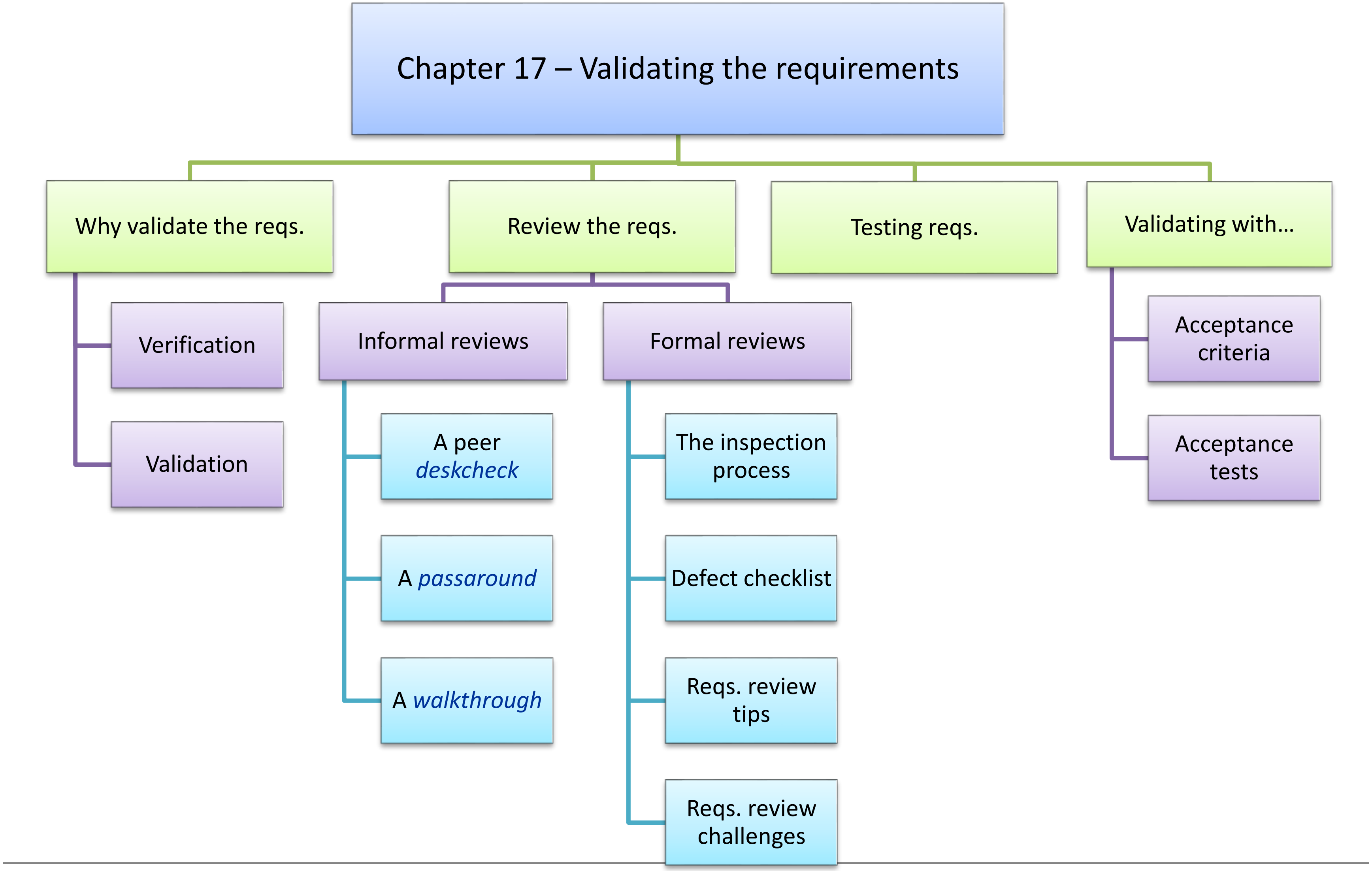
ĐẠI HỌC FPT CẦN THƠ

CHAPTER 18

Requirements reuse

Review chapter 17



Objectives

- Describes several kinds of requirements reuse, identifies some classes of requirements information that have reuse potential in various contexts, and offers suggestions about how to perform requirements reuse.
- It presents some issues around making requirements reusable.
- Concluding with both barriers to effective reuse and success factors that can help organization better take advantage of its existing body of requirements knowledge.

