PMI-ACP Exam Glossary

Find more at BareMetalCyber.com

1. **Acceptance Criteria**
   Clear conditions that must be true for a backlog item to be considered complete and acceptable. On the exam, this shows up when deciding whether work is "done" versus merely "finished," and when choosing the next best action to reduce rework and ambiguity.

2. **Adaptive Planning**
   Planning that is intentionally revisited as new information appears, rather than treated as a fixed contract. Exam questions often test whether a candidate chooses feedback-driven re-planning over rigid adherence to an outdated plan.

3. **Agile Chartering**
   A structured way to align a team and stakeholders on purpose, boundaries, working agreements, and how decisions get made. On the exam, it commonly appears in early-team scenarios where confusion exists about goals, roles, or ways of working.

4. **Agile Coaching**
   Guidance that helps individuals and teams improve their agile behaviors, often through observation, facilitation, and targeted feedback. On the exam, the best choice frequently involves enabling learning and self-organization rather than issuing directives.

5. **Agile Manifesto**
   A foundational statement of values and principles that prioritizes people, working outcomes, collaboration, and responsiveness to change. Exam items often use it as a "compass" to identify the most agile-aligned option when multiple answers sound reasonable.

6. **Agile Mindset**
   A set of attitudes and behaviors that favor transparency, experimentation, continuous learning, and adaptation. The exam regularly tests mindset indirectly by presenting pressure scenarios where the right move is to shorten feedback loops and reduce uncertainty.

7. **Agile Metrics**
   Measures used to understand progress and improve decision-making, ideally focused on trends and outcomes rather than blame. On the exam, a common

confusion is treating metrics like velocity as a performance score instead of a planning and forecasting input.

8. **Agile Release Planning**
   Planning what to release and in what order based on value, risk, dependencies, and learning needs, while staying flexible. Exam questions often test whether you prefer incremental releases and validation over "big-bang" delivery.

9. **Agile Risk Management**
   Identifying uncertainty early and reducing it through small experiments, frequent inspection, and adaptive decisions. On the exam, strong answers typically make risk visible and reduce it with learning, rather than deferring risk until late stages.

10. **Andon**
    A Lean concept for signaling a problem immediately so the team can stop, swarm, and restore flow and quality. On the exam, it appears as a decision point about whether to surface issues early and fix root causes instead of pushing defective work downstream.

11. **Backlog**
    An ordered list of work items that represents what the team may deliver, typically evolving as learning occurs. On the exam, backlog questions often test whether you keep work transparent and value-ordered instead of treating the plan as fixed or hiding work in side channels.

12. **Backlog Refinement**
    Ongoing work to clarify, split, and add detail to backlog items so they are ready for selection and delivery. The exam commonly uses refinement as the "right first step" when requirements are vague, items are too large, or the team keeps missing commitments.

13. **Behavior-Driven Development (BDD)**
    A practice that expresses expected system behavior in plain language examples to create shared understanding and testable outcomes. On the exam, BDD often appears as an answer choice when the best move is to reduce ambiguity by turning expectations into concrete examples.

14. **Burn Chart**
    A visual that shows remaining work or progress over time, such as burn-down or burn-up, used to support transparency and forecasting. Exam traps include treating

the chart as a performance weapon rather than a planning signal, or ignoring what it reveals about scope change.

15. **Cadence**
    A regular rhythm for events and delivery, such as iterations, reviews, and retrospectives, that supports predictability. On the exam, cadence usually signals healthy flow and feedback; choices that disrupt cadence without a strong reason are often wrong.

16. **Capacity**
    The realistic amount of work a team can take on during a timebox, considering availability, skills, and other constraints. On the exam, capacity matters when choosing commitments; strong answers avoid overloading the team and prefer smaller slices of value.

17. **Continuous Integration (CI)**
    A practice where code changes are frequently merged and automatically built and tested to detect issues early. Exam questions use CI to test "built-in quality" thinking, where early detection and fast feedback beat late-stage stabilization.

18. **Cumulative Flow Diagram (CFD)**
    A flow visualization that shows how work items move through states and where bottlenecks and queues are forming. On the exam, a CFD supports decisions about limiting work in progress and improving flow, rather than simply speeding up individual steps.

19. **Cycle Time**
    The elapsed time from when work starts to when it is finished, used to understand flow and predictability. The exam often tests whether you use cycle time to improve the system and forecast delivery, instead of relying on optimistic guesses or pressure.

20. **Definition of Done (DoD)**
    A shared set of completion criteria that applies to work and protects quality and consistency. On the exam, DoD appears in scenarios involving partial completion, hidden work, or quality debates; the best answer usually reinforces clear completion standards.

21. **Definition of Ready (DoR)**
    A shared set of criteria that indicates a backlog item is sufficiently understood to be

pulled into delivery work. On the exam, DoR is often the right answer when the team keeps starting unclear items and then thrashing, but it can be a trap if it becomes a gate that blocks learning.

22. **Dependency Management**
    The work of identifying and reducing reliance on external people, teams, systems, or approvals that slow delivery. Exam questions frequently reward choices that reduce dependencies through cross-functional collaboration and slicing work, rather than escalating every problem upward.

23. **DevOps**
    A set of practices and culture that improves delivery speed and reliability through collaboration, automation, and feedback between development and operations. On the exam, DevOps typically appears as an enabler of continuous delivery and quality, not as a separate team that "handles deployment."

24. **Empirical Process Control**
    A way of managing complex work by making decisions based on observation and evidence, using transparency, inspection, and adaptation. The exam tests this by offering choices between "learn fast and adjust" versus "predict perfectly and lock the plan."

25. **Estimation**
    An informed forecast of effort or complexity used to support planning, sequencing, and tradeoffs. On the exam, estimation is not the goal by itself; the best answers use it to improve shared understanding and manage risk, not to force commitments.

