Sample Exam

Edition 201710

Content

Introduction

This is the sample exam EXIN DevOps Master™ (DEVOPSM.EN). The Rules and Regulations for EXIN's examinations apply to this exam.

This exam consists of 50 multiple-choice questions. Each multiple-choice question has a number of possible answers, of which only one is the correct answer.

The maximum number of points that can be obtained for this exam is 50. Each correct answer is worth one point. If you obtain 33 points or more you will pass.

The time allowed for this exam is 120 minutes.

Good luck!

Sample exam

1 / 50
What is a good reason to implement DevOps in an organization?

A) DevOps has more frequent feedback cycles for new services, because the development speed is higher.
B) DevOps improves business continuity and agility, because the processes are optimized and only value added activities are done.
C) DevOps provides the new services Just-in-Time, because it releases software more frequently.
D) DevOps reduces waste in the collaboration between Development and Operations, because they are unified.

2 / 50
The CTO thinks that it would be most effective to apply certain Lean concepts when implementing DevOps.

Which Lean principles or practices will be **most** effective when introducing DevOps?

A) **Kaizen and 5S.** Because Agile and DevOps are based on core Lean concepts and Kaizen and 5S are the basis of Lean, they will be most effective when introducing DevOps.
B) **Kaizen in advance.** DevOps requires feedback from Operations to Development. Kaizen in advance creates an up-stream feedback loop, helping to apply this principle in DevOps.
C) **Obeya system.** DevOps integrates different management style processes. The Obeya system helps visualize the entire process, allowing for a successful DevOps introduction.
D) **One piece flow and JKK.** DevOps benefits from building up-stream processes and a single value stream flow. One-piece-flow enables this and JKK helps streamline and implement the flow.

3 / 50
Compared to a project in a conventional manner, what must be changed for a project to be successful in DevOps?

A) An IT-services supply chain, using a pull system and one-piece-flow, should be built.
B) Developers should join the Operations team for quick maintenance of the services.
C) Operations must work for the Development team. That is why it is called DevOps.
D) Team members from the Operations team should join the Development team.

4 / 50
Any team that adopts DevOps, could be described as a Compact.

What principle applies **best** to make a Compact work?

A) Dynamic collaboration
B) Ongoing communication
C) Reducing cycles
D) Shared accountability

5 / 50
To implement DevOps, there are many sources of knowledge, standards and practices available.

Which one is considered **key** for a successful DevOps implementation?

A) CMMI Level 3
B) Disciplined Agile
C) ISO 20001
D) PMI PMBok

6 / 50
Collaboration is one of the four pillars of Effective DevOps.

Why is it so important?

A) Collaboration is the only way to achieve one-piece-flow in DevOps.
B) DevOps business value is achieved through cooperation between teams.
C) DevOps organizations are small, the teams must collaborate well.
D) It drives change and is an easy win. It also reduces development costs.

7 / 50
You are the owner of a small DevOps company of 5 co-workers, that builds mobile apps for children with disabilities. One of the apps your team is most proud of is an app for children with autism, that allows them to schedule their own day.

Because this app was so successful, your team is asked to extend the capabilities of the app for other people that would benefit from scheduling their day. This request will certainly generate more complexity in the code and some technical challenges that will have to be tackled by the team.

You expect to be paid very well for this assignment, so you accept the assignment. However, after a few weeks, your team is fighting. You get angry as well and start paying more attention to what all the team members do. You regularly work along with them, so you can quickly correct any code error you spot.

Although enough work gets done, your team stays angry and you recognize that part of the DevOps mindset is missing.

What is the **best** strategy to solve your problem?

A) Add members to the team seeking more diversity, through hiring complementary non-technical personnel
B) Ask experts in DevOps outside your company to sponsor and mentor your team members
C) Find a common goal and start working towards that goal together to increase collaboration
D) Reserve a few months time to start working on team-building first and learn about each other

8 / 50
You feel that your Development team is really a team.

What is a sure sign that they are a team and not a group?

A) The team follows the rules they have agreed upon in their team meetings.
B) The team has effective meetings which they lead themselves.
C) The team keeps a steady working pace towards their common goal.
D) The team solves problems by questioning the responsible team member.

9 / 50
AppAtoZ has been growing at phenomenal rates in developing and deploying mobile applications to the iPhone and Android platforms.

The Development teams for this startup have experienced tremendous pressures in deploying rapid enhancements to their current mobile applications on an aggressive timeline. They have been working on average 60 hours per week over the last six months. Leadership has been reluctant to hire more assistance, and is more concerned about increasing revenue while reducing operational and development costs.

In the recent months there has been an increased rate of employee absenteeism, employees calling in sick, and some employees even resigned from AppAtoZ, resulting in increased workloads on current employees. Rehiring and ramp up time of a new employee does not alleviate the work pressures quickly for the Development team.

Which **long term** strategies must be considered to address employee burnout and stress?

A)
- Add a mix of permanent and contract employees to the team to better manage the workload
- Have leadership and Development identify factors in the work environment that contribute to burnout
- Develop a plan to address employee burnout and stress

B)
- Fire the leadership team, because they are incapable
- Hire a more competent leadership team that knows how to create a work-life balance and culture that is sustainable and realistic
- Provide the opportunity for the Development team to take time to find their balance

C)
- Have the Development team members take a time out
- Contract developers for managing the increased development workload at peak periods of the year
- Have developers reach out for professional mental help as needed

D)
- Have the Development team members take a time out
- Have leadership and development identify all factors in the work environment that might be contributing to burnout
- Develop a plan to address employee burnout and stress

10 / 50
You work in a DevOps team as a developer. You try to facilitate collaboration between all of your team members. Your team has two male senior developers and one junior female systems administrator.

Your team members had a rocky start and could not get along. You intervened at the time, to try to help them get along better. That worked, because they stopped fighting and got a lot more work done.

Now you start noticing a pattern where the junior administrator keeps agreeing with the senior developers.

How should you react to this pattern?

A) As long as the team members are not fighting, you should leave the situation alone. They are just getting along and you should not disrupt this.
B) It is important that Operations voices any concerns with builds, so you should coach the administrator to stop accommodating and be more assertive.
C) The senior developers have more responsibility, so you should ask them to be nicer and to think of Operations concerns if the junior does not.
D) You should start working in the Operations team for now, even though you are a developer yourself, so you can show the junior administrator the way.

11 / 50
Why do Agile and Scrum promise faster and more predictable software development?

A) By allowing better and complete requirements gathering and handling prior to design
B) By allowing small autonomous, self-organizing and self-planning teams
C) By allowing the Product Owner to participate in daily standup meetings
D) By allowing the Project Manager to quickly change priorities as needed

12 / 50
What is light-weight ITSM?

A) A business-continuity focused ITSM
B) A new ITIL version proposed as standard
C) A poor implementation of ITIL processes
D) A release-management oriented ITSM

13 / 50
You work for a company that has implemented Lean and Agile practices. Your CEO is not convinced of the value of implementing yet another new thing: DevOps.

You are a DevOps expert. You feel that DevOps is beneficial to any company that develops software.

What does DevOps add to the company you work for?

A) Creating better User Stories and functional requirements
B) Releasing less often, so customers do not have to suffer regular updates
C) Training professionals to do both Development and Operations
D) Writing code that is released faster supporting business outcomes

14 / 50
Senior business management is requesting better business support and alignment to business objectives from the IT department. As a CIO, among other measures, you decide to reduce management workload in Operations.

How will DevOps **best** help you to improve your existing Service Level Management?

A) By abandoning ITSM best practices as they are just a heavyweight approach
B) By designing better Operating Level Agreements (OLA) between Dev and Ops
C) By implementing a new set of ITSM processes based on ITIL version 3
D) By realigning ITSM to use a set of minimum required information (MRI)

15 / 50
Which DevOps implementation is **most** suitable for an enterprise that uses the System of Record (SoR) approach?

A) Collaboration
B) Continuous Delivery
C) Toyota Way

16 / 50
What is the **main** benefit of using the Obeya system?

A) Facilitates customer complaints, to ensure that the team gets enough feedback to continuously improve
B) Handles stress within teams, so that team members can keep up a sustainable pace
C) Improves daily bug reporting, which ensures less rework and reduces passing bugs to other workstations
D) Quick decision making, based upon current status, by speedy information gathering and sharing

17 / 50
You work in a software company that is just starting to work with DevOps. You realize that not only the culture of the organization should change, but also its practices and tools. The company does already monitor the software projects for any issues.

You propose to add the following measures:

- automate the Change Management process
- implement access controls to prevent anyone for making changes without approval

Why is this necessary?

A) Automation enables faster change implementations while maintaining confidence. Access control is necessary to prevent problem-solving heuristics and unplanned service disruptions.
B) Automation is necessary to slow down the number of changes. Access control should prevent customers from changing the software by themselves, without your knowledge and control.
C) Automation will help to make Operations unnecessary. Access control must be implemented, because DevOps projects work in a Cloud environment, which has higher risks involved.

18 / 50
Virtualization and Cloud computing are techniques that could help and facilitate DevOps practices.

How do they help DevOps?

A) Cloud computing makes remote access possible, which gives customers more control.
B) Virtual environments are easier to standardize and use hardware more efficiently.
C) Virtualized infrastructure is easier to understand and needs zero maintenance.

19 / 50
Service Level Agreements (SLAs) are important for every project, because they specify what you agree on with the customer. However, in DevOps, the SLAs serve another important purpose.

What is this purpose?

A) Customers are responsible to create the SLA for the DevOps team. Therefore, it replaces a formal task as an Operations responsibility.
B) In the SLA, the customer can specify all the non-functional requirements they have, so Development can focus on those.
C) The SLA specifies the acceptable Service Level. Development should understand the SLA and support Operations to maintain it.

20 / 50
For a new product, your team needs to develop a Deployment Pipeline. As part of Continuous Integration, you need to define the Commit stage of the pipeline. You discuss this stage with your team members.

The Process Master says: "The Definition of Done should be defined during or before the Commit stage. When code is not Done when it is committed, the work should be stopped".

Is this true?

A) Yes. If the work is not Done, the Process Master is not doing a good job. This should be solved immediately.
B) Yes. Work that is not Done should not be committed, because it does not add customer value.
C) No. The Definition of Done is only defined during customer meetings. Waiting for it would slow work too much.
D) No. Work in a Deployment Pipeline should always continue. If code is not Done, it just needs to be inactive.

21 / 50
What is the **main** benefit of increasing the diversity of the team to include a wider range of personal backgrounds and cultures?

A) It brings a greater number of experiences and points of view.
B) It leads to decreased friction amongst the team.
C) It limits originality and ability to come up with new insights.
D) It takes longer to come to a specific decision point.

22 / 50
You work for a small company, that has a single DevOps team. Your DevOps team works on an application that consists of multiple components. Some are new and some just need updates.

At the moment, each component has its own Deployment Pipeline. The team is proud of the Continuous Delivery they do within most of the Deployment Pipelines and their production is high and of good quality.

What should you do?

