# PMI-ACP Exam Notes

**Agile Manifesto**

1. Individuals and interactions over processes and tools.

2. Working software over comprehensive documentation

3. Customer collaboration over contract negotiation

4. Responding to change over following a plan

**Agile 12 Principles**

1. Our highest priority is **to satisfy the customer** through early and **continuous delivery** of valuable software.

2. **Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver **working software frequently**, from a **couple of weeks to a couple of months**, with a **preference to the shorter timescale.**

4. Business people and developers must work together **daily** throughout the project.

5. Build projects around **motivated individuals**. Give them the **environment and support** they need, and **trust them** to get the **job done**.

6. The **most efficient and effective method** of conveying information to and within a development team is **face-to-face conversation**.

7. **Working software** is the primary **measure of progress**.

8. Agile processes promote **sustainable** development. The sponsors, developers, and users should be able to maintain a **constant pace** indefinitely.

9. **Continuous attention** to technical excellence and good design **enhances agility.**

10. **Simplicity**--the art of **maximizing** the amount of **work not done**--is essential.

11. The best architectures, requirements, and designs emerge from **self-organizing teams**.

12. At regular intervals, the team reflects on how to become **more effective**, then tunes and **adjusts its behavior** accordingly.

**AGILE Declaration of Interdependence**

Agile and adaptive approaches for linking people, projects and value, with focus on Agile Leaders. To achieve these results

1. We **increase return on investment** by making continuous flow of value our focus.

2. We **deliver reliable results** by engaging customers in frequent interactions and shared ownership.

3. We **expect uncertainty** and manage for it through iterations, anticipation, and adaptation.

4. We **unleash creativity and innovation** by recognizing that individuals are the ultimate source of value, and creating an environment where they can make a difference.

5. We **expect uncertainty** through group accountability for results and shared responsibility for team effectiveness.

6. We **improve effectiveness and reliability** through situationally specific strategies, processes and practices.

**AGILE Various Methodologies**

**Pillars of Scrum (Three Pillars):**

Transparency/ Visibility | Inspection | Adaptation

**Roles in Scrum (Three):**

Development Team: Self-organizes to get the work done

Product Owner: Responsible for the business value of the project

Scrum Master: Ensures that the team is functional and productive and following scrum

**Scrum Artifacts:**

Product backlog : Ordered/ Prioritized list of ideas for the product

Sprint Backlog: Set of work from the product backlog that the team agrees to complete in a sprint, broken into tasks

Increment of Product: Required result of every sprint. It is an integrated version of the product, kept at high enough quality to be shippable

**Scrum Ceremonies [Planned Opportunities for Inspection and Adaptation]:**

Sprint Planning [Before Sprint]: Team meets with the product owner to choose a set of work to deliver during a sprint

Daily Standup [During Sprint]: Team meets each day to share struggles and progress

Sprint Review [After Sprint]: Team demonstrates to the product owner what it has completed during the sprint.

Sprint Retrospective [After Sprint]: Team looks for ways to improve the product and the process.

**Reasons for transitioning to an agile process like SCRUM:**

Higher Productivity and Lower Costs

Improved employee engagement and job satisfaction

Faster Time-To-Market

Higher Quality

Improved stakeholder satisfaction

What we’ve been doing no longer Work

**Activities for successful Scrum Adoption**: Five common activities necessary for the successful and lasting Scrum adoptions are **ADAPT**:

**A**wareness: that current process is not delivering accepted results.

**D**esire: to adopt SCRUM as way to address current problem.

**A**bility: to succeed with scrum.

**P**romotion: through sharing experience.

**T**ransfer: of implication of using SCRUM throughout company

**Technical Debt:** Consists of deficiencies in the code, technical documentation, development environments, 3rd-party tools, and development practices, which makes the code hard for the team to change.

Paying down technical debt by simplifying or optimizing design improves productivity hence increases velocity.

**Scrum Master’s Role and Responsibilities:** Acts a Servant Leader and

Facilitates Development Team Decision

Removes impediments

Helps product owner manage backlog

Makes sure that scrum is understood by members of team and used

Communicates the vision to the teams.

