



# **SNS COLLEGE OF ENGINEERING**

Kurumbapalayam (Po), Coimbatore – 641 107

**An Autonomous Institution**

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade  
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

## **DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY**

**COURSE NAME : 19CS302 AGILE SOFTWARE ENGINEERING**

**II YEAR /III SEMESTER**

**Unit 1- Introduction to Software Engineering**

**Topic 4: Evolutionary Process model and Specialized model**





# Brain Storming



1. How to software development differ from Iterative, Sequential and parallel?



# Evolutionary Process Model



Evolutionary process models are **iterative**. They are highly useful to a software engineer to develop more complex version soft wares. Three types in evolutionary process model:

1. The prototyping model
2. The spiral model
3. The concurrent development model



# The prototyping model



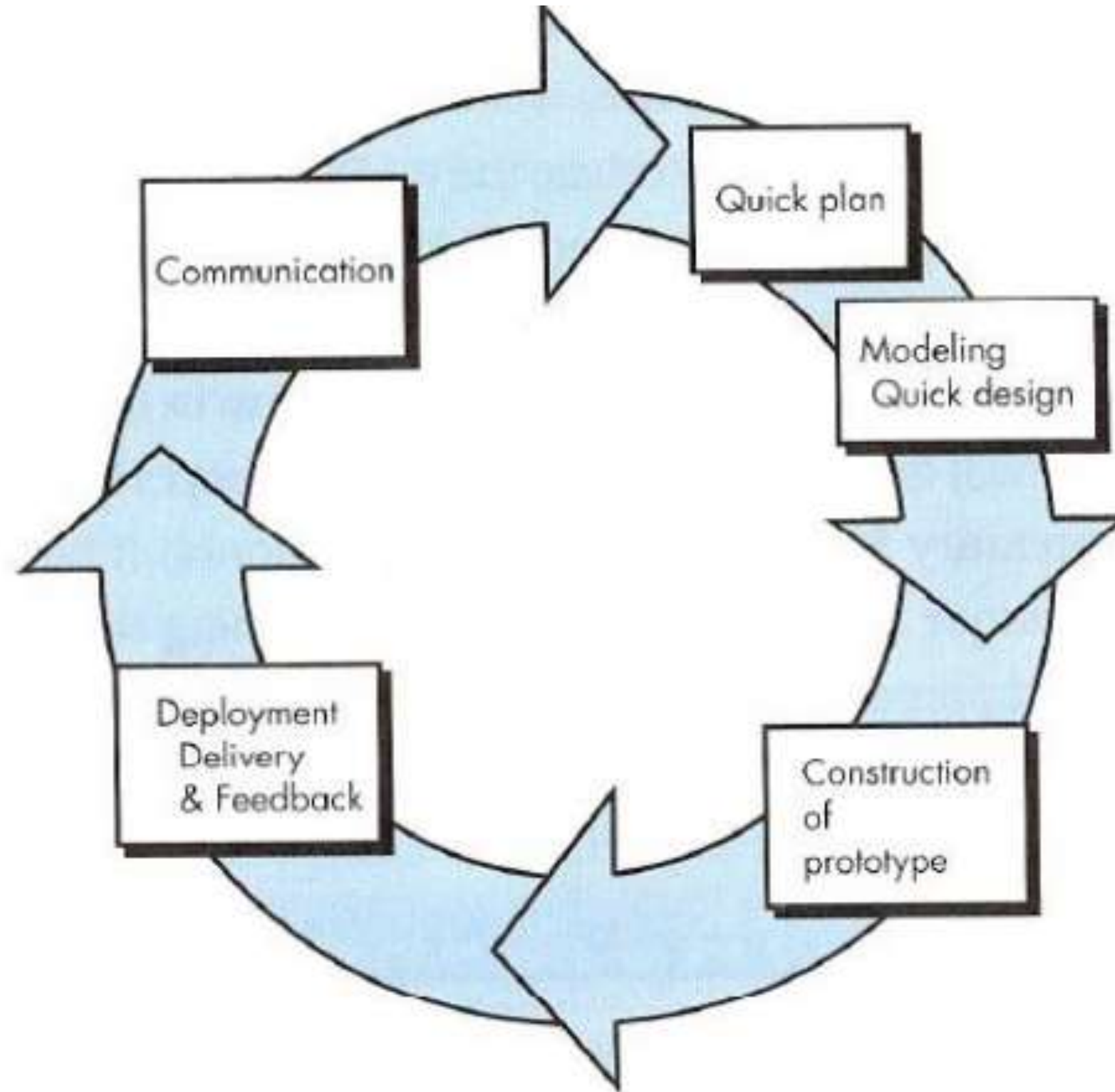
Consider the following two different situations:

