ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Nhóm chuyên môn Nhập môn Công nghệ phần mềm

NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Common Software Development Models

CONTENTS

GOALS

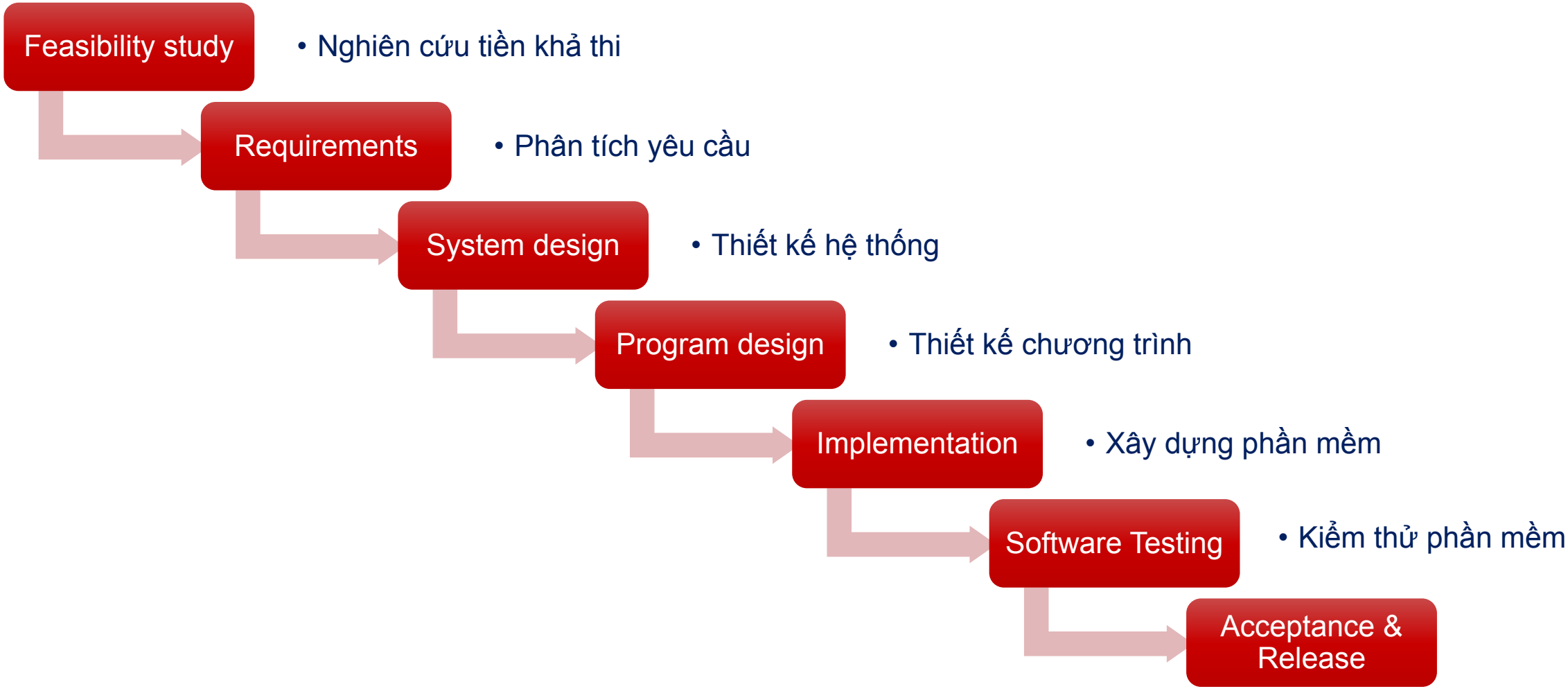
By completing this session, learners are able to:

1. Know some common software development models
2. Understand the differences between software development models
3. Gain an overview of software development models in general

