Software Engineering

Introduction to Software Engineering

1102SE01
MBA, IM, NTPU (M5010) (Spring 2022)
Wed 2, 3, 4 (9:10-12:00) (B8F40)

https://meet.google.com/ish-gzmy-pmo

Min-Yuh Day, Ph.D,
Associate Professor

Institute of Information Management, National Taipei University

https://web.ntpu.edu.tw/~myday

2022-02-23

Min-Yuh Day, Ph.D.

Associate Professor, Information Management, NTPU

Visiting Scholar, IIS, Academia Sinica

Ph.D., Information Management, NTU

Director, Intelligent Financial Innovation Technology, IFIT Lab, IM, NTPU

Artificial Intelligence, Financial Technology, Big Data Analytics,
Data Mining and Text Mining, Electronic Commerce

Course Syllabus
National Taipei University
Academic Year 110, 2nd Semester (Spring 2022)

- **Course Title: Software Engineering**
- **Instructor: Min-Yuh Day**
- **Course Class: MBA, IM, NTPU (3 Credits, Elective)**
- **Details**
  - **In-Person and Distance Learning EMI Course (3 Credits, Elective, One Semester) (M5010)**
- **Time & Place: Wed, 2, 3, 4, (9:10-12:00) (B8F40)**
- **Google Meet:** https://meet.google.com/ish-gzmy-pmo

https://meet.google.com/ish-gzmy-pmo

Course Objectives

1. Understand the fundamental concepts and research issues of software engineering.
2. Equip with Hands-on practices of software engineering.
3. Conduct information systems research in the context of software engineering.

Course Outline

- **This course introduces the fundamental concepts, research issues, and hands-on practices of software engineering.**
- **Topics include:**
  1. Introduction to Software Engineering
  2. Software Products and Project Management: Software product management and prototyping
  3. Agile Software Engineering: Agile methods, Scrum, and Extreme Programming
  4. Features, Scenarios, and Stories
  5. Software Architecture: Architectural design, System decomposition, and Distribution architecture
  6. Cloud-Based Software: Virtualization and containers, Everything as a service, Software as a service
  7. Cloud Computing and Cloud Software Architecture
  8. Microservices Architecture, RESTful services, Service deployment
  9. Security and Privacy; Reliable Programming
  10. Testing: Functional testing, Test automation, Test-driven development, and Code reviews
  11. DevOps and Code Management: Code management and DevOps automation
  12. Case Study on Software Engineering

Core Competence

- **Exploring new knowledge in information technology, system development and application 80 %**
- **Internet marketing planning ability 10 %**
- **Thesis writing and independent research skills 10 %**

Four Fundamental Qualities

- Professionalism
  - Creative thinking and Problem-solving 30 %
  - Comprehensive Integration 30 %
- Interpersonal Relationship
  - Communication and Coordination 10 %
  - Teamwork 10 %
- Ethics
  - Honesty and Integrity 5 %
  - Self-Esteem and Self-reflection 5 %
- International Vision
  - Caring for Diversity 5 %
  - Interdisciplinary Vision 5 %

College Learning Goals

- Ethics/Corporate Social Responsibility
- Global Knowledge/Awareness
- Communication
- Analytical and Critical Thinking

Department Learning Goals

- **Information Technologies and System Development Capabilities**
- **Internet Marketing Management Capabilities**
- **Research capabilities**

Syllabus

Week	Date	Subject/Topics
1	2022/02/23	Introduction to Software Engineering
2	2022/03/02	Software Products and Project Management: Software product management and prototyping
3	2022/03/09	Agile Software Engineering: Agile methods, Scrum, and Extreme Programming
4	2022/03/16	Features, Scenarios, and Stories
5	2022/03/23	Case Study on Software Engineering I
6	2022/03/30	Software Architecture: Architectural design, System decomposition, and Distribution architecture

Syllabus

Week	Date	Subject/Topics
7	2022/04/06	Make-up holiday (No Classes)
8	2022/04/13	Midterm Project Report
9	2022/04/20	Cloud-Based Software: Virtualization and containers, Everything as a service, Software as a service
10	2022/04/27	Cloud Computing and Cloud Software Architecture
11	2022/05/04	Microservices Architecture, RESTful services, Service deployment
12	2022/05/11	Industry Practices of Software Engineering

Syllabus

Week	Date	Subject/Topics
13	2022/05/18	Case Study on Software Engineering II
14	2022/05/25	Security and Privacy; Reliable Programming; Testing: Test-driven development, and Code reviews; DevOps and Code Management: DevOps automation
15	2022/06/01	Final Project Report I
16	2022/06/08	Final Project Report II
17	2022/06/15	Self-learning
18	2022/06/22	Self-learning

Teaching Methods and Activities

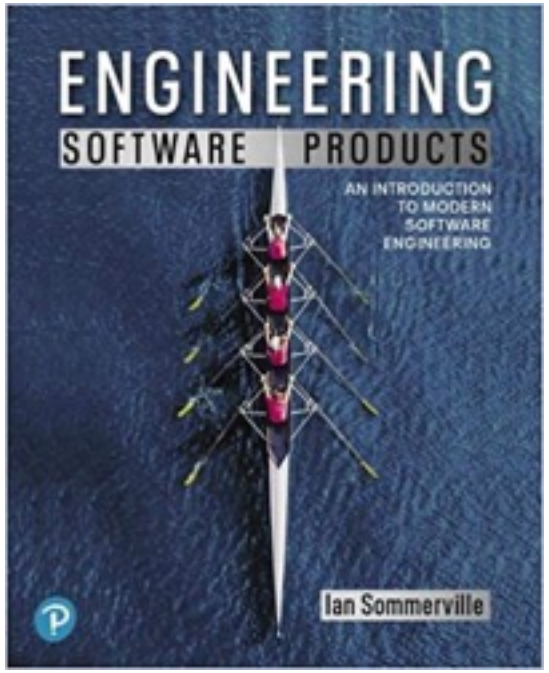
- Lecture
- Discussion
- Practicum

Evaluation Methods

- Individual Presentation 60 %
- Group Presentation 10 %
- Case Report 10 %
- Class Participation 10 %
- Assignment 10 %

Required Texts

- **Ian Sommerville (2019),**
  Engineering Software Products:
  An Introduction to Modern Software Engineering,
  Pearson.



Reference Books

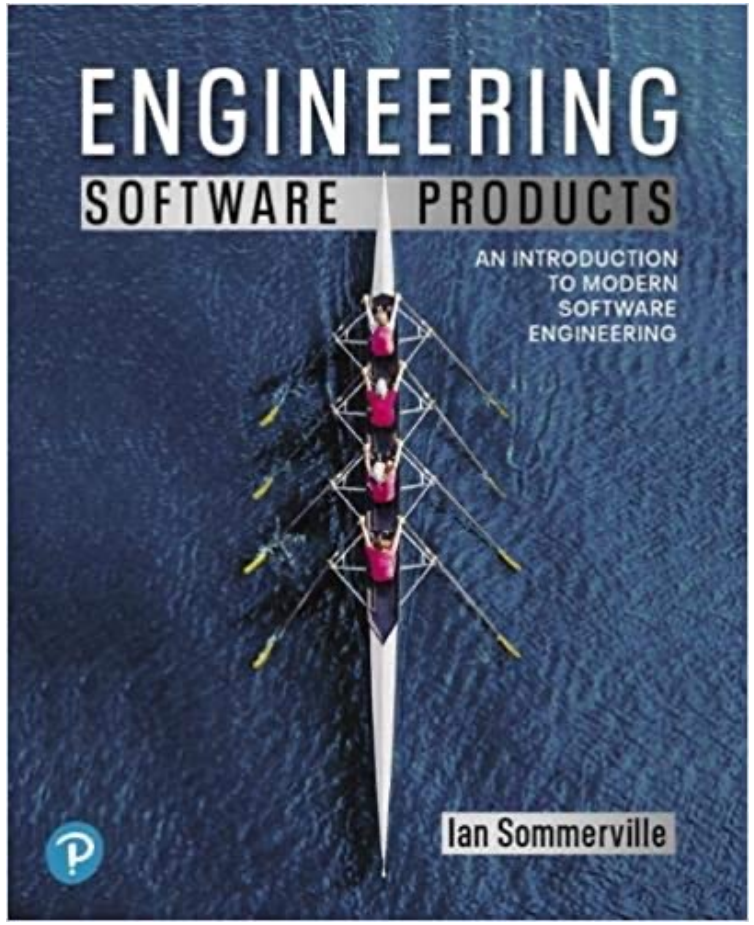
- Ian Sommerville (2015),
  Software Engineering,
  10th Edition, Pearson.
- Titus Winters, Tom Manshreck, and Hyrum Wright (2020),
  Software Engineering at Google: Lessons Learned from Programming Over Time, O'Reilly Media.
- Project Management Institute (2017),
  Agile Practice Guide, PMI
- Project Management Institute (2021),
  A Guide to the Project Management Body of Knowledge (PMBOK Guide) – Seventh Edition and The Standard for Project Management, PMI

Ian Sommerville (2019),

Engineering Software Products:

An Introduction to Modern Software Engineering,

Pearson.

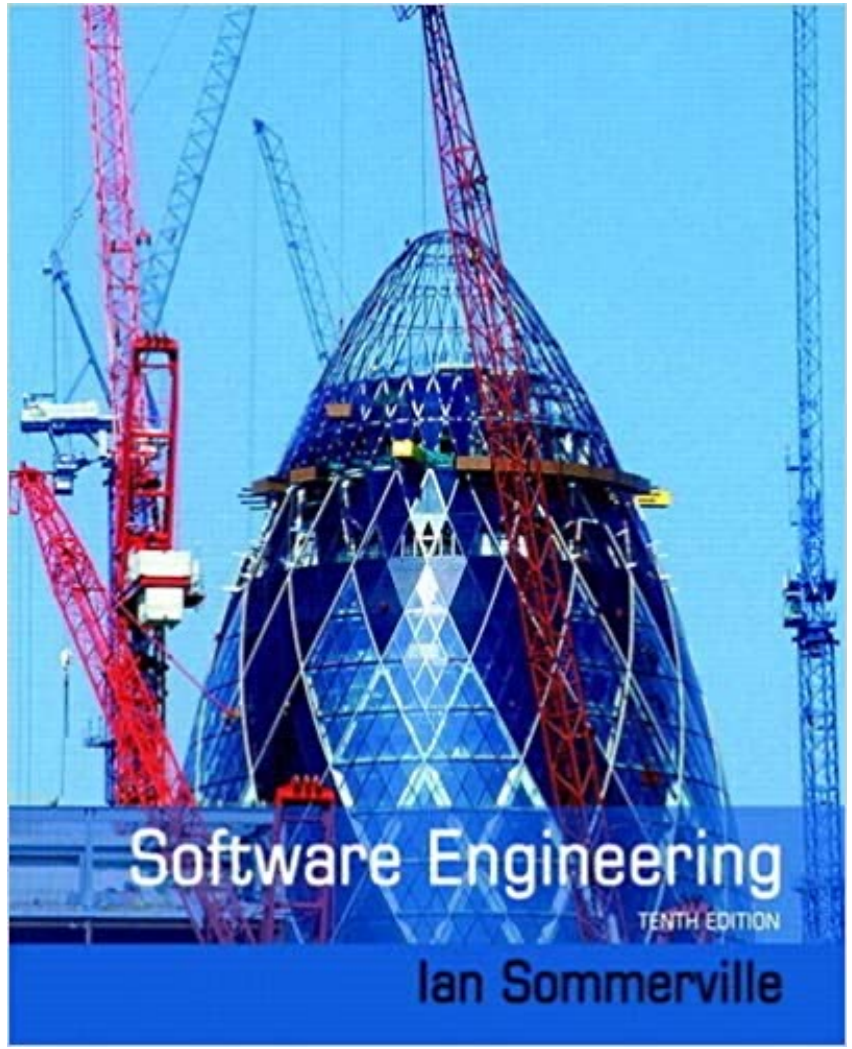


Source: https://www.amazon.com/Engineering-Software-Products-Ian-Sommerville/dp/013521064X

Ian Sommerville (2015),

Software Engineering,

10th Edition, Pearson.



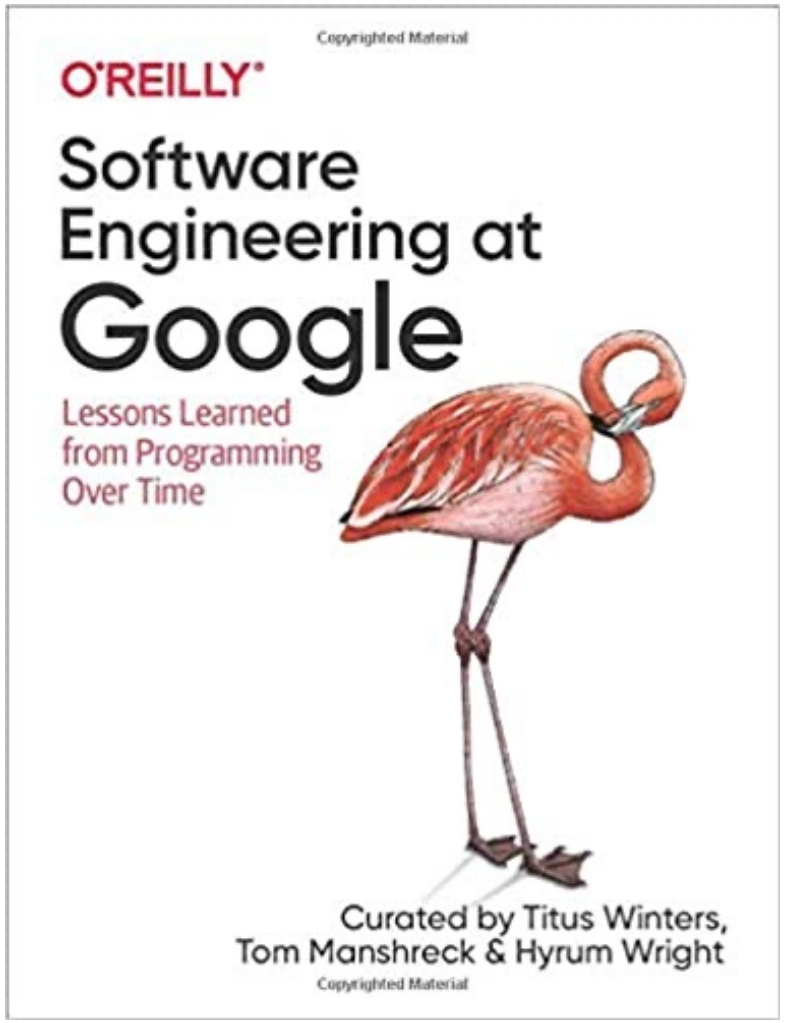
Source: https://www.amazon.com/Software-Engineering-10th-Ian-Sommerville/dp/0133943038

Titus Winters, Tom Manshreck, and Hyrum Wright (2020),

Software Engineering at Google:

Lessons Learned from Programming Over Time,

O'Reilly Media.

