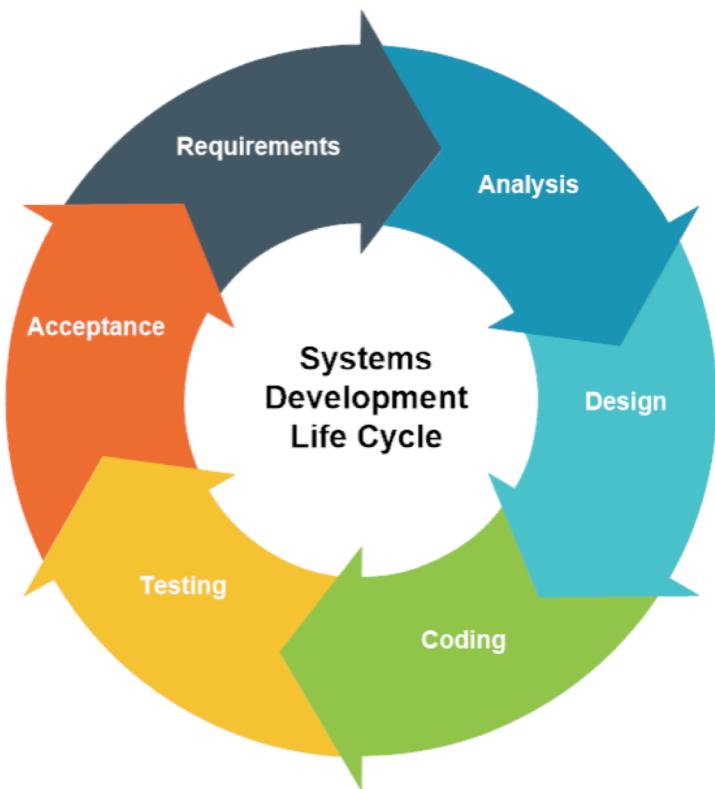


Software Development Life Cycle (SDLC)



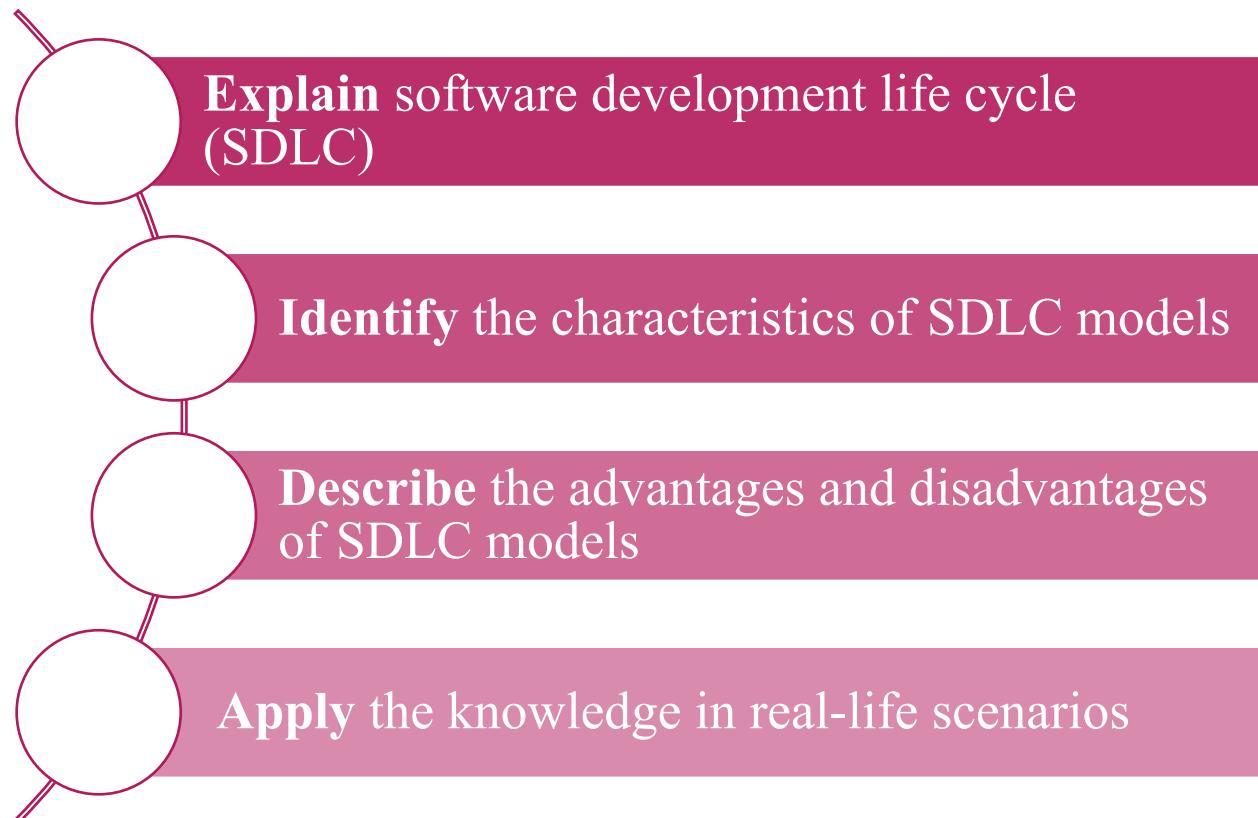
Presenter:
Md Alamgir Kabir
ID: 55340650
Department of Computer Science
City University of Hong Kong

