

# DUMPS ARENA

## Associate Certification Insurance Suite Analyst Mammoth Proctored Exam

Guidewire Insurance Suite - Analyst

Version Demo

Total Demo Questions: 11

Total Premium Questions: 114

Buy Premium PDF

<https://dumpsarena.co>

[sales@dumpsarena.co](mailto:sales@dumpsarena.co)

[sales@dumpsarena.co](mailto:sales@dumpsarena.co)  
[dumpsarena.co](https://dumpsarena.co)

## QUESTION NO: 1

Elaborate Requirements, Confirm Scope, Plan Project / Sprints, and Infrastructure Sizing are all part of this project phase?

- A. Inception
- B. Pre-Inception
- C. Development
- D. Stabilization

**ANSWER: A**

### Explanation:

Inception is the phase in which a Guidewire implementation converts the initial initiative into a delivery-ready plan. The team elaborates requirements with business and technical stakeholders so that the intended capabilities, workflows, integrations, data needs, and priorities are sufficiently understood for iterative delivery. Confirming scope is essential at this point because it establishes the agreed implementation boundary and provides the basis for release and backlog planning.

Project and sprint planning also naturally belongs in Inception. The delivery team establishes the delivery approach, sequence of work, milestones, staffing expectations, and an initial prioritized plan that can be refined as work proceeds. Infrastructure sizing is performed alongside that planning because environment capacity and topology must be estimated early enough to provision appropriate development, test, and production-supporting environments. Together, these activities reduce uncertainty before sustained configuration, integration, and testing work begins, creating a shared baseline for the implementation. Guidewire describes its implementation services as supporting customers through planning and delivery activities; see [Guidewire Implementation Services](#). Guidewire product documentation is available through the [Guidewire Documentation portal](#), which supports the technical planning and implementation work performed after this phase establishes the project foundation.

## QUESTION NO: 2

According to the training, as a non-developer, what are the common activities that you may be involved in related to integrations?

Choose 2 options.

- A. Defining integration timing requirements (real-time or batch)
- B. Defining the design of the batch file format (fixed width or csv)
- C. Defining integration trigger mechanisms (user action, data change, external call)
- D. Defining the data architecture requirements between systems (extract, transform, load)
- E. Defining the batch process sequence and error handling

**ANSWER: A C**

### Explanation:

Non-developer analysts commonly provide the business requirements that determine *when* an integration must run and *what business event* starts it. "Defining integration timing requirements (real-time or batch)" is therefore appropriate because timing is driven by operational expectations: a business process may require an immediate response while a user is completing a transaction, whereas high-volume or non-urgent work can be scheduled for batch processing. The analyst captures that need from stakeholders so the implementation team can select an appropriate integration pattern.

"Defining integration trigger mechanisms (user action, data change, external call)" is likewise a core requirements activity. Analysts identify the meaningful business events that should initiate information exchange, such as a user submitting a transaction, a record reaching a particular status, or a partner system requesting data. Clearly documenting those events helps ensure that the integration supports the intended InsuranceSuite workflow and that stakeholders have a shared understanding of its behavior.

Guidewire's integration documentation describes integrations in terms of business-facing interaction patterns, including synchronous requests and asynchronous processing, while its developer documentation shows how those requirements are implemented technically. See the [Guidewire Documentation portal](#) and the [Guidewire InsurancePlatform overview](#).

### QUESTION NO: 3

Please select User Story Card best practices from the list below. (Choose two)

- A. Include field requirements in the UI Mock-up tab
- B. Change a requirement number after the story card has been published
- C. Include a requirement number for traceability
- D. Review every requirement with the team

**ANSWER: C D**

#### Explanation:

Including a requirement number for traceability is a User Story Card best practice because it establishes a durable link between the story and the requirement it implements. That identifier supports impact analysis when scope changes, allows testers to connect acceptance and validation work to the intended business need, and helps delivery teams demonstrate that requirements have been addressed. Traceability is particularly valuable on InsuranceSuite programs, where a story commonly has relationships to configuration work, rules, integrations, test assets, and business approval decisions.

Reviewing every requirement with the team is also a best practice because a story card is a shared communication tool rather than a handoff document. Collaborative review helps the product, business, analysis, development, and testing participants reach a common understanding of the requested outcome and identify details that must be clarified before or during implementation. This supports the Agile emphasis on frequent collaboration, transparent backlog items, and ongoing refinement of work. The Scrum Guide describes the need for collaborative Product Backlog refinement and shared understanding of work items; see [The Scrum Guide](#). Guidewire product documentation is available through the [Guidewire Documentation portal](#).

### QUESTION NO: 4 - (SIMULATION)

Data Model Entities: Match the entity type with the appropriate description.