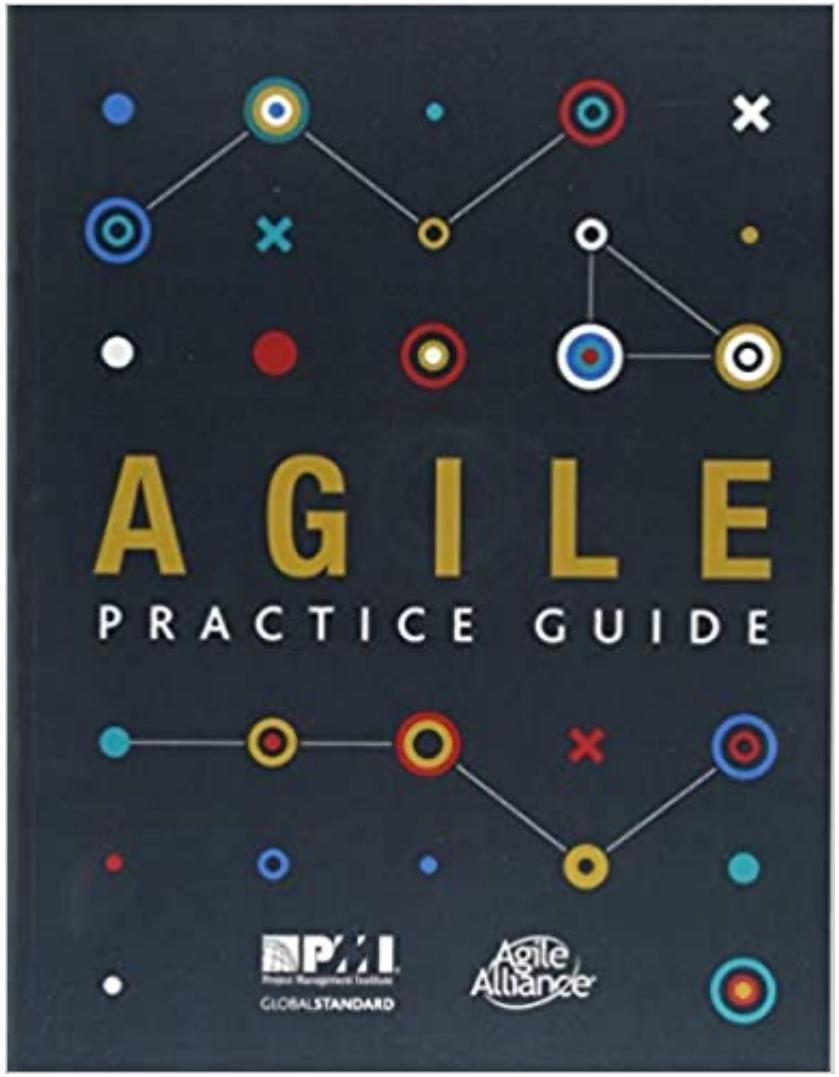


Source: https://www.amazon.com/Software-Engineering-Google-Lessons-Programming/dp/1492082791

Project Management Institute (2017),

Agile Practice Guide

PMI

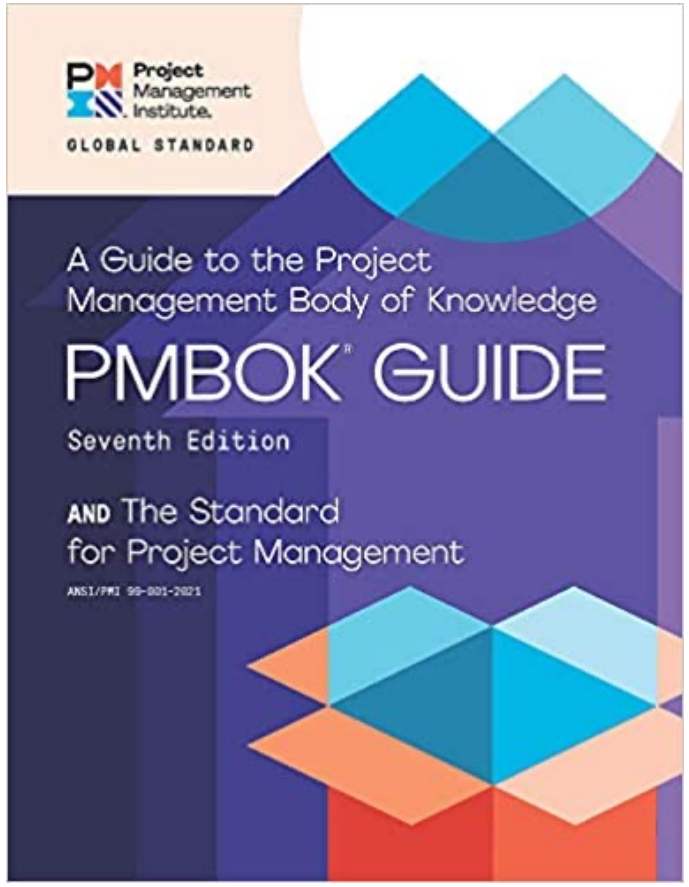


Source: https://www.amazon.com/Agile-Practice-Project-Management-Institute/dp/1628251999/

Project Management Institute (2021),

A Guide to the Project Management Body of Knowledge (PMBOK Guide) –

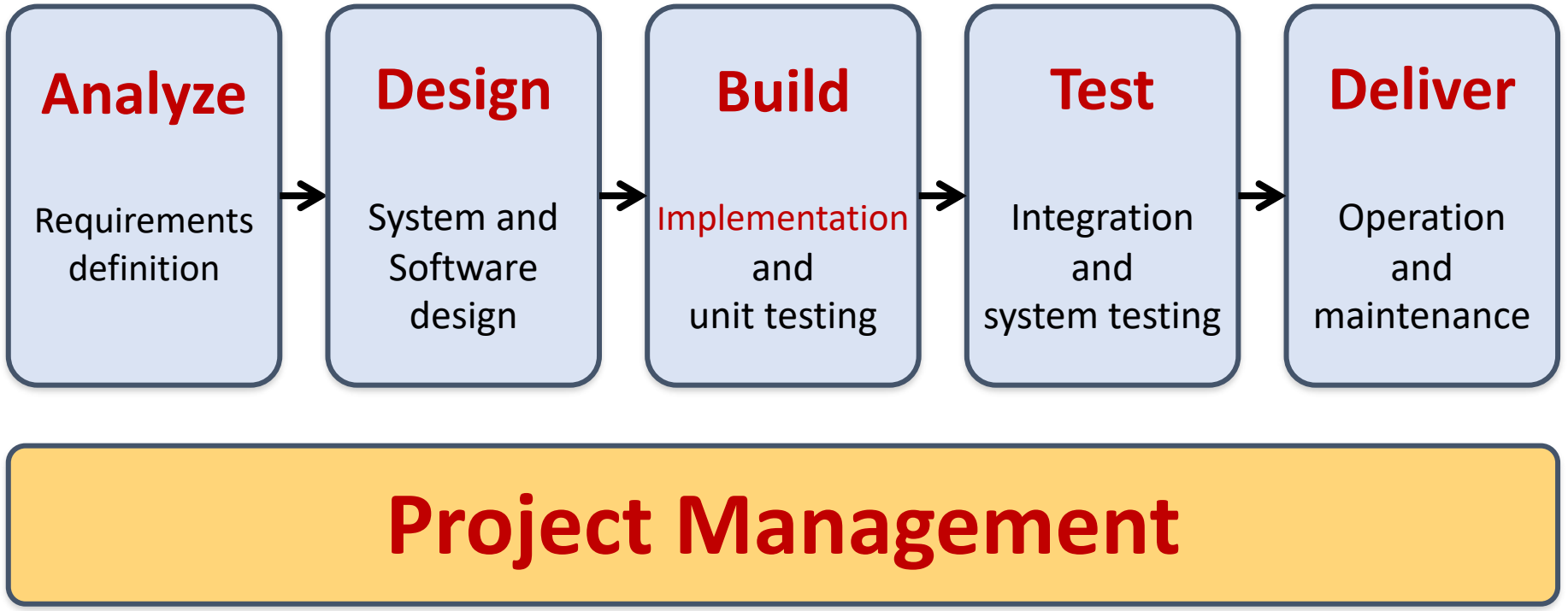
Seventh Edition and The Standard for Project Management



Source: https://www.amazon.com/Guide-Project-Management-Knowledge-PMBOK%C2%AE/dp/1628256648

Software Engineering

Software Engineering
and
Project Management

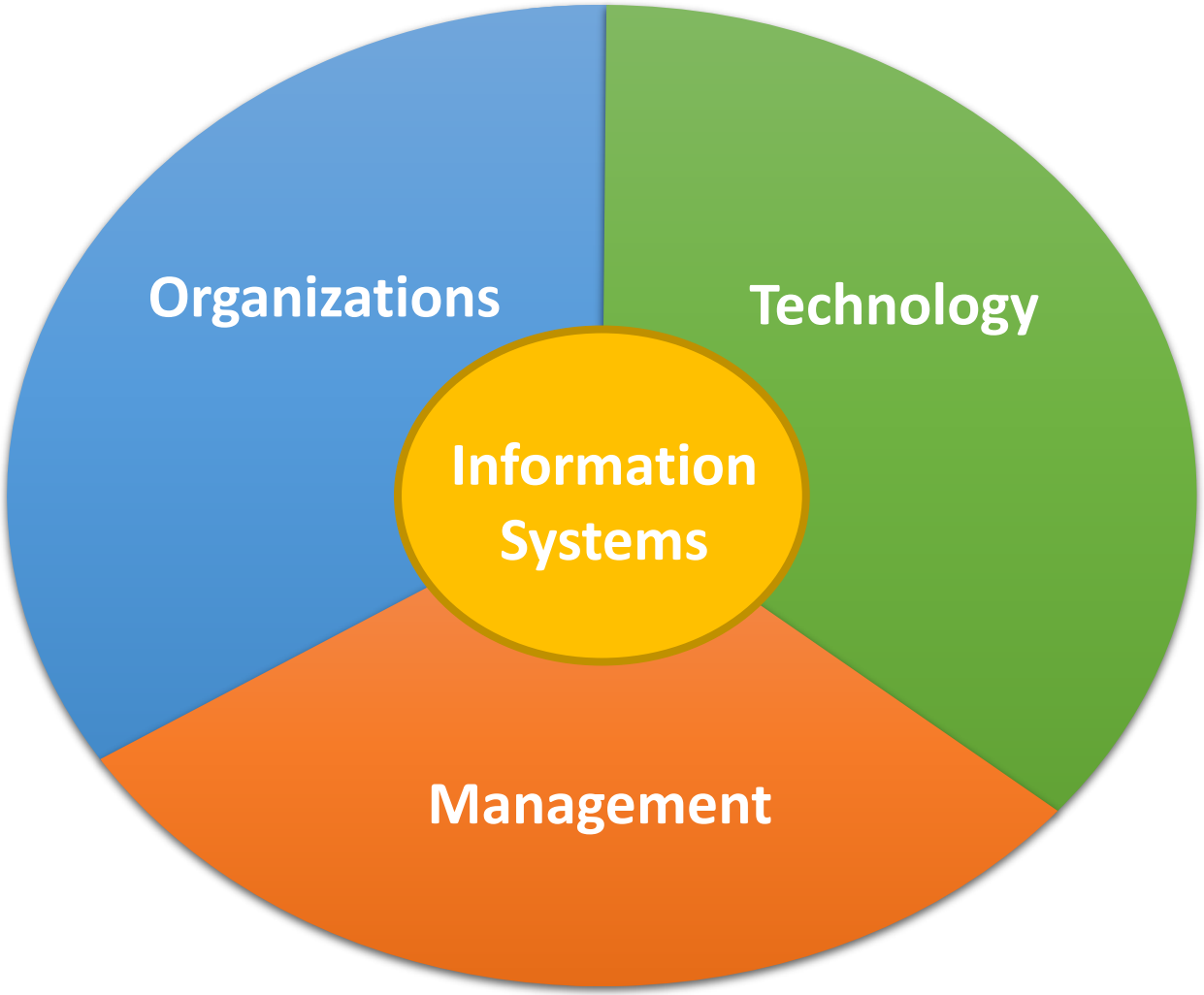


Information Management

Management Information Systems (MIS)

