

# Welcome to Introduction of Software Engineering

SOF107-2025-04

- Dr. Hejab M. Al-Fawareh
- [alfawarehhejab.Khaled@xmu.edu.my](mailto:alfawarehhejab.Khaled@xmu.edu.my)
  - 0199184370



XIAMEN UNIVERSITY MALAYSIA  
廈門大學 馬來西亞分校

**SOF107**  
**INTRODUCTION OF SOFTWARE**  
**ENGINEERING**

**DR. AL FAWAREH HEJAB**  
School of Computing and Data Science

2025/04

**Dr. Hejab Al Fawareh (PhD)**  
**Assistant Professor**

Email: [alfawarehhejab.khaled@xmu.edu.my](mailto:alfawarehhejab.khaled@xmu.edu.my)

Office: **A1-354**

Monday: 4:00 PM – 7:00 PM-(DMT&CYS)

Friday : 9:00 AM- 12:00 PM (SWE).

\*Only via **Microsoft Teams**



XIAMEN UNIVERSITY MALAYSIA

廈門大學 馬來西亞分校

# SOF107 INTRODUCTION OF SOFTWARE ENGINEERING

## Course Synopsis:

- This course is designed to provide students with fundamental concept and principles of software engineering and the process of software project.
- The students will be exposed to different techniques and methodologies in software engineering.
- The students will learn the process, practice and implementation software requirement, software design, software testing and software management.



# Introduction of Software Engineering

## SOF107

### Course Learning Outcomes (CLO)

At the end of the course, students will be able to:

1. Describe the fundamentals of software engineering and the process of software project. (C2, PLO1)
2. Apply different techniques and methodology in software engineering. (C3, PLO2)
3. Demonstrate the technique and methodology for software requirement, software design and software testing. (A3, PLO5)

# SOF107INTRODUCTION OFSOFTWARE ENGINEERING

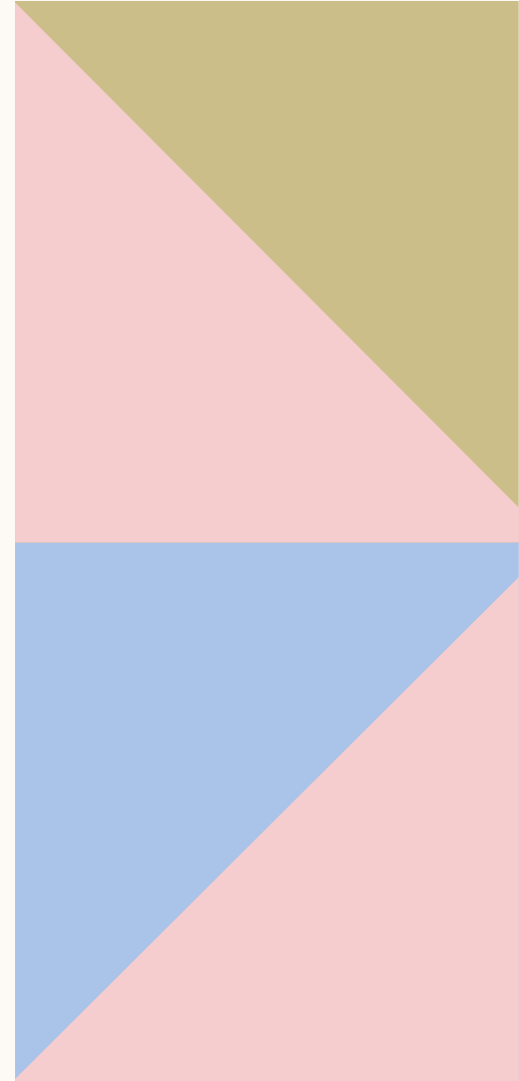
## Course Content Outline:

- ❖ Introduction to Software Engineering.
- ❖ The Software Process.
- ❖ Software Requirements.
- ❖ Software Design.
- ❖ Software Test.
- ❖ Software Management.



XIAMEN UNIVERSITY MALAYSIA

廈門大學 馬來西亞分校





# University mission and vision

- **Vision**

- Xiamen University Malaysia aspires to become a university with a distinct global outlook, featuring first-class teaching and research, and embracing cultural diversity.

- **Mission**

- To nurture young talents with dignity and wisdom, turning them into fine citizens of the region who will contribute to the prosperity of the people and social progress of Malaysia, China and Southeast Asia.



