Exam : **Development-Lifecycle-and-Deployment-Architect**

Title : Salesforce Certified Development Lifecycle and Deployment Architect

Vendor : Salesforce

Version : DEMO

NO.1 Universal Containers CUC) is an enterprise financial company that operates in EMEA, AMER, and APAC.
Because of regulatory requirements, UC has a separate Salesforce org for each region. Each org has its own customizations that fit for the region needs, but there are also standard processes that apply to all regions requirements.
As the deployment architect, what should be considered for the multi-org deployment strategy?
A. Deploy metadata to production orgs using managed packages.
B. Deploy metadata to production orgs using unmanaged packages.
C. Deploy metadata to production orgs using package development model.
D. Deploy metadata to production orgs using change sets.
Answer: C
Explanation:
Deploying metadata to production orgs using package development model is the best option for the multi-org deployment strategy, as it allows you to create modular and reusable packages that can be easily installed and updated across different orgs. Deploying metadata to production orgs using managed packages is not suitable for this scenario, as managed packages are typically used by ISVs to distribute their applications to customers, and they have some limitations and restrictions that may not fit the requirements of UC. Deploying metadata to production orgs using unmanaged packages is also not a good option, as unmanaged packages are mainly used for one-time distribution of components, and they do not support upgrades or dependencies. Deploying metadata to production orgs using change sets is not feasible for this scenario, as change sets can only be used to deploy metadata between connected orgs in the same Salesforce instance, and UC has separate orgs for each region. See [Package Development Model] for more details.

NO.2 Universal Containers (UC) is implementing Service Cloud for their contact centers for 3000 users. They have
~10 million customers. The average speed response time expected is less than 5 seconds with 1,500 concurrent users. What type of testing will help UC measure the page response time?
A. Unit Testing.
B. Load testing.
C. System Integration Testing.
D. Stress Testing.
Answer: B
Explanation:
Load testing is the type of testing that will help UC measure the page response time. Load testing simulates the expected number of concurrent users and measures how the system performs under normal load conditions. Unit testing, system integration testing, and stress testing are not designed to measure the page response time.

NO.3 Universal Containers (UC) has recently acquired other companies that have their own Salesforce orgs. These companies have been merged as new UC business units.
The CEO has requested an architect to review the org strategy, taking into consideration two main factors:
* The CEO wants business process standardization among all business units.
* Business process integration is not required as the different business units have different customers

and expertise.
Which org strategy should the architect recommend in this scenario, and why?
A. Single-org strategy, as the high level of business process standardization will be easier to implement in a single org.
B. Multi-org strategy, as it is uncommon for the diversified business units to get used to working in the same space as the other business units.
C. A Multi-org strategy, as they could deploy a common managed package into the orgs of the different business units.
D. Single-org strategy, as costs increase as the number of orgs go up.
Answer: A
Explanation:
A single-org strategy is the best option for UC, as it will enable them to achieve business process standardization among all business units, which is the main goal of the CEO. A multi-org strategy would make it harder to enforce consistent processes and policies across the different business units, and would also increase the costs and complexity of managing multiple orgs. A common managed package could help with some aspects of standardization, but it would not cover all the possible scenarios and customizations that UC might need.

NO.4 Universal Containers has five development teams. The performance of the teams has been good, but the number of bugs has been increasing. After each sprint, they need more time to understand the code and make changes.
What are two ways to improve the performance?
Choose 2 answers
A. Define a team that will analyze/approve all changes.
B. Define and follow code standards.
C. Sprint review process.
D. Version control system to identify who is generating the bugs.
Answer: B C
Explanation:
To improve the performance of the development teams, the following ways can be suggested: Define and follow code standards, and implement a sprint review process. Code standards can help ensure consistency, readability, and maintainability of the code, as well as reduce errors and bugs. A sprint review process can help evaluate the work done in each sprint, demonstrate the functionality, and gather feedback from the stakeholders and users.