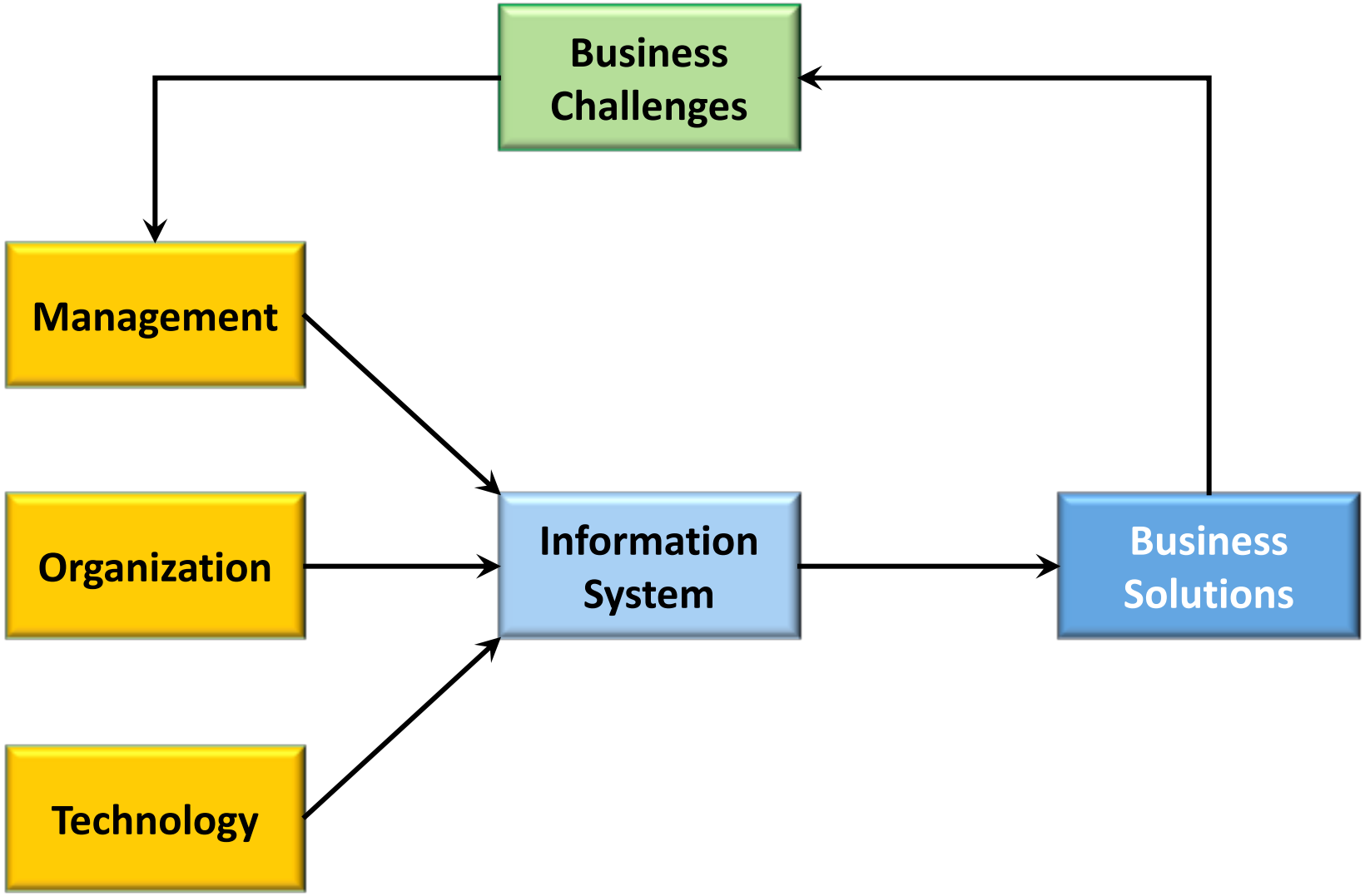
Information Systems

Information Management (MIS)
Information Systems



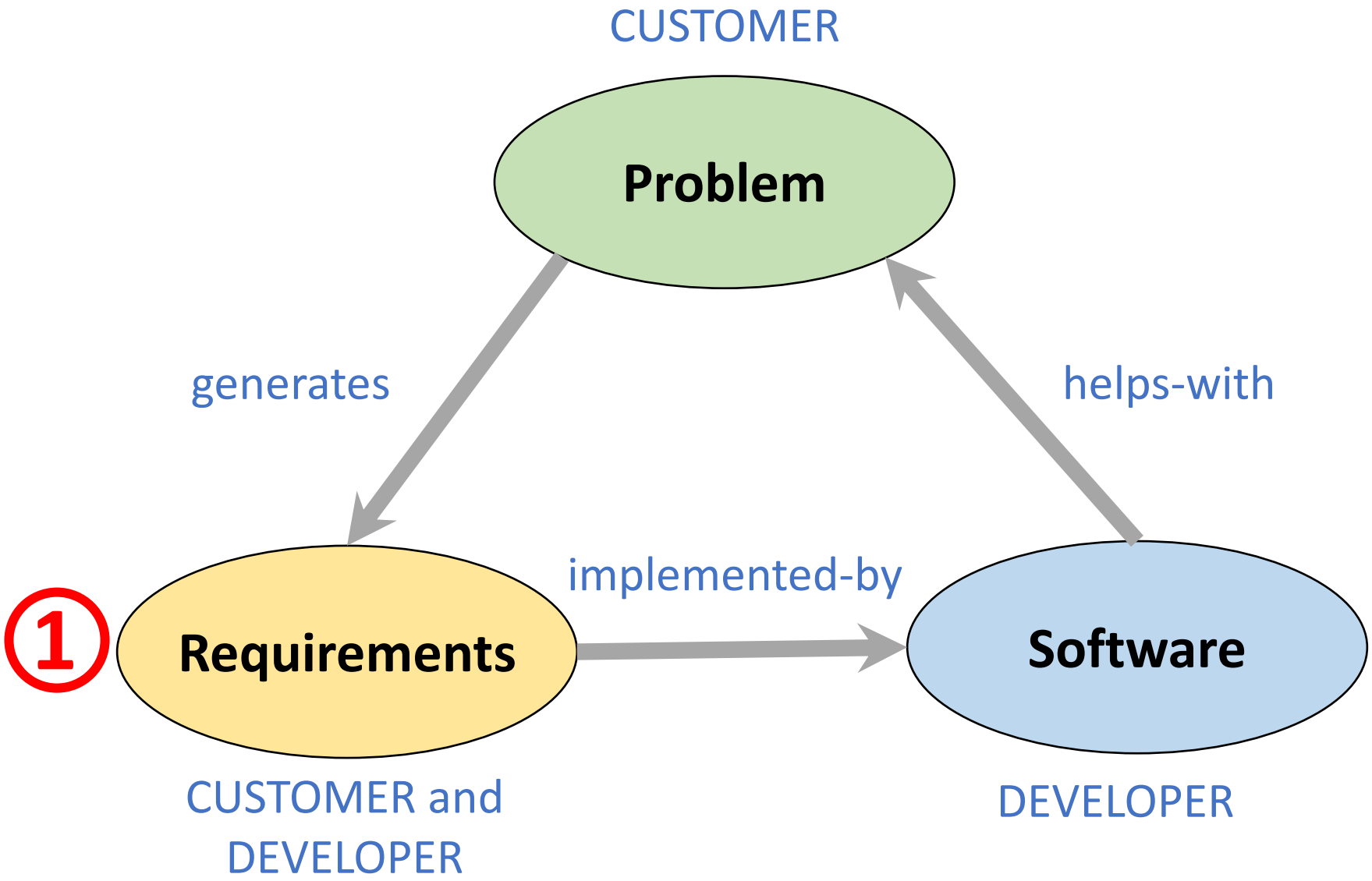
Source: Kenneth C. Laudon & Jane P. Laudon (2014), Management Information Systems: Managing the Digital Firm, Thirteenth Edition, Pearson.

Fundamental MIS Concepts



Source: Kenneth C. Laudon & Jane P. Laudon (2014), Management Information Systems: Managing the Digital Firm, Thirteenth Edition, Pearson.

Project-based software engineering



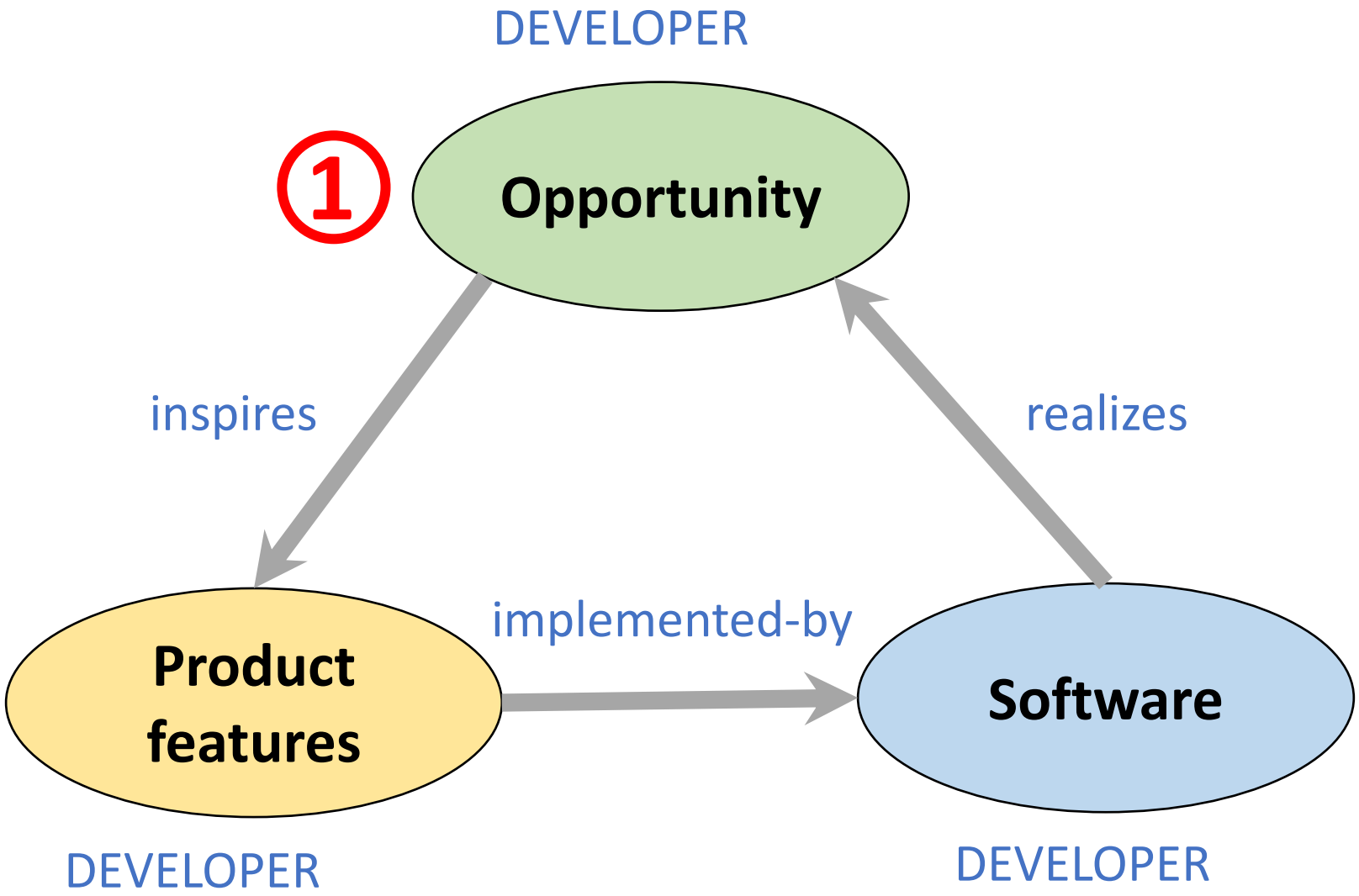
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Project-based software engineering

- The starting point for the software development is a set of 'software requirements' that are owned by an external client and which set out what they want a software system to do to support their business processes.
- The software is developed by a software company (the contractor) who design and implement a system that delivers functionality to meet the requirements.
- The customer may change the requirements at any time in response to business changes (they usually do). The contractor must change the software to reflect these requirements changes.
- Custom software usually has a long-lifetime (10 years or more) and it must be supported over that lifetime.

Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Product software engineering



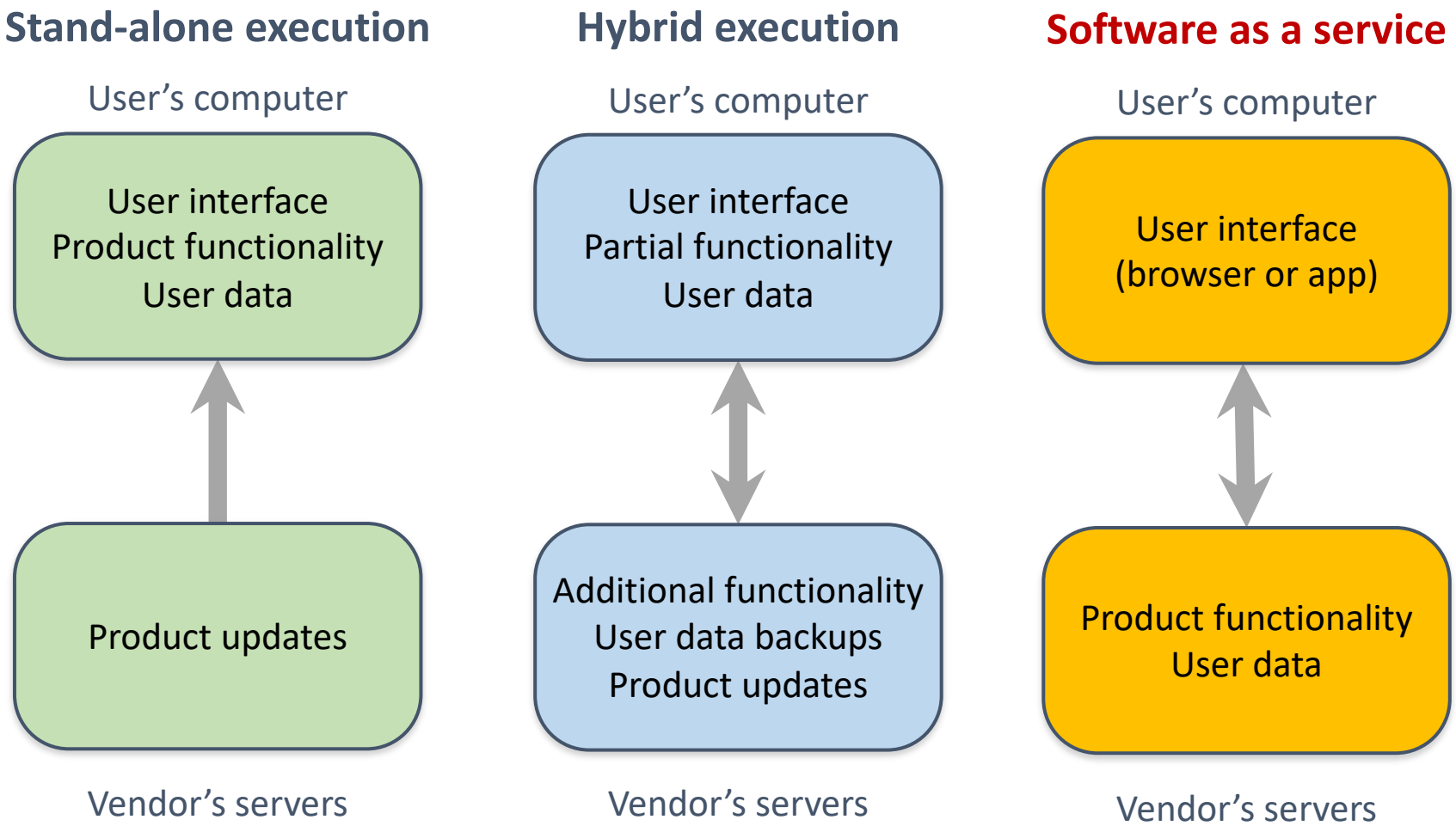
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Product software engineering

- The starting point for product development is a business opportunity that is identified by individuals or a company.
  They develop a software product to take advantage of this opportunity and sell this to customers.
- The company who identified the opportunity design and implement a set of software features that realize the opportunity and that will be useful to customers.
- The software development company are responsible for deciding on the development timescale, what features to include and when the product should change.
- Rapid delivery of software products is essential to capture the market for that type of product.

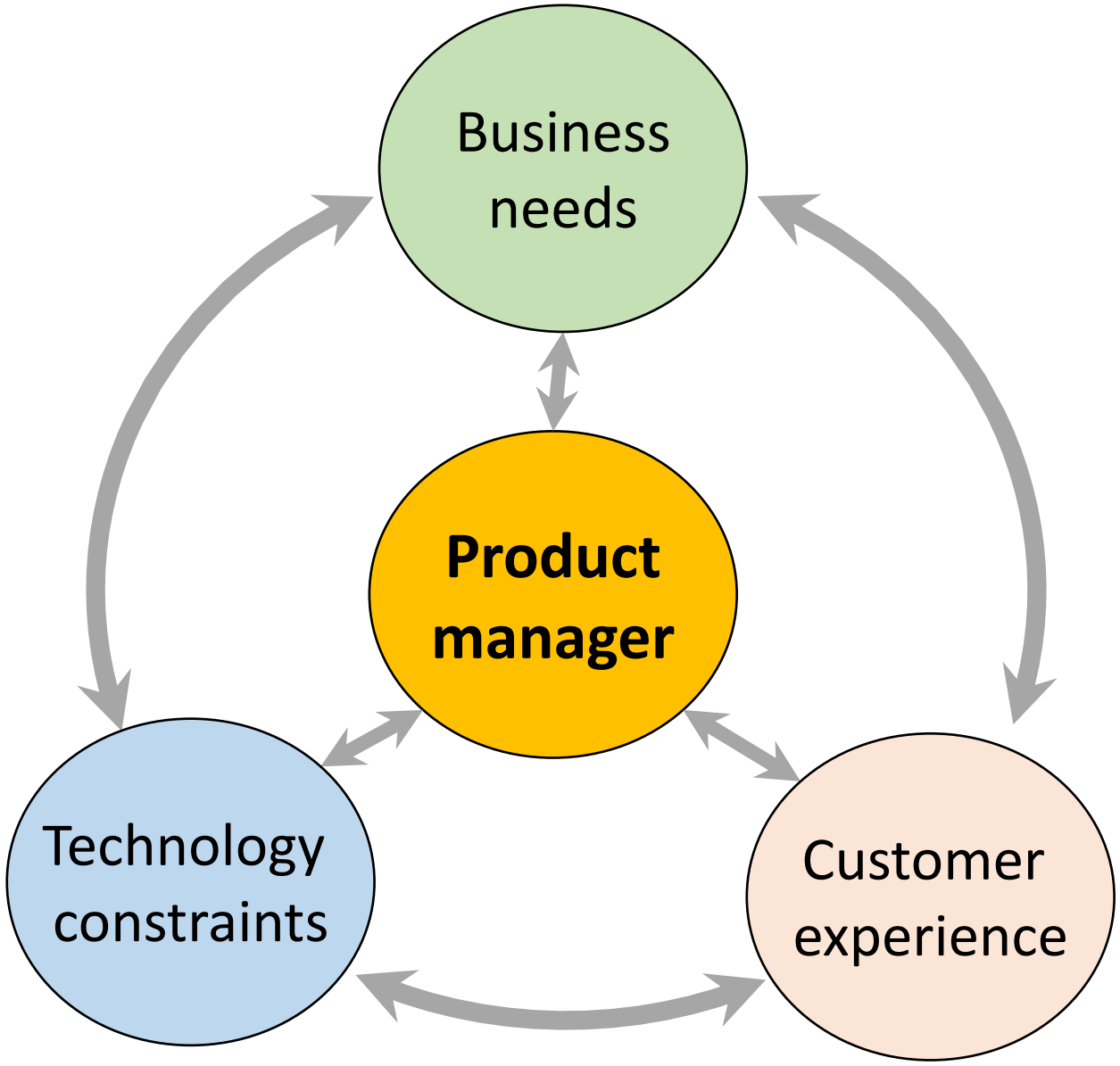
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Software execution models