**Daily Standup**

In Daily standup, team’s answers 3 questions

a. What they did since last meeting

b. What they plan to do till next meeting

c. Any roadblock / Impediments

In Daily Stand-up, the Primary intention is to inspect and adapt work plans.

In Daily Stand-up Team should answer the questions facing towards the team, they are NOT answering to scrum master neither to product owner

**Risk Audit:** It is examining the effectiveness of risk responses and is done as part of retrospective

**Velocity:**

It helps in predicting end date of release

It corrects estimation errors

It helps in customer satisfaction

**Sprint planning**

It is over when the time-box expires.

**Scrumban System**: It’s using KANBAN in combination with Scrum project management methodology.

**II. Extreme Programming (XP)**

Principles are bridge between values and practices

**XP Primary Values:** Four Values are

**a. Communication:** Constant and close communication – among team members and between the team and the customer – are central to the XP development process. All team members should be free to discuss any aspect of a project without fear of a negative reaction.

**b. Simplicity:** XP emphasizes simple design as an effective way of developing quality software and increasing efficiency. XP developers choose simple technology, algorithms, and techniques. They focus on meeting a customer's current requirements – rather than exceeding these or anticipating future requirements that may never materialize. Simplicity also helps ensure that everyone on the team can fully understand the system being developed.

**c. Feedback:** Quick and timely feedback keeps everyone updated on the project. It also enables XP developers to stay abreast of customer requirements so they can implement changes as required. XP employs a multitude of feedback mechanisms, such as daily automated unit tests, constant code reviews, customer stories, and project tracking.

**d. Courage:** In XP, team members need to have the courage to modify a system as user requirements change and in order to make continuous improvements. Because user needs may change frequently, developers need to be brave enough to discard code that's no longer suitable and move fast to try out new approaches – even if these may fail.

**e. Respect** [This value lies below the surface of 4 Values]

Secondary XP values are SAFTEY, SECURITY, PREDICTIBILTY and QUALTIY OF LIFE

**XP Principles:**

a. Humanity: Satisfy both business and personal needs (personal needs are Basic Safety, Accomplishment, belonging Growth and Intimacy)

b. Economics: Solving highest priority business need first.

c. Mutual Benefits: Looking for practice that helps you during programming after programming e.g. refactoring code to remove complexity help in future maintenance.

d. Self-Similarity: e.g. copying the structure of one solution into a new context.

e. Improvement: Find a starting place, get started and improve from there.

f. Diversity: Bringing together team with variety of skills and perspectives.

g. Reflection: Analyzing success or failures, not hiding mistakes but to learn from them.

h. Flow: Continuous flow mean deploy smaller increments of value ever more frequently. One of the example is *daily build*

i. Opportunity: Turn problems into opportunity

j. Redundancy: Sometimes having redundancy in systems helps like having testing phase after each development cycle.

k. Failure: Failure imparts knowledge, Learning from mistakes

l. Quality: Increase in quality leads to improvement in project properties like productivity and effectiveness.

m. Baby Steps: overhead of small steps is much less than big changes. Baby steps are expressed in practice like test-first programming.

n. Accepted Responsibility: e.g. suggesting that whoever signs up to do work also estimates it.

Traceability (This Principle in the case of Safety Critical System)

**XP Practices (Primary Practices):**

1) Sit Together:

2) Whole team:

3) Informative Workspace:

4) Energized Work: 40 Hour work week

5) Pair Programming:

6) Stories:

7) Weekly Cycle:

8) Quarterly Cycle:

9) Slack:

10) Ten-Minute Build:

11) Continuous Integration:

12) Test-First Programming: It addresses many problems at once like

i. Scope Creep: If not

ii. Coupling and cohesion: Loosely couples and highly cohesive code is easy to test.

iii. Trust: Automated test increases the trust in the code.

iv. Rhythm: Test case drives the development. Rhythm i.e. Test, code, refactor, test, code, refactor.