1. Often, a customer defines a set of general objectives, but does not identify a detailed specifications like input, processing and output requirements. Its a problem in customer side.
2. Next is, problems in developer side; a developer may unsure about the efficiency of an algorithm; the adoptability of an operating system; problems in human-computer interaction etc.

In these non-stabilized cases, the prototyping model may offer a best solution.



# Conti...







# Problems in prototyping model



1. Customer gets disappointed and lodge more complaints about the product after seeing an initial version of working model, because they see a temporary patch. The customer sees what appears to be a working version of the software, unaware about the prototype. When a developer informed that the product must be rebuilt so that high quality can be maintained, the customer demands that “a few fixes” be applied to make the prototype a working product.
2. The developer makes implementation compromises in order to get a working prototype. An inappropriate program / inefficient algorithm / easy tool may be used to demonstrate the capability of working model. Later, the developer become comfortable with these choices and forget all the reasons why they were inappropriate. The less-than-ideal choice has now become an integral part of the system



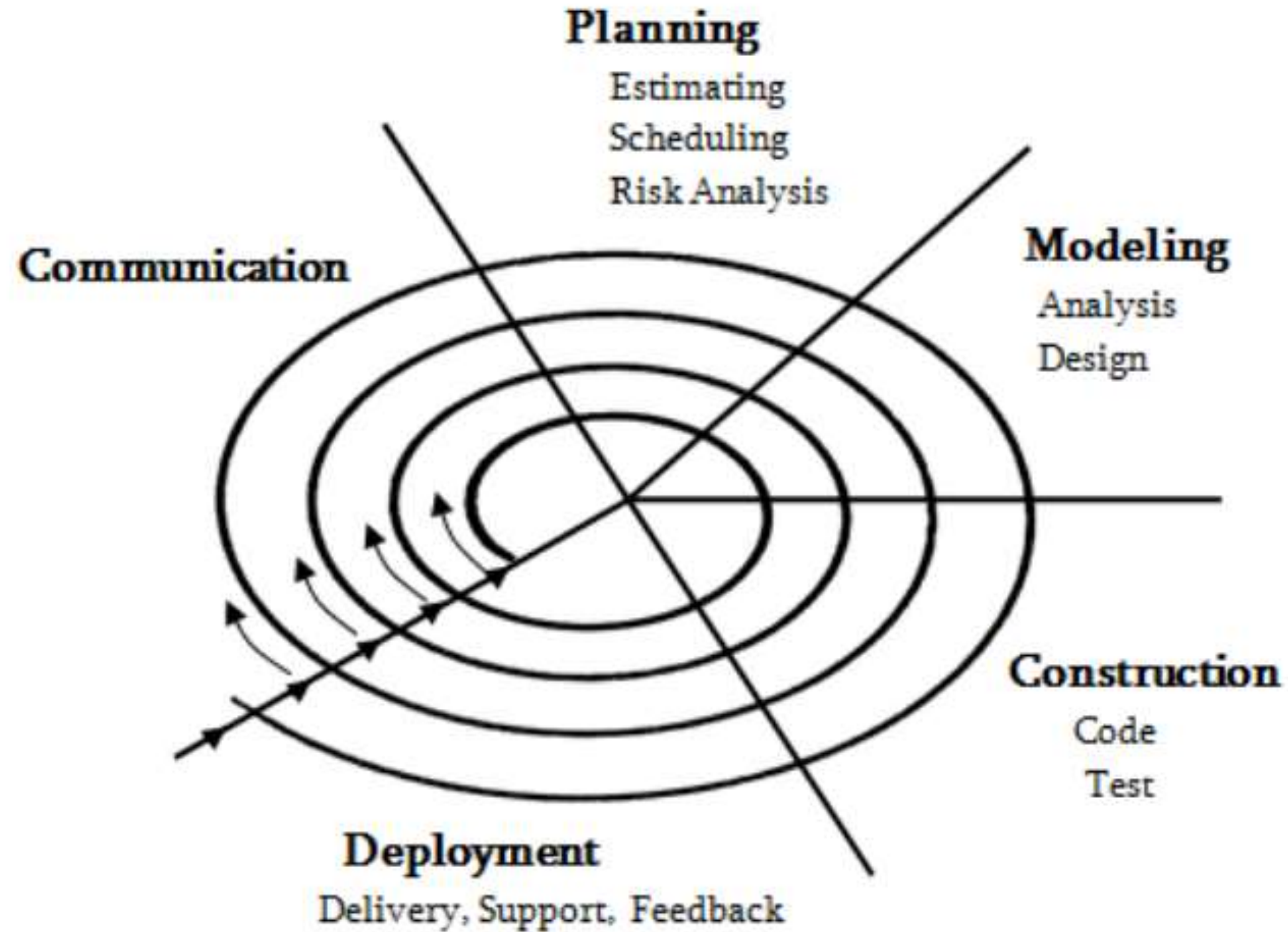
# The Spiral model



- It's a hybrid model that couples the iterative nature of prototyping model and controlled and systematic aspect of waterfall model.
- It provides potential for rapid development of more complex versions of software.
- It is a part evolutionary model.
- A spiral model is divided into a set of framework activities. Each framework activity represents one segment of the spiral path