Type key –

Select a match:

atomic data stored about the entity (non-restricted values)

- a single reference to a value in a type-list
- a set of references to another entity
- a single reference to the id of another entity

Foreign key –

Select a match:

atomic data stored about the entity (non-restricted values)

- a single reference to a value in a type-list
- a set of references to another entity
- a single reference to the id of another entity

Field –

Select a match:

atomic data stored about the entity (non-restricted values)

- a single reference to a value in a type-list
- a set of references to another entity
- a single reference to the id of another entity

Array key –

Select a match:

atomic data stored about the entity (non-restricted values)

- a single reference to a value in a type-list
- a set of references to another entity
- a single reference to the id of another entity

**ANSWER: Check the explanation for the answer**

**Explanation:**

Match the entity types as follows:

**Type key → A single reference to a value in a type-list**

**Foreign key → A single reference to the ID of another entity**

**Field → Atomic data stored about the entity (non-restricted values)**

**Array key → A set of references to another entity**

A type key stores one typelist value; a foreign key relates one entity instance to another; a field stores an ordinary atomic value such as text, number, date, or Boolean; and an array key represents a collection relationship to multiple related entities.

**QUESTION NO: 5**

A well written user story follows the INVEST model. INVEST is an acronym that stands for:

- A. Independent, Negotiable, Valuable, Estimable, Small, Testable
- B. Independent, Negotiable, Viable, Elaborate, Software, Technology
- C. Investigate, Negotiable, Viable, Elaborate, Small, Technology
- D. Investigate, Negotiable, Valuable, Estimable, Software, Testable

**ANSWER: A**

### Explanation:

Independent, Negotiable, Valuable, Estimable, Small, Testable is the correct expansion of the INVEST model, a widely used agile heuristic for assessing whether a user story is ready for collaborative refinement, estimation, implementation, and validation. Each term addresses a practical quality characteristic of a story: it should be sufficiently independent to minimize unnecessary sequencing constraints; negotiable because a story expresses an intended outcome rather than a fixed implementation contract; and valuable because it should provide a meaningful user or business benefit. It must also be estimable, meaning the delivery team has enough shared understanding to size the work reasonably. Keeping a story small helps the team complete it predictably within a short delivery iteration while reducing risk. Finally, a story must be testable, with observable acceptance criteria that allow the team to establish whether the expected behavior has been delivered. Bill Wake introduced INVEST as a concise checklist for writing effective user stories, and it remains a recognized agile practice applicable when analysts translate stakeholder needs into backlog items. The Agile Alliance describes user stories and their role in communicating customer value, while Bill Wake's original discussion defines the INVEST criteria: [Agile Alliance: User Stories](#) and [XP123: INVEST in Good Stories](#).

### QUESTION NO: 6

Which of the following are examples of standardized prescriptive designs and solutions available on Guidewire Marketplace, that can be leveraged by a project team to speed up the implementation?

Choose 2 options.

- A. Standard process flows
- B. Accelerators
- C. Out-of-the-box user stories
- D. GO Products
- E. Base product documentation

### ANSWER: B D

### Explanation:

**Accelerators** and **GO Products** are reusable, packaged assets intended to provide implementation teams with proven starting points rather than requiring every design element to be created from scratch. In the Guidewire ecosystem, Marketplace is the central catalog for solutions from Guidewire and its partner community, including applications, integrations, and implementation-focused assets that complement InsuranceSuite deployments.

Accelerators support faster delivery by supplying reusable solution components, patterns, configurations, or integration capabilities that can be adapted to a carrier's requirements. Their value is that they embody previously developed approaches and can reduce the time needed for solution design, development, and testing. GO Products likewise represent prepackaged, prescriptive offerings designed around repeatable insurance business needs and implementation scenarios. A project team can use these assets to establish a more rapid baseline while retaining the ability to configure the solution for the customer's operating model and product requirements.

For an analyst, identifying these offerings early helps shape realistic requirements conversations, informs scope and estimates, and enables the team to assess whether an available Marketplace solution can satisfy a business need. Guidewire describes the available ecosystem solutions through its [Guidewire Marketplace](#); further information on the partner ecosystem is available at [Guidewire Partners](#).

### QUESTION NO: 7

The \_\_\_\_\_ documents the entities and typelists in a Guidewire application. This tool includes information for both the base application entities and custom extensions.

- A. Data Entities
- B. Data Model

C. Data Repository

D. Data Dictionary

**ANSWER: D**

**Explanation:**

**Data Dictionary** is correct because it is Guidewire's generated reference for the application data model. It documents entities and typelists and exposes the metadata that analysts, configurators, and developers need to understand how business data is represented in an InsuranceSuite application. For an entity, the Data Dictionary presents information such as its fields, field data types, relationships, inheritance, and related metadata. For a typelist, it provides the available typecodes and their associated properties. This makes it a practical source for tracing where a business concept is stored and how it relates to other records.

A key capability is that the Data Dictionary incorporates both the product's base configuration and customer-specific configuration. Therefore, when an implementation introduces entity extensions, additional fields, or typelist extensions, the generated documentation can reflect that combined model rather than limiting users to the out-of-the-box schema. This is particularly valuable during requirements analysis, integration design, reporting, and impact assessment, because it provides a consistent view of the configured application model. Guidewire documentation is available through the [Guidewire Documentation portal](#), and Guidewire's product and platform resources are available at [Guidewire Resources](#).

