



Vorbereitungshandbuch

Ausgabe 201805

Copyright © EXIN Holding B.V. 2018. All rights reserved.

EXIN® is a registered trademark.

DevOps Master™ is a registered trademark.

No part of this publication may be published, reproduced, copied or stored in a data processing system or circulated in any form by print, photo print, microfilm or any other means without written permission by EXIN.



Inhalt

1. Überblick	4
2. Prüfungsanforderungen	8
3. Liste der Grundbegriffe	12
4. Literatur	15

1. Überblick

EXIN DevOps Master™ (DEVOPSM.DE)

Anwendungsbereich

Die Zertifizierung EXIN DevOps Master™ bestätigt, dass ein IT-Experte folgende Themen beherrscht:

- DevOps übernehmen;
- Planung, Anforderung und Design;
- Entwicklung und Bereitstellung;
- IT Betrieb und Skalierung;
- End-of-Life Management.

Zusammenfassung

Der Name DevOps setzt sich aus den ersten Silben der englischen Bezeichnungen 'Development' und 'Operations', d.h. Entwicklung und Betrieb zusammen. DevOps umfasst eine Reihe von Best Practices, die die Kollaboration und Kommunikation von IT-Experten (Entwicklern, Anwendern und Support-Mitarbeitern) im Lebenszyklus von IT-Anwendungen und IT-Services betonen und dadurch zu folgenden Ergebnissen führen:

- Continuous Integration (kontinuierliche Integration): alle entwickelten Arbeitskopien werden mehrmals täglich in den gemeinsamen Hauptzweig (Mainline) eingebracht
- Continuous Deployment (kontinuierliche Bereitstellung): neue Releases erfolgen kontinuierlich oder so oft wie möglich
- Continuous Feedback (kontinuierliches Feedback): Einholen des Feedbacks der Beteiligten in allen Phasen des Lebenszyklus

DevOps verändert die Art, in der Menschen über ihre Arbeit denken; DevOps schätzt die Diversität der ausgeführten Tätigkeiten, unterstützt durchdachte Prozesse, die die Wertschöpfung von Unternehmen beschleunigen und ermittelt die Auswirkungen gesellschaftlicher und sozialer Änderungen. DevOps ist eine Denk- und Arbeitsmethode, die es Menschen und Organisationen ermöglicht, nachhaltige Arbeitspraktiken zu entwickeln und aufrechtzuerhalten.

Effektives DevOps bedeutet:

- Eine positive Fehlerkultur geprägt von Erfahrungsaustausch und Empathie, in dem Einzelne, aber auch Teams, ihr Können wirksam und nachhaltig anwenden können.
- Anwendungen und Services Just-in-Time (JiT) an die Unternehmen bereitzustellen
- Die Kontinuität von IT-Services für betriebliche Anforderungen risikobasiert sicherzustellen
- Die Verwaltung des kompletten Lebenszyklus von IT-Anwendungen und IT-Services, einschließlich der End-of-Life Bedingungen

Diese Zertifizierung ergänzt theoretisches Wissen um praktische Fähigkeiten und ermöglicht damit dem DevOps Master™, DevOps Praktiken im Team erfolgreich zu moderieren und die Prinzipien von DevOps innerhalb der Organisation zu fördern.

Die Zertifizierung wurde gemeinsam mit DevOps-Experten entwickelt.

Kontext

Das DevOps Programm von EXIN :



Zielgruppe

Am bekanntesten ist die DevOps-Methode in der Softwareentwicklung. Die DevOps-Prinzipien lassen sich jedoch auf IT-Service-Projekte und andere Projekte ebenso anwenden. Die Schulung und Zertifizierung zum DevOps Master™ richtet sich an alle Experten, die DevOps-Praktiken in ihrer Organisation fördern möchten.

Die Zertifizierung EXIN DevOps Master™ richtet sich an jede Person, die in einem DevOps-Kontext arbeiten oder deren Organisation die Umstellung auf DevOps erwägt. Zur Zielgruppe zählen u.a. Product Owner, Agile Scrum Masters, Projekt Manager, Test Manager, IT Service Manager, Prozess Manager und Anwender von Lean IT.

Zertifizierungsvoraussetzungen

1. Erfolgreicher Abschluss einer EXIN Accredited EXIN DevOps Master™-Schulung mit praktischen Aufgabenstellungen.
2. Erfolgreicher Abschluss der Prüfung zum EXIN DevOps Master™.

Einzelheiten zur Prüfung

Prüfungsart:	Multiple-Choice Fragen
Anzahl der Fragen:	50
Mindestanforderung zum Bestehen der Prüfung:	65%
Hilfsmaterialien zugelassen?	Nein
Elektronische Geräte/Hilfsmittel erlaubt?	Nein
Examensdauer:	120 Minuten

Es gilt die EXIN Examen Regeln und Vorschriften.