13) Incremental Design:

**XP Practices (Corollary Practices):**

a. Real Customer Involvement:

b. Incremental Deployment:

c. Team Continuity:

d. Shrinking Team:

e. Root-Cause Analysis:

f. Shared Code:

g. Code and Test:

h. Single Code Base:

i. Daily Deployment

j. Negotiated Scope Contract:

k. Pay-Per-Use:

**XP Roles (XP Whole team):** Roles on mature XP team are not fixed and rigid, every one contributes their best, But initially fixed role help in learning new habits

a. Testers:

b. Interaction Designers:

c. Architects:

d. Project managers: He helps team work with rest of the organization

e. Product managers:

f. Executives:

g. Technical Writers:

h. Users:

i. Programmers:

j. Human Resources:

**Extreme Programming Notes:**

**Pair Programming** is XP technique in which two programmers work together at one workstation. The two programmers switch roles frequently. Two roles in pairs are

a. Driver, writes code while the other,

b. Observer or Navigator reviews each line of code as it is typed in

XP chooses **scope** as primary means of planning, tracking and steering project. Time and cost are often fixed.

XP commitments are made AS LATE AS possible

**Test-first programming**, one test at a time and continuous integration which integrates and tests a few hours’ worth of changes at a time. Some example tools for test-first programming are

a. Static Analysis

b. Model Checking

**XP vs. SCRUM**: SCRUM teams don’t allow changes in middle of sprint but teams do allow change.

**Coding Standards**: is an exercise in building consensus. You will learn how to disagree constructively

**Planning Game**: It’s XP Practice to write user stories and estimating them with goal of maximizing customer value.

**Cohesive Design:** means closely related concepts are at one place; it improves design quality and makes it easy to understand.

**Caves and Commons:** refers to the creation of two zones in the room.

a. The Commons area is organized to maximize osmotic communication and information transfer. For this to make sense, the people in the room must be working on the same project. It is perfect for XP’s single team of up to 12 people programming in pairs.

b. The Caves portion of the room is organized to give people a private place to do e-mail, make phone calls, and take care of their need for separation.

**III. Feature Driven Development (FDD)**

**FDD Practices:**

a. Domain Object Modeling: Team explores and explain the domain of problem to be solved

b. Developing By Feature:

c. Individual class/code ownership

d. Feature Teams

e. Inspection

f. Configuration Management System

g. Regular Builds

h. Visibility of progress and results

**IV. Dynamic Systems Development Method (DSDM)**

**DSDM Principles:**

a. Focus on the business need

b. Deliver on the time.

c. Collaborate

d. Never compromise quality

e. Build incrementally from firm foundation

f. Develop Iteratively

g. Communicate continuously and clearly

h. Demonstrate control

**V. Crystal Clear and Crystal Methods**

**Crystal Principles:**

a. Frequent Delivery

b. Reflective Improvement.

c. Osmotic Communication

d. Personal Safety

e. Focus

f. Easy access to expert users

g. Technical Environment

**VI. Lean software development**

**Lean Software Development Principles:**

a. Eliminate Waste: Waste is anything that does not add value OR any delay that keeps customer from getting value when they want. Big form of waste in Software development is “Churn” or “Requirement Churn”. Other waste is like 20% features in software are rarely used. *Write Less Code* i.e. 20%of Code Delivers 80% of value. Multitasking is waste.

b. Build Quality In/ Build Integrity In: Goal is to build quality from start rather than test it in later.

c. Create Knowledge/ Amplify Learning:

d. Defer Commitment/ Decide as late as possible:

e. Deliver Fast / Deliver as fast as possible:

f. Respect People/ Empower Team: These 3 cornerstones gives broader Idea of what respecting people means

i. Entrepreneurial Leader:

ii. Expert Technical Workforce

iii. Responsibility based Planning and Control

g. Optimize the Whole / See the whole:

**Lean Practices:**

a. Seeing Waste:

b. Value stream Mapping: involves representing steps in development process to make easier to identify waste and eliminate waste. In you start from end move backwards.

c. Set-Based Development:

d. Pull Systems:

e. Queuing Theory:

h. Motivations:

i. Measurements:

**Lean Notes:**

As per Lean agile philosophy most errors come from system (Systematic in Nature) and NOT from people. Team should respect people and fix system errors.

Lean principle of eliminating waste is similar to agile principle of simplicity-the art of maximizing the amount of work not done.

**Fast-Flexible-Flow:** It summarizes the lean principle of Optimize the whole with speed and sustainability

**Business value delivered chart:** The entire enterprise (business, management, and development teams) needs the line of sight to velocity (points/time) dashboard-type view of work management which in other terms is a business value delivered chart.

