DevOps INSTITUTE

SRE Foundation℠
Exam Study Guide
考试学习指南

DevOps Institute's SKIL Framework

DevOps Institute is dedicated to advancing the human elements of DevOps success through our human-centered SKIL framework of Skills, Knowledge, Ideas and Learning.

We develop, accredit and orchestrate SKIL through certifications, research, learning opportunities, events and community connections.

Visit the SKILup℠ CAFE at www.devopsinstitute.com to learn more.

Join Us!

Become a member and join the fastest growing global community of DevOps practitioners and professionals.

The DevOps Institute continuous learning community is your go-to hub for all things DevOps, so get ready to learn, engage, and inspire.

Visit https: www.devopsinstitute.com/become-a-community-member to join today.

You belong.

DevOps Institute is dedicated to advancing the human elements of DevOps success. We fulfill our mission through our SKIL framework of Skills, Knowledge, Ideas and Learning.

Certification is one means of showcasing your skills. While we strongly support formal training as the best learning experience and method for certification preparation, DevOps Institute also recognizes that humans learn in different ways from different resources and experiences. As the defacto certification body for DevOps, DevOps Institute has now removed the barrier to certification by removing formal training prerequisites and opening our testing program to anyone who believes that they have the topical knowledge and experience to pass one or more of our certification exams.

This examination study guide will help test-takers prepare by defining the scope of the exam and includes the following:

- Course Description
- Examination Requirements
- DevOps Glossary of Terms
- Value Added Resources
- Sample Exam(s) with Answer Key

These assets provide a guideline for the topics, concepts, vocabulary and definitions that the exam candidate is expected to know and understand in order to pass the exam. The knowledge itself will need to be gained on its own or through training by one of our Global Education Partners.

Test-takers who successfully pass the exam will also receive a certificate and digital badge from DevOps Institute, acknowledging their achievement, that can be shared with their professional online networks.

If you have any questions, please contact our DevOps Institute Customer Service team at CustomerService@DevOpsInstitute.com.

站点可靠性工程 （SRE） 基础℠
课程描述

时间 - 16 小时

介绍了通过自动化、工作方法和组织调整等手段提高服务可靠性的一系列实践。
专为专注于大规模服务可用性的人量身定制。

概述

SRE（站点可靠性 Engineering） 理 基础管理课程是原则和实践的介绍，使组织能够可靠且经济地扩展关键服务。引入站点可靠性维度需要组织重新协调，需要重新关注智能和自动化，以及采用一系列新的工作模式。

本课程重点介绍 SRE 的演变及其未来方向，并为学员提供实践、方法和工具，让整个组织的人都参与其中，使用真实情景和案例故事来论述如何实现可靠性和稳定性。课程结束后，学员在回到办公室时将有切实的收获，例如了解、设置和跟踪 服务级别目标 （SLO' s）。

本课程通过利用关键 SRE 来源、与 SRE 领域的思想领袖接触以及与采用 SRE 的组织合作来提取实际最佳实践，旨在教授开始采用 SRE 所需的关键原则和做法。

本课程为学员提供成功完成 SRE 基础认证考试的培训课程。

课程目标

SRE 基金会课程的学习目标包括以下实际内容：

- SRE的历史及其在谷歌的出现
- SRE 与 DevOps 和其他流行框架的交互关系
- SRE 背后的基本原则
- 服务级别目标 （SLO） 及其用户焦点
- 服务级别指示器 （SLI）是现代监控环境
- 错误预算和相关错误预算策略
- 琐事及其对组织生产力的影响
- 一些有助于消除琐事的实际步骤
- 可观察到一种指示服务运行状况的东西
- SRE 工具、自动化技术和安全性的重要性
- 抗脆弱性，我们处理故障和故障测试的方法
- 引入 SRE 带来的组织影响

目标学员
SRE 基金会课程的目标受众包括专业人员，包括：

- 任何人开始或领导提高可靠性
- 任何对现代 IT 领导和变革方法感兴趣的人
- 业务经理
- 业务利益相关者
- 更改代理
- 顾问
- DevOps 从业者
- IT 主管
- IT 经理
- IT 团队领导
- 产品所有者
- Scrum 大师
- 软件工程师
- 站点可靠性工程师
- 系统集成商
- 工具提供商

学员材料

- 十六 （16） 小时的讲师指导培训和锻炼促进
- 学员手册（优秀的课后参考）
- 参与旨在应用概念的独特练习
- 示例文档、模板、工具和技术
- 获得额外的增值资源和社区

先决条件
建议对常见的DevOps术语、知识和相关工作经验的理解和了解。

认证考试
成功通过 （65%）60 分钟的考试包括 40 个多选题， 获得 SRE（现场可靠性工程）基础证书。
证书认证由 DevOps 研究所管理和维护。

站点可靠性工程 （SRE） 基础SM

考试要求

站点可靠性工程 （SRE）基础SM 证书

站点可靠性工程 （SRE） 基础是 DevOps 研究所的独立认证。 此认证及其相关课程的目的是传授、测试和验证 SRE 基本词汇、原则和实践的知识。SRE 基础旨在让个人了解基本的 SRE 概念，以及如何通过应用现场可靠性工程原则和工程实践来利用 SRE 来改善运营活动。