Taxonomiestufe nach Bloom

Die Zertifizierung nach EXIN DevOps Master™ prüft Teilnehmer nach den Bloom Level 3 und 4 der überarbeiteten Bloom'schen Taxonomie:

- Bloom Level 3: Anwenden – diese Stufe zeigt, dass der Teilnehmer Inhalte in einem anderen als dem gelernten Kontext anwenden kann.
Die Fragen zu dieser Lernstufe sollen zeigen, dass der Teilnehmer Probleme in neuen Situationen lösen kann, indem er das erworbene Wissen bzw. die gelernten Tatsachen, Techniken und Regeln auf eine andere oder neue Art anwendet. Die Fragen beschreiben in der Regel ein kurzes Szenario.
- Bloom Level 4: Analysieren – diese Stufe zeigt, dass der Teilnehmer gelernte Inhalte zum besseren Verständnis in ihre Bestandteile gliedern kann. Diese Lernzielstufe nach Bloom wird in erster Linie mit Hilfe praktischer Aufgabenstellungen geprüft. Praktische Aufgabenstellungen sollen nachweisen, dass der Teilnehmer Informationen prüfen und in ihre Bestandteile zerlegen kann, indem er Motive oder Ursachen identifiziert, Schlussfolgerungen trifft und Belege für allgemein gültige Aussagen findet.

Schulung

Die Teilnahme an der Schulung ist für die Zertifizierung verpflichtend. Von den Teilnehmern wird erwartet, dass sie über Grundkenntnisse in den DevOps-Prinzipien und in den Begriffen der Lean-IT und Agile-Methoden verfügen. Die Teilnehmer können dieses Wissen auf folgende Arten erwerben:

- durch die Zertifizierung zum EXIN DevOps Professional
- oder
- durch Lektüre des Buchs „The Phoenix Project“ (siehe Literaturliste).

Präsenzstunden

Dieser Kurs umfasst mindestens 24 Präsenzstunden. Darin enthalten sind praktische Aufgabenstellungen, Prüfungsvorbereitung und kurze Pausen, nicht jedoch die Zeit, die für Hausaufgaben, die mit der Prüfung verbundene Logistik, die Prüfung und Mittagspausen benötigt wird.

Richtlinien für die praktischen Aufgabenstellungen finden Sie im Partnernet. Wir empfehlen auf die praktische Aufgabenstellung maximal 8 Stunden zu verwenden. Die praktischen Aufgabenstellungen können außerhalb der Schulung durchgeführt werden. Bei den praktischen Aufgabenstellungen können die Trainer zwischen folgenden Optionen wählen:

- Die GamingWorks-Simulation (dafür benötigt der Trainer eine GamingWorks-Akkreditierung)
- Das Spiel aus dem Buch: Introduction to DevOps with Chocolate, LEGO and Scrum Game¹
- Die Entwicklung eigener praktischer Aufgabenstellungen

Die Schulungsdauer kann je nach Umfang der im Rahmen der Schulung durchgeführten praktischen Aufgabenstellungen variieren. Möglich sind u.a.:

- 2 Tage theoretisches Präsenztraining & 1 Tag praktische Aufgabenstellung im Rahmen der Schulung
- 2 Tage theoretische Online-Schulung & 8 Stunden praktische Aufgabenstellung am Arbeitsplatz; der Teilnehmer studiert mittels e-Learning und bereitet die praktischen Aufgabenstellungen vor; der Trainer bewertet die im Rahmen der praktischen Aufgabenstellungen geleistete Arbeit.
- 2 Tage theoretische Präsenzs Schulung & 8 Stunden praktische Aufgabenstellung am Arbeitsplatz; der Trainer bewertet die im Rahmen der praktischen Aufgabenstellung geleisteten Arbeit

Regelstudiendauer

120 Stunden je nach Vorwissen. Die Literaturmatrix in Kapitel 4. *Literatur* dieses Vorbereitungshandbuchs bezieht sich auf das in der Prüfung abgefragte Wissen.

Schulungsanbieter

Eine Liste mit unseren akkreditierten Schulungsanbietern finden Sie unter www.exin.com.

¹ **Introduction to DevOps with Chocolate, LEGO and Scrum Game**

Dana Pylayeva

ISBN-13: 978-1-4842-2565-3

ISBN-10: 1484225643

Apress: 2017

2. Prüfungsanforderungen

Les conditions de l'examen sont détaillées dans les spécifications de l'examen. Le tableau suivant énumère les sujets du module (conditions de l'examen) et les sous-rubriques (spécifications de l'examen).

