

AgileBA (Agile Business Analysis) Foundation (2015) Exam

APMG-International AgileBA-Foundation

Version Demo

Total Demo Questions: 5

Total Premium Questions: 56

Buy Premium PDF

<https://passqueen.com>

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, AgileBA Overview	8
Topic 2, The AgileBA Framework	3
Topic 3, The AgileBA Process	7
Topic 4, AgileBA Practices	37
Topic 5, Mix Questions	1
Total	56

QUESTION NO: 1

After the first increment has been used, the expected reduction in processing costs has not occurred. The Business Sponsor asks whether the project should continue with the remaining increments.

What is the most appropriate AgileBA response?

- A. Continue because the original Business Case authorised the project
- B. Review and update the Agile Project Business Case
- C. Replace all remaining requirements with Could Have requirements
- D. Close the Prioritised Requirements List until costs are reduced

ANSWER: B

Explanation:

The **Agile Project Business Case should be reviewed and updated**. It provides continuing business justification, so evidence from operational use must be considered alongside expected benefits, costs, risks and assumptions. The appropriate decision-makers can then determine whether the project remains worthwhile, needs to change, or should stop.

Continuing simply because the Foundations phase has been completed ignores the need for ongoing justification. Re-prioritising requirements may be part of a response, but it should follow assessment of whether the investment still represents value. The Prioritised Requirements List is not a substitute for the Business Case.

QUESTION NO: 2

During the Foundations phase, how is the Prioritised Requirements List used for Requirements Planning?

- A. Requirements allocated to each Development Timebox are arranged into a Story Map
- B. Requirements are broken down into tasks and prioritised
- C. Requirements are displayed on a Story Board for visibility
- D. Requirements and priorities are negotiated to allow the first increment to be planned

ANSWER: D

Explanation:

Requirements and priorities are negotiated to allow the first increment to be planned. In AgileBA, the Foundations phase establishes a sufficiently robust shared understanding of the project's business, solution, and delivery foundations before iterative development begins. The Prioritised Requirements List is central to this planning: requirements are assessed using agreed priorities, typically MoSCoW, and the business and delivery team negotiate an achievable set of requirements for the first increment. This creates a realistic basis for incremental delivery while protecting the project's business objectives and ensuring early work focuses on the greatest-value outcomes. The list remains dynamic rather than becoming a fixed, detailed task plan; its purpose at this point is to support agreement on scope and priorities at the level needed to begin planning delivery. This use reflects AgileBA's integration with DSDM, where prioritisation supports incremental development, early delivery of value, and ongoing collaboration between business representatives and the project team. Further information on AgileBA's guidance and its relationship to the Agile Business Consortium's DSDM framework is available from [APMG International's AgileBA page](#) and the [Agile Business Consortium](#).

QUESTION NO: 3

A team is planning an increment with a fixed deadline. A proposed requirement is needed to meet a statutory reporting obligation, and there is no acceptable manual workaround. Several stakeholders argue that it should be classified as a Should Have because it is unlikely to be used often.

How should the requirement be prioritised?

- A. Must Have
- B. Should Have
- C. Could Have
- D. Won't Have this time

ANSWER: A

Explanation:

Must Have is correct. In MoSCoW prioritisation, a requirement is a Must Have where the solution or increment would be unacceptable without it. The frequency of use does not reduce the priority of a statutory obligation when no viable workaround exists.

A Should Have is important but can be omitted temporarily if necessary without making the solution unviable. The requirement is not merely a desirable enhancement, and classifying it as Could Have or Won't Have would leave the delivered solution unable to meet a mandatory obligation.

QUESTION NO: 4

A product team is considering an additional feature that customers have not requested and may not expect. Research suggests that, if it works well, the feature could create a significant positive reaction and differentiate the service from competitors.

How would this type of need be classified in the Kano Model?

- A. Expected need
- B. Normal need
- C. Exciter
- D. Must Have requirement

ANSWER: C

Explanation:

Exciter is correct. An Exciter is a feature that customers may not expect, but which can create disproportionate satisfaction or delight when it is provided. Kano analysis helps the team understand this relationship between fulfilment of a need and customer satisfaction.

An Expected need is taken for granted and causes dissatisfaction if absent. A Normal need produces satisfaction in proportion to the degree to which it is met. The proposed feature is not described as essential or explicitly requested, so these categories are less appropriate.

QUESTION NO: 5

A reporting feature would make monthly work easier for managers, but the existing manual report can continue to be used if the feature is not delivered in the current increment. Stakeholders agree that its absence would be inconvenient but would not make the solution unviable.

How should the feature be classified using MoSCoW prioritisation?

- A. Must Have
- B. Should Have

C. Could Have

D. Won't Have

ANSWER: B

Explanation:

Should Have is correct. A Should Have requirement is important and valuable, but the solution remains viable if it is not delivered in the current increment. There is normally a workaround, although it may be less efficient or less satisfactory.

A Must Have is essential: without it the project or increment would fail. A Could Have is desirable but less important than a Should Have and is normally included only if time and resources permit. A Won't Have is deliberately excluded from the current delivery scope.