Contents

1. Why reuse requirements?
2. Dimensions of requirements reuse.
3. Types of requirements information to reuse.
4. Common reuse scenarios.
5. Requirement patterns.
6. Tools to facilitate reuse.
7. Making requirements reusable.
8. Requirements reuse barriers and success factors.

Why reuse requirements?

- Faster delivery, lower development costs, consistency both within and across applications, higher team productivity, fewer defects, and reduced rework.
- Reusing trusted requirements can save review time, accelerate the approval cycle, and speed up other project activities, such as testing.
- Reuse can improve your ability to estimate implementation effort if you have data available from implementing the same requirements on a previous project.
- From the user's perspective, requirements reuse can improve functional consistency across related members of a product line or among a set of business applications.
- If the implementation varies in different environments, the requirements might be the same.

Dimensions of requirements reuse

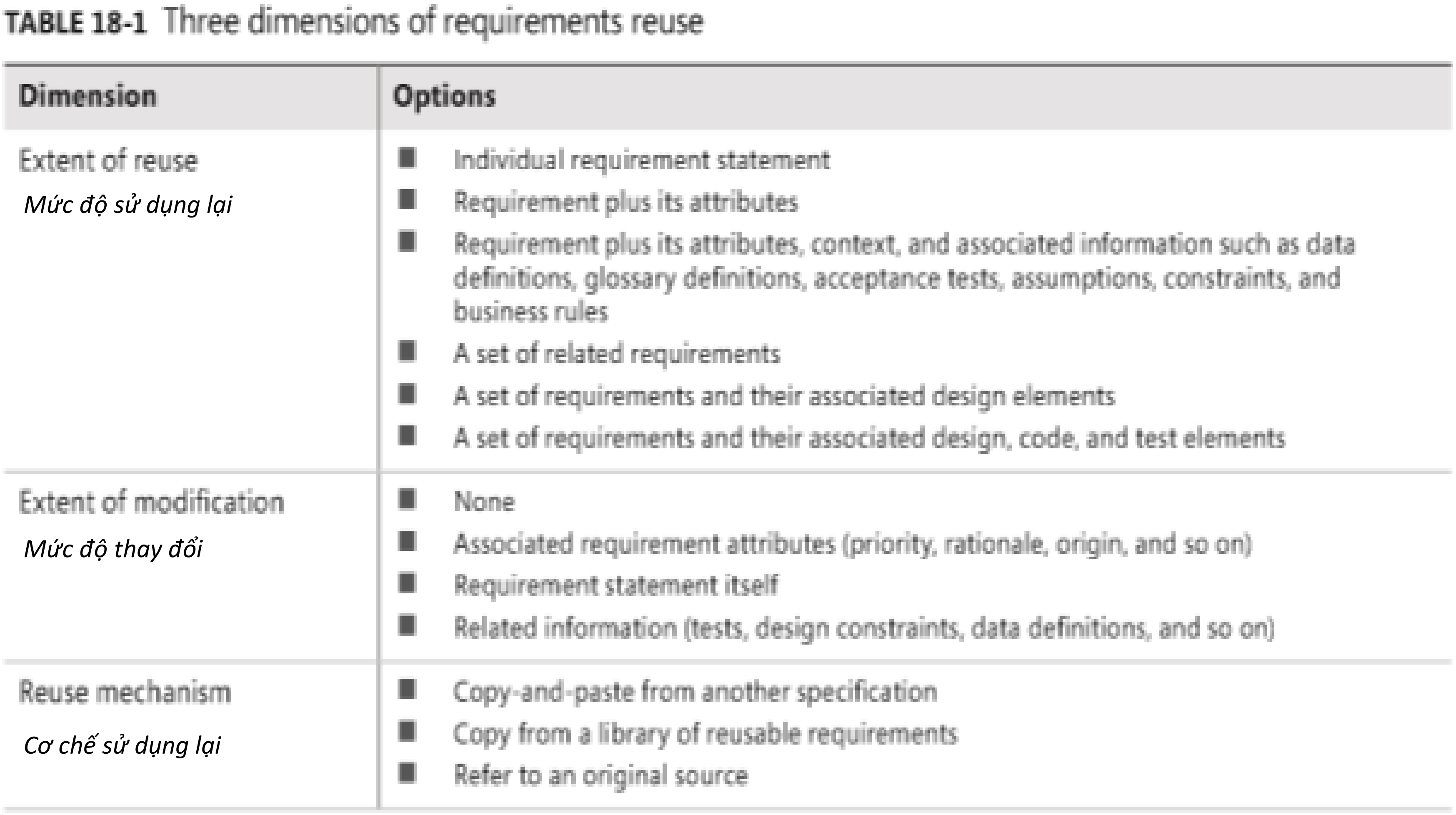


TABLE 18-1 Three dimensions of requirements reuse

Dimension	Options
Extent of reuse *Mức độ sử dụng lại*	■ Individual requirement statement ■ Requirement plus its attributes ■ Requirement plus its attributes, context, and associated information such as data definitions, glossary definitions, acceptance tests, assumptions, constraints, and business rules ■ A set of related requirements ■ A set of requirements and their associated design elements ■ A set of requirements and their associated design, code, and test elements
Extent of modification *Mức độ thay đổi*	■ None ■ Associated requirement attributes (priority, rationale, origin, and so on) ■ Requirement statement itself ■ Related information (tests, design constraints, data definitions, and so on)
Reuse mechanism *Cơ chế sử dụng lại*	■ Copy-and-paste from another specification ■ Copy from a library of reusable requirements ■ Refer to an original source