考试资格

虽然没有正式的考试先决条件，但 DevOps 研究所强烈建议以下为考生准备获得 SRE 基础认证的考试：

- 建议考生完成至少 16 个接触时间（指导和实验室），作为 DevOps 学院认证教育合作伙伴提供的正式、批准的培训课程的一部分

考试

管理 SRE 基础认证根据 DevOps 研究所的严格协议和标准进行认证、管理和管理。

难度级别

SRE 基础认证在内容与考试的构建中均采用教育目标的绽放分类法。

- SRE 基础考试包含布鲁姆 1 个问题，测试 ***学员对*** SRE 概念和词汇术语的了解（请参阅下面的列表）
- 考试还包含 Bloom 2 个问题，测试学员 ***在上下文中*** 对这些概念的理解

考试格式

考生必须取得通过分数才能获得 SRE 基础证书。

考试类型	40 个多选题
时间	60 分钟
先决条件	建议考生从经过认证的 DevOps 学院教育合作伙伴完成现场可靠性工程 （SRE） 基础课程
监督	否
打开图书	是
通过分数	65%
交货	基于 Web
徽章	SRE 基础认证

考试主题区域和问题权重

SRE 基础考试需要了解下面介绍的主题领域。

主题区域	描述	最大问题
SRE = 1：SRE 原则和实践	SRE 定义、历史、SRE 和 DevOps 以及 SRE 原则和实践	4
SRE = 2：服务级别目标和错误预算	了解服务级别目标 （SLO）、错误预算、错误预算策略、为组织设置 SLO 策略	6
SRE = 3： 减少琐事	了解琐事，以及为什么减少琐事是糟糕的人力和组织机会	5
SRE = 4：监控和服务级别指示器	了解服务级别指标 （SLI）及其与服务级别目标 （SLO） 的关系、监控环境、可观察性以及设定可衡量的服务目标	7
SRE = 5：SRE 工具和自动化	自动化定义、DevOps 和 SRE 自动化重点、SRE 自动化类型、工具环境概述	6
SRE = 6：抗脆弱，从失败中学习	从失败中学习的好处， 定义反脆弱性，改变组织平衡， 混乱工程	4
SRE = 7：SRE 的组织影响	为什么组织接受 SRE、SRE 采用模式、SRE 的组织影响、可持续事件响应、无可指责的事后检查、扩展 SRE	4
SRE-8：SRE、其他框架、趋势	SRE 和 DevOps、敏捷、ITSM、SRE 的演变、SRE 分拆，如网络可靠性工程和客户可靠性工程	4

概念和术语列表

应聘者应理解、理解和应用以下 SRE 基础概念和词汇，在 Blooms 1（知识）和 2（理解）

敏捷	监测
抗脆弱	平均恢复服务时间 （MTRS）
应用程序性能管理 （APM）	检测缺陷平均时间 （MTTD）
自动缩放	平均修复 /恢复时间 （MTTR）
自动回滚	网络可靠性工程师 （NRE）
可用 性	非功能测试
亚马逊网络服务 （AWS）	可观测
无可指责的事后复盘	On-call
爆炸半径	病理
官僚文化	可靠性
业务连续性	弹性
金丝雀测试	响应时间
混沌工程	可伸缩性
云计算	软件交付生命周期 （SDLC）
查托普	安全自动化
客户可靠性工程师 （CRE）	自我修复
数据库可靠性工程师 （DBRE）	服务级别协议 （SLA）
DevOps	服务级别指示器 （SLI）
错误预算	服务级别目标 （SLO）
错误预算策略	Simian Army
外部自动化	现场可靠性工程师 （SRE）
消防钻头	软件定义网络 （SDN）
功能测试	稳定性
传统可靠性工程师 （HRE）	遥测
不可更改的基础架构	三种方式
事件响应	琐事
内部自动化	流量
ITIL	速度
IT 服务管理 （ITSM）	韦斯特鲁姆 （组织类型）
Kubernetes	
延迟	

DEVOPS

术语表

本词汇表仅供参考，因为它包含可能或可能无法审查的关键术语。

DevOps术语表

术语	定义	课程外观
12要素应用程序设计	一种用于构建现代的，可伸缩的，可维护的软件即服务应用程序的方法。	持续交付架构
两要素或两步身份验证	两要素身份验证（也称为2FA或TFA）或两步身份验证是指用户提供两个身份验证因素时；通常首先是密码，然后是第二层验证，例如发给设备的文本短信，共享机密，物理令牌或生物识别信息。	Devsecops工程
A / B测试	将EUT的不同版本部署到不同的客户，并让客户反馈确定哪个是最好的。	持续交付架构
A3问题解决	一种结构化的问题解决方法，使用一种称为A3问题解决报告的精益工具。 术语“ A3”代表历史上用于报告的纸张尺寸（尺寸大致等于11“ x 17”）。	DevOps基金会
访问管理	基于定义的标准（例如，映射角色），授予对授权资源（例如，数据，服务，环境）的身份验证访问权限，同时防止对资源的未授权身份访问。	Devsecops工程
访问配置	访问供应是协调用户帐户，规则和角色形式的电子邮件授权以及其他任务（例如供应与使新用户进入系统或环境相关联的物理资源的供应）的创建过程。	Devsecops工程
行政测试	该测试的目的是确定最终用户测试（EUT）是否能够按预期处理管理任务。	持续交付架构

