

CSE 470

Software Engineering

SDLC

Imran Zahid

Lecturer

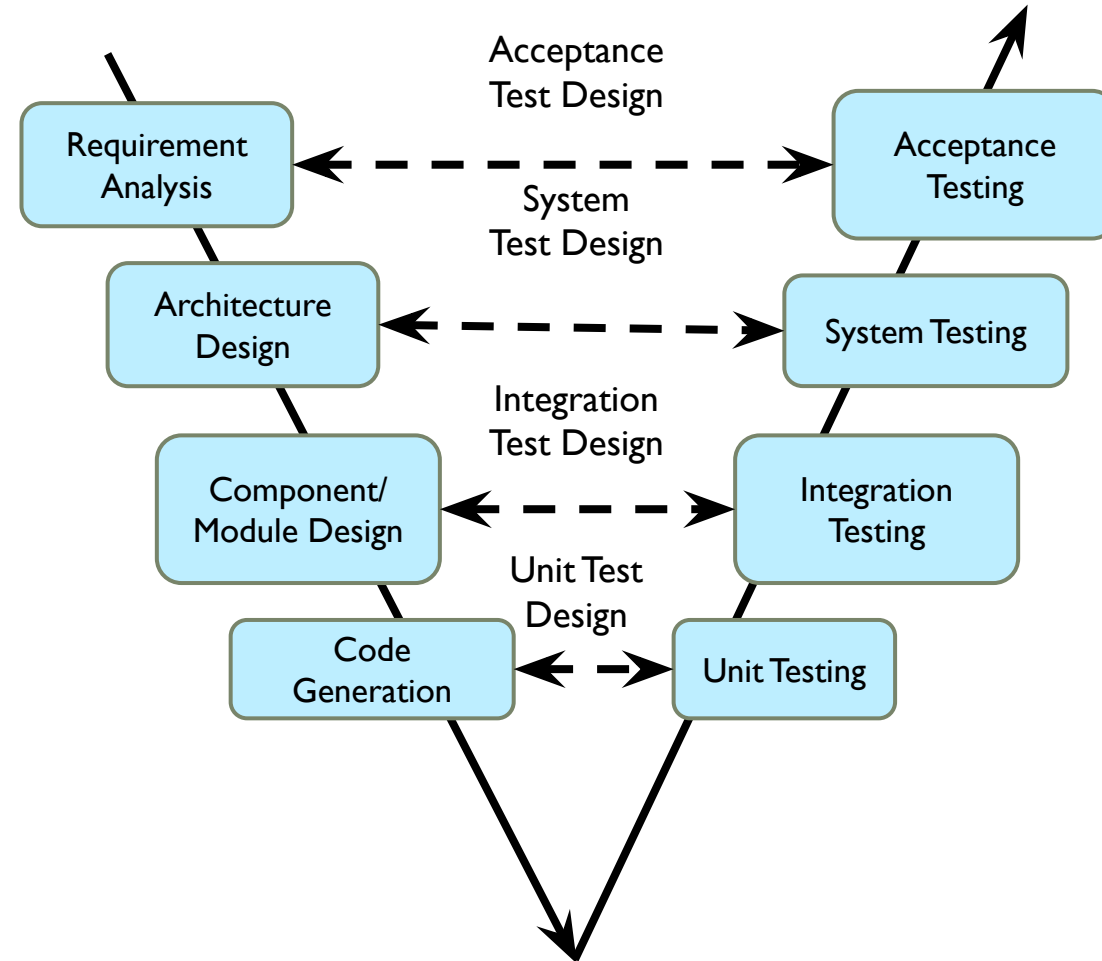
Computer Science and Engineering, BRAC University



