FPT UNIVERSITY

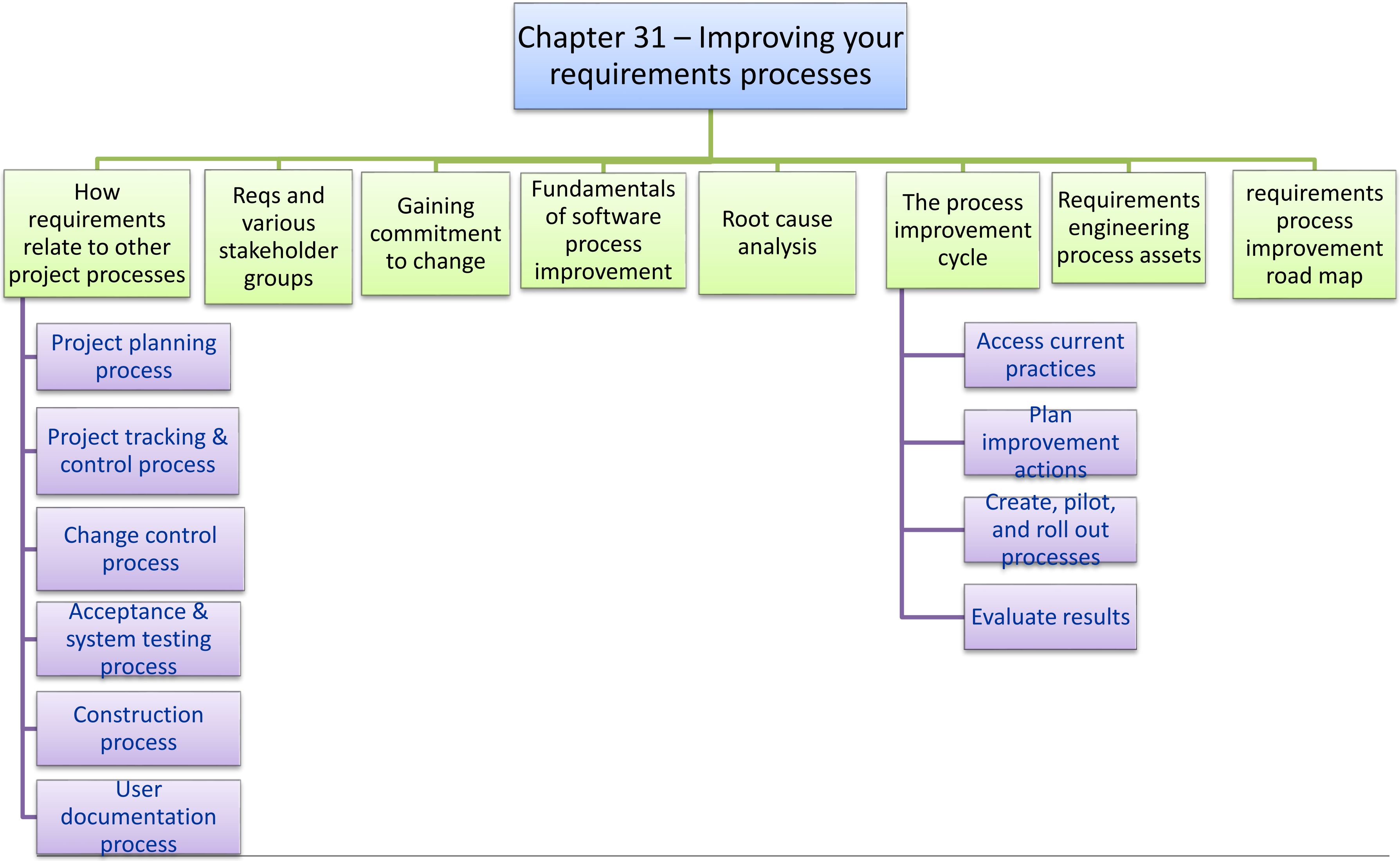
ĐẠI HỌC FPT CẦN THƠ

Chapter 32

Software requirements and risk management

cantho.fpt.edu.vn

Review chapter 31



Chapter 31 – Improving your requirements processes

- How requirements relate to other project processes
  - Project planning process
  - Project tracking & control process
  - Change control process
  - Acceptance & system testing process
  - Construction process
  - User documentation process
- Reqs and various stakeholder groups
- Gaining commitment to change
- Fundamentals of software process improvement
- Root cause analysis
- The process improvement cycle
  - Access current practices
  - Plan improvement actions
  - Create, pilot, and roll out processes
  - Evaluate results
- Requirements engineering process assets
- requirements process improvement road map

