



Engineering Primer

Agenda

 Day 1. Introduction to Software Dev Methodologies

 Day 2. Introduction to Technologies

 Day 3. Coding Best Practices - Overview

 Day 4. Introduction to Database

 Day 5. Cloud and Logical Reasoning

 Day 6. And much more...



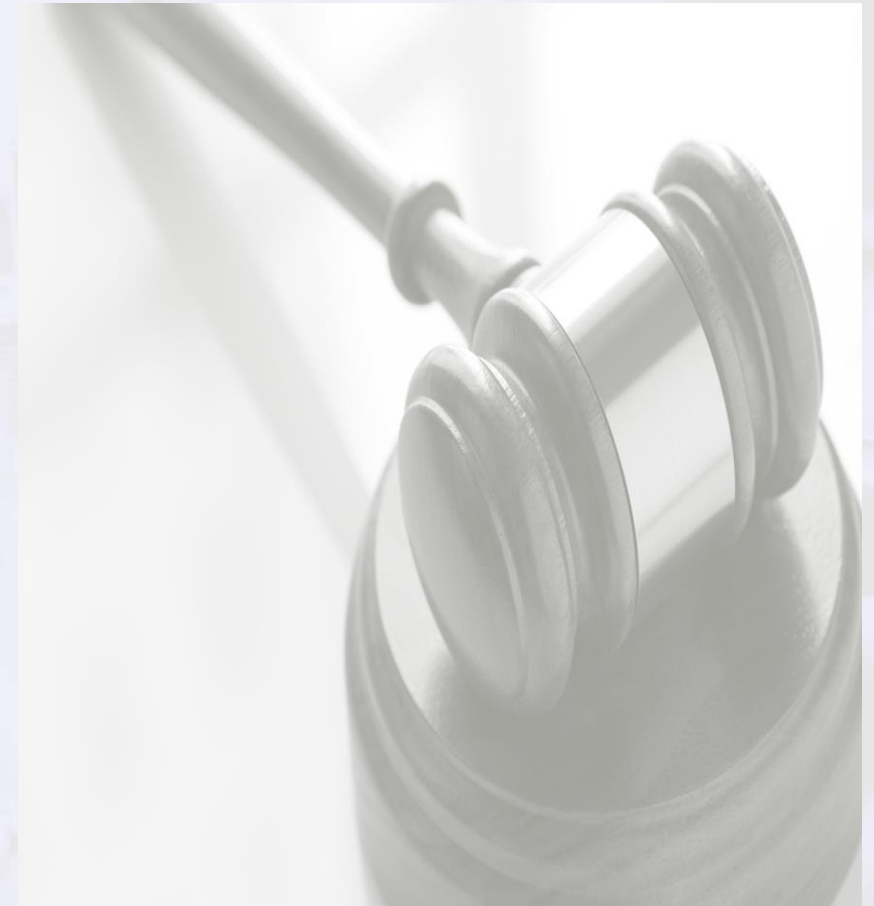
Know your Trainer..

- Neeraj Gupta, a technologist by heart, having more than 22+ years of experience of developing, designing, managing, delivering and coaching the software solutions & tools.
- A Microsoft Alumni, worked for various large and mid size organizations e.g., Citi Bank, Royal Bank of Canada, IRIS Software, Xceedance consulting etc.
- Delivered 60+ software products / solutions during my work tenure and still counting.
- Carrying PMP from Project Management Institute, USA and data analytics degree from Grate Lakes Institute of Management.
- Trainer by passion..



Some Ground Rules..

- Please put your phones on silent or better switch off, if you can..
- Request you to make silence in classroom.
- We will have two 15 minutes breaks and one ½ hour for lunch etc.
- Raise your hand if you have any query..
- Any question, switch on video before talking please.

