

Software Engineering Introduction

- Himasri Lekkala
- CSCE Graduate student

Software Engineering

Application of engineering methodologies to design, develop and maintain a software within budget, on schedule and with high quality.



Why Software Engineering?

1

Helps in utilizing the budget effectively

2

Delivers the product on schedule

3

Makes possible to handle large projects

4

Increases reliability of a software

5

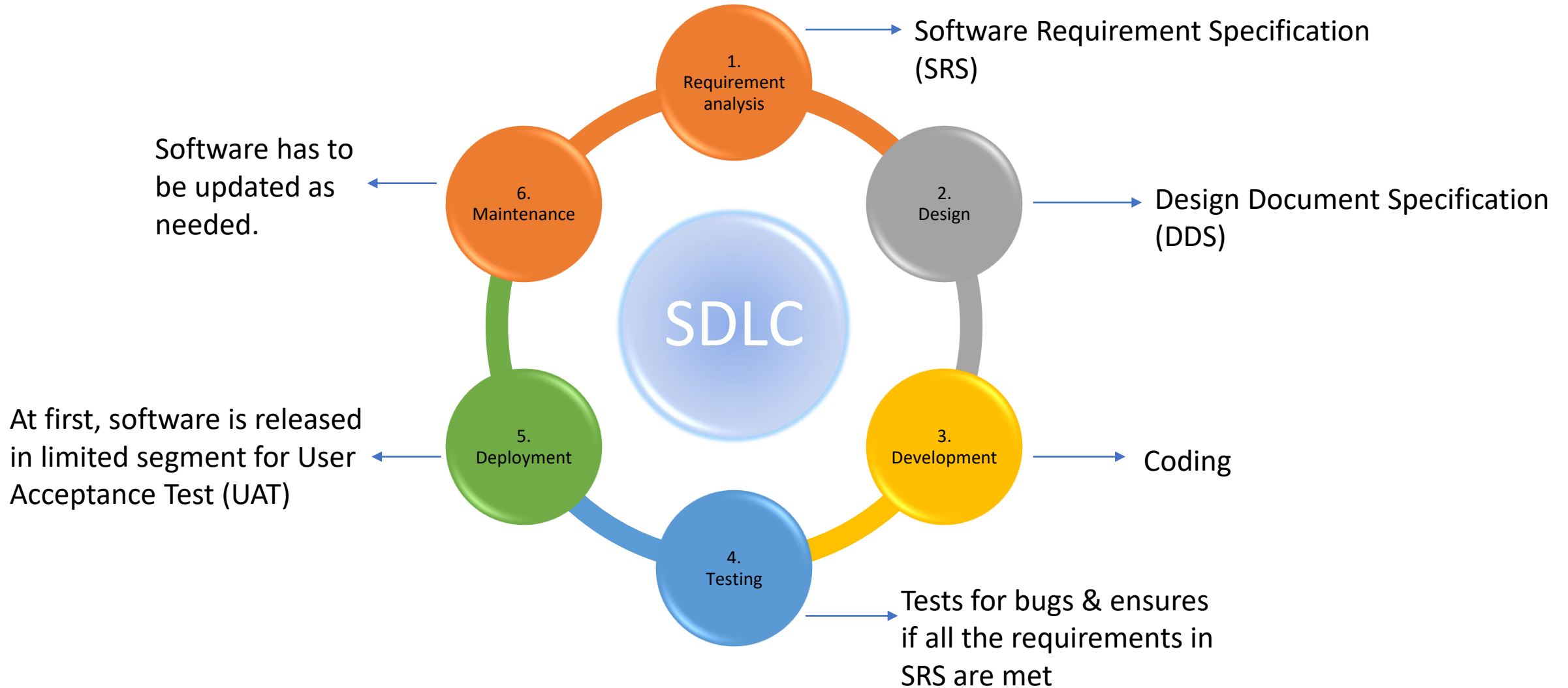
Customer satisfaction

Little Bug, Big Bang

- Took 10 years and \$7 billion to produce Ariane 5, a giant rocket.
- Exploded in less than a minute after its launch due to a small bug in the software.