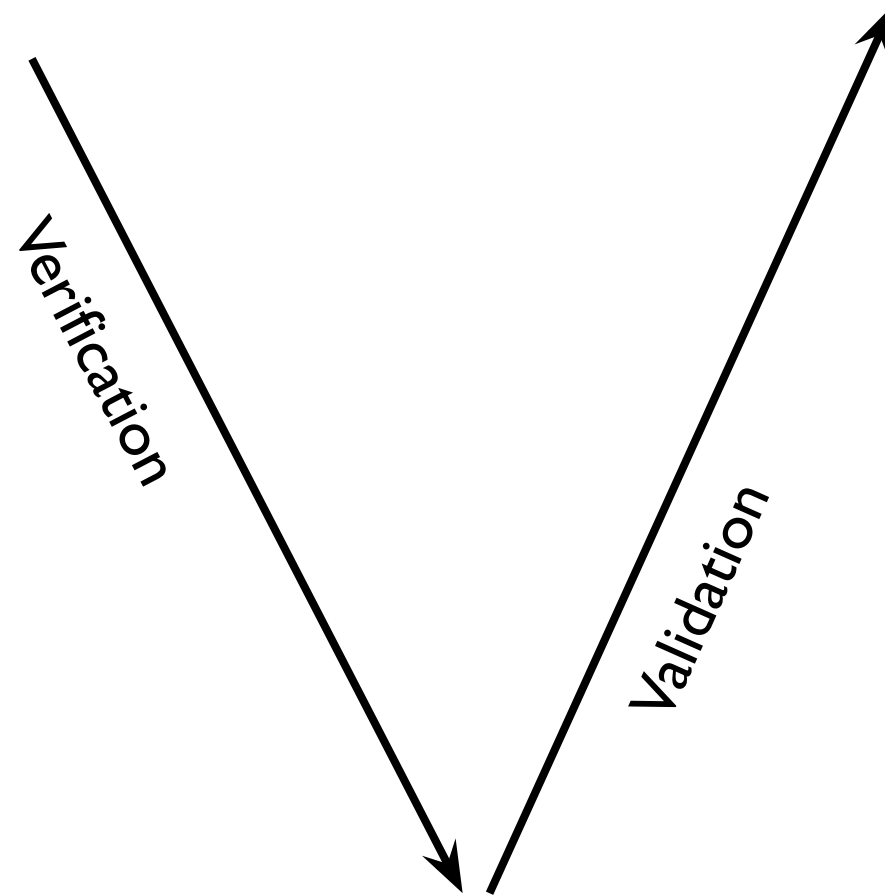
Recap



V Model



Verification and Validation



Pros

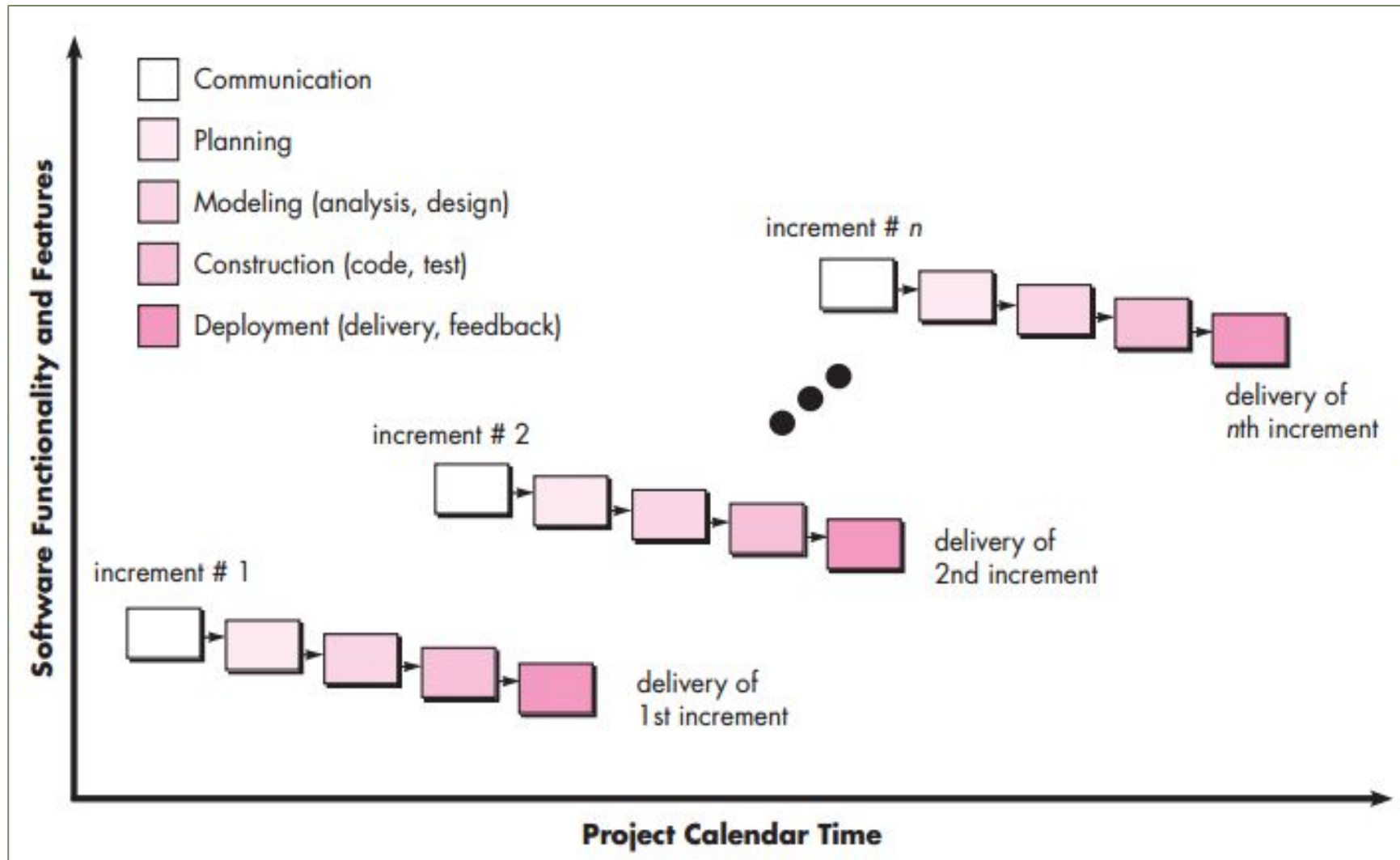
- Along with waterfall advantages, it emphasizes planning for verification and validation (V&V) of the product from the very beginning of requirement collection.
- Test activities planned before testing
- Saves time over waterfall, higher chance of success