# Course Information

- Course code :SOF107
- Course Lecturer : Dr. Al Fawareh Hejab  
[alfawarehhejab.khaled@xmu.edu.my](mailto:alfawarehhejab.khaled@xmu.edu.my)  
0199184370
- Room No. : A1-354
- Course Assessment:
  - Assignment (1600 words) / 20%,
  - Report with Presentation (1600 words) / 30%
  - Final exam / 50%



# COURSE EXPECTATIONS

- ❖ **Alert to course announcement and assessment instructions in Moodle and Email**
- ❖ **Punctual - No late submission of assessment tasks. Late submission will have penalty by deduct the marks!**
- ❖ **Attendance rate must reach at least 80% to be able to sit for the final exam according to University Policy.**



# Introduction of Software Engineering

## SOF107

### Required References

Pressman, Roger S.; Maxim, Bruce; Software Engineering: A Practitioner's Approach 9th edition. McGraw Hill Higher Education Press, 2020. (ISBN13: 9781259872976)

<b>1. Additional Reference(s)</b>	<ul style="list-style-type: none"><li>• Sommerville, Ian. Software Engineering. Pearson. 10th edition. 2018. (ISBN13: 978-9332582699).</li><li>• Sommerville, Ian. Engineering Software Products: An Introduction to Modern Software Engineering. Pearson. 1st Edition. 2020. (ISBN13: 978-0135210642).</li></ul>
-----------------------------------	---

- Triple pass requirement applies to all ICT courses
- Minimum of 40% in total continuous assessment;  
AND
- Minimum of 40% in total final assessment;  
AND
- The passing mark for the course (sum of continuous and
- Final assessment component) is 55% (Grade C).

# Consultation hour

Lecturer	Day	Time	Duration
Dr. Hejab Al Fawareh	Monday	3:00 pm -4:00 pm	1 –hour
Dr. Hejab Al Fawareh	Tuesday	4:00 pm- 6:00 pm	2 -hour
Dr. Hejab Al Fawareh	Wednesday	3:00 pm -4:00 pm	1 -hours

## *Steps to Prepare a Program According to Software Engineering*

---

- **Subtitle:** A Guide to Software Development Life Cycle (SDLC)

**Step 1 - Requirement Analysis**

**Step 2 – Planning**

**Step 3 - System Design**

**Step 4 - Implementation (Coding)**

**Step 5 - Testing**

**Step 6 - Deployment**

**Step 7 - Maintenance & Updates**

# Best Practices

---

- Choose development methodology (Agile, Waterfall, etc.).
- Implement Continuous Integration/Continuous Deployment (CI/CD).
- Maintain proper documentation.

# Introduction of Software Engineering

## SOF107

### CHAPTER 1: INTRODUCTION

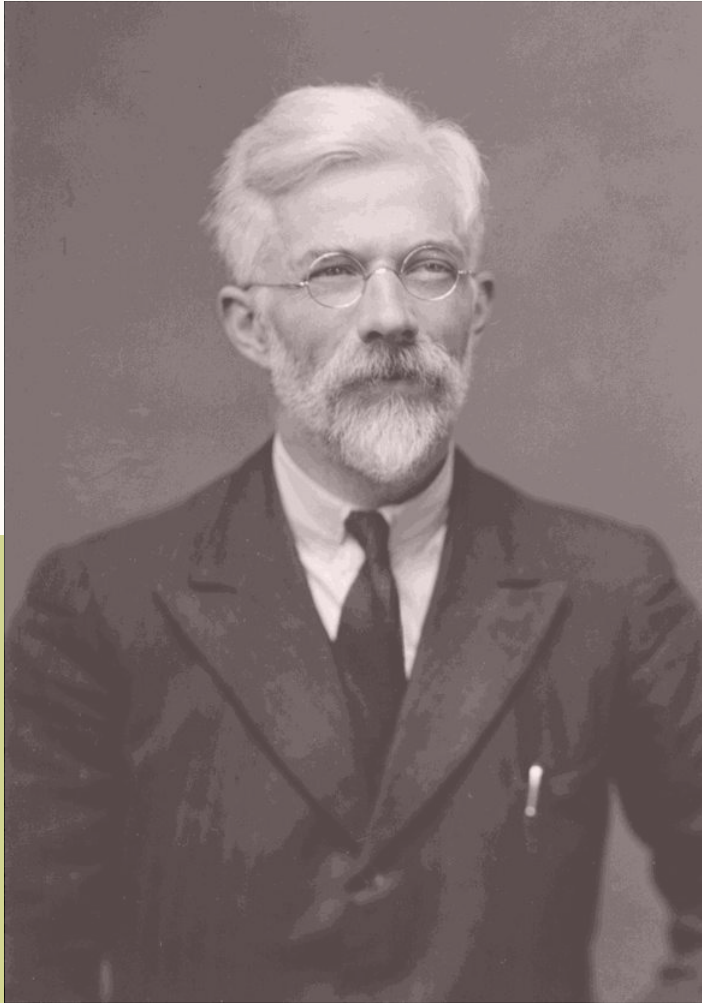
# Learning Outcome

- Describe appropriate knowledge of the fundamentals of software engineering.



