system designers may well have designed the system differently had they known that future requirements of a certain type were desired.

Key Points

- For nontrivial applications, requirements must be captured and recorded in a document, database, model, or tool.
- Different types of projects require different requirements organization techniques.
- Complex systems require that requirements sets be developed for each subsystem.