Types of requirements information to reuse

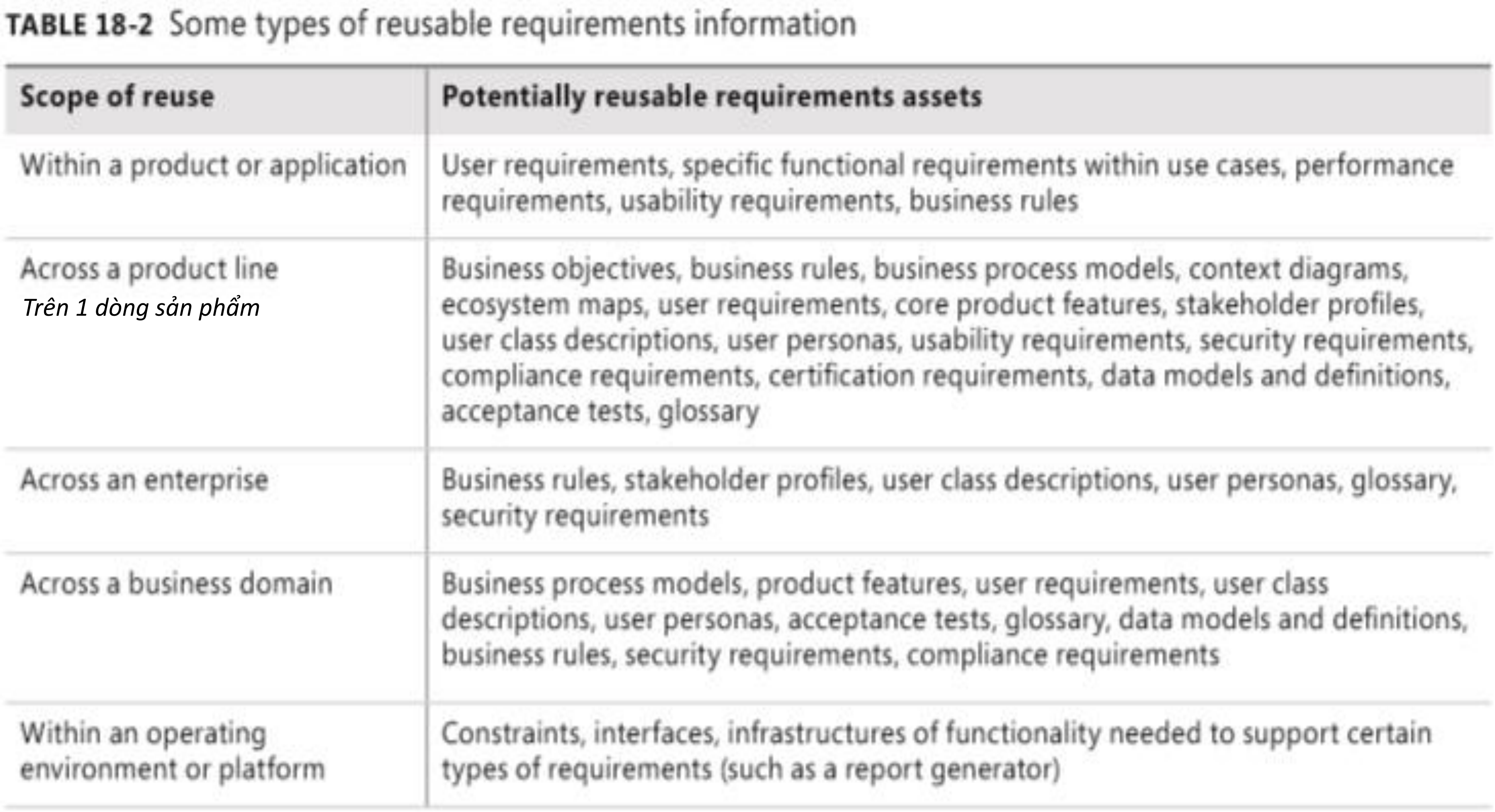


TABLE 18-2 Some types of reusable requirements information

Scope of reuse	Potentially reusable requirements assets
Within a product or application	User requirements, specific functional requirements within use cases, performance requirements, usability requirements, business rules
Across a product line *Trên 1 dòng sản phẩm*	Business objectives, business rules, business process models, context diagrams, ecosystem maps, user requirements, core product features, stakeholder profiles, user class descriptions, user personas, usability requirements, security requirements, compliance requirements, certification requirements, data models and definitions, acceptance tests, glossary
Across an enterprise	Business rules, stakeholder profiles, user class descriptions, user personas, glossary, security requirements
Across a business domain	Business process models, product features, user requirements, user class descriptions, user personas, acceptance tests, glossary, data models and definitions, business rules, security requirements, compliance requirements
Within an operating environment or platform	Constraints, interfaces, infrastructures of functionality needed to support certain types of requirements (such as a report generator)

Common reuse scenarios

Let’s look at some scenarios where requirements reuse offers good potential.

- Software product lines: Anytime you’re creating a set of products in a family—a software product line—those products will have a lot of functionality in common. Sometimes you’re producing variations (biến thể) of a base product for different customers or markets.
- Reengineered and replacement systems: Reengineered and replacement systems always reuse some requirements from the original incarnation. Often, you can harvest (thu hoạch) business rules that were embedded in an old system and reuse them on future projects, updating them as necessary.
- Other likely reuse opportunities: Table 18-3 (next slide) lists several other situations in which reusing requirements information is common. If you previously worked on a project similar to the current one, consider whether you can use any artifacts from the earlier project again.

