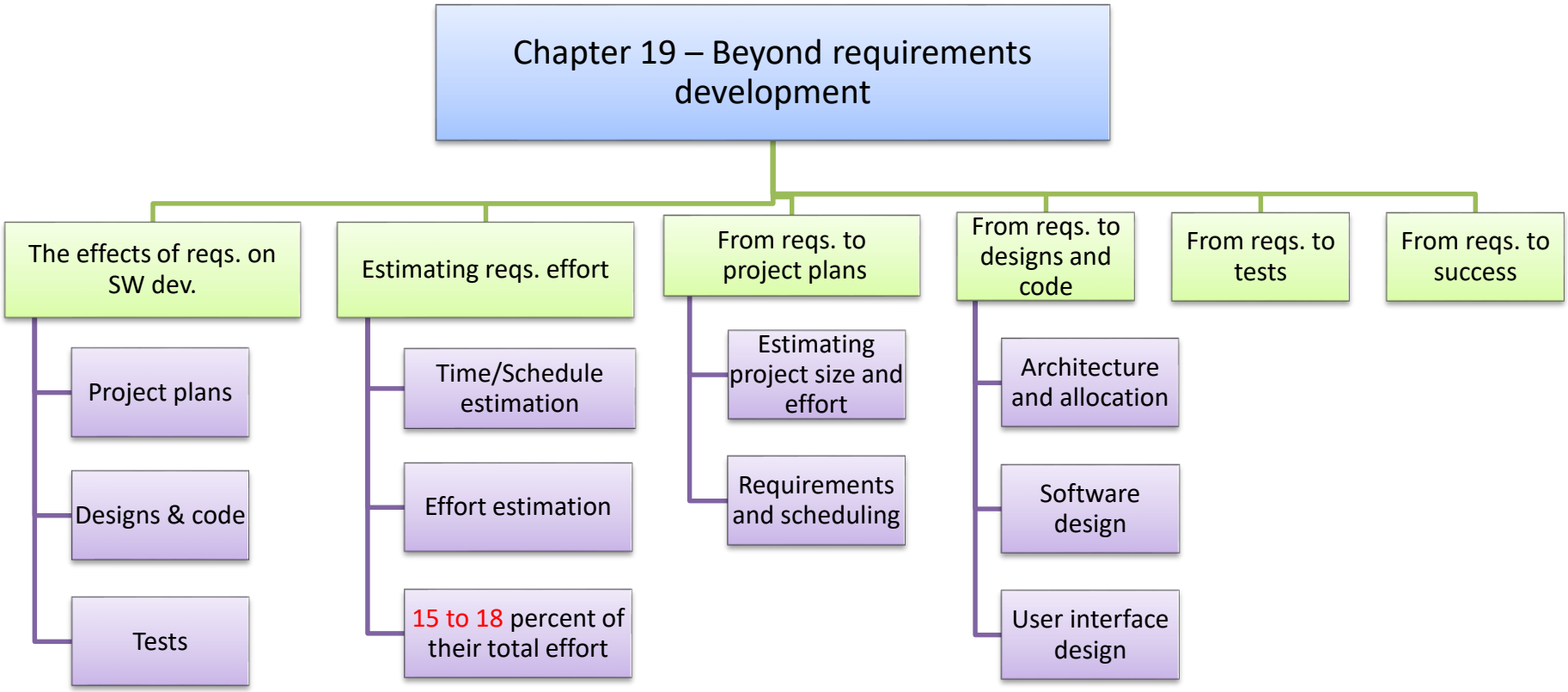
FPT UNIVERSITY

ĐẠI HỌC FPT CẦN THƠ

CHAPTER 20

Agile projects

Review chapter 19



Objectives

- Describes the characteristics of agile approaches as they relate to the requirements activities for a software project, the major adaptations of traditional requirements practices for an agile project, and a road map of where to find more detailed guidance.

