軟體工程
(Software Engineering)

軟體工程概論
(Introduction to Software Engineering)

1101SE01
MBA, IM, NTPU (M6131) (Fall 2021)
Thu 11, 12, 13 (19:25-22:10) (209)

Min-Yuh Day
戴敏育
Associate Professor
副教授
Institute of Information Management, National Taipei University
國立臺北大學 資訊管理研究所
https://web.ntpu.edu.tw/~myday

2021-09-23

戴敏育 博士
(Min-Yuh Day, Ph.D.)

國立台北大學 資訊管理研究所 副教授

中央研究院 資訊科學研究所 訪問學人

國立台灣大學 資訊管理 博士

Publications Co-Chairs, IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013-)

Program Co-Chair, IEEE International Workshop on Empirical Methods for Recognizing Inference in TExt (IEEE EM-RITE 2012-)

Publications Chair, The IEEE International Conference on Information Reuse and Integration (IEEE IRI)

aws educate | Cloud Ambassador
2020 Cohort

aws academy Accredited Educator

aws certified Solutions Architect Associate

aws certified Cloud Practitioner

軟體工程
(Software Engineering)
Contact Information

戴敏育 博士 (Min-Yuh Day, Ph.D.)

副教授 (Associate Professor)

國立臺北大學 資訊管理研究所

Institute of Information Management, National Taipei University

電話： 02-86741111 ext. 66873

研究室： 商8F12

地址： 23741 新北市三峽區大學路 151 號

Email：myday@gm.ntpu.edu.tw

網址：http://web.ntpu.edu.tw/~myday/

國立臺北大學
110學年度第1學期
課程大綱
Fall 2021 (2021.09 - 2022.01)

- 課程名稱：**軟體工程 (Software Engineering)**
- 授課教師：戴敏育 (Min-Yuh Day)
- 開課系所：資管所碩士班 (資訊碩2; 電子商務碩士學程)
- 開課資料：選修 半學年 3 學分 (3 Credits, Elective)
- 上課時間：週四 11, 12, 13 (19:25-22:10)
- 上課教室：209 (台北大學民生校區)

Google Meet: http://meet.google.com/zqs-amev-anr

教學目標

1. 瞭解軟體工程基本概念、與研究議題。
2. 具備軟體工程實務操作能力。
3. 進行軟體工程相關之資訊管理研究。

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.

內容綱要

- 本課程介紹軟體工程基本概念、研究議題、與實務操作。
- 課程內容包括
  1. 軟體工程概論
  2. 軟體產品與專案管理：軟體產品管理，原型設計
  3. 敏捷軟體工程：敏捷方法、Scrum、極限程式設計
  4. 功能、場景和故事
  5. 軟體架構：架構設計、系統分解、分散式架構
  6. 基於雲的軟體：虛擬化和容器、軟體即服務
  7. 雲端運算與雲軟體架構
  8. 微服務架構：RESTful服務、服務部署
  9. 安全和隱私; 可靠的程式設計
  10. 測試：功能測試、測試自動化、測試驅動的開發、程式審查
  11. DevOps和程式碼管理：程式碼管理和DevOps自動化
  12. 軟體工程個案研究

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

資訊管理研究所
系核心能力
(Core Competence)

- 資訊科技新知探索與系統開發應用 90％
- 網路行銷企劃能力
- 論文寫作與獨立研究能力 10％

校四大基本素養
(Four Fundamental Qualities)

- 專業 (Professionalism)
  - 創意思考與問題解決 (Creative thinking and Problem-solving) 30 %
  - 綜合統整(Comprehensive Integration) 30 %
- 人際 (Interpersonal Relationship)
  - 溝通協調 (Communication and Coordination) 10 %
  - 團隊合作 (Teamwork) 10 %
- 倫理 (Ethics)
  - 誠信正直(Honesty and Integrity) 5 %
  - 尊重自省(Self-Esteem and Self-reflection) 5 %
- 國際觀 (International Vision)
  - 多元關懷 (Caring for Diversity) 5 %
  - 跨界宏觀 (Interdisciplinary Vision) 5 %

商學院學習目標
(College Learning Goals)

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

系所學習目標
(Department Learning Goals)

- Information Technologies and System Development Capabilities
- Research capabilities

課程大綱 (Syllabus)

國立臺北大學 National Taipei University

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
1	2021/09/23	軟體工程概論 (Introduction to Software Engineering)
2	2021/09/30	軟體產品與專案管理：軟體產品管理，原型設計 (Software Products and Project Management: Software product management and prototyping)
3	2021/10/07	敏捷軟體工程：敏捷方法、Scrum、極限程式設計 (Agile Software Engineering: Agile methods, Scrum, and Extreme Programming)
4	2021/10/14	功能、場景和故事 (Features, Scenarios, and Stories)
5	2021/10/21	軟體工程個案研究 I (Case Study on Software Engineering I)
6	2021/10/28	軟體架構：架構設計、系統分解、分散式架構 (Software Architecture: Architectural design, System decomposition, and Distribution architecture)

課程大綱 (Syllabus)

國立臺北大學 National Taipei University

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
7	2021/11/04	基於雲的軟體：虛擬化和容器、軟體即服務 (Cloud-Based Software: Virtualization and containers, Everything as a service, Software as a service)
8	2021/11/11	期中報告 (Midterm Project Report)
9	2021/11/18	雲端運算與雲軟體架構 (Cloud Computing and Cloud Software Architecture)
10	2021/11/25	微服務架構：RESTful服務、服務部署 (Microservices Architecture, RESTful services, Service deployment)
11	2021/12/02	軟體工程產業實務 (Industry Practices of Software Engineering)
12	2021/12/09	軟體工程個案研究 II (Case Study on Software Engineering II)

課程大綱 (Syllabus)

國立臺北大學 National Taipei University

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
13	2021/12/16	安全和隱私 (Security and Privacy); 可靠的程式設計 (Reliable Programming)
14	2021/12/23	測試：功能測試、測試自動化、 測試驅動的開發、程式碼審查 (Testing: Functional testing, Test automation, Test-driven development, and Code reviews); DevOps和程式碼管理：程式碼管理和DevOps自動化 (DevOps and Code Management: Code management and DevOps automation)
15	2021/12/30	期末報告 I (Final Project Report I)
16	2022/01/06	期末報告 II (Final Project Report II)
17	2022/01/13	學生自主學習 (Self-learning)
18	2022/01/20	學生自主學習 (Self-learning)

教學方法與教學活動
(Teaching methods and activities)

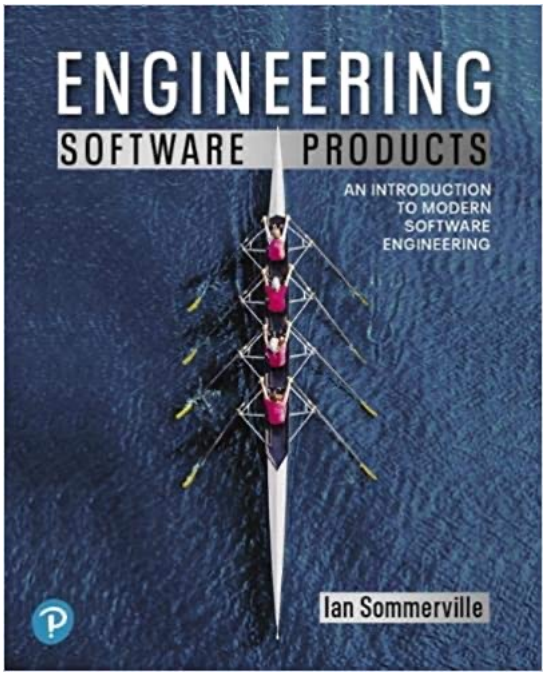
- 講授 (Lecture)
- 討論 (Discussion)
- 實習 (Practicum)

評量方式
(Evaluation Methods)

- 個人報告 (Individual Presentation) 60 %
- 團體報告 (Group Presentation) 10 %
- 個案分析報告 (Case Report) 10 %
- 課堂參與 (Class Participation) 10 %
- 作業 (Assignment) 10 %

指定用書
(Required Texts)

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



參考書目
(Reference Books)

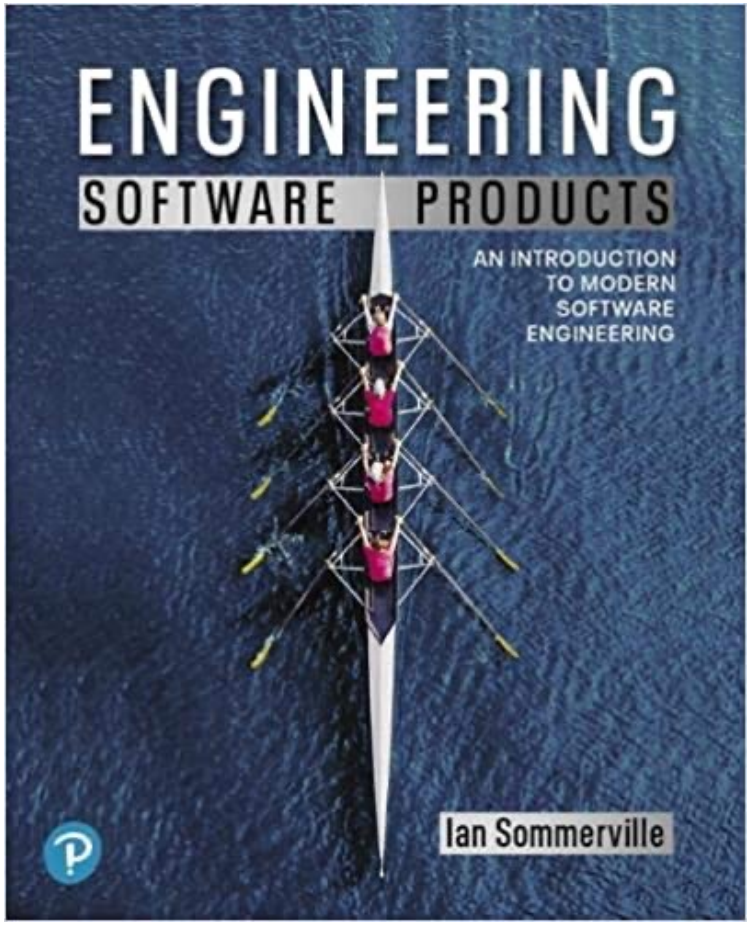
- Ian Sommerville (2015),
  Software Engineering,
  10th Edition, Pearson.
- Titus Winters, Tom Manshreck, and Hyrum Wright (2020),
  Software Engineering at Google: Lessons Learned from Programming Over Time, O'Reilly Media.
- Project Management Institute (2017),
  Agile Practice Guide, PMI
- Project 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.

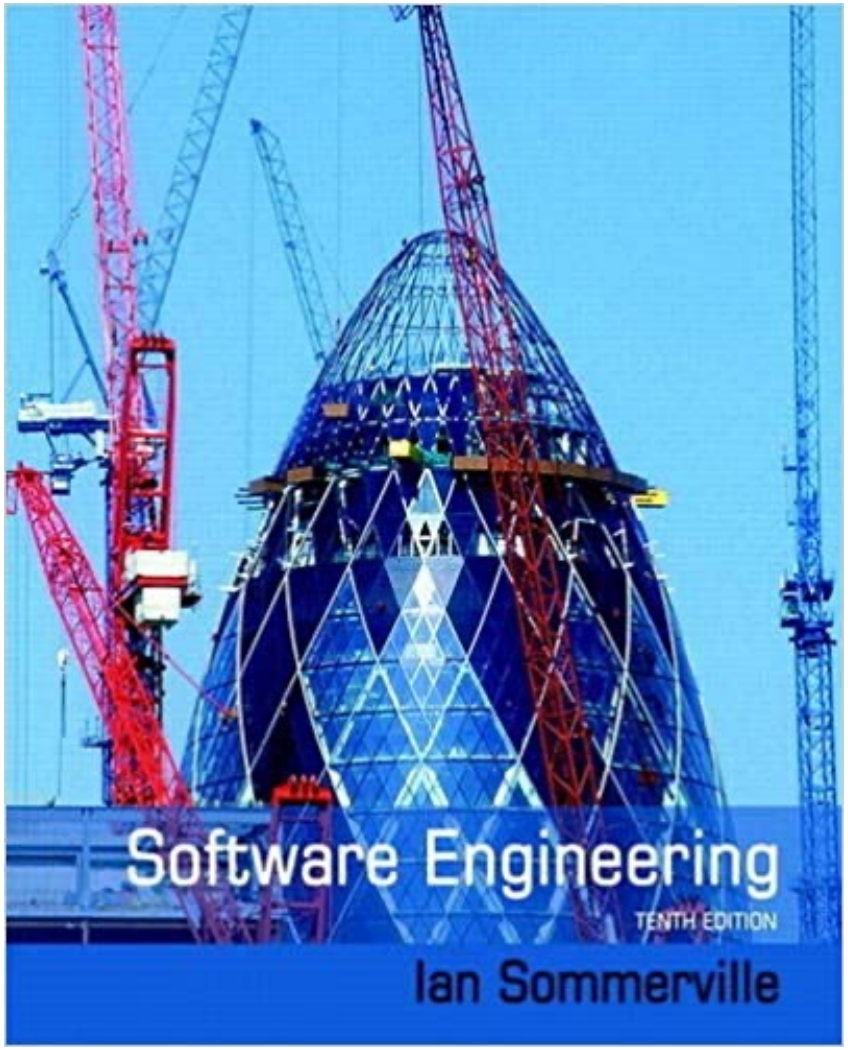


Source: https://www.amazon.com/Engineering-Software-Products-Ian-Sommerville/dp/013521064X

Ian Sommerville (2015),

Software Engineering,

10th Edition, Pearson.



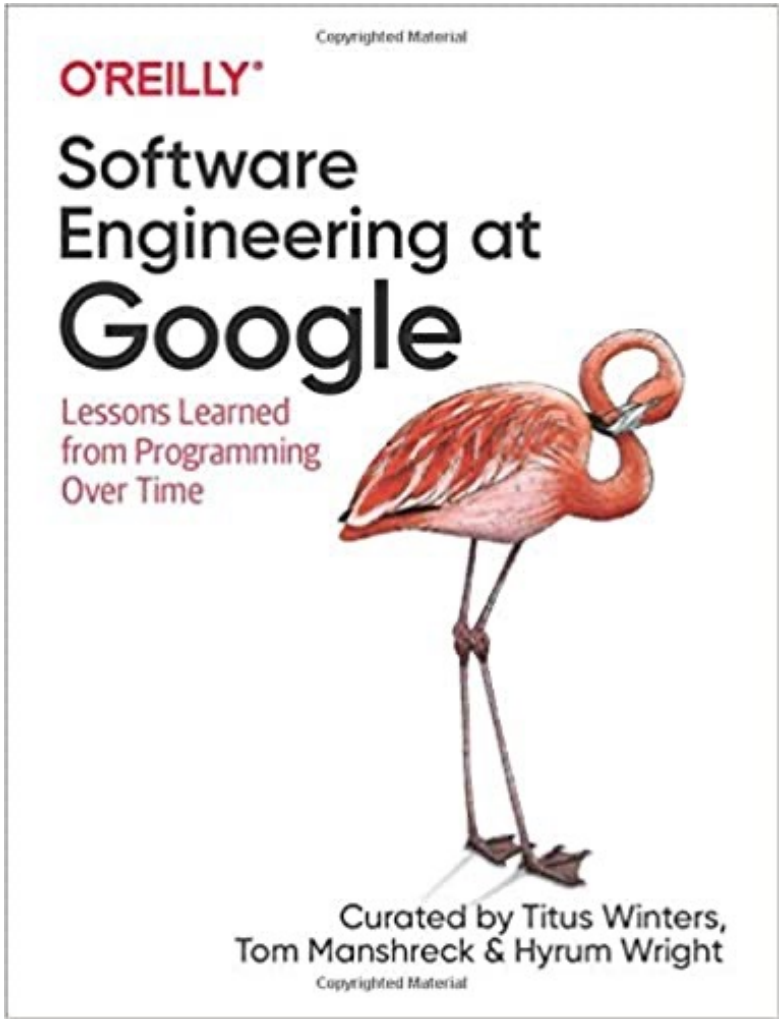
Source: https://www.amazon.com/Software-Engineering-10th-Ian-Sommerville/dp/0133943038

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

Software Engineering at Google:

Lessons Learned from Programming Over Time,

O'Reilly Media.