CONTENTS

1. ANALYSIS OF COMMON SOFTWARE DEVELOPMENT MODELS

1.1 Waterfall Model



Steps in the Waterfall Model

1. ANALYSIS OF COMMON SOFTWARE DEVELOPMENT MODELS

1.1 Waterfall Model

- The Waterfall Model is the oldest software lifecycle model, proposed by Winston Royce in 1970.
- It is called the "waterfall" model because it is typically illustrated as a sequence of activities flowing “downward” from left to right through the stages of the lifecycle: analysis, requirements, specification, design, implementation, testing, and maintenance.
- There are many versions of the waterfall model:
  - The phases/activities can be structured at different levels of detail
  - The degree of feedback can be more or less flexible

1. ANALYSIS OF COMMON SOFTWARE DEVELOPMENT MODELS

1.1 Waterfall Model

- Advantages:
  - Clearly separates tasks in each phase
  - Good visibility and easy tracking
  - Quality control at each step
  - Cost monitoring at every stage
- Disadvantages:
  - Relies heavily on requirements being defined early
  - Not feasible in cases where frequent changes are needed
  - In practice, each phase often brings improvements or feedback about previous phases, which typically requires modifications or adjustments

The Waterfall model lacks flexibility

1. ANALYSIS OF COMMON SOFTWARE DEVELOPMENT MODELS

1.1 Modified Waterfall Model

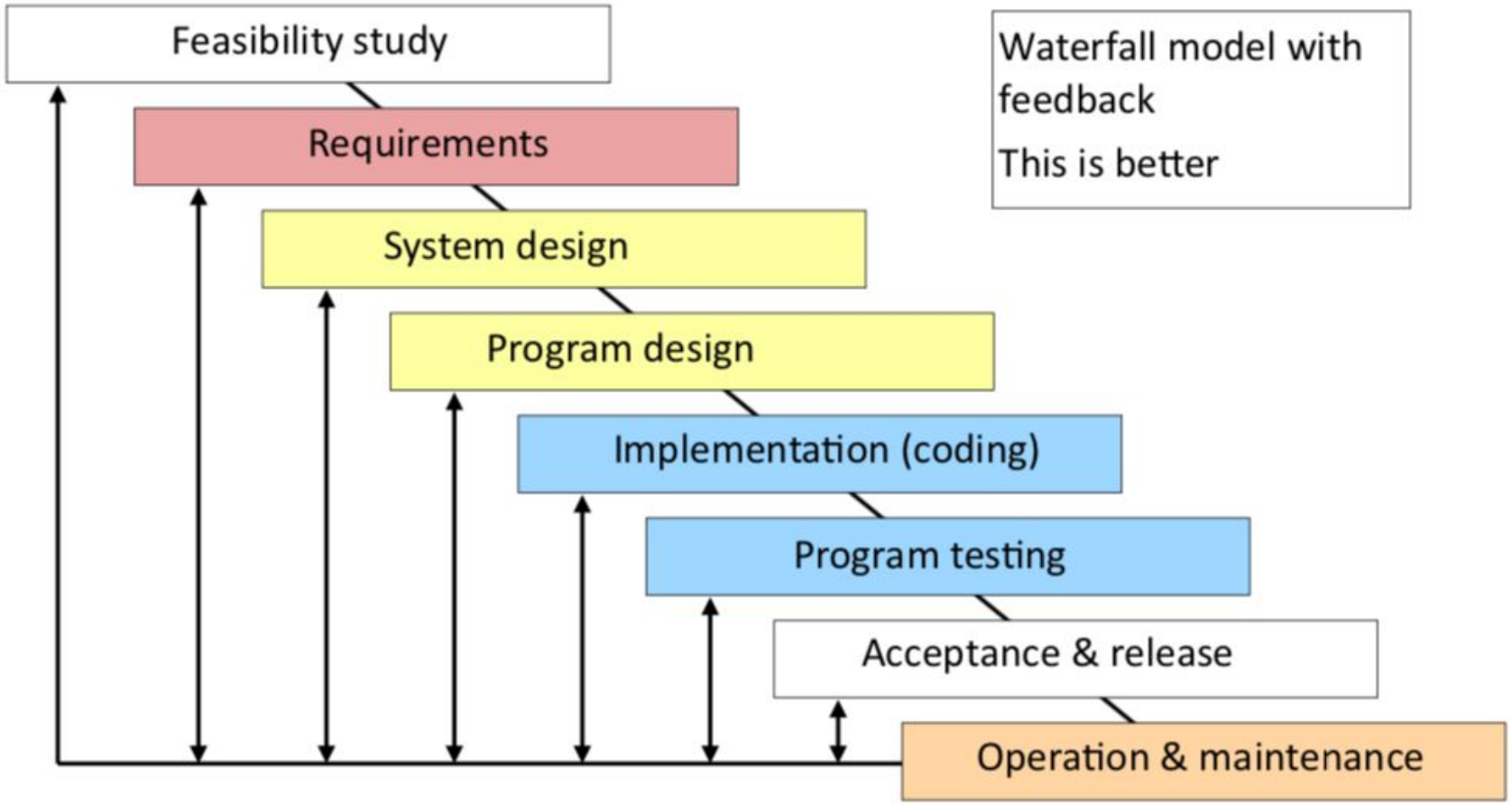


Figure 1.1: Steps in Modified waterfall model

Reference: Utami Jayadi, software Process Models and the feasibility study

1. ANALYSIS OF COMMON SOFTWARE DEVELOPMENT MODELS

1.1 Modified Waterfall Model

This model is typically applied to:

1. The modified Waterfall model works best when the requirements are well understood and the design is straightforward

   Examples:

   - Converting a manual data processing system where the requirements are already well understood
   - A new version of a system that has functionality similar to the previous product

2. Components of a large system where certain parts have clearly defined requirements and are clearly separated from the rest of the system