26. **Facilitation**
    Guiding a group conversation so it stays focused, inclusive, and productive, especially during planning, reviews, and retrospectives. The exam often frames facilitation as a leadership behavior that helps a team reach clarity and agreement without command-and-control tactics.

27. **Flow**
    The smooth movement of work through a process with minimal waiting, rework, and handoffs. On the exam, flow is a decision lens: strong choices limit work in progress, remove bottlenecks, and shorten feedback loops instead of simply "working harder."

28. **Forecasting**
    Predicting likely outcomes such as delivery dates or scope based on evidence like throughput, cycle time, and trends. Exam questions often contrast evidence-based forecasting with wishful planning or fixed-date promises that ignore uncertainty.

29. **Iteration**
    A short, fixed timebox in which a team plans, delivers, and reviews a slice of value, then reflects to improve. On the exam, iterations emphasize learning and predictability; answers that skip feedback events or defer testing to the end are commonly wrong.

30. **Kanban**
    A method that manages flow by visualizing work, limiting work in progress, and improving continuously. The exam uses Kanban scenarios to test whether you optimize the system with WIP limits and bottleneck management rather than adding more parallel work.

31. **Lean**
    A set of principles focused on maximizing value by reducing waste, improving flow, and continuously learning. On the exam, Lean shows up when you must choose actions that simplify work, reduce delays, and prevent defects instead of adding process overhead.

32. **Lean Waste**
    Work or activity that consumes effort without adding customer value, such as waiting, handoffs, rework, or overproduction. Exam questions often test whether you can spot waste in a scenario and pick the next step that removes the cause rather than treating symptoms.

33. **Minimum Viable Product (MVP)**
    The smallest complete version of a product that can validate a key assumption and deliver real learning or value. On the exam, MVP is a decision tool for uncertainty: the best choice usually favors early validation over building a large feature set before feedback.

34. **MoSCoW Prioritization**
    A technique that groups items into Must have, Should have, Could have, and Won't have (for now) to support tradeoffs. On the exam, it appears as a way to clarify scope under time pressure, but it can be a trap if it replaces true value-based ordering.

35. **Pair Programming**
    Two people working together on the same task, typically one driving and one reviewing, to improve quality and shared understanding. On the exam, it often appears as a quality and knowledge-sharing option, especially when defects or skill gaps threaten delivery.

36. **Product Backlog**
    A backlog specifically representing product work, ordered to maximize value and learning over time. On the exam, product backlog scenarios test whether you keep items value-focused and adaptable, rather than treating the backlog as a static requirements document.

37. **Product Owner**
    The role accountable for maximizing product value through backlog ordering, clarity, and stakeholder alignment. Exam questions often test whether decisions about value and priority stay with the product owner, while the team controls how the work is executed.

38. **Refactoring**
    Improving internal code structure without changing external behavior, to reduce complexity and support future change. On the exam, refactoring matters as a quality investment; the best answers treat it as part of sustainable delivery, not optional "nice to have" work.

39. **Retrospective**
    A regular team meeting to reflect on how work went and decide improvements for the next cycle. The exam frequently frames retrospectives as the primary mechanism for continuous improvement, and traps include skipping them when the schedule feels tight.

40. **Risk Burndown**
    A chart or view showing how overall risk exposure changes over time as risks are addressed. On the exam, it appears in risk-focused scenarios where the right move is to reduce uncertainty early and track whether mitigation actions actually lower risk.

41. **Servant Leadership**
    A leadership approach that focuses on enabling the team by removing impediments, supporting growth, and fostering collaboration. On the exam, servant

leadership often distinguishes the best answer from “manage harder” options, especially in conflict, morale, or performance scenarios.

42. **Spike**
    A timeboxed exploration used to reduce uncertainty, such as investigating a technical approach or clarifying requirements. The exam uses spikes as a risk-reduction move; a common trap is letting a spike turn into open-ended research with no decision output.

43. **Stakeholder**
    Anyone who affects or is affected by the product, including customers, sponsors, users, and internal partners. On the exam, stakeholder engagement appears as a frequent decision point, where the best option increases collaboration and feedback rather than delaying involvement.

44. **Story Points**
    A relative estimation unit that represents effort, complexity, and uncertainty, used to support planning and forecasting. Exam questions often test whether you treat story points as team-specific and relative, not as a direct measure of hours or individual performance.

45. **Sustainable Pace**
    A way of working that can be maintained over time without burnout and quality collapse. On the exam, sustainable pace matters when choosing between short-term heroics and system changes that preserve quality and predictability.

46. **Team Charter**
    A shared agreement that defines purpose, boundaries, roles, working norms, and how the team will operate. The exam often uses charters as a solution when a team lacks alignment, but it is not a substitute for ongoing feedback and adaptation.

47. **Test-Driven Development (TDD)**
    A practice where tests are written first and code is written to make the tests pass, supporting correctness and design clarity. On the exam, TDD shows up as a built-in quality choice that reduces defects and supports safe change.

48. **Timeboxing**
    Setting a fixed time limit for an activity to encourage focus and fast decision-making. The exam commonly rewards timeboxing for meetings and exploration work, but it can be misused if timeboxes become rigid barriers to needed learning.

49. **User Story**
    A short, user-centered description of a need that supports conversation, slicing, and prioritization. On the exam, user stories are often part of “make it testable and small” decisions, where the correct choice improves clarity and acceptance.

50. **Work in Progress (WIP) Limit**
    A constraint on how many items can be actively worked on at once, used to improve flow and reduce multitasking and queues. The exam tests WIP limits in scenarios where “start less, finish more” is the best path to faster delivery and higher quality.