# At the end of these lectures students are able to:

- Define the terms and concepts, “software”, and “software engineering”.
- Differentiate categories of software.
- Analyze main causes of software project failures.
- Explain the essence of practices.

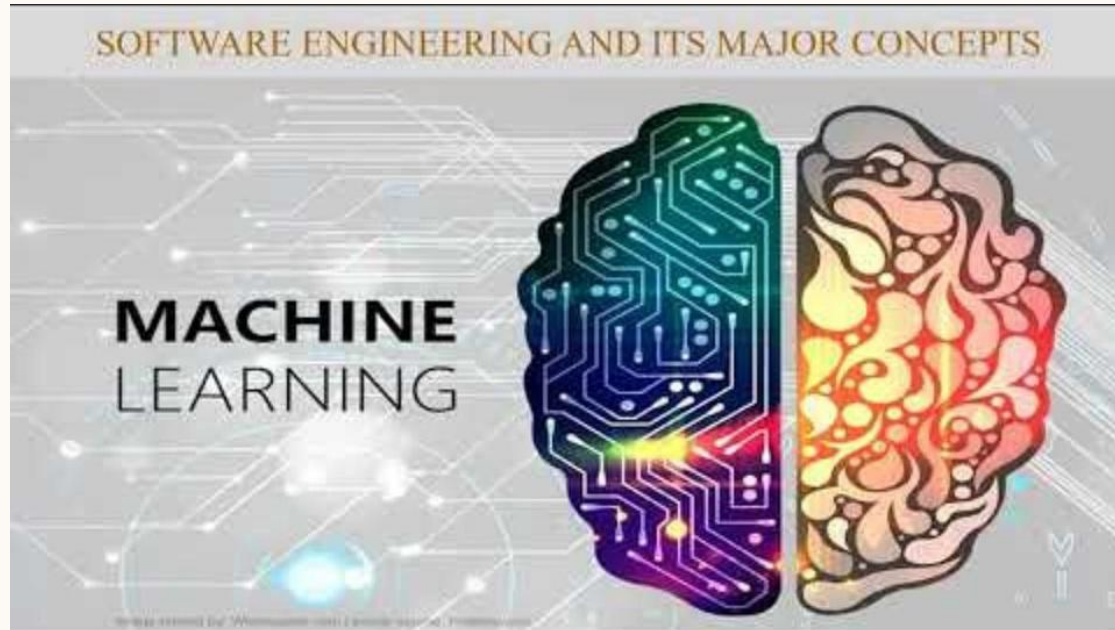


# INTRODUCTION TO SOFTWARE ENGINEERING

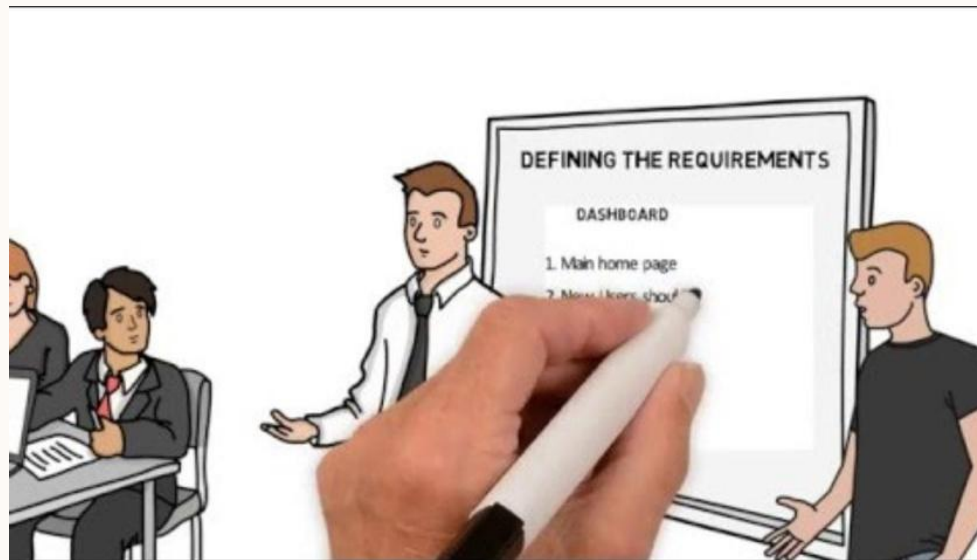


XIAMEN UNIVERSITY MALAYSIA  
廈門大學 馬來西亞分校

# INTRODUCTION TO SOFTWARE ENGINEERING



# INTRODUCTION TO SOFTWARE ENGINEERING



# Do I need a degree to be computer professional?

## PROGRAMMERS

write source code to  
program computers for  
specific tasks  
merging databases,  
processing online  
orders,

routing  
communications,  
conducting searches  
or  
displaying text and  
graphics  
interpret instructions  
from software  
developers and  
engineers

## SOFTWARE ENGINEERS

apply engineering  
principles to build  
software and systems to  
solve problems.  
use modeling language  
and other tools to devise  
solutions.  
adhere to the scientific  
method and must work in  
the real world.