A) Keep the different pipelines and encourage the team to expand Continuous Delivery
B) Only keep pipelines that have Continuous Delivery and merge the other pipelines without Continuous Delivery
C) Only keep two different pipelines: one for development and one for maintenance
D) Work towards a single pipeline with your team, by explaining the risks of multiple pipelines

23 / 50
You are assessing Company Builders, which is a medium to large organization that has adopted DevOps practices a couple of years ago.

They have hired you to determine their current maturity state. When you are done, you should give suggestions for improvement. They want to know on which area they should focus to reach the next maturity level: *Level 2 - Quantitatively Managed.*

You find that most areas are within *Level 1 - Consistent*, with two exceptions:

1. **Environments and Deployment.** This area manages orchestrated deployments and has tested release and rollback processes.
2. **Build Management and Continuous Integration.** In this area, you find regular automated builds and testing, and any build can be re-created from source control using an automated process.

First determine the level of maturity in these two areas, based on the information given. Then give your recommendation for the focus of improvement.

Which of these two areas should Company Builders work on, before progressing to Level 2?

A) Environments and Deployment, and Build Management and Continuous Integration are both at level 0. The work should be done on both environments at the same time.
B) Environments and Deployment, and Build Management and Continuous Integration are either at level 1 or above. The work should be done in the other areas to progress.
C) Environments and Deployment is at level 0. Build Management and Continuous Integration is at level 1. The focus should be on Environments and Deployment first.
D) Environments and Deployment is at level 2. Build Management and Continuous Integration is at level 0. The focus should be on Build Management and Continuous Integration only.

24 / 50
Your company sells an online back-up service to its customers. Now, one of the customers has asked to implement new features in your service. They want the new features within a week, or they will seek business elsewhere.

You believe that the new features are important and you know the Development team can build it quickly. However, you encounter the following problems:

- It takes a long time for bugs to be closed by testers.
- Testers are finding bugs that developers fixed a long time ago.
- The application can rarely be demonstrated to be working.
- Showcases rarely happen.

What is your problem and how should you solve it?

A) You deploy buggy codes. You should solve this by increasing the understanding of the deployment process, increasing the collaboration and by working in a more disciplined manner.
B) You have poor Configuration Management. You should solve this by increasing the collaboration between Development and Operations, by increasing monitoring and logging as well as virtualization.
C) Your Continuous Integration process is not managed properly. You should solve this by speeding up automated tests and the Commit stage, and increase the understanding of the Continuous Integration process.
D) Your testing strategy is not effective. You should solve this by automating tests and increasing the collaboration between testers and the rest of the team.

25 / 50
Which is a benefit of Continuous Integration within Effective DevOps?

A) Extensive testing cycles before feature releases
B) Long periods of time between feature releases
C) More frequent and timely feature releases

26 / 50
A multinational organization is having many challenges in merging commits from their distributed locations into a central repository based in Dallas, TX. The distributed locations include Mexico City, Paris, San Diego and England. There have been inconsistent manners on when the regions are running their commits and at times it is not clear if some tests failed or not.

There are four possible practices:

1. Do not check in on a broken build
2. Always run all commit tests locally before committing, or get your Continuous Integration server to do it for you
3. Wait for commit tests to pass before moving on
4. Do not comment out failing tests

Which of these are **most** applicable to make sure that the distributed locations address their current pain points?

A) 1 & 2
B) 1, 2 & 3
C) 2 & 3
D) 2, 3 & 4

27 / 50
Consider the anatomy of a basic Deployment Pipeline.

Which stage asserts that the system works at the functional and non-functional level?

A) Automated acceptance test
B) Build and unit test
C) Manual acceptance test
D) Version Control

28 / 50
It is a DevOps best practice to use the same process to deploy to every environment in which your application runs. This ensures that the build is tested effectively. You are using scripts to automate your build and deployment process.

What is the **best** way to do this?

A) Use one script for each environment and maintain them as part of the Version Control system
B) Use one specific script for each environment to address the differences between environments
C) Use the same scripts for each environment, taking manual parameters for specific configurations
D) Use the same scripts to deploy to each environment and manage configuration information separately

29 / 50
Upon release of a new IT service, a job in operation ends unexpectedly.

What could **not** be a valid cause for this?

A) The Gatekeeper did not take the interrelation between work items and SAC into account.
B) The non-functional requirements were not clear, due to an imperfect User Story.
C) The Service Master did not agree with the user on the End of Life of the service to be released.
D) There was a lack of work in the development processes, as is supported by the Service Acceptance Criteria (SAC).

30 / 50
Company AppBC is using DevOps. They have implemented Continuous Deployment and a solid Deployment Pipeline, with highly automated acceptance tests and they are delivering daily new software to production.

AppBC has a large database and many users. They have a comprehensive and solid capacity testing strategy in place. As their environment is quite large and complex, with each new version some bugs appear in production.

What strategy could **best** help AppBC prevent these bugs?

A) Adopt canary releasing
B) Automate capacity testing
C) Decrease the delivery rate
D) Use Blue-Green deployment

31 / 50
DevOps takes very important concepts from Agile, derived from the Toyota Production System.

Why is one-piece-flow important for a DevOps adoption?

A) It allows your team to work in a sustainable pace with predictable velocity.
B) It helps the team to focus on adding features with the most business value possible.
C) It increases the shared responsibility of work between the different team members.
D) It reduces bottlenecks by limiting the number of tasks you are doing simultaneously.

32 / 50
The S Corporation, a medium sized automobile parts supplier, supplies a large company, T Motors Corporation. They supply automobile parts to T Motors Corporation, which amounts to almost 60% of their total sales.

The board meets to discuss the new partnership. T Motors Corporation demands that S Corporation change their way of delivery to Just-in-Time delivery, or they will stop their business with S Corporation. S Corporation would not survive this loss of business, so there is a sense of urgency to change to Just-in-Time. This change must happen within 6 months, so there are 5 months at most to prepare.

One of the things to be implemented is tracking of parts through Radio Frequency ID's (RFID). This should help keep the production process transparent. A quick review of the current process is in order to facilitate the change to an RFID enabled process.

The CIO is asked to manage the change process. She believes this will be possible, if a DevOps approach is used to create a minimal release. Ideally, the concept for production with RFID should be developed first. As a last step, a production control system using the RFID data should be implemented. However, there is not enough time to do these steps sequentially. Therefore, these three things should be done concurrently.

The CIO assigns Em, who is a Scrum Master, to the project. Development prepares to build a Deployment Pipeline.

Em can see that Development is enthusiastic and works hard, but they could use more discipline. In addition, the release frequency needs to be higher.

What should Em focus on **first**?

A) Em should focus on communication, since it is the most important thing in DevOps. Em should start with breaking the ice with the team and setting some rules for communication.
B) Em should start with discussing a Value Stream Map and building one-piece-flow with the team, since flow and a streamlined process are very important.
C) Em should start with discussing the infrastructure and the working environment with the team members, since DevOps is most effective when all tools and practices work.
D) Em should start with gathering all stakeholders, educate them on DevOps and ask for their support in spreading the cultural change, since cultural change is required for DevOps.

33 / 50
Your DevOps team works well together at a sustainable pace. By building enough slack into the process, the team has the time and concentration to carefully check and test the builds. Currently, your team tests and deploys manually. Their pace is high enough to deliver high value to the business on a regular basis.

Your CEO has requested your advice on automation within this team.

Which advice should you give?

A) Automate as much as you can, so that the team can add more features and sooner demonstrate business value
B) Automate the acceptance tests, but not the deployments, because the manual process is more secure
C) Automate the deployments to improve cycle time, but not the tests, to allow learning from bugs
D) Do not add automation to the methods of this team, because the methods currently executed by the team are providing amazing results

34 / 50
A CIO assigns her most reliable employee, Michael, who is a Scrum Master, to a project. The Development team prepares to build a Deployment Pipeline.

Michael has confidence in the good intentions and spontaneity of the development team, but would like them to become more disciplined. In addition, there should be a higher release frequency. Michael wants the Development team to implement more frequent releases.

One of the team members says: "The most important thing about this new Deployment Pipeline is automating it. We should first automate the Deployment Pipeline".

Is this statement correct?

A) Yes, this is correct. Automating the Deployment Pipeline is the most important factor for increasing the efficiency.
B) Yes, this is correct. By focusing on creating a Deployment Pipeline that is automated, you overcome potential problems that you may encounter later.
C) No, this is not correct. Achieving single-piece-flow and a solid deployment process should be the first priority. Automation of the process can come later.
D) No, this is not correct. Instead of automating the Deployment Pipeline, the testing process should be automated first.

35 / 50
Your company is changing its ways and starting to work with DevOps. Your team is on board with this change. You are discussing best practices for the Commit stage of the code.

Your co-worker Sun says: "When a build breaks, and nobody takes responsibility, we should find out who did it and call them out on it, so that they can fix the build."

Is this a good idea?

A) Yes. Only the person who breaks a build can fix it, so you should identify them, even if this makes them uncomfortable.
B) Yes. You should always find the person responsible for breaking a build. If you do not, your co-workers may enforce this rule.
C) No. DevOps is a blame free environment. If a co-worker does not take responsibility, do not force them to.
D) No. You should fix the build first. Then make time to identify the person responsible and punish them for it.

36 / 50
The Development team at XAppGo has been running into numerous challenges with their current testing practices. Currently, they use a manual acceptance testing process. The developers believe that the unit test suite that they have created is thorough enough to protect against regressions.

The Development team has to spend 1 million dollars on manual acceptance testing for every release. Senior leadership has mandated that the Development team should implement automated acceptance testing to reduce overall costs of testing and also minimize the number of code defects and regressions introduced into the production environment.

What principles must be followed when defining acceptance criteria for your application with automation in mind?

A) Agile principles
B) ATAM principles
C) INVEST principles

37 / 50
What is the **most** effective mechanism for migrating data in an automated way?

A) Create a database versioning schema and keep it under Version Control
B) Create and manage smaller datasets, so that the migration becomes easier
C) Ensure all your scripts have been properly tested prior to migrating the data
D) Ensure you have a rollback procedure in place in case the migration fails

38 / 50
Company X-AppGo has been having challenges with their roll-back processes. This often results in critical data losses within their production application databases, when executing roll-back scripts.

When is it **not** possible to run roll-back scripts without losing critical data?

A) The roll-back script deletes data that only the new version uses.
B) The roll-back script involves moving a column between tables.
C) The roll-back script will add data back from temporary tables.

39 / 50
ACMECONST has encountered many application and hardware failures after deploying application software upgrades and hardware refreshes to their routers and switches.

It has been very difficult for them to recover back to their original state, after having these failures during their maintenance windows. This has resulted in extended hours of recovery, beyond the normal maintenance windows, and extended downtime for their critical applications.

Automated provisioning and autonomic infrastructure can help within this situation, but some considerations apply.

Which items need to be managed carefully to reduce the risk of disruption when deploying to the production environment?

A) Detailed monitoring logs to troubleshoot application upgrade failures
B) External integration points, such as external systems and services
C) Server configurations and underlying user account information
D) The set of automation provisioning tools and autonomic architecture

40 / 50
Company X-AppGo has been having challenges in their core application. The application is not properly interfacing with other external applications. These external applications need to effectively obtain specific data variables, so that specific calls can be executed. The core application is being developed by one team, and the company wants to maintain that for very good business reasons.