NO.5 Sales and Service products will be created by two teams that will use second-generation managed package(s).
The Sales team will use a specific function of the Service product, but the architect wants to ensure that this team will only use the functions exposed by the Service team. No other team will use these same functions.
What should an architect recommend?
A. Create two second generation managed packages with the same namespace and set the methods that should be shared with the @namespaceAccessible annotation.
B. Create two managed packages with Sales and service namespaces. Set the methods to be shared with the salesAccessible annotation

C. Create a managed package with both products and create a code review process with an approver from each team.

D. Create two managed packages. Create an authentication function in the Service package that will return a token if a Sales user is authorized to call the exposed function. Validate the token in the Service functions.

Answer: A

Explanation:

The architect should recommend creating two second generation managed packages with the same namespace and setting the methods that should be shared with the @namespaceAccessible annotation. This will allow the Sales team to access the specific functions of the Service product without exposing them to other teams or customers. Creating two managed packages with different namespaces will not allow the Sales team to access the Service functions, unless they are declared as global, which will expose them to everyone. Creating a managed package with both products will not allow the separation of the products and the control of the functions. Creating an authentication function in the Service package will add unnecessary complexity and overhead to the solution.

NO.6 Universal Containers has developed teams working on multiple projects. They are exploring a source control tool to track and manage their code/config. What two benefits does a source control tool provide? Choose 2 ans

A. Provide the ability for distributed teams to work in isolation.

B. Provides automated code/configuration deployments.

C. Provides the ability to backup code/configuration changes.

D. Provides the ability to automatically identify issues in code/configuration.

Answer: A C

Explanation:

A source control tool provides the ability for distributed teams to work in isolation, as they can create branches and merge their changes later. It also provides the ability to backup code/configuration changes, as they can be stored in a remote repository and retrieved if needed.

NO.7 By to What three tools should an architect recommend to support application lifecycle methodology Choose 3 answers

A. Database management systems

B. Version control repository

C. Middleware

D. Continuous integration tool

E. Issue tracking Tool

Answer: B D E

Explanation:

To support application lifecycle methodology, you need tools that can help you manage the source code, automate the deployment process, and track the issues and bugs. A version control repository is a tool that allows you to store, track, and collaborate on the source code of your application. A continuous integration tool is a tool that allows you to automate the deployment of your code to different environments, as well as run tests and validations. An issue tracking tool is a tool that allows you to record, monitor, and resolve the issues and bugs that arise during the development and testing phases. A database management system is a tool that allows you to store, manipulate, and

query data, but it is not directly related to application lifecycle methodology. A middleware is a software layer that facilitates communication and data exchange between different applications, but it is not directly related to application lifecycle methodology either.

NO.8 An architect is working on a Universal Containers (UC) project, and due to security concerns, the UC security team cannot provide the Architect with production access. Instead, a central release management team will be responsible for performing production deployments for all development teams.
How should an architect leverage the Metadata API to ensure any metadata components necessary to deploy the project's functionality Are properly communicated to the release management team?
A. Provide a spreadsheet of all components and utilize the metadata API's read Metadata()call.
B. Communicate the unlocked package version to the release management team.
C. Create a change set in each sandbox and download the package.xml file for the release management team.
D. Provide the release management team a copy of the audit trail from the sandbox you wish to deploy from.
Answer: B
Explanation:
The best way to leverage the Metadata API to communicate the metadata components to the release management team is to use unlocked packages. Unlocked packages are a collection of metadata components that can be easily deployed and updated using the Metadata API. They also provide versioning and dependency tracking features that can help ensure the integrity and compatibility of the components.
Providing a spreadsheet, a change set, or an audit trail are not effective ways to use the Metadata API, as they do not capture the full metadata information and require manual intervention.

