CSE4708: Software Project Management

Unit II : Project Evaluation & Estimation

Topic: Spiral Model

Name: Manka Sharma

Delivered on: 20th August 2020

Process Models

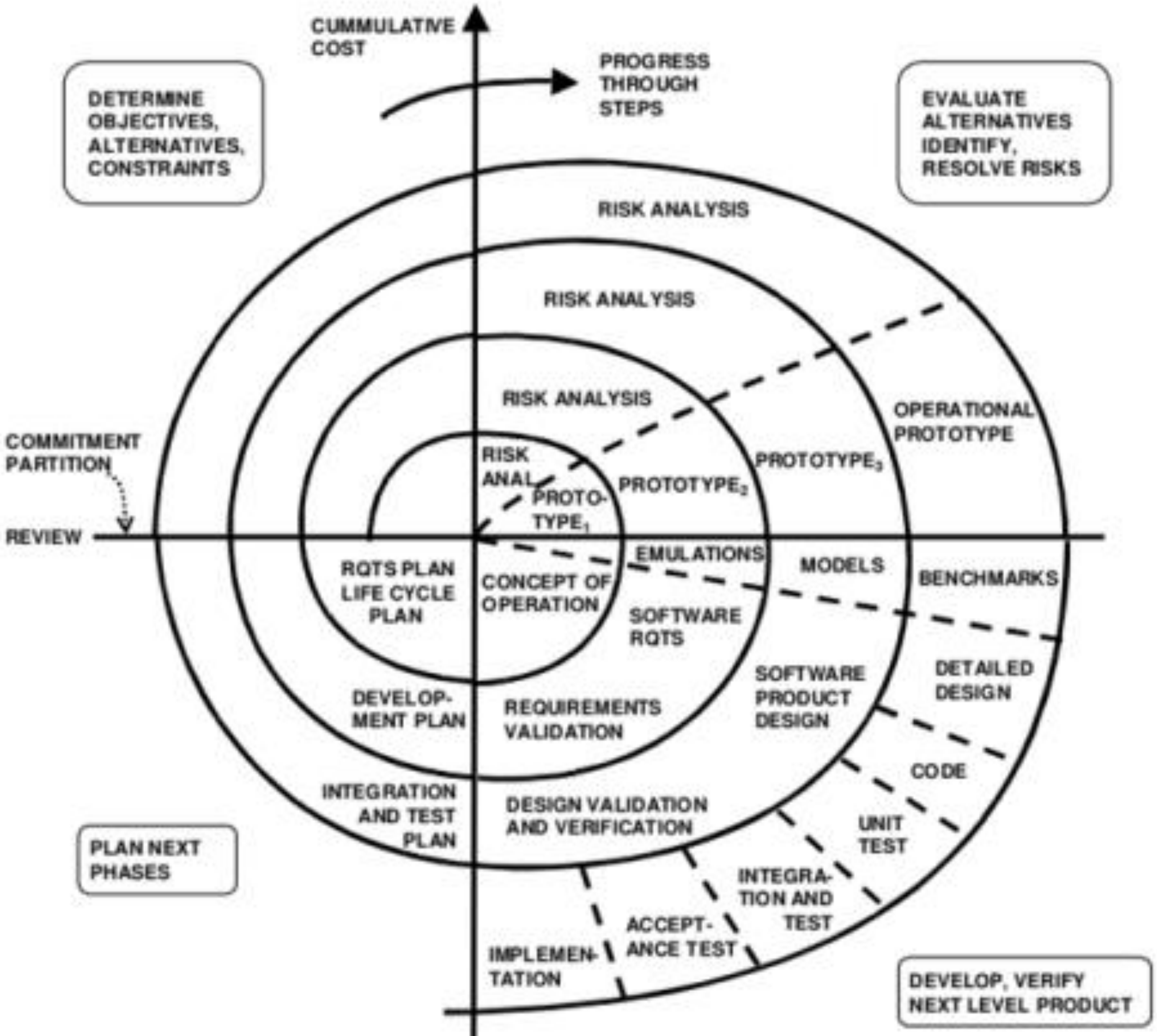
Spiral Model

- Developed by Barry Boehm in 1988.
- Combines the features of the prototyping model and the waterfall model.
- It is favored for large, expensive, and complicated projects.
- This model uses many of the same phases as the waterfall model, in essentially the same order, separated by planning, risk assessment, and the building of prototypes and simulations.
- It is used in the large applications and systems which built-in small phases or segments.