Cons

- Changes are not welcomed
- Software developed at end of all phases, so no dummy prototypes can be found
- If any test fails, then test document and code both needs to be updated



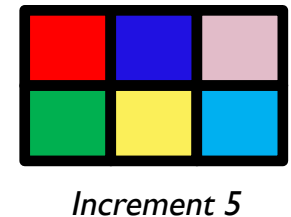
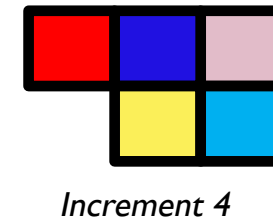
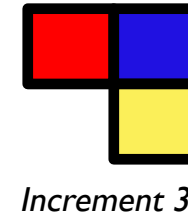
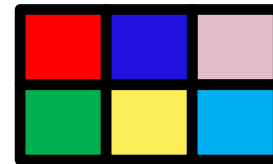
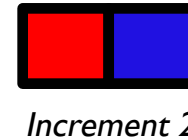
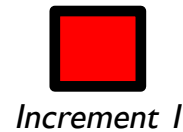
Incremental Process Model



Incremental Process Model

- For example, in first increment Login module is delivered.
- In second increment, another new module such as Navigation module is also added.
- In third, one more added.

That is, this model “adds onto” new section as increments.



Pros

- After each iteration, during testing, faulty elements of the software can be quickly identified.
- It is generally easier to test and debug because relatively smaller changes are made during each iteration.
- Customers can respond to features and review the product for any needed or useful changes.
- Initial product delivery is faster and costs less.
- Parallel stages such as requirement collection, planning etc. can take place.

Cons

- The resulting cost may exceed the cost of the organization.
- As additional functionality is added to the product, problems may arise related to system architecture which were not evident in earlier prototypes.

Incremental Process Model - When to Use?