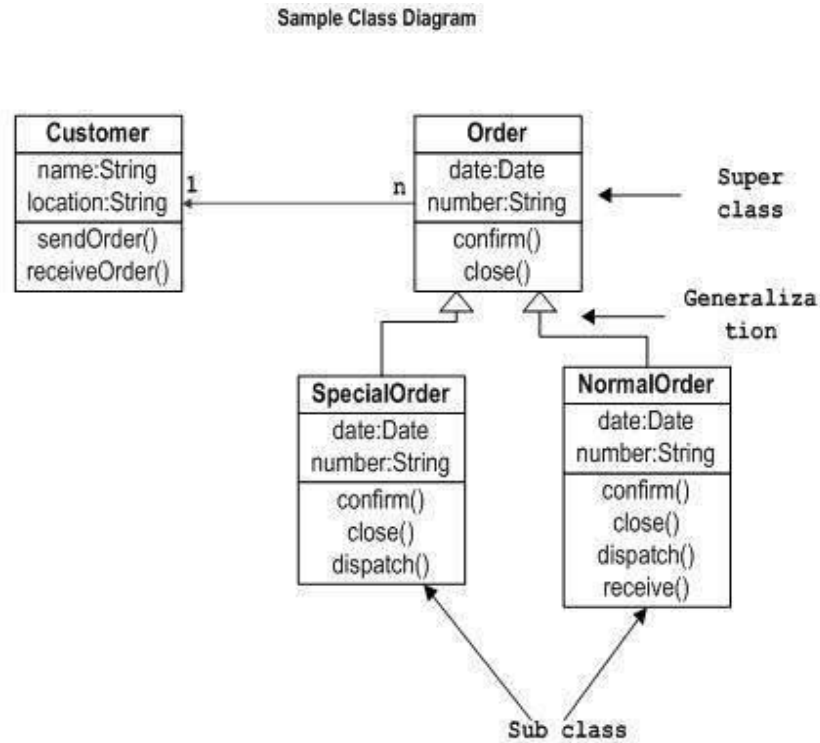


Software Development Life Cycle (SDLC)

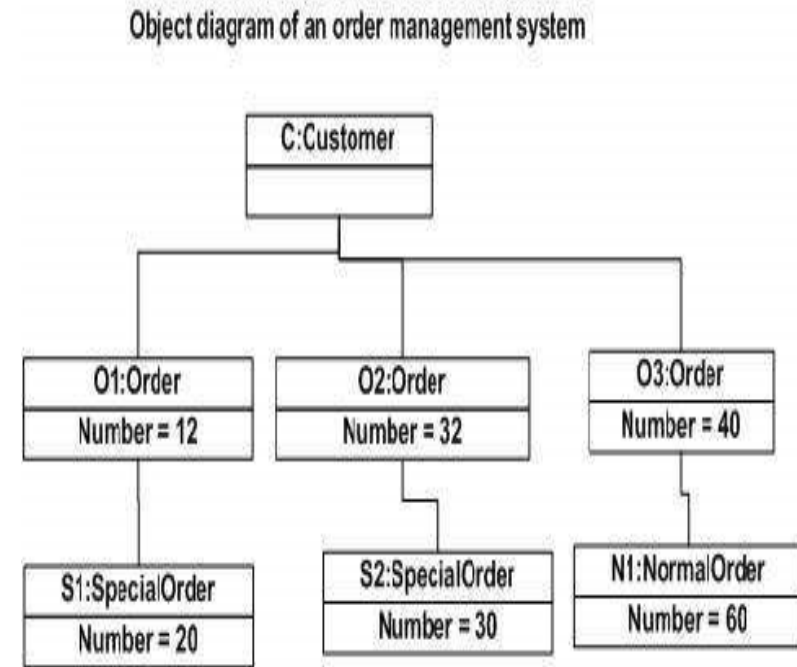


UML Diagrams (Unified Modeling Language)