Spiral Model

- The spiral development model is a risk-driven process model generator that is used to guide multi-stakeholder concurrent engineering of software-intensive systems.
- It has two main distinguishing features.
  - One is a cyclic approach for incrementally growing a system's degree of definition and implementation while decreasing its degree of risk.
  - The other is a set of anchor point milestones for ensuring stakeholder commitment to feasible and mutually satisfactory system solutions

Spiral Model



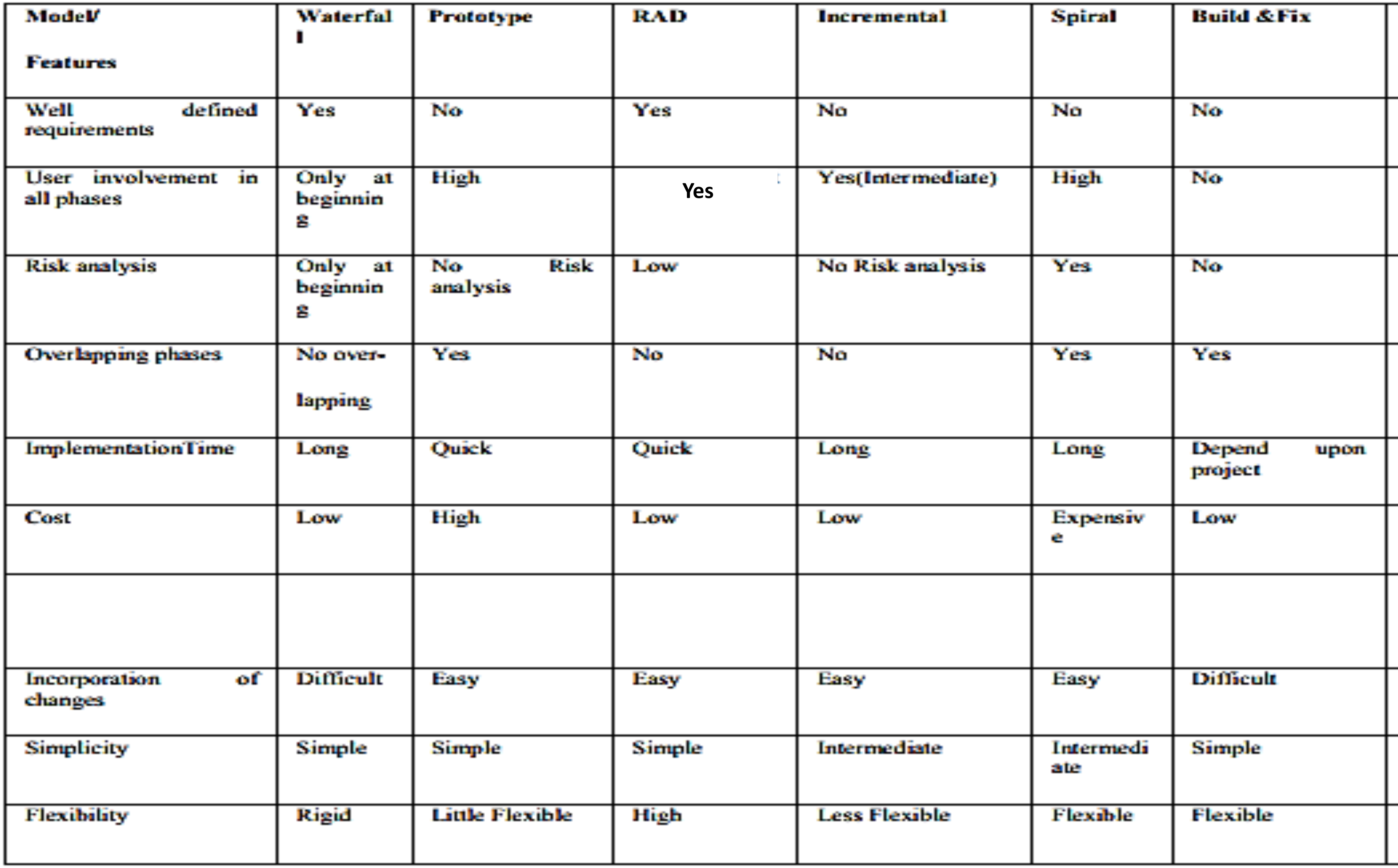
Advantages

- Estimates i.e. budget, schedule, cost etc. become more realistic as work progressed because important issues are discovered earlier.
- Early involvement of developers.
- Manages risks and develops the system into phases.

Disadvantages

- High cost and time to reach the final product.
- Needs special skills to evaluate the risks and assumptions.
- Highly customized limiting re-usability

Comparison of various SDLC Models



Model/ Features	Waterfall	Prototype	RAD	Incremental	Spiral	Build &Fix
Well defined requirements	Yes	No	Yes	No	No	No
User involvement in all phases	Only at beginnings	High	Yes	Yes(Intermediate)	High	No
Risk analysis	Only at beginnings	No Risk analysis	Low	No Risk analysis	Yes	No
Overlapping phases	No overlapping	Yes	No	No	Yes	Yes
ImplementationTime	Long	Quick	Quick	Long	Long	Depend upon project
Cost	Low	High	Low	Low	Expensive	Low
Incorporation of changes	Difficult	Easy	Easy	Easy	Easy	Difficult
Simplicity	Simple	Simple	Simple	Intermediate	Intermediate	Simple
Flexibility	Rigid	Little Flexible	High	Less Flexible	Flexible	Flexible

References

- Bob Hughes and Mike Cotterell, “Software Project Management ”, Tata McGraw Hill, 4th edition, 2006 .
- Roger S. Pressman, “Software Engineering – A practitioners Approach”, 5th edition.
- Software Project Management, Tutorialspoint. https://www.tutorialspoint.com/software_engineering/software_project_management.htm (accessed on 18th July 2020).