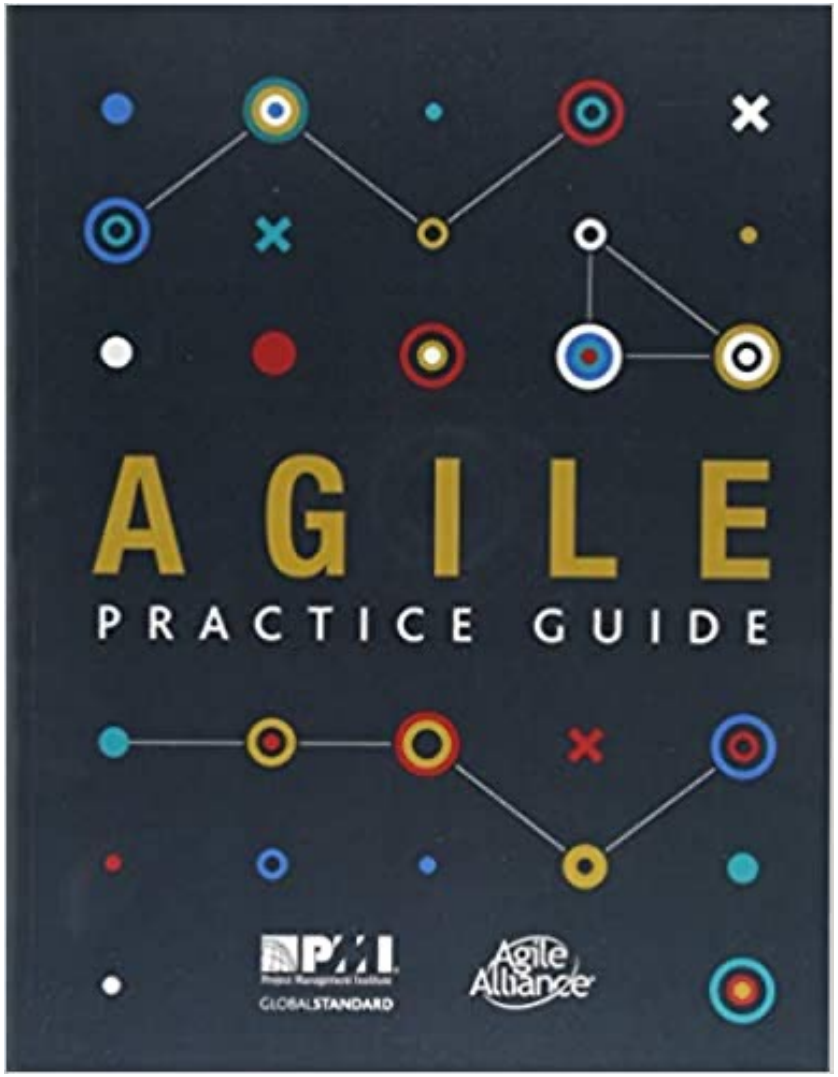


Source: https://www.amazon.com/Software-Engineering-Google-Lessons-Programming/dp/1492082791

Project Management Institute (2017),

Agile Practice Guide

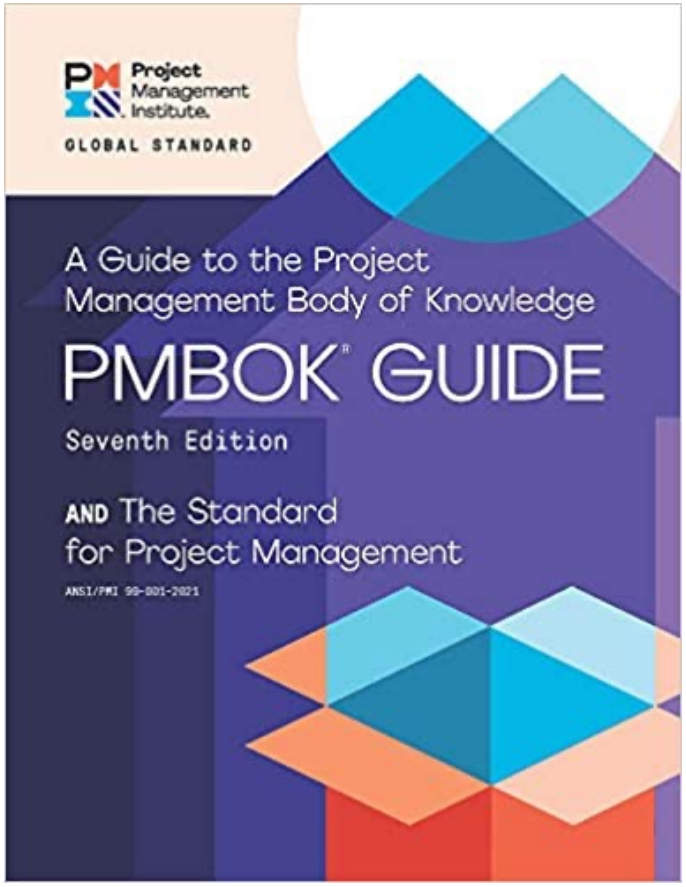
PMI



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

Project Management Institute (2021),

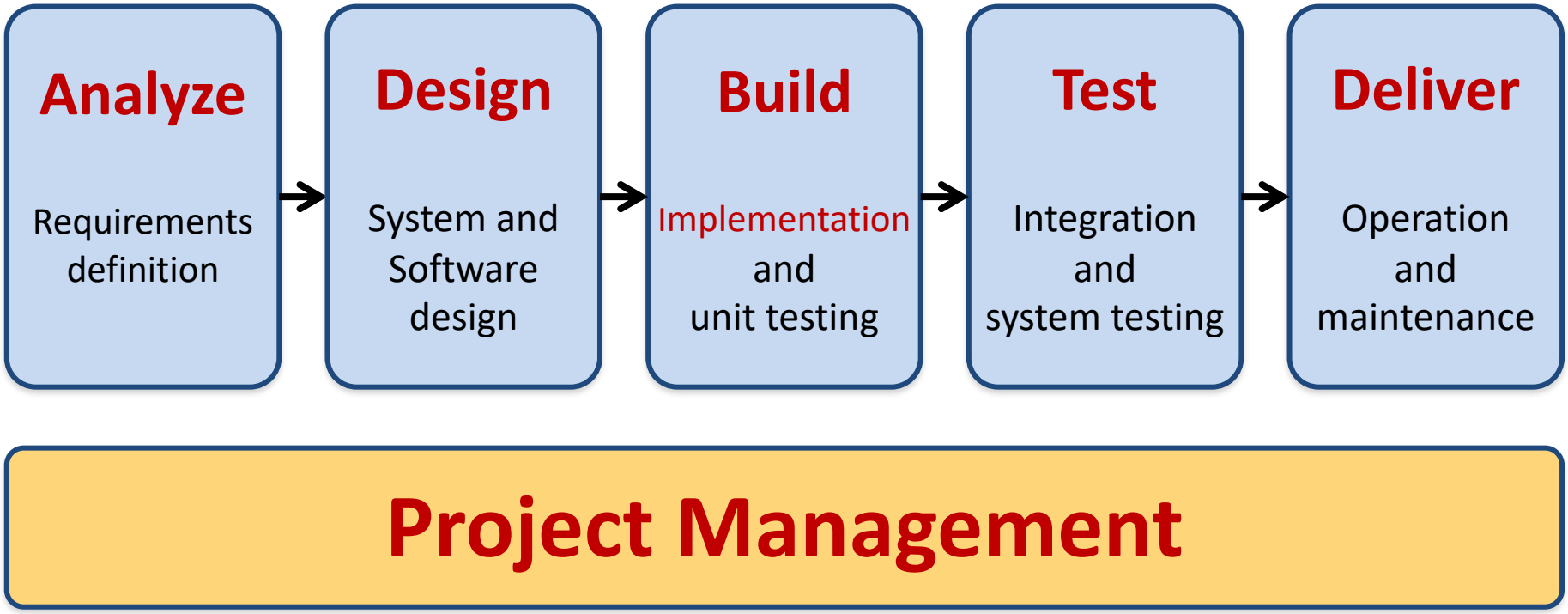
A Guide to the Project Management Body of Knowledge (PMBOK Guide) – Seventh Edition and The Standard for Project Management



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

Software Engineering

Software Engineering and Project Management

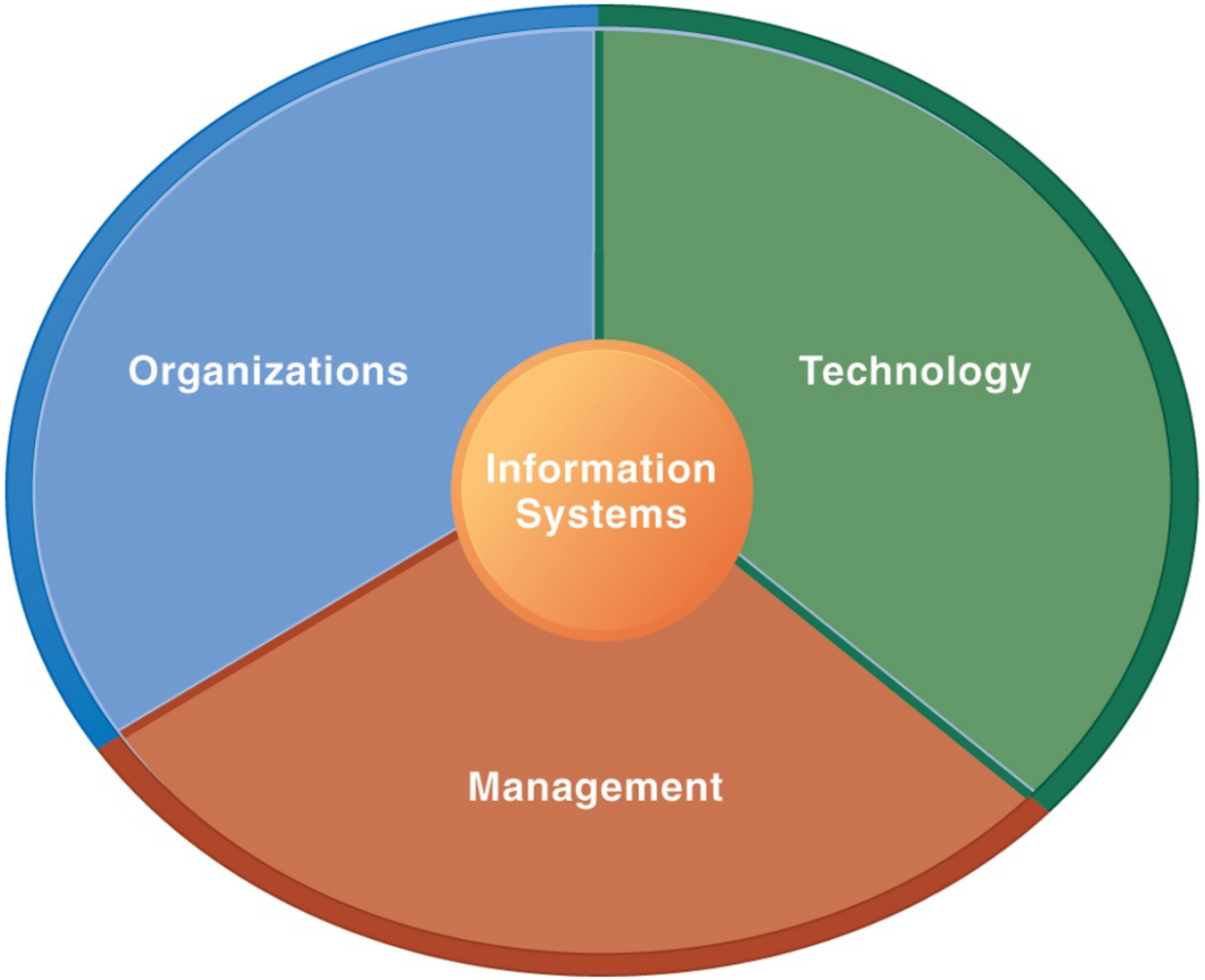


Information Management

Management Information Systems (MIS)