## SOFTWARE DEVELOPERS

a less formal role than  
engineers.  
can be closely involved  
with specific project areas  
(including writing code)  
drive the overall software  
development lifecycle  
working across functional  
teams to transform  
requirements into  
features,  
managing development  
teams and processes, and  
conducting software  
testing and maintenance.

# DO I NEED A DEGREE TO BE COMPUTER PROFESSIONAL?

From Prisoner to Software Engineers



# DO I NEED A DEGREE TO BE COMPUTER PROFESSIONAL?

- [MIT OpenCourseWare | Free Online Course Materials Coursera |](#)
- [Online Courses From Top Universities. Join for Free Build new](#)
- [skills. Advance your career. | edX YouTube](#)

**THEY ARE ALL FREE!**

THEY ARE ALL FREE!



XIAMEN UNIVERSITY MALAYSIA

廈門大學 馬來西亞分校

# link

Students are required to view this video clip by following the link given to understand the concept of software engineering

- <https://www.youtube.com/watch?v=ilxZrYzJJ7I>
  - <https://www.youtube.com/watch?v=i-QyW8D3ei0>
- <https://www.youtube.com/watch?v=avdDEZCcluo>



# Software Principles and Aspects

- Software engineering is intended to support professional software development, rather than individual programming.
- It includes techniques that support program specification, design, and evolution, none of which are normally relevant for personal software development.

# What is software?

Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.

“Software is an instructions (computer programs), data structures and descriptive information.”

# Categories of computer software

## System Software

### Software Application Domains

- a collection of programs written to service other programs.

## Application Software

- stand-alone programs that solve a specific business need.

## Engineering/Scientific Software

- satisfies the needs of a scientific or engineering user to perform enterprise specific tasks.

## Embedded Software

- resides within a product or system and is used to implement and control features and functions for the end user and for the system itself.

# Categories of computer software

## Product-line Software

Software Application Domains

- designed to provide a specific capability for use by many different customers

## Web/Mobile Application

- browser-based apps and software that resides on mobile devices

## Artificial Intelligent Software

- non numerical algorithms to solve complex problems

# What are the attributes of good software?

- Good software should deliver the required functionality and performance to the user and should be maintainable, dependable, and usable

# What is software engineering?

- Software engineering is an engineering discipline that is concerned with all aspects of software
- **Software is not just the programs themselves but also all associated documentation and configuration data that is required to make these programs operate correctly.**

# Types of Software Products

- **Generic products:** Stand-alone systems that are produced by a development organization and sold on the open market to any customer who is able to buy them.
- Examples: databases, word processors, drawing packages, and project-management tools, library information systems, accounting systems, or systems for maintaining dental records

- **Customized products:** Systems that are commissioned by a particular customer. A software contractor develops the software especially that customer
- Examples: electronic devices, systems written to support a particular business process, and air traffic control systems