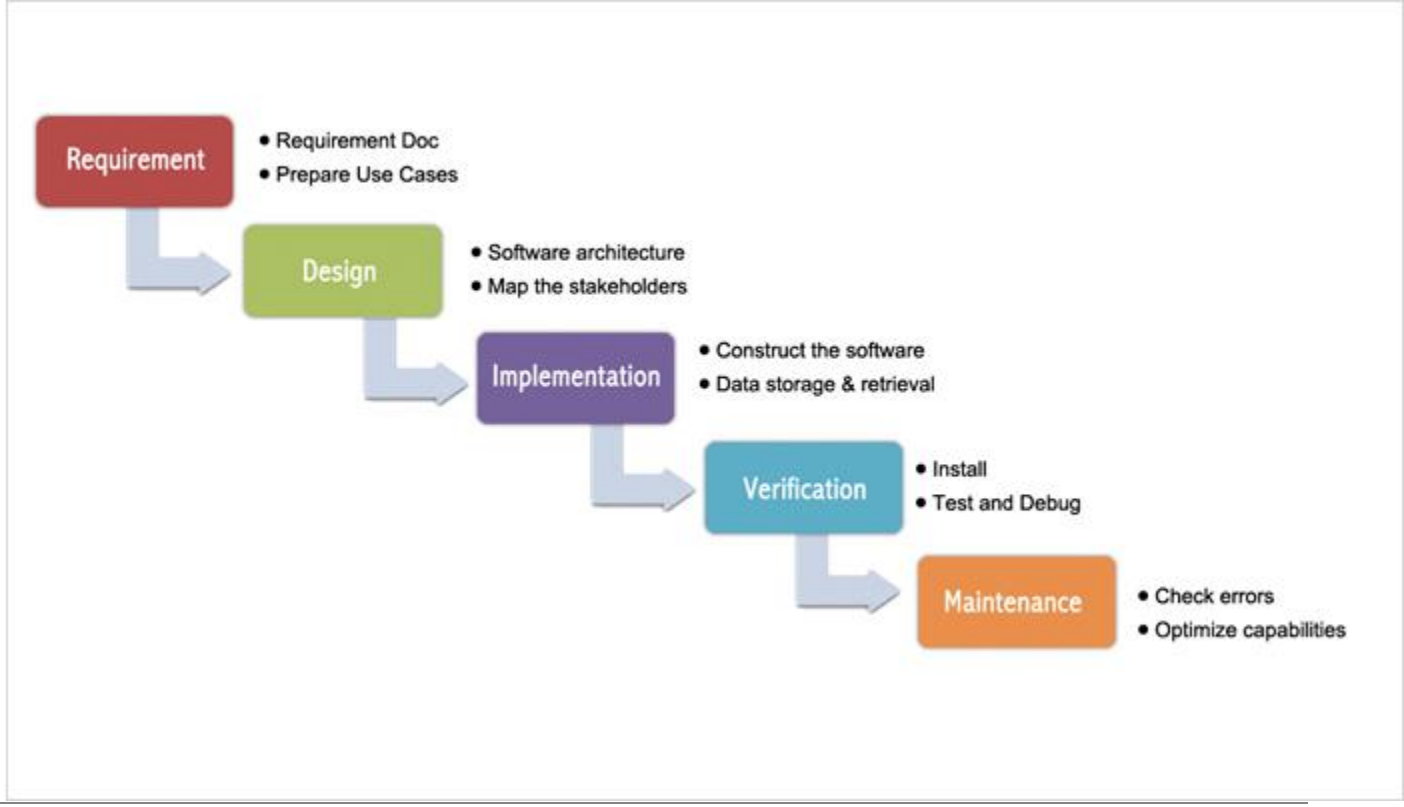
- Student should recognize what and how they have to do in every stage of agile projects

Contents

1. Limitations of the waterfall.
2. The agile development approach.
3. Essential aspects of an agile approach to requirements.
4. Adapting requirements practices to agile projects.
5. Transitioning to agile: Now what?