- The software can be conceptualized as thin slices of the overall software.
- Customer wants prototype version of the software from the beginning of the project
- Increments needs to be prioritized by customers, so better chance of success
- Better for small or medium size projects



Iterative Process Model

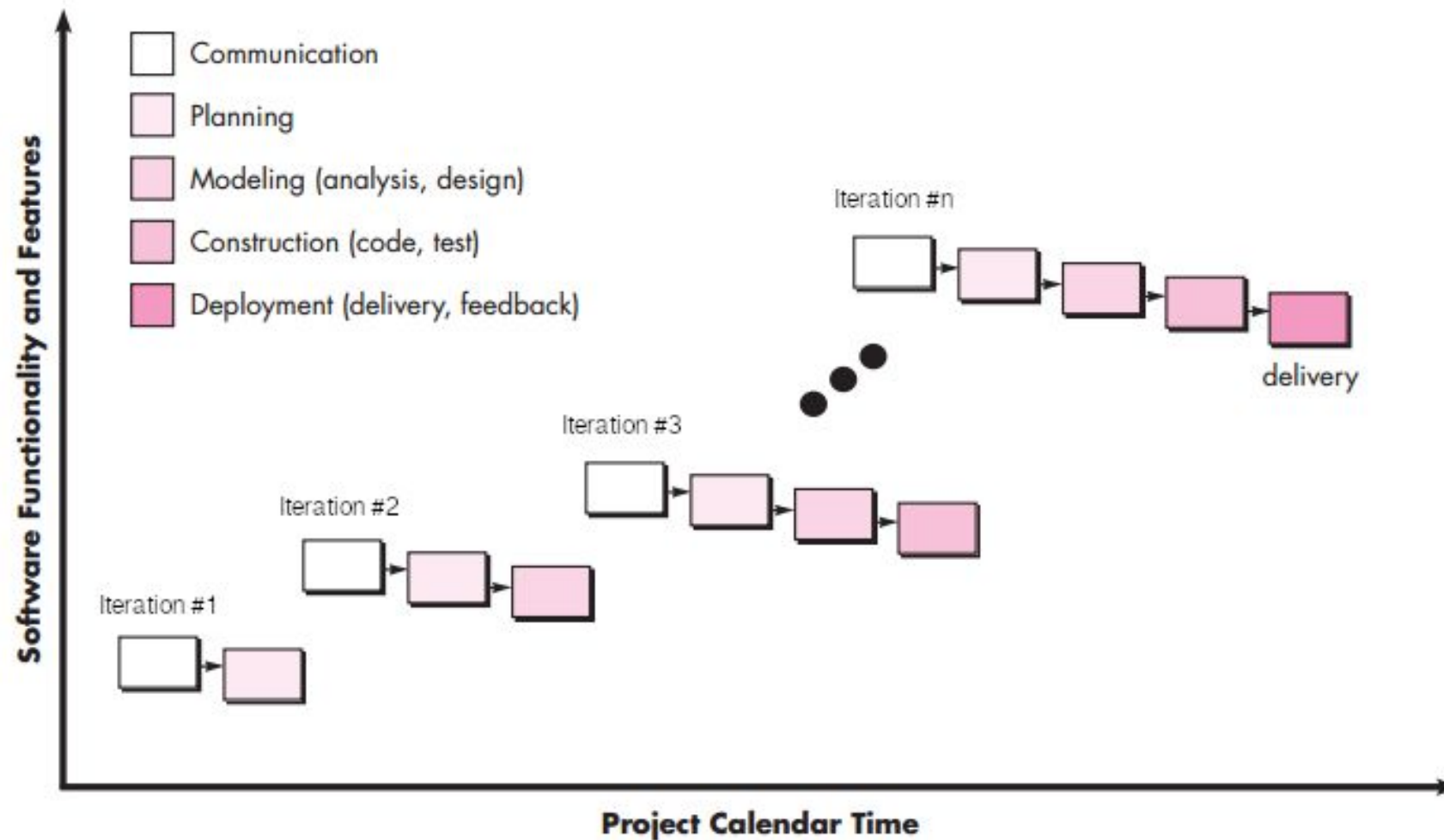


Iterative Process Model

- A software process model where the software will be delivered in iterations
- Customer is not sure about the requirements
- Even development team may not be sure about which technology or approaches will be used.
- There is no fixed limit of iterations, that is time is set aside.