One of the developers suggests to separate out a component from the X-AppGo codebase to tackle the interfacing issues.

What are good reasons to separate out a component in this case?

A) Convert a set of plugins in the codebase into a monolithic codebase
B) Limit the impact of changes and make changing the codebase easier
C) The X-AppGo codebase will have to be split and managed by different teams
D) There are no good reasons and this will require more time to compile

41 / 50
Even the smallest applications will have a dependency on other components or libraries. Therefore, understanding and managing dependencies is a key activity within Continuous Deployment in order to keep flow within the Deployment Pipeline.

You have built an application that uses two libraries. Each of these libraries rely on a third, underlying library, although they refer to different versions. This creates a specific dependency.

What is the **best** solution to solve or prevent this dependency?

A) Assemble all libraries into a single library, so that you can refer to the library directly and prevent the problem
B) Manage the libraries by using Version Control, so that you see it directly if you create this type of dependency
C) Keep a visual overview of all your dependencies on sticky notes on a big board, so you can track the flow
D) Only check in small parts of your toolchain, so that you can easily debug problems that may occur upon check-in

42 / 50
Within a Continuous Deployment environment, it is important that everything is version controlled, so that you can find errors quickly, or roll back whenever necessary.

However, it is **not** recommended to keep binary output within Version Control.

Why is this exception made?

A) Binary output tends to be in large files that change with every build and are updated automatically.
B) Multiple team members work on the binary files, so it is not practical to keep this in version control.
C) The binary output is the input for your compilers, which are already kept in version control.
D) There is no need to do this, since recompilation is done as a regular part of the normal build process.

43 / 50
You want to take a holistic approach to managing all of your IT infrastructure.

On which two principles can this approach be based **best**?

A)
1. The desired state of your infrastructure should be specified through change-controlled configuration.
2. You should always know the actual state of your infrastructure through monitoring and Event Management.

B)
1. The desired state of your infrastructure should be specified through change-controlled configuration.
2. You should always know the actual state of your infrastructure through instrumentation and Incident Management

C)
1. The desired state of your infrastructure should be specified through version-controlled configuration.
2. You should always know the actual state through current Incident and Event Management.

D)
1. The desired state of your infrastructure should be specified through version-controlled configuration.
2. You should always know the actual state of your infrastructure through instrumentation and monitoring.

44 / 50
Teams with good collaboration practices have synchronized work tickets. A CTO used 'Go and See', to investigate how the Operations team functions. After releasing, the Operations team always redefines the Operational infrastructure.

What is the **best** advice to improve upon this practice?

A) They should do nothing. There is no improvement possible, because the redefinition step should always be done.
B) They should examine a way to model the operational infrastructure and access control of the operational environment.
C) They should review the operational infrastructure, so that that becomes an automated process.
D) They should start to share their knowledge of the deployment process with the Development team.

45 / 50
When is a good time for Operations to inform Development of an operational change?

A) Development does not have to be informed. Operational changes are for the Operations team only.
B) Immediately. Development must be informed as soon as possible.
C) In the Scrum of the Scrum's meeting the next morning.
D) When the Operations team has done the acceptance testing.

46 / 50
You want your DevOps organization to mature. There are many ways to do this.

What is **not** a way to help your DevOps organization mature?

A) Clearly define targets as milestones to help your team members judge if their daily activities are valuable
B) Define processes clearly and support and enable the team members to improve the process daily
C) Keep recordings of all meetings so that your team members have easy access to all communication
D) Monitor and record daily activities to help identify small areas of day-to-day progress and celebrate them

47 / 50
You work for an IT service provider. As a part of your business continuity plan, you want to guarantee that you can always meet the minimum agreed service levels.

You want to ensure IT service continuity.

How can DevOps help you with IT Service Continuity Management?

A) DevOps cultural values, affinity and collaboration, make sure that service is highly valued by the DevOps team members.
B) DevOps prepares the team's disaster routines and Obeya practices by deliberately introducing chaos in the system.
C) Risk reduction measures and recovery options are likely coded in, because Operations is working together with Development.
D) Service Level Management becomes more important in DevOps, because the Process Master's task is to monitor this.

48 / 50
ACMECONST has aggressively expanded its global presence by increasing the number of hires and engineering teams distributed throughout the world. It also has been increasing its customer base at a dramatic pace of 30% per year.

Decisions that were once easily made when the engineering team was in one room are now taking much longer, causing frustration across the organization. There are more layers of management approvals to go through and the process is more extensive, which is causing many of the engineers to get disenchanted with the entire decision making process.

There is also increased confusion around the ownership of the various problems that are presented, causing a hesitation on making decisions. The engineers also feel their creativity has been stifled by the additional processes and bureaucracy, which has started to impact their morale.

What is the **best** way to address this scenario?

A) Keep current processes, but establish clear roles, accountability and ownership for each process, establish an effective method for weighing productivity versus risk, make incremental changes and create safe places for experiments
B) Re-examine processes to identify where things can be streamlined and establish clear roles, accountability and ownership for each process, establish an effective method for weighing productivity versus risk, make incremental changes and create safe places for experiments
C) Re-examine processes to identify where things can be streamlined and establish clear roles, accountability and ownership for each process, establish an effective method for weighing productivity versus risk, make incremental changes and minimize the amount of experimentation to prevent unnecessary application failures

49 / 50
Within company X-AppGo there is a conflict between the Operations team in Colombia and the Development team in Ireland, caused by the fact that they have different priorities and goals. Due to this conflict, the amount of time and effort it takes to resolve issues that impact the business is increasing.

Which key practices should X-AppGo consider in order to reduce conflict and improve collaboration between the Development and Operations teams?

A)
1. Allow Development and Operations teams to work separately from each other, if they prefer, to avoid conflicts.
2. Obtain complete executive board buy-in on supporting the Development and Operations teams.

B)
1. Get a sponsor from the executive board of the company to talk to the DevOps team about the importance of working together.
2. Train the Development and Operations teams in DevOps practices, so they learn to do each other's work.

C)
1. Make sure that the Development and Operations teams visit other companies where DevOps works well.
2. Increase funding to better support the increased demands that both the Operations and Development teams are facing.

D)
1. Recommend site visits between the Development and Operations teams to build rapport, develop trust and understanding.
2. Spread knowledge between Development and Operations teams so they work together more effectively.

50 / 50
A Development team is interested in DevOps. They are mainly interested in Continuous Integration (CI). They currently develop and maintain 3 major solutions and 4 smaller ones. They use Scrum practices. Each Sprint takes 4 weeks, creating an average of 1 committed release to the test environment each 10 or 15 days and 1 release to production per month. They want to create a qualitative business case for their management to support their investment and effort to create a CI practice.

Which tangible benefits of CI help that business case **most**?

A) Deploying to test environment once per day could increase business benefits and greatly decrease development costs.
B) It helps the team spirit. As they are already using Scrum, CI will **not** generate measurable benefits for the business.
C) It increases release stability and quality with better and automated testing, facilitating and increasing the overall release speed.
D) Releasing to production once per day could increase business benefits and greatly decrease development costs.

Answer key

1 / 50
What is a good reason to implement DevOps in an organization?

A) DevOps has more frequent feedback cycles for new services, because the development speed is higher.
B) DevOps improves business continuity and agility, because the processes are optimized and only value added activities are done.
C) DevOps provides the new services Just-in-Time, because it releases software more frequently.
D) DevOps reduces waste in the collaboration between Development and Operations, because they are unified.

A) Incorrect. Feedback cycles are more frequent when development speed is higher, but this is usually due to Scrum or other Agile methodology, not DevOps per se.
B) Correct. Adding value and optimizing processes are the keys to improving business continuity and the agility of the company. You should think of what it means that IT services should always support the business, what the value and purpose are of DevOps. *(Literature: C, Chapter 2)*
C) Incorrect. Delivering Just-in-Time is great, but it is not a good reason on its own to implement DevOps. Implementing Lean processes would work better for this goal.
D) Incorrect. Just adding two teams together will not ensure removal of Muda (waste). They need to change their practices to start removing waste.

2 / 50
The CTO thinks that it would be most effective to apply certain Lean concepts when implementing DevOps.

Which Lean principles or practices will be **most** effective when introducing DevOps?

A) **Kaizen and 5S.** Because Agile and DevOps are based on core Lean concepts and Kaizen and 5S are the basis of Lean, they will be most effective when introducing DevOps.
B) **Kaizen in advance.** DevOps requires feedback from Operations to Development. Kaizen in advance creates an up-stream feedback loop, helping to apply this principle in DevOps.
C) **Obeya system.** DevOps integrates different management style processes. The Obeya system helps visualize the entire process, allowing for a successful DevOps introduction.
D) **One piece flow and JKK.** DevOps benefits from building up-stream processes and a single value stream flow. One piece flow enables this and JKK helps streamline and implement the flow.

A) Incorrect. Although Lean, Agile and DevOps are interconnected, Kaizen and 5S are not best suited to help support the success of the launch DevOps. Once DevOps has been introduced, Kaizen can be used for Continuous Improvement and 5S can be used to maintain good practices. However, both of these are after successful introduction of DevOps.
B) Incorrect. Feedback is always welcome, but this does not necessarily guarantee the most effective application of Lean when implementing DevOps.
C) Incorrect. Visualization can be helpful, but it is not the most impactful Lean practice when implementing DevOps.
D) Correct. Building a workable, single piece, deployment pipeline will help implement successful DevOps. The most important thing in DevOps is building up-stream processes from Development to Operations, specifically for a single deployment pipeline. JKK is the most effective working behavior to achieve this. *(Literature: C, Chapter 4)*

3 / 50
Compared to a project in a conventional manner, what must be changed for a project to be successful in DevOps?

A) An IT-services supply chain, using a pull system and one-piece-flow, should be built.
B) Developers should join the Operations team for quick maintenance of the services.
C) Operations must work for the Development team. That is why it is called DevOps.
D) Team members from the Operations team should join the Development team.

A) Correct. A project is a success when the process uses a Jidoka based pull system to create valuable IT services (or products). *(Literature: C, Chapter 4 and Literature B, Chapter 1)*
B) Incorrect. DevOps does not mean that a Developer joins Operations.
C) Incorrect. Just collaborating does not make the project a DevOps success.
D) Incorrect. DevOps does not just mean that Operations joins Development.

4 / 50
Any team that adopts DevOps, could be described as a Compact.

What principle applies **best** to make a Compact work?

A) Dynamic collaboration
B) Ongoing communication
C) Reducing cycles
D) Shared accountability

A) Incorrect. Dynamic collaboration is not a compact principle.
B) Correct. The principles of a DevOps compact are ongoing communication, shared, clearly defined goals and dynamic adjustment and repairs of understanding. *(Literature: A, Chapter 2)*
C) Incorrect. Reducing cycles is not a compact principle.
D) Incorrect. Shared accountability is not a compact principle.