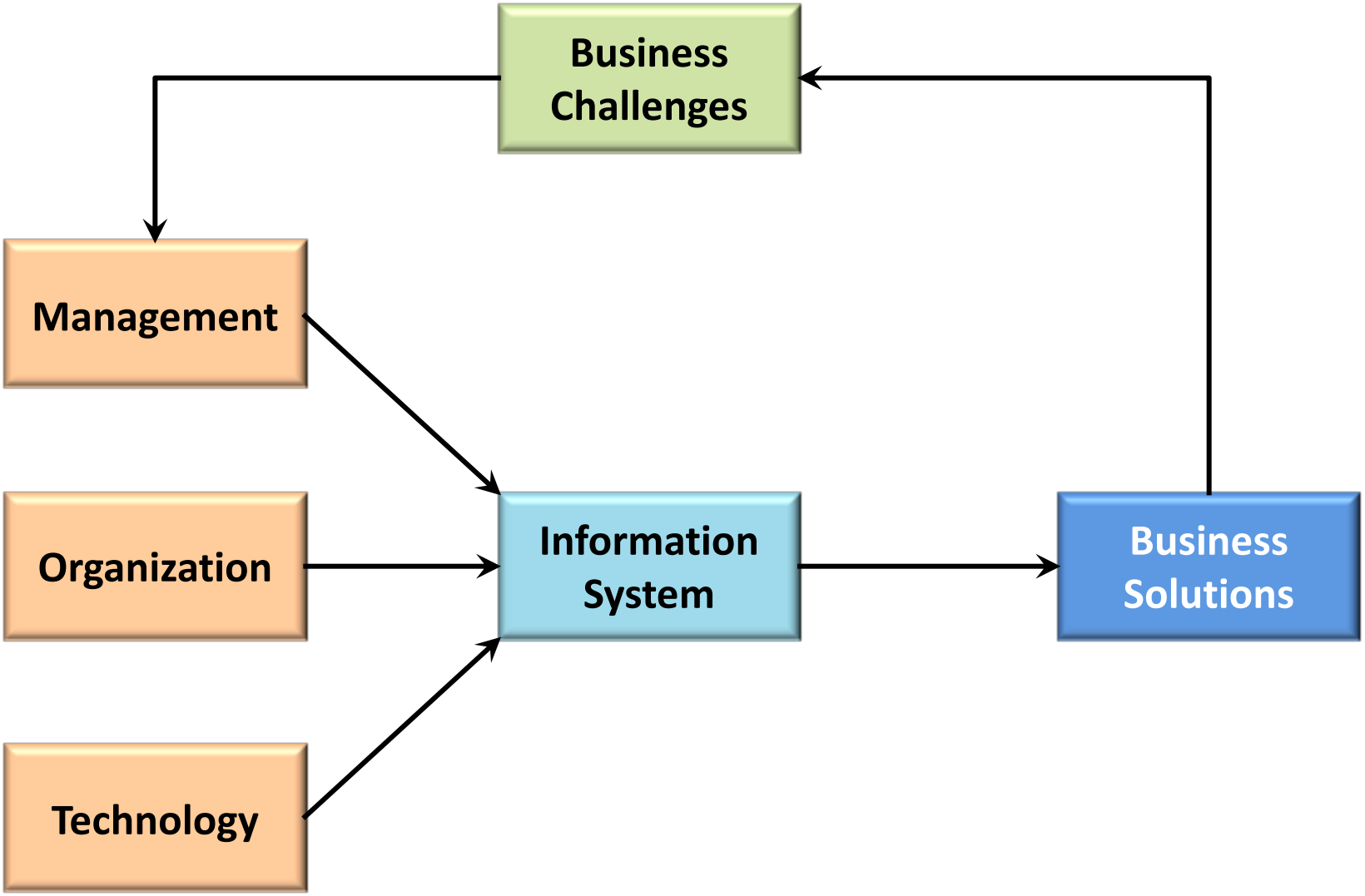
Information Systems

Information Management (MIS) Information Systems



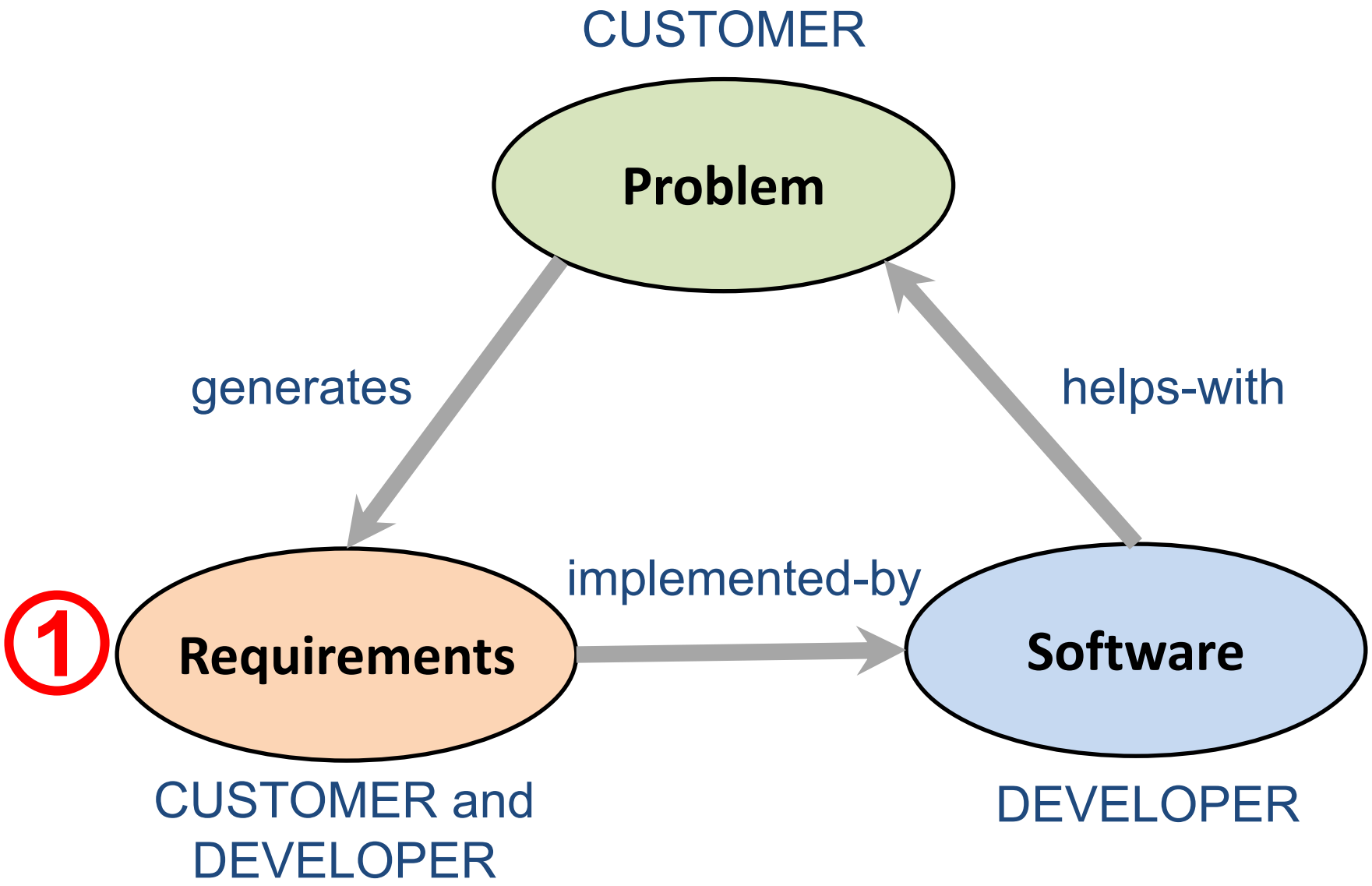
Source: Kenneth C. Laudon & Jane P. Laudon (2014), Management Information Systems: Managing the Digital Firm, Thirteenth Edition, Pearson.

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



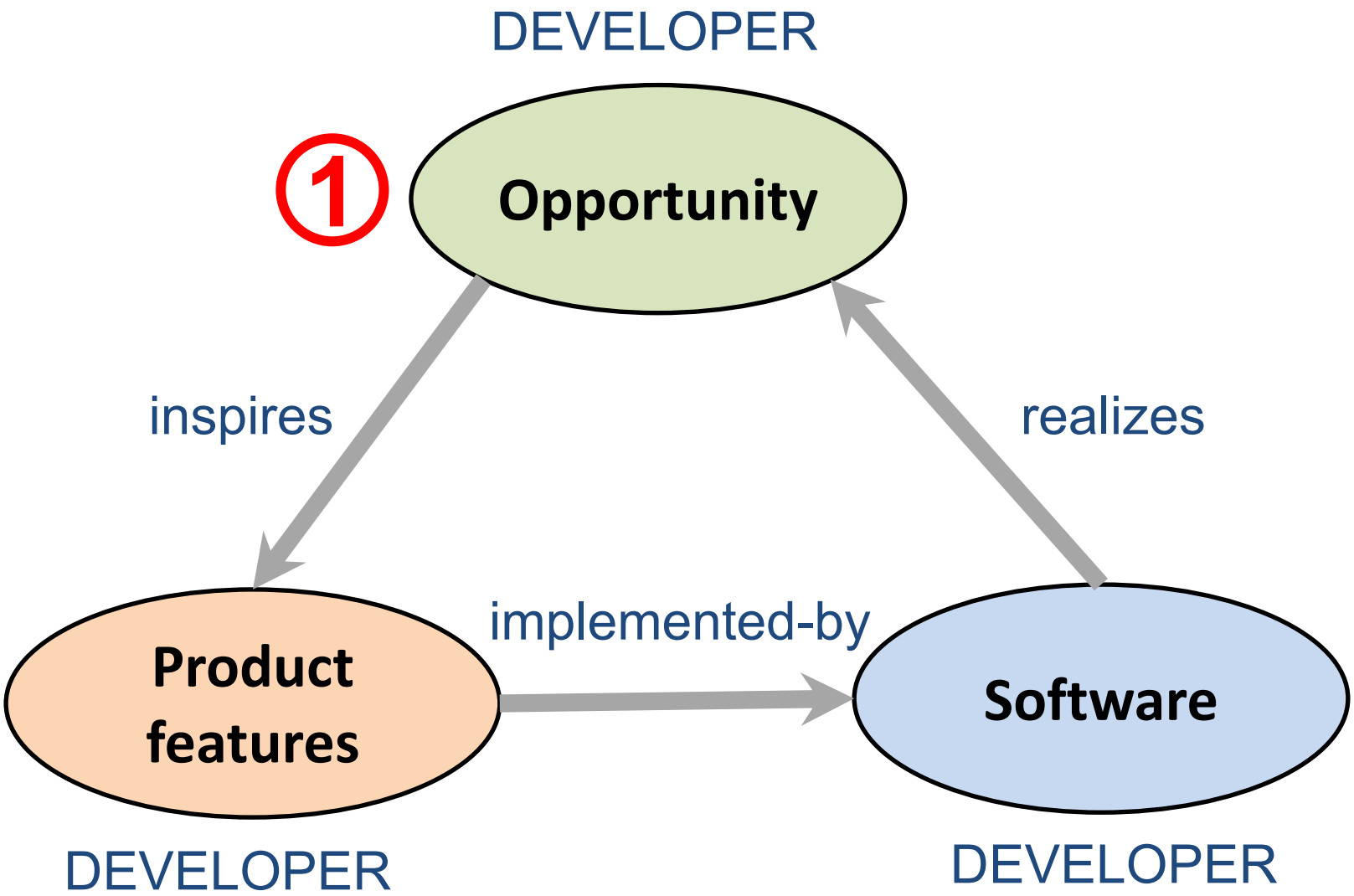
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

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

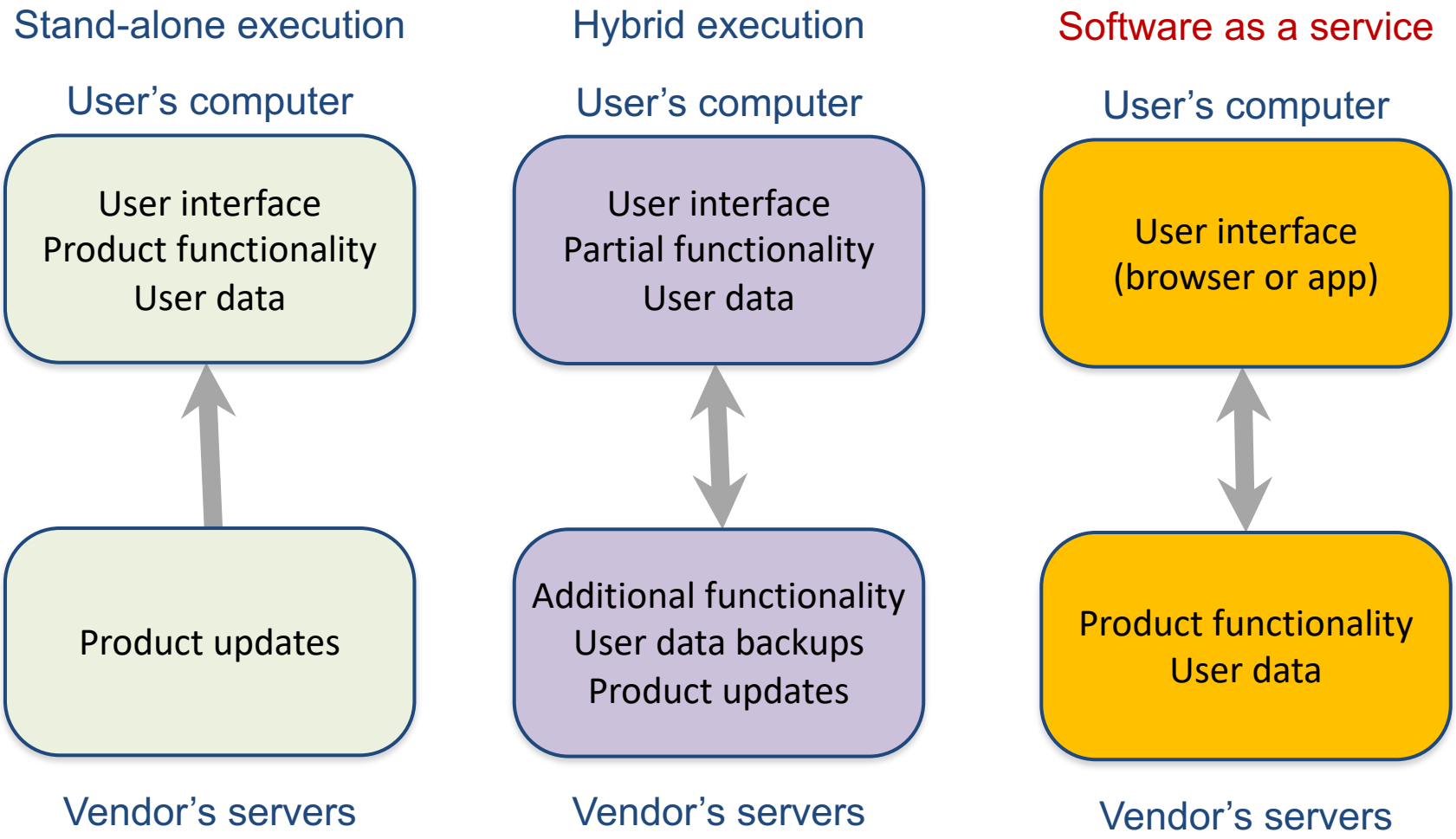


Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Product software engineering

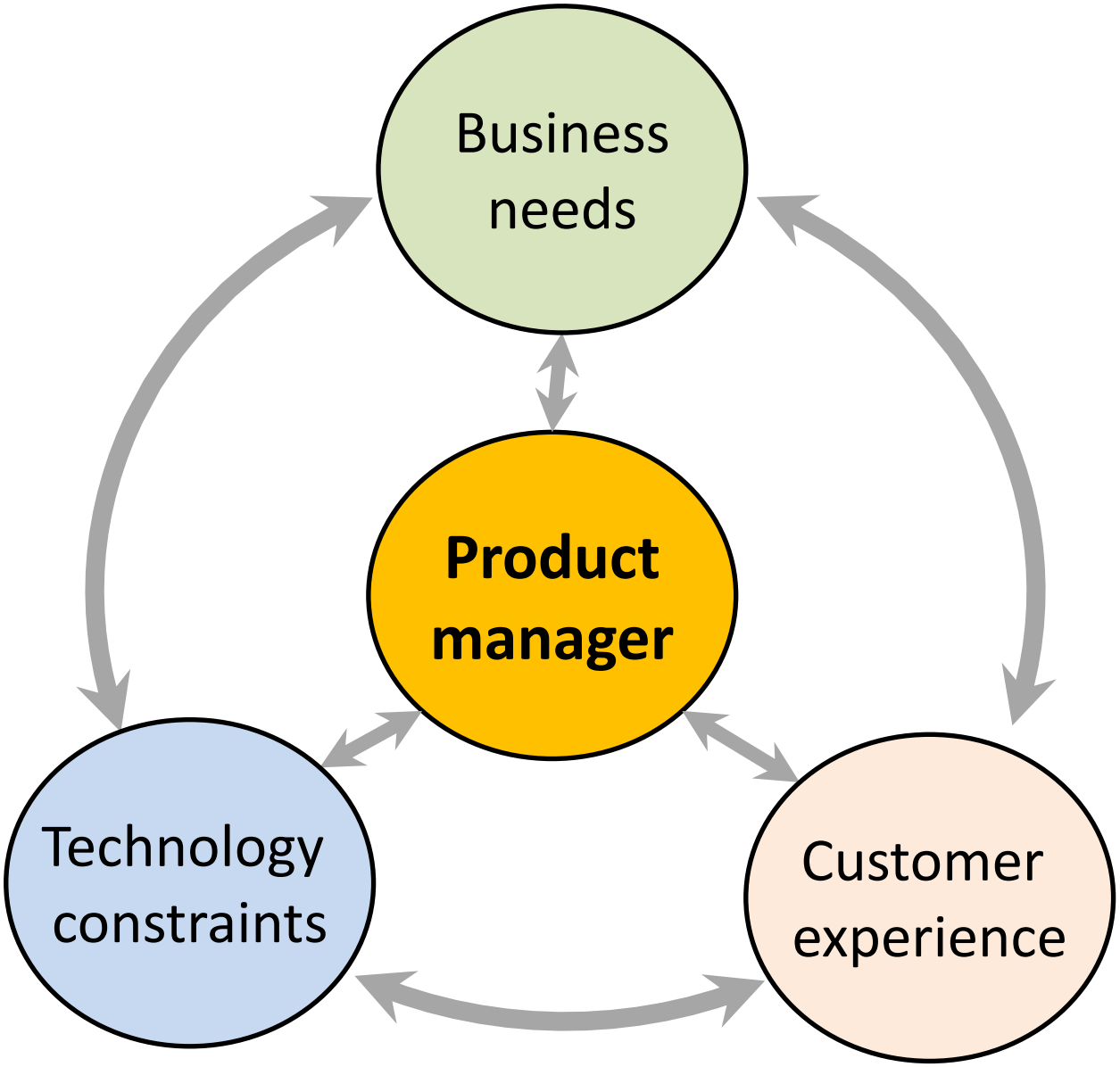
- The starting point for product development is a business opportunity that is identified by individuals or a company. They develop a software product to take advantage of this opportunity and sell this to customers.
- The company who identified the opportunity design and implement a set of software 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.

