Principles

1. Core Principle
    - Provide value to end user (the reason it all exists)
    - Keep it simple stupid
    - Maintain the vision
    - Be open to future
    - Others will consume what you produce
    - Plan ahead for reuse
    - Think, then act

2. Communication principles
    - Listen
    - Prepare before you communicate
    - Someone should facilitate the communication
    - Face to face communication is the best
    - Take notes and document decision
    - Collaborate with customer
    - Stay focused on a topic, modularize your decision
    - If something is unclear draw the picture
    - Move on to next topic
        a) After you agree to something
        b) If you cannot agree to something
        c) If a function or feature is unclear and cannot be clarified at the moment
    - Negotiation is not a contest or a game , it works the best when both parties win
3. Deployment principles
    - Customer expectation from the software product should me managed
    - A complete delivery package should be assembled and tested
    - A support system must be established before the software is delivered
    - Appropriate instructional material should be provided to end user
    - Buggy software should be fixed first ,delivered later
4. Modelling practices principles

- Primary goal of software team is to build software not create a model
- Don't create more model than you need
- Try to produce simple model
- Build models in such a way that makes then agreeable to change
- Be able to state an explicit purpose for each model that is created
- Adopt the model you develop to the system at hand
- Try to build useful model but forget about building useful model
- Don't become inflexible about the syntax of the model if it communicate contents successfully , representation is secondary
- If your talent tell you a model isn't right even it seems ok on paper, you probably have reason to be concerned

5. Testing principles
    - It is the process of executing a program with the intent of finding errors
    - A good test is one that has a high probability of finding an as yet undiscovered error
    - A successful test is one that uncover an as yet undiscovered error
6. Planning principles
    - Understand the project scope
    - Involve the customer in the planning activity
    - Recognize that planning is iterative
    - Estimate based on what you know
    - Consider risk as you define the plan
    - Be realistic
    - Adjust granularity as you proceed further with the planning
    - Define how quality will be achieved
    - Define how you will accommodate changes
7. Scheduling principles
    - Define clear project objectives
    - Break down work into task
    - Estimate effort and duration
    - Communication and collaboration
    - Regular monitoring and tracking

- **Review and adapt**
- **Allocate resources**

Difference between

1. **Software quality management and software quality assurance**

Software Quality Assurance (QA)	Software Quality Control (QC)
• It is a procedure that focuses on providing assurance that quality requested will be achieved	• It is a procedure that focuses on fulfilling the quality requested.
• QA aims to prevent the defect	• QC aims to identify and fix defects
• It is a method to manage the quality- Verification	• It is a method to verify the quality-Validation
• It does not involve executing the program	• It always involves executing a program
• It's a Preventive technique	• It's a Corrective technique
• It's a Proactive measure	• It's a Reactive measure

2. Waterfall vs Incremental model

Parameters	Waterfall	Incremental
Simplicity	Simple	Intermediate
Risk involvement	High	Comparatively low

manageability	Difficult	Easy
User involvement	At the beginning	Throughout
Flexible	rigid	Less flexible
Maintenance	least	High
duration	Long	Very long