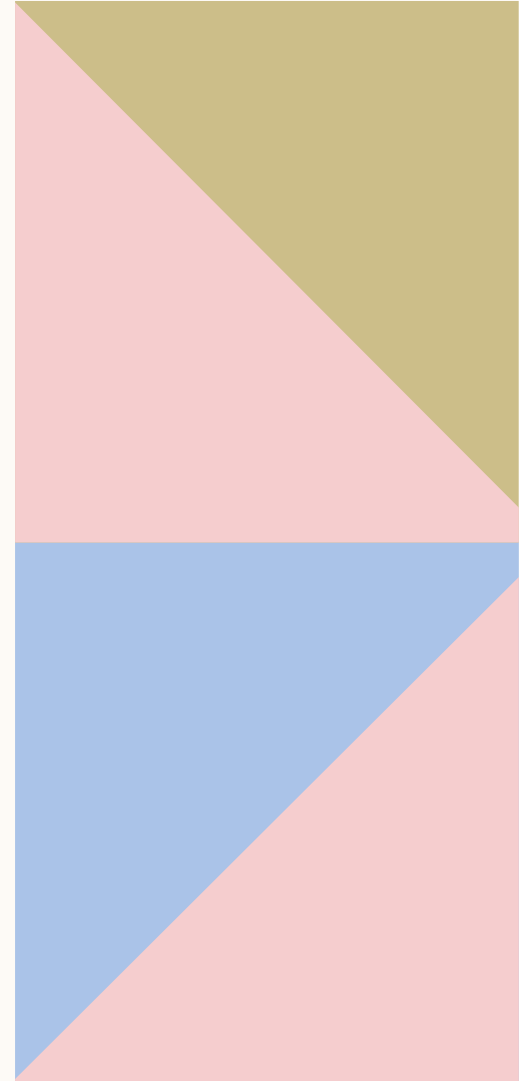


# Why Software Engineering is Important?

- More and more, individuals and society rely on advanced software systems. We need to be **able to produce reliable and trustworthy systems economically and quickly**.
- It is usually cheaper, to use software engineering methods and techniques for software systems rather than just write the programs as if it was a personal programming project. For most types of systems, the majority of costs are the costs of changing the software after it has gone into use

# WHY SOFTWARE ENGINEERING IS IMPORTANT?

- [Mars Climate Orbiter -Failure and Vacuity of Software engineering How](#)
- [lines of code made a rocket explode 11 of the most costly software](#)
- [errors in history · Raygun Blog](#)



# Software process activities

- **Software specification**, where customers and engineers define the software that is to be produced and the constraints on its operation.
- **Software development**, where the software is designed and programmed.
- **Software validation**, where the software is checked to ensure that it is what the customer requires.
- **Software evolution**, where the software is modified to reflect changing customer and market requirements.



# General issues that affect most software



## Heterogeneity

Systems are increasingly needed to operate as distributed systems across networks which include various types of computers and mobile devices.



## Business and social change

Business and society are evolving extremely rapidly. They need to develop their current software and create new software quickly to be in advance.



## Security and trust

We must ensure that unauthorized users are unable to target our applications, and that information is protected and secured.



## Scales

Software must be built over a broad range of scales, from very tiny embedded systems in portable or wearable devices through the Internet sizes, a cloud-based framework that serves a global population

# General issues that affect software

- **Heterogeneity**
  - Increasingly, systems are required to operate as distributed systems across networks that include different types of computer and mobile devices.
- **Business and social change**
  - Business and society are changing incredibly quickly as emerging economies develop and new technologies become available. They need to be able to change their existing software and to rapidly develop new software.

# General issues that affect software

- Security and trust
  - As software is intertwined with all aspects of our lives, it is essential that we can trust that software.
- Scale
  - Software has to be developed across a very wide range of scales, from very small embedded systems in portable or wearable devices through to Internet-scale, cloud-based systems that serve a global community.

# Software engineering diversity

- There are many **different** types of software system and there is no universal set of software techniques that is applicable to all of these.
- The software engineering methods and tools used depend on the type of application being developed, the requirements of the customer and the background of the development team.

# Application types

- Stand-alone applications
  - These are application systems that run on a local computer, such as a PC. They include all necessary functionality and do not need to be connected to a network.
- Interactive transaction-based applications
  - Applications that execute on a remote computer and are accessed by users from their own PCs or terminals. These include web applications such as e-commerce applications.
- Embedded control systems
  - These are software control systems that control and manage hardware devices. Numerically, there are probably more embedded systems than any other type of system.



# Application types

- **Batch processing systems**
  - These are business systems that are designed to process data in **large batches**. They process large numbers of individual inputs to create corresponding outputs.
- **Entertainment systems**
  - These are systems that are primarily for **personal use** and which are intended to entertain the user.
- **Systems for modelling and simulation**
  - These are systems that are developed by scientists and engineers to model physical processes or situations, **which include many, separate, interacting objects**.

# Application types

- Data collection systems
  - These are systems that collect data from their environment using a set of sensors and send that data to other systems for processing.
- Systems of systems
  - These are systems that are composed of a number of other software systems.

# Software engineering fundamentals

- Some fundamental principles apply to all types of software system, irrespective of the development techniques used:
  - Systems should be developed using a managed and understood development process. Of course, different processes are used for different types of software.
  - Dependability and performance are important for all types of system.
  - Understanding and managing the software specification and requirements (what the software should do) are important.
  - Where appropriate, you should reuse software that has already been developed rather than write new software.

# Internet software engineering

- The Web is now a platform for running application and organizations are increasingly developing web-based systems rather than local systems.
- Web services allow application functionality to be accessed over the web.
- Cloud computing is an approach to the provision of computer services where applications run remotely on the 'cloud'.
  - Users do not buy software buy pay according to use.