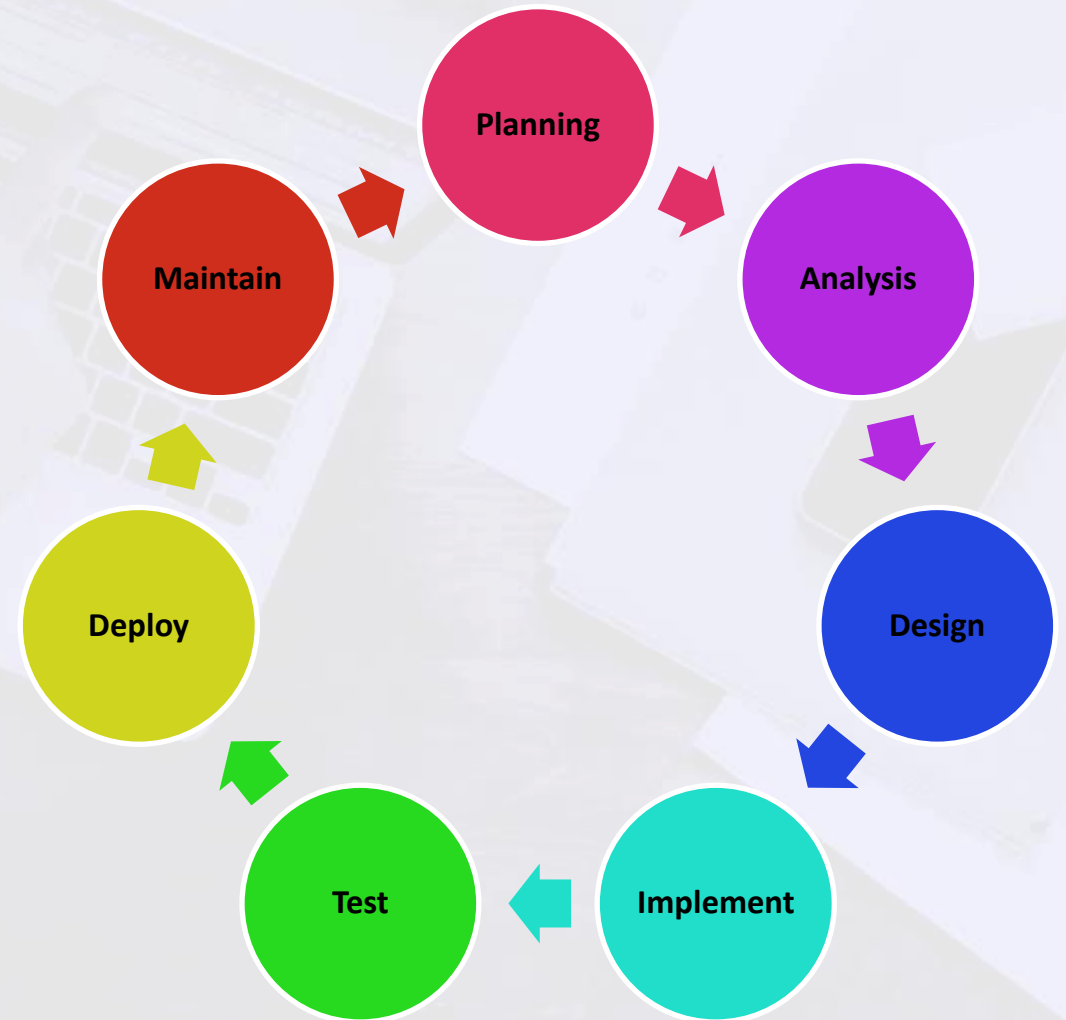


Software Development Life Cycle

FRAMEWORK(S) FOR SOFTWARE DEVELOPMENT

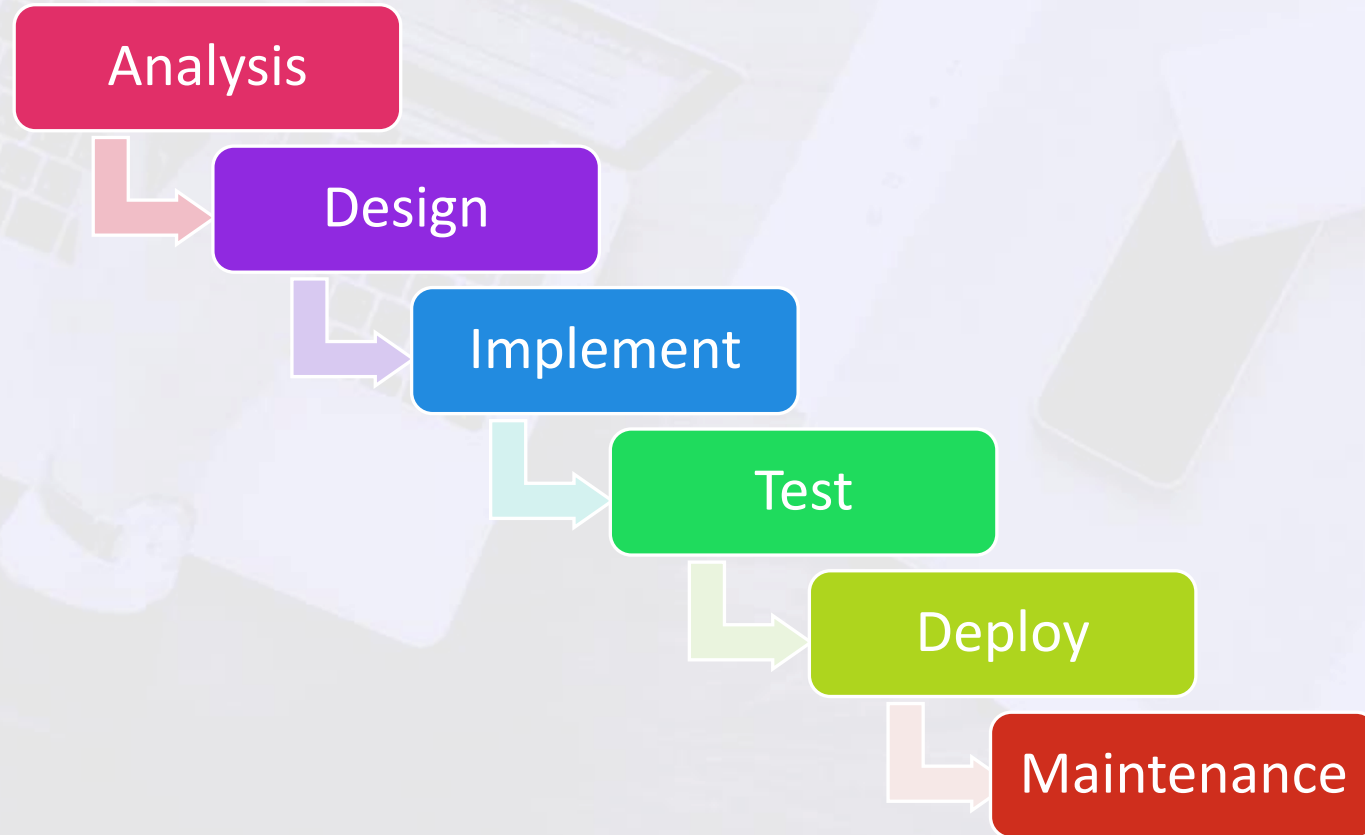
What is SDLC?

- A framework that describes the activities performed at each stage of a software development project.
- Helps in producing high quality products within defined time and budget.
- Each stage in SDLC have an entry & exit points.
- Models: Waterfall, V-Shaped, Spiral, Agile etc.



Waterfall.. Classic Approach

- Classic approach for developing software.
- Step wise moving to next stage from current stage.
- The output of one stage is input for next stage.



Waterfall.. Benefits

- Easy to understand, easy to use
- Provides structure to inexperienced staff
- Milestones are well understood
- Sets requirements stability
- Good for management control (plan, staff, track)
- Works well when quality is more important than cost or schedule