Software execution models



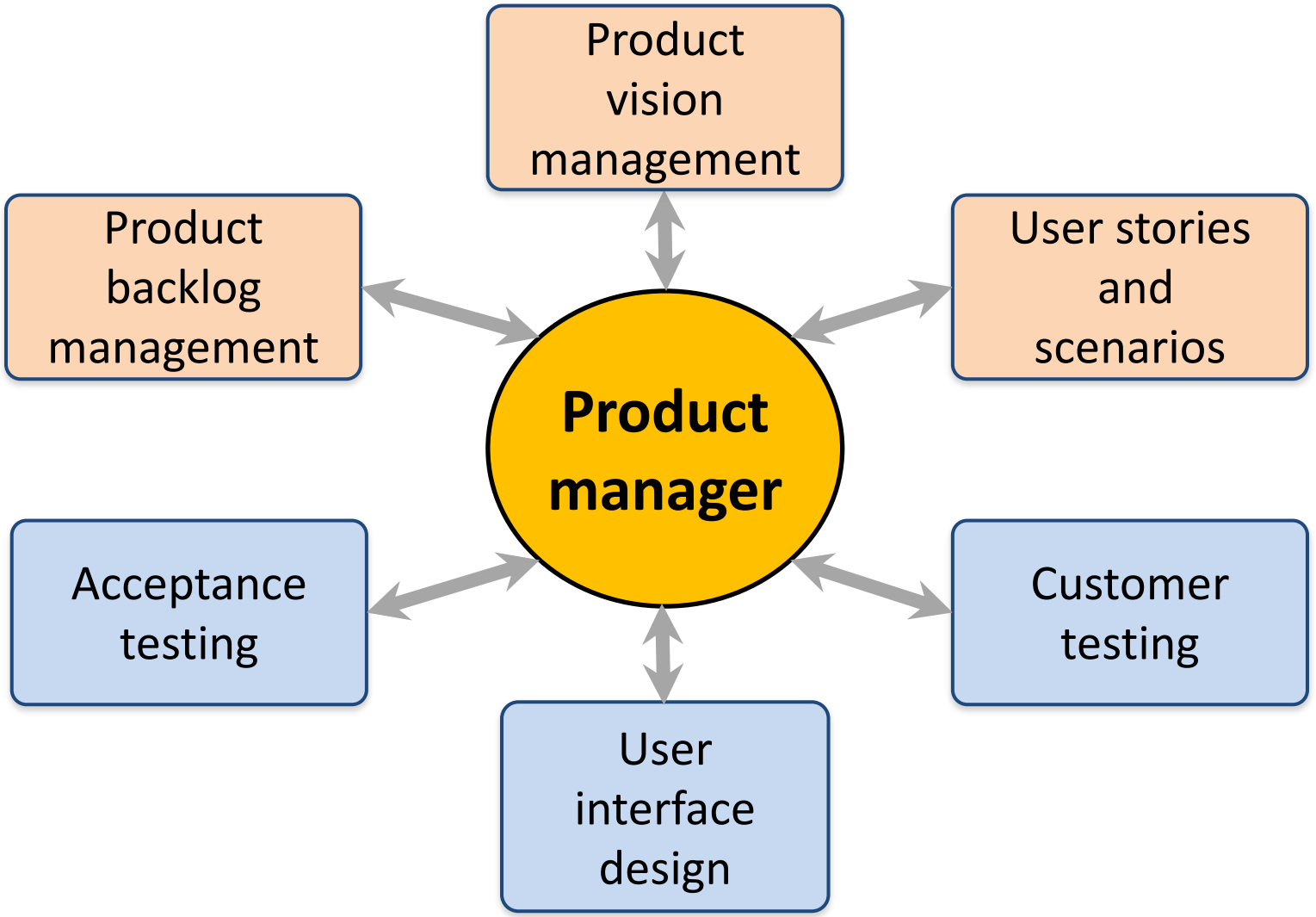
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Product management concerns



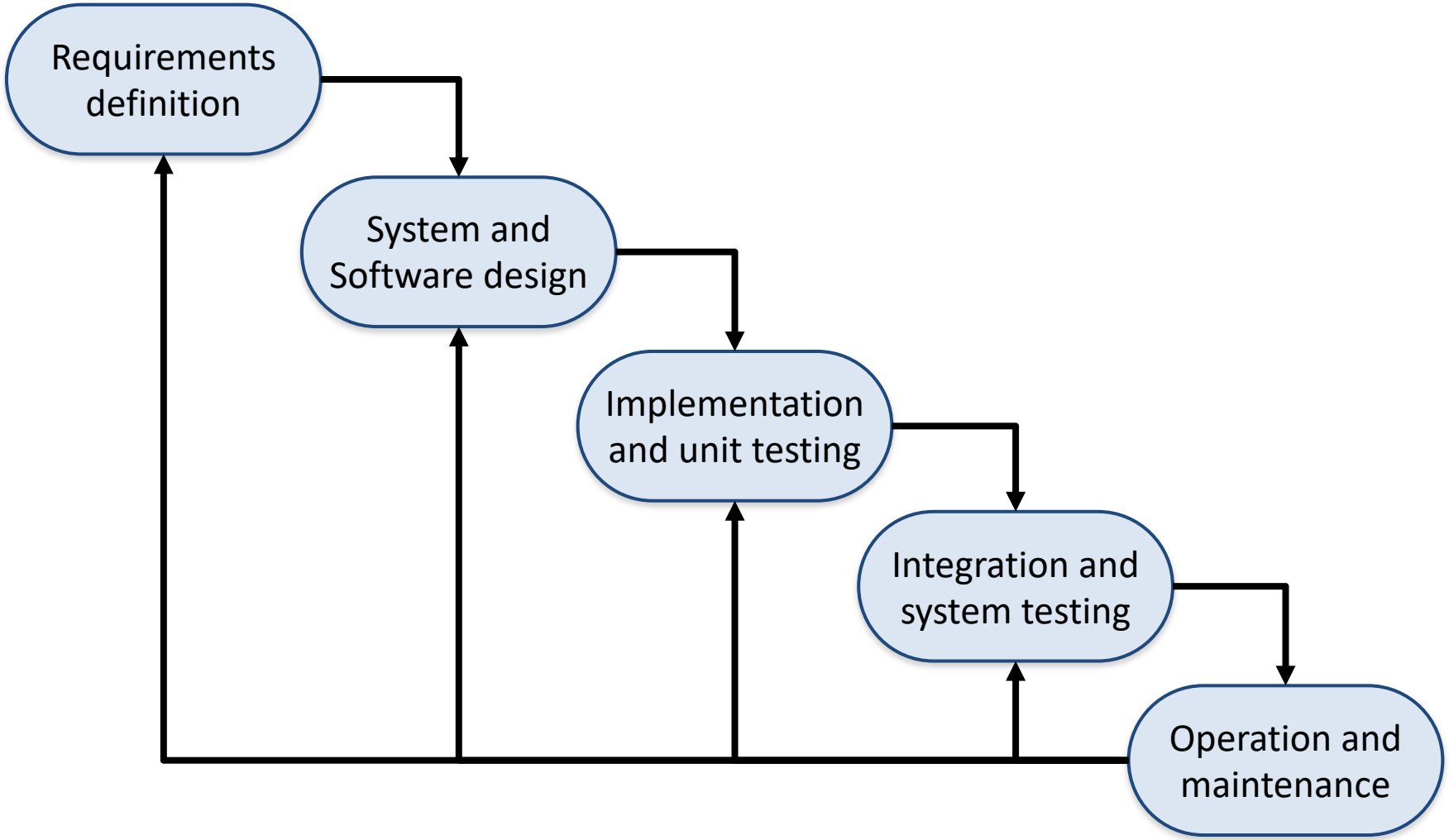
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Technical interactions of product managers



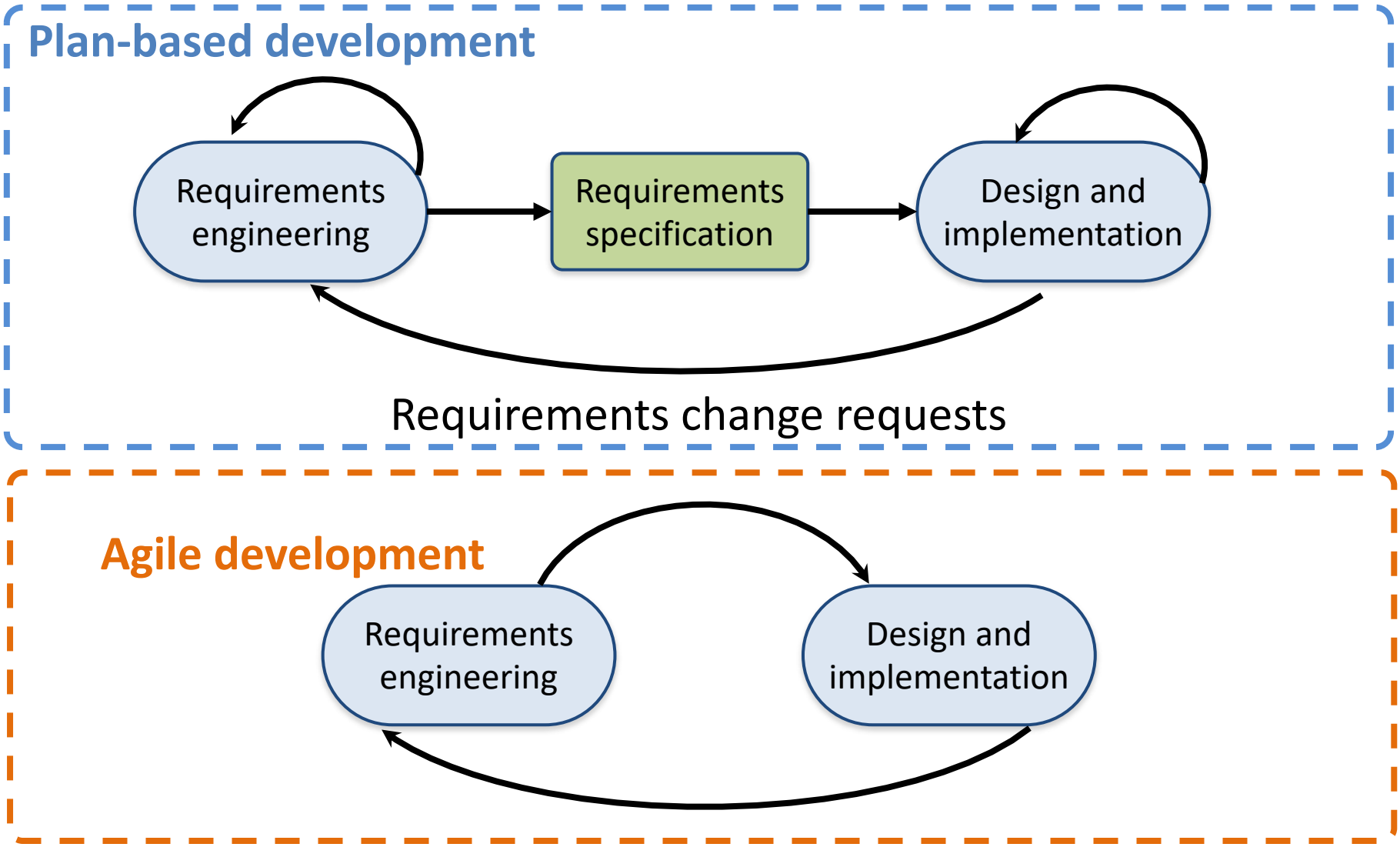
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Software Development Life Cycle (SDLC)
The waterfall model



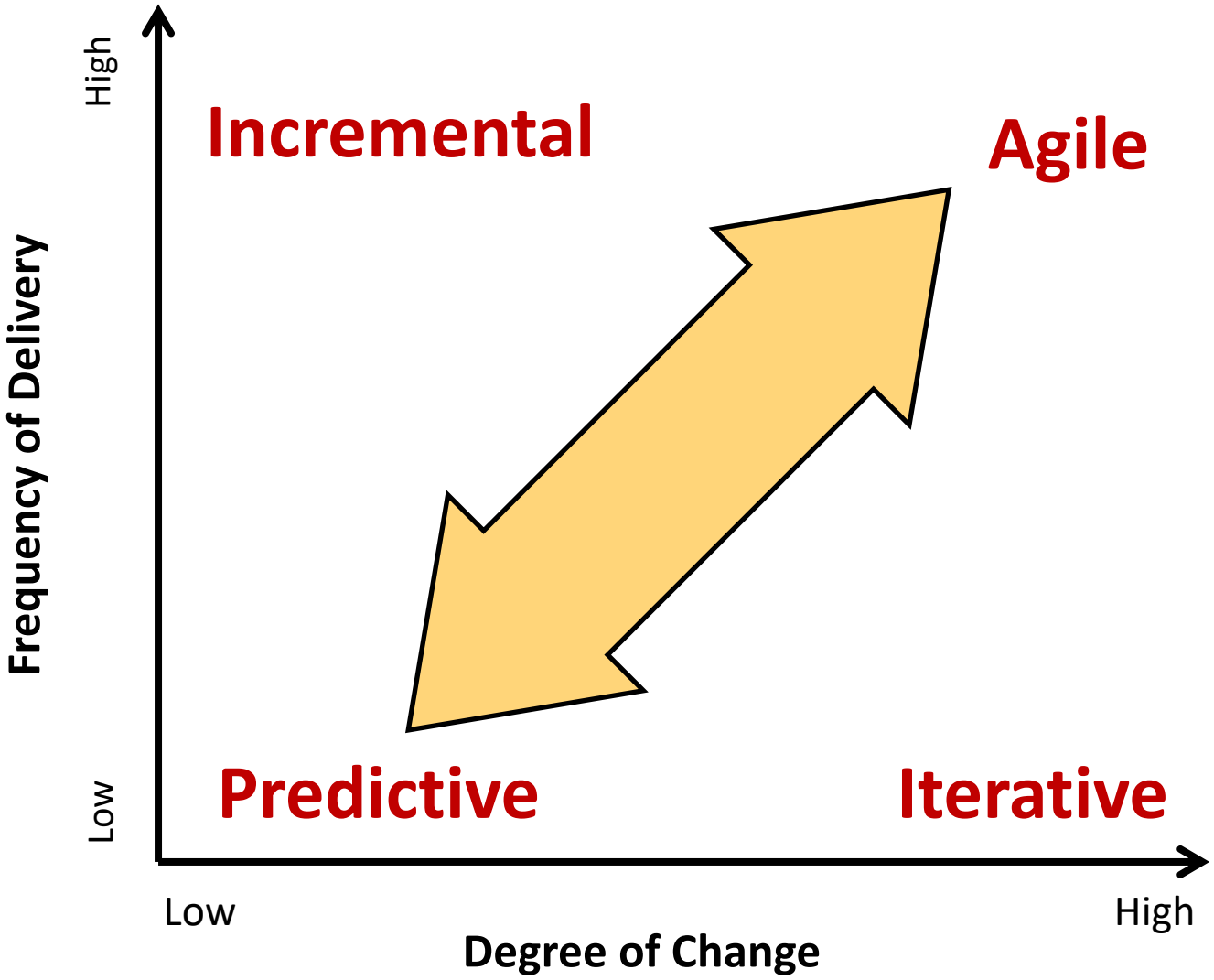
Source: Ian Sommerville (2015), Software Engineering, 10th Edition, Pearson.