Objectives

- Presents a brief tutorial on software risk management and a number of risk factors that can raise their ugly heads (giúp ích) during requirements engineering activities.
- Student should understand that potential problems might have an adverse impact on the project’s cost, schedule, or technical success; the product’s quality; or the team’s effectiveness.
- Student could identify a list of potential requirements risks facing in their project.

Contents

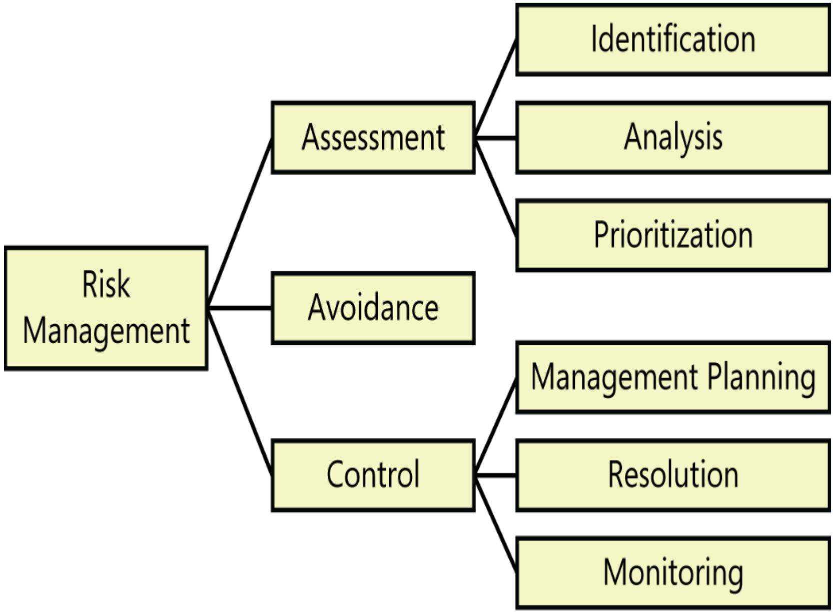
- Fundamentals of software risk management.
- Requirements-related risks.
- Risk management is your friend.

What's risk and risk management?

- A *risk* is a condition that could cause some loss or otherwise threaten the success of a project.
- This condition causes actually problems. These potential problems might have an adverse impact on the project's cost, schedule, or technical success; the product's quality; or the team's effectiveness.
- *Risk management* is the process of identifying, evaluating, and controlling risks before they harm your project.
- Because no one can predict the future with certainty, risk management is used to minimize the likelihood (khả năng xảy ra) or impact of potential problems. Risk management means dealing with a concern before it becomes a crisis.
- Typical *requirements risks* include misunderstanding the requirements, inadequate user involvement, uncertain or changing project scope and objectives, and continually changing requirements. Project managers can control requirements risks only through collaboration with customers and other stakeholders.