NO.9 Universal Containers (UC) is preparing for the new Salesforce release in a couple of months, and has several ongoing development projects that may be affected. Which three steps should the team at UC take to prepare for this release? Choose 3 answers
A. Contact Salesforce to schedule a time to upgrade the full Sandbox.
B. Refresh a Sandbox during the Release Preview Window to ensure they have the upcoming release.
C. Run regression tests in an upgraded sandbox to detect any issues with the Upgrade.
D. Review the release notes for automatically-enabled features and technical debt.
E. Upgrade any SOAP integrations to the newest WSDL as early as possible
Answer: B C D
Explanation:
Refreshing a sandbox during the release preview window to ensure they have the upcoming release is a step that the team at UC should take to prepare for the new Salesforce release, as it allows them to test their application in an environment that matches the production environment after the upgrade. Running regression tests in an upgraded sandbox to detect any issues with the upgrade is also a step that the team at UC should take to prepare for the new Salesforce release, as it helps them to verify that the existing functionality is not affected by the new features or changes introduced by the upgrade. Reviewing the release notes for automatically-enabled features and technical debt is also a step that the team at UC should take to prepare for the new Salesforce release, as it helps them to understand the impact and benefits of the new features or changes, as

well as to identify and resolve any technical debt that may cause issues or conflicts with the upgrade. Contacting Salesforce to schedule a time to upgrade the full sandbox is not a step that the team at UC should take to prepare for the new Salesforce release, as it is not possible to request a specific time for the upgrade of the full sandbox, which is determined by Salesforce and depends on the release window and the pod assignment. Upgrading any SOAP integrations to the newest WSDL as early as possible is not a step that the team at UC should take to prepare for the new Salesforce release, as it is not necessary to upgrade the SOAP integrations to the newest WSDL, unless they want to use the new features or fields introduced by the upgrade. The SOAP integrations will continue to work with the previous WSDL versions, as they are backward compatible.

NO.10 Universal Containers (UC) is implementing a governance framework and has asked the Architect to make recommendations regarding release planning. Which two decisions should the Architect make when planning for releases? Choose 2 answers

A. How to test existing functionality to ensure no regressions are introduced.

B. Whether Salesforce will wait to upgrade the pod until after a UC release is complete.

C. How to roll back to the previous Salesforce release if there are issues.

D. When to test a new UC feature release if there are issues.

Answer: A D

Explanation:

How to test existing functionality to ensure no regressions are introduced is a decision that the Architect should make when planning for releases, as it is part of the quality assurance process and helps to ensure that the new changes do not break the existing functionality. When to test a new UC feature release is also a decision that the Architect should make when planning for releases, as it is part of the release schedule and helps to coordinate the testing activities with the development and deployment activities. Whether Salesforce will wait to upgrade the pod until after a UC release is complete is not a decision that the Architect can make, as it is determined by Salesforce and depends on the release window and the pod assignment. How to roll back to the previous Salesforce release if there are issues is not a decision that the Architect can make, as it is not possible to roll back to a previous Salesforce release once the upgrade is done.

NO.11 Universal Containers is building a new complex integration to a legacy system. the legacy system is also going through a major upgrade. Senior leadership has committed to the board that the combined programs will be completed on time. What is the risk with this plan?

A. The deadline is scheduled during a Salesforce release

B. The project team has decided to use the Waterfall methodology

C. The legacy system team is using an Agile methodology

D. Multiple work -streams with dependencies could impact the go-live

Answer: D

Explanation:

D is the correct answer, as the risk with this plan is that multiple work-streams with dependencies could impact the go-live. If the legacy system upgrade and the new integration are not aligned and coordinated, there could be delays, errors, or failures in the project delivery. A is incorrect, as the deadline being scheduled during a Salesforce release is not a risk, but an opportunity to leverage the new features and enhancements that Salesforce provides. B is incorrect, as the project team using the Waterfall methodology is not a risk, but a choice that depends on the project scope, complexity,

and requirements. C is incorrect, as the legacy system team using an Agile methodology is not a risk, but a choice that depends on the project scope, complexity, and requirements. You can learn more about the risk management in the Project Management Strategies for Salesforce Implementations module on Trailhead.