Prüfungsanforderungen	Prüfungsspezifikationen	Gewichtung
1. DevOps übernehmen		28%
	DevOps Denkweise und Vorteile	10%
	Organisationskultur	12%
	DevOps Prinzipien und Konzepte	6%
2. Planung, Anforderung und Design		18%
	Anwendungs - oder Service-Lifecycle Management	4%
	Projektcharta (Defining Scope) und visuelle Kontrolle	4%
	Infrastruktur- und Architektur-Design	4%
	Service Level Requirements (SLR) und Service Level Agreements (SLA)	2%
	Implementieren einer Test-Strategie	4%
3. Entwicklung und Bereitstellung		30%
	Continuous Delivery (kontinuierliche Lieferung) und Continuous Integration (kontinuierliche Integration)	12%
	Deployment-Pipeline	4%
	Continuous Deployment (kontinuierliche Bereitstellung)	4%
	Ji-Kotei-Kanketsu, Rhythm (Rhythmus), Work-in-Progress und One-piece-flow (In einem Rutsch)	4%
	Automatisierung, Tools (Hilfsmittel) und Testen	6%
4. IT Betrieb und Skalierung		22%
	Management von Daten; Infrastruktur und Umgebung, Komponenten und deren Abhängigkeiten	10%
	Konfigurationsmanagement und Versionskontrolle	4%
	Cloud und unveränderbare (Immutable) Infrastruktur	2%
	Business Continuity	2%
	Skalierung	4%
5. End-of-Life Management		2%
	Bedingungen für End-of-Life eines Produktes oder einer Dienstleistung	2%
Total		100%

Prüfungsspezifikationen

1. DevOps übernehmen

1.1 DevOps Denkweise und Vorteile

Der Kandidat kann...

- 1.1.1 schlechte Lösungsansätze in einem DevOps-Szenario analysieren
- 1.1.2 die Vorteile von DevOps erklären
- 1.1.3 erklären, warum DevOps sehr gut zum aktuellen Software-Entwicklungsprozess passt
- 1.1.4 erklären, warum man eine spezielle Denkweise für die Umsetzung von DevOps benötigt
- 1.1.5 erklären, wie DevOps mit Methoden wie Lean, Agile und Scrum zusammen passt

1.2 Organisationskultur

Der Kandidat kann...

- 1.2.1 erklären, warum die vier Säulen von DevOps (Kollaboration, Affinität, Tools, und Skalierung) so wichtig sind
- 1.2.2 ein Szenario auf das Fehlen von Teilen der DevOps Denkweise analysieren
- 1.2.3 erklären, wie man aus einer Gruppe von Menschen durch Kollaboration, der DevOps Denkweise, sowie Empathie und Vertrauen ein Team entwickelt
- 1.2.4 eine Situation analysieren, in der es ein Missverständnis in Bezug auf die Kollaboration gibt und die richtige Problembehandlung identifizieren
- 1.2.5 eine Situation analysieren, in der Konfliktmanagement benötigt wird und die beste Lösung identifizieren
- 1.2.6 erklären, wie das Personalmanagement die kulturelle Vielfalt fördern kann und welche Vorteile das für die Organisation bringt

1.3 DevOps Prinzipien und Konzepte

Der Kandidat kann...

- 1.3.1 die Verwendung und den Nutzen verschiedener Softwareentwicklungsmethoden (Wasserfall, Agile, Scrum) und deren Grundlagen erläutern
- 1.3.2 die Verwendung und den Nutzen verschiedener Methoden im IT Betrieb (Operations) (IT Service Management) erläutern
- 1.3.3 die Verwendung und den Nutzen der Lean Methode erläutern

2. Planung, Anforderung und Design

2.1 Anwendungs - oder Service-Lifecycle Management

Der Kandidat kann...

- 2.1.1 erklären, wie DevOps einen Mehrwert zum modernen Anwendungs-Lifecycle liefern kann
- 2.1.2 erklären, wie DevOps das Kundenerlebnis verbessert, wenn es im Rahmen vom Service-Lifecycle Management eingesetzt wird

2.2 Projektcharta (Defining Scope) und visuelle Kontrolle

Der Kandidat kann...

- 2.2.1 erklären, wie ein DevOps-Projektumfang ermittelt werden sollte
- 2.2.2 erklären, warum visuelle Kontrolle über ein DevOps-Projekt die DevOps-Praktiken erleichtert

2.3 Infrastruktur- und Architektur-Design

Der Kandidat kann...

- 2.3.1 erklären, wie DevOps das Design von IT-Infrastruktur und deren Architektur verändert oder beeinflusst

- 2.3.2 erklären, warum Cloud Computing und Virtualisierungstechniken die Integration von „Dev“ und „Ops“ vereinfachen
- 2.4 Service Level Requirements (SLR) und Service Level Agreements (SLA)
Der Kandidat kann...
 - 2.4.1 erläutern, wie DevOps die Service Level Requirements und Service Level Agreements ändert
- 2.5 Implementieren einer Test-Strategie
Der Kandidat kann...
 - 2.5.1 erklären, warum und wie bei einem Wechsel zu DevOps die Teststrategie geändert werden muss
 - 2.5.2 User Stories auf Vollständigkeit analysieren