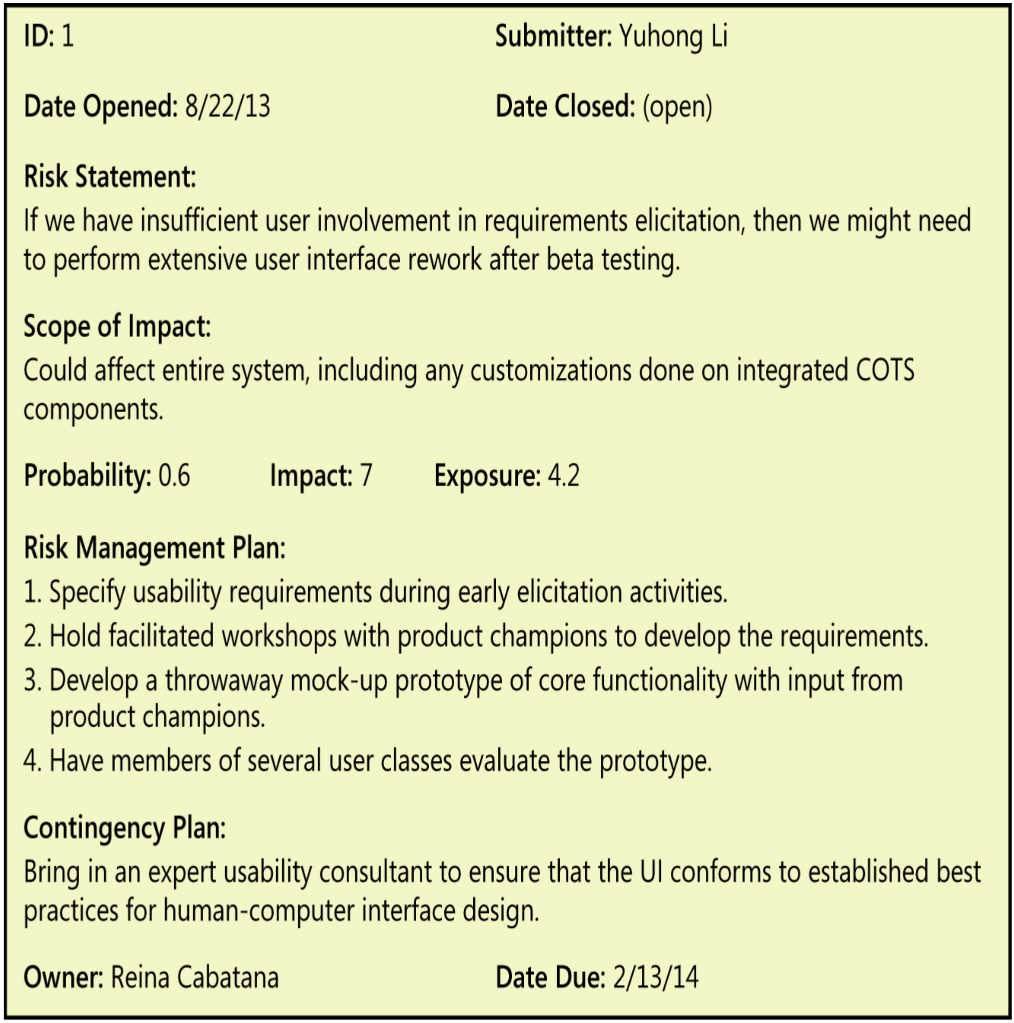
Fundamentals of software risk management

- Elements of risk management:
  - Risk management involves (liên quan) the application of tools and procedures to contain project risk within acceptable limits. It provides a standard approach to identify and document risk factors, evaluate their potential severity, and propose strategies for mitigating them.
  - Risk management includes the activities shown in Figure 32-1:
  - *Risk assessment* is the process of examining a project to identify potential threats.
    - Facilitate *risk identification* with lists of common risk factors.
    - During *risk analysis*, you'll examine the potential consequences of specific risks to your project.
    - *Risk prioritization* helps you focus on the most severe risks by assessing the potential *risk exposure* from each.
  - *Risk avoidance* is one way to deal with a risk: don't do the risky thing.
  - Most of the time you'll have to perform *risk control* activities to manage the top-priority risks you identified.
    - *Risk management planning* produces a plan for dealing with each significant risk, including mitigation approaches, contingency plans, owners, and timelines.
    - *Risk resolution* involves executing the plans for mitigating each risk.
    - Track your progress toward resolving each risk item through *risk monitoring*.



Fundamentals of software risk management

- Documenting project risks:
  - Use a *condition-consequence* format when you document risk statements. That is, state the risk condition that you are concerned about, followed by the potential adverse (bất lợi) outcome—the consequence—from that condition.
  - Figure 32-3 illustrates a risk that the Chemical Tracking System team leaders discussed at the beginning of this chapter. The team estimated the probability and impact on the basis of their previous experience.