In lean agile testing is done to improve the process and quality.

**VII. KANBAN**

**KANBAN** is agile method for software development of products and processes with an emphasis on just-in-time (JIT) delivery while not overloading the software developers. In this approach, the process, from definition of a task to its delivery to the customer, is displayed for participants to see and developers pull work from a queue.

**KANBAN Principles:**

a. Visualize the workflow

b. Limit WIP

c. Manage Flow

d. Make Process Policies Explicit

e. Improve Collaboratively (using models & the scientific method)

**KANBAN Notes:**

**KANBAN** is Lean based methodology

**Pull System:** is KANBAN scheduling system that signals what to produce and only produces item customer needs.

**Limited WIP:** KANBAN has huge focus on limited work in progress (WIP)

**MMF (Minimal Marketable Features):** is a feature that is minimal, because if it is any smaller, it would not be marketable. This concept originated in KANBAN and Lean.

**Agile Important Points**

Agile project can have fixed end date, fixed cost but it **CAN NOT** have fixed scope.

**Agile Approaches** work well when there are *complex requirements and complex technology*.

Agile management favors adaptation while traditional management methods favor anticipation

**Acceptance Test Driven Development (ATDD)** is a practice in which the whole team collaboratively discusses acceptance criteria, with examples, and then distills them into a set of concrete acceptance tests before development begins.

**Active listening practice:** The Listening skill progression steps are in the following order

a. Internal Listening (How will this affect me), then to

b. Focused Listening (What are they really trying to say) and then to

c. Global Listening (What other clues I notice to help me understand what they are saying)

**Agile projects** we refine requirement From

a. Backlog to Iteration goal then from

b. Iteration goal to Iteration plan then from

c. Iteration plan to user stories then we continue to refine

d. During discussion in daily standup

**Agile Games (Collaborative and Innovation Games)**

a. Remember the future: exercise to vision-setting and requirements-elicitation

b. Prune the product tree: exercise helps gather and shape requirement

c. Speedboat/ Sail boat: exercise to identify threat and opportunities (risk) for the product

d. Buy a feature: exercise for prioritization

e. Bank-for-the-Buck: exercise look at value vs. cost ranking

**Agile Coach Role:** Agile coaching is 40 percent DOING and 60 percent BEING. Important aspect of agile coaching is being model.

a. During Retrospective agile coaches role is as of Facilitator

**Agile Coaching Stages**: Agile team as they practice and then get good at agile:

a. **Shu**: Follow the Rule, Learning the basics, So here coaching style is *Teaching or Mentoring* [1st Stage]

b. **Ha**: Break the Rule, Attained basic knowledge, Here the coaching style is *Coaching* [2nd Stage]

c. **Ri**: Be the Rule , i.e. mastered attained and are in self-discovery mode, here the coaching style is *Advising*[3rd Stage]

**Adaptive Leadership:** Ron Heifetz’s conveys that leader adapts to the environment to lead most effectively i.e. leadership is contingent on situation.

**Agile Conflicts [Lea’s Conflict Model] :** Five Levels of Conflicts

a. Level 1: Problem to Solve

b. Level 2: Disagreement

c. Level 3: Contest

d. Level 4: Crusade

e. Level 5: World War

**Agile Coaching** should be provided to a **new team** before team start up for just 1-2 days, rest can be done during sprint

**Coaching in Sprint**, in the beginning and end whole team coaching is most impactful and less disruptive BUT during the middle of spring individual one-one is most impactful.

In Agile project Product owner is NOT Project Sponsor, So Product owner is not responsible of funding of project.

**Communication Effectiveness** in various communication models in ascending order is:

a. email (*Lowest Effectiveness*)

b. Phone

c. Video Conference

d. Face to face

e. Face to face conversation with white Board *(Highest Effectiveness )*

**Coaching term “Meet them half step ahead”:** It means first determine where coachee stands on agile path. When meet them half-step ahead instead 10 step ahead, i.e. coachee is taking small steps/ baby steps on agile journey

**Fully Burdened Labor Costs** include everything needed to employ a person including his or her salary and benefits, the office space and supplies

**Facilitated workshop** techniques are used to help stakeholders understand complex or ambiguous issues.

**Five Failure Modes (Alistair Cockburn):** are

a. Making mistakes,

b. Preferring to fail conservatively,

c. Inventing rather than researching,

d. Being creatures of habit

e. Being inconsistent.