3. Entwicklung und Bereitstellung

- 3.1 Continuous Delivery (kontinuierliche Lieferung) und Continuous Integration (kontinuierliche Integration)
Der Kandidat kann...
 - 3.1.1 erklären, warum Continuous Delivery (kontinuierliche Lieferung) für den Einsatz von effektivem DevOps essentiell ist
 - 3.1.2 analysieren, wie Continuous Delivery (kontinuierliche Lieferung) in ein Szenario integriert wird
 - 3.1.3 analysieren, wie Continuous Delivery (kontinuierliche Lieferung) Probleme in einem Szenario lösen kann
 - 3.1.4 erklären, warum Continuous Integration (kontinuierliche Integration) für effektives DevOps wichtig ist
 - 3.1.5 analysieren, wie Continuous Integration (kontinuierliche Integration) in einem Szenario mit einem verteilten Team oder einem verteilten Versionskontrollsystem (VCS) erreicht wird
 - 3.1.6 analysieren, wie Probleme mit Continuous Integration (kontinuierliche Integration) in einem Szenario gelöst werden
- 3.2 Deployment Pipeline
Der Kandidat kann...
 - 3.2.1 die Logik die einzelnen Bestandteile einer DevOps-Deployment Pipeline erklären
 - 3.2.2 erklären, wie man Build- und Bereitstellungs-Skripte benutzt
- 3.3 Continuous Deployment (kontinuierliche Bereitstellung)
Der Kandidat kann...
 - 3.3.1 erklären, warum der Iterations- und Releaseplan für effektives DevOps geändert werden sollten
 - 3.3.2 analysieren, wie Continuous Deployment (kontinuierliche Bereitstellung) in einem Szenario implementiert wird
- 3.4 Ji-Kotei-Kanketsu, Rhythm (Rhythmus), Work-in-Progress und One-piece-flow (In einem Rutsch)
Der Kandidat kann...
 - 3.4.1 die Konzepte von Ji-Kotei-Kanketsu, Rhythm (Rhythmus), Work-in-Progress und One-piece-flow (In einem Rutsch) erklären
 - 3.4.2 ein Szenario auf ein Problem mit Ji-Kotei-Kanketsu, Rhythm (Rhythmus), Work-in-Progress oder One-piece-flow (In einem Rutsch) untersuchen und eine passende Lösung finden
- 3.5 Automatisierung, Tools und Testen
Der Kandidat kann...
 - 3.5.1 erklären, warum Automatisierung für effektives DevOps wichtig ist
 - 3.5.2 erklären, wie man Tools verwendet, um DevOps im Allgemeinen zu erleichtern
 - 3.5.3 erklären wie man Tools verwendet, um die DevOps-Denkweise und Kultur zu unterstützen

- 3.5.4 erklären, warum es wichtig ist, dass DevOps-Tests automatisiert sind
- 3.5.5 ein Szenario analysieren und die richtige Art der Automatisierung eines Akzeptanztests wählen

4. IT Betrieb und Skalierung

- 4.1 Management von Daten; Infrastruktur und Umgebung, Komponenten und deren Abhängigkeiten
Der Kandidat kann...
 - 4.1.1 erklären, welche Probleme bei der Verwaltung von Daten in Datenbanken innerhalb von DevOps auftreten können
 - 4.1.2 ein Szenario analysieren, in dem eine Datenbank in DevOps verwendet wird und die beste Lösung für ein Problem anbieten
 - 4.1.3 ein Szenario analysieren und den besten Weg identifizieren, um eine Infrastruktur für die Bereitstellung vorzubereiten oder diese nach der Bereitstellung zu verwalten
 - 4.1.4 ein Szenario analysieren und eine allgemeine Strategie zur Verwaltung der Komponenten vorschlagen
 - 4.1.5 erklären, wie man Abhängigkeiten verwaltet
- 4.2 Konfigurationsmanagement und Versionskontrolle
Der Kandidat kann...
 - 4.2.1 erklären, warum Versionskontrolle ein Schlüssel zu effektivem DevOps ist
 - 4.2.2 erläutern, wie die Versionskontrolle über Daten, Infrastruktur und Komponenten beibehalten wird
 - 4.2.3 ein Szenario analysieren und die beste Strategie zur Behandlung eines Konfigurations-Problems vorschlagen
- 4.3 Cloud und unveränderbare (Immutable) Infrastruktur
Der Kandidat kann...
 - 4.3.1 erklären, wann es nötig ist und wann es nicht nötig ist, auf eine cloudbasierte Infrastruktur für effektive DevOps zu wechseln
 - 4.3.2 erklären, wie cloudbasierte Infrastruktur innerhalb von DevOps verwaltet werden sollte
- 4.4 Business Continuity
Der Kandidat kann...
 - 4.4.1 erklären, wie DevOps Business Continuity-Praktiken erleichtern kann
- 4.5 Skalierung
Der Kandidat kann...
 - 4.5.1 ein Szenario analysieren und erläutern, ob und warum es wichtig ist, in dieser Situation die Skalierung anzupassen und den besten Weg zu finden, dies zu tun
 - 4.5.2 ein Szenario analysieren in dem die Skalierung fehlgeschlagen ist und einen guten Weg identifizieren, um das Problem zu lösen
 - 4.5.3 erklären, wie Sozialpolitik und Einstellungsmethoden im Unternehmen die Skalierung von DevOps unterstützen können