5 / 50
To implement DevOps, there are many sources of knowledge, standards and practices available.

Which one is considered **key** for a successful DevOps implementation?

A) CMMI Level 3
B) Disciplined Agile
C) ISO 20001
D) PMI PMBok

A) Incorrect. CMMI certification could help but is not key for a DevOps implementation. Even less a specific CMMI Level.
B) Correct. Disciplined Agile is a key requirement for a successful DevOps implementation. *(Literature: C, Chapter 4i)*
C) Incorrect. ISO 20001 certification could help but is not key for a DevOps implementation.
D) Incorrect. PMI PMBok is a Project Management reference literature that is not key for a DevOps implementation.

6 / 50
Collaboration is one of the four pillars of Effective DevOps.

Why is it so important?

A) Collaboration is the only way to achieve one-piece-flow in DevOps.
B) DevOps business value is achieved through cooperation between teams.
C) DevOps organizations are small, the teams must collaborate well.
D) It drives change and is an easy win. It also reduces development costs.

A) Incorrect. You can achieve one-piece-flow without collaborating. It is a different mechanism.
B) Correct. Collaboration between all the teams involved (Development and Operations, included) is of key importance to achieve higher business value, through increased communication, automatization and higher quality software. *(Literature: A, chapter 6 and 7)*
C) Incorrect. DevOps organizations can be very large. Teams need to collaborate to add enough business value.
D) Incorrect. Tools drive change and are easy wins. Collaboration is hard. Collaboration can reduce development costs, but this is not the main goal. The goal is to increase quality.

7 / 50
You are the owner of a small DevOps company of 5 co-workers that builds mobile apps for children with disabilities. One of the apps your team is most proud of is an app for children with autism, that allows them to schedule their own day.

Because this app was so successful, your team is asked to extend the capabilities of the app for other people that would benefit from scheduling their day. This request will certainly generate more complexity in the code and some technical challenges that will have to be tackled by the team.

You expect to be paid very well for this assignment, so you accept the assignment. However, after a few weeks, your team is fighting. You get angry as well and start paying more attention to what all the team members do. You regularly work along with them, so you can quickly correct any code error you spot.

Although enough work gets done, your team stays angry and you recognize that part of the DevOps mindset is missing.

What is the **best** strategy to solve your problem?

A) Add members to the team seeking more diversity, through hiring complementary non-technical personnel
B) Ask experts in DevOps outside your company to sponsor and mentor your team members
C) Find a common goal and start working towards that goal together to increase collaboration
D) Reserve a few months time to start working on team-building first and learn about each other

A) Incorrect. Expanding the team may be advisable when there is no expertise to achieve what needs to be done. But hiring people with no technical duties is never advisable nor desirable to maintain the team efficient. However, in this case, enough work is getting done, so this is only going to increase any problems that you experience now, because there is no trust and no collaboration.
B) Incorrect. This is a great idea, but it will not solve the missing collaboration and affinity part of the DevOps mindset in your company. Collaboration is increased by sharing a goal and succeeding together.
C) Correct. This is a good way to solve the current problems. It should reduce the fighting, without taking a lot of time to solve the problem and potentially losing the assignment. Sharing goals builds trust, affinity and collaboration. *(Literature: A, Chapter 7)*
D) Incorrect. This may work and build a team that trusts each other and collaborates, which should be your goal. It is not the best way to solve the problem, though, because you risk losing the assignment by taking too much time to resolve the issue.

8 / 50
You feel that your Development team is really a team.

What is a sure sign that they are a team and not a group?

A) The team follows the rules they have agreed upon in their team meetings.
B) The team has effective meetings which they lead themselves.
C) The team keeps a steady working pace towards their common goal.
D) The team solves problems by questioning the responsible team member.

A) Incorrect. Groups of people can be very good in following rules. This does not necessarily make a team.
B) Incorrect. Groups of people can hold very effective meetings. This is not necessarily a sign of a team.
C) Correct. A true team ensures a steady working pace and will keep working towards their common goal. *(Literature: A, Chapter 9)*
D) Incorrect. Teams solve problems together and do not start questioning a team member. DevOps has a blame-free culture.

9 / 50
AppAtoZ has been growing at phenomenal rates in developing and deploying mobile applications to the iPhone and Android platforms.

The Development teams for this startup have experienced tremendous pressures in deploying rapid enhancements to their current mobile applications on an aggressive timeline. They have been working on average 60 hours per week over the last six months. Leadership has been reluctant to hire more assistance, and is more concerned about increasing revenue while reducing operational and development costs.

In the recent months there has been an increased rate of employee absenteeism, employees calling in sick, and some employees even resigned from AppAtoZ, resulting in increased workloads on current employees. Rehiring and ramp up time of a new employee does not alleviate the work pressures quickly for the Development team.

Which **long term** strategies must be considered to address employee burnout and stress?

A)
- Add a mix of permanent and contract employees to the team to better manage the workload
- Have leadership and Development identify factors in the work environment that contribute to burnout
- Develop a plan to address employee burnout and stress

B)
- Fire the leadership team, because they are incapable
- Hire a more competent leadership team that knows how to create a work-life balance and culture that is sustainable and realistic
- Provide the opportunity for the Development team to take time to find their balance

C)
- Have the Development team members take a time out
- Contract developers for managing the increased development workload at peak periods of the year
- Have developers reach out for professional mental help as needed

D)
- Have the Development team members take a time out
- Have leadership and development identify all factors in the work environment that might be contributing to burnout
- Develop a plan to address employee burnout and stress

A) Correct. This is the only answer where all choices are long-term. The rest of the answers have at least one short-term choice. *(Literature: A, Chapter 8)*

B) Incorrect. Firing the leadership team does not solve any underlying issues in the long term. It may be a fix in the short term. Taking time to find balance is a wise idea, but without a plan it may not spontaneously work.

C) Incorrect. Taking a time-out may be a great idea, but it only works in the short-term. Mental help is also great, but again, without underlying changes, a short-term solution. Contracting developers may work and is a great idea.

D) Incorrect. Developing a plan and identifying work-environment factors are great ideas. There is a better answer than this answer though, because taking a time-out works only short-term.

10 / 50
You work in a DevOps team as a developer. You try to facilitate collaboration between all of your team members. Your team has two male senior developers and one junior female systems administrator.

Your team members had a rocky start and could not get along. You intervened at the time, to try to help them get along better. That worked, because they stopped fighting and got a lot more work done.

Now you start noticing a pattern where the junior administrator keeps agreeing with the senior developers.

How should you react to this pattern?

A) As long as the team members are not fighting, you should leave the situation alone. They are just getting along and you should not disrupt this.
B) It is important that Operations voices any concerns with builds, so you should coach the administrator to stop accommodating and be more assertive.
C) The senior developers have more responsibility, so you should ask them to be nicer and to think of Operations concerns if the junior does not.
D) You should start working in the Operations team for now, even though you are a developer yourself, so you can show the junior administrator the way.

A) Incorrect. This situation needs conflict resolution. The junior team member uses accommodation as a conflict resolution style, and that is not a productive style. Each role has to perform its responsibility to avoid introducing risks of concerns not being addressed on time in the build.
B) Correct. This is the correct solution for this situation. This situation needs conflict resolution. The junior team member uses accommodation as a conflict resolution style, and that is not a productive style. In addition, it brings a risk of Operations not voicing any concerns with the build. *(Literature: A, Chapter 7 and 14)*
C) Incorrect. Everyone is just as responsible. This does not depend on the seniority or gender. In addition, developers cannot see problems with builds in the same way as operators can. This is why DevOps needs the interaction of the different areas.
D) Incorrect. You cannot just switch teams, even though showing the way is a great way to coach. Developers cannot see problems with builds in the same way as operators can, so it will harm the quality of the builds by doing this.

11 / 50
Why do Agile and Scrum promise faster and more predictable software development?

A) By allowing better and complete requirements gathering and handling prior to design
B) By allowing small autonomous, self-organizing and self-planning teams
C) By allowing the Product Owner to participate in daily standup meetings
D) By allowing the Project Manager to quickly change priorities as needed

A) Incorrect. This approach is more of a Waterfall model.
B) Correct. This is the way proposed by Scrum and Agile principles to be able to deliver faster and better. *(Literature: A, Chapter 4)*
C) Incorrect. Even though the Product Owner can participate, this is not relevant to deliver on the promise of faster and more predictable software development.
D) Incorrect. The Project Manager should not change priorities, it is the Product Owner who does.

12 / 50
What is light-weight ITSM?

A) A business-continuity focused ITSM
B) A new ITIL version proposed as standard
C) A poor implementation of ITIL processes
D) A release-management oriented ITSM

A) Correct. ITIL seems heavyweight and not suited for the quick processes of DevOps. Light-weight ITSM is ITSM realigned for DevOps focused on business continuity with a set of Minimum required information. *(Literature: C, Chapter 4iii)*
B) Incorrect. There is not such ITIL Version yet proposed.
C) Incorrect. Light-weight ITSM is not a poor implementation, rather a skimmed version, focused on business continuity and reducing management workload.
D) Incorrect. ITSM is oriented to Service Management, not Release management. Within the ITSM concept, Release is a process that underpins the Service.

13 / 50
You work for a company that has implemented Lean and Agile practices. Your CEO is not convinced of the value of implementing yet another new thing: DevOps.

You are a DevOps expert. You feel that DevOps is beneficial to any company that develops software.

What does DevOps add to the company you work for?

A) Creating better User Stories and functional requirements
B) Releasing less often, so customers do not have to suffer regular updates
C) Training professionals to do both Development and Operations
D) Writing code that is released faster supporting business outcomes

A) Incorrect. Agile practices already have you writing great User Stories and functional requirements. DevOps adheres to Agile principles as its foundation. Even with the addition of early operations engagement, the user stories and functional requirements created by following Agile will not be much changed within DevOps.
B) Incorrect. DevOps uses faster, continuous releasing, to bring value to the business faster. This is also a Lean principle. Customers updating often is not a problem, if the new features really add value. In addition, you can release often and still only have the end-users update on scheduled times.
C) Incorrect. The goal of DevOps is that Operations and Development work together, not as individuals fulfilling both roles. Even though this is theoretically possible in very small companies, it is not DevOps per se.
D) Correct. Lean and Agile practices already ensure focus on business value and changing feature requests. DevOps adds the increased release to production frequency in direct support of business outcomes, by focusing on creating a Continuous Deployment Pipeline with early Operations involvement and as much automation as possible. *(Literature: C, Chapter 1)*

14 / 50
Senior business management is requesting better business support and alignment to business objectives from the IT department. As a CIO, among other measures, you decide to reduce management workload in Operations.

How will DevOps **best** help you to improve your existing Service Level Management?