# Spiral Model



**Figure : Spiral Model**

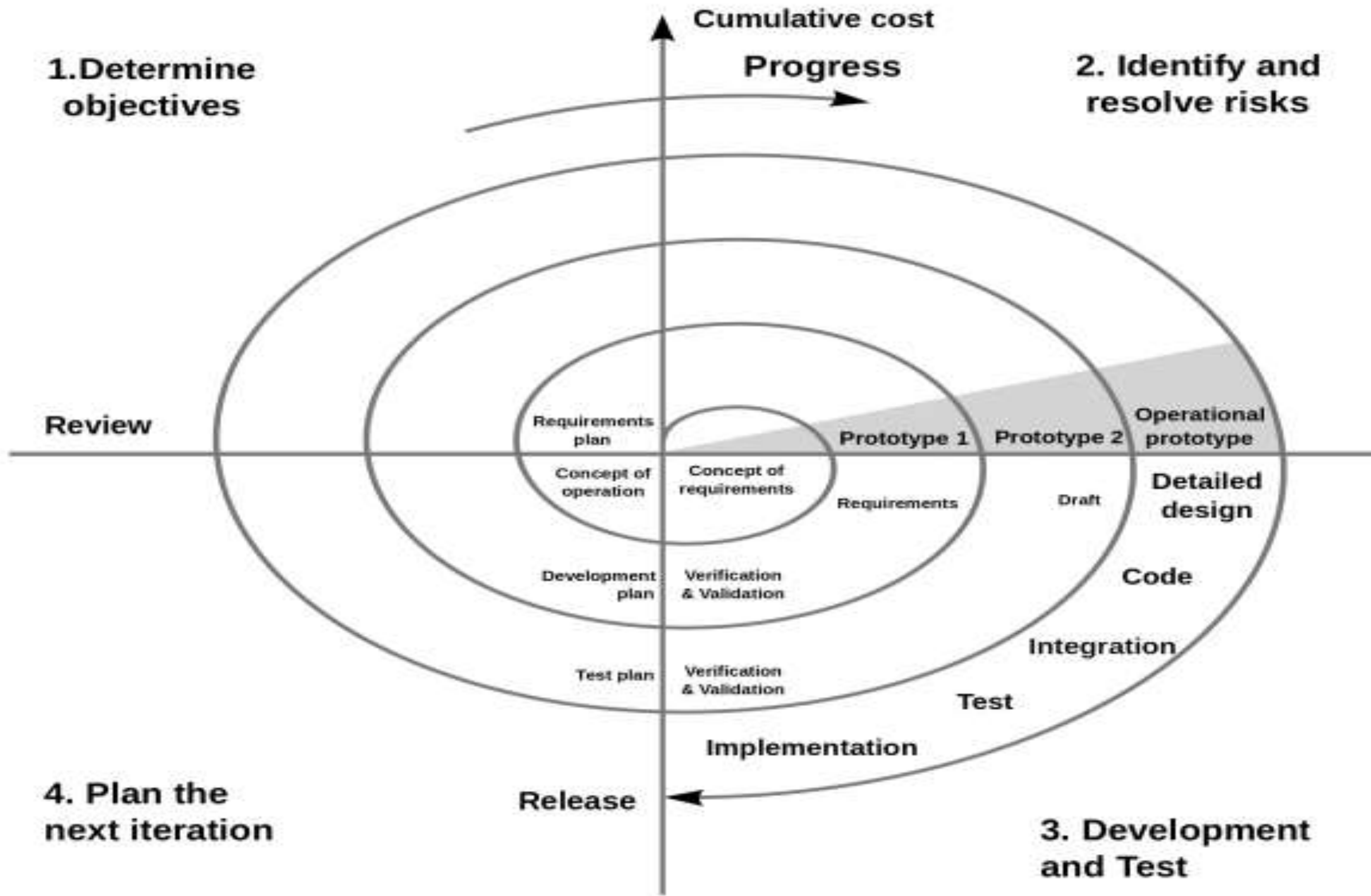




## Conti...



- As this process begins, the software team performs activities that are implied by a circuit around the spiral in a clockwise direction, beginning at the center.
- Anchor path milestones – a combination of work products and conditions are attained along the path of spiral.





# The positives and negatives of Spiral model



## Positives are:

- The spiral model is a realistic approach to the development of large complex systems. It helps developer to understand and react to the risks at each evolutionary levels.
- It sees prototype as one of risk reducing mechanism, and it enables the developer to use prototyping approach at every evolutionary levels.

## Problems are:

- It is very difficult to convince a customer (particularly in contract situations) that the model is controllable.
- It demands risk assessment expertise and relies on this expertise on success. If a major risk is uncovered, problems will occur,