Class diagram



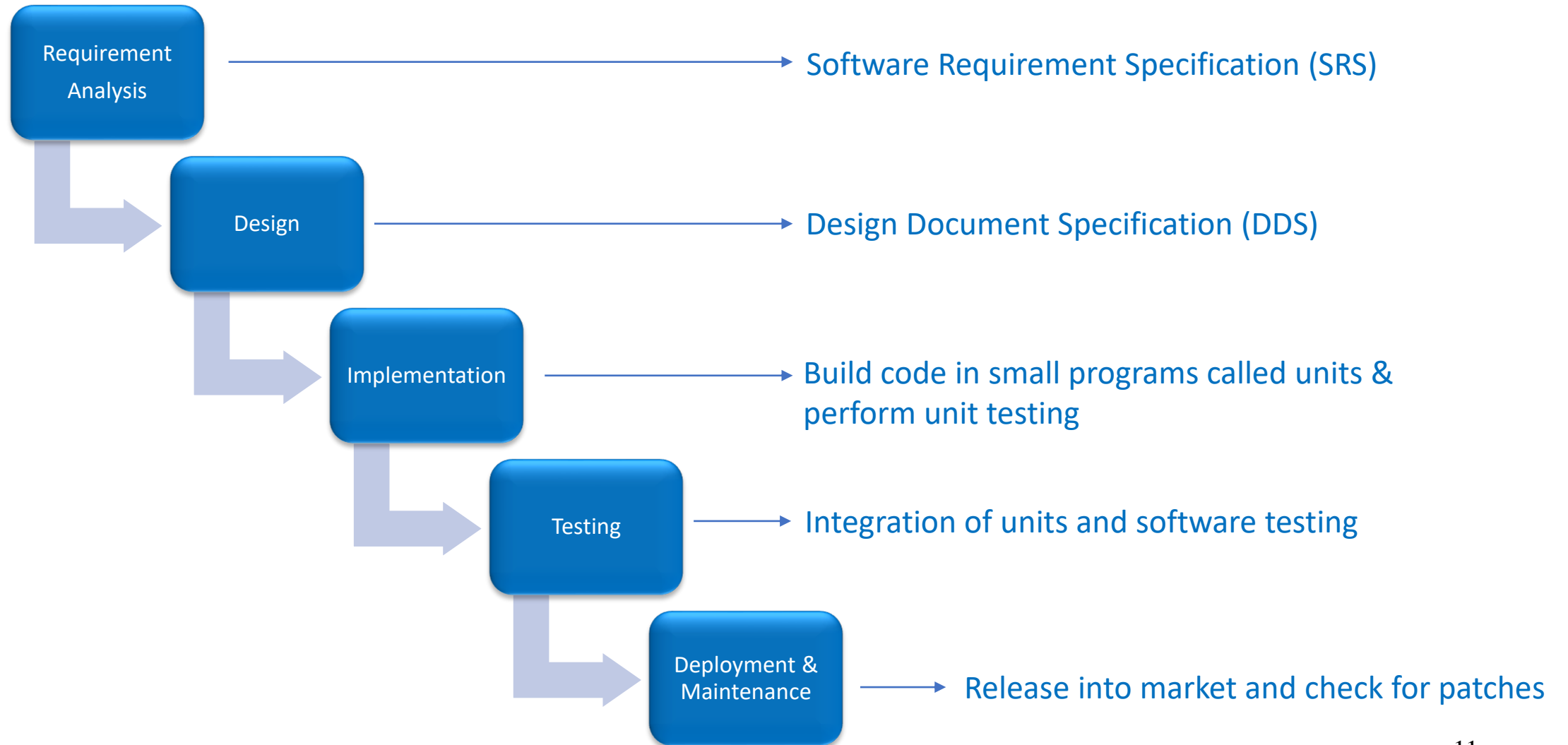
Object diagram



SDLC Models

- Waterfall Model
- Incremental Model
- Spiral Model
- Agile Model
- Chaos Model
- Scrum Model
- V-Model
- Sawtooth Model

Waterfall Model



Waterfall Model - Application

1

Requirements are very well documented, clear and fixed.