NO.12 Universal Containers is a global organization that maintains regional production instances of Salesforce. One region has created a new application to track shipping containers.
The CIO has requested that this new application be used globally by all the Salesforce instances and further maintained and modified regionally by local administrators.
Which two deployment tools will support the request?
Choose 2 answers

A. Change Sets B

B. Developer Console

C. ANT Migration Tool

D. VS Code with Salesforce Extension

Answer: C D

Explanation:
The two deployment tools that will support the request are the ANT Migration Tool and VS Code with Salesforce Extension. These tools allow the developers to deploy metadata components from one Salesforce org to another, and also to maintain and modify the code locally in their own machines. Change Sets and Developer Console are not suitable for this scenario, because they do not support deploying to multiple orgs or working offline.

NO.13 Universal Containers CUC) is embarked on an enterprise salesforce transformation journey, UC would like to streamline and automate deployment to different sandboxes during the build phase. Upon customer acceptance in UAT, the company requested to automate the production deployment as well.
As the deployment architect, what is the recommendation to satisfy the customer requirements?

A. Recommend using the Continues integration and the Continues deployment tool and build the pipeline to deploy to sandboxes and production.

B. Recommend using SFDX and documents the deployment commands with steps to be executed for each environment.

C. Recommend using the ANT script and build a custom application to run the script and use change sets to deploy supported metadata.

D. Recommend using an AppExchange solution that packages the deployment components and you can run the deployment wizard to track deployment result.

Answer: A

Explanation:
The recommendation to satisfy the customer requirements is to use a Continuous Integration and Continuous Deployment tool and build the pipeline to deploy to sandboxes and production. A Continuous Integration and Continuous Deployment tool can automate the process of building, testing, and deploying the changes to different environments, as well as provide feedback and visibility into the deployment status and results. This can help to streamline and accelerate the deployment process, as well as to ensure consistency and quality across the environments. Using SFDX and documenting the deployment commands with steps to be executed for each environment

is not a good recommendation, as it still requires manual intervention and execution, which can be error-prone and time-consuming. Using ANT script and building a custom application to run the script and use change sets to deploy supported metadata is not a good recommendation, as it involves using multiple tools and methods, which can increase the complexity and risk of the deployment process. Using an AppExchange solution that packages the deployment components and you can run the deployment wizard to track deployment result is not a good recommendation, as it may not support all the metadata types and features that need to be deployed, and it may not integrate well with the SFDX tools and methodologies that UC is using.

NO.14 Universal Containers has a full sandbox that will be used to analyze and fix bugs found in production.
Which two items should the architect recommend to ensure that bugs found in production are more easily analyzed in this full sandbox?
Choose 2 answers

A. Refresh the full sandbox after every deployment in production.

B. Create a daily process of copying new and changed data in production to the full sandbox.

C. Before any deployment in production, the same process must be performed in this sandbox.

D. Perform a Refresh Data in the full sandbox.

Answer: A C
Explanation:
To ensure that bugs found in production are more easily analyzed in the full sandbox, the architect should recommend to refresh the full sandbox after every deployment in production, and to perform the same process in the sandbox before any deployment in production. This way, the full sandbox will be in sync with the production org and will have the same metadata and code. Creating a daily process of copying new and changed data in production to the full sandbox is not necessary, as the full sandbox already has a copy of the production data. Performing a Refresh Data in the full sandbox is not possible, as this option is only available for partial copy and developer pro sandboxes.

NO.15 What are three advantages of the package development model?
Choose 3 answers

A. Improving team development and collaboration.

B. Eliminating the need of using change set, which should no longer be used as it can get messy working with package development models.