# The concurrent development model

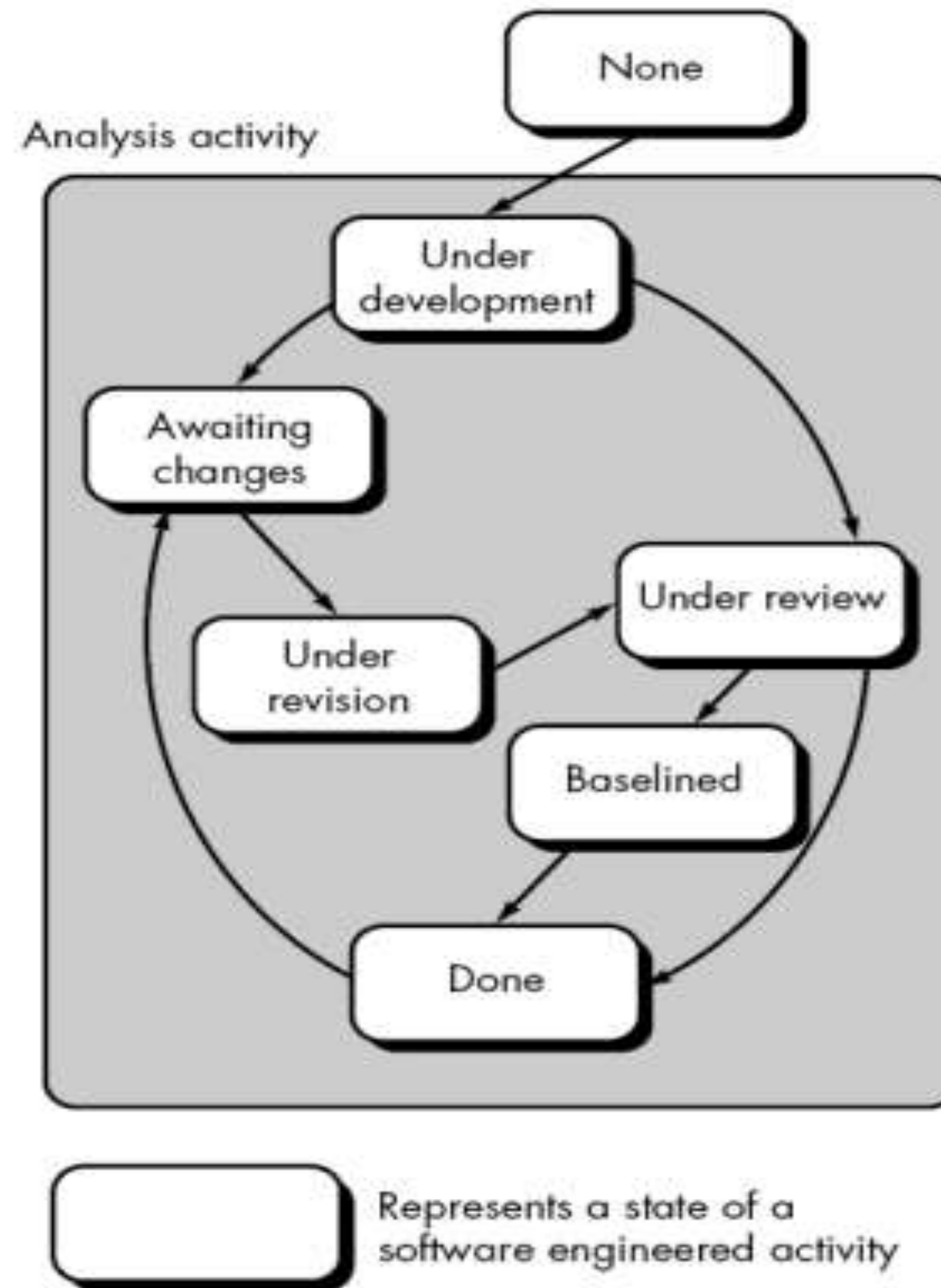


- Sometimes called concurrent engineering, can be represented schematically as a series of framework activities, software engineering actions, tasks and their associated states.
- The model defines a series of events that will trigger transitions from state to state for each of the software engineering activities, actions, or tasks.