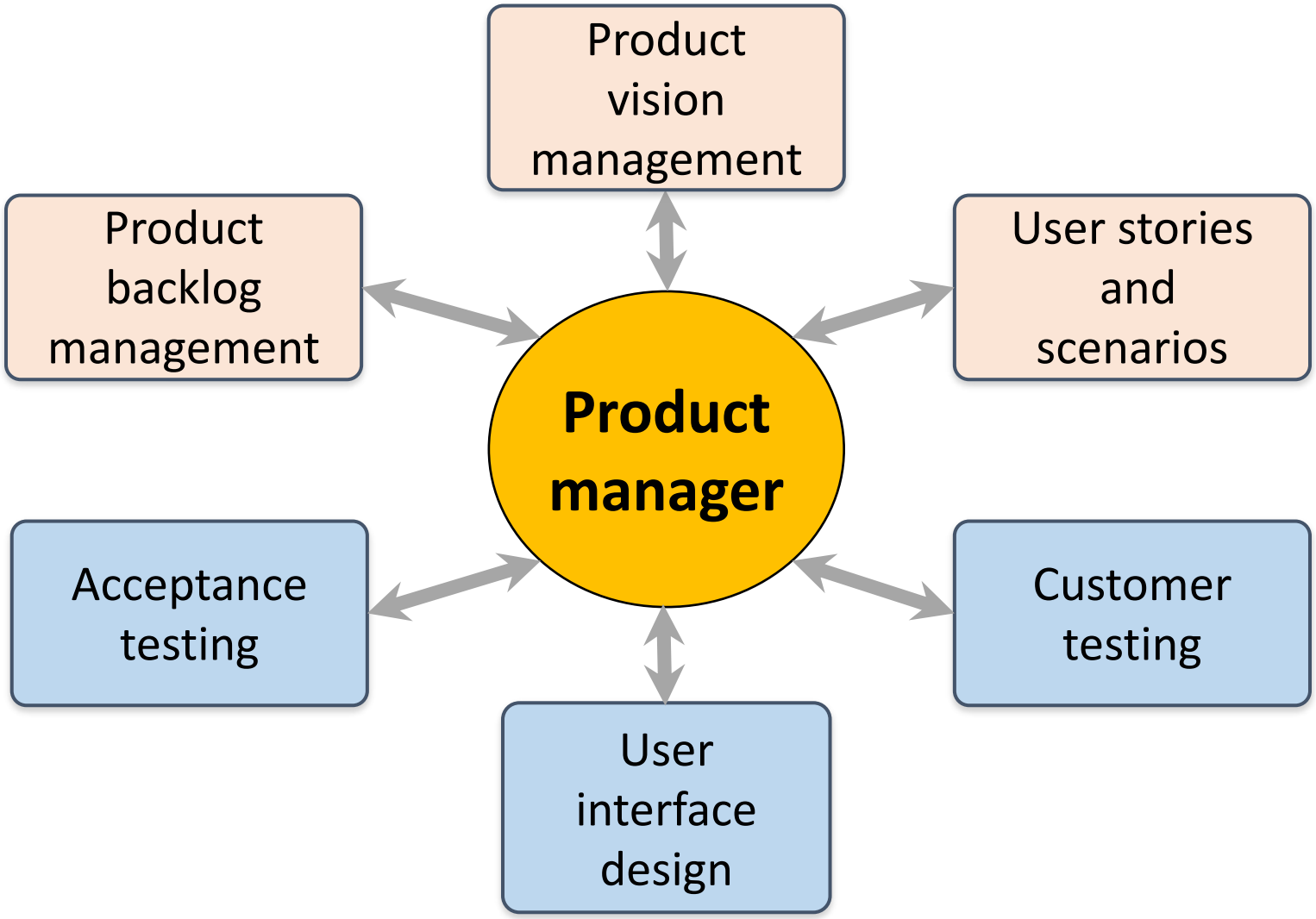
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Product management concerns



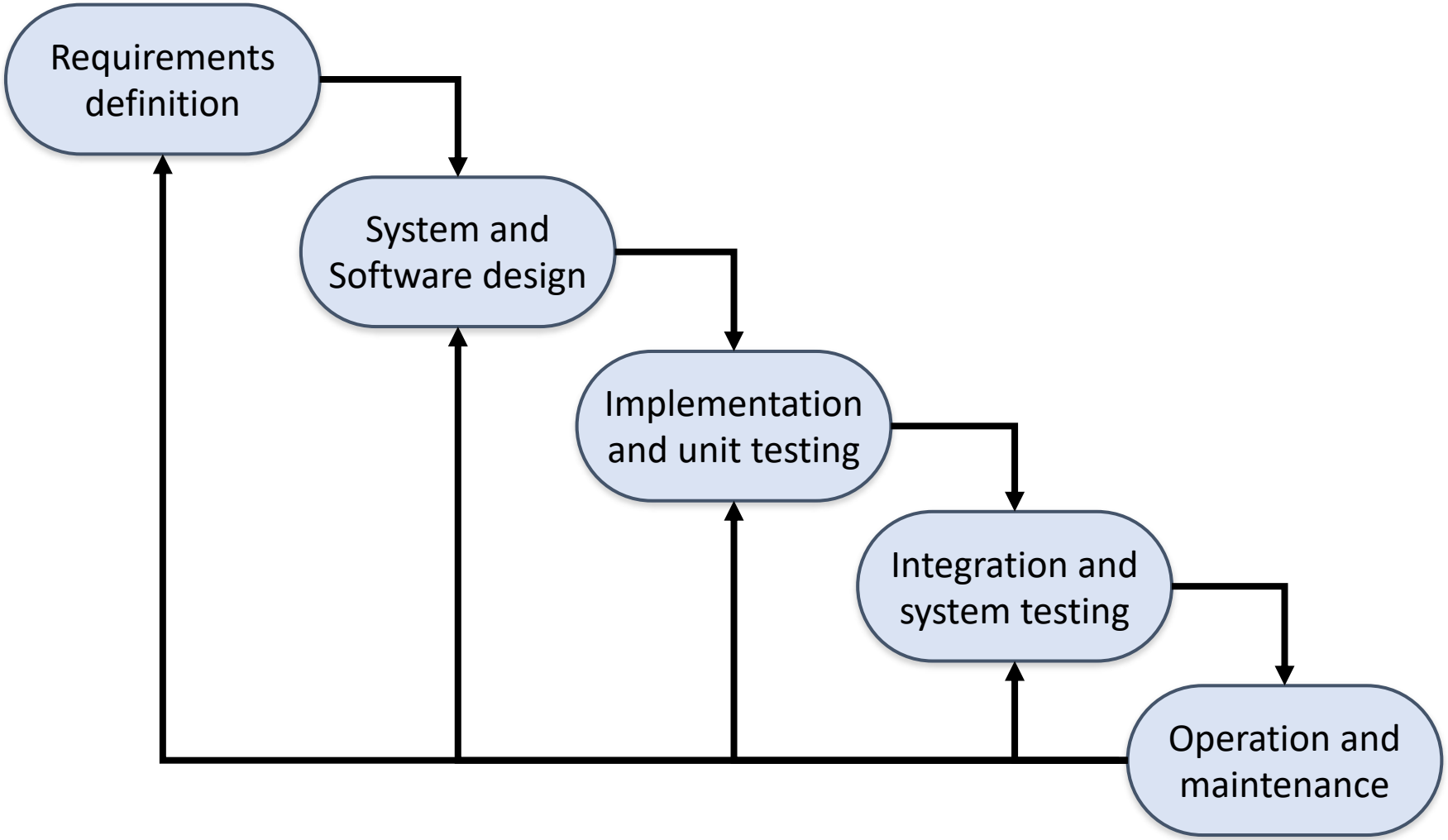
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Technical interactions of product managers



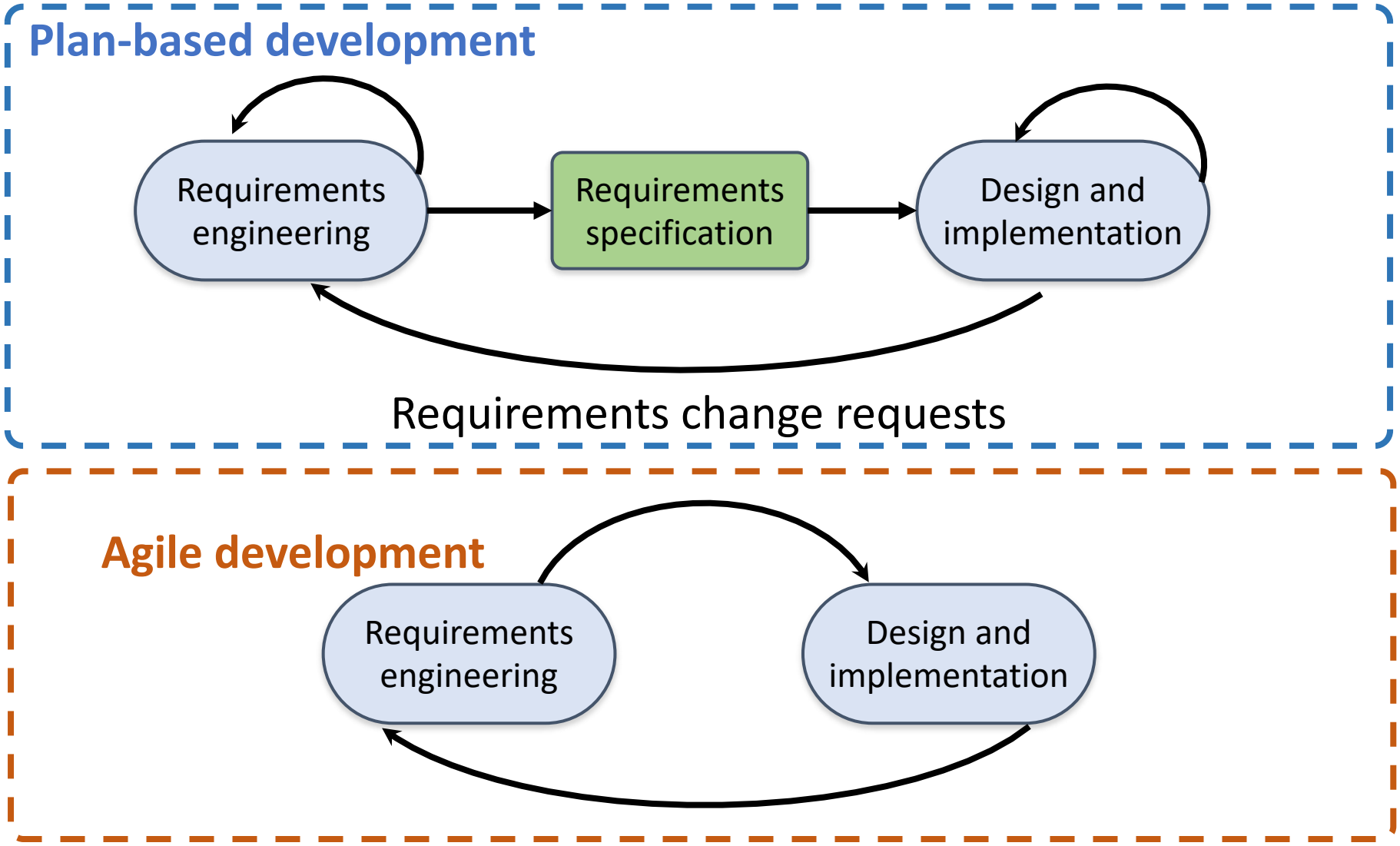
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Software Development Life Cycle (SDLC)
The waterfall model



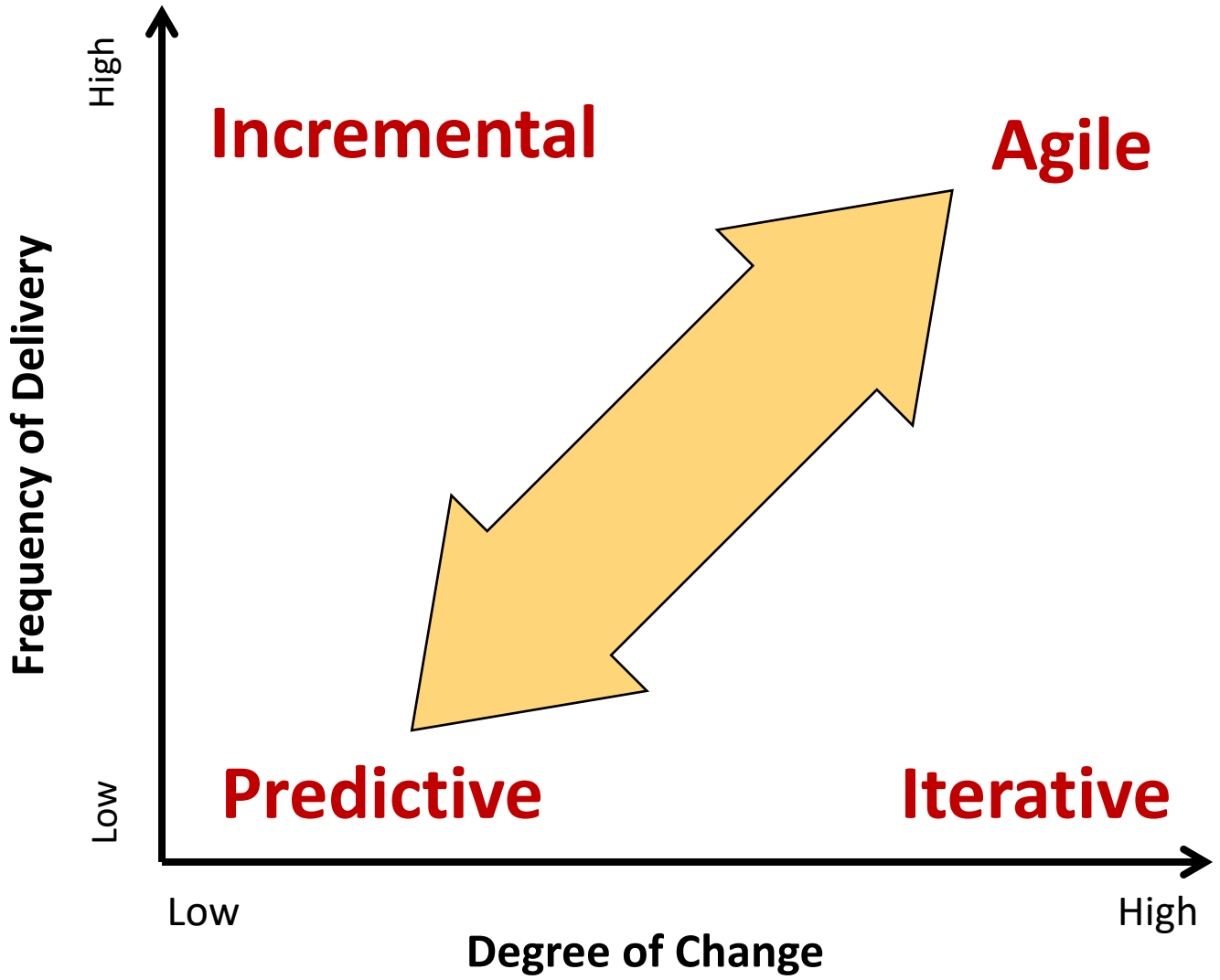
Source: Ian Sommerville (2015), Software Engineering, 10th Edition, Pearson.