Plan-based and Agile development



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

The Continuum of Life Cycles

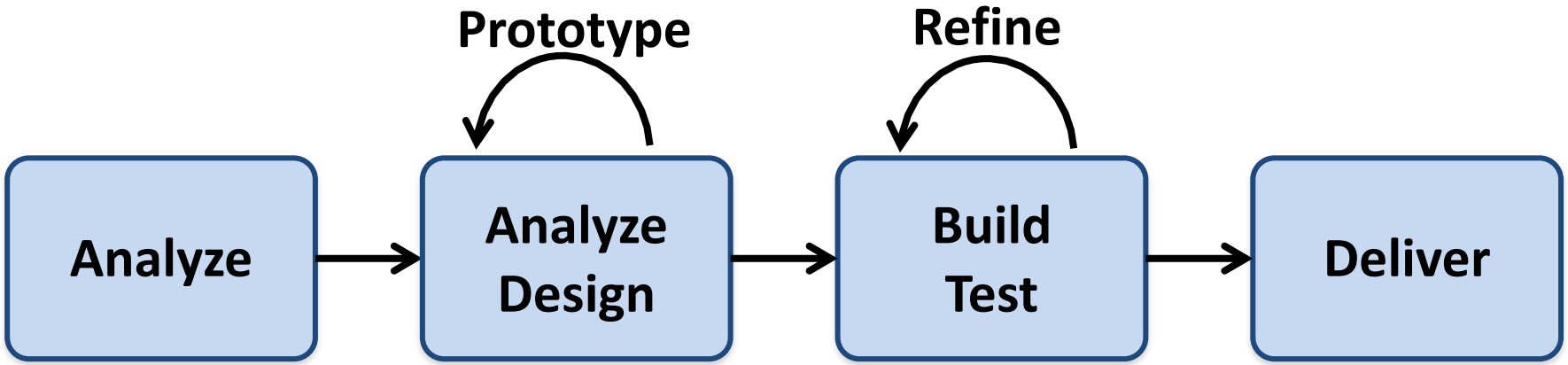


Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

Predictive Life Cycle

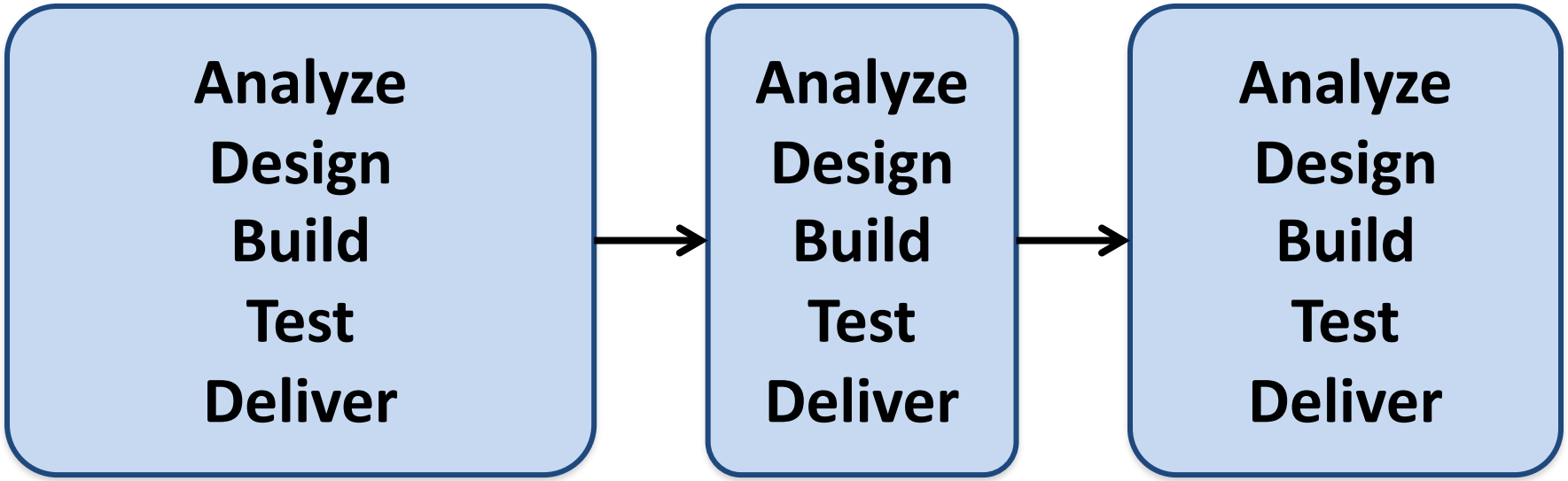
Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

Iterative Life Cycle



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

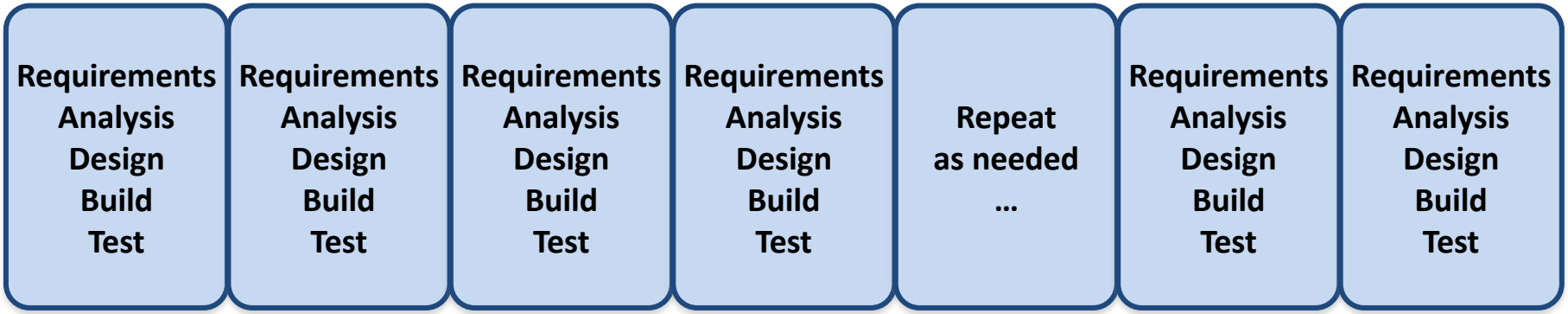
A Life Cycle of Varying-Sized Increments



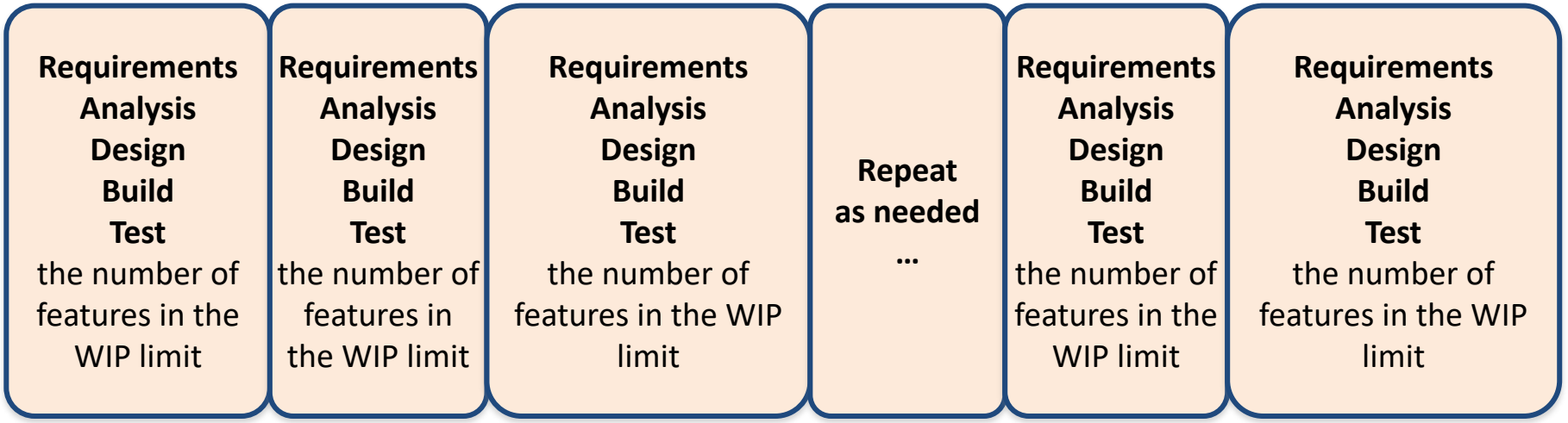
Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

Iteration-Based and Flow-Based Agile Life Cycles

Iteration-Based Agile

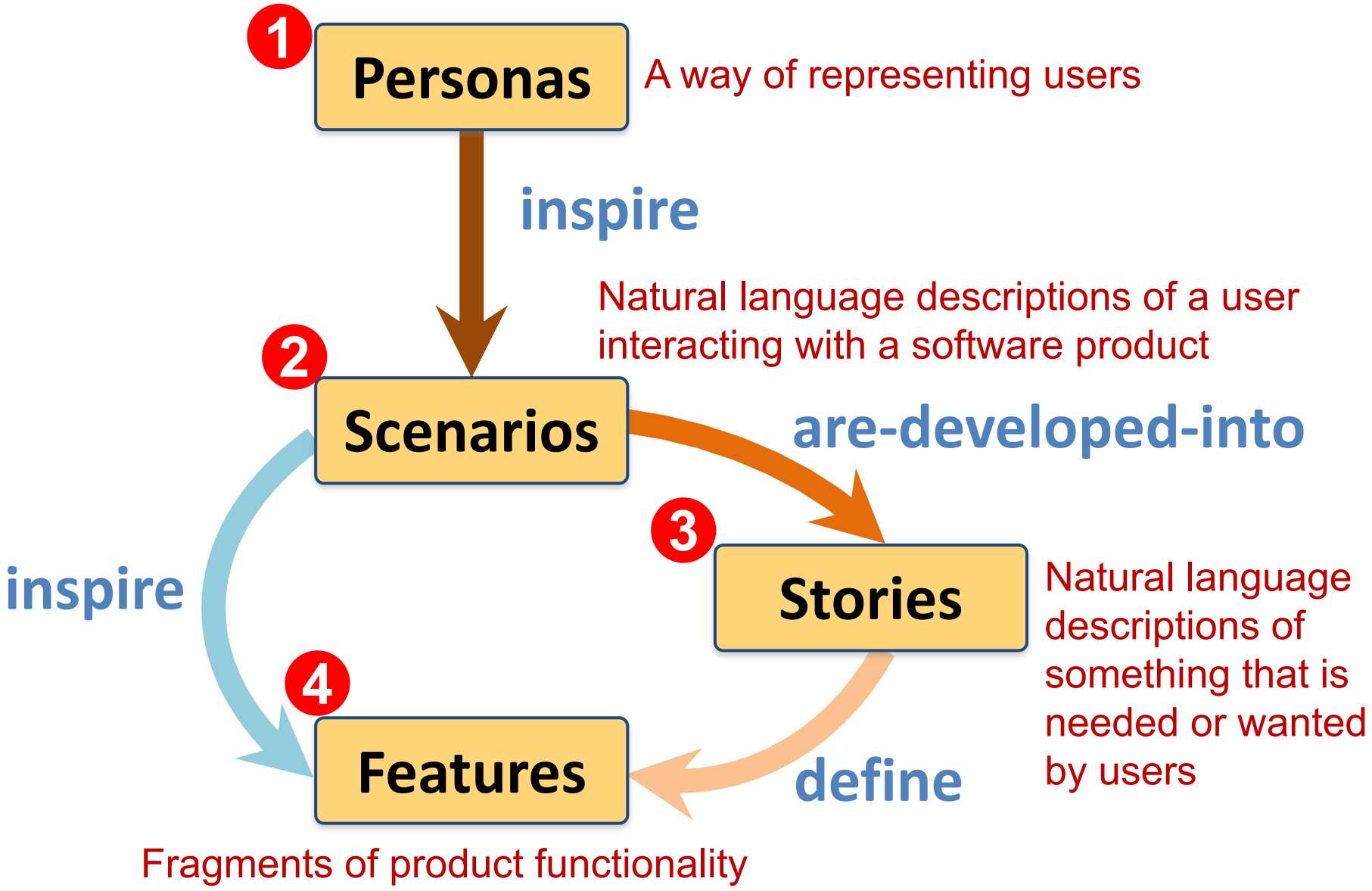


Flow-Based Agile



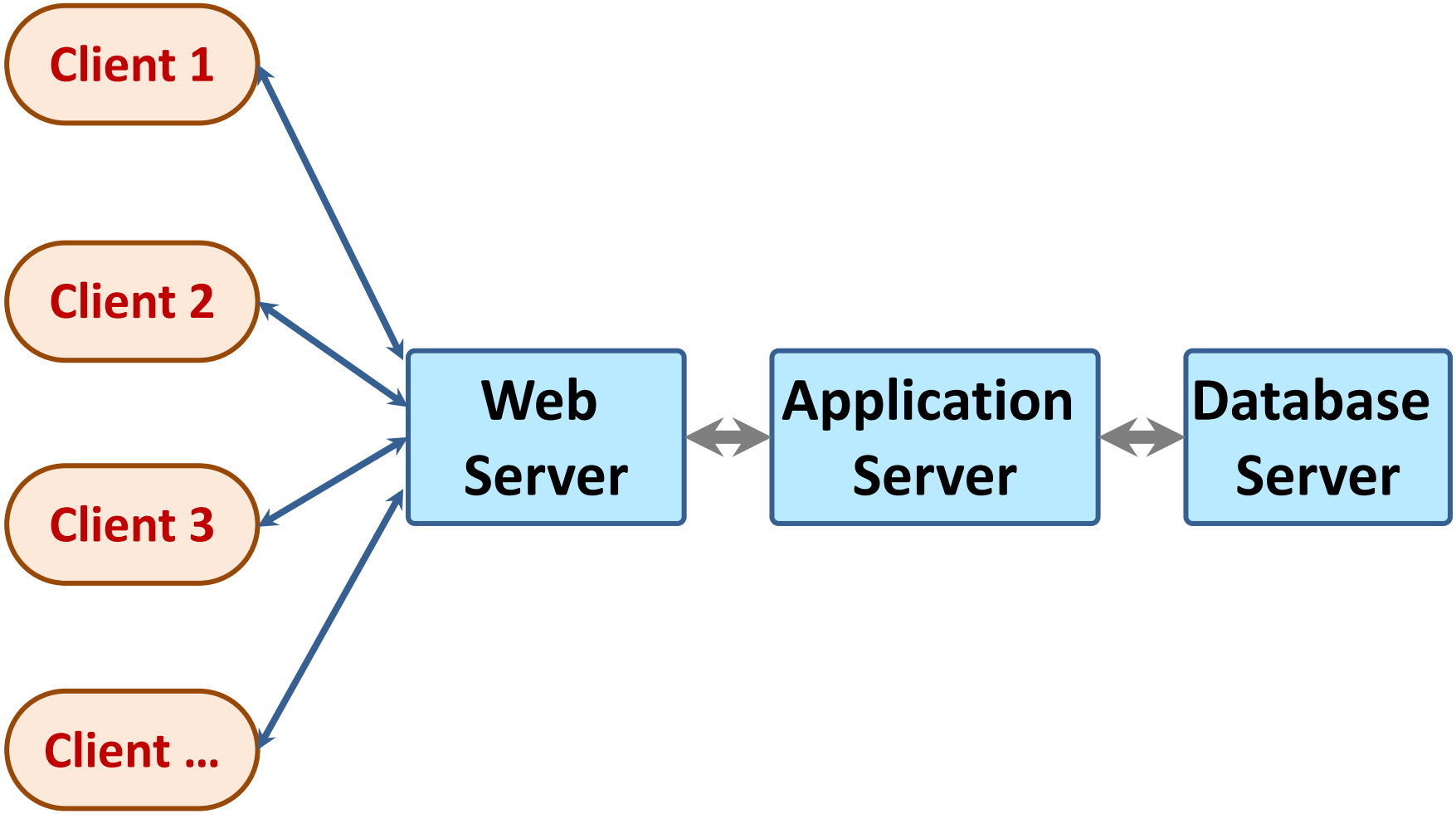
Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