**BART Analysis** is to check team dynamics. BART stands for Boundary, Authority, Role and Task

**Informative Workspace** is good way to keep everyone in and it also gives whole team opportunity to notice what’s going well and what’s not.

**Project charter** is important both for Agile and traditional projects

In Agile project Customer specifies the Acceptance Tests.

**Vision**, reveals where the project is going and why it is going there

**Draft:** In the context of information flow “Draft” refers to unwanted information flowing osmotically between collocated teams.

**High Bandwidth Communication:** Face to face communication is also refereed as high bandwidth communication.

**Face to Face communication** has highest efficiency and highest richness

Purpose of Brainstorming is to generate large volume of ideas.

Large payback period results in higher risk

**James Shore’s** quiz focused on XP practices is *a self-assessment quiz*.

Agile team go about **developing a schedule** by **estimating story points and making use of the team velocity**

**Agile earned value management** is most appropriately applied at **Iteration Level**

**WIP:** Work in Progress OR Work in Process OR Work in Play. Excessive WIP can be problem

a. It TIES CAPITAL no return on investment till converted into accepted product

b. It hides bottlenecks in process that slows the workflow and masks efficiency issues

c. It is risk in form of REWORK, since there may be still changes to item till its accepted

**Little’s Law:** Cycle time is directly proportional to amount of WIP

**Throughput:** WIP/Cycle Time

**Prioritizing Agile Stories** : Order of consideration is

a. Compliance,

b. Value base,

c. Working software

Agile variants such as SCRUM, XP and LEAN. They all complement other.

Basic Idea of Communication Management is timely collection and dissemination of project related information.

**Escaped Defect:** A problem or error that was delivered to the customer escaped the development team, validation, verification, and acceptance.

**Hardening Iteration (Iteration H):** also known as final sprint or iteration: Many teams will devote the final iteration or sprint to production readiness. That involves some final testing, administration, documentation - many steps that may be unique to the major step of releasing the major increment to the customer.

**Osmotic Communication:** It is indirect information transfer through overhearing conversations, background conversation or simply noticing things happening around you. It’s most effective in collocated team.

As a Project Manager for team that was following traditional waterfall model for long time, you have been asked to follow agile, FIRST thing you should do is to start daily stand-up meetings.

**Horizontal-Market Software** is intended to be used across wide range / many industries

**Vertical-Market Software** is developed for many organizations, However it’s built for PARTTICULAR Industry.

**Lost Opportunity Cost:** is the cost of fixing errors introduces due to team member not communicating with each other.

**Sprint Backlog**

a. It is created and maintained by TEAM

**Product Backlog**

a. It is created and maintained by Product Owner

b. It acts as common collaboration tool between product owner and agile team

**Burn up chart** has two trend lines, one show the progress rate and other for scope.

**Risk Burn down Chart:** It shows cumulative risk severities over time.

**Burn down chart:**

a. Horizontal – Timeline in days, Vertical – Remaining effort in days

**Burn Rate:** is based on velocity and cost

**Release burn down Chart:**

a. It is updated during iteration review.

**Scrum Cycle** starts in the MIDDLE of sprint planning meeting, As Sprint planning meeting as two four hours sessions, 1st part is for prioritizations and agreeing on which stories to be picked up for the sprint. In 2nd part team discuss about the tasks involved. This when sprint cycle starts.

**Trade of Matrix:** guides project manager about relative priority of project constrains (scope, schedule and cost). If team must adept to inevitable changes. The project manager should be clear about whether a particular constraint is fixed, flexible or accept.

**Agile estimation approach is Top down Estimation**.

**Affinity estimation** is based or relative sizing.

a. Example shirt size S M L

b. There should be minimum 20 items in product back log for this technique to work.

c. It confirms the size of story point is consistent and hasn’t drifted over time

**IKIWISI:** I know it when I see it.

**Seven Software Related Wastes:**

a. Partially Done Work

b. Extra Processes

c. Extra Features

d. Waiting

e. Motion

f. Task switching: e.g. Multi-Tasking or working on multiple projects

g. Defects

**Agile Documentation** should be barely sufficient

**Agile Good User Story** should be INVEST**:**

a. **I**ndependent

b. **N**egotiable

c. **V**aluable

d. **E**stimable

e. **S**mall

f. **T**estable

**Extreme Persona** is the people who are not typical users of the product. Considering them helps in identifying the stories which could be missed otherwise.