Plan-based and Agile development



Source: Ian Sommerville (2015), Software Engineering, 10th Edition, Pearson.

The Continuum of Life Cycles

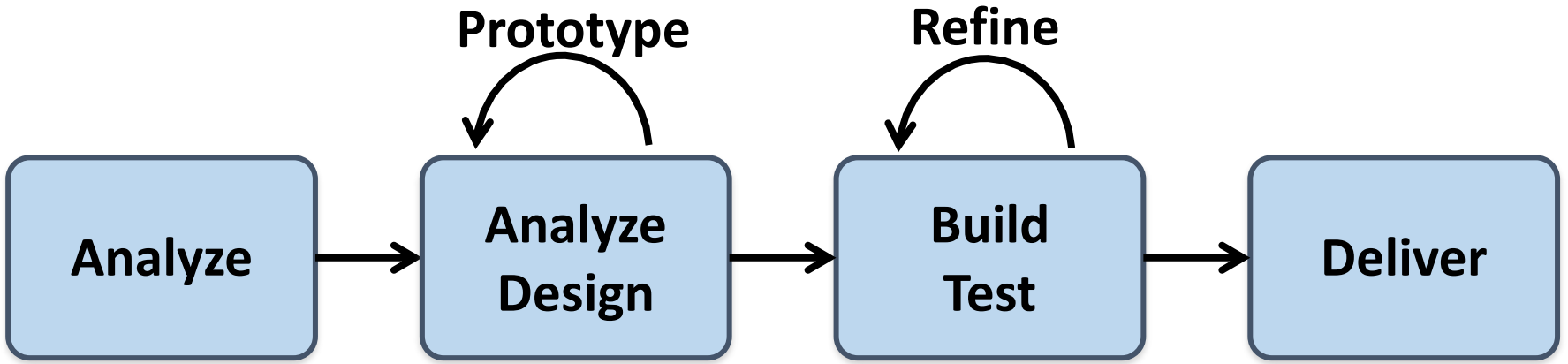


Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

Predictive Life Cycle

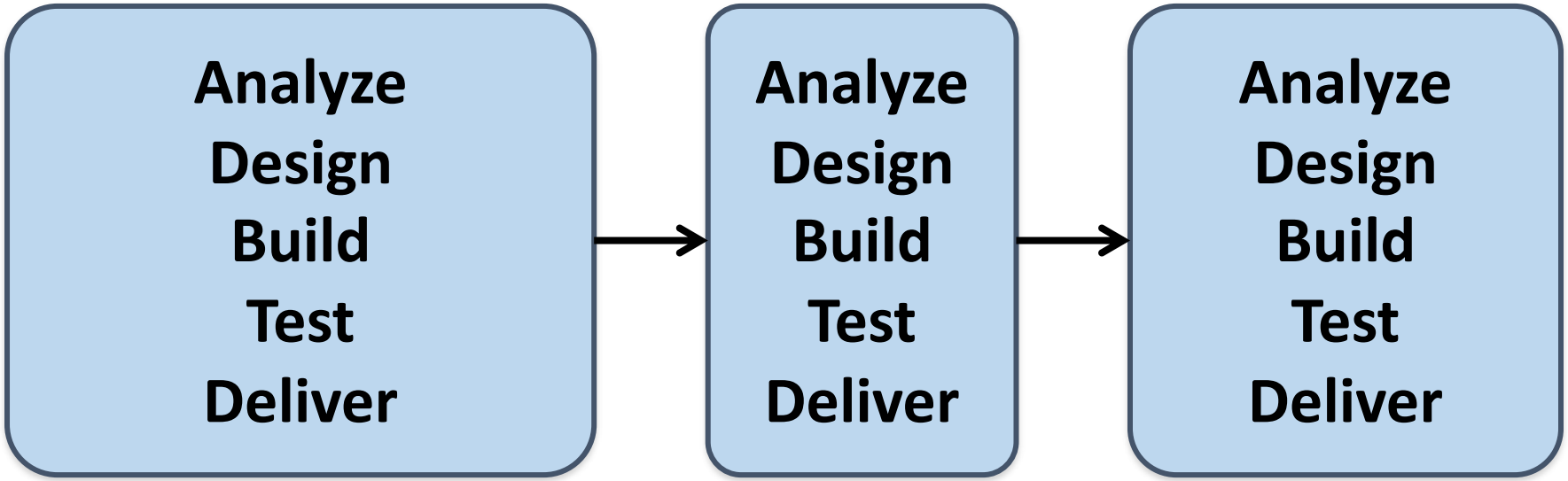
Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

Iterative Life Cycle



Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

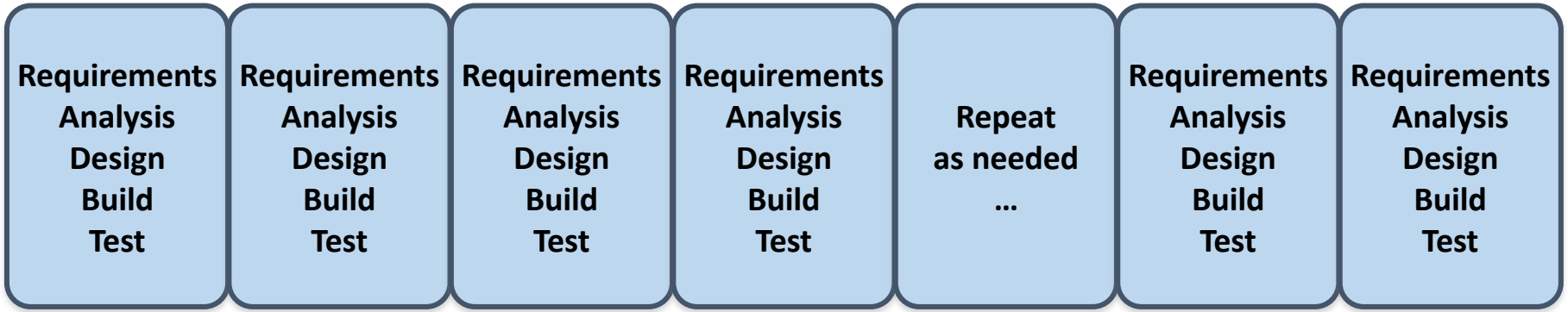
A Life Cycle of Varying-Sized Increments



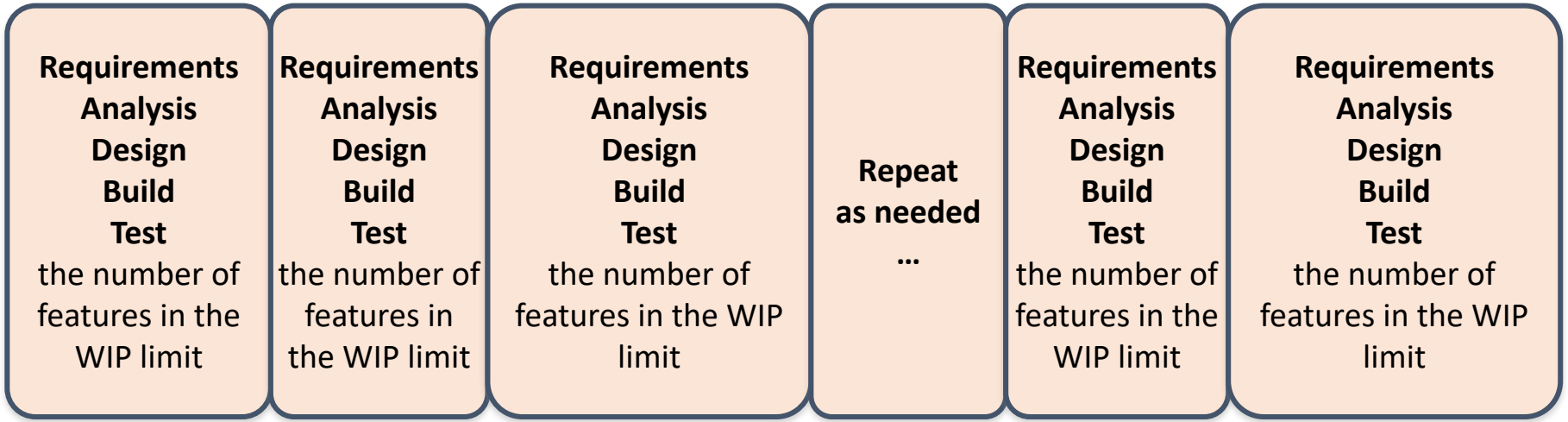
Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

Iteration-Based and Flow-Based Agile Life Cycles

Iteration-Based Agile

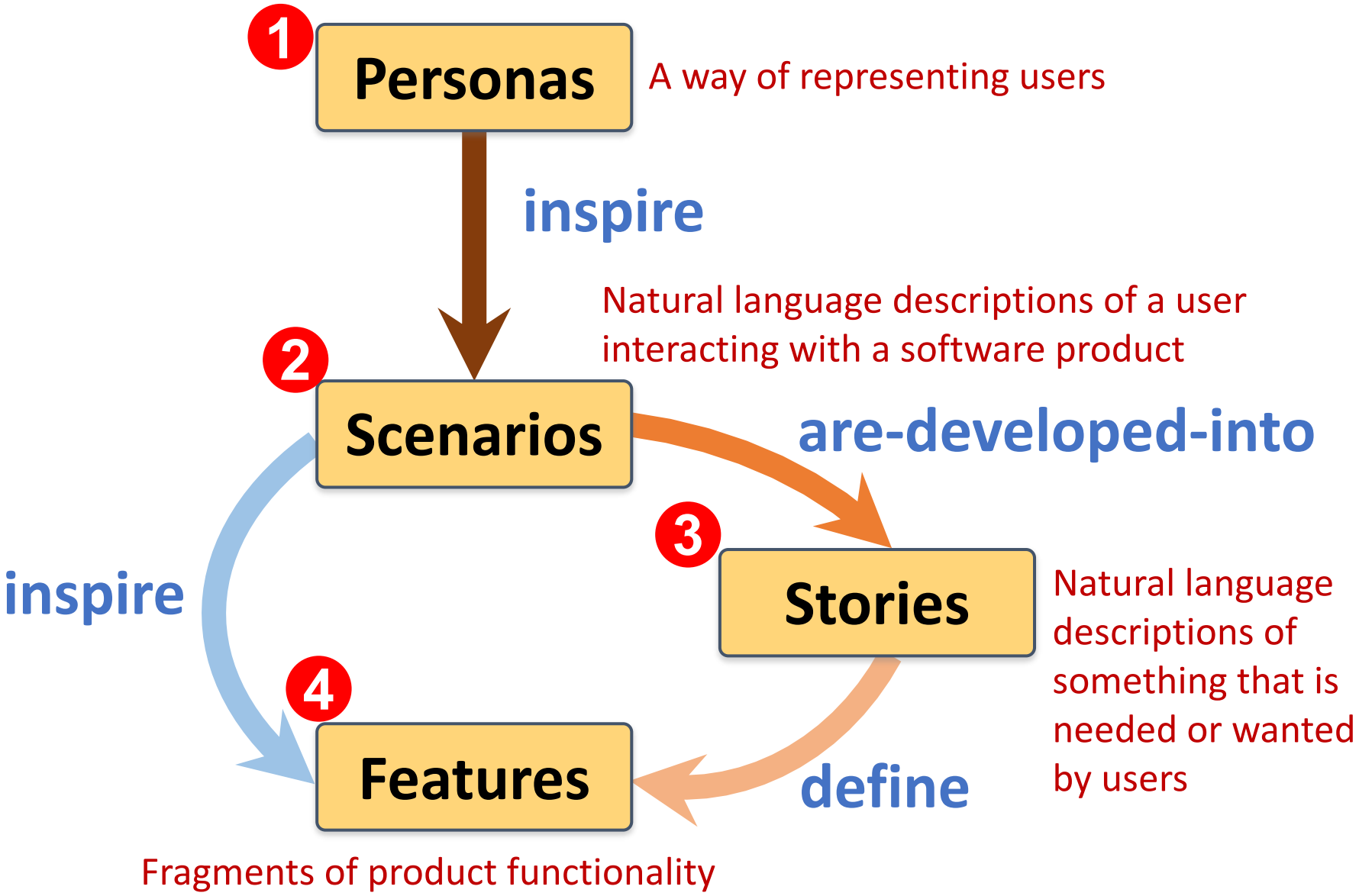


Flow-Based Agile



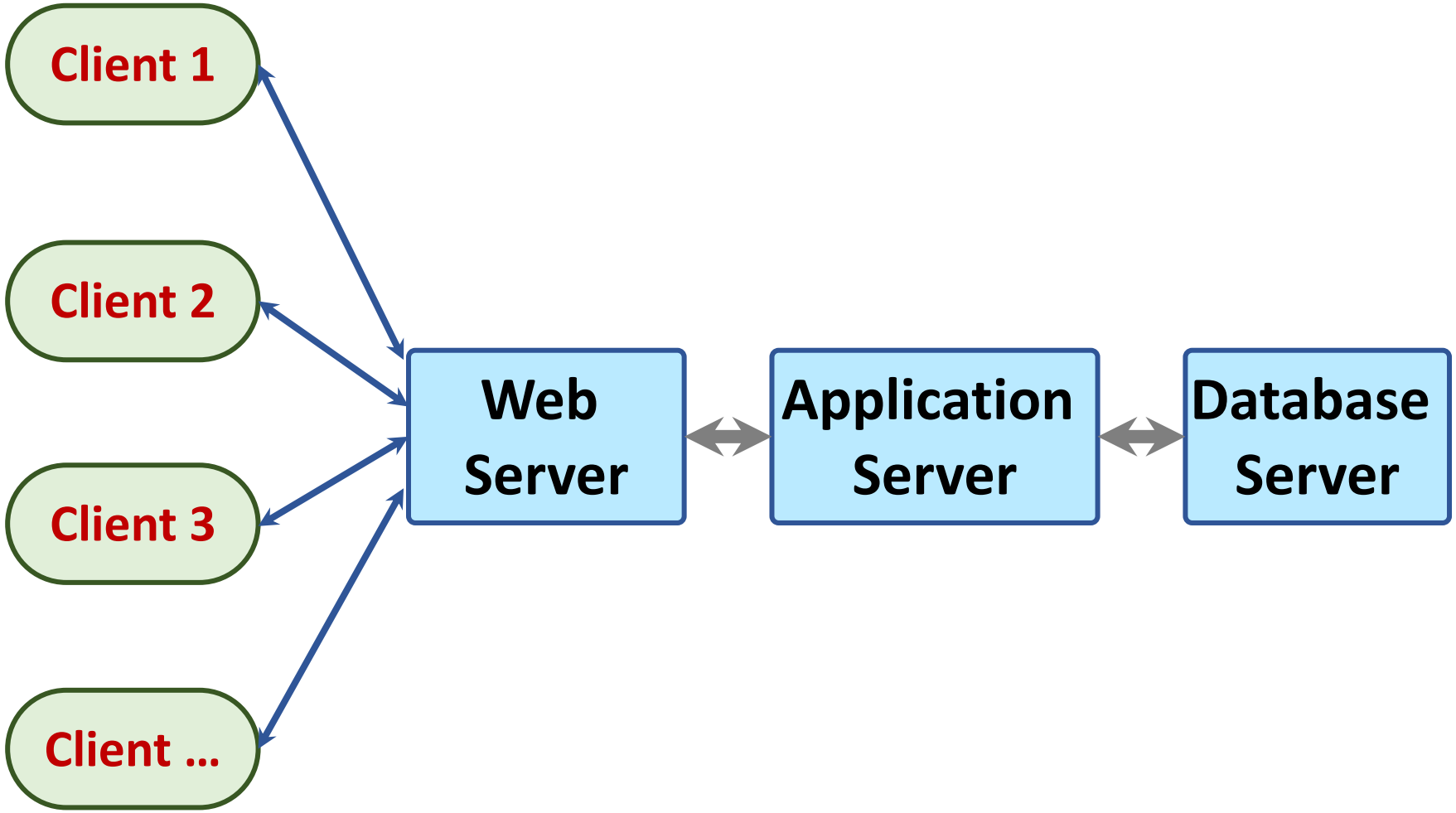
Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