Waterfall.. Drawbacks

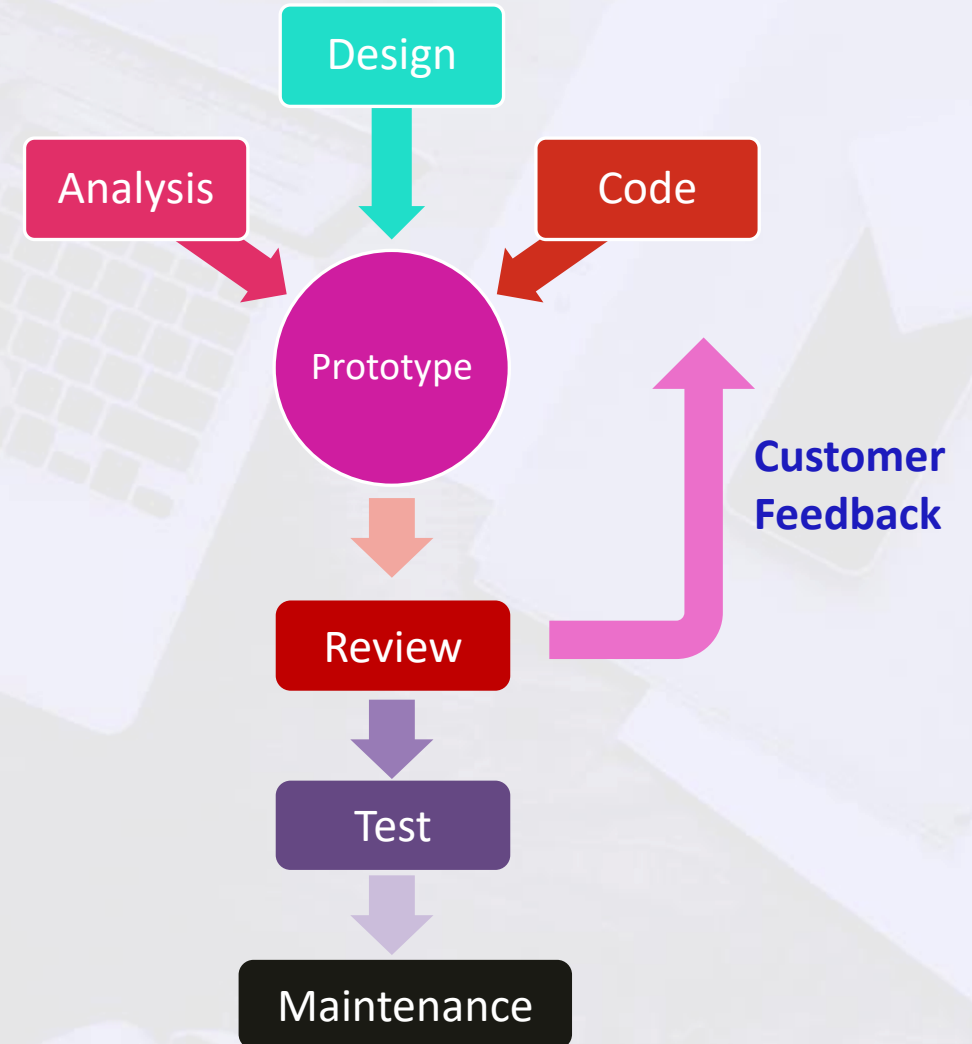
- All requirements must be known upfront
- Deliverables created for each phase are considered frozen – inhibits flexibility
- Can give a false impression of progress
- Does not reflect problem-solving nature of software development – iterations of phases
- Integration is one big bang at the end
- Little opportunity for customer to preview the system (until it may be too late)

Waterfall.. When to go with?

- 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.. Review / feedback

- Developers build a **prototype** during the **requirements phase**
- 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.. Benefits

- 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**
- Interaction with the prototype **stimulates awareness of additional needed functionality**