咨询流程	任何做出决定的人都必须向受到该决定有意义影响的每个人以及对此事具有专门知识的人寻求建议。 尽管不必接受或遵循建议，但必须考虑收到的建议。 建议过程的目的不是形成共识，而是通知决策者，以便他们做出最佳决策。不遵循建议流程会破坏信任，并给业务带来不必要的风险。	Devsecops工程
敏捷	一种用于复杂项目的项目管理方法，该方法将任务划分为小的工作"冲刺"，并需要频繁地重新评估和调整计划。	认证的敏捷流程所有者，认证的敏捷服务经理，站点可靠性工程
敏捷（形容词）	能够快速轻松地移动；协调良好。 能够快速思考和理解；能够解决问题并有新想法。	Devops[，Devsecops]
敏捷教练	帮助团队掌握敏捷开发和DevOps实践；实现高效的工作和协作方式。	DevOps负责人
敏捷企业	快速发展，灵活而强大的公司能够快速应对意外的挑战，事件和机遇。	Devops[，Devsecops]
敏捷宣言	对价值和原则进行正式声明，以指导以迭代和以人为中心的软件开发方法。http://agilemanifesto.org	DevOps基金会
敏捷投资组合管理	涉及评估飞行中的项目以及拟议的未来计划，以塑造和管理对项目和全权委托工作的持续投资。 CA的Agile Central和VersionOne就是示例。	站点可靠性基金会
敏捷原则	敏捷宣言的十二项原则。	敏捷服务认证经理
敏捷过程设计	敏捷服务管理（Agile SM）的方面，与开发人员对软件开发的应用相同的敏捷方法用于流程设计。	敏捷服务认证经理
敏捷流程改进	通过持续改进使敏捷价值与ITSM流程保持一致的敏捷SM方面。	敏捷服务认证经理

敏捷流程负责人	使用敏捷和Scrum原理和实践来设计，管理和衡量单个流程的ITSM或其他类型的流程所有者。	DevOps基金会
敏捷服务管理	确保ITSM流程反映敏捷价值的框架，并设计有“足够多”的控制和结构，以便有效，高效地提供服务，以在需要时以及如何按需促进客户结果。	敏捷服务认证经理
敏捷服务管理工件	流程积压，Sprint积压，燃尽图，流程增量	敏捷流程所有者认证
敏捷服务管理事件	Sprint审查，Sprint回顾 [（），冲刺），Sprint*，冲刺， *** * * * * *，* *	敏捷流程所有者认证
敏捷服务管理角色	流程所有者，流程改进团队（团队）和敏捷服务经理。 另请参阅Scrum角色。	敏捷流程所有者认证
敏捷服务经理	操作等效于Dev的ScrumMaster。 IT组织中的角色，了解如何利用敏捷和Scrum方法来改进ITSM流程的设计，速度和敏捷性。	DevOps基金会
敏捷软件开发	一组软件开发方法，其中需求和解决方案通过自组织，跨职能团队之间的协作发展。 通常使用Scrum或可扩展敏捷框架方法来应用。	持续交付架构，DevOps基础，DevSecOps工程
亚马逊网络服务（AWS）	Amazon Web Services（AWS）是一个安全的云服务平台，提供计算能力，数据库存储，内容交付和其他功能，以帮助企业扩展和发展。	DevSecOps工程，站点可靠性工程
分析工具	根据分析方法和标准以有组织的方式处理和呈现测试结果。	持续交付架构，DevOps测试工程
安东	系统使流水线工人有能力（而且还赋予他们权力）在发现缺陷时停止生产并立即寻求帮助。	持续交付架构

反模式	通常重新发明但解决问题的方法较差。	DevOps基金会

抗漏洞	抗脆弱性是系统的一种属性，由于压力，冲击，波动，噪声，错误，错误，故障，攻击或故障的结果，系统的抗衰老能力得以增强。	DevOps Foundation，站点可靠性工程
API测试	测试的目的是确定EUT的API是否按预期运行。	持续交付架构，DevOps测试工程
应用程序性能管理（APM）	APM是对软件应用程序的性能和可用性的监视和管理。 APM努力检测和诊断复杂的应用程序性能问题，以维持预期的服务水平。	现场可靠性工程
应用程序编程接口（API）	一组协议，用于为特定的OS创建应用程序或作为模块或应用程序之间的接口。	Devops[，Devsecops]
应用程序编程接口（API）测试		持续交付架构
申请发布	该测试的目的是确定EUT的API是否按预期运行。	持续交付架构
应用程序发布自动化（ARA）或编排（ARO）	受控的连续交付管道功能，包括自动化（在提交代码后释放）。	持续交付架构
应用测试驱动开发（ATDD）	受控的连续交付管道功能，包括自动化（代码提交后发布），环境建模（端到端管道阶段以及将应用程序二进制文件，程序包或其他工件部署到目标环境）和发布协调（项目，日历和调度管理，集成）变更控制和/或IT服务支持管理）。	持续交付架构
应用测试	验收测试驱动开发（ATDD）是一种做法，整个团队共同讨论示例的验收标准，然后在开发开始之前将其提炼为一组具体的验收测试。	持续交付架构