From personas to features



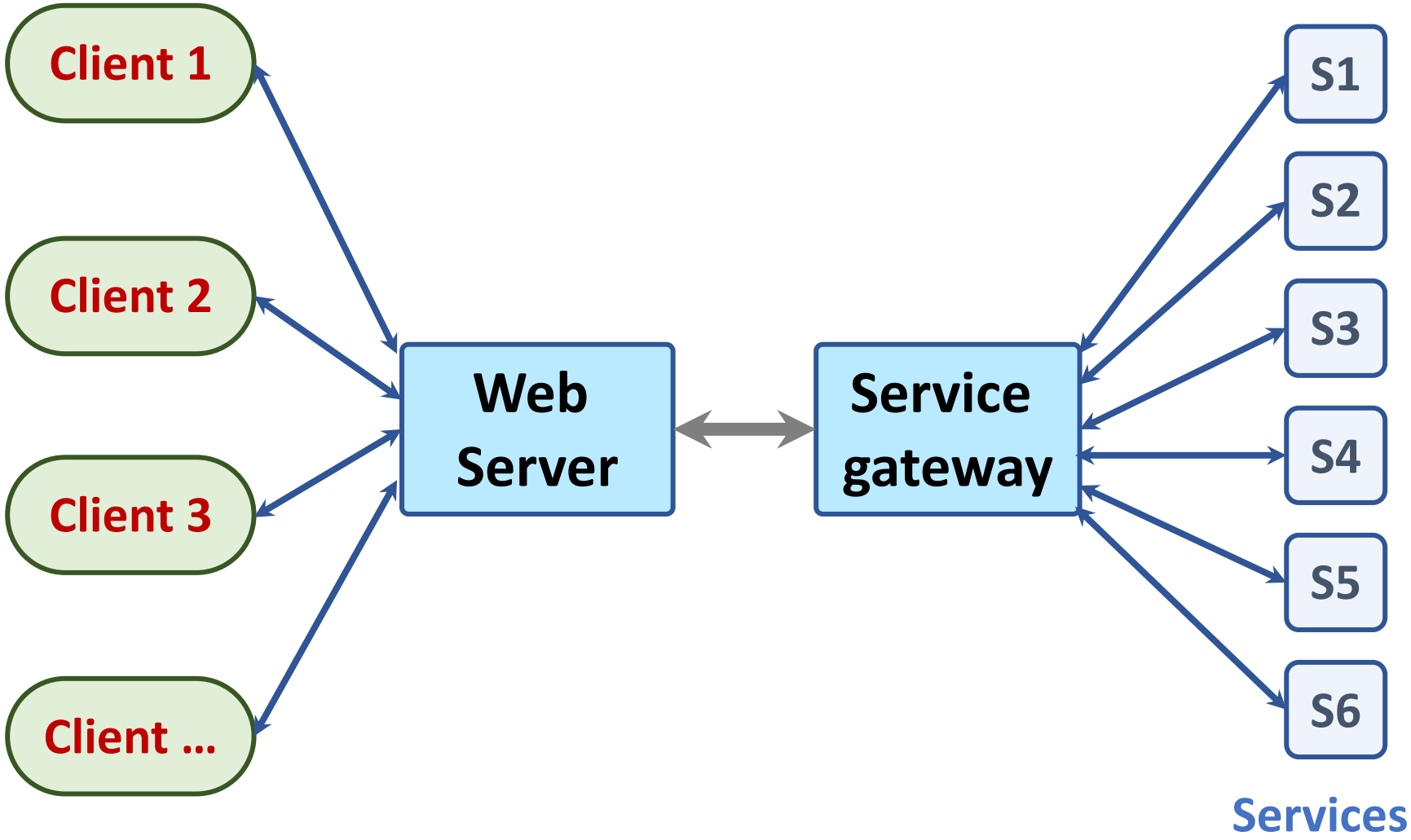
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Multi-tier client-server architecture

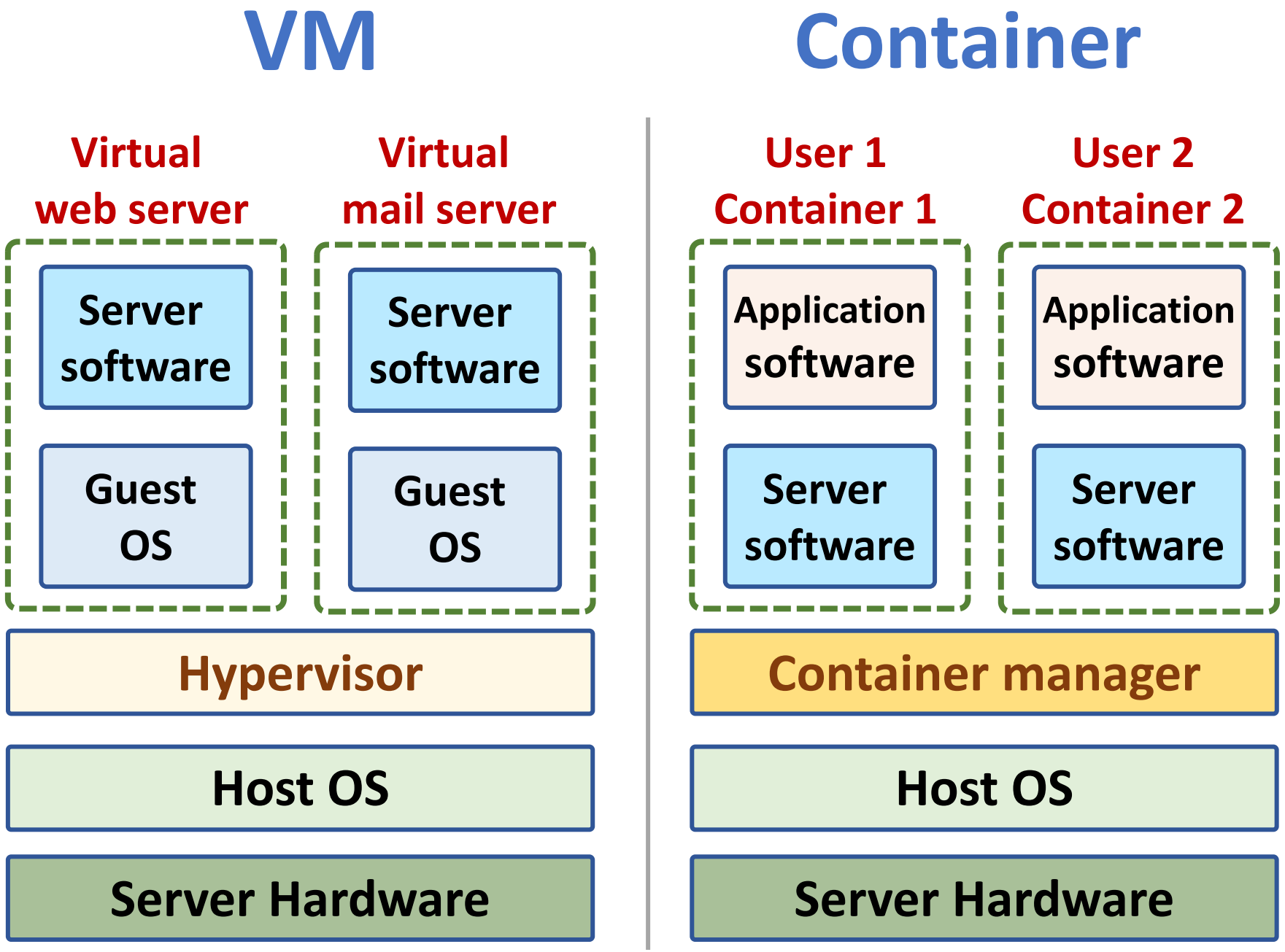


Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Service-oriented Architecture

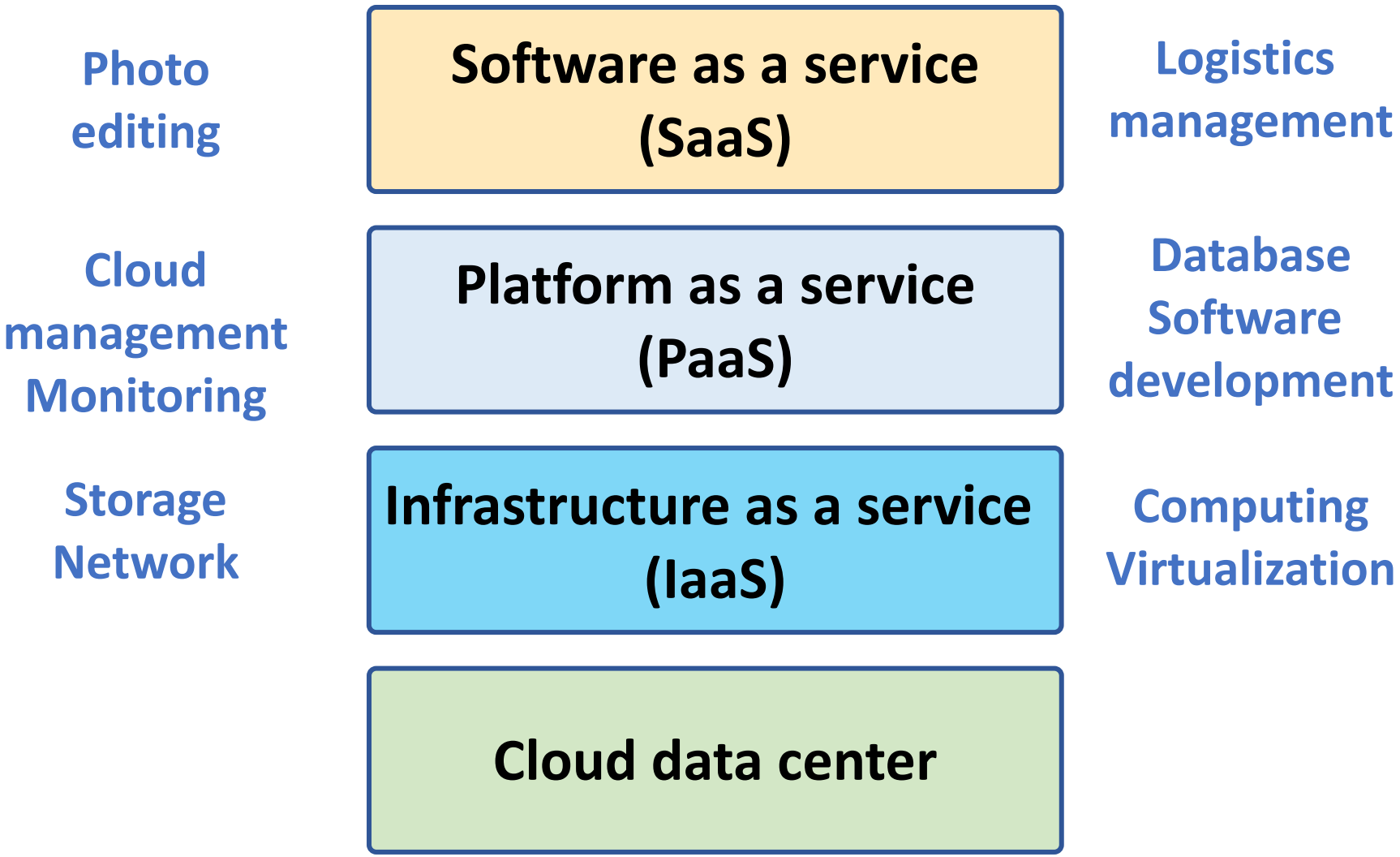


Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.



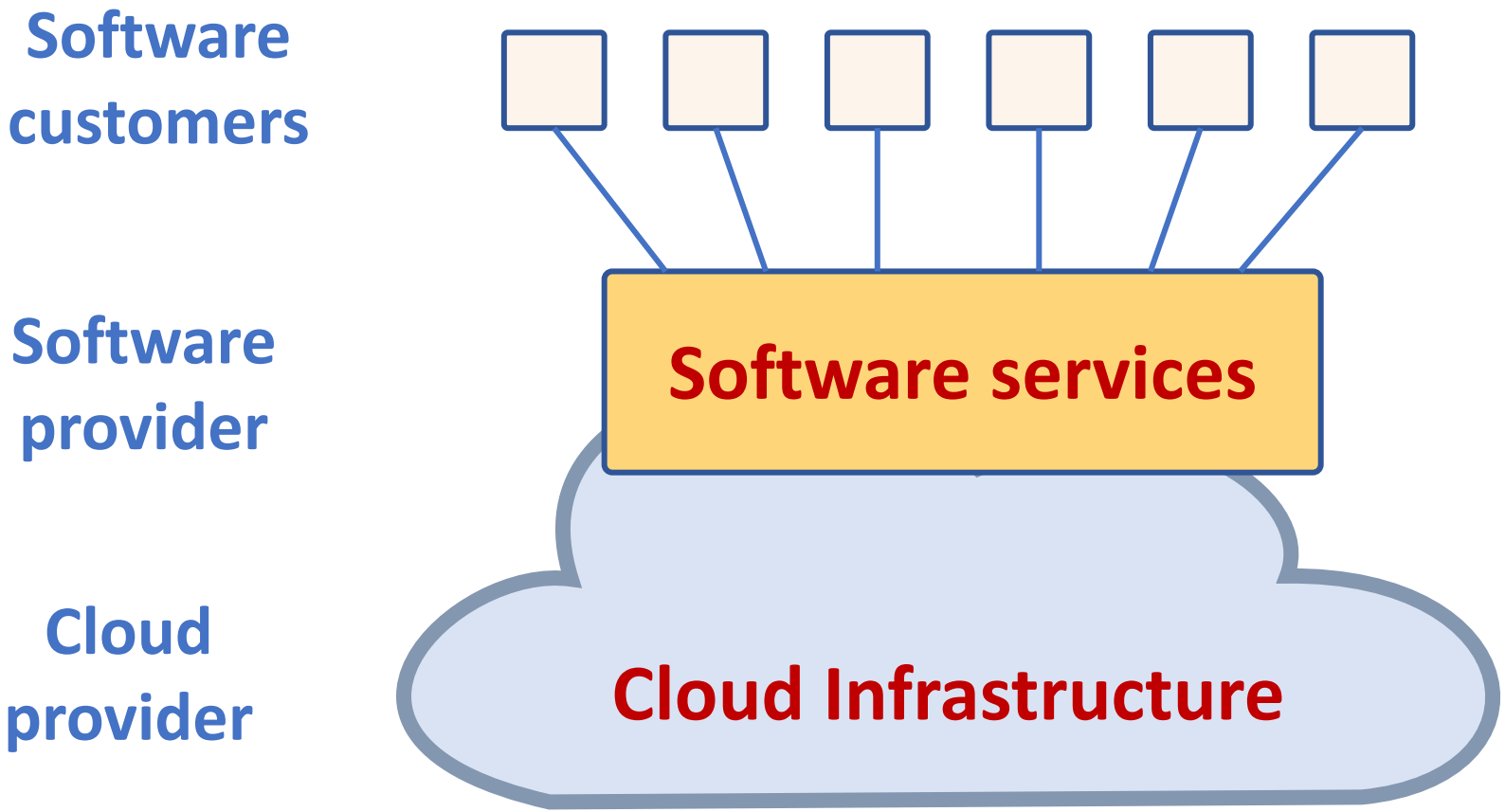
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Everything as a service