Common reuse scenarios

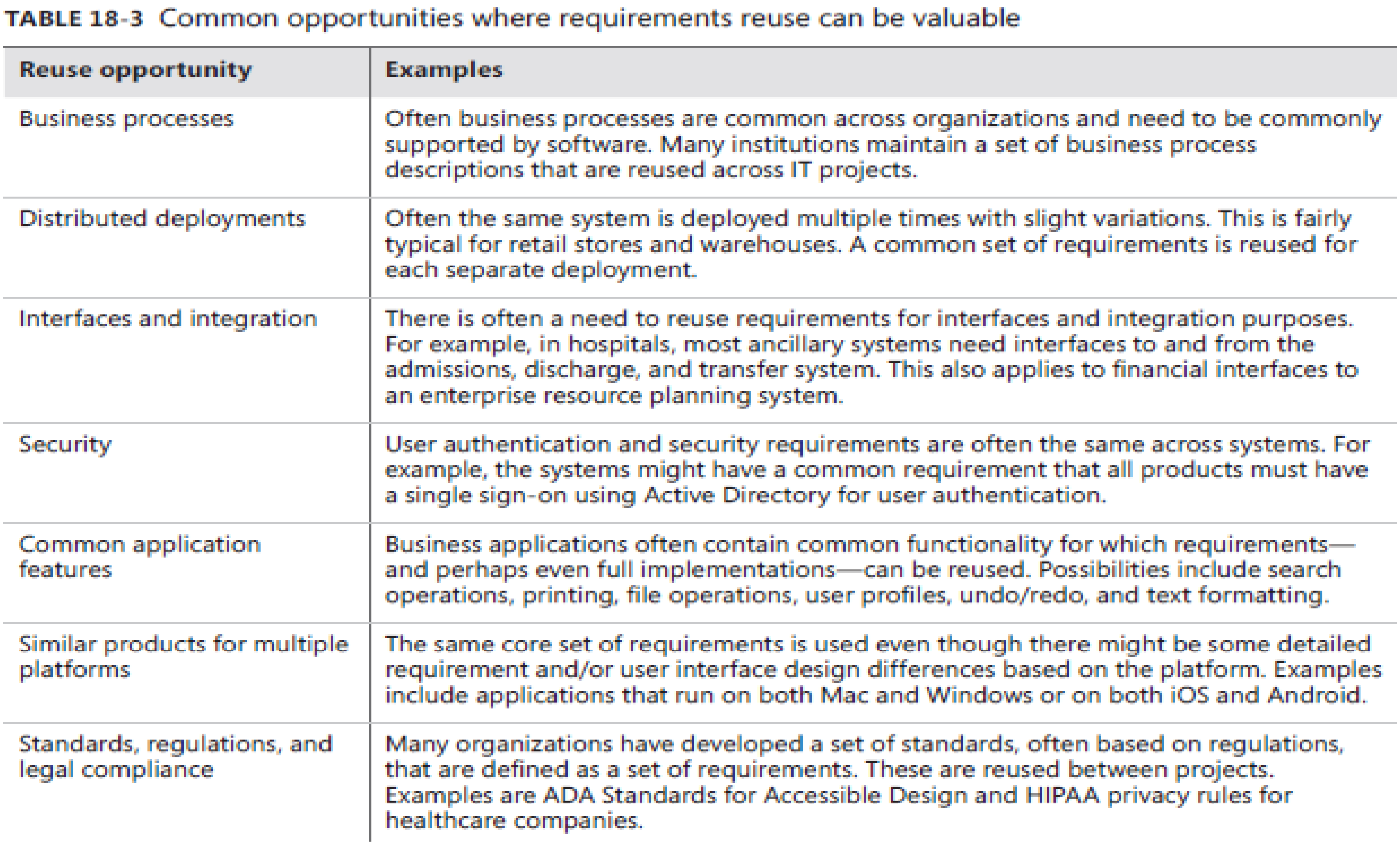


TABLE 18-3 Common opportunities where requirements reuse can be valuable

Reuse opportunity	Examples
Business processes	Often business processes are common across organizations and need to be commonly supported by software. Many institutions maintain a set of business process descriptions that are reused across IT projects.
Distributed deployments	Often the same system is deployed multiple times with slight variations. This is fairly typical for retail stores and warehouses. A common set of requirements is reused for each separate deployment.
Interfaces and integration	There is often a need to reuse requirements for interfaces and integration purposes. For example, in hospitals, most ancillary systems need interfaces to and from the admissions, discharge, and transfer system. This also applies to financial interfaces to an enterprise resource planning system.
Security	User authentication and security requirements are often the same across systems. For example, the systems might have a common requirement that all products must have a single sign-on using Active Directory for user authentication.