A) By abandoning ITSM best practices as they are just a heavyweight approach
B) By designing better Operating Level Agreements (OLA) between Dev and Ops
C) By implementing a new set of ITSM processes based on ITIL version 3
D) By realigning ITSM to use a set of minimum required information (MRI)

A) Incorrect. Abandoning ITSM best practices is not something that will improve SLM.
B) Incorrect. Designing better OLA between Dev and Ops will not help to reduce the workload, it will rather create more bureaucracy. It is certainly not a focus of DevOps.
C) Incorrect. This will create more Heavyweight processes unfitted for DevOps.
D) Correct. This is a key requirement for a DevOps implementation, to produce a light-weight ITSM. *(Literature: C, Chapter 4iii)*

15 / 50
Which DevOps implementation is **most** suitable for an enterprise that uses the System of Record (SoR) approach?

A) Collaboration
B) Continuous Delivery
C) Toyota Way

A) Correct. This focuses on just providing quick and frequent IT Services and reliable operation, most suited for SoE and SoR. *(Literature: C - Chapter 8)*
B) Incorrect. Most suitable for Digital product vendors.
C) Incorrect. Most suitable for IT Service providers.

16 / 50
What is the **main** benefit of using the Obeya system?

A) Facilitates customer complaints, to ensure that the team gets enough feedback to continuously improve
B) Handles stress within teams, so that team members can keep up a sustainable pace
C) Improves daily bug reporting, which ensures less rework and reduces passing bugs to other workstations
D) Quick decision making, based upon current status, by speedy information gathering and sharing

A) Incorrect. Obeya does not facilitates customer complaints.
B) Incorrect. Even though it could help to handle stressful situations, this is not a main benefit.
C) Incorrect. Obeya does not help to improve bug reporting.
D) Correct. Obeya room or War room is a Toyota Production System / Lean tool that helps project teams to have all relevant information available at sight and also promotes quick interactions and information sharing among small teams, speeding the information gathering for decision making. *(Literature: C, Chapter 7iii)*

17 / 50
You work in a software company that is just starting to work with DevOps. You realize that not only the culture of the organization should change, but also its practices and tools. The company does already monitor the software projects for any issues.

You propose to add the following measures:
- automate the Change Management process
- implement access controls to prevent anyone for making changes without approval

Why is this necessary?

A) Automation enables faster change implementations while maintaining confidence. Access control is necessary to prevent problem-solving heuristics and unplanned service disruptions.
B) Automation is necessary to slow down the number of changes. Access control should prevent customers from changing the software by themselves, without your knowledge and control.
C) Automation will help to make Operations unnecessary. Access control must be implemented, because DevOps projects work in a Cloud environment, which has higher risks involved.

A) Correct. Quote: "While in general we are not fans of locking things down and establishing approval processes, when it comes to your production infrastructure it is essential. As a corollary of that, since we believe that you should treat your testing environments the same way you treat your production environments, the same process should apply to both. It is essential to lock down the production environments to prevent unauthorized access not only from people outside your organization, but also from people within it—even operations staff. Otherwise it is just too tempting, when something goes wrong, to log into the environment in question and poke around to resolve problems (a process sometimes politely called a problem-solving heuristic). This is almost always a terrible idea for two reasons. First, it usually leads to service disruptions (people tend to try rebooting or applying service packs at random). Second, if something goes wrong later, there is no record of who did what when, which means it's impossible to work out the cause of whatever problem you're facing. In this situation, you may as well re-create the environment from scratch so it is in a known state." *(Literature: B, Chapter 11)*
B) Incorrect. Automation helps speed up the number of changes you can make, without losing confidence in the process. Access control may keep customers out, but this is not the main goal.
C) Incorrect. Operations is never unnecessary. They may feel less annoyed and burdened, though. DevOps projects may or may not be in a Cloud environment. Even though access control helps secure software in a Cloud environment, this is not the main goal.

18 / 50
Virtualization and Cloud computing are techniques that could help and facilitate DevOps practices.

How do they help DevOps?

A) Cloud computing makes remote access possible, which gives customers more control.
B) Virtual environments are easier to standardize and use hardware more efficiently.
C) Virtualized infrastructure is easier to understand and needs zero maintenance.

A) Incorrect. Cloud computing makes remote access easier, certainly, but that on its own does not means customers will have more control at all. Both things do not help DevOps in any way.
B) Correct. Virtualization makes it easy to consolidate CI and testing infrastructure so it can be offered as a service to delivery teams. It is also more efficient in terms of hardware usage. Virtualization allows you to standardize on a single hardware configuration for physical environments but run a variety of heterogeneous environments and platforms virtually. *(Literature: B, Chapter 11)*
C) Incorrect. Virtualized infrastructure is not easier to understand per se. In addition, it needs maintenance and management as well. Not the same as on premise infrastructure though. So this is not a valid reason.

19 / 50
Service Level Agreements (SLAs) are important for every project, because they specify what you agree on with the customer. However, in DevOps, the SLAs serve another important purpose.

What is this purpose?

A) Customers are responsible to create the SLA for the DevOps team. Therefore, it replaces a formal task as an Operations responsibility.
B) In the SLA, the customer can specify all the non-functional requirements they have, so Development can focus on those.
C) The SLA specifies the acceptable Service Level. Development should understand the SLA and support Operations to maintain it.

A) Incorrect. The SLA is always a contract between the customer and the area that provides the service, so both sides have something to say about the content.
B) Incorrect. Some of the non-functional and functional requirements may come from the SLA, but the most are given directly to the DevOps team, not via the SLA. Also providing non-functional requirements is not the purpose of the SLA by itself.
C) Correct. Usually, the terms described in the SLA are most relevant for Operations. Development should support Operations, by making their job as easy as possible. This is what makes DevOps different than regular development. *(Literature: B, Chapter 12)*

20 / 50
For a new product, your team needs to develop a Deployment Pipeline. As part of Continuous Integration, you need to define the Commit stage of the pipeline. You discuss this stage with your team members.

The Process Master says: "The Definition of Done should be defined during or before the Commit stage. When code is not Done when it is committed, the work should be stopped".

Is this true?

A) Yes. If the work is not Done, the Process Master is not doing a good job. This should be solved immediately.
B) Yes. Work that is not Done should not be committed, because it does not add customer value.
C) No. The Definition of Done is only defined during customer meetings. Waiting for it would slow work too much.
D) No. Work in a Deployment Pipeline should always continue. If code is not Done, it just needs to be inactive.

A) Incorrect. The Process Master has a job to ensure that there is a Definition of Done and when code is committed that is not Done, work should be stopped. However, the Process Master is not necessarily doing a bad job when code is committed that is not Done.
B) Correct. When work is not Done, there is not enough value for the customer to start it in the Deployment Pipeline. Considering one-piece-flow, this would delay the flow of more valuable work. *(Literature: B, Chapter 3)*
C) Incorrect. Definition of Done is one of the first things that is agreed upon in a project. It is not defined during customer meetings. When starting coding, we should already know a Definition of Done. Otherwise, how would you know when to stop coding?
D) Incorrect. When there is something wrong with the code, or it does not add value, this is enough reason to stop the Deployment Pipeline and get it fixed, or get something more valuable in the one-piece-flow pipeline.

21 / 50
What is the **main** benefit of increasing the diversity of the team to include a wider range of personal backgrounds and cultures?

A) It brings a greater number of experiences and points of view.
B) It leads to decreased friction amongst the team.
C) It limits originality and ability to come up with new insights.
D) It takes longer to come to a specific decision point.

A) Correct. Diversity includes a wide range of backgrounds that includes aspects of race, gender, sexuality, class, education level, language, and amount work experience. All of these unique aspects bring to an organization a greater number of experiences and points of view to the table. *(Literature: A, Chapter 7)*
B) Incorrect. There is a possibility that increasing the diversity might even lead to increased pressure and friction, because different cultural values have to work together.
C) Incorrect. More diversity means more different viewpoints. Usually, this leads to more originality.
D) Incorrect. This is considered a disadvantage, usually. (Although slower decision processes could be beneficial.) More diversity might lead to taking longer to reach consensus.

22 / 50
You work for a small company that has a single DevOps team. Your DevOps team works on an application that consists of multiple components. Some are new and some just need updates.

At the moment, each component has its own Deployment Pipeline. The team is proud of the Continuous Delivery they do within most of the Deployment Pipelines and their production is high and of good quality.

What should you do?

A) Keep the different pipelines and encourage the team to expand Continuous Delivery
B) Only keep pipelines that have Continuous Delivery and merge the other pipelines without Continuous Delivery
C) Only keep two different pipelines: one for development and one for maintenance
D) Work towards a single pipeline with your team, by explaining the risks of multiple pipelines

A) Incorrect. Having more than one Deployment Pipeline per team has risks. It is difficult to prioritize the different pipelines and work on some items will have to be stopped if something in another pipeline gets priority. That brings back a certain level of chaos and multi-tasking that DevOps tries to avoid. Efficiency and business value is increased using a single Deployment Pipeline.
B) Incorrect. Any Deployment Pipeline greatly benefits from Continuous Delivery. However, the existence of Continuous Delivery practices do not reduce the risks of multiple Deployment Pipelines.
C) Incorrect. Even if you just have two Deployment Pipelines, and even if they are split functionally, the same risks apply.
D) Correct. This is the way to go. Hopefully, your team has learned enough from their practice with Continuous Delivery to build this into the entire pipeline. *(Literature: B, Chapter 13)*

23 / 50
You are assessing Company Builders, which is a medium to large organization that has adopted DevOps practices a couple of years ago.

They have hired you to determine their current maturity state. When you are done, you should give suggestions for improvement. They want to know on which area they should focus to reach the next maturity level: *Level 2 - Quantitatively Managed.*

You find that most areas are within *Level 1 - Consistent*, with two exceptions:

1. **Environments and Deployment.** This area manages orchestrated deployments and has tested release and rollback processes.
2. **Build Management and Continuous Integration.** In this area, you find regular automated builds and testing, and any build can be re-created from source control using an automated process.

First determine the level of maturity in these two areas, based on the information given. Then give your recommendation for the focus of improvement.

Which of these two areas should Company Builders work on, before progressing to Level 2?

A) Environments and Deployment, and Build Management and Continuous Integration are both at level 0. The work should be done on both environments at the same time.
B) Environments and Deployment, and Build Management and Continuous Integration are either at level 1 or above. The work should be done in the other areas to progress.
C) Environments and Deployment is at level 0. Build Management and Continuous Integration is at level 1. The focus should be on Environments and Deployment first.
D) Environments and Deployment is at level 2. Build Management and Continuous Integration is at level 0. The focus should be on Build Management and Continuous Integration only.

A) Incorrect. Area 1 is at level 2 and Area 2 is at Level 0.
B) Incorrect. Area 1 is at level 2 and Area 2 is at Level 0.
C) Incorrect. Environments and Deployment is already at level 2 and does not need more work.
D) Correct. Area 2 is at Level 0 and it should be first matured to Level 1 in order for the organization to aim for Level 2. *(Literature: B, Chapter 15)*

24 / 50
Your company sells an online back-up service to its customers. Now, one of the customers has asked to implement new features in your service. They want the new features within a week, or they will seek business elsewhere.

You believe that the new features are important and you know the Development team can build it quickly. However, you encounter the following problems:

- It takes a long time for bugs to be closed by testers.
- Testers are finding bugs that developers fixed a long time ago.
- The application can rarely be demonstrated to be working.
- Showcases rarely happen.

