

# Software Lifecycle and Steps in Software Development

CPEN333 – System Software Engineering  
2021 W2  
University of British Columbia

© *Farshid Agharebparast*



# Introduction

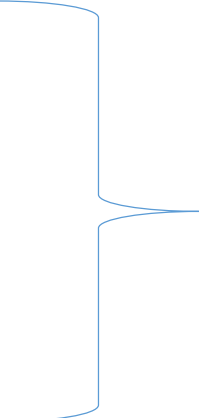
---

- Although software products are very varied, successful software development projects all need to address similar issues.
- So we can discuss a number of software development steps that should be part of all software projects.
- In this set of slides, we will discuss these software process steps and we will discuss examples of software engineering life cycle.

# Basic software development steps

---

- **Feasibility**
- **Requirements**
- **System/program design**
- **Implementation**
- **Acceptance and release**
- **Maintenance**



These steps may be repeated as many times as needed

We are distinguishing among these software process steps to be able to clearly identify what we are doing at any given time, allow review of work and specify actions to perform next.

Considerations for **testing**, **security** and **performance** are part of many of these steps.

# Note

---

- Normally the terminology used is “software process” or “software process steps”.
- Since we have spent a lot of time in this course using the term *process* for a different concept, we have used the term process as rarely as possible as a part of the terminology here.

# Steps

---

- **Feasibility:** The feasibility study helps us make a **go or no-go decision**.
  - ❖ Is it technically feasible?
  - ❖ What are the available resources?
  - ❖ What are the objectives and scope of the project?
  
- **Requirements:** They are the **descriptions of the services** that a system should provide and the **constraints** on its operation.
  - ❖ Since they are to reflect the needs of the costumer (**client's viewpoint**), the requirements are developed with direct influence and consultation with the client.

# Steps (cont.)

---

- **Design** describes the system from the software **developer's** viewpoint
- **System design**: the whole system design including hardware
- **Program design**: the software design considering implementation of a number of units or modules
  - ❖ We can use models (for example UML) to communicate the program design.

# Steps (cont.)

---

- **Implementation:** coding
  - ❖ **Testing** is also an integral part of implementation.
- **Acceptance:** It is the testing of the system by the client against the requirements.
  - ❖ If it is accepted, a release will be scheduled for the system to be delivered to the client.
- **Maintenance:** It is the general process of changing a system after it has been delivered:
  - ❖ Bug fixes and security patches after it is put in practical use
  - ❖ System evolution over time if the requirements change or if there is a need to adapt to a new technical environment.

# Sequences of the steps

---

- Every software project includes these basic steps in some shape and form.
  - ❖ We may carry them out in various sequences
  - ❖ Some steps may be less formal.
- The bigger the project and the more involved it is, the more important it is to decide and follow specific software development process.
- Software development **lifecycle** is a clearly defined series of phases/steps through which software is produced.
  - ❖ This can take days to years to complete.



# Some Lifecycle Models

---

- **Code-and-fix:** write some code, fix it when it breaks, repeat
- **Waterfall model:** perform each step in order with feedback
- **Agile (Incremental/iterative development):** project is divided into a large number of small tasks, each to be completed incrementally using iterations

# Code-and-Fix

---

## ➤ **some advantage:**

- ❖ applicable to very small and short-lived projects
- ❖ can see progress quickly with little to no overhead