Prototyping model.. Drawbacks

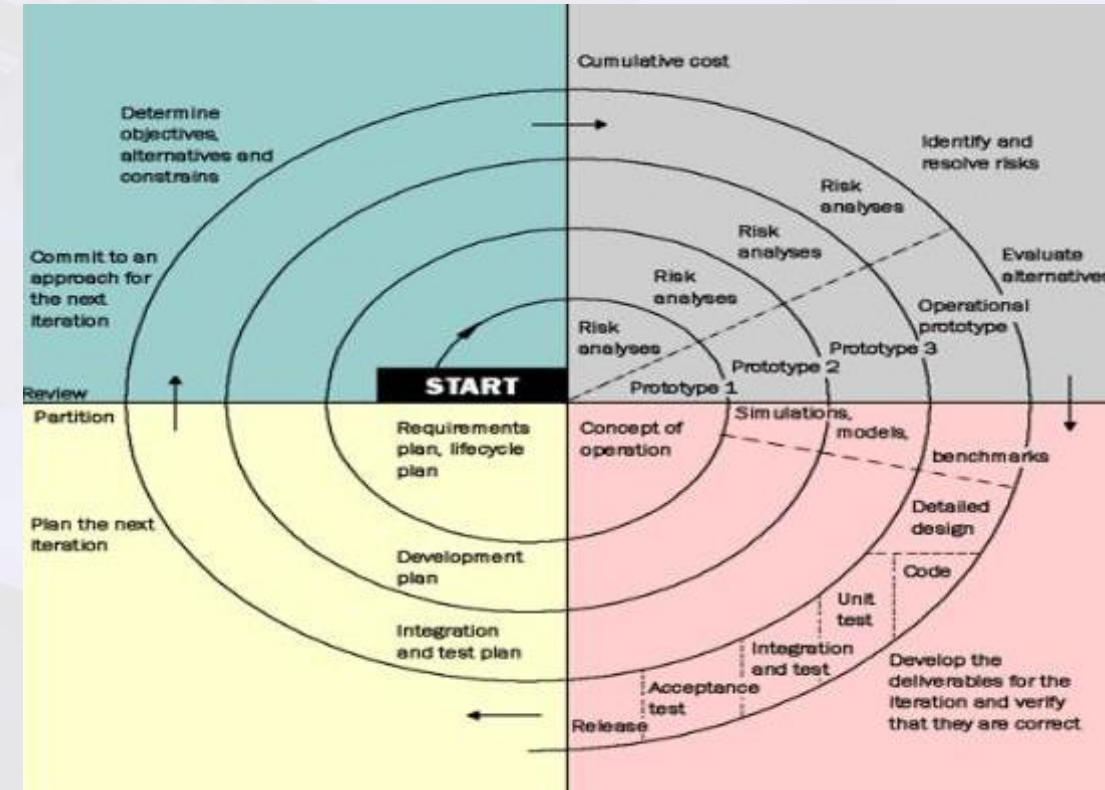
- Tendency to abandon structured program development for “code-and-fix” development
- Bad reputation for “quick-and-dirty” methods
- Overall maintainability may be overlooked
- The customer may want the prototype delivered.
- Process may continue forever (scope creep)

Prototyping.. When to go with?

- Requirements are unstable or must be clarified
- As the requirements clarification stage of a waterfall model
- Develop user interfaces
- Short-lived demonstrations
- New, original development
- With the analysis and design portions of object-oriented development.

Spiral.. Cyclical Iteration

- Evaluation of production in multiple cycles.
- Each cycle involves same sequence of steps as the waterfall process model.
- Adds risk analysis & resolution for stable outcome.



Spiral.. Quadrants

Determine Objectives, alternatives and constraints

- **Objectives:** functionality, performance, hardware/software interface, critical success factors, etc.
- **Alternatives:** build, reuse, buy, sub-contract, etc.
- **Constraints:** cost, schedule, interface, etc.

Evaluate alternatives, identify and resolve risks

- Study alternatives relative to objectives and constraints
- Identify risks (lack of experience, new technology, tight schedules, poor process, etc.)
- Resolve risks (evaluate if money could be lost by continuing system development)

Spiral
Quadrants

Plan next phase

Typical activities

- Develop project plan
- Develop configuration management plan
- Develop a test plan
- Develop an installation plan

Develop next-level product

Typical Activities:

- Create a design
- Review design
- Develop code
- Inspect code
- Test product

Spiral.. Benefits

- Provides **early indication of insurmountable risks**, without much cost
- Users see the system early because of **rapid prototyping tools**
- Critical **high-risk functions** are developed first
- The **design** does not have to be perfect
- Users can be **closely tied to all lifecycle steps**
- Early and **frequent feedback** from users
- Cumulative **costs assessed frequently**