1.2 Prototyping Model

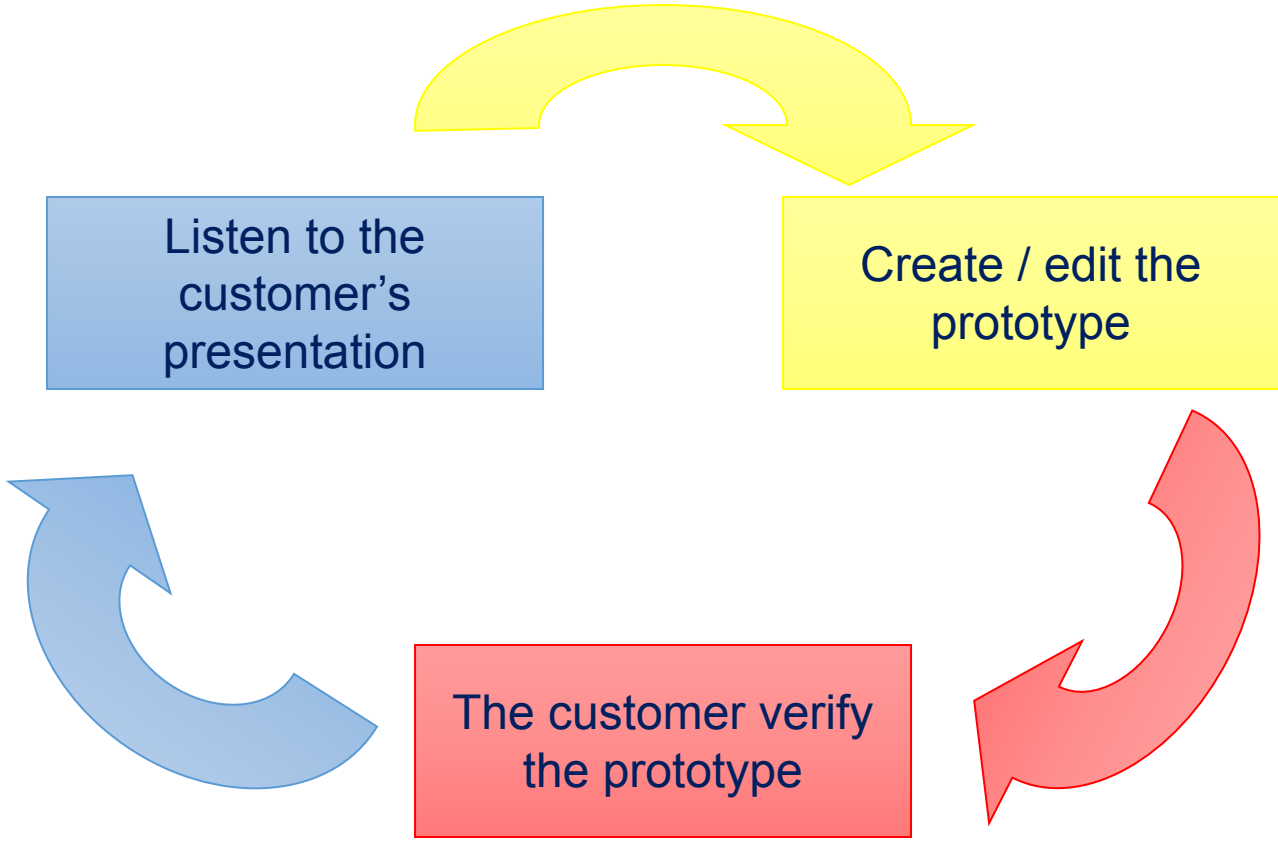


Figure 1.2: Prototyping model

1.2 Prototyping Model – When?

- When only the general purpose of the software is known, but input, processing, or output requirements are still unclear
- Used to gather requirements through rapid prototyping
- Algorithms and technical solutions used in the prototype may not be optimal or efficient, as long as they serve as a basis for discussion to elicit user requirements

1.3 Growth model

- Most complex software systems evolve over time: environments change, new requirements emerge, and additional functions and features are needed.
- Evolutionary models are iterative in nature. Software engineers create versions that become increasingly refined and complex.
- Typical models include:
  - Incremental
  - Spiral
  - WINWIN spiral
  - Concurrent development

1. ANALYSIS OF COMMON SOFTWARE DEVELOPMENT MODELS

1.4 Incremental model

- Combines the sequential model with the iterative concept of prototyping
- A product with the most basic system requirements is developed first
- Additional functions with other requirements are developed incrementally afterward
- The process is repeated to gradually improve and complete the system