## The concurrent development model

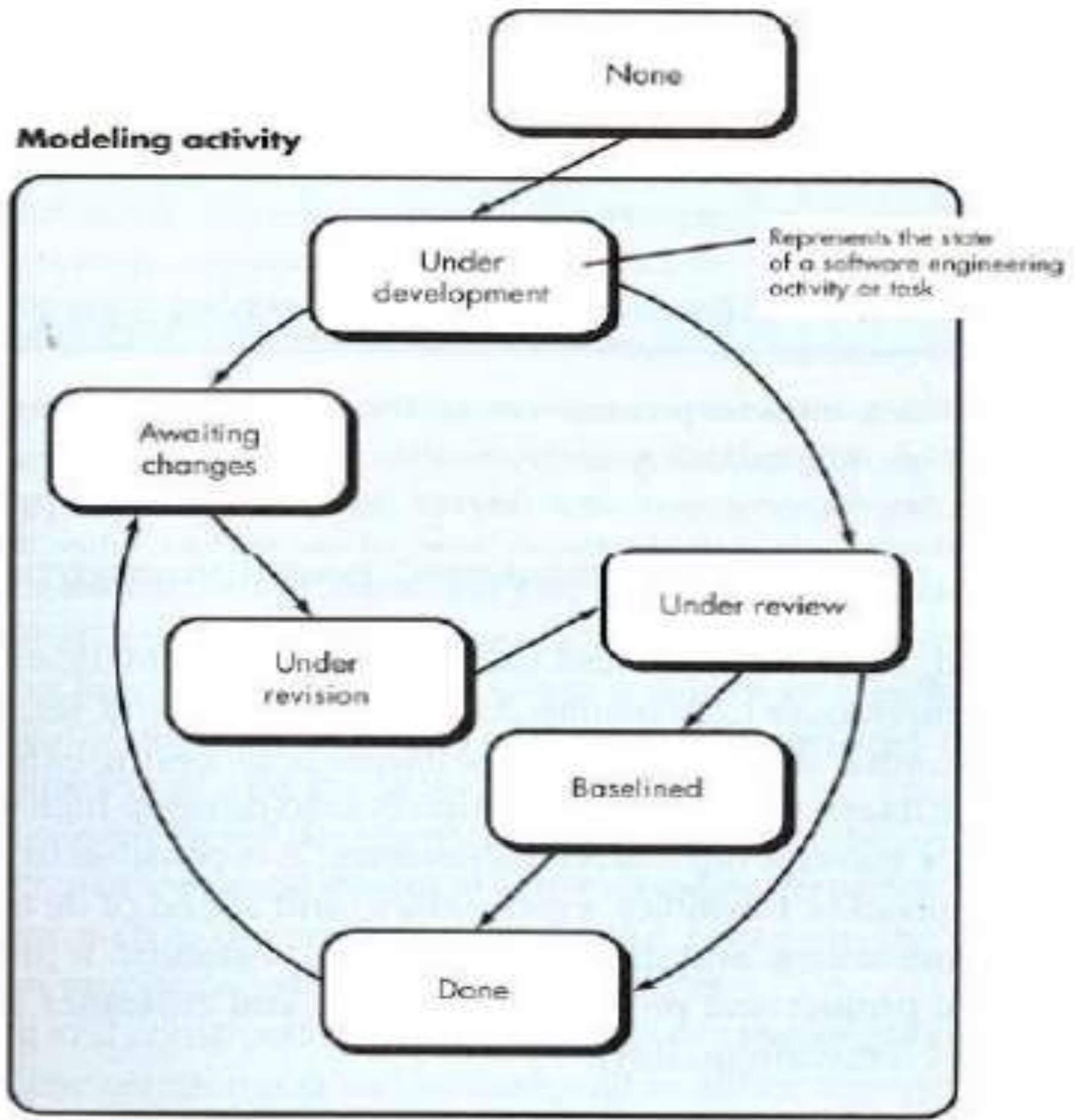




## Conti...



- Consider analysis activity – an action may go on anyone of the given states. Similarly other activities or tasks can be represented in an analogous (meaning: comparable in certain respects) manner.
- For your consideration, a modelling activity is given in next slide.





# Importance of concurrent development model



- The concurrent development model is applicable to all types of software development and provides an accurate picture of the current state of a project. It defines a network activity, not like other software engineering actions.
- Each activity, actions, tasks on the network exists simultaneously with other activities, actions, tasks.





# Specialized Process model



Special process models take many features from one or more conventional models. However these special models tend to be applied when a narrowly defined software engineering approach is chosen. Types in Specialized process models:

1. Component based development (Promotes reusable components)
2. The formal methods model (Mathematical formal methods are backbone here)
3. Aspect oriented software development (Uses crosscutting technology)