被测应用程序（AUT）	测试的目的是确定应用程序是否正在根据其要求和预期行为执行。	持续交付架构，DevOps测试工程

建筑	计算机硬件，软件或两者结合的基本基础设计。	Devsecops工程
神器	软件开发项目中的任何元素，包括文档，测试计划，图像，数据文件和可执行模块。	持续交付架构，DevOps基础，DevSecOps工程
工件存储库	存储二进制文件，报告和元数据。 示例工具包括：JFrog Artifactory，Sonatype Nexus。	持续交付架构，DevOps基金会
攻击路径	威胁可以利用一系列弱点来实现攻击者的目标。 例如，攻击路径可能首先破坏用户的凭据，然后在易受攻击的系统中使用该凭据升级特权，而特权又又被用于访问受保护的信息数据库，该数据库被复制到攻击者自己的服务器上）。	Devsecops工程
审计管理	使用自动化工具来确保产品和服务是可审核的，包括保留构建，测试和部署活动的审核日志，审核配置和用户以及生产操作的日志文件。	现场可靠性工程
认证方式	验证断言身份的过程。 身份验证可以基于您知道的信息（例如密码或PIN），拥有的信息（令牌或一次性代码），您的身份（生物统计信息）或上下文信息。	Devsecops工程
授权书	向用户授予角色访问资源的过程。	Devsecops工程
自动开发	Auto DevOps通过自动配置软件开发生命周期，将DevOps最佳实践带到您的项目中。 它会自动检测，构建，测试，部署和监视应用程序。	现场可靠性工程
自动缩放	在保持成本控制的同时，能够根据流量和容量的变化自动，灵活地伸缩基础架构和缩小规模。	持续交付架构

自动回滚	如果在部署过程中检测到故障，则操作员（或自动化过程）将验证故障并将失败的版本回滚到以前的已知工作状态。	现场可靠性工程

可用性	可用性是系统处于运行状态并因此可供（用户）使用的时间的比例。	现场可靠性工程
后门	后门绕过用于访问系统的常规身份验证。 其目的是即使组织已纠正最初用于攻击系统的漏洞，也仍可以授予网络罪犯将来访问系统的权限。	Devsecops工程
积压	系统要求，通常以"用户故事"的形式表示为产品待办事项的优先列表。 产品积压由产品负责人优先处理，应包括功能，非功能和技术团队生成的需求。	持续交付架构，DevOps基金会
基本安全卫生	一组通用的最低安全实践，必须毫无例外地应用于所有环境。 实践包括基本的网络安全性（防火墙和监视），强化，漏洞和补丁管理，日志记录和监视，基本策略和实施（可以通过"策略作为代码"方法来实施）以及身份和访问管理。	Devsecops工程
批次大小	指单个代码版本中涉及的功能量。	DevOps负责人
贝特森利益相关者地图	反映利益相关者对正在进行的计划的参与的工具。	DevOps负责人
行为驱动开发（BDD）	通过模拟EUT的外部可观察输入和输出来创建测试用例。	持续交付架构
超越预算	示例工具：黄瓜。	DevOps负责人
黑盒子	一种超越命令和控制的管理模型，而朝着更加授权和自适应的状态发展。	持续交付架构，DevOps测试工程

无责的尸检	测试用例仅使用有关EUT外部可观察行为的知识。	现场可靠性工程

爆炸半径	用于服务事件的影响分析。 当特定的IT服务失败时， 用户， 客户以及其他相关服务也会受到影响。	现场可靠性工程
蓝色/绿色测试或部署	使用两个标记为Blue和Green的环境将软件从测试的最后阶段带到现场生产。 软件在绿色环境中运行后， 请切换路由器， 使所有传入请求都进入绿色环境-蓝色的请求现在处于空闲状态。	持续交付架构， DevOps测试工程
虫子	软件中的错误或缺陷， 导致意外或系统降级的情况。	Devsecops工程
官僚文化	官僚组织可能会使用在危机中可能不够用的标准渠道或程序（Westrum）。	DevOps负责人
燃尽图	该图显示了剩余工作量随时间的变化。	DevOps基金会认证的敏捷服务经理
爆裂	根据需要添加公共云资源， 以临时增加私有云的总计算能力。	持续交付架构
商业案例	根据预期的商业利益为拟议项目或事业的理由。	DevOps负责人
业务连续性	业务连续性是组织确保操作和核心业务功能不会受到使关键服务脱机的灾难或计划外事件的严重影响的能力。	现场可靠性工程
业务转型	改变业务运作方式。 实现这一目标意味着要改变文化， 流程和技术， 以更好地使每个人围绕执行组织的使命而团结一致。	Devsecops工程
商业价值	关键业务KPI的方法的好处。	DevOps负责人