**User Story Cards** are considers relevant or useful for the project UNTIL work related to story us completed.

**User Story Component**

a. Role [As a …]

b. Function [I can …]

c. Benefit [So That…]

**Adaptive Leadership Style During Team Phase**

a. Forming Phase [Leadership Style Directing]

b. Storming Phase [Leadership Style Coaching]

c. Norming Phase [Leadership Style Supporting]

d. Performing Phase [Leadership Style Delegating]

**Feature Buffer:** There is project in which releasing by particular date is extremely important. Feature Buffer can be used to manage risk. As it will not only help delivering all MUST HAVE features on time but also GOOD TO HAVE features can be picked depending on time.

**Project Roadmap:**

a. In agile high-level features are recorded in the beginning in this document.

b. It provides high level view of which feature will be released and when.

**Risk Severity:** Risk Severity = Risk Probability \* Risk Impact.

**Wideband Delphi:** is method for generating estimates, the term wideband because compare to existing Delphi, the new method involved greater interaction and communication between participants.

**Project Data Sheet (PDS):** It is single page summary of Key business and quality objectives, product capabilities and project management information like Tradeoff-Matrix, Major Milestones, Risks.

**Tacit Knowledge:** is undocumented information supported through team communication

**Process Tailoring** is that the process should add value to the project - not vice versa. Projects should not be shaped to fit the process or tools - the shapes of processes and tools should be driven by the project need.

Technical debt is NOT an XP Practice

Scrum Master OR XP Coach define the ground rules to bind the team together as a unit

Agile Team

a. AVOIDS Low Value and High risk work Features.

b. PICKS High Value and High Risk Feature

**Story Maps:** comprised of component stories that **do not need** to add up to be 100% of the epic. They are typically shown in 2 dimensions, with story cards breaking down the component stories.

**Story Maps:** help select and group features for release , It consists of

a. Backbone: The essential functionality product makes up backbone

b. Walking Skeleton: Its smallest system that could work i.e. the minimum set of features that could create a working product.

c. Additional Features: Any remaining features

**Spike:** A spike solution, or spike is technical investigation.

a. Its’ small experiment to research the answer to problem- can be termed as brief learning period.

b. Is undertaken as early as possible to minimize project risk profile OR reach fast failure

**Triangulation:** In story points estimate we should triangulate i.e. we should NOT compare single baseline, but instead we should compare against assortment of those that have already been estimated.

**Epic Story:**

a. Large user story that can be broken down further.

b. They are found at the bottom of product backlog, as they move up they need to be broken down

**Refactoring:** It enable programmer to improve quality without changing behavior.

**DRY concept:** Avoiding redundant coding. DRY concept recommends removing or avoiding duplicate code.

**Scope Creep in Agile:** In agile scope creep can be take care of by keeping the release plan highly visible and allowing customer to make changes to the priority of the items in backlog at the end of every iteration.

**Global team:** Hold a face to face kick-off meeting and have them work together for 1 or 2 iteration at same location.

**Karl Wieger’s Requirement Prioritization Model OR Relative Weighing Technique:** Relative benefit of feature creation and relative penalty of not creating the features are key factors that provide relative value. Value percentage is divided by cost percentage to derive relative priority.

**Kano Analysis**: It’s a prioritization technique, it classifies the customer preferences in 4 categories

a. Exciters/ Delighters: It brings high value to customer

b. Satisfiers: It brings value to customer, more the better

c. Dis-satisfiers: It cause user to dislike product if those features are not there

d. Indifferent: No impact on customer one way or another

**Iteration planning:**

a. Amount of **slack** you would build depends on **Randomness** of the problem team experiences

b. 8 Hour Duration

**Release Planning Outputs:** are

a. Release plan

b. Release backlog

c. Actions / action items

d. Risk

e. Assumptions

f. Dependencies

**Team Members raising hands showing one to five fingers** is a way of checking consensus

a. Five Fingers: I love this Idea

b. Four Fingers: I am happy with the idea

c. Three Fingers: I can live with and support this idea

d. Two Fingers: I have reservation about this

e. One Finger: I have grave misgiving

**Three-step intervention path/method for** Agile conflict

a. Have you shared your concerns and feelings about this with \_\_\_\_\_\_\_\_\_?

b. \_\_\_\_\_\_\_ should know of your concerns. Would it help if I go with you?

c. May I tell \_\_\_\_\_\_\_\_\_ that you have these concerns?