5. End-of-Life Management

- 5.1 Bedingungen für End-of-Life eines Produktes oder einer Dienstleistung
Der Kandidat kann...
 - 5.1.1 erläutern, welche Bedingungen vor der Beendigung einer Dienstleistung oder eines Produktes erfüllt werden sollten

3. Liste der Grundbegriffe

Dieses Glossar enthält die Begriffe und Abkürzungen, mit denen die Teilnehmer vertraut sein sollten.

Bitte beachten Sie, dass die Kenntnis dieser Begriffe alleine nicht ausreicht. Die Teilnehmer müssen diese Begriffe auch verstehen und mit Beispielen belegen können.

English

A/B testing
 Acceptance tests
 Affinity (*in DevOps*)
 Agile
 Anti-pattern
 Application Deployment
 Artifact Management
 (Artifact) repository
 ATAM
 Automated testing
 Automation
 Binary files
 Blamelessness
 Blue-Green deployment
 Build (Management)
 Build-time
 Canary releases
 Capacity tests
 Change Management
 Check-in
 Cloud Computing
 Collaboration (*in DevOps*)
 Commit (stage)
 Communication styles
 Compact
 Component (tests)
 Configuration Management
 Containers
 Continuous Delivery
 Continuous Deployment

 Continuous Integration

 Cycle time
 Definition of Done (*in Agile Scrum*)

 Dependency
 (Deployment) Pipeline
 Development Team
 DevOps Engineer
 Disciplined Agile
 Distributed Team
 Effective DevOps
 Event Management

German

A/B Testing
 Automatisierte Abnahmetests
 Affinität
 Agile (Agil)
 Anti-Pattern
 Anwendungsbereitstellung
 Management von Artefakten
 (Artefakt-)Repository
 ATAM
 Automatisiertes Testen
 Automatisierung
 Binären Dateien
 Frei von Schuldzuweisung
 Blue-Green Deployment
 Build (Management)
 Build-Zeit
 Canary-Releases
 Kapazitätstest
 Change Management
 Check-in
 Cloud Computing (Cloud)
 Kollaboration (*in DevOps*)
 Commit(-Phase)
 Kommunikationstile
 Teamverbund
 Komponenten(tests)
 Konfigurationsmanagement
 Container
 Continuous Delivery (kontinuierliche Lieferung)
 Continuous Deployment (kontinuierliche Bereitstellung)
 Continuous Integration (kontinuierliche Integration)
 Zykluszeit
 Definition of Done (Definition von 'fertiggestellt') (*in Agile Scrum*)
 Abhängigkeit
 (Deployment) Pipeline
 Entwicklungsteam
 DevOps Engineer
 Disziplinierte Agilität
 Dezentralen Entwicklungsteams
 Wirksames DevOps
 Event Management

Exploratory testing	Versuchsforschung
Flow	Fluss
Functional acceptance tests	Funktionelle Akzeptanztests
Forensic tools	Forensischer Tools (Hilfsmittel)
Gatekeeper	Gatekeeper
Happy path	Happy Path
Human error	Menschlicher Fehler
Incident Management	Incident Management
Information radiators	Informations-Verbreiter
Infrastructure Automation	Infrastruktur-Automatisierung
INVEST	INVEST
Integration tests	Integrationstests
Iteration	Iteration
ITSM (IT Service Management)	ITSM (IT Service Management)
Ji-Kotei-Kanketsu (JKK)	Ji-Kotei-Kanketsu (JKK)
Just-in-Time (JiT)	Just-in-Time (JiT)
Kaizen (<i>in Lean</i>)	Kaizen (<i>in Lean</i>)
Lean	Lean
Libraries	Libraries
Light-weight ITSM	Lightweight ITSM
Manual testing	Manuelles Testen
Minimum Viable Product	Produkt mit Mindestfunktionalität (den minimalen Anforderungen entsprechendes, lauffähiges Produkt)
Monitoring strategy	Überwachungsstrategie
Negotiation styles	Verhandlungsstile
Non-functional testing	Nicht-funktionales Testen
Obeya	Obeya
Ochestration	Orchestrierung
One-piece-flow	One-piece-flow (In einem Rutsch)
Operations Team	Betriebsteam
Organizational Learning	organisationsbezogenes Lernen
Plan-Do-Check-Act cycle (PDCA cycle)	PDCA-Zyklus (Plan-Do-Check-Act)
Process Master	Process Master
(Product) Backlog	(Product) Backlog
Product Owner (<i>in Agile Scrum</i>)	Product Owner (<i>in Agile Scrum</i>)
Project charter	Projekt Charter
Pull system	Pull-System
Quality Assurance (QA)	Qualitätssicherung
Regression testing	Regressions-testen
Release Coordinator	Release Coordinator
Reliability Engineer	Reliability Engineer
Retrospective	Retrospektive
Rhythm (<i>in Lean</i>)	Rhythm (Rhythmus) (<i>in Lean</i>)
Root Cause Analysis	Ursachenanalyse
Run time	Laufzeit
Sad path	Sad Path
Scaling (<i>of DevOps or Agile Scrum</i>)	Skalierung (<i>von DevOps or Agile Scrum</i>)
Scrum	Scrum
Scrum Master (<i>in Agile Scrum</i>)	Scrum Master (<i>in Agile Scrum</i>)
Service Continuity	Service Continuity
Service Level Agreement (SLA)	Service Level Agreement (SLA)
Service Master	Service Master
Silos	Silos
Sprint	Sprint
System of Engagement (SoE)	System-of-Engagement (SoE)