ID: 1 **Submitter:** Yuhong Li

Date Opened: 8/22/13 **Date Closed:** (open)

Risk Statement:
If we have insufficient user involvement in requirements elicitation, then we might need to perform extensive user interface rework after beta testing.

Scope of Impact:
Could affect entire system, including any customizations done on integrated COTS components.

Probability: 0.6 **Impact:** 7 **Exposure:** 4.2

Risk Management Plan:

1. Specify usability requirements during early elicitation activities.
2. Hold facilitated workshops with product champions to develop the requirements.
3. Develop a throwaway mock-up prototype of core functionality with input from product champions.
4. Have members of several user classes evaluate the prototype.

Contingency Plan:
Bring in an expert usability consultant to ensure that the UI conforms to established best practices for human-computer interface design.

Owner: Reina Cabatana **Date Due:** 2/13/14

FIGURE 32-3 *Sample risk item from the Chemical Tracking System.*

Fundamentals of software risk management

- Planning for risk management:
  - For a small project, you can include your plans for controlling risks in the software project management plan. A large project should write a separate risk management plan that spells out (nêu rõ) the approaches it intends to take to identify, evaluate, document, and track risks.
  - This plan should include the roles and responsibilities for the risk management activities.
  - Many projects appoint a project risk manager to be responsible for the things that could go wrong.
  - Establish a rhythm of periodic risk monitoring.
  - Keep the 10 or so risks that have the highest risk exposure visible, and track the effectiveness of your mitigation approaches regularly.
  - When a mitigation action is completed, reevaluate the probability and impact for that risk item and then update the risk list.

Requirements-related risks

The risk factors described on the following pages are organized by the five requirements engineering subdisciplines (phân chi) of elicitation, analysis, specification, validation, and management. Techniques are suggested that can reduce each risk's probability or impact.

- Requirements elicitation: Numerous factors can conspire (âm mưu) to hamper (cản trở) your requirements elicitation efforts. Following are several areas of potential elicitation risk and suggestions for how to avoid them.
  - Product vision and project scope: Scope creep → Early in the project, write a vision and scope document that contains your business requirements, and use it to guide decisions about new or modified requirements.
  - Time spent on requirements development: Tight project schedules often pressure managers and customers into glossing over (che đậy) the requirements because they believe that if the developers don't start coding immediately, they won't finish on time.

Requirements-related risks

- Customer engagement: Insufficient customer involvement during the project increases the chance of an expectation gap. → Identify stakeholders, customers, and user classes early in the project.
- Completeness and correctness of requirements specifications: Elicit user requirements that map to business requirements to ensure that the solution will deliver what the customers really need.
- Requirements for innovative products: It's easy to misgauge (đánh giá sai) market response to products. → Emphasize market research, and use focus groups (nhóm tập trung) to obtain early and frequent customer feedback about your innovative product visions.

Requirements-related risks

- Defining nonfunctional requirements: Because of the emphasis on product functionality, it’s easy to neglect nonfunctional requirements → Ask customers about quality characteristics such as performance, usability, security, and reliability. Document these nonfunctional requirements and their acceptance criteria as precisely as you can.
- Customer agreement on requirements: If your customers don’t agree on what you should build, they will be unhappy with the result → Determine who the primary customers are, and use the product champion approach to get adequate customer representation and involvement. Make sure you’re relying on the right people for making decisions about requirements.
- Unstated requirements: Customers often hold implicit expectations that are neither communicated nor documented. → Try to identify any assumptions the customers might be thinking. Use open-ended questions to encourage customers to share more of their thoughts, wishes, ideas, information, and concerns than you might otherwise hear.