2

Ample resources with required expertise are available to support the product.

3

Technology is understood and is not dynamic.

4

Suitable for short term projects.

Waterfall Model – Pros and Cons

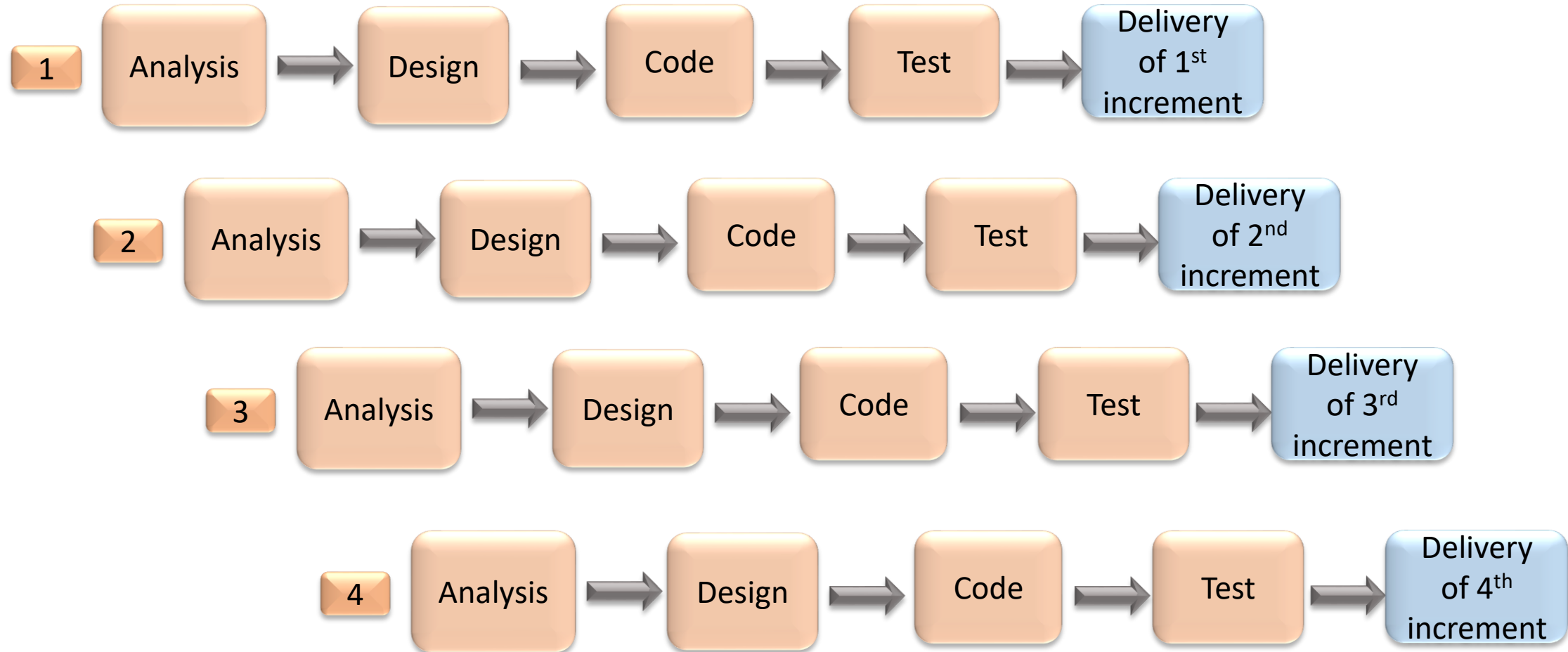


- Simple and easy to understand
- No overlap of phases, avoiding ambiguity
- Works well for small projects where requirements are well understood
- Easy to manage



- Not Agile – not flexible to accommodate changes
- No working software till the end of the SDLC
- Not suitable for long term projects
- Difficult to fully specify requirements at the beginning

Incremental Model



Incremental Model – Characteristics

(Waterfall model +
iterative feature) =
Incremental model

Partial systems are
built to produce the
final system.