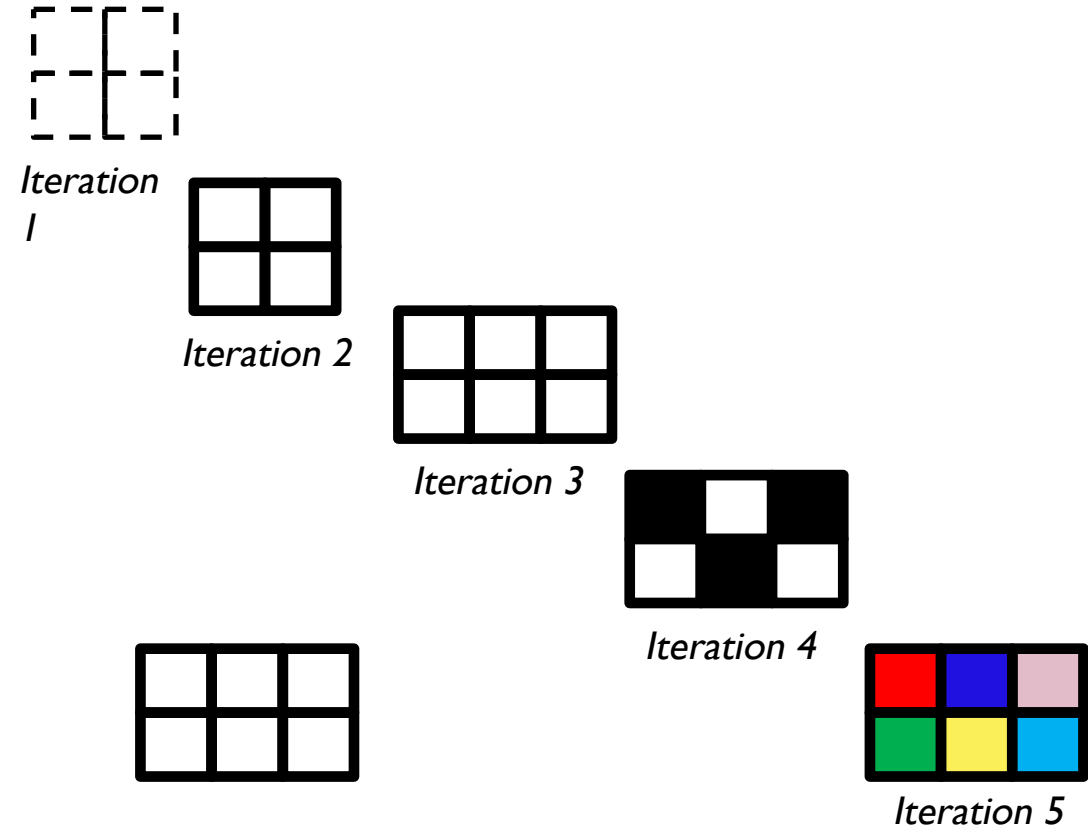
Iterative Process Model



Iterative Process Model

- For example, in first iteration Login module is delivered.
- In second iteration, Login module is updated again.
- In third iteration, Navigation module is added . And in fourth iteration some refinement is done.

That is, this model “changes or reworks” on same section in iterations until customer accepts it.



Pros

- Testing is facilitated by the modules being relatively small
- Most risks can be identified during iteration and higher risks can be dealt with as an early priority
- Inconsistencies among requirements, designs, and implementations are detected early.

Cons

- The output of each iteration is not necessarily a working/deployable product, it may be a partial product.
- It is hard to determine an end date for the project conclusion
- Each successive phase is rigid with no overlaps



Iterative Process Model - When to Use?

- Requirements are not fixed
- Technological tools or requirements are not identified yet.
- Instead of fixed time, quality of the features is refined with time
- Customer feedbacks with repetitive iterations increase the product quality
- Better for long-term and complex projects



Incremental vs Iterative

	Feature 1	Feature 2	Feature 3
UI layer			
App layer			
Middleware			
Database			

	Feature 1	Feature 2
UI layer	?	
App layer	?	
Middleware	✓	
Database	?	