What is your problem and how should you solve it?

A) You deploy buggy codes. You should solve this by increasing the understanding of the deployment process, increasing the collaboration and by working in a more disciplined manner.
B) You have poor Configuration Management. You should solve this by increasing the collaboration between Development and Operations, by increasing monitoring and logging as well as virtualization.
C) Your Continuous Integration process is not managed properly. You should solve this by speeding up automated tests and the Commit stage, and increase the understanding of the Continuous Integration process.
D) Your testing strategy is not effective. You should solve this by automating tests and increasing the collaboration between testers and the rest of the team.

A) Correct. Deploying buggy code or not deploying enough leads to long deployment times, low velocity, skepticism about release dates, loss of trust in the Continuous Integration environment, extended time to fix bugs, finding bugs that developers fixed a long time ago, and few demonstrations and showcases. The solution given solves these problems. *(Literature: B, Chapter 15)*
B) Incorrect. Poor Configuration Management specifically leads to unexplained failures in production, unmanageable deployment events, more time for environment configuration and a long recovery time in the event of failure. The solution given solves these problems.
C) Incorrect. Not managing the Continuous Integration process properly specifically leads to fewer than once-a-day deployments, broken commit stage, and long integration phase between releases. The solution given solves these problems.
D) Incorrect. A non-effective testing strategy leads to recurring bugs, much time spent fixing bugs, many complaints from customers, a low quality product and stressed developers. The solution given solves these problems.

25 / 50
Which is a benefit of Continuous Integration within Effective DevOps?

A) Extensive testing cycles before feature releases
B) Long periods of time between feature releases
C) More frequent and timely feature releases

A) Incorrect. There is need for sound testing practices, but they do not have to be extensive cycles. In addition, this is not a benefit.
B) Incorrect. This is the opposite of what happens.
C) Correct. Continuous Integration helps to release quicker and more frequent, because integration is automated. *(Literature: B, Chapter 3)*

26 / 50
A multinational organization is having many challenges in merging commits from their distributed locations into a central repository based in Dallas, TX. The distributed locations include Mexico City, Paris, San Diego and England. There have been inconsistent manners on when the regions are running their commits and at times it is not clear if some tests failed or not.

There are four possible practices:

1. Do not check in on a broken build
2. Always run all commit tests locally before committing, or get your Continuous Integration server to do it for you
3. Wait for commit tests to pass before moving on
4. Do not comment out failing tests

Which of these are **most** applicable to make sure that the distributed locations address their current pain points?

A) 1 & 2
B) 1, 2 & 3
C) 2 & 3
D) 2, 3 & 4

A) Incorrect. 1 is not applicable here. 3 & 4 are also important.
B) Incorrect. 1 is not important.
C) Incorrect. 4 is also important.
D) Correct. All three of these practices are most applicable to the current scenario. There is no evidence they are checking in on broken builds, so this is not as applicable. *(Literature: B, Chapter 3)*

27 / 50
Consider the anatomy of a basic Deployment Pipeline.

Which stage asserts that the system works at the functional and non-functional level?

A) Automated acceptance test
B) Build and unit test
C) Manual acceptance test
D) Version Control

A) Correct. The automated acceptance test stage asserts that the system works at the functional and nonfunctional level, that behaviorally it meets the needs of its users and the specifications of the customer. *(Literature: B, Chapter 8)*
B) Incorrect. Build tests and unit tests ensure that the new piece of code is sound in itself. It does not check the integration with the existing build.
C) Incorrect. This could be a correct answer. However, in a functioning Deployment Pipeline we expect the acceptance tests to be automated.
D) Incorrect. Version Control is used to fix broken builds or problems and issues. It is not used to show that the system works at a functional or non-functional level.

28 / 50
It is a DevOps best practice to use the same process to deploy to every environment in which your application runs. This ensures that the build is tested effectively. You are using scripts to automate your build and deployment process.

What is the **best** way to do this?

A) Use one script for each environment and maintain them as part of the Version Control system
B) Use one specific script for each environment to address the differences between environments
C) Use the same scripts for each environment, taking manual parameters for specific configurations
D) Use the same scripts to deploy to each environment and manage configuration information separately

A) Incorrect. The amount of effort to maintain this and the potential errors you can introduce by adding this complexity makes this answer a bad choice.
B) Incorrect. Different scripts could create a problem with modifications thus generating problems within the process that are not easy to track and solve.
C) Incorrect. No manual interaction should be done when building and deploying, as this process should be automated, both for celerity and to make it error free.
D) Correct. Scripts should be the same to ensure both the build and the delivery process can be tested effectively. The differences between environments, such as URI, IP, etc., should be managed as part of the Configuration Management process. *(Literature: B, Chapter 6)*

29 / 50
Upon release of a new IT service, a job in operation ends unexpectedly.

What could **not** be a valid cause for this?

A) The Gatekeeper did not take the interrelation between work items and SAC into account.
B) The non-functional requirements were not clear, due to an imperfect User Story.
C) The Service Master did not agree with the user on the End of Life of the service to be released.
D) There was a lack of work in the development processes, as is supported by the Service Acceptance Criteria (SAC).

A) Incorrect. This is a possible cause.
B) Incorrect. This is a possible cause.
C) Correct. This is not a possible cause, because this cannot be determined beforehand and will always have to be determined once the service is live. *(Literature: C, Chapter 7)*
D) Incorrect. This is a possible cause.

30 / 50
Company AppBC is using DevOps. They have implemented Continuous Deployment and a solid Deployment Pipeline, with highly automated acceptance tests and they are delivering daily new software to production.

AppBC has a large database and many users. They have a comprehensive and solid capacity testing strategy in place. As their environment is quite large and complex, with each new version some bugs appear in production.

What strategy could **best** help AppBC prevent these bugs?

A) Adopt canary releasing
B) Automate capacity testing
C) Decrease the delivery rate
D) Use Blue-Green deployment

A) Correct. Canary releasing involves rolling out a new version of an application to a subset of the production servers to get fast feedback. This quickly uncovers any problems with the new version without impacting the majority of users by gradually ramping up the load, while measuring response times and other performance metrics, reducing the risk of releasing a new version and helping to find and fix the bugs quicker. *(Literature: B, Chapter 10)*
B) Incorrect. Within this context capacity testing should already be automated, but automation of these tests will not help to detect the bugs in this scenario.
C) Incorrect. This is against DevOps practices.
D) Incorrect. Blue-Green will require too much resources that will be very expensive in this scenario. Also using this strategy with a large database could produce down-times or read-only situations if rollback is needed. Also it will not help produce better capacity testing.

31 / 50
DevOps takes very important concepts from Agile, derived from the Toyota Production System.

Why is one-piece-flow important for a DevOps adoption?

A) It allows your team to work in a sustainable pace with predictable velocity.
B) It helps the team to focus on adding features with the most business value possible.
C) It increases the shared responsibility of work between the different team members.
D) It reduces bottlenecks by limiting the number of tasks you are doing simultaneously.

A) Incorrect. This is what rhythm does. By setting a rhythm that is sustainable, you can ensure that your pace stays predictable, your team does not burnout and that your employees have a good work-life balance.
B) Correct. This is what one-piece-flow achieves. One-piece-flow lets you pick the feature or update the feature that provides the most value and puts it next in the pipeline. This keeps you agile. By working on a single feature, you tend to be able to limit the Work-in-Progress so the features also actually get finished. *(Literature: C, Chapter 7)*
C) Incorrect. This is something that is important for DevOps, but is not directly related to Work in Progress, one-piece-flow, rhythm or Ji Koutei Kanketsu.
D) Incorrect. This is what Work-in-Progress is all about. By limiting the number of tasks you can work on simultaneously with your team, you actually successfully finish the assigned tasks. You avoid bottlenecks, because you regulate the Work-in-Progress limit so that there is always someone waiting for the finished work.

32 / 50
The S Corporation, a medium sized automobile parts supplier, supplies a large company, T Motors Corporation. They supply automobile parts to T Motors Corporation, which amounts to almost 60% of their total sales.

The board meets to discuss the new partnership. T Motors Corporation demands that S Corporation change their way of delivery to Just-in-Time delivery, or they will stop their business with S Corporation. S Corporation would not survive this loss of business, so there is a sense of urgency to change to Just-in-Time. This change must happen within 6 months, so there are 5 months at most to prepare.

One of the things to be implemented is tracking of parts through Radio Frequency ID's (RFID). This should help keep the production process transparent. A quick review of the current process is in order to facilitate the change to an RFID enabled process.

The CIO is asked to manage the change process. She believes this will be possible, if a DevOps approach is used to create a minimal release. Ideally, the concept for production with RFID should be developed first. As a last step, a production control system using the RFID data should be implemented. However, there is not enough time to do these steps sequentially. Therefore, these three things should be done concurrently.

The CIO assigns Em, who is a Scrum Master, to the project. Development prepares to build a Deployment Pipeline.

Em can see that Development is enthusiastic and works hard, but they could use more discipline. In addition, the release frequency needs to be higher.

What should Em focus on **first**?

A) Em should focus on communication, since it is the most important thing in DevOps. Em should start with breaking the ice with the team and setting some rules for communication.
B) Em should start with discussing a Value Stream Map and building one-piece-flow with the team, since flow and a streamlined process are very important.
C) Em should start with discussing the infrastructure and the working environment with the team members, since DevOps is most effective when all tools and practices work.
D) Em should start with gathering all stakeholders, educate them on DevOps and ask for their support in spreading the cultural change, since cultural change is required for DevOps.

A) Incorrect. Although communication is important for a functioning DevOps team, they have been working together before. There is lower hanging fruit to start with. More importantly, the switch to thinking in terms of value for the customer must be made. Once this way of thinking is in place, Em can switch over to working on fine-tuning communication skills.

B) Correct. The processes need to be streamlined and the Value Stream Map should be created, so that the team can start adding as much value with as little effort as possible. After this, it is time to define what could and should change and what already works great; tools, communication and culture included. DevOps does not have to look and feel the same in each company, but it does have to focus the team on adding value for the customer. *(Literature: B, Chapter 1 and A; Chapter 1 and 2)*

C) Incorrect. Tools and automation are important parts of DevOps and should not be forgotten. However, they are not necessarily the very first focus. Instead, it is wise to see if you can make the whole production or service process easier, shorter and cheaper by creating a Value Stream Map and discussing it with the team. This will start the inevitable and necessary cycle of Continuous Improvement. All other steps should come secondary.

D) Incorrect. In many companies, a cultural change must be made. However, this is not necessarily the first thing to do. Neither should the stakeholders be gathered to discuss this change in culture. The cultural change should follow from how the most value is created for the customers or stakeholders. It would be a great idea to ask different stakeholders to have a look at the Value Stream Map and see what they can add.

33 / 50

Your DevOps team works well together at a sustainable pace. By building enough slack into the process, the team has the time and concentration to carefully check and test the builds. Currently, your team tests and deploys manually. Their pace is high enough to deliver high value to the business on a regular basis.

Your CEO has requested your advice on automation within this team.

Which advice should you give?