First tackles highest
priority requirements
to deliver core product

New
features/functionalities
are added in the form
of increments.

Incremental Model – Pros and Cons

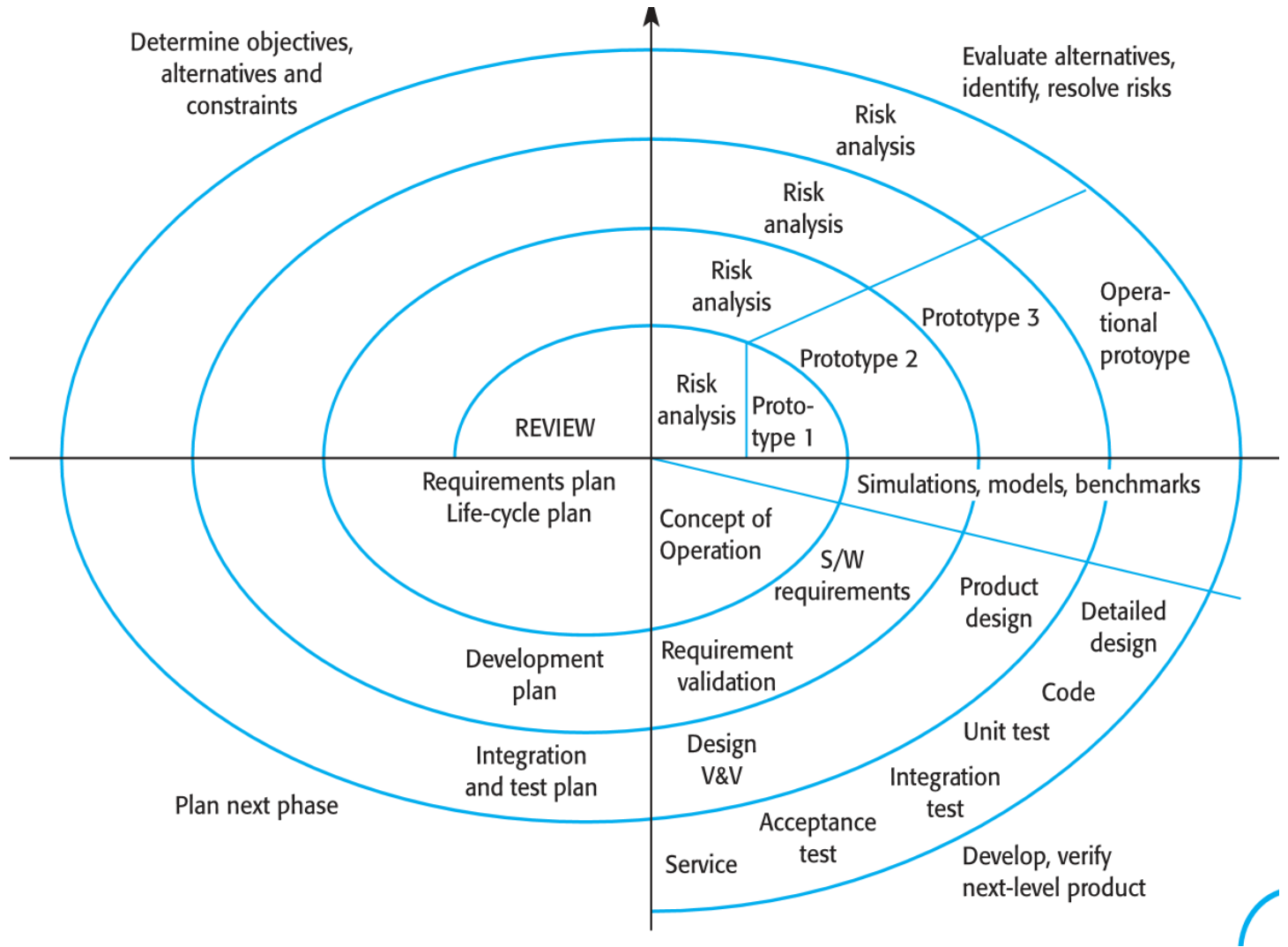


- Flexible to changes in requirements
- Working software is available immediately after first iteration
- Facilitates customer evaluation/feedback after each iteration
- Can start with initial requirements and expertise