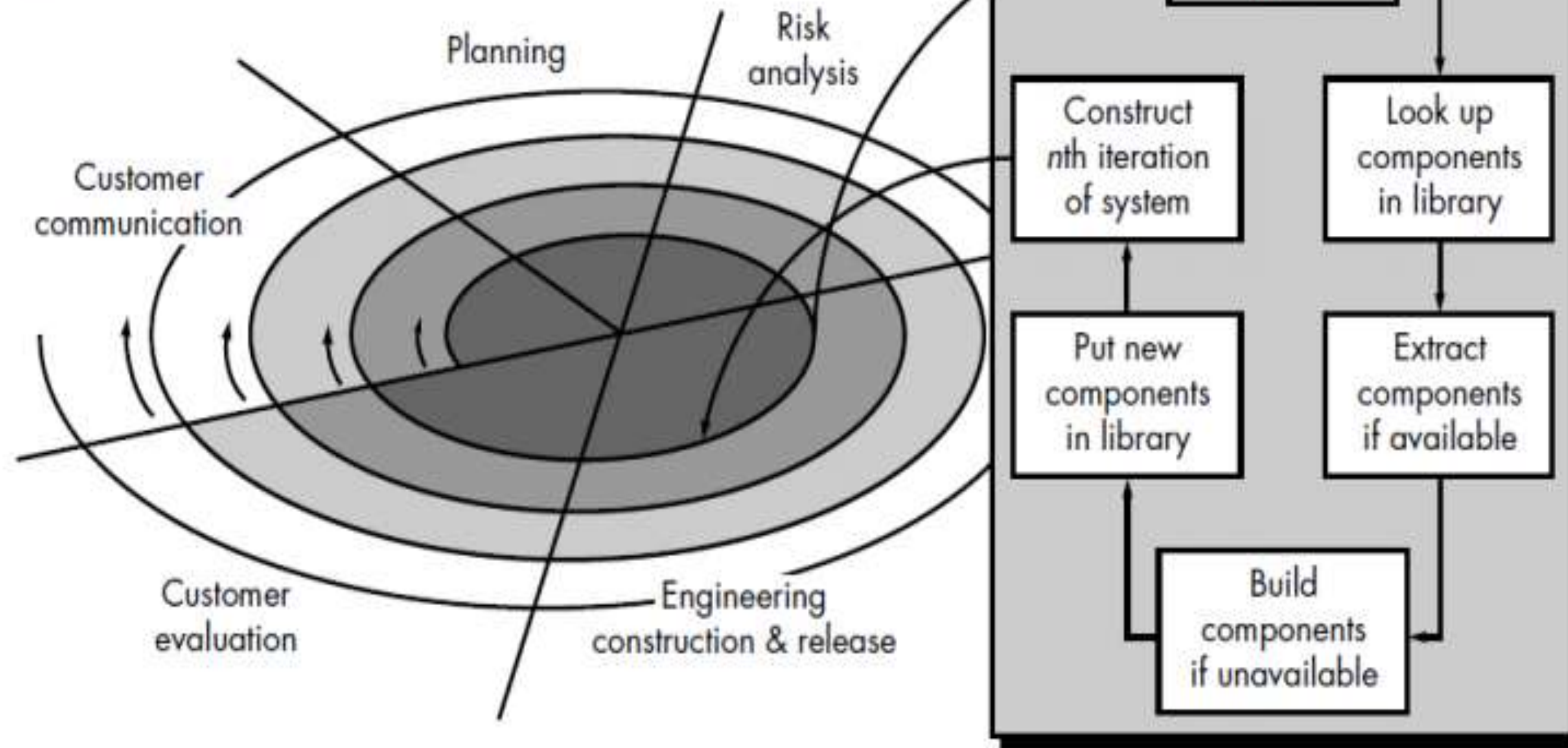
# Component based development



- The component based development model incorporates many of the characteristics of the spiral model.
- It is evolutionary in nature, demanding an iterative approach to the creation of software.
- However, the model focuses on prepackaged software components. It promotes software reusability.



Component-based development





# Conti...



- Modeling and construction activities begin with the identification of candidate components.
- Candidate components can be designed as either conventional software modules or object oriented packages.

## **Component based development has the following steps:**

1. Available component based products are researched and evaluated for the application domain.
2. Component integration issues are considered.
3. A software architecture is designed to accommodate the components.
4. Components are integrated into the architecture.
5. Comprehensive testing is conducted to ensure proper functionality.





# Problems in Component based development



**Component trustworthiness** - how can a component with no available source code be trusted?

**Component certification** - who will certify the quality of components?

**Requirements trade-offs** - how do we do trade-off analysis between the features of one component and another?



# The formal methods model



- The formal methods model encompasses a set of activities that leads to formal mathematical specification of software. Formal methods enable a software engineer to specify, develop and verify a computer system by applying rigorous mathematical notation.
- When mathematical methods are used during software development, they provide a mechanism for eliminating many of the problems that are difficult to overcome using other software engineering paradigms.
- Formal methods provide a basis for software verification and therefore enable software engineer to discover and correct errors that might otherwise go undetected.



# Problems in formal methods model



- The development of formal models is currently time-consuming and expensive.
- Because few developers have the necessary background knowledge to apply formal methods, extensive training is required to others.
- It is difficult to use this model as a communication mechanism for technically unsophisticated people.