节奏	事件的流程或节奏。	Devops[，Devops]，负责人，Devsecops工程

平静模型	Jez 提出John Damon [DevOps]：]，[，]，]，[DevOps约翰·威利斯，达蒙·爱德华兹]））。	DevOps基金会
金丝雀测试	金丝雀（也称为金丝雀测试）是将代码更改推送给少数自愿自愿进行任何测试的最终用户。与增量部署类似，它是一小部分用户群首先被更新到新版本的地方。 该子集即金丝雀，然后成为俗称的"煤矿中的金丝雀"。 如果出现问题，则会回滚发行版，并且只会影响一小部分用户。	持续交付架构，站点可靠性工程
容量测试	该测试的目的是确定EUT是否可以处理预期的负载，例如用户数量，会话数量，聚合带宽。	持续交付架构
捕获重播	通过捕获与EUT的实时交互来创建测试用例，其格式可以由工具重播。 例如。 硒	持续交付架构，DevOps测试工程
萝卜	积极激励措施，用于鼓励和奖励期望的行为。	Devsecops工程
目标链	Roman Pichler设计的一种方法，用于确保通过产品开发过程在各个级别上都实现目标的链接和共享。	DevOps负责人
更改	添加，修改或删除任何可能影响IT服务的内容。 （ITIL®定义）	Devops[，Devsecops]
变更失败率	度量失败/回滚更改的百分比。	持续交付架构，DevOps基金会
改变疲劳	对个人或团队对组织变革的冷漠或被动辞职的一般感。	Devsecops工程
更改提前期	从更改请求到更改交付的时间度量。	DevOps基金会
变革领导者发展模式	吉姆·坎特鲁奇（Jim Canterucci）的模式具有五个级别的变革领导能力。	DevOps负责人

更换管理层	在整个生命周期中控制所有变更的过程。（ITIL定义）	Devops[，Devops]，负责人，Devsecops工程

变更管理（组织）	一种将个人，团队和组织从当前状态转移或转移到当前状态的方法	DevOps负责人
基于变更的测试选择方法	所需的未来状态。 包括管理变更人员以实现所需业务成果的流程，工具和技术。	持续交付架构，DevOps测试工程
混沌工程	根据将测试属性与构建中更改的代码的属性相匹配的标准选择测试。	现场可靠性工程
章节负责人	在生产中的软件系统上进行实验的纪律，以便对系统抵御动荡和意外情况的能力建立信心。	DevOps负责人
章节	Spotify模型中的班级经理负责传统的人员管理职责，参与日常工作并提高个人和部门的能力。	DevOps负责人
聊天操作	一小群人，他们具有相似的技能，并且在同一部落的同一一般能力领域内工作。	持续交付架构，DevOps基础，DevOps测试工程，站点可靠性工程
报到	各章定期开会，讨论挑战和专业领域，以促进共享，技能开发，重用和解决问题。	持续交付架构，DevOps测试工程
CI回归测试	一种管理技术和业务运营的方法（由GitHub创造），涉及群组聊天和与DevOps工具的集成的结合。 示例工具包括：Atlassian HipChat / Stride，Microsoft Teams，Slack。	持续交付架构

透明盒	将软件更改提交到系统版本管理系统的操作。	持续交付架构，DevOps测试工程
云计算	构建软件组件后立即运行的回归测试的子集。与烟雾测试相同。	DevSecOps工程，站点可靠性工程

云原生	本机云应用程序（NCA）专为云计算而设计。	持续交付架构
云蜂	Cloudbees是商业上受支持的专有自动化框架工具，可通过提供企业级支持和附加功能与Jenkins协同工作并对其进行增强。	DevOps测试工程
集群成本优化	诸如Kubecost，Replex和Cloudability之类的工具使用监视来分析容器集群并优化资源部署模型。	现场可靠性工程
集群监控	可让您了解在Kubernetes等群集中运行的部署环境的运行状况的工具。	现场可靠性工程
聚类	一组计算机（称为节点或成员）作为群集一起工作，这些群集通过充当单个系统的快速网络连接在一起。	持续交付架构
代码覆盖率	通过计算测试执行的代码单元来衡量白盒测试覆盖率。 代码单元可以是代码语句，代码分支或通过代码模块的控制路径或数据路径。	持续交付架构，DevOps测试工程
代码质量	另请参阅静态代码分析，Sonar和Checkmarks是自动检查代码质量的七个主要维度的工具示例，这些维度包括注释，体系结构，重复项，单元测试覆盖率，复杂性，潜在缺陷，语言规则。	现场可靠性工程
代码库	开发人员可以在其上提交和协作其代码的存储库。 它还跟踪历史版本，并可能识别同一代码的冲突版本。 也称为“存储库”或“存储库”。	Devsecops工程
代码审查	软件工程师检查彼此的源代码以检测编码或代码格式错误。	持续交付架构，DevOps测试工程

认知偏差	认知偏见是客观思维的局限性，它是由于人脑倾向于通过个人经验和偏好过滤器来感知信息：偏离规范或理性判断的系统模式。	DevOps负责人

合作	人们与他人共同努力实现共同的目标。	Devops[，Devsecops]
合作文化	一种适用于每个人的文化，其中包含一种预期的行为，语言和公认的相互合作的方式，这种方式由领导层加强。	持续交付架构
相容性测试	测试目的在于确定和EUT是否与另一个EUT（例如对等应用程序或协议）进行互操作。	持续交付架构，DevOps测试工程
配置管理	配置管理（CM）是一个系统工程过程，用于建立和维持产品性能，功能和物理属性与其在整个生命周期中的需求，设计和操作信息的一致性。	持续交付架构，DevOps基础，DevSecOps工程
符合性测试	该测试的目的是确定EUT是否符合标准。	持续交付架构，DevOps测试工程
约束	限制或约束；约束的东西。 另请参见瓶颈。	Devops[，Devsecops]
容器	一种将软件打包为轻量，独立，可执行程序包的方式，包括运行该程序以进行开发，运输和部署所需的一切（代码，运行时，系统工具，系统库，设置）。	DevOps 基金会， 德夫科波普斯
容器网络安全	用来证明可以在容器群集上与任何其他应用程序一起运行的任何应用程序都可以确信，没有其他应用程序的意外使用或它们之间没有任何意外的网络流量。	现场可靠性工程
集装箱登记处	容器映像的安全和私有注册表。 通常，它允许从构建工具轻松上传和下载图像。 Docker Hub，Artifactory，Nexus是示例。	现场可靠性工程