**QUESTION NO: 8**

**Business case completed, business resources trained, and identified are all deliverables of which phase?**

A. Stabilization

B. Development

C. Inception

D. Pre-Inception

**ANSWER: D**

**Explanation:**

**Pre-Inception** is the correct phase because it establishes the organizational and business foundation required before the implementation team begins the detailed project work. A completed business case confirms the initiative's expected value, scope rationale, sponsorship, and commitment to proceed. Identifying business resources ensures that the appropriate subject-matter experts, decision-makers, and product owners are available to participate in the implementation. Training those resources early prepares them to work effectively with the Guidewire delivery approach, terminology, and project responsibilities from the start of the engagement.

These deliverables are readiness outcomes: they demonstrate that the customer organization has both a justified initiative and capable business participation in place. This early alignment is essential for effective discovery, requirements clarification, prioritization, and governance throughout the rest of the implementation lifecycle. Guidewire emphasizes the importance of combining its platform and implementation expertise with active customer participation to achieve successful delivery; see [Guidewire Implementation Services](#) and the [Guidewire corporate site](#) for its implementation and platform-services context.

**QUESTION NO: 9**

**Guidewire Marketplace is a website designed for browsing and downloading \_\_\_\_\_ and product add-ons.**

A. User story cards

B. End-user documentation

C. Detailed requirements documentation

**ANSWER: D**

**Explanation:**

**Accelerators** is correct because Guidewire Marketplace is Guidewire's catalog for solution extensions that complement InsuranceSuite products. Marketplace offerings include accelerators: reusable, packaged assets intended to help insurers implement capabilities more quickly, extend product functionality, and address common business or integration needs. These assets can be supplied by Guidewire, consulting partners, and technology partners, and may include implementation-ready content, integrations, and other prebuilt solution components. Product add-ons are similarly listed through Marketplace, making "accelerators" the term that accurately completes the description of what users can browse and download. In practical InsuranceSuite work, Marketplace helps teams identify assets that can reduce custom development effort and speed delivery while remaining aligned to the Guidewire ecosystem. Guidewire describes Marketplace as a location for finding solutions and integrations that extend its cloud platform and applications. See the [Guidewire Marketplace](#) and Guidewire's overview of its [partner ecosystem](#) for further context on the solutions available through this channel.

**QUESTION NO: 10**

**At the completion of Inception: (Select 2)**

- A. Test cases are written to test end-to-end system functionality
- B. A confirmed scope and estimate is completed with associated user story cards
- C. A conceptual sprint plan is established to guide when user story cards will be built
- D. Documented acceptance criteria is tested to ensure the who, how, and why of story cards is defined

**ANSWER: B C**

**Explanation:**

At the end of Inception, the project team should have a shared, validated basis for proceeding into delivery. A confirmed scope and estimate completed with associated user story cards is a key outcome because it captures the agreed business outcomes at an appropriate level of detail, provides the basis for estimating the work, and aligns stakeholders on the intended implementation boundary. The story cards connect the scope to tangible user and business needs while still allowing the delivery team to refine details as work approaches.

A conceptual sprint plan established to guide when user story cards will be built is also an Inception outcome. This plan is a high-level delivery forecast that sequences the agreed work and helps establish expectations for iterative implementation. It supports planning discussions about priorities, dependencies, team capacity, and likely delivery timing without treating the plan as a fixed, detailed task schedule. This reflects the iterative delivery approach used with Guidewire implementations: establish a reliable scope and plan, then progressively elaborate and build the prioritized stories through sprints. Guidewire describes its implementation and delivery services at [Guidewire Services](#); the iterative planning principles underlying sprint-based delivery are also described in the [Scrum Guide](#).

**QUESTION NO: 11**

**Which of the following statements describe the importance of acceptance criteria in a software implementation project? (Select three)**

- A. They describe desired system functionality when "done" from the business perspective
- B. They describe how to correctly configure and code requirements
- C. They facilitate the writing of automated test scenarios with BDD
- D. They are acceptance tests
- E. They are used to confirm whether the user story can be accepted

**Explanation:**

Acceptance criteria establish the observable, business-focused conditions that a delivered user story must satisfy before stakeholders regard it as complete. Therefore, “They describe desired system functionality when “done” from the business perspective” is correct: criteria turn a business need into specific expected outcomes that product, delivery, and testing teams can evaluate consistently. This shared definition of done is especially important in an InsuranceSuite implementation, where configuration and integrations must meet agreed business behavior.

“They facilitate the writing of automated test scenarios with BDD” is also correct. Concise, testable criteria provide the expected behavior and examples that can be expressed as behavior-driven scenarios and then implemented as automated tests. This improves traceability from a story’s intended outcome through validation and regression testing. The Agile Alliance describes acceptance criteria as conditions a product must satisfy to be accepted by a user, customer, or other stakeholder: [Agile Alliance: Acceptance Criteria](#).

Finally, “They are used to confirm whether the user story can be accepted” is correct because the criteria provide the objective basis for acceptance by the appropriate business or product representative. This aligns delivery decisions with validated business value and supports clear completion decisions. Guidewire’s product documentation portal provides the authoritative documentation context for InsuranceSuite implementation work: [Guidewire Documentation](#).