From personas to features



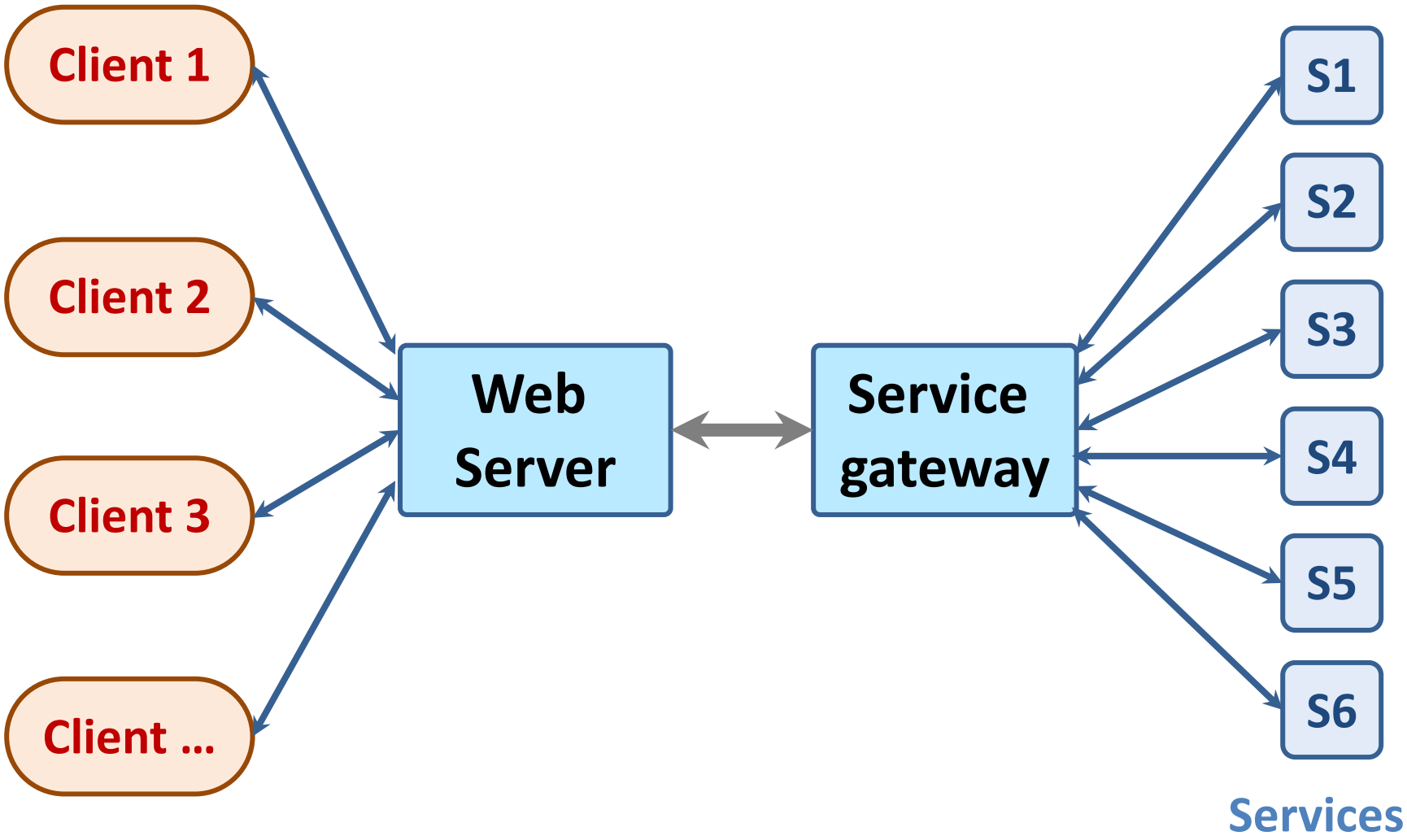
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Multi-tier client-server architecture

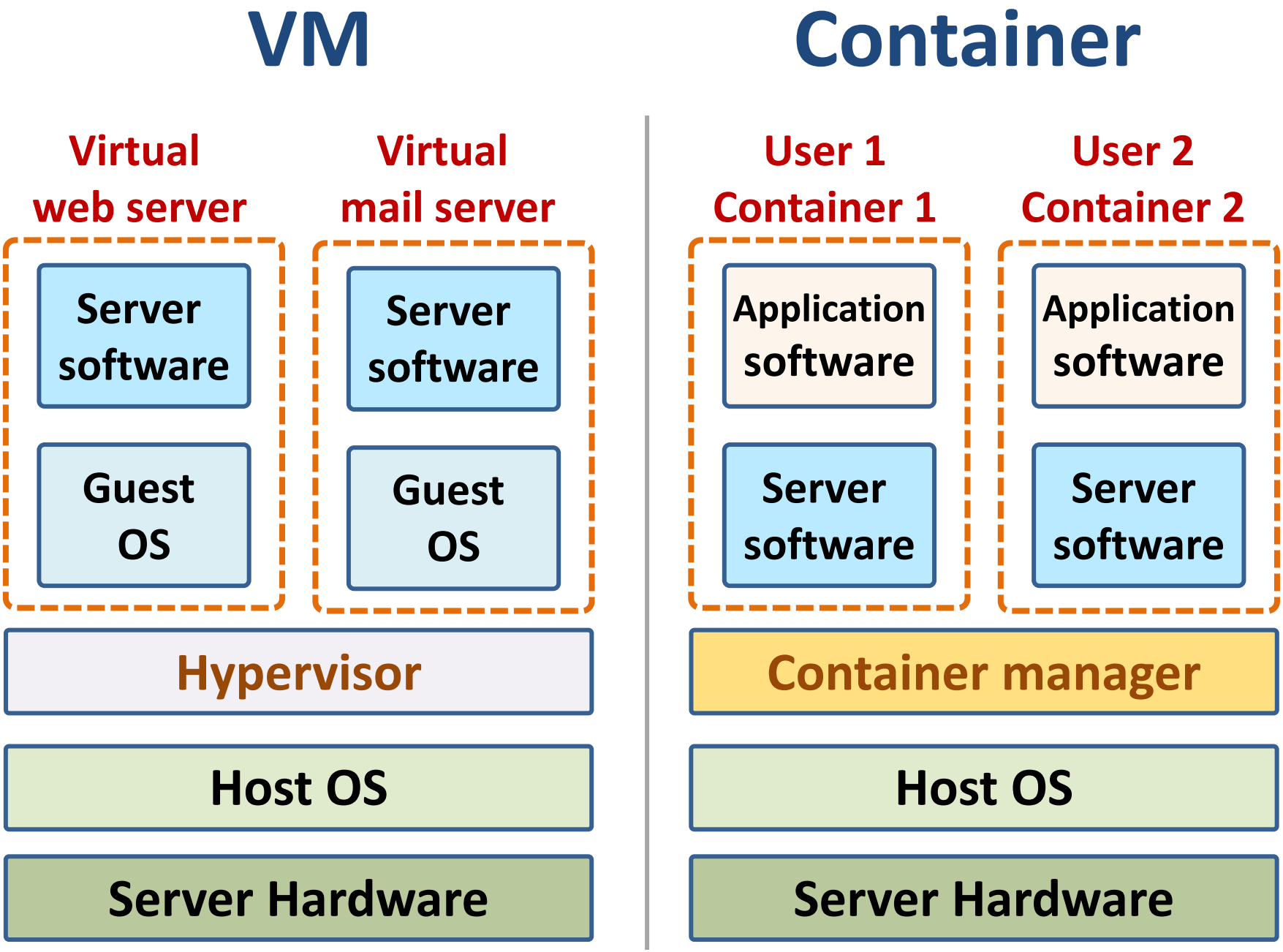


Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Service-oriented Architecture

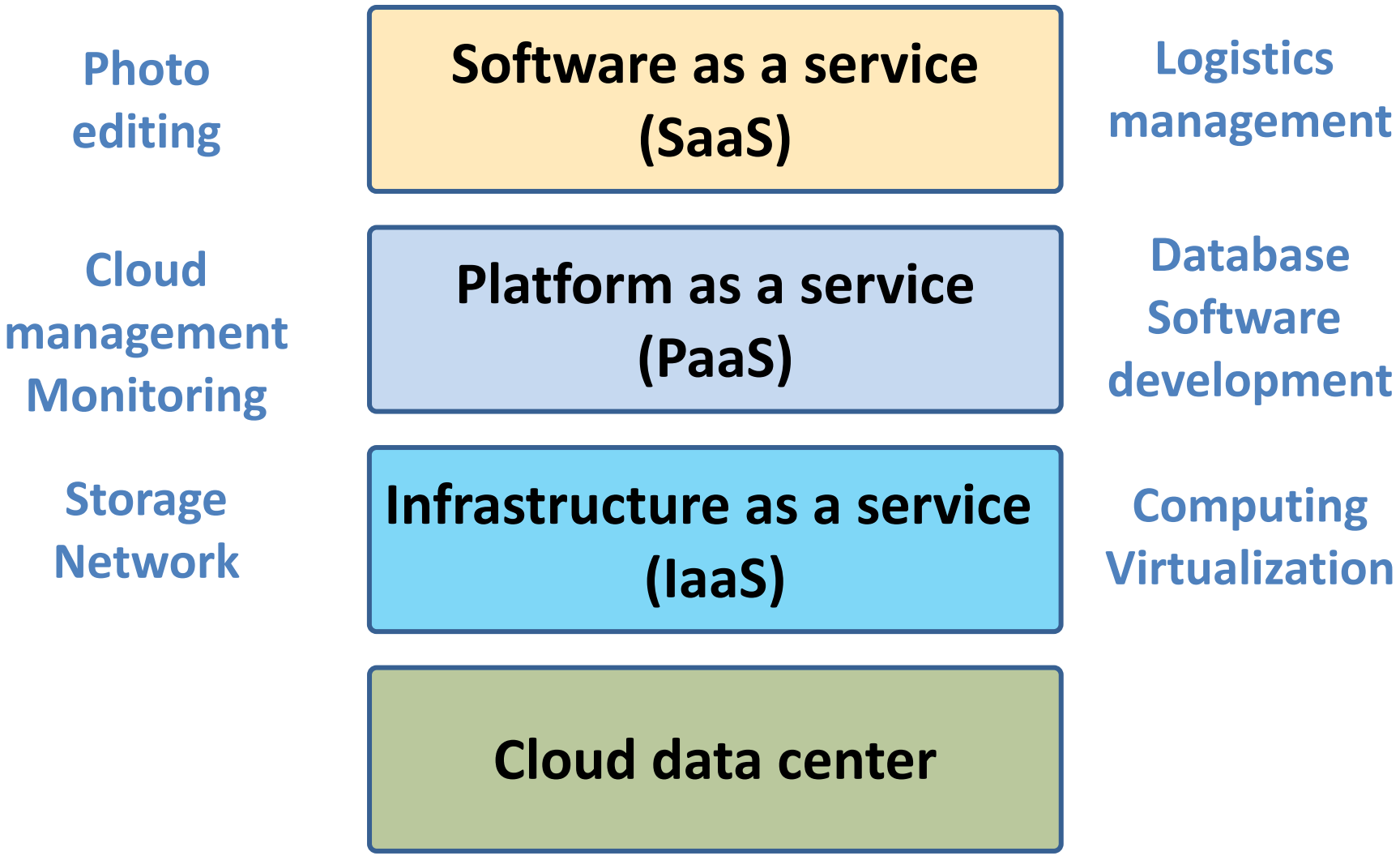


Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.