Spiral.. Drawbacks

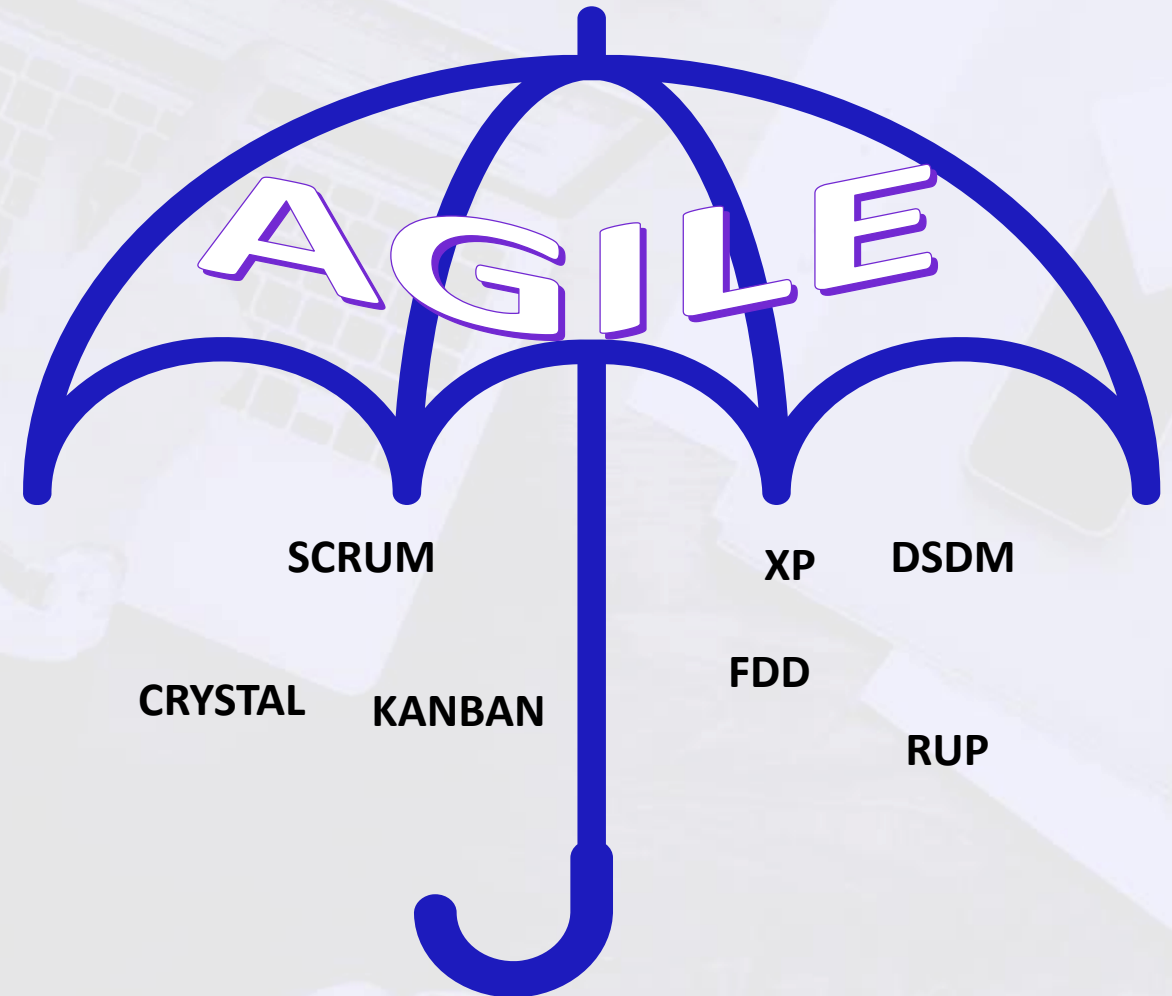
- Time spent for evaluating risks too large for small or low-risk projects
- Time spent planning, resetting objectives, doing risk analysis and prototyping may be excessive
- The model is complex
- Risk assessment expertise is required
- Spiral may continue indefinitely
- Developers must be reassigned during non-development phase activities
- May be hard to define objective, verifiable milestones that indicate readiness to proceed through the next iteration

Spiral.. When to go with?

