Preparation Guide

Edition 202306

Content

1. Overview

EXIN DevOps Master™ (DEVOPSM.EN)

Scope

EXIN DevOps Master™ is a certification that validates a professional's knowledge about:

- implementing continuous delivery
- designing a DevOps architecture
- improving product and process
- using Lean management and monitoring
- implementing cultural change
- assessing maturity

Summary

The word DevOps is a contraction of 'Development' and 'Operations'. DevOps is a set of best practices that emphasizes the collaboration and communication of IT-professionals (developers, operators, and support staff) in the lifecycle of applications and services, leading to:

- continuous integration: merge developed working copies to a shared mainline several times a day
- continuous deployment: release continuously or as often as possible
- continuous feedback: seek feedback from stakeholders during all lifecycle stages

DevOps changes how individuals think about their work. DevOps values diversity, supports processes that accelerate the rate by which businesses realize value, and measures the effect of social and technical change. DevOps is a way of thinking and working that enables individuals and organizations to develop and maintain sustainable work practices.

EXIN DevOps Master™ focuses on introducing DevOps with respect for the individuals involved and an eye for customer needs. DevOps actively promotes a blame-free culture of sharing stories and developing empathy, enabling people and teams to practice their crafts in effective and lasting ways. The ultimate driver behind DevOps is providing applications and services that bring value for the business. It reaches this goal by managing the complete lifecycle of applications and services through creating flow in the value stream, continuous feedback loops, and automation.

This certification focuses on adding practical skills to knowledge, enabling a DevOps Master™ to facilitate DevOps successfully in teams and to promote its principles in the organization.

Context

The EXIN DevOps Master™ certification is part of the EXIN DevOps qualification program.

Target group

DevOps is best known in the field of software development, but its principles are applicable in IT service projects and other projects as well. The EXIN DevOps Master™ certification is aimed at professionals who are interested in facilitating DevOps practices in the organization.

The EXIN DevOps Master™ certification is meant for anyone working within a DevOps context or in an organization that considers the transition to a DevOps way of working. The target group includes business analysts, developers, security experts, Product Owners, process managers, and members of DevOps teams.

Requirements for certification

- Successful completion of the EXIN DevOps Master™ exam.
- Accredited EXIN DevOps Master™ training, including completion of the Practical Assignments.

Examination details

Examination type:	Multiple-choice questions
Number of questions:	40
Pass mark:	65% (26/40 questions)
Open book:	No
Notes:	No
Electronic equipment/aides permitted:	No
Exam duration:	90 minutes

The Rules and Regulations for EXIN's examinations apply to this exam.

Bloom level

The EXIN DevOps Master™ certification tests candidates at Bloom level 3 and 4 according to Bloom's Revised Taxonomy:

- Bloom level 3: Application - shows that candidates have the ability to make use of information in a context different from the one in which it was learned. This type of questions aims to demonstrate that the candidate is able to solve problems in new situations by applying acquired knowledge, facts, techniques and rules in a different, or new way. These questions usually contain a short scenario.
- Bloom level 4: Analysis - shows that candidates have the ability to break learned information down into its parts to understand it. This Bloom level is mainly tested in the Practical Assignments. The Practical Assignments aim to demonstrate that the candidate is able to examine and break information into parts by identifying motives or causes, make inferences and find evidence to support generalizations.

Training

Contact hours

The recommended number of contact hours for this training course is 21. This includes practical assignments, exam preparation and short breaks. This number of hours does not include lunch breaks, homework, and the exam.

Indication study effort

112 hours (4 ECTS), depending on existing knowledge.

Training organization

You can find a list of our Accredited Training Organizations at www.exin.com.

2. Exam requirements

The exam requirements are specified in the exam specifications. The following table lists the topics of the module (exam requirements) and the subtopics (exam specifications).

Exam requirements	Exam specifications	Weight
1. Implementing continuous delivery		**32.5%**
	1.1 Implementing version control	5%
	1.2 Automating deployment	7.5%
	1.3 Establishing continuous integration	2.5%
	1.4 Introducing test automation	5%
	1.5 Managing data and data security (DevSecOps)	5%
	1.6 Establishing continuous delivery	5%
	1.7 Trunk-based development	2.5%
2. Designing a DevOps architecture		**5%**
	2.1 Creating a loosely coupled architecture	5%
3. Improving product and process		**30%**
	3.1 Using customer feedback	5%
	3.2 Visualizing flow and the value stream	12.5%
	3.3 Working in small batch sizes	5%
	3.4 Fostering team experimentation	7.5%
4. Using Lean management and monitoring		**10%**
	4.1 Implementing lightweight change approval processes	2.5%
	4.2 Monitoring and checking system health	2.5%
	4.3 Limiting work in progress (WIP)	5%
5. Implementing cultural change		**17.5%**
	5.1 Supporting a generative culture	2.5%
	5.2 Becoming a learning organization	2.5%
	5.3 Fostering collaboration	5%
	5.4 Making work meaningful	5%
	5.5 Promoting transformational leadership	2.5%
6. Assessing maturity		**5%**
	6.1 Assessing maturity with the help of DevOps capability	2.5%
	6.2 Visualizing progress in DevOps maturity	2.5%
	Total	**100%**

Exam specifications

1 Implementing continuous delivery

1.1 Implementing version control

The candidate can…

1.1.1 explain why version control is important.