Limitations of the waterfall

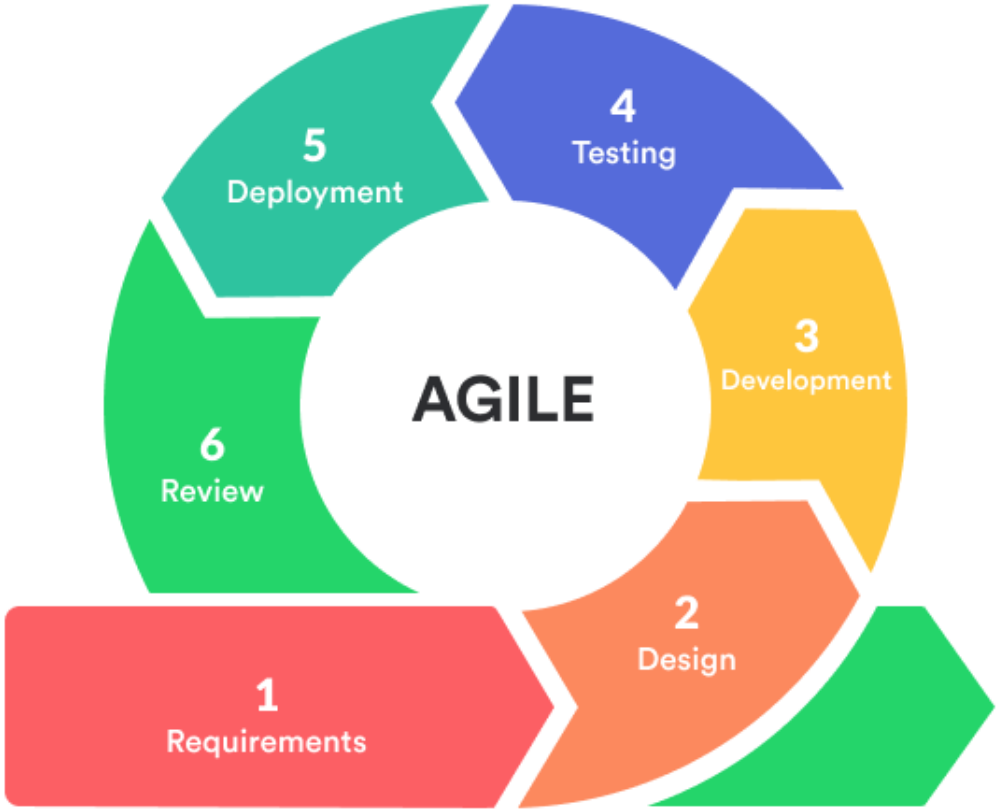
- What it is?
  - Organizations often think of a waterfall development process as involving a linear sequence of activities, where project teams fully specify the requirements, then create designs, then write code, and finally test the solution.



Limitations of the waterfall

- What are advantages?
  - The team can catch any flaws (nắm bắt những thiếu sót) in the application's requirements and design early on rather than during construction, testing, or maintenance, when fixing an error is much more costly.
  - If the requirements are correct up front, it is easy to allocate budget and resources, to measure progress, and to estimate an accurate completion date.
- What are disadvantages?
  - Large projects that use a waterfall approach are often delivered late, lack necessary features, and fail to meet users' expectations.
  - Stakeholders often change their requirements during the course of a long project, and projects struggle (tìm cách tồn tại) when the software development teams cannot respond to these changes effectively.
- When we use waterfall development?
  - For the projects, if the requirements are correct up front, it is easy to allocate budget and resources, to measure progress, and to estimate an accurate completion date.