# Web-based software engineering

- Web-based systems are complex distributed systems but the fundamental principles of software engineering discussed previously are as applicable to them as they are to any other types of system.
- The fundamental ideas of software engineering apply to web-based software in the same way that they apply to other types of software system.

# Web software engineering

- **Software reuse**
  - Software reuse is the dominant approach for constructing web-based systems.
  - When building these systems, you think about how you can assemble them from pre-existing software components and systems.
- **Incremental and agile development**
  - Web-based systems should be developed and delivered incrementally.
  - It is now generally recognized that it is impractical to specify all the requirements for such systems in advance.

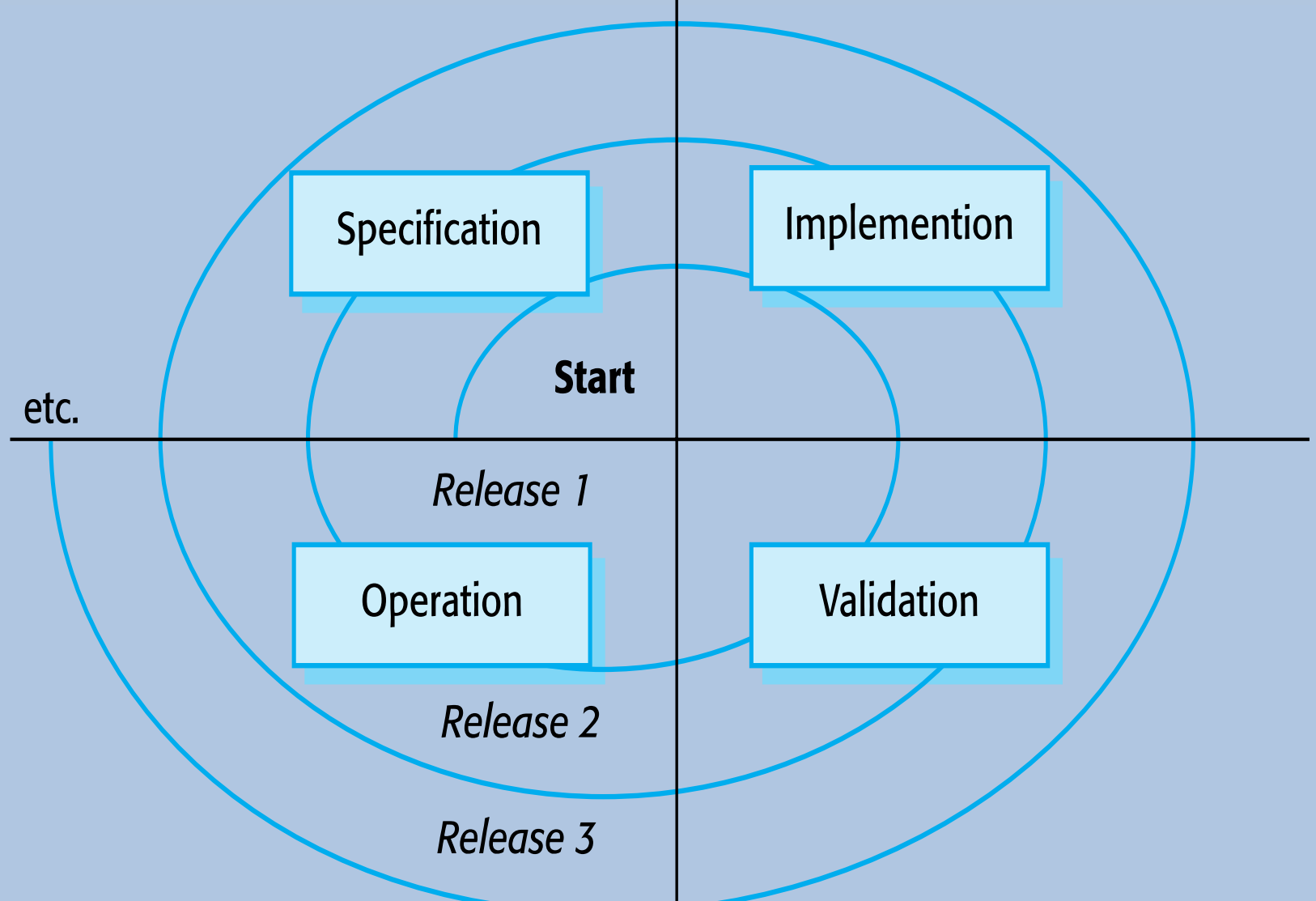
# Web software engineering

- **Service-oriented systems**
  - Software may be implemented using service-oriented software engineering, where the software components are stand-alone web services.
- **Rich interfaces**
  - Interface development technologies such as AJAX and HTML5 have emerged that support the creation of rich interfaces within a web browser.