Common application features	Business applications often contain common functionality for which requirements—and perhaps even full implementations—can be reused. Possibilities include search operations, printing, file operations, user profiles, undo/redo, and text formatting.
Similar products for multiple platforms	The same core set of requirements is used even though there might be some detailed requirement and/or user interface design differences based on the platform. Examples include applications that run on both Mac and Windows or on both iOS and Android.
Standards, regulations, and legal compliance	Many organizations have developed a set of standards, often based on regulations, that are defined as a set of requirements. These are reused between projects. Examples are ADA Standards for Accessible Design and HIPAA privacy rules for healthcare companies.

Requirement patterns

- *A requirement pattern* offers a systematic approach to specifying a particular type of requirement.
- A pattern defines a template with categories of information for each of the common types of requirements a project might encounter.
- A requirement pattern contains several sections:
  - Guidance: Basic details about the pattern, including related patterns, situations to which it is (and is not) applicable, and a discussion of how to approach writing a requirement of this type.
  - Content: A detailed explanation of the content that such a requirement ought to (nên) convey, item by item.
  - Template: A requirement definition with placeholders wherever variable pieces of information need to go.
  - Examples: One or more illustrative requirements of this type.
  - Extra requirements: Additional requirements that can define certain aspects of the topic, or an explanation of how to write a set of detailed requirements.
  - Considerations for development and testing: Factors for developers to keep in mind when implementing a requirement of the type specified by the pattern, and factors for testers to keep in mind when testing such requirements.

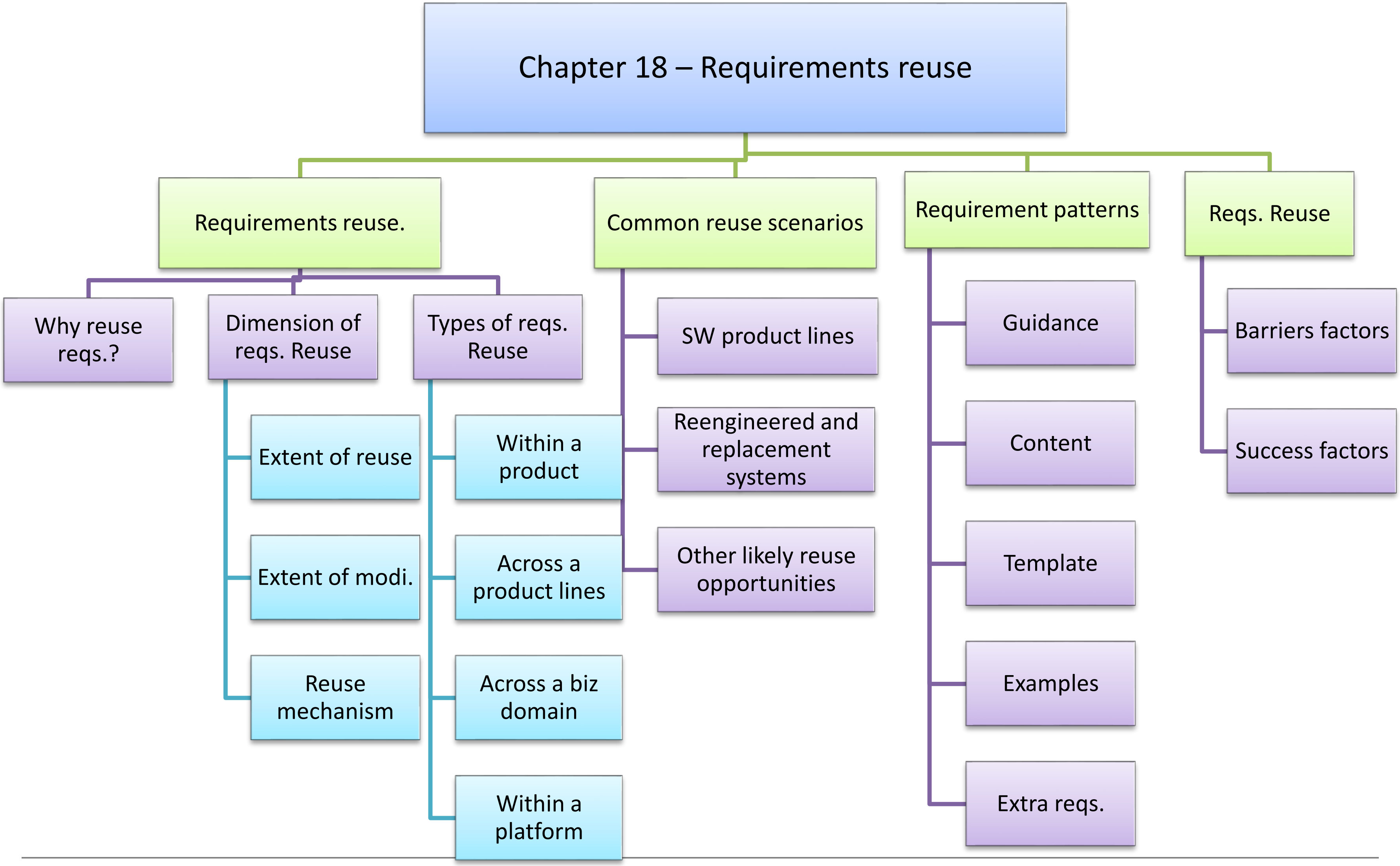
Requirements reuse barriers and success factors