Topics

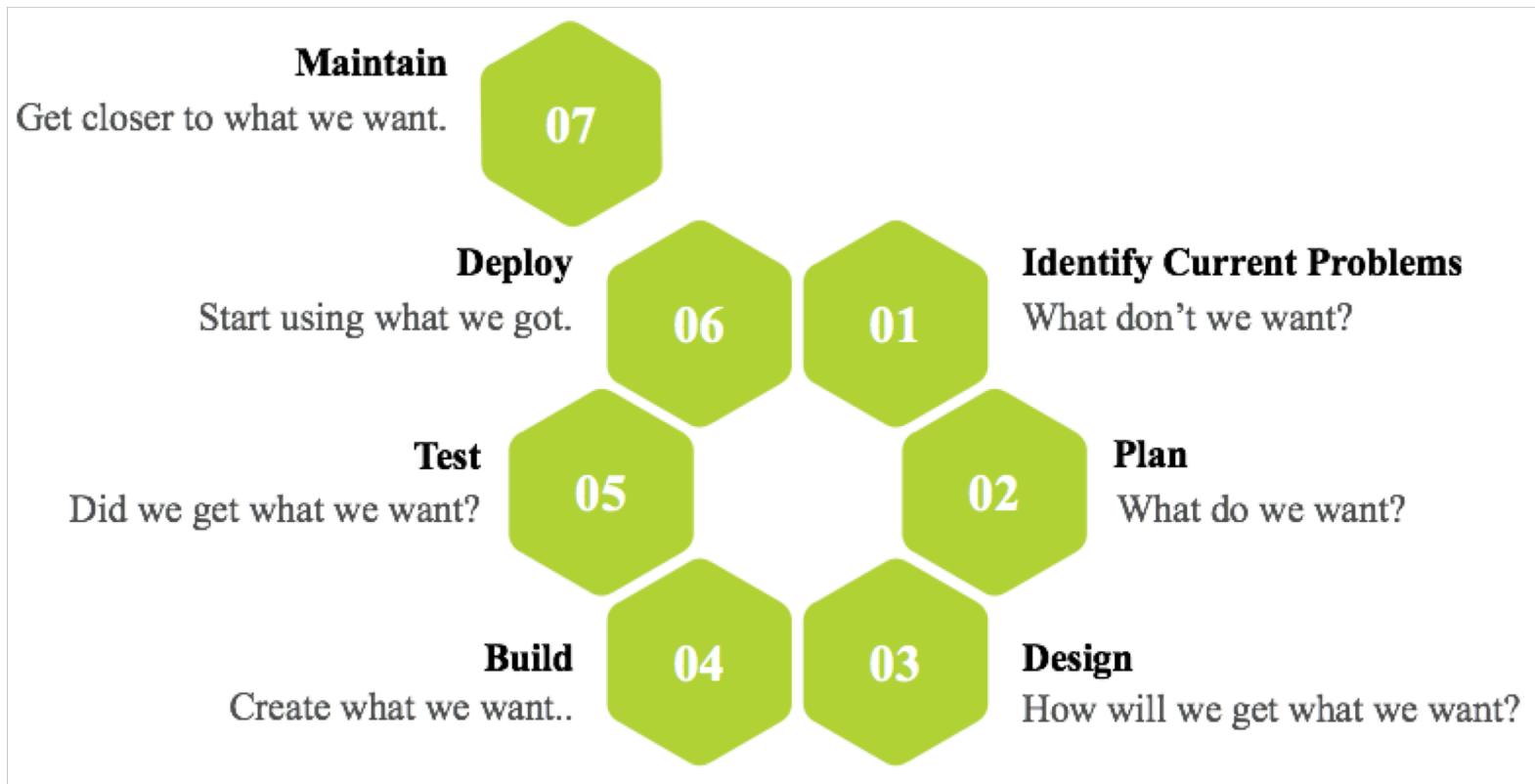
- Intended learning outcomes
- Software development life cycle
- Waterfall Model
- Prototyping Model
- Assessment

Intended Learning Outcomes

Upon completion of this lecture the audience should able to:



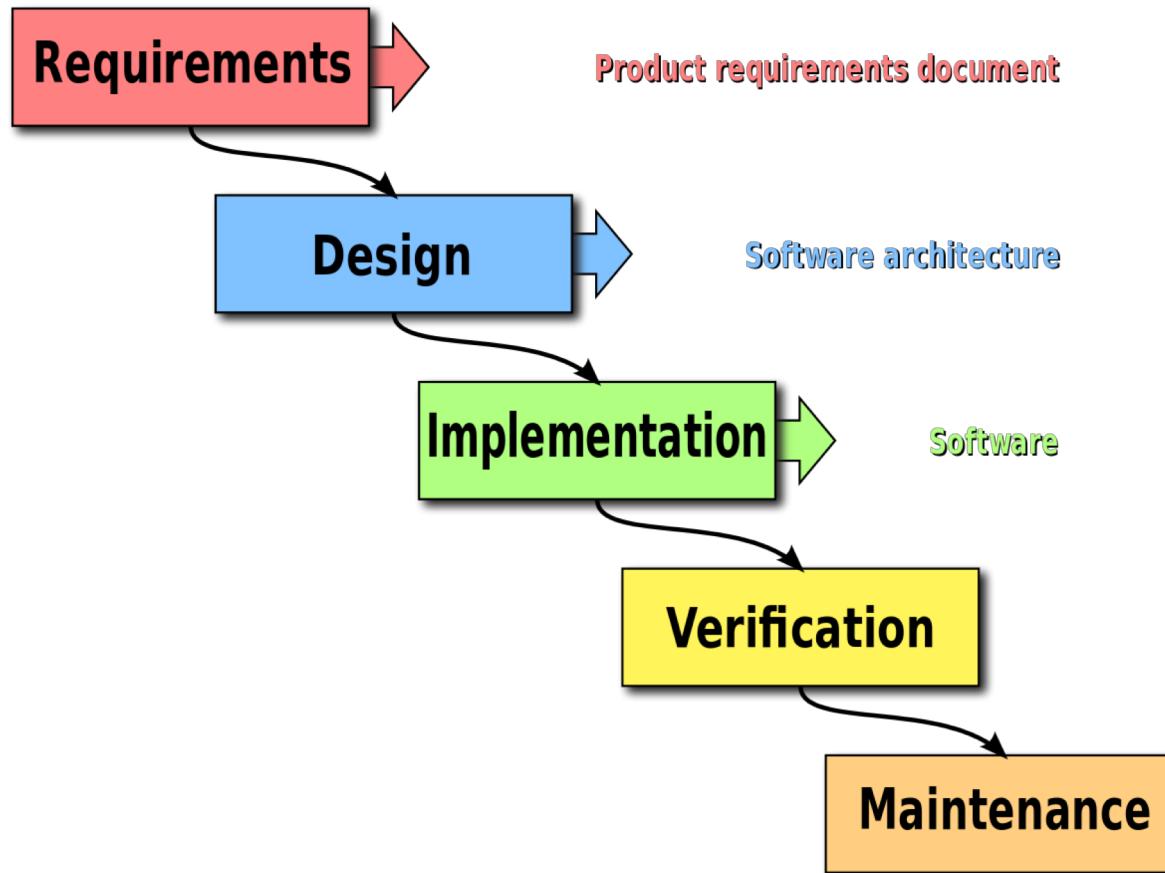
Software Development Life Cycle (SDLC)



SDLC Models

Waterfall and Prototyping

Waterfall Model



Strength and Weakness of Waterfall Model

Strength

- Easy to understand
- Provides structure
- Milestones are well understood
- Good for management control
- Works well when quality is more important than cost or schedule

Weakness

- All requirements must be known upfront
- Deliverables created for each phase are considered frozen – inhibits flexibility
- Integration is one big bang at the end
- Little opportunity for customer to preview the system