- When creation of a prototype is appropriate
- When costs and risk evaluation is important
- For medium to high-risk projects
- Long-term project commitment unwise because of potential changes to economic priorities
- Users are unsure of their needs
- Requirements are complex
- New product line
- Significant changes are expected (research and exploration)

Agile Methodologies

- Agile is time-boxed, iterative approaches to deliver software solutions.
- Incremental release.
- An umbrella term for many methodologies under “Agile”.
- Helps in early customer involvement, quality improvement, minimizing risk & much more...



Agile.. Benefits

- Customer satisfaction by rapid, continuous delivery of useful software.
- People engaging as emphasize on individual & interactions over processes and tools.
- Frequent delivery (usually fortnightly, weekly...)
- Changes in requirements are welcomed.
- Highly adaptive, as frequent delivery cause frequent feedback.
- Agile is not a method but mindset.

Agile.. Drawbacks

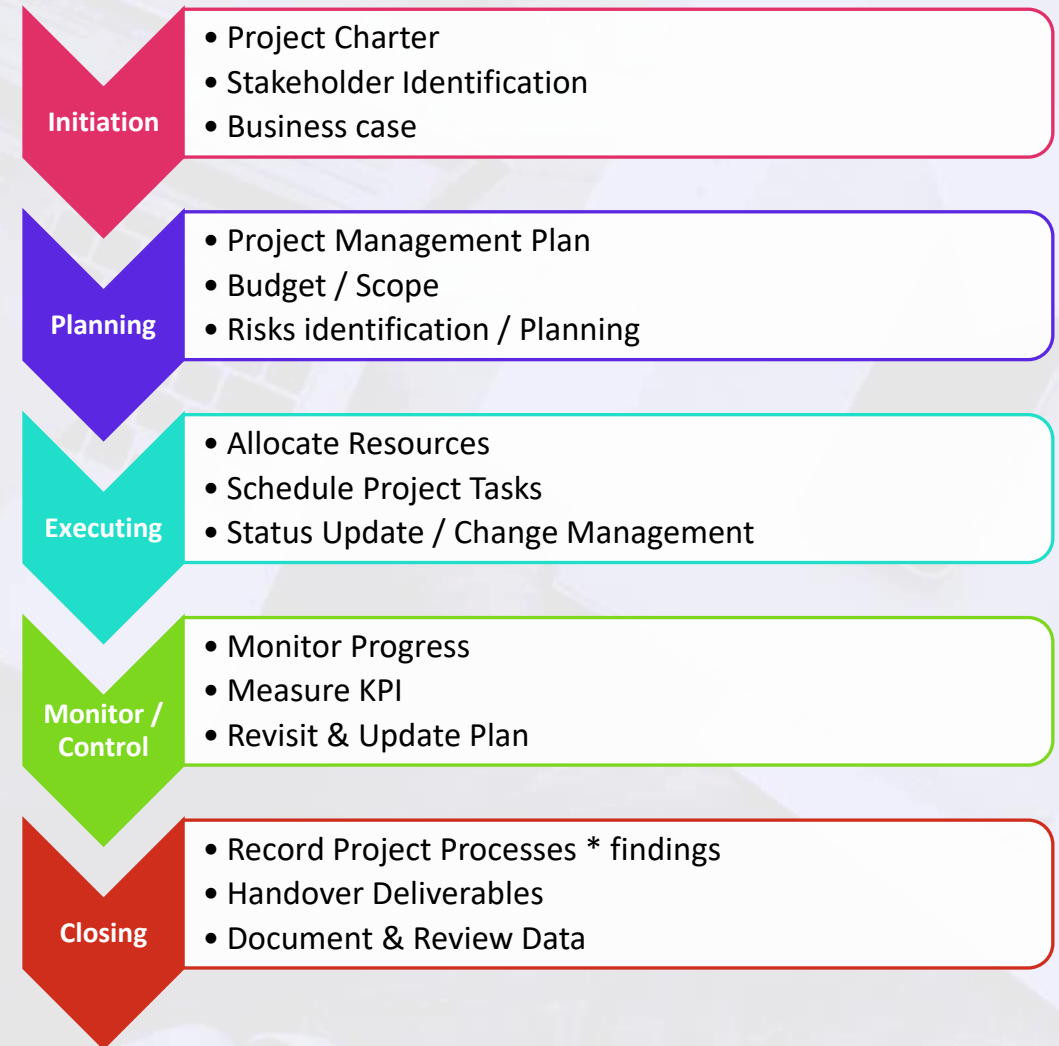
- It is easy to understand, difficult to master.
- It can be difficult to keep the interest of customers / users who are involved in the process
- Team members may be unsuited to the intense involvement that characterizes agile methods.
- Less emphasis on documentation - harder to maintain when you get a new team for maintenance
- Agile is expensive at initial.