1.1.2 explain how to keep version control over data, infrastructure, and components.

1.2 Automating deployment

The candidate can…

1.2.1 explain the logic of the anatomy of a deployment pipeline.

1.2.2 recommend the best way to reach continuous deployment in a scenario.

1.3 Establishing continuous integration

The candidate can…

1.3.1 explain how to implement continuous integration in a scenario.

1.4 Introducing test automation

The candidate can…

1.4.1 explain why test automation is important.

1.4.2 explain how to automate testing in a scenario.

1.5 Managing data and data security (DevSecOps)

The candidate can…

1.5.1 explain which problems can be encountered when managing data in databases within DevOps.

1.5.2 recommend the best way to guarantee information security (DevSecOps) in a scenario.

1.6 Establishing continuous delivery

The candidate can…

1.6.1 explain why continuous delivery is essential.

1.6.2 recommend the best way to achieve continuous delivery in a scenario.

1.7 Trunk-based development

The candidate can…

1.7.1 explain the benefits of trunk-based development.

2 Designing a DevOps architecture

2.1 Creating a loosely coupled architecture

The candidate can…

2.1.1 explain the benefits of a loosely coupled architecture.

2.1.2 analyze the best way to prepare an infrastructure environment for deployment and manage it after deployment in a scenario.

3 Improving product and process

3.1 Using customer feedback

The candidate can…

3.1.1 explain why feedback loops are necessary in DevOps.

3.1.2 identify ways to enable feedback in a scenario.

3.2 Visualizing flow and the value stream

The candidate can…

3.2.1 explain how DevOps adds value to the business through focusing on the value stream.

3.2.2 explain why DevOps improves customer experience through focusing on adding value.

3.2.3 explain how visual control over a DevOps project facilitates DevOps practices.

3.2.4 analyze a scenario for a problem with automation or flow and find a suitable solution.

3.2.5 explain why visualization is key to DevOps.

3.3 Working in small batch sizes
The candidate can…
3.3.1 explain how a DevOps project's scope should be determined.
3.3.2 explain why single-piece flow is beneficial.
3.4 Fostering team experimentation
The candidate can…
3.4.1 explain why experimentation and low-risk releases are important.
3.4.2 explain why experimentation is necessary in DevOps.
3.4.3 identify opportunities for experimentation and hypothesis-driven development in a scenario.

4 Using Lean management and monitoring

4.1 Implementing Lean management practices
The candidate can…
4.1.1 explain why Lean management is useful for DevOps.
4.1.2 explain the use of a lightweight change management process.
4.2 Monitoring and checking system health
The candidate can…
4.2.1 identify ways to prevent problems or issues based on telemetry in DevOps in a scenario.
4.3 Limiting work in progress (WIP)
The candidate can…
4.3.1 explain how imposing work-in-progress limits (WIP-limits) helps establish flow in DevOps.
4.3.2 identify opportunities to improve flow through limiting work in progress (WIP) in a scenario.

5 Implementing cultural change

5.1 Supporting a generative culture
The candidate can…
5.1.1 explain the benefits of a generative culture in contrast to pathological and bureaucratic cultures.
5.2 Becoming a learning organization
The candidate can…
5.2.1 explain why organizational learning and improvement are important.
5.3 Fostering collaboration
The candidate can…
5.3.1 explain how human resource management can foster diversity and which benefits this brings to the organization.
5.3.2 explain how to manage distributed teams or hybrid teams through fostering collaboration in a scenario.
5.4 Making work meaningful
The candidate can…
5.4.1 explain how lean management practices decrease burnout and increase employee loyalty and engagement by providing meaningful work.
5.5 Promoting transformational leadership
The candidate can…
5.5.1 identify the best place to start a DevOps introduction in a scenario.
5.5.2 explain how to expand DevOps throughout an organization by promoting transformational leadership.

6 Assessing maturity

6.1 Assessing maturity with the help of DevOps capability

The candidate can…

6.1.1 explain the need to continuously monitor and increase the maturity of DevOps in an organization.

6.2 Visualizing progress in DevOps maturity

The candidate can…

6.2.1 explain how to visualize DevOps maturity and its progress based on the DevOps continuous everything model or the DevOps cube model.

3. List of basic concepts

This chapter contains the terms and abbreviations with which candidates should be familiar.

Please note that knowledge of these terms alone does not suffice for the exam; the candidate must understand the concepts and be able to provide examples.