3. Agile vs Prescriptive model

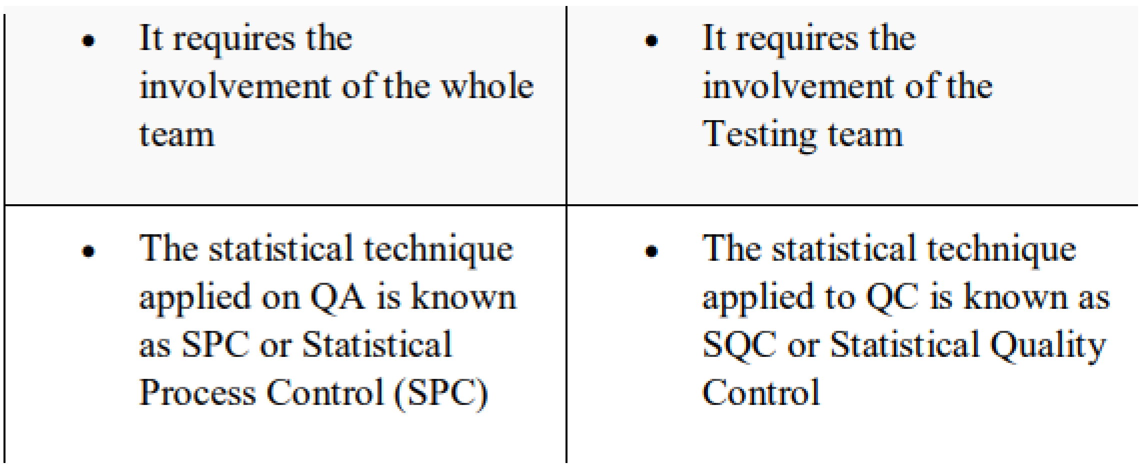
Agile	Prescriptive
These models satisfy customer through fast delivery	Developed to bring order and structure in software process model
Comparatively less popular	More popular
People oriented	Process oriented
It follows iterative development model	It follows life cycle model(waterfall,spiral)
Informal communication is required	Formal communication is required
Customer role is critical	Customer role is important
e.g Extreme programming,Scrum	Waterfall,incremental model

4. White box vs Black Box

Whitebox	Blackbox
Tester need to have knowledge Of internal code	Tester doesn't need to have knowledge of internal code
It aims at testing the structure	It aims at testing the functionality
Aka structural testing, clear box testing , code based testing or glass box testing	Aka data driven , box testing , Data and functional testing
Suited for lower level of testing like unit testing	Suited for higher level of testing Like system testing
Based on detailed designed documents	Based on requirement specification document
Statement , branch and path coverage are white box testing techniques	Equivalence partitioning and Boundary value analysis are black box testing techniques

5.

• It is the procedure to create the deliverables	• It is the procedure to verify that deliverables
• QA involves in full software development life cycle	• QC involves in full software testing life cycle
• In order to meet the customer requirements, QA defines standards and methodologies	• QC confirms that the standards are followed while working on the product
• It is performed before Quality Control	• It is performed only after QA activity is done
• It is a Low-Level Activity, it can identify an error and mistakes which QC cannot	• It is a High-Level Activity, it can identify an error that QA cannot
• Its main motive is to prevent defects in the system. It is a less time-consuming activity	• Its main motive is to identify defects or bugs in the system. It is a more time-consuming activity
• QA ensures that everything is executed in the right way, and that is why it falls under verification activity	• QC ensures that whatever we have done is as per the requirement, and that is why it falls under validation activity



• It requires the involvement of the whole team	• It requires the involvement of the Testing team
• The statistical technique applied on QA is known as SPC or Statistical Process Control (SPC)	• The statistical technique applied to QC is known as SQC or Statistical Quality Control

6. CMMI vs ISO

Parameters	CMMI	ISO
Scope	Broad coverage of software development and management areas.	Focuses primarily on quality management and software life cycle processes.
Approach	Staged representation with maturity levels.	Process-based approach with specific requirements.
Implementation	Involves a systematic and staged approach, including regular appraisals.	Requires establishing and maintaining a QMS, followed by certification audits.
Focus	Process improvement and maturity levels	Quality management and adherence to specified processes throughout the software life cycle.
Industry Application	Software development, IT services , and system engineering	Software development manufacturing ,health care and services

7. Waterfall and spiral

Parameters	Waterfall Model	Spiral Model
Working Nature	Sequential method	Evolutionary method
Involvement of Customer	Minimum, at end of project completion	Maximum, earlier in project development
Identification & rectification of errors	After completion of all stages	Earlier while developing the project
Simplicity	Easy Model	Complex Model
Flow of the Phases	One after another, difficult to go back	In iteration, easy to go back
Project size	Small	Large
Flexibility to change contents	Difficult	Easy
Cost	Less	More
Framework type	Linear	Iterative