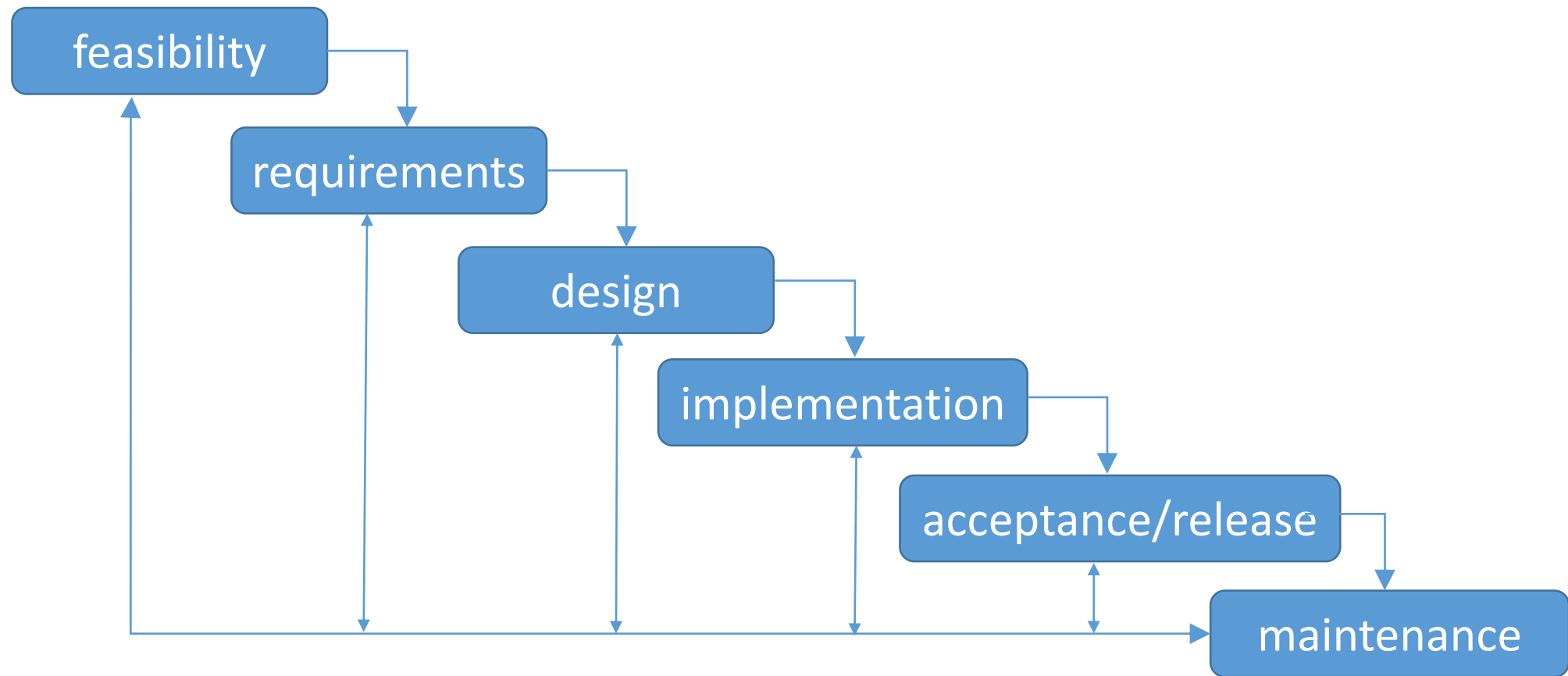
## ➤ **some disadvantage:**

- ❖ no good way to assess progress, quality, or risks
- ❖ accommodating change normally means a major design overhaul

➤ The lack of a process is the process here.

# Waterfall

- Waterfall model is the traditional sequential approach to software development



# Waterfall (cont.)

---

## ➤ **Some advantages:**

- ❖ suitable for projects that are well understood but complex
  - tackles most planning upfront
- ❖ no midstream change so could be efficient

## ➤ **Some disadvantages:**

- ❖ not easy to have most planning upfront
  - requirements must be very clear and well-understood
- ❖ not much adaptable
- ❖ no sense of progress until the end

# Waterfall (cont.)

---

- It is used for long-term, expensive, critical projects



image source: lockheedmartin.com

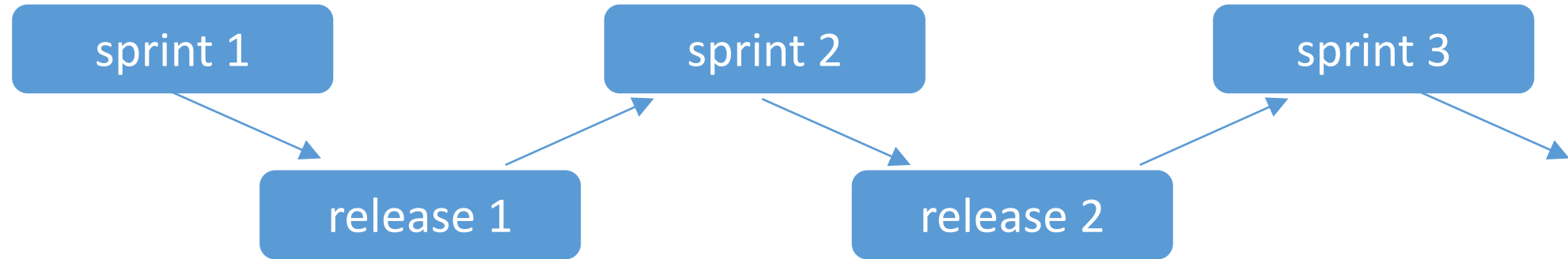


image source: nasa.org

# Agile

---

- An incremental development model with iterations
- The project is divided into a number of smaller iterations, known as **sprints**, to incrementally develop.
- Each sprint includes its own planning, design, coding, ... and **release** (e.g. a small-scale waterfall)



# Agile (cont.)

---

## ➤ **Some advantage:**

- ❖ easier for a small team to develop a small sprint correctly
  - e.g. start-up company developing a web-based service
- ❖ allows early release (payback on investment begins soon)
- ❖ can get feedback from customers early and that can be incorporated in later sprints

## ➤ **Some disadvantage:**

- ❖ requirements and design emerge incrementally, so inevitably some early sprints must be reworked
  - this requires changes to code that is already released (maybe awkward)
- ❖ product must be decomposable

# References

---

➤ ***Software Engineering***, I. Sommerville