System of Record (SoR)
System tests
Test-Driven Development
Tools
Toyota Production System (TPS)
Unit test
Usability test
User Acceptance Testing (UAT)
User Story
Value Stream Mapping
Velocity (*in Agile Scrum*)
(Vendor) lock-in
Version Control
Virtualization
Waste (*in Lean*)
Waterfall
Work-in-Progress (WiP)

System-of-Records (SoR)
Systemtests
Testgetriebene Entwicklung
Tools (Hilfsmittel)
Toyota Produktions-System (TPS)
Unit-Tests
Test der Benutzerfreundlichkeit
Benutzerabnahmetest
User Story
Wertstromanalyse
Teamgeschwindigkeit (*in Agile Scrum*)
(Vendor-)Lock-in
Versionskontrolle
Virtualisierung
Zeitverschwendung (*in Lean*)
Wasserfall (-Modell)
Work-in-Progress (WiP)

4. Literatur

Fachliteratur zur Prüfung

Das für die Prüfung EXIN DevOps Master™ benötigte Wissen wird durch folgende Literatur abgedeckt.

- A. Jennifer Davis, Katherine Daniels
Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale
O'Reilly Media; 1 edition (June 25, 2016)
ISBN-13: 978-1491926307
ISBN-10: 1491926309
- B. Jez Humble, David Farley
Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation
Addison-Wesley Professional; 1 edition (August 6, 2010)
ISBN-13: 978-0321601919
ISBN-10: 0321601912
- C. Koichiro (Luke) Toda, President Strategic Staff Services Corporation and Director of TPS Certificate Institution, and Nobuyuki Mitsui, CTO of Strategic Staff Services Corporation
Success with Enterprise DevOps
White Paper; June 2016 (download from EXIN DevOps Master™ product page)

Weiterführende Literatur

- D. Gene Kim, Kevin Behr, George Spafford
The Phoenix Project
IT Revolution Press (January 10, 2013)
ISBN-10: 0988262576
ISBN-13: 978-0988262577
- E. Gene Kim, Jez Humble, Patrick Debois, John Willis
The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations
IT Revolution Press; 1 edition (2016)
ISBN-10: 1942788002
ISBN-13: 978-1942788003
- F. Other sources:
<http://newrelic.com/devops>
<http://devops.com/>

Auswahlkriterien

Weiterführende Literatur dient nur als Referenz und zur Vertiefung des Wissens. Wir empfehlen Ihnen dringend das Buch The Phoenix Project vor Beginn der Schulung zu lesen.