Requirements-related risks

- Existing product used as the requirements reference: Requirements development might not be deemed (được coi) important on enhancement or replacement projects. → Developers are sometimes told to use the existing product as their source for requirements, with a list of changes and additions.
- Solutions presented as needs (theo nhu cầu): User-proposed solutions can mask (che đậy) the users' actual needs, lead to automating ineffective business processes, and overconstrain the developers' design options. The analyst must drill down (đi sâu vào) to understand the intent (ý định)—the real requirement—behind a solution the customer has presented.
- Distrust between the business and the development team: As you know, effective requirements engineering demands close collaboration among various stakeholders, particularly customer communities and developers. If these parties do not feel that their counterparts (đối tác) are working in good faith (thiện chí) toward a mutually beneficial outcome, conflicts can arise and requirements elicitation can be threatened (bị đe dọa).

Requirements-related risks

- Requirements analysis: It isn’t prudent (không ngoan/thận trọng) to just record whatever the customer tells you and dive into development. Requirements analysis poses its own threat areas, as described below.
  - Requirements prioritization: Ensure that every functional requirement, feature, or user requirement is prioritized and allocated to a specific system release or iteration.
  - Technically difficult features: Evaluate the feasibility of each requirement to identify those that might take longer than anticipated (dự kiến) to implement. → Take corrective action as early as possible. Prototype the novel (new) or risky requirements to select effective approaches.
  - Unfamiliar technologies, methods, languages, tools, or hardware: Don’t underestimate the learning curve (học tập) for new techniques that are needed to satisfy certain requirements. → Identify those high-risk requirements early on, and work with the development team to allow sufficient time for false starts, learning, and experimentation.

Requirements-related risks

- Requirements specification: Requirements are all about communication. Just because requirements are communicated on paper or in writing doesn't mean they are actually understood.
  - Requirements understanding: Different interpretations of the requirements by developers and customers lead to expectation gaps, in which the delivered product fails to satisfy customer needs. → Peer review of requirements by developers, testers, and customers can mitigate this risk.
  - Time pressure to proceed open issues: It is a good idea to mark the open issues as TBD (to be determined), but it's risky to proceed with construction if these haven't been resolved. → Record who is responsible for closing each open issue and the target date for resolution.
  - Ambiguous terminology: Create a glossary to define business and technical terms that might be interpreted differently by different readers. → Requirements reviews can help participants reach a common understanding of terms and concepts.
  - Design included in requirements: Design elements that are included in the requirements and its constraints. → Review the requirements to make sure they emphasize what needs to be done to solve the business problem.