	Feature 1	Feature 2	Feature 3
UI layer			
App layer			
Middleware			
Database			

	Feature 1	Feature 2
UI layer	??	
App layer	??	
Middleware	✓	
Database	✓	

	Feature 1	Feature 2	Feature 3
UI layer			
App layer			
Middleware			
Database			

	Feature 1	Feature 2
UI layer	✓	
App layer	✓	
Middleware	✓	
Database	✓	

Spiral Model



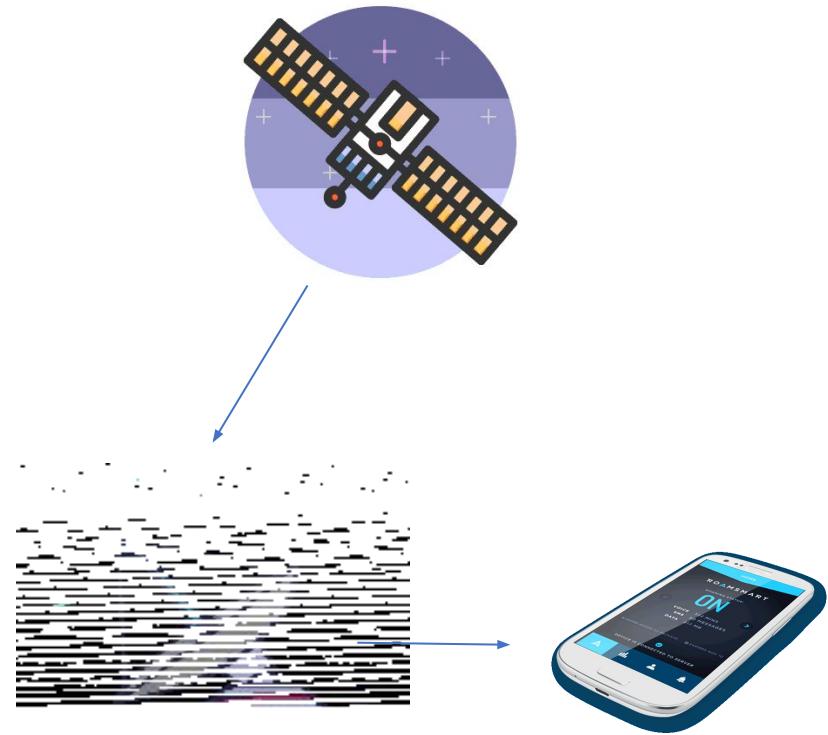
Spiral Model

- Rarely Used but an important model
- Its a model that works for projects with unlimited budget, time and projects that has huge risk factors.
- Example, making a heavy lift system for space shuttle and international space station.
- Another example can be about a company name Galaxy inc.



Spiral Model

- Suppose you wanted to send 6 dozen satellites to space and build a satellite based cellular system.
- So that remote places like even in Antarctica where you don't have any BTS (mobile tower) you can still communicate using your cell, that is, you are never out of network.