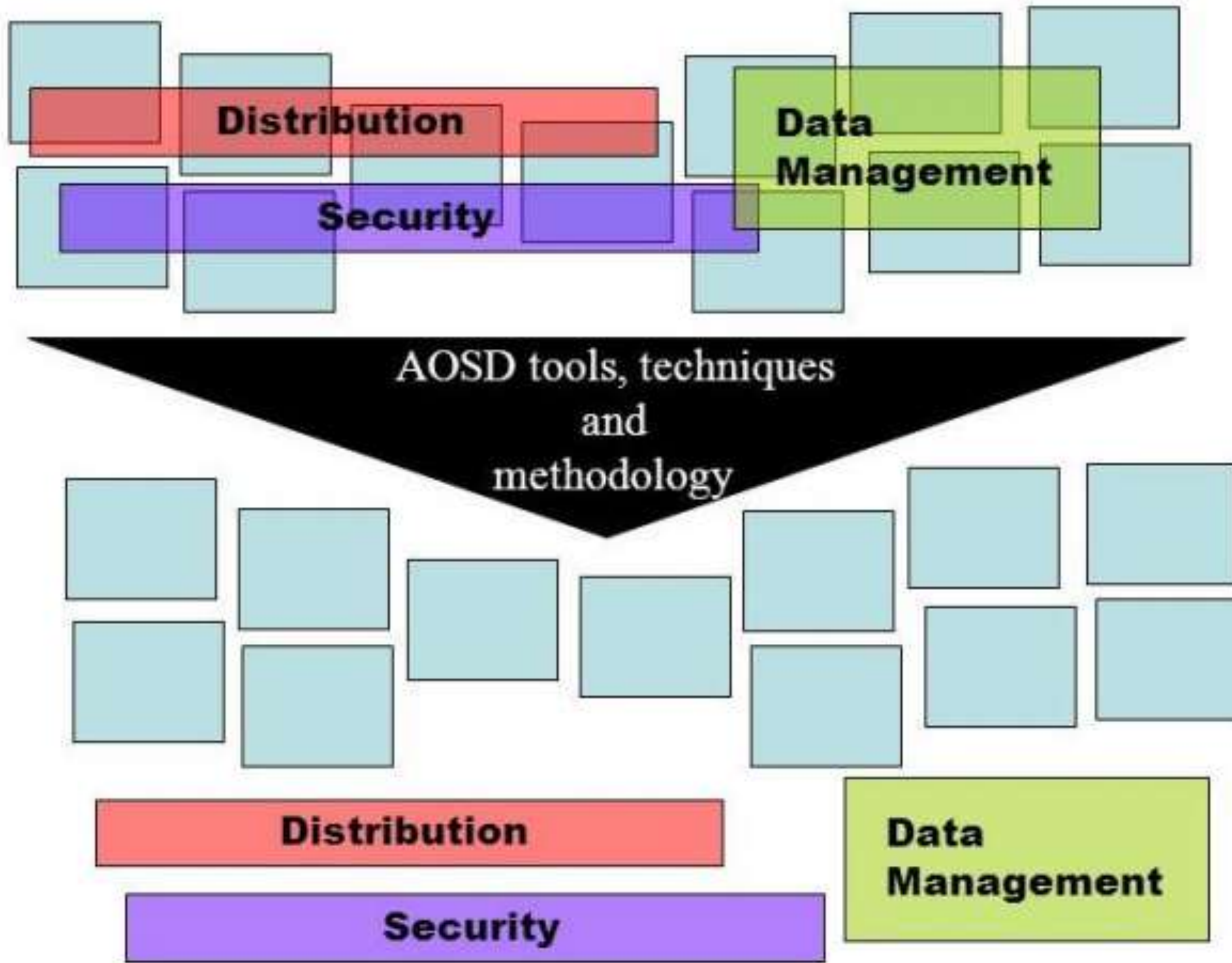


# Aspect oriented Software Development



- Aspect Oriented Software Development (AOSD) often referred to as aspect-oriented programming (AOP), a relatively new paradigm that provides process and methodology for defining, specifying designing and constructing aspects.
- It addresses limitations inherent in other approaches, including object-oriented programming.
- AOSD aims to address crosscutting concerns by providing means for systematic identification, separation, representation and composition.
- This results in better support for modularization hence reducing development, maintenance and evolution costs.







# Assessment 1



1. What is Evolutionary process model?

Ans : \_\_\_\_\_

2. Write about Specialized software model?

Ans : \_\_\_\_\_





# References



1. Roger S. Pressman, Software engineering- A practitioner's Approach, 10th Edition, McGraw-Hill, 2017.
2. Ken Schawber, Mike "Agile Software Development with Scrum" Pearson Education, 2<sup>nd</sup> Edition, 2015.

**Thank You**