The agile development approach



The agile development approach

- What is agile development approach?
  - Agile methods focus on iterative and incremental development, breaking the development of software into short cycles called iterations (or, in the agile method known as Scrum (nghĩa đen tụ nhau giành bóng), "sprints" (chạy hết tốc lực)).
  - Iterations can be as short as one week or as long as a month. During each iteration, the development team adds a small set of functionality based on priorities established by the customer, tests it to make sure it works properly, and validates it with acceptance criteria established by the customers.
- What are advantages?
  - Minimum the defects at the final product, because of subsequent increments modify what already exists, enrich the initial features, add new ones, and correct defects that were discovered.
  - Ongoing customer participation enables the team to spot (phát hiện) problems and changes in direction early (sớm thay đổi đúng hướng), thereby guiding developers to adjust their course before they are too far down the wrong path.
- When we use agile development approach?
  - Agile development approach can applied to all SW projects.

Essential aspects of an agile approach to requirements

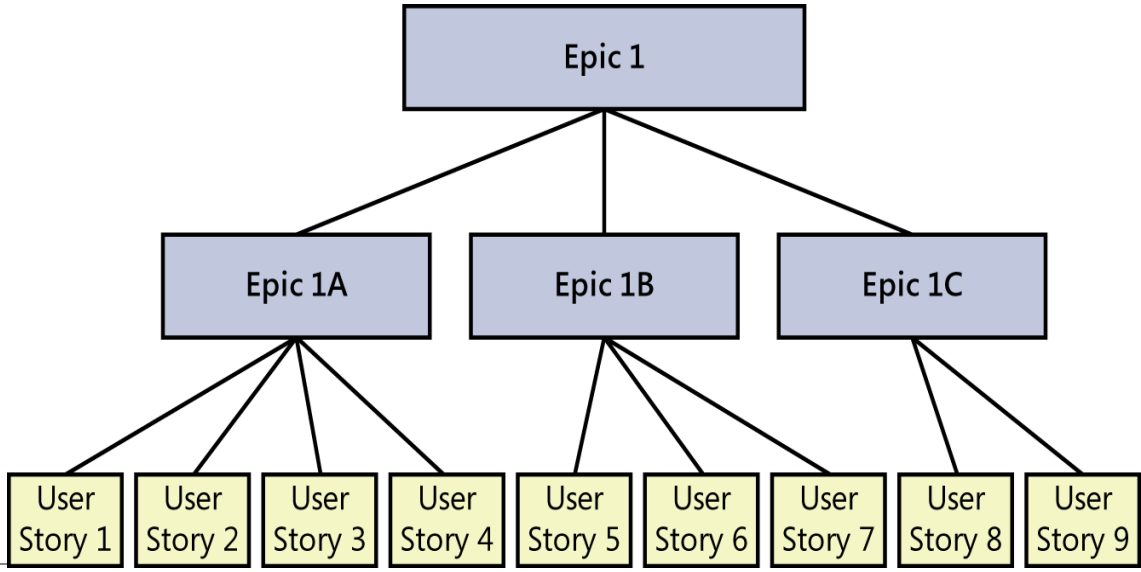
- Customer involvement:
  - Collaborating with customers on software development projects always increases the chances of project success. This is true for waterfall projects as well as for agile projects.
  - On waterfall projects, customers typically dedicate considerable time up front, helping the BA understand, document, and validate requirements. During the later phases, there is generally little customer involvement, which makes it difficult for a project to adapt to changing customer needs.
  - On agile projects, customers (or a product owner) are engaged continuously throughout the project. During an initial planning iteration on some agile projects, customers work with the project team to identify and prioritize user stories that will serve as the preliminary road map for the development of the product.
- Documentation detail:
  - Because developers have little interaction with customers in the later phases on waterfall projects, the requirements must specify system behavior, data relationships, and user experience expectations in considerable detail.
  - The close collaboration of customers with developers on agile projects generally means that requirements can be documented in less detail than on traditional projects.