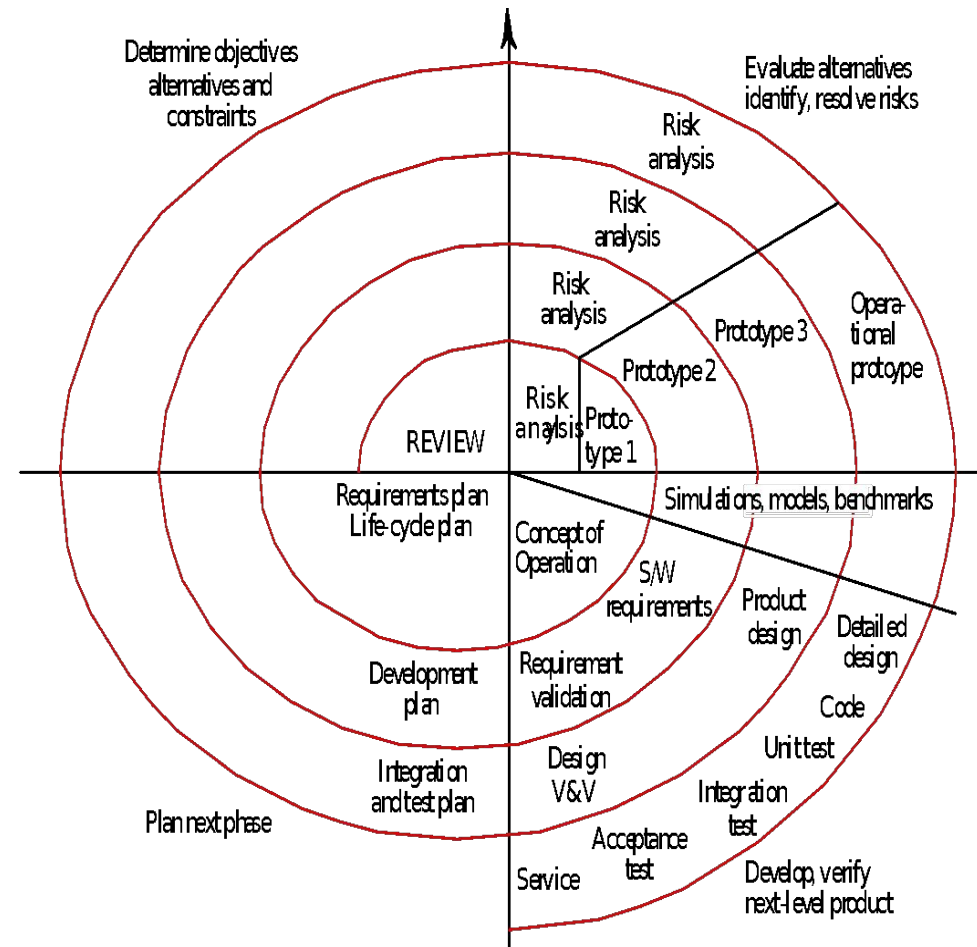
Spiral Model

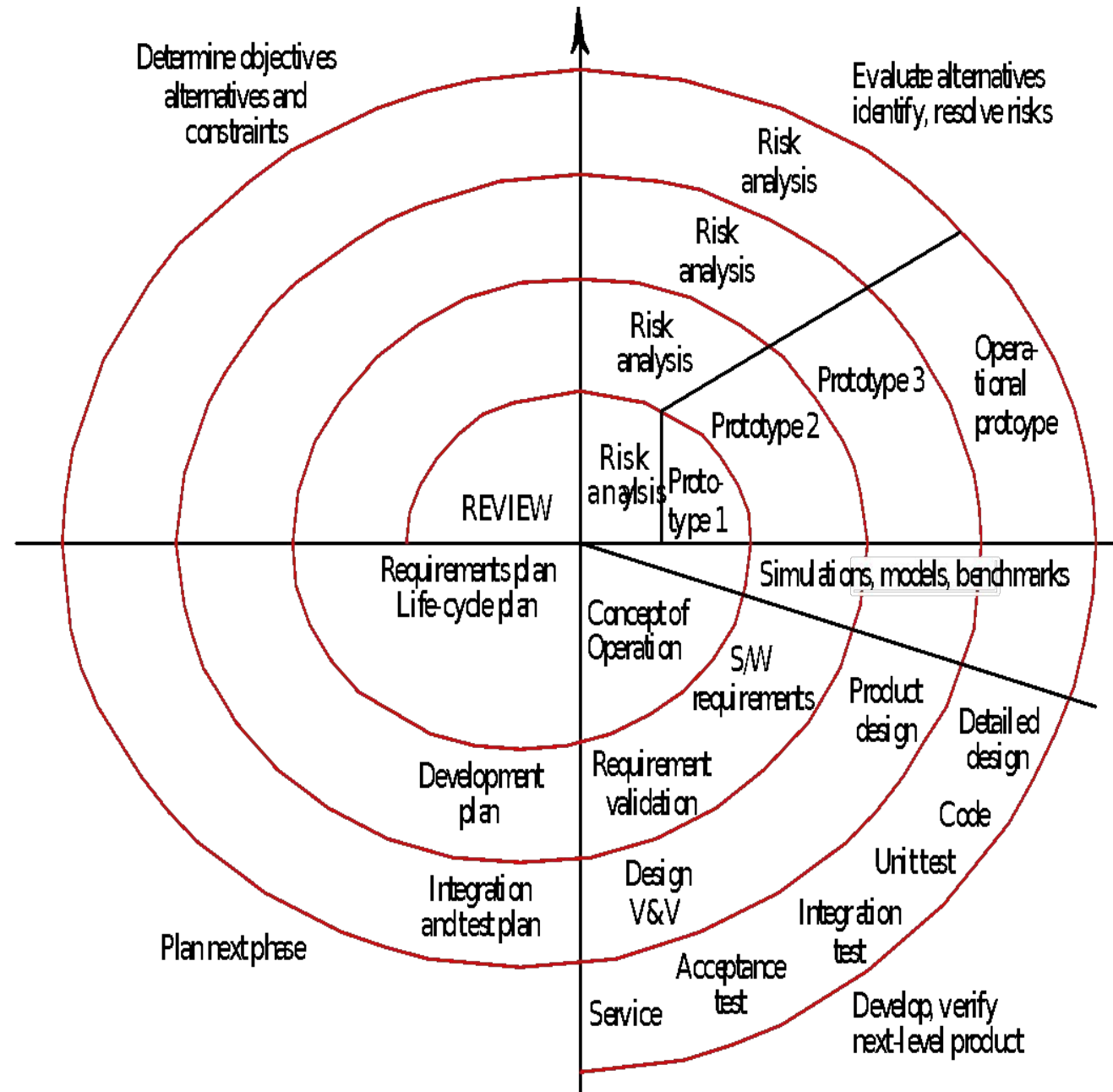
- As you can see for this project:
 - Risks were enormous
 - Needed a huge budget
 - No published materials or experienced workers
 - Risks will come and be identified once the project kicks off
 - Several million lines of code had to be written and you don't even have Stackoverflow.