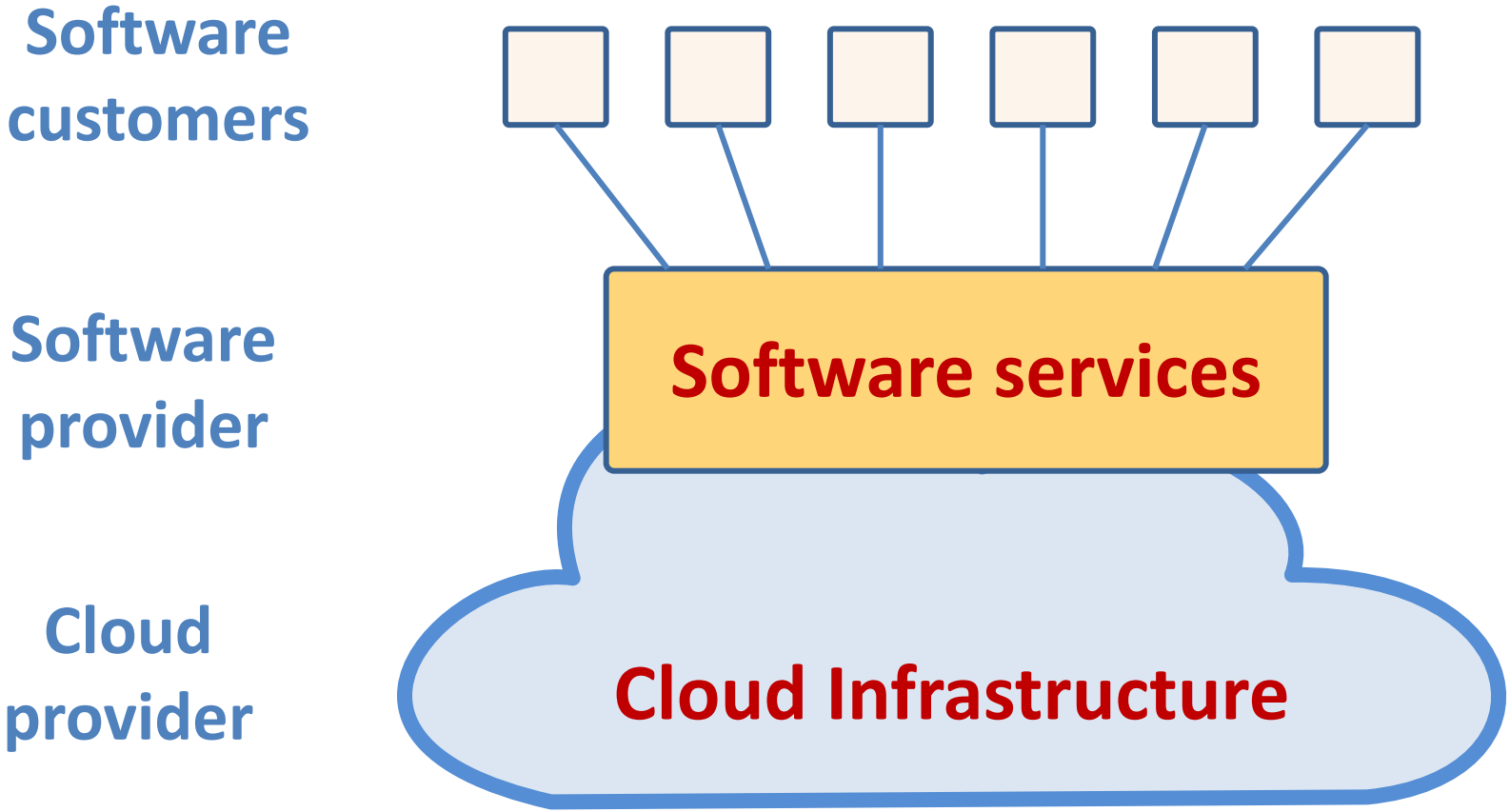
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Everything as a service



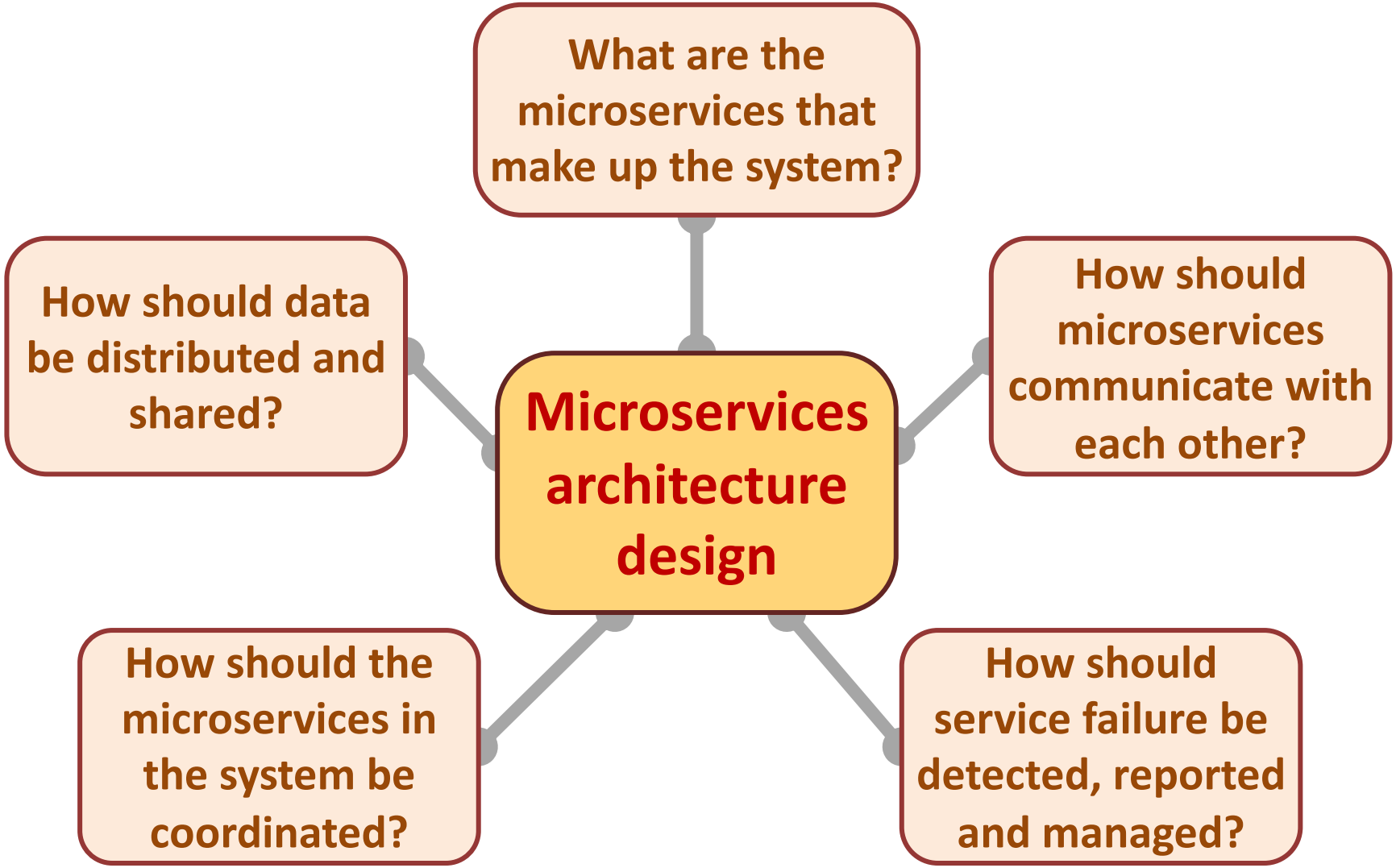
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Software as a service



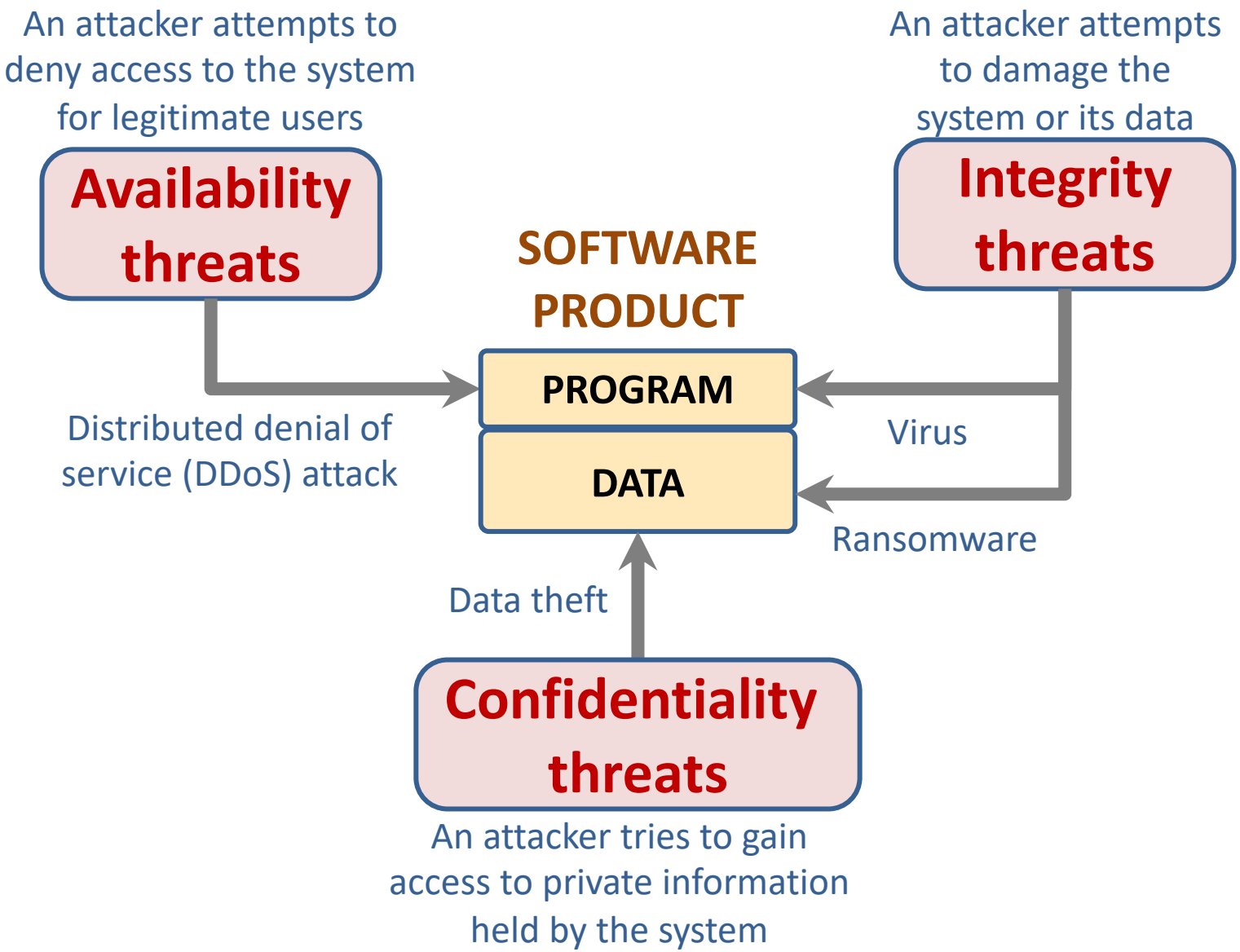
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Microservices architecture – key design questions



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

Types of security threat

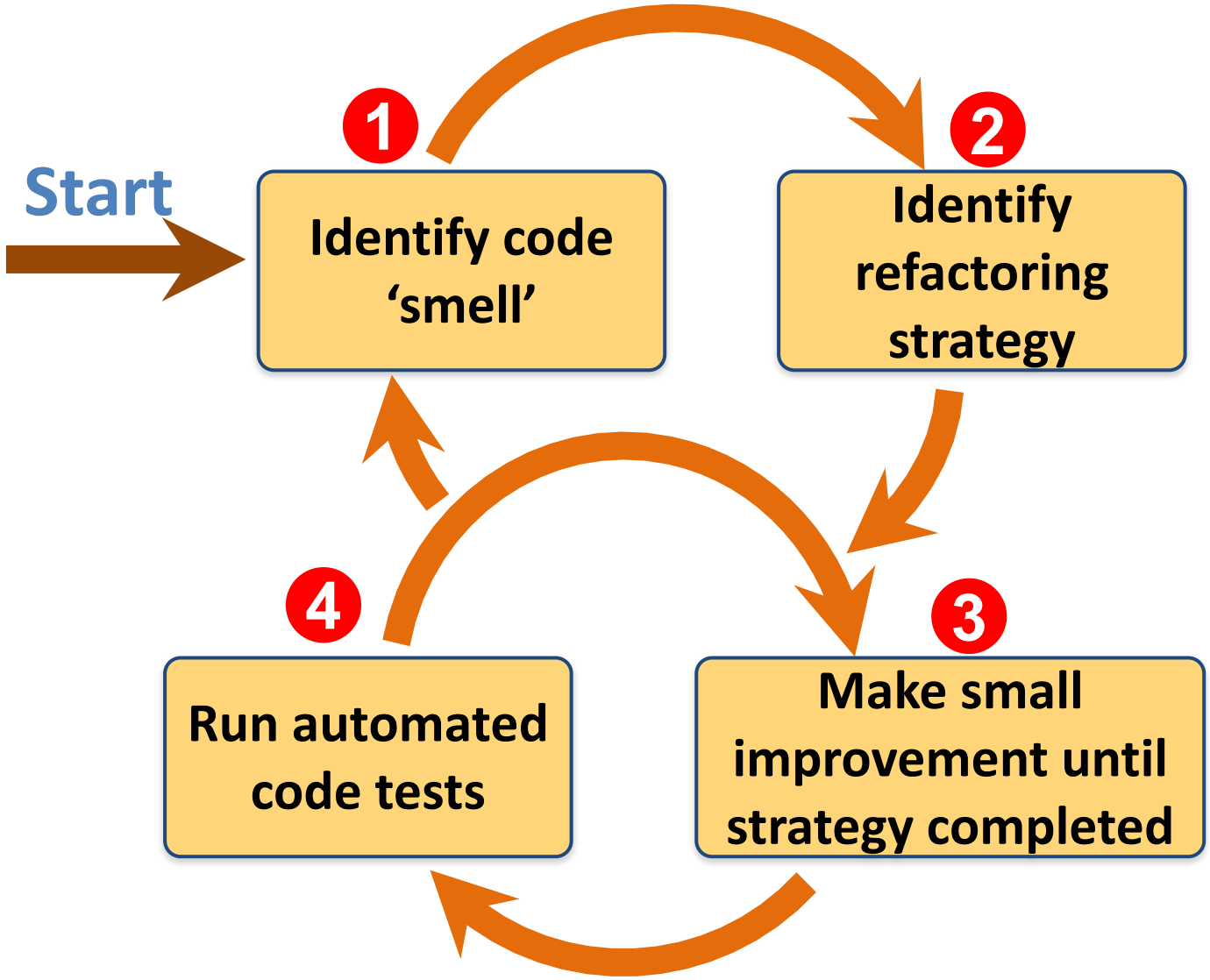


Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Software product quality attributes

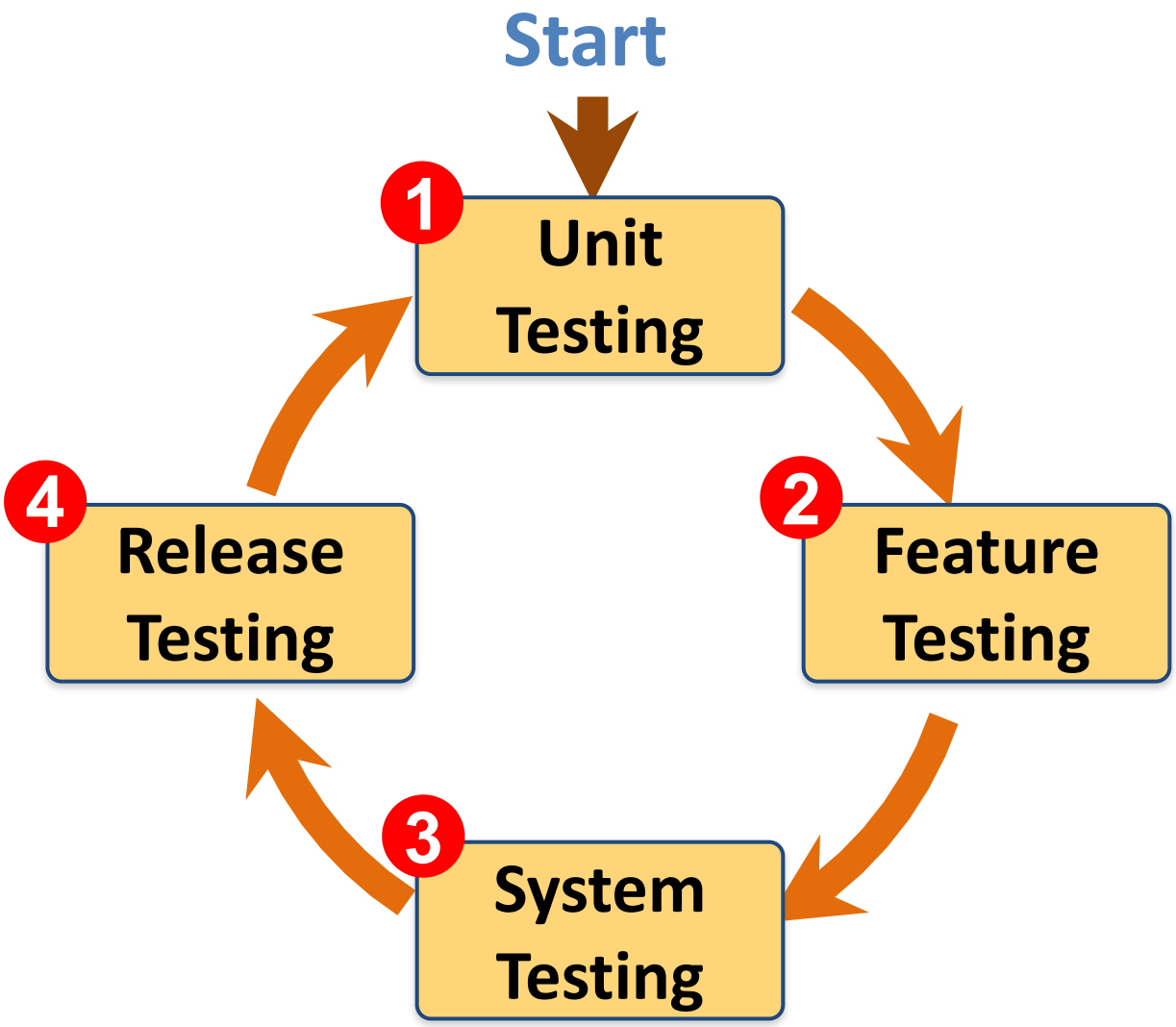
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