容器扫描	在为应用程序构建容器映像时，工具可以运行安全扫描，以确保在代码交付环境中不存在任何已知漏洞。 Blackduck，Synopsis，Synk，Claire和klar就是例子。	现场可靠性工程

持续服务改进（CSI）	ITIL核心出版物之一，以及服务生命周期的一个阶段。	DevOps基金会
（CD）	专注于确保软件在其整个生命周期中始终处于可发布状态的方法。	开发运营基金会，敏捷运维基金会，敏捷运维基金会认证的敏捷服务经理
持续交付（CD）架构师	负责指导持续交付流程的实施和最佳实践的人员。	工程，DevOps测试工程
连续交付管道	连续交付流水线是指分阶段对产品更改执行的一系列过程。 在管道的开始处注入更改。 更改可能是应用程序的代码，数据或图像的新版本。 每个阶段都处理前一阶段产生的工件。 最后阶段导致部署到生产。	持续交付架构
连续交付阶段	每个过程都在一个连续的交付管道中。 这些不是标准的。 例如设计：确定实施变更；创建：实施设计更改的未集成版本；整合：合并	持续交付架构，DevOps基础课程，DevOps负责人
持续部署	一组实践，使通过自动测试的每个更改都可以自动部署到生产中。	持续交付架构
连续流	将人员或产品从流程的第一步平稳地移动到最后一步，而步骤之间的缓冲最少（或没有）。	Devops[，Devsecops]
连续的提高	基于Deming的“计划-执行-检查-行动”，该模型可确保不断进行改进产品，流程和服务的工作。	Devops[，Devops]，负责人，Devsecops工程

它 -第一）次	一种开发实践，要求开发人员至少每天至少将其代码合并到主干或精通，并在每次提交代码时执行测试（即单元，集成和验收）。	Devops[，Devops]负责人

持续集成工具	通过定期合并，构建和测试代码来提供即时反馈循环的工具。 示例工具包括：Atlassian Bamboo，Jenkins，Microsoft VSTS / Azure DevOps，TeamCity。	Devops[，Devops]负责人
[（CM）	这是与测试结果信息的日志记录，通知，警报，显示和分析有关的一类术语。	持续交付架构，DevOps测试工程
* （CT）	这是与在DevOps环境中测试和验证EUT有关的一类术语。	开发运维
对话咖啡厅	对话咖啡厅是开放式的，可以在咖啡馆以及会议和教室中进行对话，人们可以在任何地方聚集在一起以了解我们的世界。	基础，持续交付架构，DevOps测试工程
康威定律	设计系统受约束的组织必须生成设计，这些设计是这些组织的通信结构的副本。	DevOps负责人
合作与竞争	关键的文化价值转向高度协作和合作，而不再具有内部竞争力和分歧性。	持续交付架构，DevOps负责人
栈	商用现货解决方案	Devsecops工程
关键成功因素（CSF）	为了使IT服务，流程，计划，项目或其他活动成功，必须发生的事情。	持续交付架构，DevOps测试工程
CSI寄存器	在整个生命周期中记录和管理改进机会的工具（持续服务改进）。	DevSecOps Engineering的DevOps Foundation的认证敏捷流程所有者，认证的敏捷服务经理

文化冰山	隐喻，可视化（在水面之上）和不可观察（在水线以下）文化元素之间的差异。	敏捷服务认证经理
文化（组织文化）	有助于组织独特的社会心理环境的价值观和行为。	DevOps负责人

累积流程图	累积流程图是用于敏捷软件开发和精益产品开发的工具。 它是一个	DevOps负责人
当前状态图	区域图，描述了给定状态下的工作量，显示了到达，排队的时间，排队的数量和出发。	DevOps负责人
客户可靠性工程师（CRE）	价值流图的一种形式，可以帮助您确定当前流程的工作方式以及断开连接的位置。	父亲可靠性工程
周期	当您学习SRE的原理和课程并将其应用于客户时，您将获得CRE。	Devops[，Devops]负责人。
·斯库姆	从开始工作到准备交付所需时间的量度。	Devsecops工程
仪表板	每天15分钟或更短的计时活动，以便团队在Sprint期间重新计划第二天的工作。	DevOps基金会认证的敏捷服务经理
数据丢失保护（DLP）	图形显示汇总的测试结果。	持续交付架构，DevOps测试工程
数据库可靠性工程师（DBRE）	防止从服务环境或组织内删除文件和内容的工具。	现场可靠性工程
缺陷密度	负责保持支持生产中所有面向用户的服务的数据库系统平稳运行的人员。	现场可靠性工程
完成的定义	单元中发现的故障数每个KLOC＃个缺陷，每个更改＃个缺陷。	持续交付架构，DevOps测试工程