- Reuse barriers
  - Missing or poor requirements: A common barrier is that the requirements developed on previous projects weren't documented, so it's impossible to reuse them. And even if you find a relevant requirement, it might be badly written, incomplete, or a poor fit for your present circumstances.
  - NIH and NAH: Two barriers to reuse are NIH and NAH syndromes (hội chứng). NIH means "not invented here." Some people are reluctant to reuse requirements from another organization or generic requirements found in a public collection. NAH, or "not applicable here," when practitioners protest that a new process or approach does not apply to their project or organization.
  - Writing style: Requirements written in natural language are ambiguities, missing information, and hidden assumptions. Requirements that have embedded design constraints will offer little opportunity for reuse in a different environment. These issues reduce their reuse potential.
  - Inconsistent organization: It can be difficult to find requirements to reuse because authors organize their requirements in many different ways: by project, process flow, business unit, product feature, category, subsystem or component, and so forth.
  - Project type: Requirements that are tightly coupled (gắn chặt) to specific implementation environments or platforms are less likely (ít có khả năng) to generate reusable requirements or to benefit from an existing pool of requirements knowledge.
  - Ownership: If you're developing a software product for a specific customer, its requirements are likely the proprietary intellectual property of the customer.

Requirements reuse barriers and success factors

- Reuse success factors
  - Repository: You can't reuse something if you can't find it. A collection of requirements stored in a requirements management tool that can be searched across projects.
  - Quality: No one wants to reuse junk (tạm nham). Potential reusers need confidence in the quality of the information.
  - Interactions: Requirements often have logical links or dependencies on each other. Reused requirements must conform to existing business rules, constraints, standards, interfaces, and quality expectations.
  - Terminology: Establishing common terminology and definitions across your projects will be helpful for reusability. Terminology variations won't prevent you from reusing requirements.
  - Organizational culture: The individuals, project teams, and organizations that practice reuse most effectively are likely to enjoy (có khả năng tận hưởng) the highest productivity. In a reuse culture, BAs look at the reusable requirements repository before creating their own requirements.

Review chapter 18



THE END

THANK YOU!