1.4 Incremental model

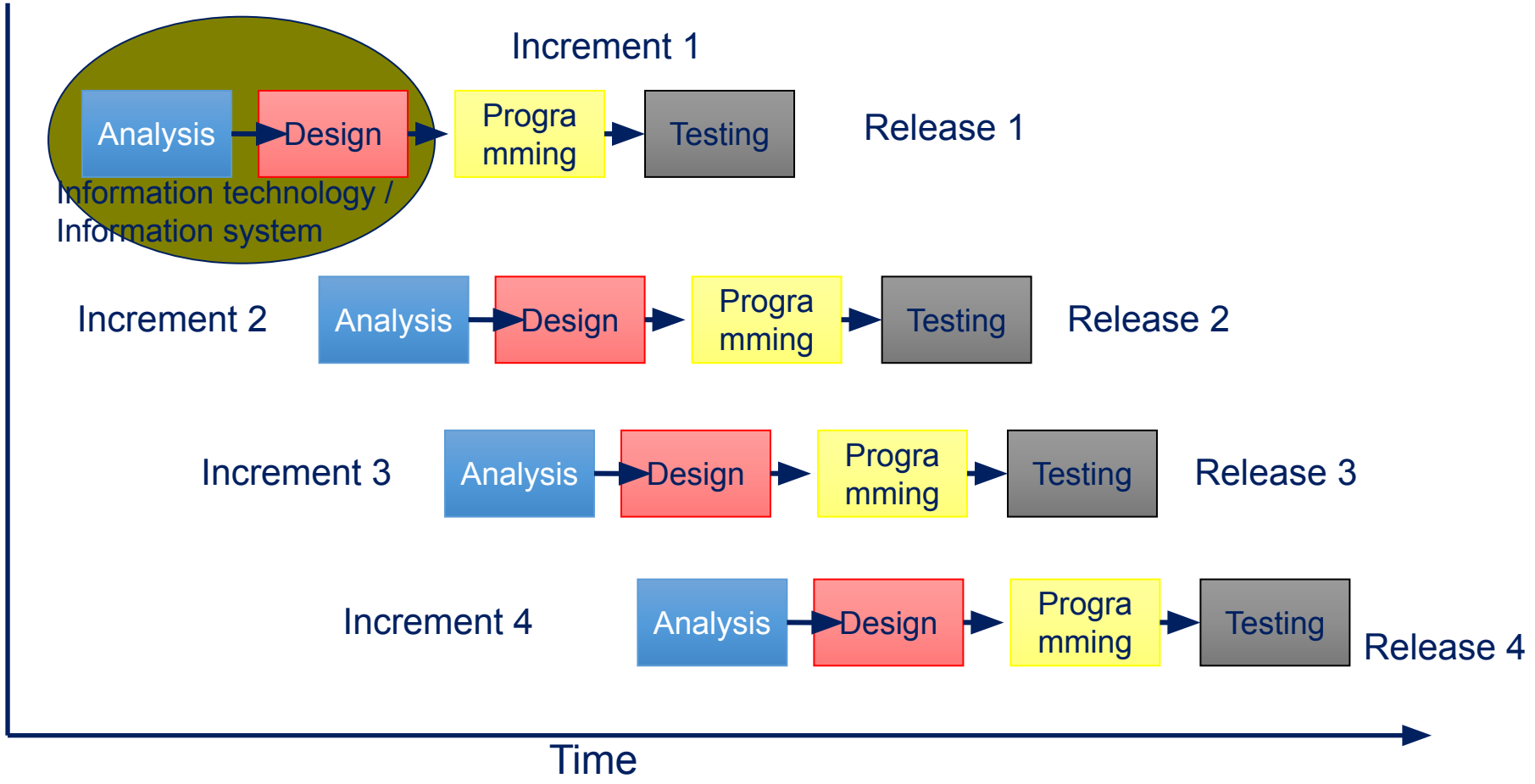


Figure 1.3: Incremental model

1.5 Rapid Application Development

- It is an incremental software development process, with each development cycle being very short (60–90 days).
- It is built on component-based construction with a strong focus on reusability.
- The process involves multiple teams, with each team carrying out a Rapid Application Development (RAD) process through the following phases: Business Modeling, Data Modeling, Process Modeling, Application Generation, Testing and Evaluation.