A) Automate as much as you can, so that the team can add more features and sooner demonstrate business value

B) Automate the acceptance tests, but not the deployments, because the manual process is more secure

C) Automate the deployments to improve cycle time, but not the tests, to allow learning from bugs

D) Do not add automation to the methods of this team, because the methods currently executed by the team are providing amazing results

A) Correct. Always automate what you can, so you gain more control over your process and sooner demonstrate more business value. *(Literature: B, Chapters 1 and 8)*

B) Incorrect. Automating the acceptance tests is a great idea, but manual deployments are not more secure.

C) Incorrect. Automating the releases is a great idea, but manual testing does not allow learning from bugs more than automated testing.

D) Incorrect. The team does amazing work, but in a way you are wasting its potential by not automating the activities that can be automated.

34 / 50
A CIO assigns her most reliable employee, Michael, who is a Scrum Master, to a project. The Development team prepares to build a Deployment Pipeline.

Michael has confidence in the good intentions and spontaneity of the development team, but would like them to become more disciplined. In addition, there should be a higher release frequency. Michael wants the Development team to implement more frequent releases.

One of the team members says: "The most important thing about this new Deployment Pipeline is automating it. We should first automate the Deployment Pipeline".

Is this statement correct?

A) Yes, this is correct. Automating the Deployment Pipeline is the most important factor for increasing the efficiency.
B) Yes, this is correct. By focusing on creating a Deployment Pipeline that is automated, you overcome potential problems that you may encounter later.
C) No, this is not correct. Achieving single-piece-flow and a solid deployment process should be the first priority. Automation of the process can come later.
D) No, this is not correct. Instead of automating the Deployment Pipeline, the testing process should be automated first.

A) Incorrect. The Deployment Pipeline should always be a single-piece-flow Deployment Pipeline first. This can work well without any automation. Once it is solidly in place, there is opportunity for automating the process where feasible. However, this should always be secondary to building a solid Deployment Pipeline.
B) Incorrect. The Deployment Pipeline should always be a single-piece-flow Deployment Pipeline first. This can work well without any automation. Once it is solidly in place, there is opportunity for automating the process where feasible. However, this should always be secondary to building a solid Deployment Pipeline.
C) Correct. The Deployment Pipeline should always be a single-piece-flow Deployment Pipeline first. This can work well without any automation. Once it is solidly in place, there is opportunity for automating the process where feasible. However, this should always be secondary to building a solid Deployment Pipeline. *(Literature: B, Chapter 5)*
D) Incorrect. Automating tests is a key activity. However, when faced with the option of the choice for creating a solid Deployment Pipeline and automating working tests, you should always focus first on creating a solid Deployment Pipeline. Once this is in place, there is opportunity to create increased efficiencies through test automation.

35 / 50
Your company is changing its ways and starting to work with DevOps. Your team is on board with this change. You are discussing best practices for the Commit stage of the code.

Your co-worker Sun says: "When a build breaks, and nobody takes responsibility, we should find out who did it and call them out on it, so that they can fix the build."

Is this a good idea?

A) Yes. Only the person who breaks a build can fix it, so you should identify them, even if this makes them uncomfortable.
B) Yes. You should always find the person responsible for breaking a build. If you do not, your co-workers may enforce this rule.
C) No. DevOps is a blame free environment. If a co-worker does not take responsibility, do not force them to.
D) No. You should fix the build first. Then make time to identify the person responsible and punish them for it.

A) Incorrect. It is probably easiest to let the person who creates a problem find the problem, but it is not necessary. DevOps is a blame free environment. If a co-worker does not take responsibility, do not force them to. Forcing anybody to do anything is not respectful.
B) Incorrect. DevOps is a blame free environment. If a co-worker does not take responsibility, do not force them to. Forcing anybody to do anything is not respectful.
C) Correct. Forcing anybody to do anything is not respectful. It is OK to make errors. The team members work in a collaborative manner to get through any errors or challenges. *(Literature: B, Chapter 3; and A, Chapter 4)*
D) Incorrect. The build does not have to be fixed; you can revert to a previous version. In addition, fixing the build might not be a bad idea, but punishing someone who made a mistake is a bad idea. DevOps is a blame free environment. If a co-worker does not take responsibility, do not force them to. Forcing anybody to do anything is not respectful.

36 / 50
The Development team at X-AppGo has been running into numerous challenges with their current testing practices. Currently, they use a manual acceptance testing process. The developers believe that the unit test suite that they have created is thorough enough to protect against regressions.

The Development team has to spend 1 million dollars on manual acceptance testing for every release. Senior leadership has mandated that the Development team should implement automated acceptance testing to reduce overall costs of testing and also minimize the number of code defects and regressions introduced into the production environment.

What principles must be followed when defining acceptance criteria for your application with automation in mind?

A) Agile principles
B) ATAM principles
C) INVEST principles

A) Incorrect. INVEST is the recommended set of principles to adopt when creating maintainable acceptance test suites. There is no specific info on why ATAM and Agile are not recommended. Agile does not provide any specific guidance or principles on test automation.
B) Incorrect. INVEST is the recommended set of principles to adopt when creating maintainable acceptance test suites. There is no specific info on why ATAM and Agile are not recommended.
C) Correct. Acceptance tests are derived from acceptance criteria, so the acceptance criteria for your application must be written with automation in mind and must follow the INVEST principles, which stands for independent, negotiable, valuable, estimable, small and testable. *(Literature: B, Chapter 8)*

37 / 50
What is the **most** effective mechanism for migrating data in an automated way?

A) Create a database versioning schema and keep it under Version Control
B) Create and manage smaller datasets, so that the migration becomes easier
C) Ensure all your scripts have been properly tested prior to migrating the data
D) Ensure you have a rollback procedure in place in case the migration fails

A) Correct. Versioning your database is the best mechanism to migrate data in an automated fashion. *(Literature: B, Chapter 12)*
B) Incorrect. This does not provide the best mechanism to support automated migrations, and is primarily focused on how to more effectively manage datasets.
C) Incorrect. This is not the best answer and really is more focused on testing activities versus migration activities.
D) Incorrect. This is focused on recovery actions to take if the migration fails.

38 / 50
Company X-AppGo has been having challenges with their roll-back processes. This often results in critical data losses within their production application databases, when executing roll-back scripts.

When is it **not** possible to run roll-back scripts without losing critical data?

A) The roll-back script deletes data that only the new version uses.
B) The roll-back script involves moving a column between tables.
C) The roll-back script will add data back from temporary tables.

A) Incorrect. The roll-back script is deleting only some data that only the new version uses and does not result in critical data loss when you roll-back.
B) Incorrect. The roll-back script that is modifying the database schema in a way that will not result in data loss.
C) Correct. This is a scenario where a roll-back script will not be possible. *(Literature: B, Chapter 12)*

39 / 50
ACMECONST has encountered many application and hardware failures after deploying application software upgrades and hardware refreshes to their routers and switches.

It has been very difficult for them to recover back to their original state, after having these failures during their maintenance windows. This has resulted in extended hours of recovery, beyond the normal maintenance windows, and extended downtime for their critical applications.

Automated provisioning and autonomic infrastructure can help within this situation, but some considerations apply.

Which items need to be managed carefully to reduce the risk of disruption when deploying to the production environment?

A) Detailed monitoring logs to troubleshoot application upgrade failures
B) External integration points, such as external systems and services
C) Server configurations and underlying user account information
D) The set of automation provisioning tools and autonomic architecture

A) Incorrect. Troubleshoot will happens after the upgrade so this is not a valid item to reduce risk of deployment to the production environment.
B) Correct. This item needs to be managed carefully to reduce the risk of deployment to any production-like environment. *(Literature: B, Chapter 11)*
C) Incorrect. This is covered within the information of the operating system and its configuration for both test and production environments, so it is not an item that you manage separately.
D) Incorrect. It is not important to manage the tools itself to reduce risk. They help as a providing the proper process implementation and context to produce automated provisioning and autonomic infrastructure.

40 / 50

Company X-AppGo has been having challenges in their core application. The application is not properly interfacing with other external applications. These external applications need to effectively obtain specific data variables, so that specific calls can be executed. The core application is being developed by one team, and the company wants to maintain that for very good business reasons.

One of the developers suggests to separate out a component from the X-AppGo codebase to tackle the interfacing issues.

What are good reasons to separate out a component in this case?

A) Convert a set of plugins in the codebase into a monolithic codebase
B) Limit the impact of changes and make changing the codebase easier
C) The X-AppGo codebase will have to be split and managed by different teams
D) There are no good reasons and this will require more time to compile

A) Incorrect. Creating a component is going from a monolithic codebase to a modular one based on components, so this is actually the opposite.
B) Correct. They encourage us to design and maintain software with clear delineation of responsibilities, which in turn limit the impact of change, and makes understanding and changing the codebase easier *(Literature: B, Chapter 13)*
C) Incorrect. Since this is not separating the X-AppGo application into independent components, there is no need to challenge business reasons and split the team.
D) Incorrect. There are several good reasons and option 2 is one. Also creating a modularized codebase will require less time to compile and link the code not more.

41 / 50
Even the smallest applications will have a dependency on other components or libraries. Therefore, understanding and managing dependencies is a key activity within Continuous Deployment in order to keep flow within the Deployment Pipeline.

You have built an application that uses two libraries. Each of these libraries rely on a third, underlying library, although they refer to different versions. This creates a specific dependency.

What is the **best** solution to solve or prevent this dependency?

A) Assemble all libraries into a single library, so that you can refer to the library directly and prevent the problem
B) Manage the libraries by using Version Control, so that you see it directly if you create this type of dependency
C) Keep a visual overview of all your dependencies on sticky notes on a big board, so you can track the flow
D) Only check in small parts of your toolchain, so that you can easily debug problems that may occur upon check-in

A) Incorrect. This is not a good idea. Instead, use version control to refer to the latest library, or use an automation tool that can help you automate using the latest version of libraries.
B) Correct. This is a good solution. In addition, Version Control on your libraries helps you debug problems from users running old versions of your software. Another good solution is to use an automation tool to help you use the latest version of libraries. *(Literature: B, Chapter 13)*
C) Incorrect. This may help track flow, but is not a solution for dealing with dependencies. Kanban boards do not really help you in implementing Version Control.
D) Incorrect. Instead, you should check-in the full toolchain. This helps you find interdependencies or incompatibilities much faster and more reliably.

42 / 50
Within a Continuous Deployment environment, it is important that everything is version controlled, so that you can find errors quickly, or roll back whenever necessary.

However, it is **not** recommended to keep binary output within Version Control.

Why is this exception made?

A) Binary output tends to be in large files that change with every build and are updated automatically.
B) Multiple team members work on the binary files, so it is not practical to keep this in version control.
C) The binary output is the input for your compilers, which are already kept in version control.
D) There is no need to do this, since recompilation is done as a regular part of the normal build process.