Literaturmatrix

Prüfungsanforderung		Literatur
1. DevOps übernehmen		
1.1	DevOps Denkweise und Vorteile	
1.1.1	schlechte Lösungsansätze in einem DevOps-Szenario analysieren	A: Kapitel 5 B: Kapitel 1 C: Kapitel 2
1.1.2	die Vorteile von DevOps erklären	B: Kapitel 1 C: Kapitel 4
1.1.3	erklären, warum DevOps sehr gut zum aktuellen Software-Entwicklungsprozess passt	A: Kapitel 3 B: Kapitel 1 C: Kapitel 4
1.1.4	erklären, warum man eine spezielle Denkweise für die Umsetzung von DevOps benötigt	A: Kapitel 1, 2, 4 C: Kapitel 4
1.1.5	erklären, wie DevOps mit Methoden wie Lean, Agile und Scrum zusammen passt	B: Kapitel 15 C: Kapitel 1, 4
1.2	Organisationskultur	
1.2.1	erklären, warum die vier Säulen von DevOps (Kollaboration, Affinität, Tools, und Skalierung) so wichtig sind	A: Kapitel 6, 7
1.2.2	ein Szenario auf das Fehlen von Teilen der DevOps Denkweise analysieren	A: Kapitel 7
1.2.3	erklären, wie man aus einer Gruppe von Menschen durch Kollaboration, der DevOps Denkweise, sowie Empathie und Vertrauen ein Team entwickelt	A: Kapitel 7, 9
1.2.4	eine Situation analysieren, in der es ein Missverständnis in Bezug auf die Kollaboration gibt und die richtige Problembehandlung identifizieren	A: Kapitel 7, 8, 9, 16
1.2.5	eine Situation analysieren, in der Konfliktmanagement benötigt wird und die beste Lösung identifizieren	A: Kapitel 7, 14
1.2.6	erklären, wie das Personalmanagement die kulturelle Vielfalt fördern kann und welche Vorteile das für die Organisation bringt	A: Kapitel 7
1.3	DevOps Prinzipien und Konzepte	
1.3.1	die Verwendung und den Nutzen verschiedener Softwareentwicklungsmethoden (Wasserfall, Agile, Scrum) und deren Grundlagen erläutern	A: Kapitel 4 B: Kapitel 1 C: Kapitel 5
1.3.2	die Verwendung und den Nutzen verschiedener Methoden im IT Betrieb (Operations) (IT Service Management) erläutern	C: Kapitel 3, 4
1.3.3	die Verwendung und den Nutzen der Lean Methode erläutern	A: Kapitel 4
2. Planung, Anforderung und Design		
2.1	Anwendungs - oder Service-Lifecycle Management	
2.1.1	erklären, wie DevOps einen Mehrwert zum modernen Anwendungs-Lifecycle liefern kann	C: Kapitel 1
2.1.2	erklären, wie DevOps das Kundenerlebnis verbessert, wenn es im Rahmen vom Service-Lifecycle Management eingesetzt wird	C: Kapitel 4
2.2	Projektcharta (Defining Scope) und visuelle Kontrolle	
2.2.1	erklären, wie ein DevOps-Projektumfang ermittelt werden sollte	B: Kapitel 10 C: Kapitel 8
2.2.2	erklären, warum visuelle Kontrolle über ein DevOps-Projekt die DevOps-Praktiken erleichtert	B: Kapitel 5 C: Kapitel 7

	2.3	Infrastruktur- und Architektur-Design	
	2.3.1	erklären, wie DevOps das Design von IT-Infrastruktur und deren Architektur verändert oder beeinflusst	B: Kapitel 11
	2.3.2	erklären, warum Cloud Computing und Virtualisierungstechniken die Integration von „Dev“ und „Ops“ vereinfachen	B: Kapitel 11
	2.4	Service Level Requirements (SLR) und Service Level Agreements (SLA)	
	2.4.1	erläutern, wie DevOps die Service Level Requirements und Service Level Agreements ändert	B: Kapitel 12 C: Kapitel 4
	2.5	Implementieren einer Test-Strategie	
	2.5.1	erklären, warum und wie bei einem Wechsel zu DevOps die Teststrategie geändert werden muss	B: Kapitel 4
	2.5.2	User Stories auf Vollständigkeit analysieren	B: Kapitel 4
3. Entwicklung und Bereitstellung			
	3.1	Continuous Delivery (kontinuierliche Lieferung) und Continuous Integration (kontinuierliche Integration)	
	3.1.1	erklären, warum Continuous Delivery (kontinuierliche Lieferung) für den Einsatz von effektivem DevOps essentiell ist	B: Kapitel 13, 15
	3.1.2	analysieren, wie Continuous Delivery (kontinuierliche Lieferung) in ein Szenario integriert wird	B: Kapitel 3, 5, 15
	3.1.3	analysieren, wie Continuous Delivery (kontinuierliche Lieferung) Probleme in einem Szenario lösen kann	B: Kapitel 15
	3.1.4	erklären, warum Continuous Integration (kontinuierliche Integration) für effektives DevOps wichtig ist	B: Kapitel 3
	3.1.5	analysieren, wie Continuous Integration (kontinuierliche Integration) in einem Szenario mit einem verteilten Team oder einem verteilten Versionskontrollsystem (VCS) erreicht wird	B: Kapitel 3
	3.1.6	analysieren, wie Probleme mit Continuous Integration (kontinuierliche Integration) in einem Szenario gelöst werden	B: Kapitel 3
	3.2	Deployment Pipeline	
	3.2.1	die Logik die einzelnen Bestandteile einer DevOps-Deployment Pipeline erklären	B: Kapitel 5, 8 C: Kapitel 7
	3.2.2	erklären, wie man Build- und Bereitstellungs-Skripte benutzt	B: Kapitel 1, 6
	3.3	Continuous Deployment (kontinuierliche Bereitstellung)	
	3.3.1	erklären, warum der Iterations- und Releaseplan für effektives DevOps geändert werden sollten	C: Kapitel 5, 7
	3.3.2	analysieren, wie Continuous Deployment (kontinuierliche Bereitstellung) in einem Szenario implementiert wird	B: Kapitel 10
	3.4	Ji-Kotei-Kanketsu, Rhythm (Rhythmus), Work-in-Progress und One-piece-flow (In einem Rutsch)	
	3.4.1	die Konzepte von Ji-Kotei-Kanketsu, Rhythm (Rhythmus), Work-in-Progress und One-piece-flow (In einem Rutsch) erklären	C: Kapitel 4, 7
	3.4.2	ein Szenario auf ein Problem mit Ji-Kotei-Kanketsu, Rhythm (Rhythmus), Work-in-Progress oder One-piece-flow (In einem Rutsch) untersuchen und eine passende Lösung finden	A: Kapitel 1, 2, 9 B: Kapitel 1 C: Kapitel 4