Spiral Model

- The figure depicts spiral model.
- As you can see, it has spiral loops showing phase by phase development.
- You can see we perform risk analysis at every phase, and plan and keep building prototypes until we reach our goals.





Spiral Model - Formal Definition

- The spiral model is a risk-driven process model generator for software projects. Based on the unique risk patterns of a given project, the spiral model guides a team to adopt elements of one or more process models, such as incremental, waterfall, or evolutionary prototyping.
- This model was first described by Barry Boehm in his 1986 paper "A Spiral Model of Software Development and Enhancement".



Spiral Model - Phases

- Objective setting
 - Specific objectives for the phase are identified
- Risk assessment and reduction
 - Risks are assessed and activities put in place to reduce key risks
- Development and validation
 - A development model for the system is chosen which could be any of the generic models
- Planning
 - The project is reviewed and next phase of the spiral is planned



Spiral Model - Key Points

- The exact number of spirals needed to develop the product can be varied by the project manager depending upon the project risks.
- As the project manager dynamically determines the number of spirals, the project manager has an important role in developing a product using the spiral model.
- It is based on the idea of a spiral, with each iteration of the spiral representing a complete software development cycle, from requirements gathering and analysis to design, implementation, testing, and maintenance.



Spiral Model - Usage

- Spiral model has been very influential in helping people to think about iteration in software processes and introducing the risk-driven approach to development.
- In practice, however as mentioned, the model is rarely used as published for practical software development.



Spiral Model - When to Use?

- So if you are a project manager or lead developer when would you suggest spiral model? It is if you have:
 - Long term project commitment and budget
 - Users and developers unsure of the needs
 - Requirements are complex
 - New product line
 - Significant changes are expected (research and explanation)



Examples



Example Scenario

Being a project manager of a software company, you have got a project request for developing a coronavirus awareness app. The customers initially want the app to show testing info, take appointments and visualize affected area data. Currently, the software should support only Bangla language, however English language support can be added later if the app gets promising end user feedback.

In such a case, which would you apply – Waterfall, V, Incremental or Iterative?



Example Scenario

Being a project manager of a software company, you have got a project request for developing a coronavirus awareness app. The customers initially want the app to show testing info, take appointments and visualize affected area data. Currently, the software should support only Bangla language, however English language support **can be added later** if the app gets promising **end user** feedback.

In such a case, which would you apply – Waterfall, V, Incremental or Iterative?



Example Scenario

Being a project manager of a software company, you have got a project request for developing a coronavirus awareness app. The customers want the app to show testing info, take appointments, visualize affected area data and other features.

In such a case, which would you apply – Waterfall, V, Incremental or Iterative?



Example Scenario

Being a project manager of a software company, you have got a project request for developing a coronavirus awareness app. The customers want the app to show testing info, take appointments, visualize affected area data and **other features**.

In such a case, which would you apply – Waterfall, V, Incremental or **Iterative**?



Example Scenario

Being a project manager of a software company, you have got a project request for developing a coronavirus awareness app. The customers want the app to show testing info, take appointments, visualize affected area data.

In such a case, which would you apply – Waterfall, V, Incremental or Iterative?



Example Scenario

Being a project manager of a software company, you have got a project request for developing a **coronavirus** awareness app. The customers want the app to show testing info, take appointments, visualize affected area data.

In such a case, which would you apply – Waterfall, V, Incremental or Iterative?



Thank you