C. Facilitating automated testing and continuous integration.

D. Significantly reducing the need for manually tracking changes.

E. Providing its own source control, so the source can be deployed In any sandbox orgs.

Answer: A C D
Explanation:
The advantages of the package development model are improving team development and collaboration, facilitating automated testing and continuous integration, and significantly reducing the need for manually tracking changes. The package development model allows the developers to work on modular and reusable components that can be easily tested and deployed. The package development model does not eliminate the need of using change sets, as they can still be used for deploying non-packaged components or metadata. The package development model does not provide its own source control, but rather relies on external source control systems such as Git.

NO.16 Universal Containers CUC) has developed extensions of Salesforce Service Cloud for the use of its customer service teams using the change set development model.
Recently, UC acquired a company that develops extensions of an AppExchange app. The development team of the acquired company uses the org development model. The Universal Containers CTO wants both teams to work on a single org and follow the same set of processes.
Which development model should the architect recommend to be used by the consolidated development team?

A. Org development model, because the acquired company's team is already using it, and it is better than the change set development model.

B. Package development model, because it allows packages to be created and deployed using declarative (point-and-click) development tools, without writing code.

C. Package development model, so teams can build release artifacts that can be tested and released independently from artifacts for other projects.

D. Change set development model, because UC is already using it, so it will face less resistance.

Answer: C
Explanation:
The development model that the architect should recommend to be used by the consolidated development team is the package development model. This model allows teams to build release artifacts that can be tested and released independently from artifacts for other projects, using unlocked packages or second-generation managed packages. This model can help improve the modularity, reusability, and maintainability of the code and configuration, as well as enable source-driven development and continuous integration and delivery. The org development model is not a good choice for the consolidated development team, as it is based on working directly in an org and using change sets or metadata API tools to deploy changes. This model can lead to conflicts, dependencies, and governance issues, especially when working on multiple projects or with multiple teams. The change set development model is also not a good choice for the consolidated development team, as it is based on using change sets to deploy changes between connected orgs. This model can be slow, error- prone, and limited, as it does not support all metadata types, dependencies, or automation. The package development model does not allow packages to be created and deployed using declarative (point-and-click) development tools, without writing code. This is a wrong statement, as the package development model supports both declarative and programmatic development tools, and requires writing code to create and install packages using the Salesforce CLI or APIs. See Package Development Model for more details.

NO.17 Universal Containers (UC) have developed a managed package targeted for AppExchange. The product includes some Apex code to customize and create layouts. UC is in the testing phase of the package, so it's not certified yet. During testing on the target org, the Apex code for the layouts fails.
Why are the Apex classes not able to access the metadata of the target org during testing?

A. Apex Settings to allow the access to metadata is not switched on.

B. UC needs to turn on Apex Settings within the custom metadata type.

C. The solution is flawed. UC should utilize the Tooling API from a web service call to modify the layouts.

D. UC needs to get the managed package certified by the Salesforce security review.

Answer: D
Explanation:
The reason why the Apex classes are not able to access the metadata of the target org during testing is that UC needs to get the managed package certified by the Salesforce security review. This is because Apex code in a managed package can only access the metadata of the target org if the package has passed the security review and has been granted the Modify Metadata permission. See Apex Metadata API for more details.

NO.18 Universal Containers (UC) works with different partners and has few admin resources that take care of the day-to-day deployment tasks. As a result, UC would like to find a way to automate the deployments using Metadata API. Which two limitations of Metadata API should be considered when using Metadata API-based Deployments? Choose 2 answers

A. Deploy up to 10,000 files, but retrieve more than 10,000 files.

B. Maximum size of deployed .zip file is 400MB.

C. Maximum Size of deployed .zip file is 39MB.

D. Deploy and retrieve up to 10,000 files at once.

Answer: A C
Explanation:
The maximum size of deployed .zip file is 39MB for Metadata API. You can deploy up to 10,000 files, but retrieve more than 10,000 files using Metadata API.