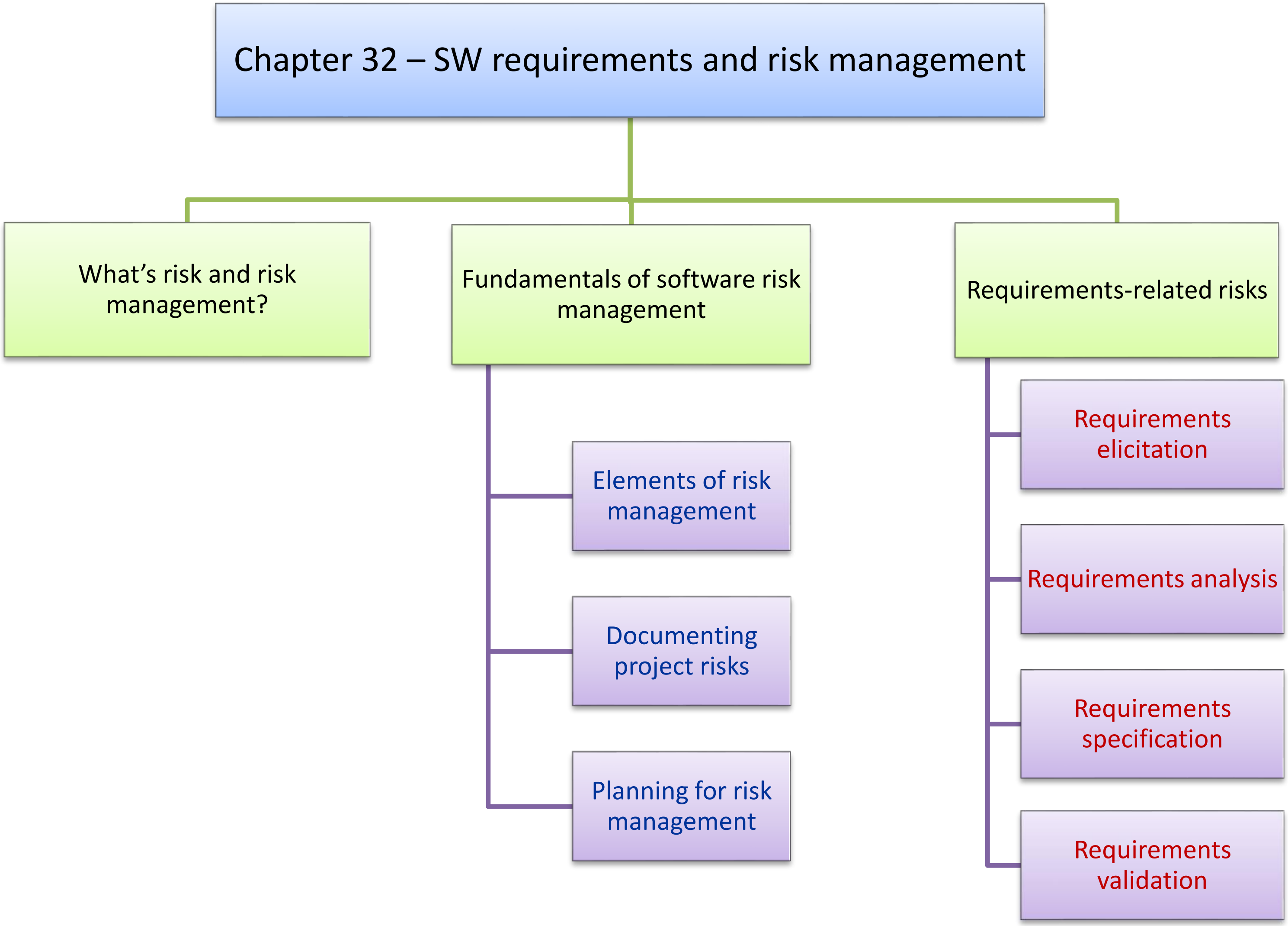
Requirements-related risks

- Requirements validation: Even if you've done a good job on requirements elicitation, it's important to confirm the quality and validity of the solution that the requirements specify. Validation offers the following pitfalls (cạm bẩy).
  - Unvalidated requirements: If you confirm the correctness and quality of each set of requirements before their implementation, you can avoid considerable expensive rework later. → Gain commitment from your customer representatives to participate in requirements reviews. Perform incremental, informal reviews to find problems as early and cheaply as possible.
  - Inspection proficiency (giỏi thanh tra): If inspection participants do not know how to inspect requirements effectively, they might miss serious defects. → Train all team members who will participate in inspections of requirements documents. Invite an experienced inspector from your organization or an outside consultant to observe your early inspections to coach the participants.

Requirements-related risks

- Requirements management: Much of the requirements-related risk on a software project comes from how changes are handled. The requirements management risks are mentioned below.
  - Changing requirements: You can control scope creep by using documented business requirements and scope definitions as the benchmark for approving changes. A collaborative elicitation process with extensive user involvement can cut requirements creep nearly in half (Jones 1996a).
  - Requirements change process: Risks related to how requirements changes are included lacking of a defined change process, using ineffective change mechanisms, failing to incorporate valuable changes efficiently, and incorporating changes that bypass the process.
  - Unimplemented requirements: Requirements tracing helps you avoid overlooking any requirements during design, construction, or testing.
  - Expanding project scope: If requirements are poorly defined initially, further clarification can expand the scope of the project. Vaguely specified areas of the product will consume more effort than anticipated. → Should choose agile project approache.

Review chapter 32


Chapter 32 – SW requirements and risk management
What's risk and risk management?
Fundamentals of software risk management
Elements of risk management
Documenting project risks
Planning for risk management
Requirements-related risks
Requirements elicitation
Requirements analysis
Requirements specification
Requirements validation

THE END
THANK YOU!