# Software Evolution

- Software development does not stop when a system is delivered but continues throughout the lifetime of the system.
- After a system has been deployed, it inevitably has to change if it is to remain useful
- Software evolution may be triggered by **changing business requirements**, by **reports of software defects**, or by **changes to other systems** in a software system's environment. Useful software systems often have a very long lifetime. We should think it is as a spiral model of development and evolution.





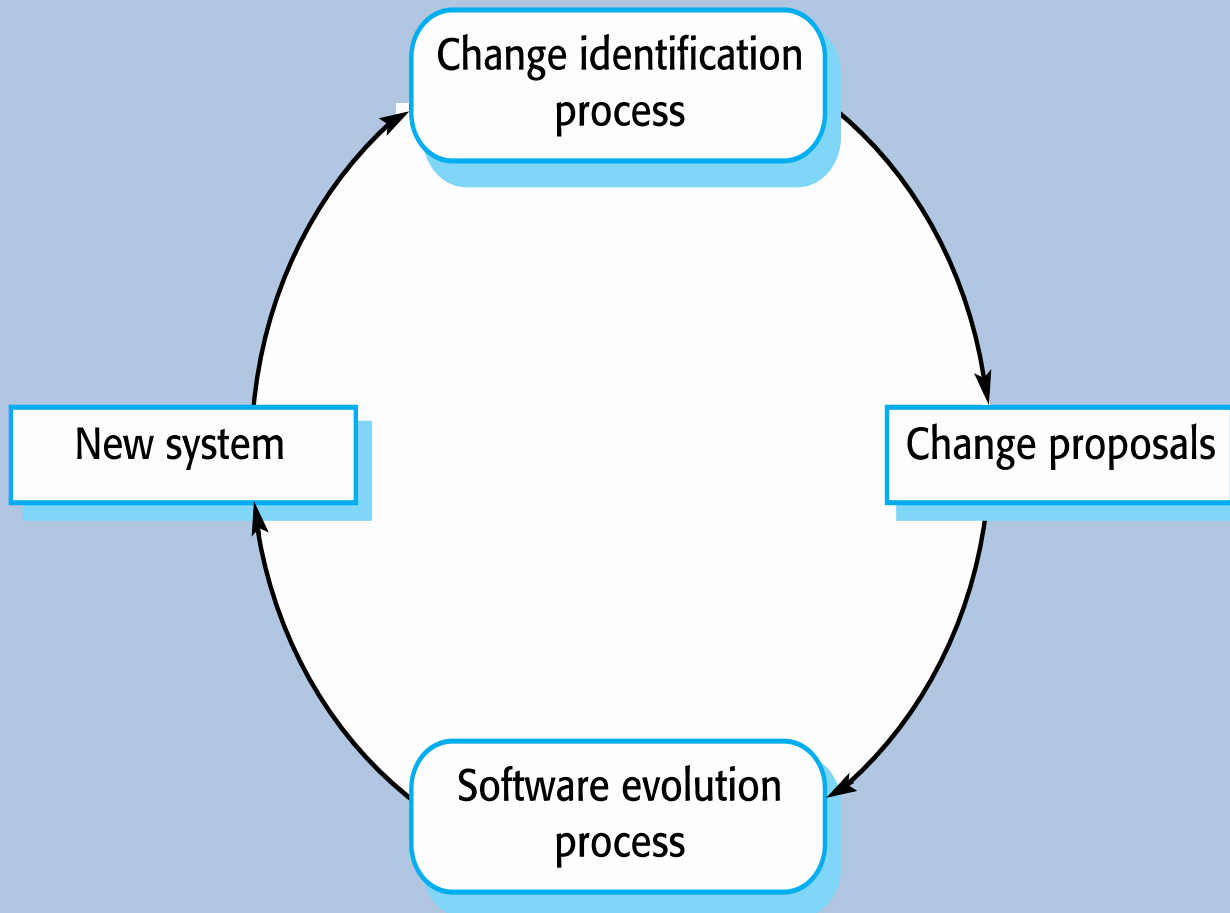
**Figure 1.1:** A spiral model of development and evolution

# Evolution Processes

- Software evolution processes vary depending on the type of software being maintained, the development processes used in an organization and the skills of the people involved
- System change proposals are the driver for system evolution in all organizations.