NO.19 Universal Containers has multiple projects being developed in parallel. One of the projects is in the testing phase and the testing team found a list of issues on the items that will be deployed to production. As the project deadline is short, the customer team proposes that the fixes be done in the test sandbox and then deployed to production. What should be the Architect recommend?

A. Recommend the customer team's proposal to fix the issues in the testing environment and deploy them to production.

B. Recommend fixing the issues in the development environment and deploying the changes to production.

C. Recommend fixing the issues in the development sandbox, migrating them to testing, and deploy to production after testing.

D. Recommend fixing the issues in the test environment and migrating the changes to the development sandbox.

Answer: C
Explanation:
The best practice is to fix the issues in the development sandbox, migrate them to the testing sandbox, and deploy to production after testing. This ensures that the development and testing environments are in sync and that the changes are properly tested before going to production. Fixing the issues in the testing environment and deploying them to production may introduce errors or conflicts, as the development environment may not have the same changes. Fixing the issues in the test environment and migrating them to the development sandbox may also cause errors or conflicts, as the development environment may have other changes that are not ready for testing.

NO.20 Universal Containers has just initiated a project to implement partner community. The application will be deployed into a production environment currently in use by a large Salesforce user

base. The project manager has insisted that the development and testing team use a single developer sandbox. What is the risk with this approach?

A. Tester will encounter platform limits due to developer sandbox capacity limits.

B. Testers will experience functional changes throughput testing due to not having isolation form development.

C. Testers will hit governor limits due to large volume of users in the developer sandbox.

D. Refreshing the developer sandbox will take significant time.

Answer: B

Explanation:

Testers will experience functional changes throughout testing due to not having isolation from development.

Using a single developer sandbox for both development and testing is not a good practice, as it does not provide a stable and consistent environment for testing. The developers may make changes to the code or configuration that affect the functionality or behavior of the application, which may cause the testers to encounter unexpected results or errors.

NO.21 Universal Containers (UC) is considering updating their Salesforce Production Deployment as a part of their Release Mgmt process. Which three best practices should UC consider for Production Deployment? Choose 3 ans.

A. Announce the maintenance window ahead of time.

B. Define a rollback strategy.

C. Lert all users on the day of deployment.

D. Schedule releases with Salesforce upgrades.

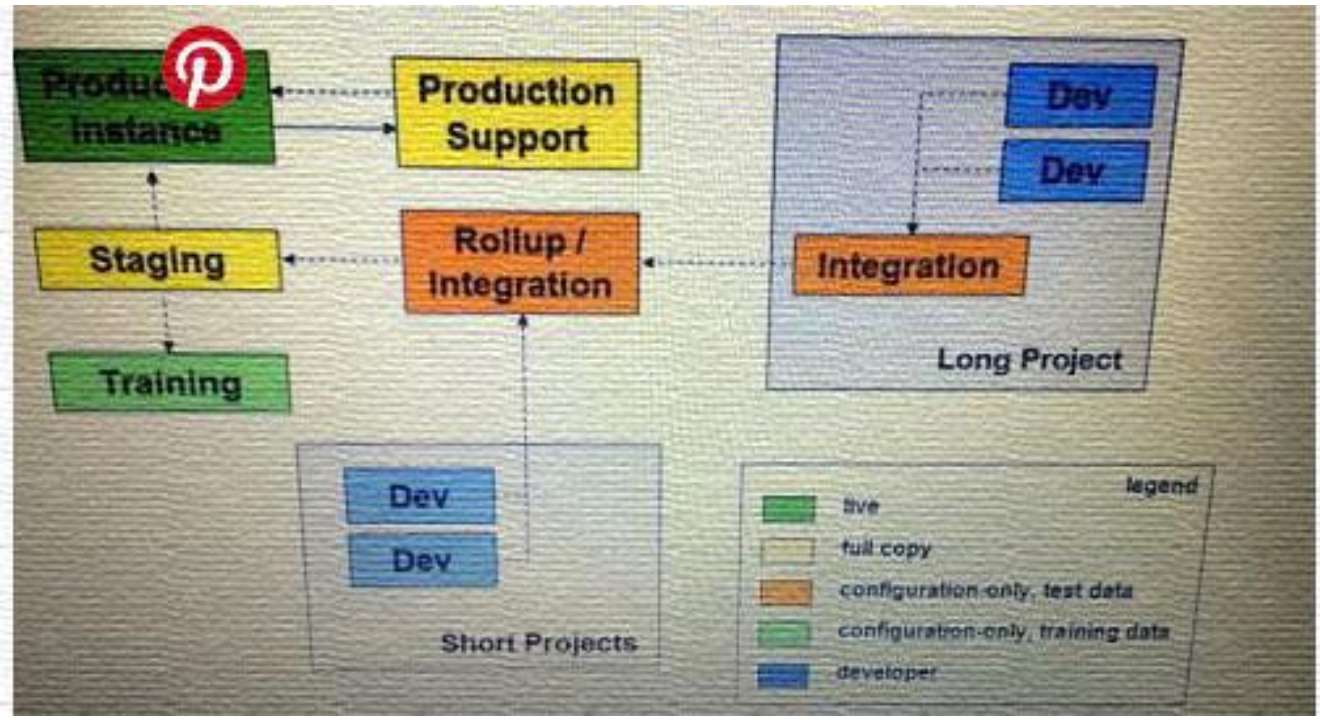
E. Temporarily suspend configuration changes in production.