A/B-testing
acceptance tests
affinity (in DevOps)
Agile
(application) deployment
artifact management
(artifact) repository
automated testing
automation
backlog
binary files
blamelessness
build (management)
build time
canary releasing
capacity testing
change management
check-in
cloud computing
collaboration (in DevOps)
commit (stage)
communication styles
compact
component test
configuration management
containers/containerization
continuous delivery
continuous deployment
continuous integration
cycle time
DataOps
definition of done (DoD)
dependency
(deployment) pipeline
Development (team)
DevSecOps
distributed team
experimentation
exploratory testing
feedback loops
flow
human error
incident management
information radiator
infrastructure automation
infrastructure management
integration tests
INVEST
- independent
- negotiable
- valuable
- estimable
- small
- testable

iteration
IT service management (ITSM)
kaizen (in Lean)
Lean
libraries
manual testing
minimum viable product (MVP)
monitoring strategy
nonfunctional requirements
Operations team
organizational learning
PDCA cycle
- plan
- do
- check
- act

product backlog
Product Owner (in Agile Scrum)
push system/pull system
quality assurance (QA)
regression testing
retrospective
rhythm (in Lean)
root cause analysis
run time
scaling (in DevOps or Agile Scrum)
Scrum
Scrum Master (in Agile Scrum)
service continuity
service level agreement (SLA)
single-piece flow
system of engagement (SoE)
system of record (SoR)
system tests
takt time
telemetry
test-driven development (TDD)
tools/tooling
Toyota Production System (TPS)
unit test

usability tests
user acceptance testing (UAT)
user story
value stream mapping (VSM)
velocity (in Agile Scrum)
version control
virtualization
waste (in Lean)
waterfall
work in progress (WIP)
work-in-progress limit (WIP-limit)

4. Literature

Exam literature

The knowledge required for the exam is covered in the following literature:

A. Nicole Forsgren, Jez Humble & Gene Kim
Accelerate: The Science of Lean Software and DevOps: Building and Scaling High-Performing Technology Organizations
IT Revolution (1st edition, 2018)
ISBN 9781942788331 (hardcopy)
ISBN 9781942788355 (eBook)

B. Jez Humble, David Farley
Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation
Addison-Wesley Professional (1st edition, 2010)
ISBN 9780321601919 (hardcopy)

C. Bart de Best
DevOps Continuous Assessment: A publication in the Continuous Everything series
Leonon Media (2022)
ISBN 9789492618696 (hardcopy)

Additional literature

D. Koichiro Toda & Nobuyuki Mitsui
Success with Enterprise DevOps
EXIN (2022)
Freely available from www.exin.com. Click on 'Certifications' to find the exam. The download can be found under 'Downloads'.

E. Bart de Best
Continuous Everything
Leonon Media (2021)
ISBN 9789492618597 (hardcopy)

F. Gene Kim, Kevin Behr & George Spafford
The Phoenix Project
IT Revolution Press (January 10, 2013)
ISBN 9780988262577

Comment

Additional literature is for reference and depth of knowledge only.

Literature matrix

Exam requirements	Exam specifications	Reference
1. Implementing continuous delivery		
	1.1 Implementing version control	B, Chapter 2, 12, 13 & 14
	1.2 Automating deployment	B, Chapter 1, 3, 5, 6, 8 & 10
	1.3 Establishing continuous integration	B, Chapter 3 & 14
	1.4 Introducing test automation	B, Chapter 8 & 9
	1.5 Managing data and data security (DevSecOps)	A, Chapter 4 & 6 B, Chapter 12
	1.6 Establishing continuous delivery	B, Chapter 3, 5 & 15
	1.7 Trunk-based development	A, Chapter 4 B, Chapter 4
2. Designing a DevOps architecture		
	2.1 Creating a loosely coupled architecture	A, Chapter 5
3. Improving product and process		
	3.1 Using customer feedback	A, Chapter 4 & 8
	3.2 Visualizing flow and the value stream	A, Chapter 1, 2, 7, 8 & 16 B, Chapter 1 & 15
	3.3 Working in small batch sizes	A, Chapter 2 & 8
	3.4 Fostering team experimentation	A, Chapter 4 & 8
4. Using Lean management and monitoring		
	4.1 Implementing lightweight change approval processes	A, Chapter 7
	4.2 Monitoring and checking system health	A, Chapter 7 & 13 B, Chapter 11
	4.3 Limiting work in progress (WIP)	A, Chapter 7
5. Implementing cultural change		
	5.1 Supporting a generative culture	A, Chapter 1, 2 & 3
	5.2 Becoming a learning organization	A, Chapter 3, 10, 16 and Foreword
	5.3 Fostering collaboration	A, Chapter 3, 5, 10 & 11
	5.4 Making work meaningful	A, Chapter 10, 11 & Appendix A
	5.5 Promoting transformational leadership	A, Chapter 11 & 16
6. Assessing maturity		
	6.1 Assessing maturity with the help of DevOps capability	C, Chapter 3, 4 & Appendix B
	6.2 Visualizing progress in DevOps maturity	C, Chapter 5, 6, 7 & 8

Contact EXIN

www.exin.com