Essential aspects of an agile approach to requirements

- The backlog and prioritization:
  - The product backlog on an agile project contains a list of requests for work that the team might perform (IIBA 2013). Product backlogs typically are composed of user stories to be built, and defects to be corrected. Each project should maintain only one backlog (Cohn 2010).
  - Prioritization of the backlog is an ongoing activity to select which work items go into upcoming (sắp tới) iterations and which items are discarded (loại bỏ) from the backlog.
- Timing:
  - Detailed requirements are not documented all at once at the beginning of an agile project. Instead, high-level requirements, typically in the form of user stories, are elicited to populate (đưa vào) a product backlog early in a project for planning and prioritization.
  - User stories are allocated to specific iterations for implementation, and the details for each story are further clarified during that iteration.

Essential aspects of an agile approach to requirements

- Epics, user stories, and features:
  - User stories are sized so as to be fully implementable in a single iteration.
  - Mike Cohn (2010) defines an ***epic*** as being a user story that is too large to fully implement in a single iteration. Because epics span iterations, they must be split into sets of smaller stories.
  - Sometimes epics are large enough that they must be subdivided into multiple epics, each of which is then split into multiple stories until each resulting story can be reliably estimated and then implemented and tested within a single iteration.
  - A feature is a grouping of system capabilities that provides value to a user. In the context of an agile project, features could encompass an individual user story, multiple user stories, an individual epic, or multiple epics.



Essential aspects of an agile approach to requirements

- Expect change:
  - The biggest adaptation (thích ứng) that BAs need to make when a requirement change arises on an agile project is to say not, “Wait, that’s out of scope”, but rather, “Okay, let’s talk about the change.”
  - This encourages customer collaboration to create or change user stories and prioritize each change request against everything else that’s already in the backlog.
  - Change also includes removing items from scope. Items can be removed from an iteration’s scope for various reasons, including the following:
    - Implementation issues prevent an item from being completed within the current time frame.
    - Issues discovered by product owners or during testing make the implementation of a particular story unacceptable.
    - Higher-priority items need to replace less important ones that were planned for an iteration.

Adapting requirements practices to agile projects

Many other chapters in this book address how to adapt the practices described in the chapter to suit an agile project.

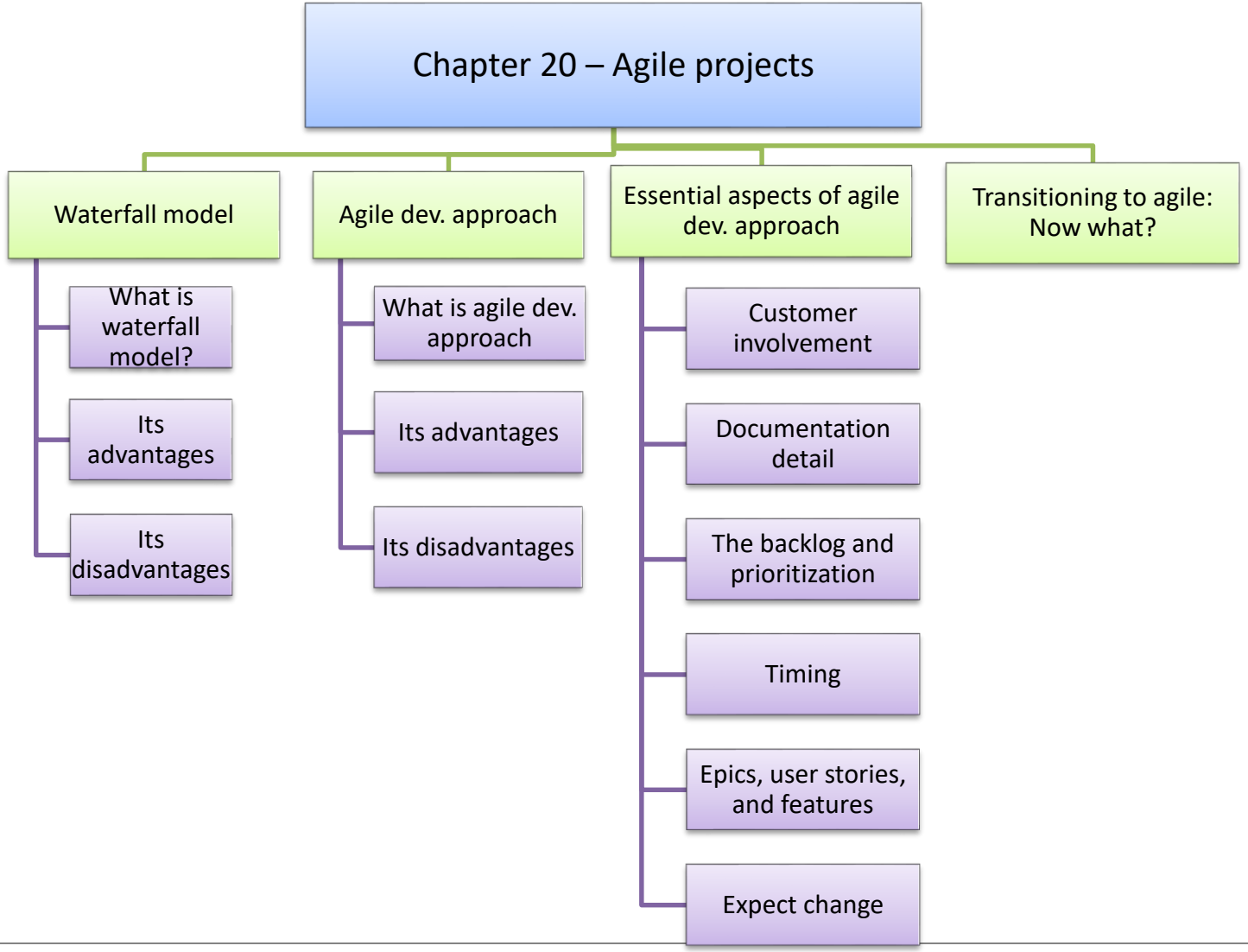
TABLE 20-1 A road map to chapters that address agile development topics