# Evolution Processes

- Change proposals may come from existing requirements that have not been implemented in the released system, requests for new requirements, bug reports from system stakeholders, and new ideas for software improvement from the system development team.
- The changes are implemented and validated, and a new version of the system is released



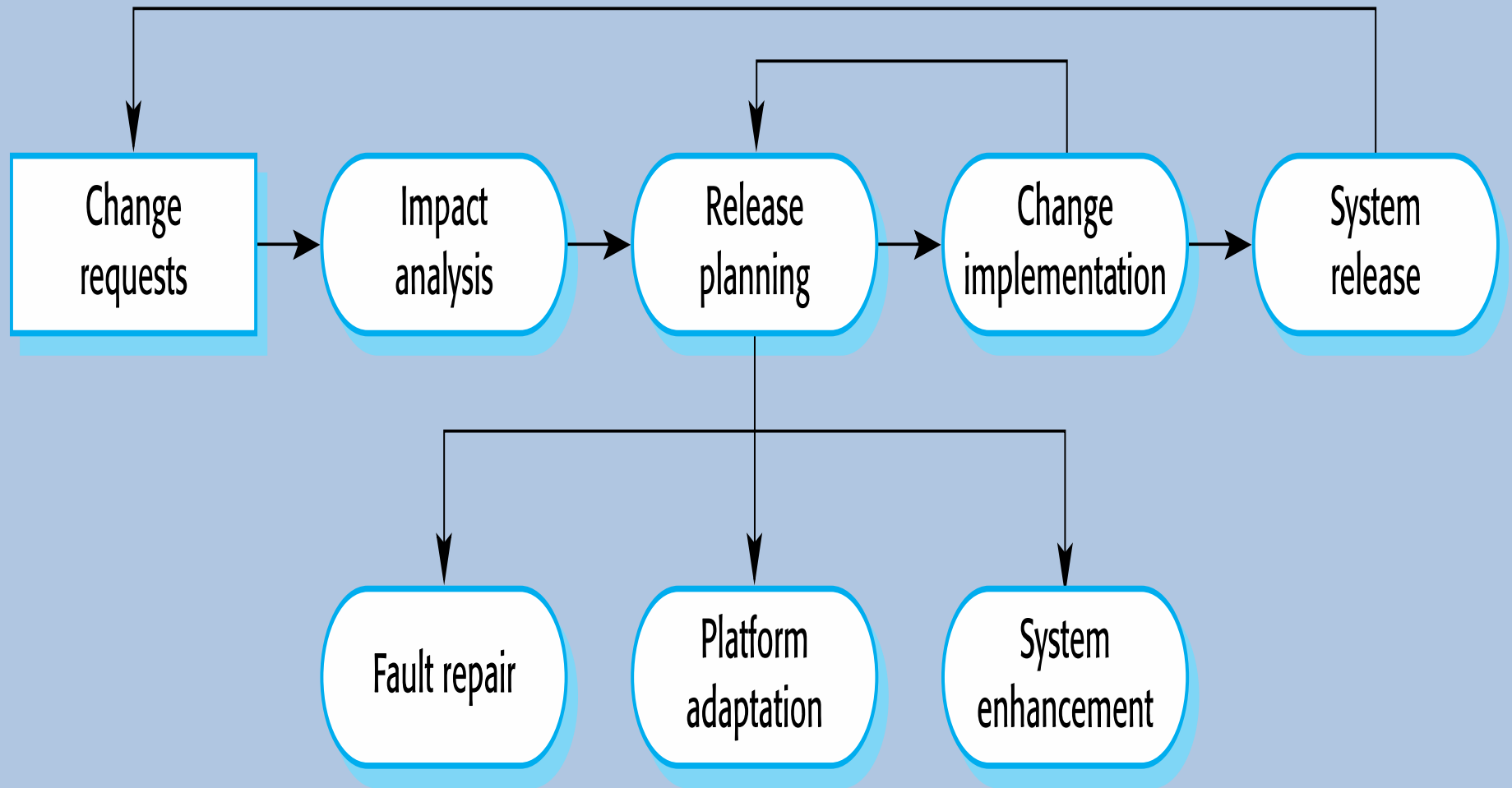


Figure 2: The evolution Processes

- Change requests sometimes relate to system problems that have to be tackled urgently. **There are 3 reasons for the urgent changes:**
  - ❖ If a serious system fault occurs that has to be repaired to allow normal operation to continue.
  - ❖ If changes to the systems operating environment have unexpected effects that disrupt normal operation.

- ❖ If there are unanticipated changes to the business running the system, such as the emergence of new competitors or the introduction of new legislation that affects the system.

# Exercises

1. Scenario: Imagine a scenario where a medium-sized healthcare facility is considering two options for managing their patient records and administrative processes. They are debating whether to invest in professional healthcare management software or to have custom software developed specifically for their needs. Explain why professional software would be the better choice in this scenario.
2. What is the most important difference between generic software product development and custom software development?