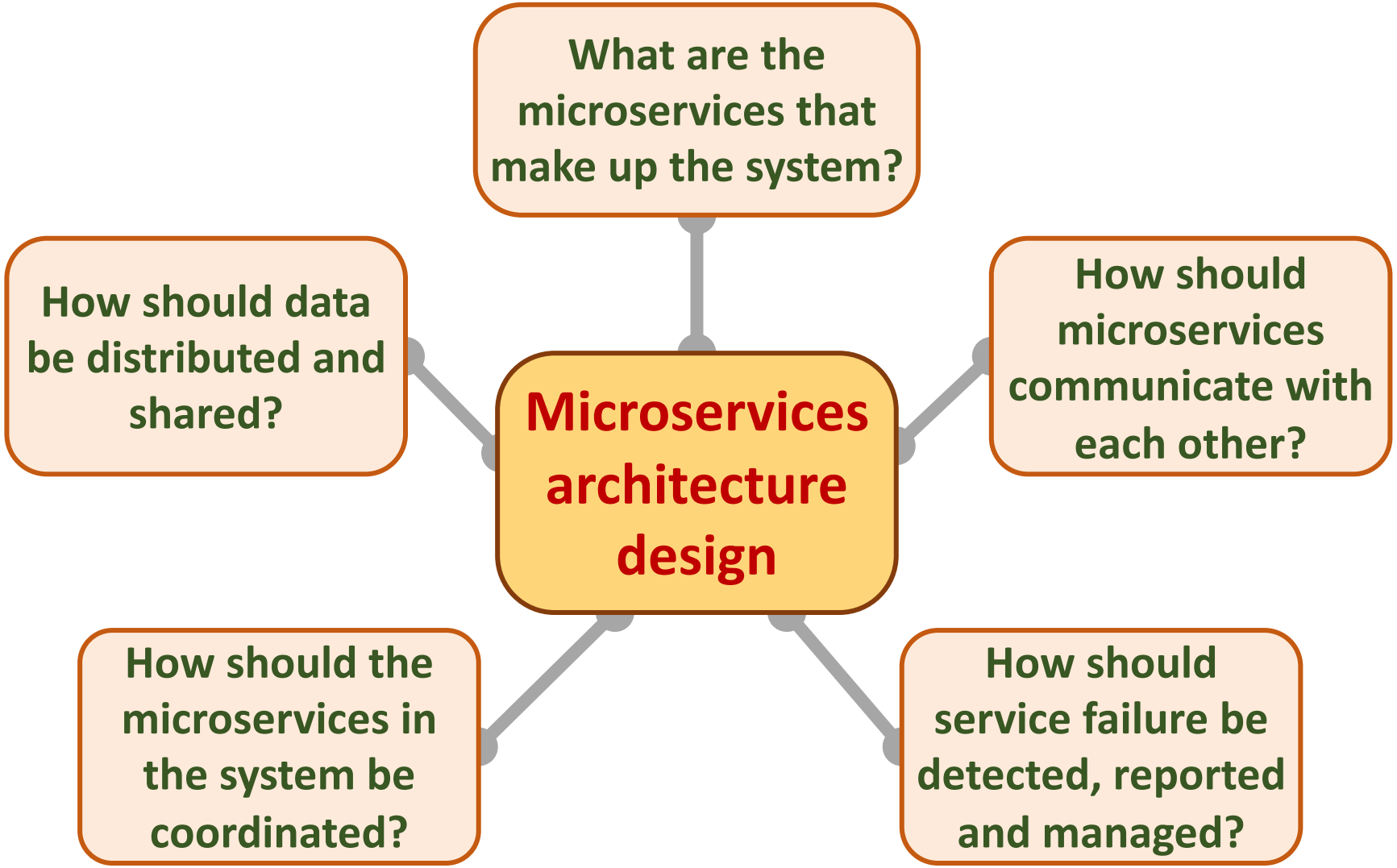
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Software as a service



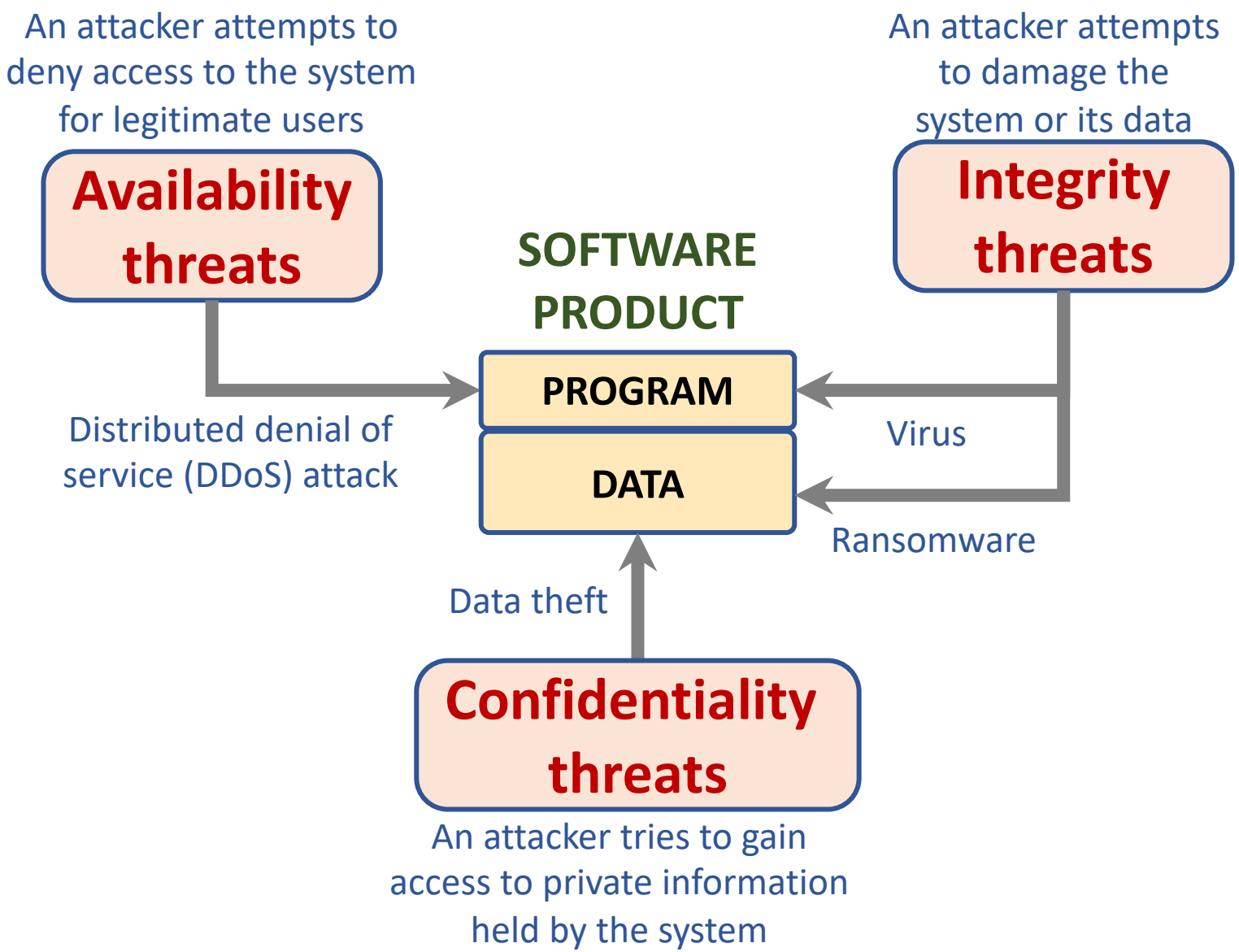
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Microservices architecture – key design questions



Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Types of security threat

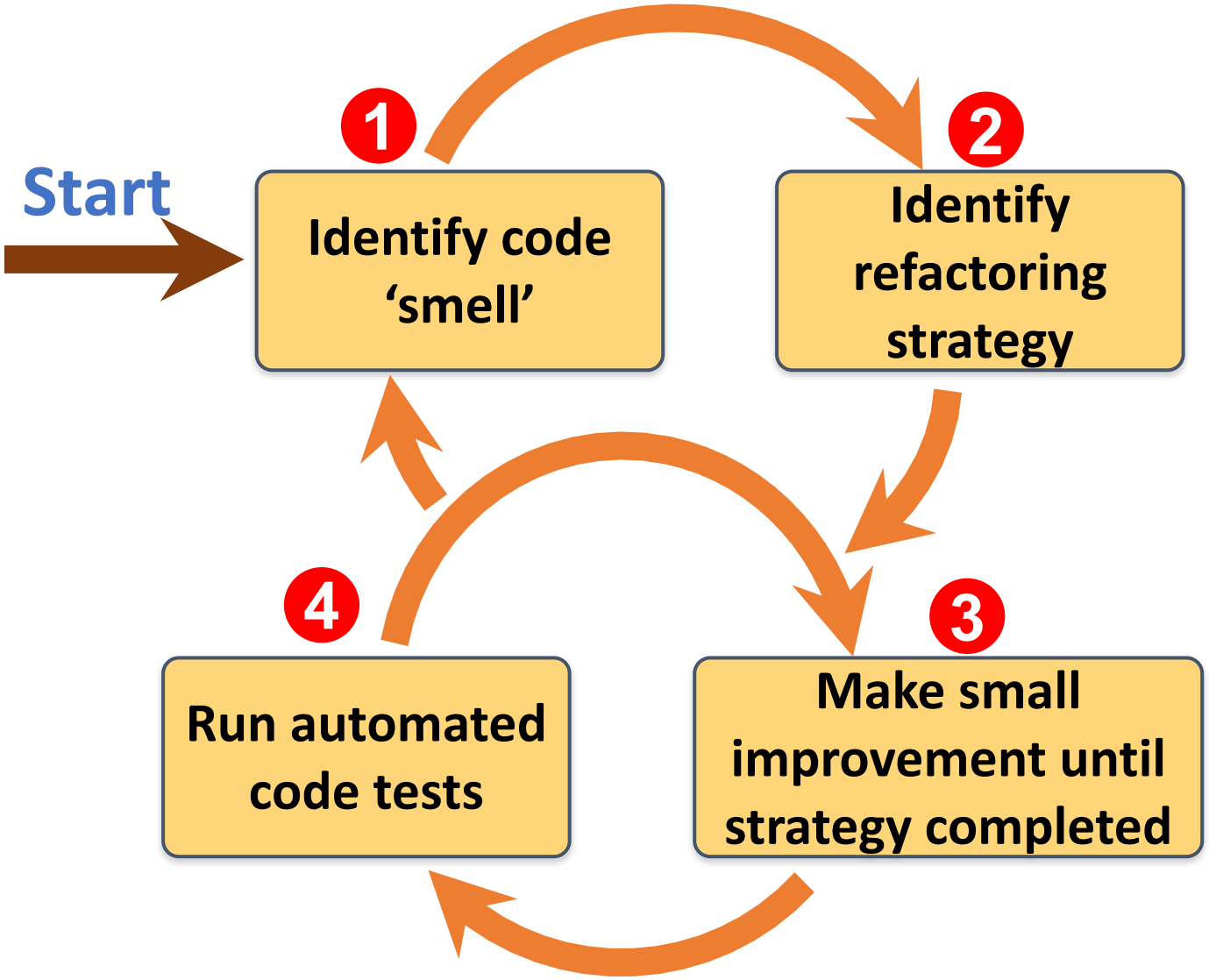


Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Software product quality attributes

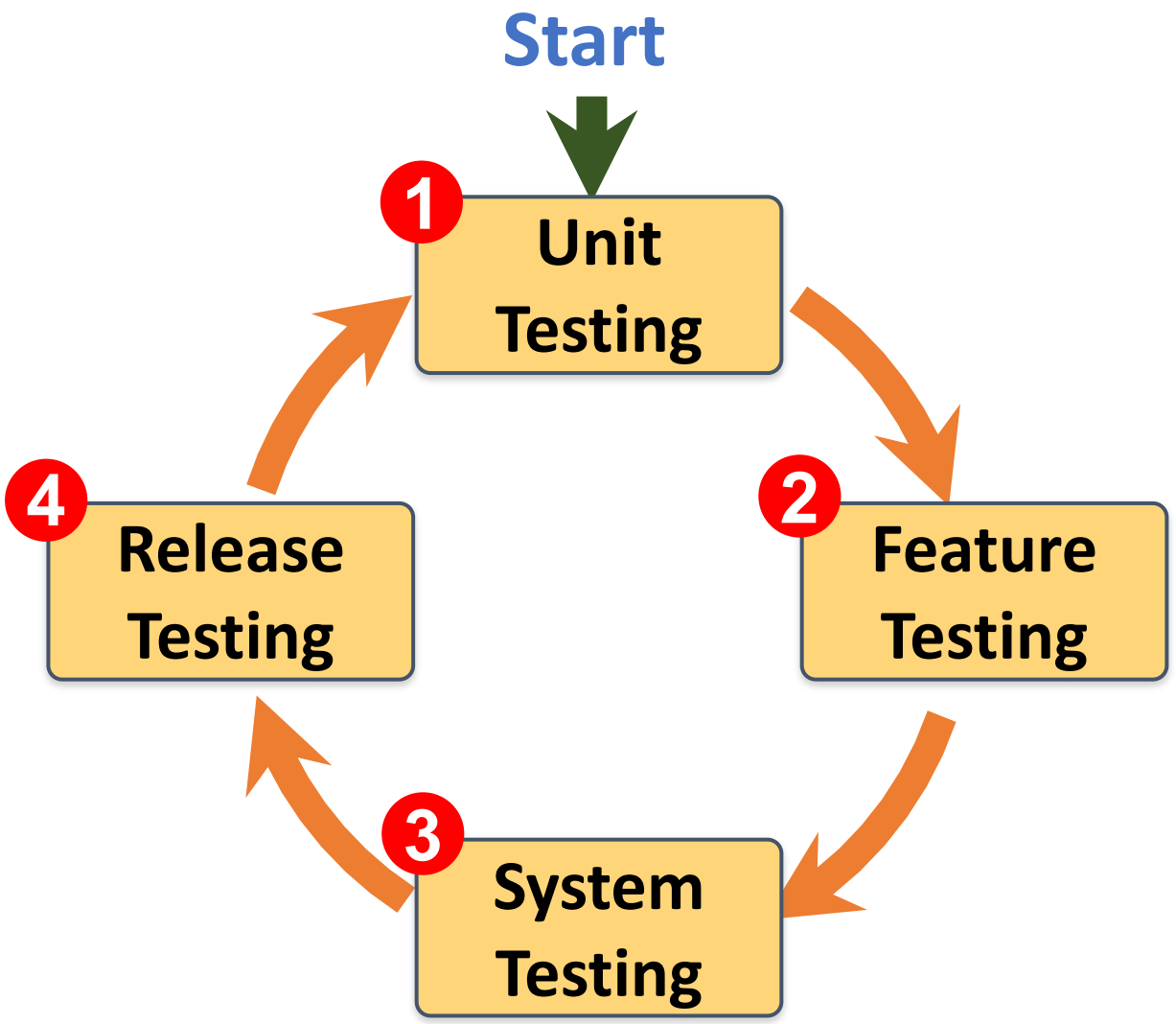
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