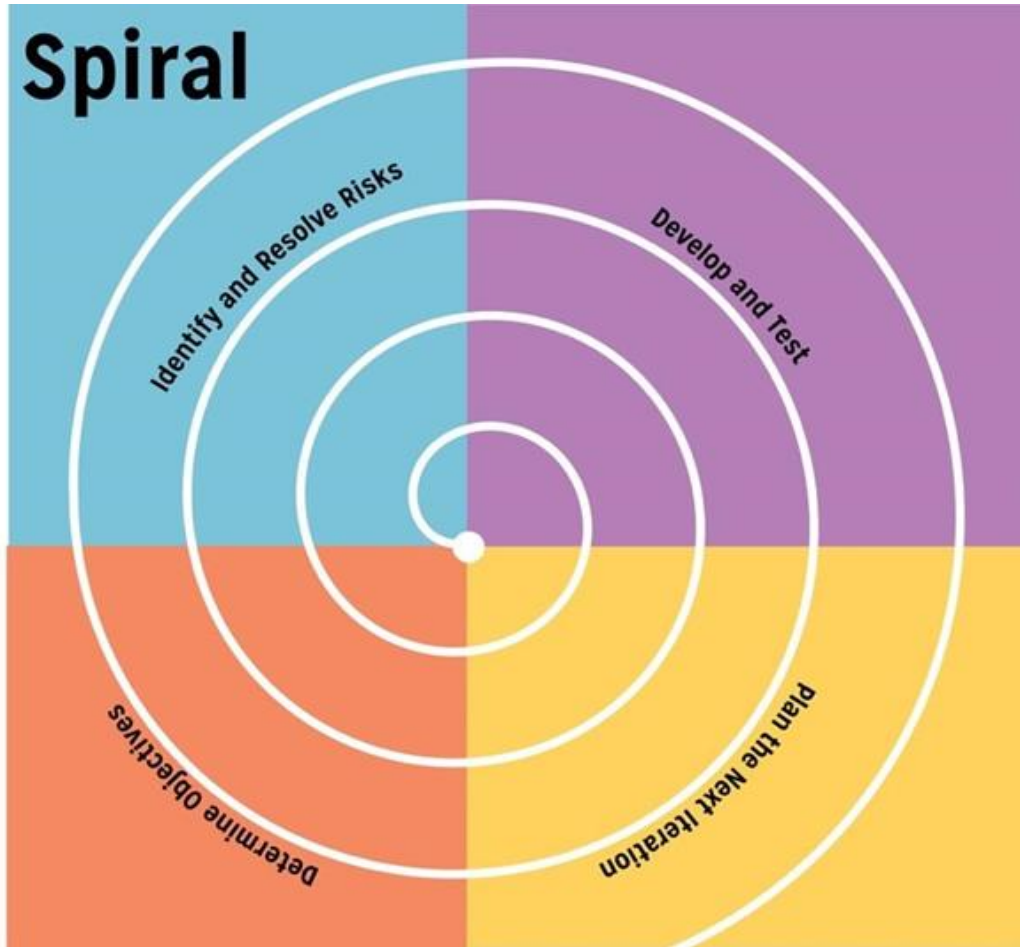


- Applicable only to bulky projects
- End of the project may not be known
- Hard to decide on size of each increment
- Difficult to manage

Spiral Model



Spiral Model



- Q1-Determine objectives
- Q2-Identify and resolve risks
- Q3-Develop and test
- Q4-Plan the next iteration
- A software project repeatedly passes through these phases in spirals
- First spiral is baseline spiral