1. ANALYSIS OF COMMON SOFTWARE DEVELOPMENT MODELS

1.5 Rapid Application Development

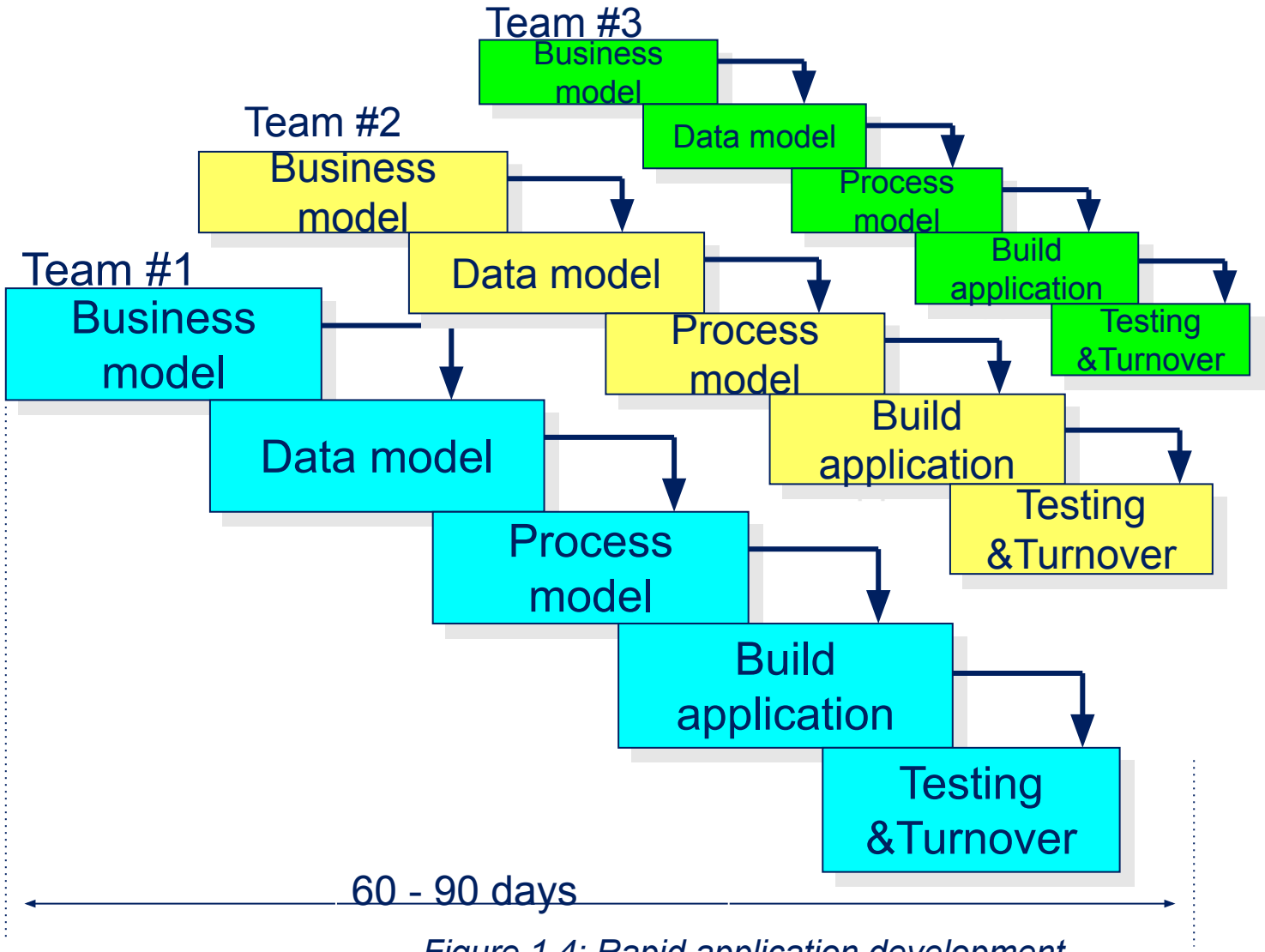


Figure 1.4: Rapid application development

1.6 Spiral Model

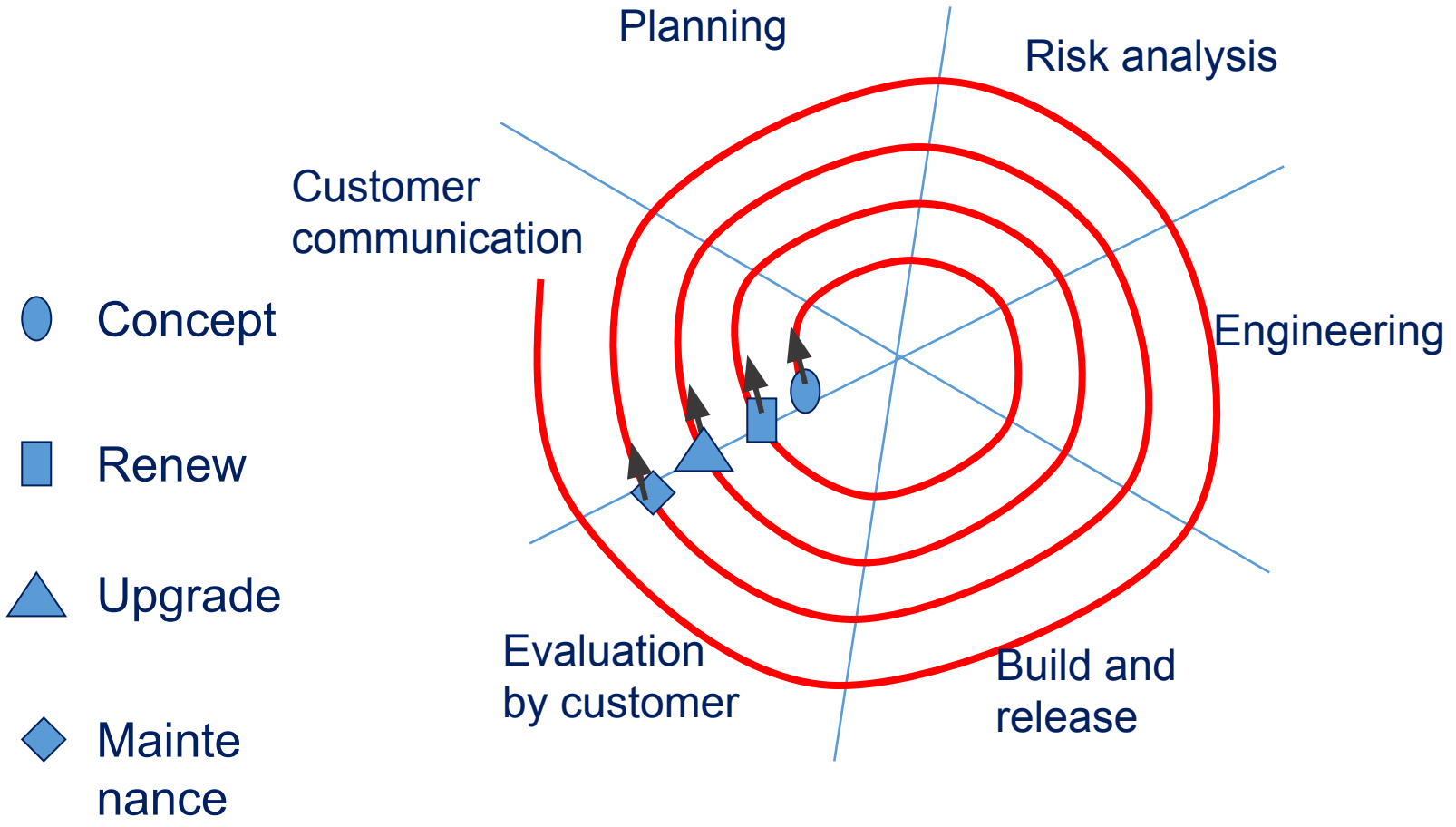


Figure 1.5: Spiral model

1. ANALYSIS OF COMMON SOFTWARE DEVELOPMENT MODELS

1.6 Spiral Model

- Customer Communication: Interaction between developers and customers to understand requirements and gather feedback
- Planning: Define resources, timelines, and other relevant information
- Risk Analysis: Evaluate technical and management risks
- Engineering: Build one or more representations of the application
- Construction and Release: Develop, test, install, and provide user support (documentation, training, etc.)
- Customer Evaluation: Receive user feedback on the software representation during the engineering and deployment phases

1.6 Spiral Model