A) Correct. First, the output is very big and recreated for every check-in that is compiled and passes the automated tests. Second, they can be recreated from source code by rerunning the build script. *(Literature: B, Chapter 2)*
B) Incorrect. This is not the reason.
C) Incorrect. Binary output is the output of compilers, not the input. Rest of the reasoning is correct.
D) Incorrect. It is not advisable to do recompilation as a regular part of the normal build process. It is correct that recompilation would give you a new binary output.

43 / 50
You want to take a holistic approach to managing all of your IT infrastructure.

On which two principles can this approach be best **best**?

A)
1. The desired state of your infrastructure should be specified through change-controlled configuration.
2. You should always know the actual state of your infrastructure through monitoring and Event Management.

B)
1. The desired state of your infrastructure should be specified through change-controlled configuration.
2. You should always know the actual state of your infrastructure through instrumentation and Incident Management

C)
1. The desired state of your infrastructure should be specified through version-controlled configuration.
2. You should always know the actual state through current Incident and Event Management

D)
1. The desired state of your infrastructure should be specified through version-controlled configuration.
2. You should always know the actual state of your infrastructure through instrumentation and monitoring.

A) Incorrect. The desired state of your infrastructure should be specified through version-controlled configuration and not change-controlled configuration. Also point 2 is incorrect: event management is not correct and the principle of instrumentation is missing.
B) Incorrect. The desired state of your infrastructure should be specified through version-controlled configuration and not change-controlled configuration. Also point 2 is incorrect: incident management is not correct and the principle of monitoring is missing.
C) Incorrect. Point 1 is correct. Point 2 is not correct: it is not one of the principles on which a holistic approach can be based best.
D) Correct. These are two of the principles on which performing a holistic approach to managing all infrastructure can be based best. *(Literature: B, Chapter 11)*

44 / 50
Teams with good collaboration practices have synchronized work tickets. A CTO used 'Go and See', to investigate how the Operations team functions. After releasing, the Operations team always redefines the Operational infrastructure.

What is the **best** advice to improve upon this practice?

A) They should do nothing. There is no improvement possible, because the redefinition step should always be done.
B) They should examine a way to model the operational infrastructure and access control of the operational environment.
C) They should review the operational infrastructure, so that that becomes an automated process.
D) They should start to share their knowledge of the deployment process with the Development team.

A) Incorrect. This is unnecessary work and can be improved upon.
B) Incorrect. Although this may be nice, they will keep on repeating it, which is Waste.
C) Incorrect. There is no clear way to start automating this process without involving Development.
D) Correct. This is the way to go: sharing knowledge and then taking further steps. *(Literature: A, Chapter 17)*

45 / 50
When is a good time for Operations to inform Development of an operational change?

A) Development does not have to be informed. Operational changes are for the Operations team only.
B) Immediately. Development must be informed as soon as possible.
C) In the Scrum of the Scrum's meeting the next morning.
D) When the Operations team has done the acceptance testing.

A) Incorrect. Development must be informed immediately, so they can foresee possible risks and problems.
B) Correct. Development must be informed immediately, so they can foresee possible risks and problems. *(Literature: C, Chapter 5 and 7)*
C) Incorrect. Development must be informed immediately, so they can foresee possible risks and problems.
D) Incorrect. Development must be informed immediately, so they can foresee possible risks and problems.

46 / 50
You want your DevOps organization to mature. There are many ways to do this.

What is **not** a way to help your DevOps organization mature?

A) Clearly define targets as milestones to help your team members judge if their daily activities are valuable
B) Define processes clearly and support and enable the team members to improve the process daily
C) Keep recordings of all meetings so that your team members have easy access to all communication
D) Monitor and record daily activities to help identify small areas of day-to-day progress and celebrate them

A) Incorrect. This is helpful to help the DevOps organization mature.
B) Incorrect. This is helpful to help the DevOps organization mature.
C) Correct. This does not help your DevOps organization to mature. There is no strict need to write down all records of a meeting and revisit it. There is a need for writing down agreements, but not entire meetings. *(Literature: B, Chapter 15)*
D) Incorrect. This is helpful to help the DevOps organization mature.

47 / 50
You work for an IT service provider. As a part of your business continuity plan, you want to guarantee that you can always meet the minimum agreed service levels.

You want to ensure IT service continuity.

How can DevOps help you with IT Service Continuity Management?

A) DevOps cultural values, affinity and collaboration, make sure that service is highly valued by the DevOps team members.
B) DevOps prepares the team's disaster routines and Obeya practices by deliberately introducing chaos in the system.
C) Risk reduction measures and recovery options are likely coded in, because Operations is working together with Development.
D) Service Level Management becomes more important in DevOps, because the Process Master's task is to monitor this.

A) Incorrect. The cultural values help people to feel better, work better, at a steady pace, make mistakes and learn from them. In itself it does not help Service Level Management.
B) Incorrect. Introducing chaos, as Netflix's Chaos Monkey does, might help you to start coding in risk reduction measures and recovery options, but in itself it does not help the Service Level Management. Furthermore, chaos in the system should be solved by coding, not by war-room style solutions.
C) Correct. It is a systematic process to prevent, predict and manage infrastructure and application risks that if not addressed could lead to disruptions and incidents, which have the potential to disrupt services. *(Literature: B, Chapter 11; and literature C, Chapter 4)*
D) Incorrect. The Process Master does not have a priority task to keep Service Level Management in check. At the very least, this is not a priority in every DevOps team.

48 / 50
ACMECONST has aggressively expanded its global presence by increasing the number of hires and engineering teams distributed throughout the world. It also has been increasing its customer base at a dramatic pace of 30% per year.

Decisions that were once easily made when the engineering team was in one room are now taking much longer, causing frustration across the organization. There are more layers of management approvals to go through and the process is more extensive, which is causing many of the engineers to get disenchanted with the entire decision making process.

There is also increased confusion around the ownership of the various problems that are presented, causing a hesitation on making decisions. The engineers also feel their creativity has been stifled by the additional processes and bureaucracy, which has started to impact their morale.

What is the **best** way to address this scenario?

A) Keep current processes, but establish clear roles, accountability and ownership for each process, establish an effective method for weighing productivity versus risk, make incremental changes and create safe places for experiments
B) Re-examine processes to identify where things can be streamlined and establish clear roles, accountability and ownership for each process, establish an effective method for weighing productivity versus risk, make incremental changes and create safe places for experiments
C) Re-examine processes to identify where things can be streamlined and establish clear roles, accountability and ownership for each process, establish an effective method for weighing productivity versus risk, make incremental changes and minimize the amount of experimentation to prevent unnecessary application failures

A) Incorrect. Keeping current processes is not the best idea: there has been growth and the current processes are not working anymore. Creating a safe place for experiments is a good idea, though.
B) Correct. These actions work well with a growing company. Re-examining the processes is necessary, because the old processes do not work anymore. The safe place for experiments is also necessary to facilitate Continuous Improvement. *(Literature: A, Chapter 15)*
C) Incorrect. Re-examining the processes is necessary. However, minimizing the amount of experimentation is a bad idea: this suffocates the Continuous Improvement that needs to happen, by not allowing for experiments.

49 / 50

Within company X-AppGo there is a conflict between the Operations team in Colombia and the Development team in Ireland, caused by the fact that they have different priorities and goals. Due to this conflict, the amount of time and effort it takes to resolve issues that impact the business is increasing.

Which key practices should X-AppGo consider in order to reduce conflict and improve collaboration between the Development and Operations teams?

A)
1. Allow Development and Operations teams to work separately from each other, if they prefer, to avoid conflicts.
2. Obtain complete executive board buy-in on supporting the Development and Operations teams.

B)
1. Get a sponsor from the executive board of the company to talk to the DevOps team about the importance of working together.
2. Train the Development and Operations teams in DevOps practices, so they learn to do each other's work.

C)
1. Make sure that the Development and Operations teams visit other companies where DevOps works well.
2. Increase funding to better support the increased demands that both the Operations and Development teams are facing.

D)
1. Recommend site visits between the Development and Operations teams to build rapport, develop trust and understanding.
2. Spread knowledge between Development and Operations teams so they work together more effectively.

A) Incorrect. The teams should be encouraged to work together, not be allowed to work apart. They need each other and they need to learn from each other. Collaboration does not happen when you do not see each other. Buy-in from the executive board does not really help the teams themselves to work with each other.

B) Incorrect. Although teams may feel appreciated more by a sponsor from the executive board, it does not help their collaboration with each other. Training only goes so far. In order to help the teams work together better, they just need to, well, work together.

C) Incorrect. Visiting other companies can be inspirational, but DevOps is highly unique to the setting of the company. This does not directly help with collaboration and conflict reduction. Instead, the teams should work together and share knowledge. Increasing funding may help a little when there is too much work to do for too few people, but it only goes so far in reducing conflicts and increasing collaboration.

D) Correct. These are the most appropriate practices to best support the current scenario of reducing conflict and improving collaboration between the Operations and Development teams. *(Literature: A, Chapter 15)*

50 / 50
A Development team is interested in DevOps. They are mainly interested in Continuous Integration (CI). They currently develop and maintain 3 major solutions and 4 smaller ones. They use Scrum practices. Each Sprint takes 4 weeks, creating an average of 1 committed release to the test environment each 10 or 15 days and 1 release to production per month. They want to create a qualitative business case for their management to support their investment and effort to create a CI practice.

Which tangible benefits of CI help that business case **most**?

A) Deploying to test environment once per day could increase business benefits and greatly decrease development costs.
B) It helps the team spirit. As they are already using Scrum, CI will **not** generate measurable benefits for the business.
C) It increases release stability and quality with better and automated testing, facilitating and increasing the overall release speed.
D) Releasing to production once per day could increase business benefits and greatly decrease development costs.

A) Incorrect. Deploying to test environment faster is OK and a consequence of CI but it won't create any business benefits.
B) Incorrect. CI can help them to deliver faster to production, finding bugs sooner with less cost, whether they use Scrum or not is irrelevant.
C) Correct. Increasing release speed will be a consequence from Continuous Integration, as each change will be functional and integrated in the main code, ready to be delivered to production, as the product will always be in working state. Also it will increase release stability and quality out of finding and fixing bugs sooner by automated testing. *(Literature: B, Chapter 3)*
D) Incorrect. Faster release to production is one of the main benefits of Continuous Delivery, not the direct benefit of Continuous Integration, which aims to delivery to a production-like environment to assure the release is in working state and automated tests have been passed.

Evaluation

The table below shows the correct answers to the questions in this set of sample questions.

Question	Answer Key	Question	Answer Key
1	B	**26**	D
2	D	**27**	A
3	A	**28**	D
4	B	**29**	C
5	B	**30**	A
6	B	**31**	B
7	C	**32**	B
8	C	**33**	A
9	A	**34**	C
10	B	**35**	C
11	B	**36**	C
12	A	**37**	A
13	D	**38**	C
14	D	**39**	B
15	A	**40**	B
16	D	**41**	B
17	A	**42**	A
18	B	**43**	D
19	C	**44**	D
20	B	**45**	B
21	A	**46**	C
22	D	**47**	C
23	D	**48**	B
24	A	**49**	D
25	C	**50**	C

Contact EXIN

www.exin.com

YOUR ICT COMPETENCE PARTNER