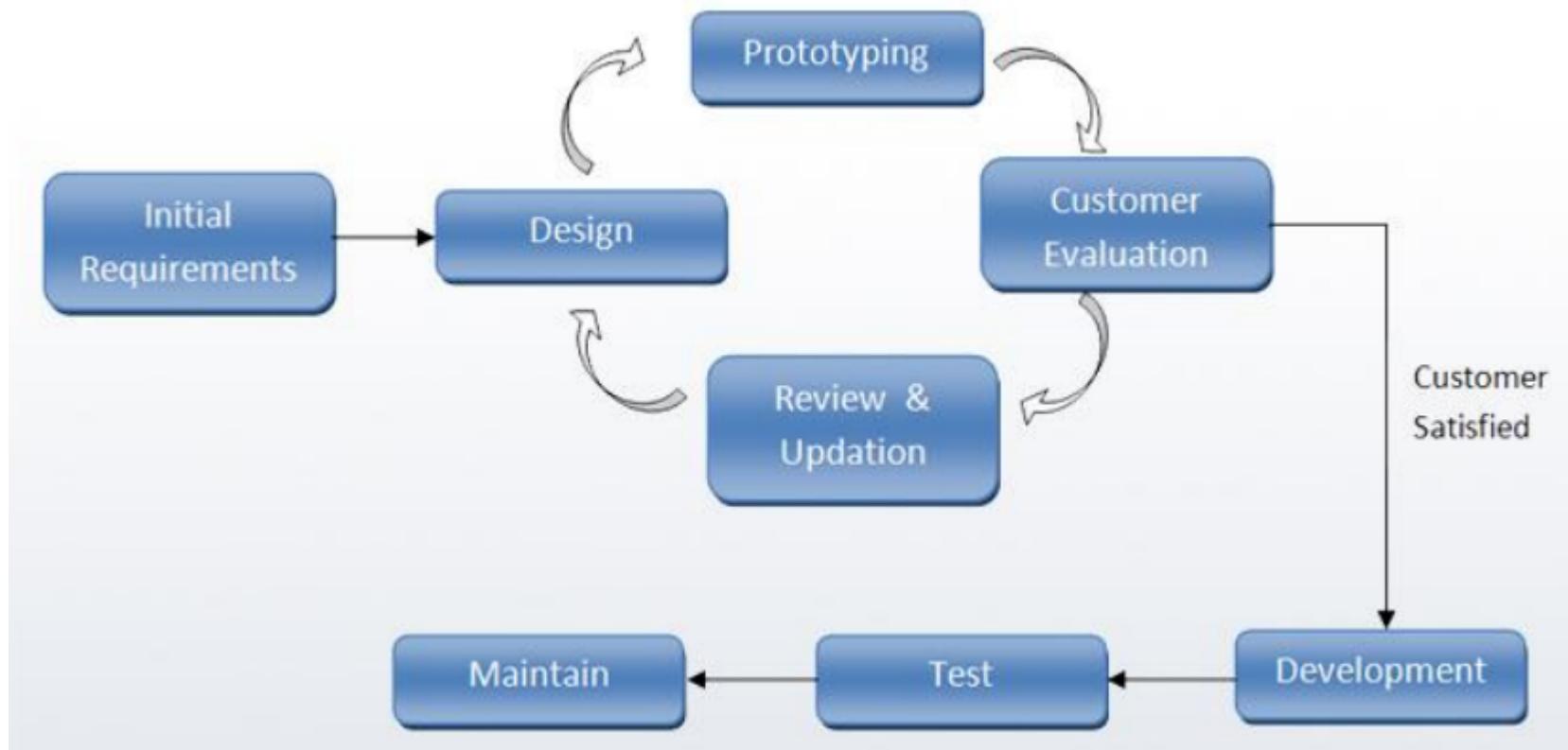
When to Use the Waterfall Model

-  Requirements are very **well known**
-  Product definition is **stable**
-  Technology is **understood**
-  New **version** of an existing product
-  Porting an **existing product** to a new platform

Prototyping Model

- Developers build a prototype after getting all the requirements
- Prototype is evaluated by end users
- Users give corrective feedback
- Developers further refine the prototype
- When the user is satisfied, the prototype code is brought up to the standards needed for a final product

Prototyping Model



Strength and Weakness of Prototyping Model

Strength

- Customers can “see” the system requirements as they are being gathered
- Developers learn from customers
- A more accurate end product
- Unexpected requirements accommodated
- Allows for flexible design and development
- Steady, visible signs of progress produced
- Additional needed functionality

Weakness

- Bad reputation for “quick-and-dirty”
- Overall maintainability may be overlooked
- The customer may want the prototype delivered
- Process may continue forever

When to Use the Prototyping Model

- Requirements are unstable or have to be clarified
- As the requirements clarification stage of a waterfall model
- Develop user interfaces
- Short-lived demonstrations
- New, original development

“You’ve got to be very careful if you don’t know where you’re going, because you might not get there.”

Yogi Berra

Scenario - Assessment

Let us take an example for making a website for a shopping mall which contains many shops. Making a website for a shopping mall is not an easy task because it requires lots of time. It is also not possible to gather all the requirements of all the shops at once.

Which SDLC model is perfect for developing this system?

Scenario – Answer

Developing a website for a mall the waterfall model is not appropriate

Sources for Further Studies

1. Book: The Software Development Lifecycle - A Complete Guide by Richard Murch
2. Sommerville, I. (2011). Software engineering 9th Edition. *ISBN-10, 137035152.*
3. Davis, A. M., Bersoff, E. H., & Comer, E. R. (1988). A strategy for comparing alternative software development life cycle models. *IEEE Transactions on software Engineering*, 14(10), 1453-1461.

Question?

Thanks for
Listening!