Spiral Model

- First described by Barry Boehm in 1986
- Also an Linear + Iterative model like Incremental model
- Special emphasis is given to risk analysis
- When there is a budget constraint and risk evaluation is important
- Medium to high-risk projects
- When customer is not sure of the requirements

Spiral Model – Pros and Cons



- Special emphasis on risk management
- Flexible to changes in requirements
- Working software is available after every iteration
- Facilitates customer evaluation/feedback after each iteration



- Complex
- Spiral can go into an infinite loop
- Difficult to manage
- Large number of intermediate stages requires excessive documentation
- Expensive for small projects or low risk projects

Agile Model

Major focus on customer satisfaction by rapid delivery.

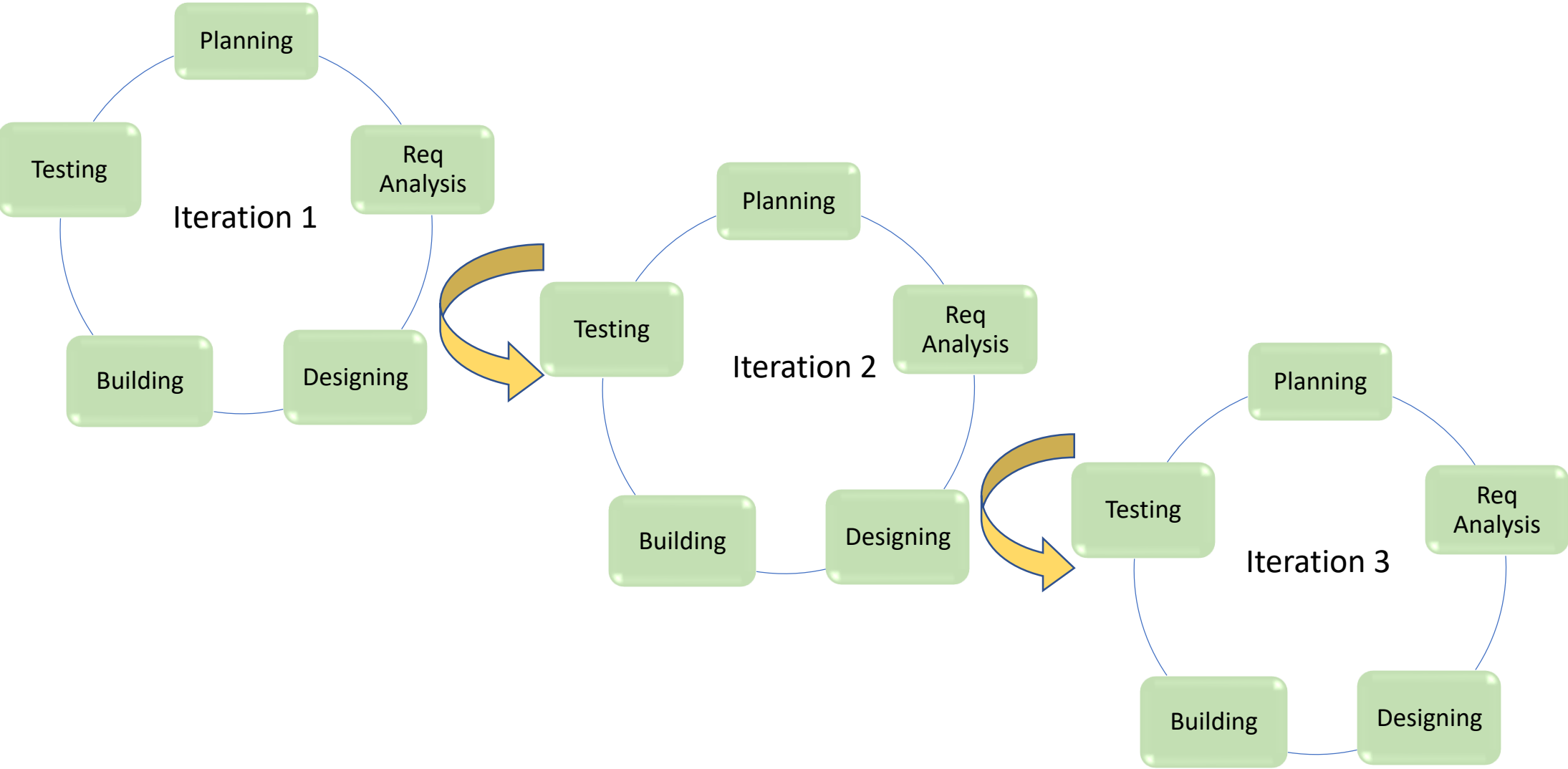
Break the product into small builds and run them in iterations.