A refactoring process



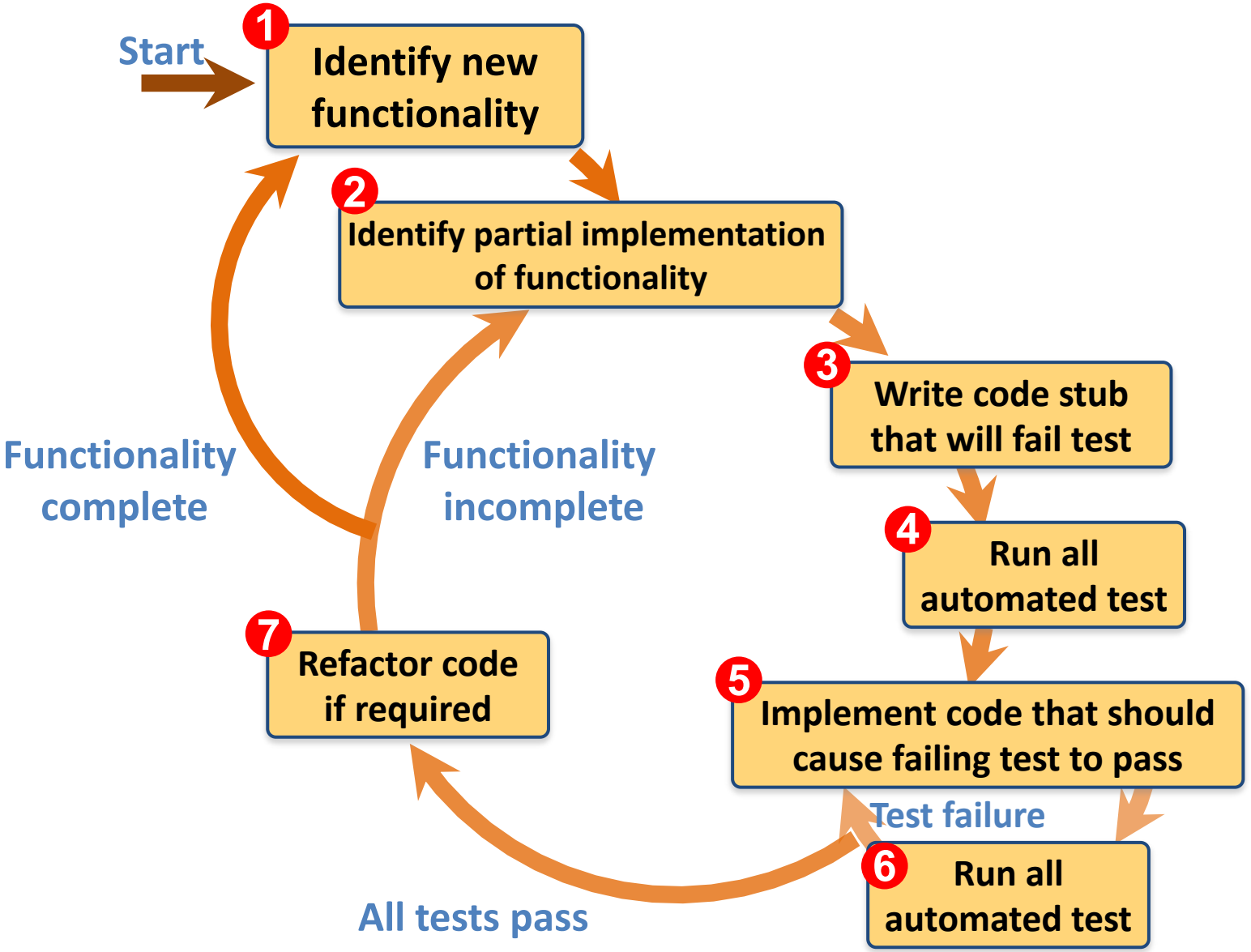
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Functional testing



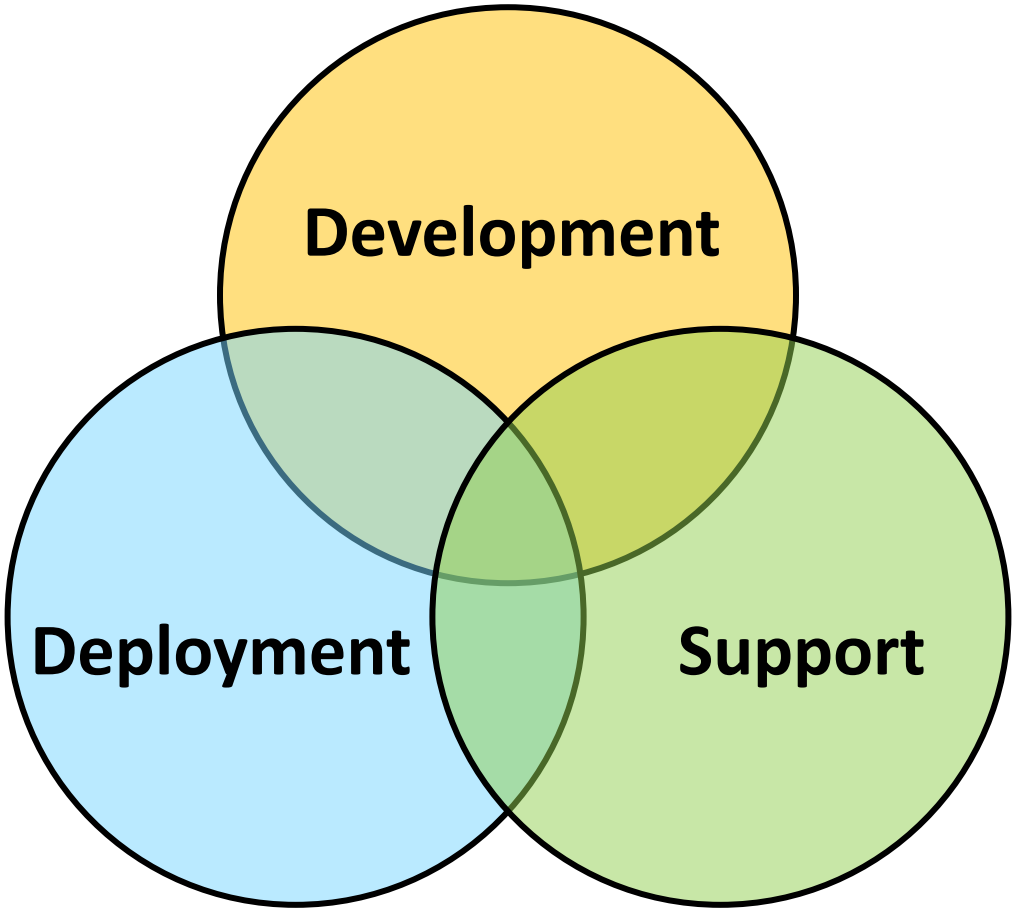
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Test-driven development (TDD)



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

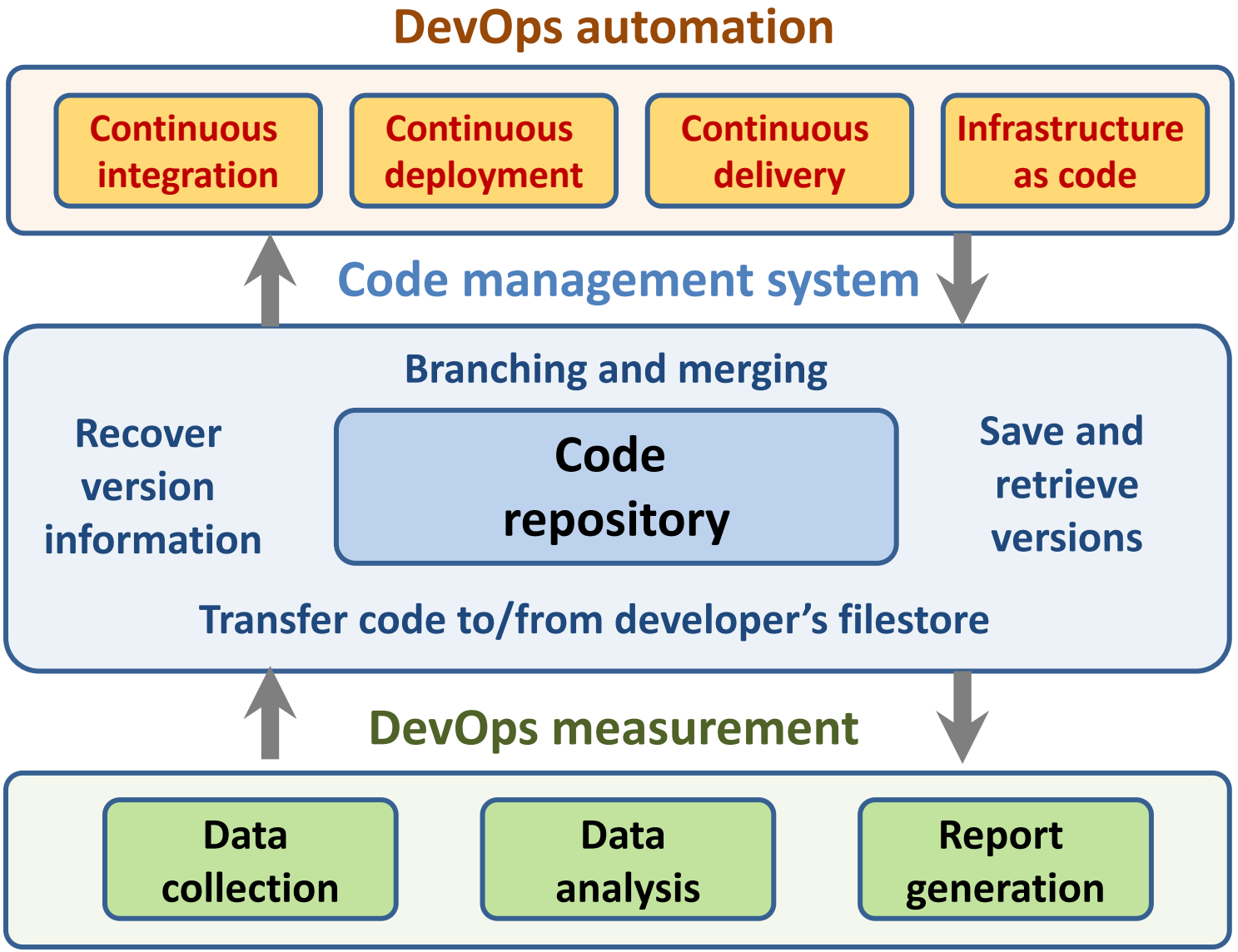
DevOps



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**."

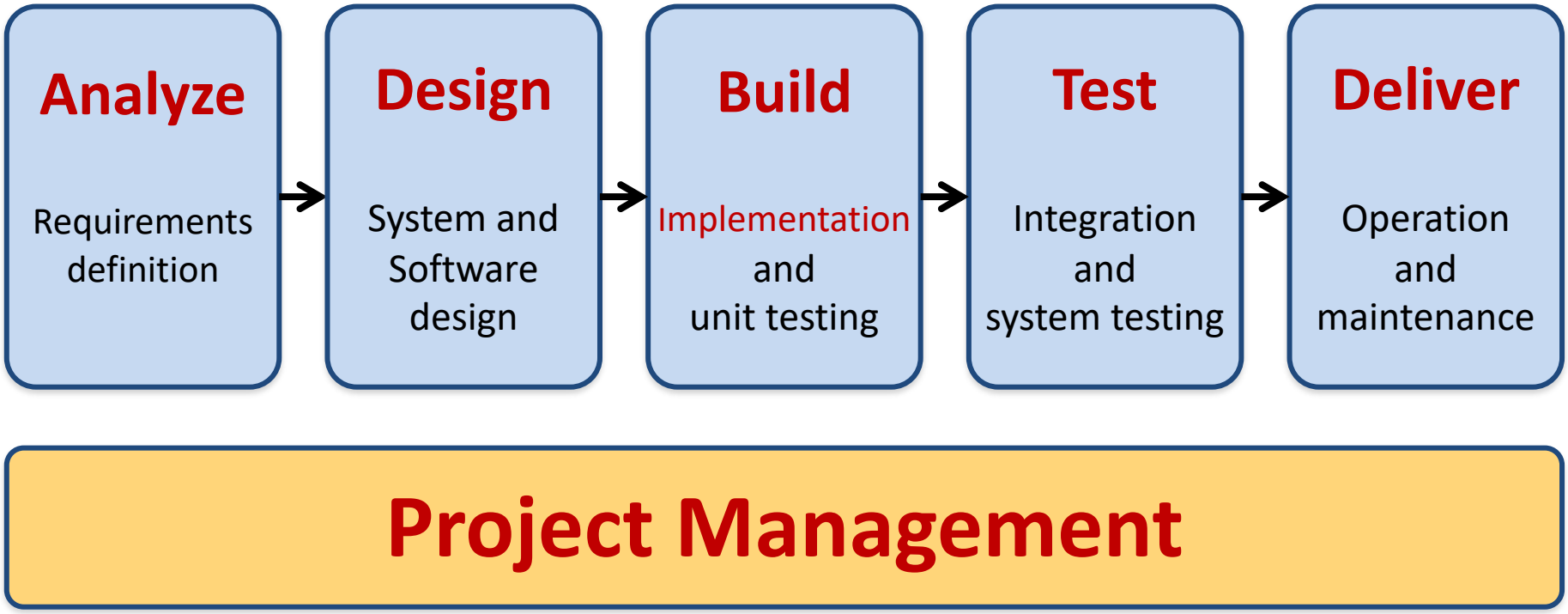
Source: Philip Kotler and Kevin Lane Keller (2016), Marketing Management, 15th edition, Pearson.

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

aws educate | Cloud Ambassador
2020 Cohort

aws academy Accredited Educator

aws certified Solutions Architect Associate

aws certified Cloud Practitioner

軟體工程
(Software Engineering)
Contact Information

戴敏育 博士 (Min-Yuh Day, Ph.D.)

副教授 (Associate Professor)

國立臺北大學 資訊管理研究所

Institute of Information Management, National Taipei University

電話： 02-86741111 ext. 66873

研究室： 商8F12

地址： 23741 新北市三峽區大學路 151 號

Email：myday@gm.ntpu.edu.tw

網址：http://web.ntpu.edu.tw/~myday/