A refactoring process



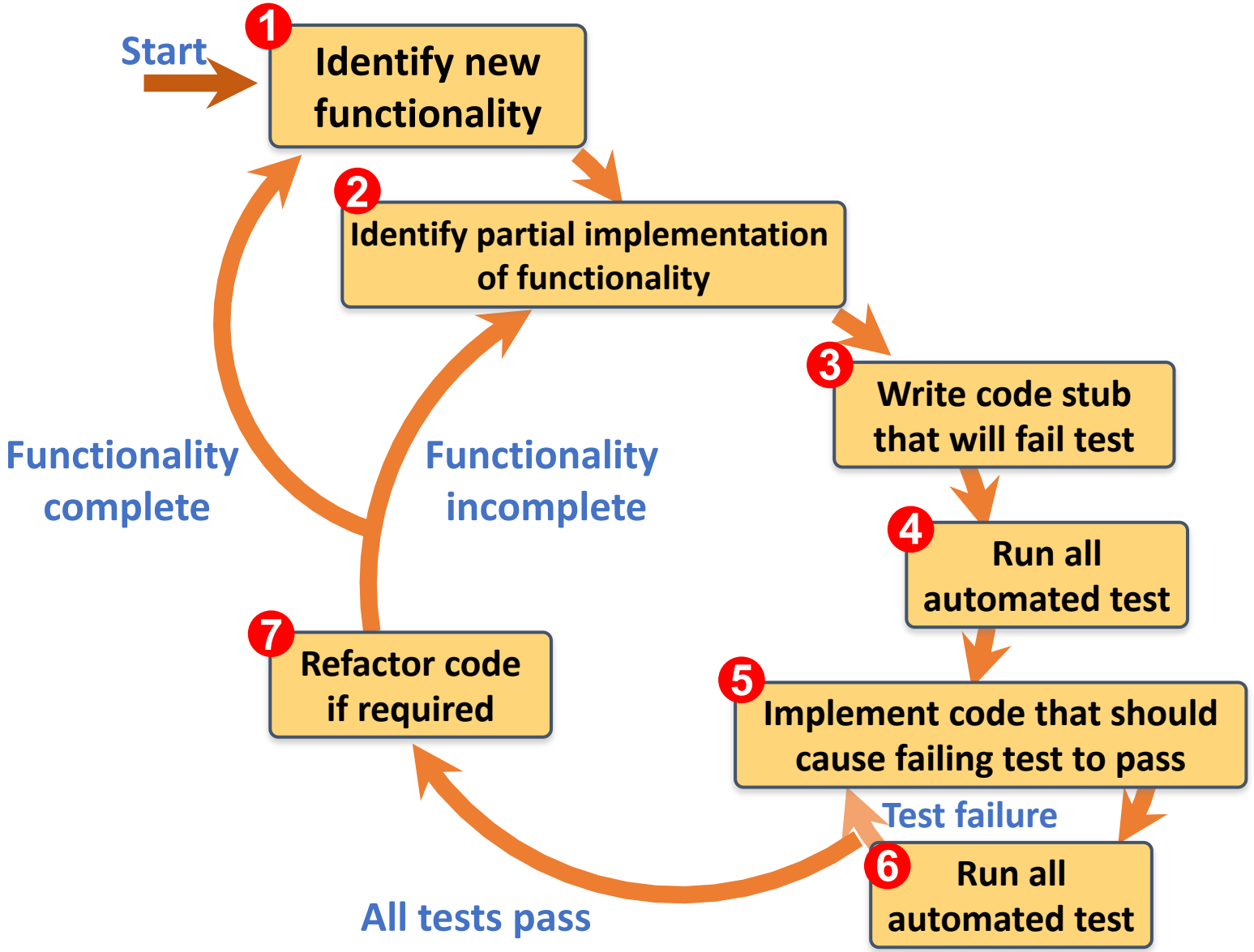
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Functional testing



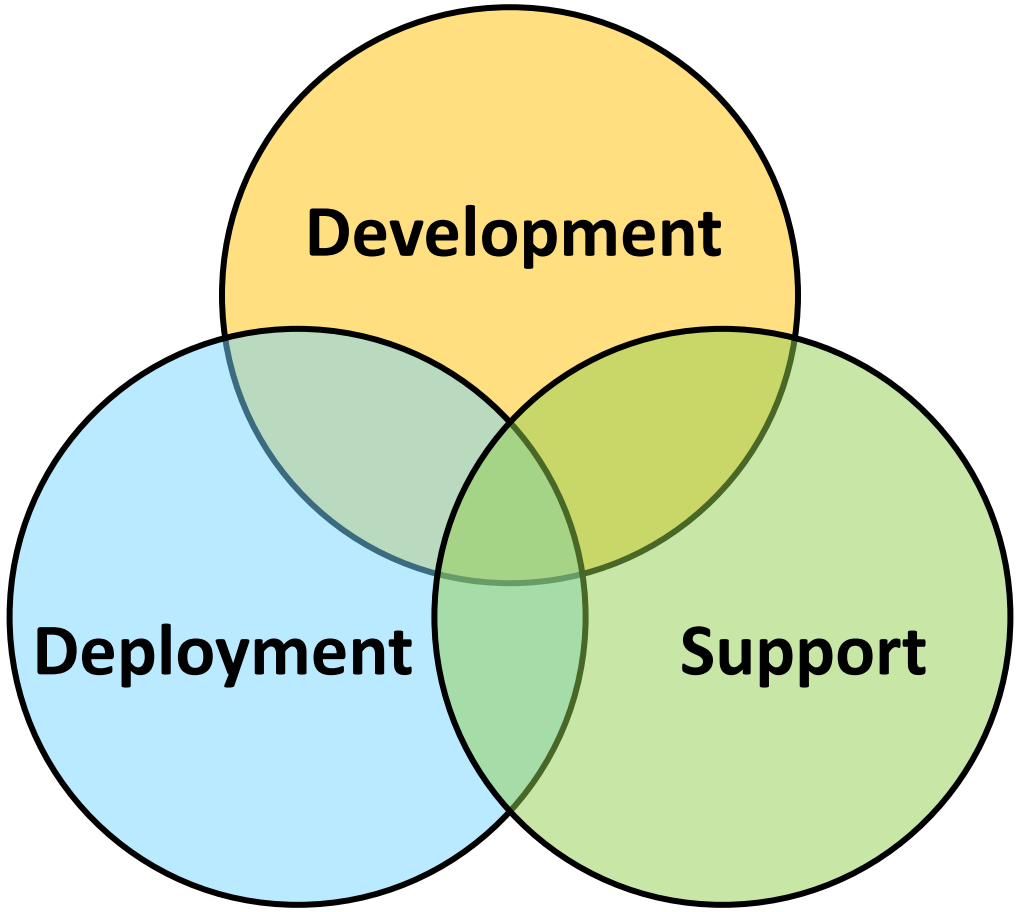
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Test-driven development (TDD)



Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

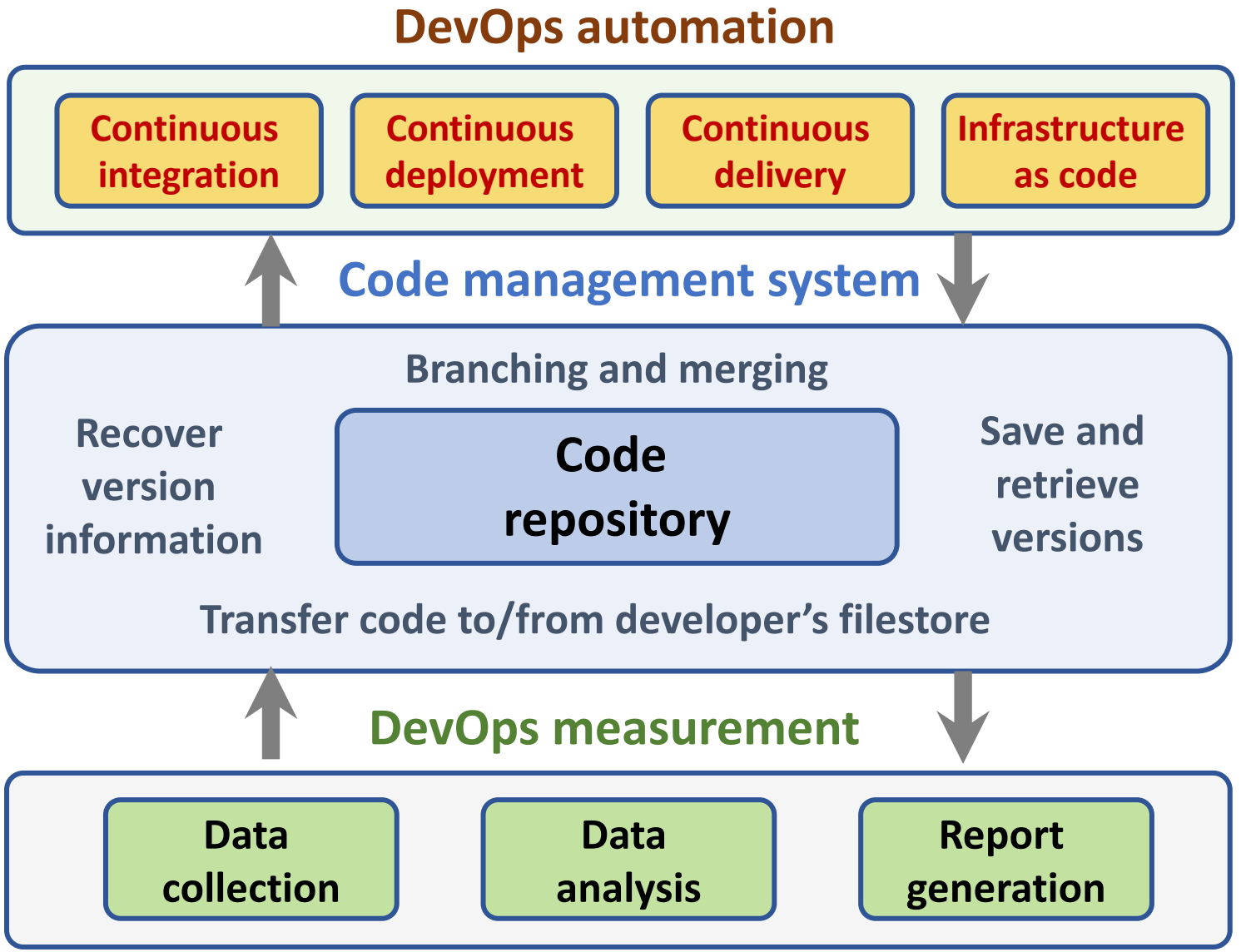
DevOps



Multi-skilled DevOps team

Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Code management and DevOps



Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Marketing

Marketing

“Meeting

needs

profitably”

Source: Philip Kotler and Kevin Lane Keller (2016), Marketing Management, 15th edition, Pearson.

Marketing

"Marketing is an organizational function
and a set of processes for
creating, communicating, and delivering
value to customers and
for managing customer relationships
in ways that benefit the organization and its stakeholders."

Source: Philip Kotler and Kevin Lane Keller (2016), Marketing Management, 15th edition, Pearson.

Marketing Management

Marketing Management

"Marketing management is the art and science of choosing target markets and getting, keeping, and growing customers through creating, delivering, and communicating superior customer value."

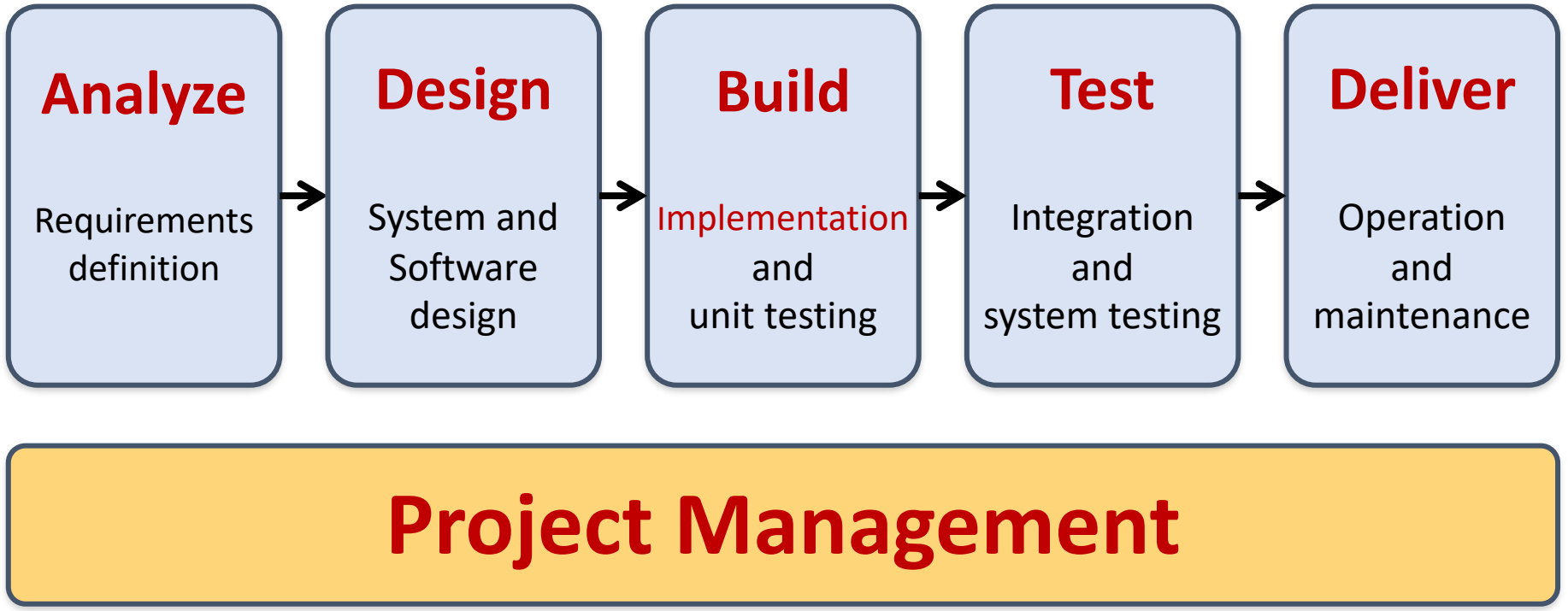
Source: Philip Kotler and Kevin Lane Keller (2016), Marketing Management, 15th edition, Pearson.

Marketing Management

1. Understanding Marketing Management
2. Capturing Marketing Insights
3. Connecting with Customers
4. Building Strong Brands
5. Creating Value
6. Delivering Value
7. Communicating Value
8. Conducting Marketing Responsibly for Long-term Success

Source: Philip Kotler and Kevin Lane Keller (2016), Marketing Management, 15th edition, Pearson.

Software Engineering
and
Project Management



Summary

- **This course introduces the fundamental concepts, research issues, and hands-on practices of software engineering.**
- **Topics include:**
  1. Introduction to Software Engineering
  2. Software Products and Project Management: Software product management and prototyping
  3. Agile Software Engineering: Agile methods, Scrum, and Extreme Programming
  4. Features, Scenarios, and Stories
  5. Software Architecture: Architectural design, System decomposition, and Distribution architecture
  6. Cloud-Based Software: Virtualization and containers, Everything as a service, Software as a service
  7. Cloud Computing and Cloud Software Architecture
  8. Microservices Architecture, RESTful services, Service deployment
  9. Security and Privacy; Reliable Programming
  10. Testing: Functional testing, Test automation, Test-driven development, and Code reviews
  11. DevOps and Code Management: Code management and DevOps automation
  12. Case Study on Software Engineering

Software Engineering

國立臺北大學
National Taipei University

aws educate | Cloud Ambassador
2020 Cohort

aws academy
Accredited Educator

aws certified
Solutions Architect
Associate

aws certified
Cloud Practitioner

Contact Information

Min-Yuh Day, Ph.D.

Associate Professor

Institute of Information Management, National Taipei University

Tel: 02-86741111 ext. 66873

Office: B8F12

Address: 151, University Rd., San Shia District, New Taipei City, 23741 Taiwan

Email: myday@gm.ntpu.edu.tw

Web: http://web.ntpu.edu.tw/~myday/