交付节奏	对增量必须实现的期望有共同的理解，才能将其发布到生产中。 （Scrum.org）	DevOps基金会，敏捷运营负责人，敏捷流程负责人，敏捷服务经理
送货包裹	分娩的频率。 例如。 每天，每周等的＃次交付。	持续交付架构，DevOps测试工程
戴明周期	打包用于部署的一组发布项目（文件，图像等）。	持续交付架构，DevOps测试工程

依赖防火墙	许多项目依赖于可能来自未知或未验证提供商的软件包，从而引入了潜在的安全漏洞。 有一些工具可以扫描依赖项，但这是在它们下载之后进行的。 这些工具可防止从一开始就下载这些漏洞。	现场可靠性工程
依赖代理	对于许多组织而言，希望为经常使用的上游映像/包提供一个本地代理。 在CI / CD的情况下，代理负责接收请求并从注册表中返回上游映像，充当拉入式缓存。	现场可靠性工程
依赖扫描	用于在开发和测试应用程序时自动在依赖项中查找安全漏洞。 Synopisis，Gemnasium，Retire.js和bundler-audit是该领域中流行的工具。	现场可靠性工程
部署方式	将特定版本的软件安装到给定的环境中（例如，将新的版本推广到生产环境中）。	Devops[，Devsecops]
可测试性设计	EUT设计具有使其能够被测试的功能。	持续交付架构，DevOps测试工程
设计原则	设计，组织和管理DevOps交付运营模型的原则。	DevOps负责人
开发人员	参与软件开发活动的个人，例如应用程序和软件工程师。	Devops[，Devsecops]
[（开发）	负责为EUT开发变更的个人。 备选：参与软件开发活动的人员，例如应用程序和软件工程师。	持续交付架构，DevOps测试工程

开发测试	确保开发人员的测试环境可以很好地代表生产测试环境。	持续交付架构，DevOps测试工程
被测设备（DUT）	被测设备是设备。 例如。 路由器或交换机正在测试中。	持续交付架构，DevOps测试工程

开发运维	一种文化和专业运动，它强调软件开发人员与IT运营专业人员之间的沟通，协作和集成，同时使软件交付和基础架构变更的过程自动化。 它旨在建立一种文化和环境，使构建，测试和发布软件的过程可以快速，频繁且可靠地进行。 ”（资料来源：维基百科）	DevSecOps工程 [DevOps基金会] DevOps
Devops教练	帮助团队掌握敏捷开发和DevOps实践；实现高效的工作和协作方式。	DevOps负责人
DevOps基础架构	[-德沃普] CT、CMCTCM，和和CI，CD。 工具。	持续交付架构，DevOps测试工程
德沃普 · 凯森	Kaizen是日语单词，紧密地翻译为“为了更好而改变”，这种不断改进的想法（无论大小）涉及所有员工并跨越组织边界。 达蒙·爱德华兹（Damon Edwards）的DevOps Kaizen显示了进行小规模的增量改进（小J值）如何长期改善生产率。	DevOps负责人
Devops管道	构成DevOps基础架构的整套互连过程。	持续交付架构，DevOps测试工程
德沃普斯	显示整个组织内DevOps采用率以及对交付速度的相应影响的指标。	现场可靠性工程
DevOps工具链	支持从概念到价值实现的DevOps持续开发和交付周期所需的工具。	[，Devops]，基础，Devsecops[，Devops]测试工程
开发安全	一种“每个人都对安全负责”的心态，其目标是在不牺牲所需安全性的前提下，将安全决策快速，大规模地分发给拥有最高级别上下文的人员。	持续交付架构，DevOps基础，DevSecOps工程

分布式版本控制系统（DVCS）	软件修订版本存储在分布式修订版本控制系统（DRCS）中，也称为分布式版本控制系统（DVCS）。	持续交付架构

非军事区（非军事区）	网络安全用法中的 DMZ 是公共 互联网和内部保护资源之间的网络区域。任何需要外部公开的应用程序、服务器或服务（包括 API）通常放置在 DMZ 中。并行有多个 DMZ 的情况并 不少见。	vSecOps 工程
动态分析	动态分析是测试应用程序，通过实时执行数据，目的是在应用程序运行期间检测缺陷，而不是通过脱机重复检查代码。	连续交付架构，DevOps 测试工程
动态应用程序安全测试 （DAST）	一种针对生成代码运行的测试类型，用于测试公开的接口。	DevSecOps 工程
茄子	企业应用程序的自动化 功能和回归测试。由测试工厂许可。	DevOps 测试工程
弹性基础设施	弹性是云计算中常用的术语，用于描述 IT 基础架构在不妨碍或危及其危害的情况下快速扩展 或削减容量和服务的能力 基础设施的稳定性、性能、安全性、治理或合规性协议。	连续交付架构
电梯间距	用于快速简单地定义流程、产品 、服务、组织或事件及其价值主张的简短摘要。	经过认证的敏捷流程所有者
经验过程控制	过程控制模型，其中决策基于观察和实验（而不是详细的向上规划）和决策基于已知内容。	经过认证的敏捷流程所有者
enps	员工净促进者分数 （eNPS） 是组织衡量员工忠诚度的一种方式.净促进者分数最初是客户服务工具，后来在 内部用于员工而不是客户。	DevOps 基金会，DevOps 领导者