Answer: A B E

Explanation:

Announcing the maintenance window ahead of time, defining a rollback strategy, and temporarily suspending configuration changes in production are all best practices for production deployment. These practices help to minimize the risk of errors, downtime, and conflicts during the deployment process.

NO.22 The release will be deployed over a weekend, one week after Salesforce updates the production environment (e.g., from Winter to Spring). UC has found that a full sandbox refresh can take several days. What should the architect suggest as an optimal deployment plan?



A. Two weeks before go -live, deploy to Staging and then refresh the Staging and Production support sandboxes. Deploy from Staging to Production at go-live

B. Approximately six weeks before go -live, ensure the sandbox will be on the release preview. One week before go live, deploy to Staging. Deploy from Staging to Production at go-live

C. One month before go -live, deploy to Staging and to Production Support. Deploy from Production Support to Production at go-live

D. One week before go -live, initiate the Staging sandbox refresh and then immediately deploy to Staging.Deploy from Staging to Production at go-live

Answer: B

Explanation:

The best option is to ensure the sandbox will be on the release preview, which means it will be upgraded to the new platform release before the production environment. This will allow the team to test the deployment in a realistic scenario and catch any potential issues. Option A is not ideal, as the staging and production support sandboxes will not be on the same platform version as the production environment. Option C is also not ideal, as the production support sandbox will not be on the same platform version as the production environment. Option D is risky, as the staging sandbox refresh may not complete in time for the go-live.

NO.23 Universal Containers wants to do weekly releases to production. What approach will mitigate the risk of bugs being inadvertently introduced to production?

A. User Acceptance Testing

B. Use of an Agile Methodology

C. Requirements Traceability

D. Automated Testing

Answer: D

Explanation:

D is the correct answer, as automated testing is the best approach to mitigate the risk of bugs being

inadvertently introduced to production. Automated testing can help to ensure that the code meets the quality and functionality standards, as well as detect and prevent any errors or regressions before deploying to production. A is incorrect, as user acceptance testing is not enough to mitigate the risk of bugs, as it depends on the manual testing and feedback of the end users, which can be subjective and incomplete. B is incorrect, as the use of an agile methodology is not directly related to mitigating the risk of bugs, as it is a way of managing the project scope, requirements, and delivery. C is incorrect, as requirements traceability is not directly related to mitigating the risk of bugs, as it is a way of tracking the requirements and their fulfillment.
You can learn more about this topic in the Testing Strategies module on Trailhead.