Project Planning & Managing Risks

FRAMEWORK(S) FOR SOFTWARE DEVELOPMENT

Project Management

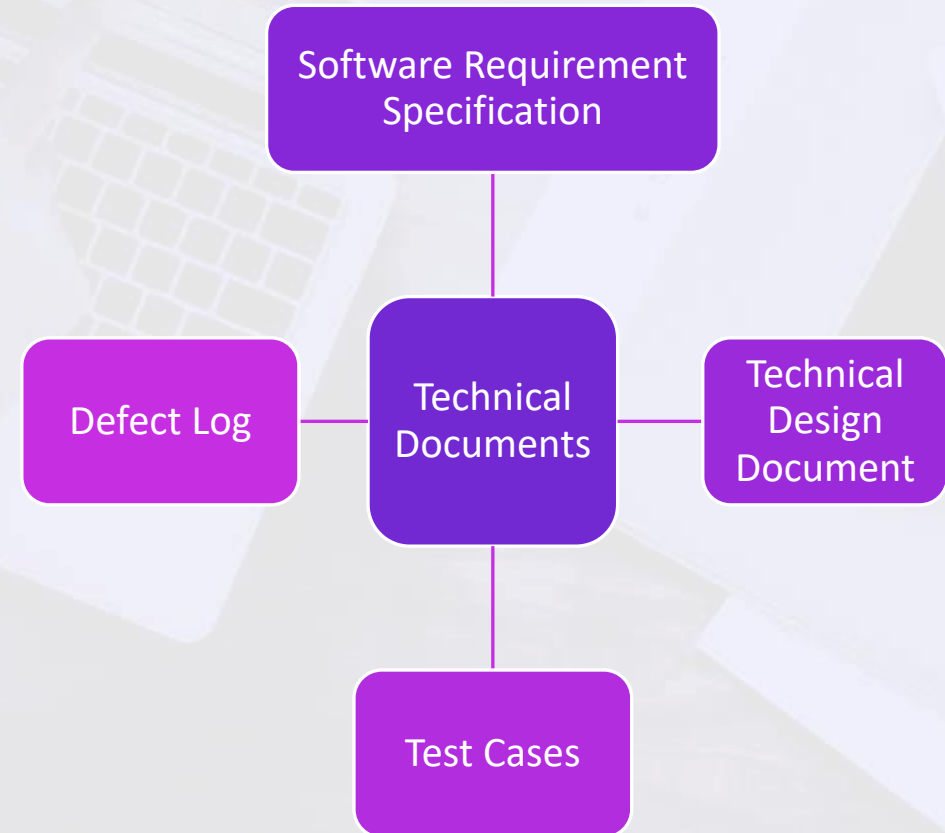
- The Software development life cycles are complimented by project management practices.
- The five process groups of project management framework, helps in managing project end to end.
- All the five groups are important, where “planning” is the most important.



Why Manage Projects? Manage unexpected...

- To accomplish objectives of project within constraints
- Balancing trade-offs between time, cost and performance, these three constraints can be mutually exclusive
- Anticipating, identifying and handling the unexpected
- As project complexity increases coordination and risk also increase
- New technology development is usually associated with increased risk and complexity
- Unexpected events will happen throughout the project (Murphy's Law)
- Risk planning is an essential component to project management considering unique project features

Documentation in Project Management



Things to consider while project planning..

1. Outline business justification and stakeholder needs
2. List of requirements and project objectives
3. Project scope statement
4. List of deliverables and estimated due dates
5. Detailed project schedule
6. Risk assessment and management plan
7. Defined roles and responsibilities
8. Resource allocation
9. Quality assurance (QA) plan
10. Communication plan



Let's us do some practical..