Each iteration typically lasts 1 to 3 weeks.

Each build adds a new feature.

Cross functional teams work simultaneously on different phases in each iteration.

Agile Model



Agile Model – Pros and Cons



- Rapid delivery
- Minimal documentation – focus on development



- Transfer of technology to new team members will be challenging due to lack of documentation
- Depends heavily on customer interaction
- Depends on teamwork which may not go well sometimes
- Difficult to manage

Introduced in 1995 by L.B.S. Raccoon

Chaos strategy is “Always resolve the most important issue first.”
Here, the issue is an incomplete programming task.

- The most important issue is a combination of big, urgent, and robust:
- Big issues : Which provide value to users in the form of functionality.
- Urgent issues : Which will hold up other work until they are resolved.
- Robust issues : That are trusted and tested when resolved.
- To resolve an issue is to bring it to a point of stability.

The behavior of a complex system emerges from the combined behavior of the smaller building blocks.

The chaos model's relationship to chaos theory is the idea that large-scale architectural issues cannot be stabilized without also stabilizing the "smaller" issues in the software. Including the individual lines of code.

It endeavors to unify the best programming methodologies with the best project management techniques

Chaos Model

Factors to be considered in choosing a software model

1. Requirements



2. Time



3. Risk Involved



4. Budget



5. Customer Interaction



6. Expertise



1. A software system is to be developed for which the requirements are well understood and the risk of failure is minimal. To meet these requirements, which of the following software development models would be most appropriate to use? (From Praxis 5651 help guide)

A. Chaos

B. Incremental

C. Spiral

D. Waterfall

2. You and your development team have been commissioned to work on a database for a major bank. For obvious reasons, your client is very concerned with security. You and your team come up with many security features that could be implemented into the product.

In what phase of a software life cycle process would this task occur?

- A. Planning and Defining
- B. Design and Implementation
- C. Verification and Validation

3. You and your development team are working hard to implement some agile practices into your development. One practice you chose to adopt is delivering a working prototype to your client every 2 weeks for feedback.

What phase of a process does this practice belong?

- A. Specification
- B. Design and Implementation
- C. Verification and Validation

4. In the Waterfall model, what phase or kinds of activities would happen right after the product has been implemented?

A. Design

B. Maintenance

C. Testing

D. Requirement gathering

5. In the Spiral software process model, each iteration begins in which quadrant or phase?

- A. Determine objectives
- B. Plan the next iteration
- C. Develop and test
- D. Identify and resolve risks

6. Which of the following represents Linear process model ?
- A. Each phase happens sequentially, and then loops back to the beginning when all the phases are complete.
 - B. Each phase happens in parallel with other phases until the product is done, with no repetition between or within phases.
 - C. Each phase happens sequentially, and never loops or repeats.
 - D. Each phase can be repeated until the product is complete.

Match the following - 1

Spiral model

Waterfall model

Incremental model

Agile model

The first phase in SDLC

The last phase in SDLC

(a) rapid delivery

(b) requirement gathering

(c) Maintenance

(d) Special emphasis on risk analysis

(e) new features are added in the form of increments

(f) Requirements are well understood

Match the following - 2

SRS

DDS

Waterfall Model

Incremental & Spiral

UAT

UML

(a) Linear + iterative

(b) Testing

(c) Class & Object diagrams

(d) Requirements phase

(e) Linear Model

(f) Design phase

Thank you