- **This model is typically applied in:**
  - Suitable for large-scale software systems
  - Easier to manage risks at each stage of evolution
  - Difficult to convince customers that the spiral evolutionary approach is controllable
  - Not as widely adopted as linear or prototyping models

CONTENTS

2. CONCLUSION

- There is no single software development process that is perfect for all projects.
- Each process has its own strengths and limitations, and it may also need to be adapted or combined to suit the specific requirements of the project and development team.
- Choosing the right process for a project is a crucial step in ensuring the success of the software development effort.

SUMMARY AND OUTLOOK

1. The lesson has provided learners with several **common software development models** along with their key characteristics. Through these models, learners can gain an overall understanding of how software development processes are structured.
2. Following this lesson, learners can further explore other software development models in more detail on their own.

NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Một số quy trình phát triển phần mềm phổ biến

Biên soạn:
TS. Nguyễn Nhất Hải
Trình bày:
TS. Nguyễn Nhất Hải

NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Bài học tiếp theo:

Hướng dẫn bài tập: Chương 2. Vòng đời phần mềm

Tài liệu tham khảo:

[1] R. Pressman, Software Engineering: A Practitioner's Approach. 8th Ed., McGraw-Hill, 2016 và bộ slide đi kèm.

[2] I. Sommerville, Software Engineering. 10th Ed., AddisonWesley, 2017.

[3] Pankaj Jalote, An Integrated Approach to Software Engineering, 3rd Ed., Springer.

[4] Shari Lawrence Pleeger, Joanne M.Atlee, Software Engineering theory and practice. 4th Ed., Pearson, 2009