Chapter	Topic
Chapter 2, "Requirements from the customer's perspective"	Reaching agreement on requirements
Chapter 4, "The business analyst"	The BA's role on agile projects and who is responsible for the requirements artifacts created
Chapter 5, "Establishing the business requirements"	Setting and managing the vision and scope
Chapter 6, "Finding the voice of the user"	User representation
Chapter 8, "Understanding user requirements"	User stories
Chapter 10, "Documenting the requirements"	Specifying requirements for agile development
Chapter 12, "A picture is worth 1024 words"	Modeling on agile projects
Chapter 14, "Beyond functionality"	Identifying quality attributes, especially those needed up front for architecture and design
Chapter 15, "Risk reduction through prototyping"	Agile projects and evolutionary prototyping
Chapter 16, "First things first: Setting requirement priorities"	Prioritization on agile projects
Chapter 17, "Validating the requirements"	Acceptance criteria and acceptance tests
Chapter 27, "Requirements management practices"	Managing requirements on agile projects through backlogs and burndown charts
Chapter 28, "Change happens"	Managing change on agile projects

Transitioning to agile: Now what?

- If you're a business analyst who is new to agile development methods, don't worry: most of the practices you already use will still apply. After all, both agile and traditional project teams need to understand the requirements for the solutions they build.
- Following are a few suggestions to help you make the conversion to an agile approach:
  - Determine what your role is on the team: Everyone has a role in the project, but encourage all team members to focus on the goals of the project, not their individual roles or titles.
  - Read a book on the agile product owner role so you understand user stories, acceptance tests, backlog prioritization,... One suggested book is *Agile Product Management with Scrum* (Pichler 2010).
  - Identify suggested agile practices that will work best in your organization.
  - Implement a small project first as a pilot for agile methods, or implement only a few agile practices on your next project.
  - If you are new to agile, bring in an experienced coach for three or four iterations to help you avoid the temptation (cám dỗ) to revert to the historical practices with which you are comfortable.

Review chapter 20



THE END
THANK YOU!