	3.5	Automatisierung, Tools und Testen	
	3.5.1	erklären, warum Automatisierung für effektives DevOps wichtig ist	B: Kapitel 1, 8
	3.5.2	erklären, wie man Tools verwendet, um DevOps im Allgemeinen zu erleichtern	B: Kapitel 8
	3.5.3	erklären wie man Tools verwendet, um die DevOps-Denkweise und Kultur zu unterstützen	A: Kapitel 12
	3.5.4	erklären, warum es wichtig ist, dass DevOps-Tests automatisiert sind	A: Kapitel 1 B: Kapitel 3
	3.5.5	ein Szenario analysieren und die richtige Art der Automatisierung eines Akzeptanztests wählen	B: Kapitel 8
4. IT Betrieb und Skalierung			
	4.1	Management von Daten; Infrastruktur und Umgebung, Komponenten und deren Abhängigkeiten	
	4.1.1	erklären, welche Probleme bei der Verwaltung von Daten in Datenbanken innerhalb von DevOps auftreten können	B: Kapitel 12
	4.1.2	ein Szenario analysieren, in dem eine Datenbank in DevOps verwendet wird und die beste Lösung für ein Problem anbieten	B: Kapitel 12
	4.1.3	ein Szenario analysieren und den besten Weg identifizieren, um eine Infrastruktur für die Bereitstellung vorzubereiten oder diese nach der Bereitstellung zu verwalten	B: Kapitel 11
	4.1.4	ein Szenario analysieren und eine allgemeine Strategie zur Verwaltung der Komponenten vorschlagen	B: Kapitel 13
	4.1.5	erklären, wie man Abhängigkeiten verwaltet	B: Kapitel 13
	4.2	Konfigurationsmanagement und Versionskontrolle	
	4.2.1	erklären, warum Versionskontrolle ein Schlüssel zu effektivem DevOps ist	B: Kapitel 2
	4.2.2	erläutern, wie die Versionskontrolle über Daten, Infrastruktur und Komponenten beibehalten wird	B: Kapitel 10, 11
	4.2.3	ein Szenario analysieren und die beste Strategie zur Behandlung eines Konfigurations-Problems vorschlagen	B: Kapitel 2
	4.3	Cloud und unveränderbare (Immutable) Infrastruktur	
	4.3.1	erklären, wann es nötig ist und wann es nicht nötig ist, auf eine cloudbasierte Infrastruktur für effektive DevOps zu wechseln	A: Kapitel 17 B: Kapitel 11 C: Kapitel 5, 7
	4.3.2	erklären, wie cloudbasierte Infrastruktur innerhalb von DevOps verwaltet werden sollte	B: Kapitel 11
	4.4	Business Continuity	
	4.4.1	erklären, wie DevOps Business Continuity-Praktiken erleichtern kann	C: Kapitel 4
	4.5	Skalierung	
	4.5.1	ein Szenario analysieren und erläutern, ob und warum es wichtig ist, in dieser Situation die Skalierung anzupassen und den besten Weg zu finden, dies zu tun	A: Kapitel 14
	4.5.2	ein Szenario analysieren in dem die Skalierung fehlgeschlagen ist und einen guten Weg identifizieren, um das Problem zu lösen	A: Kapitel 15 B: Kapitel 11
	4.5.3	erklären, wie Sozialpolitik und Einstellungsmethoden im Unternehmen die Skalierung von DevOps unterstützen können	A: Kapitel 6, 7, 14

5. End-of-Life Management		
5.1	Bedingungen für End-of-Life eines Produktes oder einer Dienstleistung	
5.1.1	erläutern, welche Bedingungen vor der Beendigung einer Dienstleistung oder eines Produktes erfüllt werden sollten	C: Kapitel 7

Anmerkung: Die Lektüre des im Literaturverzeichnis unter Punkt D aufgeführten Buchs The Phoenix Project trägt insbesondere zum Verständnis der folgenden Spezifikationen bei:

- 1.1
- 1.2
- 3.1
- 3.3
- 3.4
- 4.4

Kontakt EXIN

www.exin.com