**Layers for Agile Enterprise Framework**: Four layers are

a. Portfolio Governance Layer

b. Project Management Layer

c. Iteration Management Layer

d. Technical Practices Layer

***“Working software over comprehensive documentation*”** is most important for agile practitioners in winning stakeholders confidence. This value is equivalent to WIP.

**Definition of Agility**: It is ability to balance flexibility and stability.

*Incremental Delivery* Agile Principle helps in chaotic situation with lot of uncertainty.

Web hosting meeting are example of synchronous collaboration.

**Generalized Specialist:** People who can fit in any role on situation. This helps in keeping team size manageable. Generalized Specialist BEST fit for agile team as agile team is cross functional.

**Earned value Management:** Its one of the benefits is that it is a leading indicator. EVM looks forward and try to predict completion date and cost, and gives opportunity to change course in case of a problem.

**Pilot Agile project** should NOT be mission critical project.

**Agile Iterative process** makes progress through successive refinement.

**Agile requirement User role modeling steps:**

a. Identify Customer, then

b. Brainstorming on initial role, then

c. Organize roles, then

d. Refine roles i.e. add persona, consider role attributes; A name of persona is most useful in persona creation.

**Erg-Second:** The energy/time required to transfer important info Or measure of cost to get questions answered in team, where Erg is unit of work and second is unit of time

**Information Radiator:**

a. Task Board, burn-down charts are example of information radiators.

b. Information radiators should be highly visible

c. Information radiators should display the progress and expose the problems.

**Cumulative Flow Diagram:** Theses are like burn down chart but tracking is activity level e.g. analysis, coding, testing etc. which helps in highlighting pipeline delays and queue sizes.

**TDD:** Red Green Refactor

**Jim Highsmith’s Decision Spectrum**: Its participatory decision making tool. It allows people to indicate support for decision and also air their reservation.

**Retrospective Process** : 5 Steps

a. Set the stage:

i. Check-in

ii. Focus on/ Focus off

iii. ESVP ( Explorers, Shoppers , Vacationers and Prisoners)

iv. Working Agreement

b. Gather Data:

i. Timeline

ii. Triple Nickels

iii. Color code Dots

iv. Mad, Sad, Glad

v. Locate Strength

vi. Satisfaction histograms

vii. Team radar

viii. Like to like

c. Generate Insight:

i. Brainstorming

ii. Five Whys

iii. Fishbone

iv. Prioritize With Dots

v. Identify Theme

d. Decide What to Do:

i. Short Subjects

ii. SMART Goals

iii. Retrospective planning games

iv. Circle of Questions

e. Close the Retrospective:

i. Plus/ Delta

ii. Helped, Hindered, Hypothesis

iii. Return on Time Invested (ROTI)

iv. Appreciations

**User Stories** are used when we need to capture high-level objective of the specific requirement.

**Agile Modeling:** Its aim is to capture the intent of design in barely sufficient way.

**Velocity Measurement:** Velocity measurement accounts for work done and disruptions on the project.

Agile methods employ **combination of Management and Leadership.**

**COCOMO (Constructive cost Model):** This model was created by reverse engineering the inputs from completed software projects that had known exact costs.

**Aspect of Emotional Intelligence:** one should work on the improvement on our emotional intelligence in following order

a. Self-Awareness: Self-Confidence

b. Self-Management: Self-Control

c. Social Awareness: Empathy for others

d. Social Skills: Self-Control

**High Performance Team** trusts each other, so the **Trust** and **Transparency** are keys to team excellence.

**Product Box OR Vision Box** capture ONLY top 3 things project must deliver, the limit of 3 is to keep team focused on prioritizing feature and making trade off.