正在测试的实体 （ EUT）	这是一个术语类，它指的是正在测试的实体类型的名称。 这些术语通常缩写为"x"表示正在测试的实体类型的窗体xUT。	连续交付架构， DevOps 测试工程

史诗	大量的工作，由许多用户的故事，有一个共同的目标。	经过认证的敏捷流程所有者
埃里克森（心理社会发展阶段）	Erik Erikson（1950年，1963年）提出了一个心理社会发展的心理分析理论，包括从婴儿期到成年的八个阶段。在每个阶段，患者都经历一场心理社会危机，这种危机对人格发展有积极或消极的结果。	DevSecOps 工程
错误预算	错误预算提供了一个清晰、客观的指标，用于确定服务在特定时间段内不可靠。	站点可靠性工程
错误预算策略	错误预算策略枚举团队在特定时间段中用尽特定服务的错误预算时采取的活动。	站点可靠性工程
错误 跟踪	轻松发现和显示应用程序可能生成的错误以及关联数据的工具 。	站点可靠性工程
外部自动化	旨在减少辛劳的服务外的脚本和自动化。	站点可靠性工程
早失败	DevOps 原则，指的是在开发和交付管道中尽早发现关键问题的偏好。	连续交付架构， DevOps 测试工程
经常失败	DevOps 原则，强调优先尽可能快且经常发现关键问题。	连续交付架构， DevOps 测试工程
故障率	单位时间未判决。	DevOps 基础， 连续的 Deliv ery 架构， DevOps 测试工程
假阴性	当 EUT 实际通过测试的目的时，测试错误地报告"失败"的判断。	连续交付架构， DevOps 测试工程
误报	当 EUT 实际上未能达到测试目的时，测试错误地报告了"通过"的判断。	连续交付架构， DevOps 测试工程

功能切换	使用软件交换机隐藏或激活功能的做法。这 可实现与选定的利益干系人持续集成和测试功能。	DevOps 基础、连续交付架构、DevOps 测试工程

联合标识	用于访问各种应用程序、系统 和服务的中心标识，但特别偏向基于 Web 的应用程序。此外，通常被引用为"身份服务"（IDaas）。任何可以在多个站点上重复使用的标识，特别是通过 SAML 或 OAuth 身份验证我chanisms。	DevSecOps 工程
消防钻头	计划中的故障 测试过程侧重于 实时服务的操作，包括服务故障测试以及通信、文档 和其他人为 因素测试 。	站点可靠性工程
流	人员、产品或信息如何通过流程移动。流是"三种方式"的第一方式。	DevOps 基金会，DevOps 领导者，DevSecOps 工程
价值流	显示端到端价值流的地图 形式。此视图通常在企业中不可用。	DevOps 领导者
框架	用于插入工具的骨干。启动自动任务，从自动任务中收集结果。	连续交付架构，DevOps 测试 工程
自由和责任	一个核心的文化价值，与自我管理的自由（如 DevOps提供）来的责任是勤奋，遵循建议的过程，并取得成功和失败的所有权。	DevSecOps 工程
频率	应用程序发布之次。	DevOps 领导者
功能测试	测试以确定服务的功能操作是否如预期的那样。	站点可靠性工程
未来状态图	一种价值 流图形式，可帮助您开发和传达目标结束状态的外观以及如何处理必要的更改。	DevOps 领导者
模糊	模糊或模糊测试是一种自动软件测试实践，将无效、 意外或随机数据输入应用程序。	DevSecOps 工程

门控提交	定义并取得所有 CD 管道阶段之间促进的更改标准的共识，例如：开发到 CI 阶段/CI 到打包/交付阶段/ 交付到部署/生产阶段。	连续交付架构

生成（DevOps）文化	在生成组织中，通过与特派团的识别进行一致性。个人"购买"他或她 应该做什么，它对结果的影响。生成组织往往在任何方面积极主动地将信息收集给合适的人。必要。（韦斯特鲁姆）	DevOps 领导者
基因性	一种文化观点，其中 长期成果是首要重点，这反过来又推动投资与合作，使一个组织能够取得这些成果。	DevSecOps 工程
玻璃+盒子	与透明 +框测试和白色 +框测试相同。	连续交付 架构, DevOps 测试工程
全局流程所有者	负责监督单个全局流程的流程所有者。全局流程所有者（可能驻留在 SMO 中）可以监督一个或多个区域流程经理。	经过认证的敏捷流程所有者
目标=寻求测试	测试的目的是确定 EUT 的性能边界，使用增量应力，直到 EUT 达到最佳性能。例如，确定可以处理的最大吞吐量，而不会出错。	连续交付弧线, DevOps 测试工程
黄金圈	Simon Sinek 的模型强调在关注"什么"和"如何"之前对业务的"原因"的理解。	DevOps 基金会
黄金图像	虚拟机 （VM）、虚拟桌面 、服务器或硬盘驱动器的模板。（技术目标）	DevSecOps 工程
戈勒曼的六种领导风格	丹尼尔·戈勒曼（2002年）创立了六种领导风格,在他的研究中发现，领导者在任何时间都使用这些 风格之一。	DevOps 领导者

治理、风险管理和合规（GRC）	一个软件平台，用于集中治理、 合规性和风险管理数据，包括策略、合规性要求、漏洞数据 ，有时还包括资产清单、业务连续性计划等。从本质上讲，是用于安全治理的专用文档和数据存储库。或专门从事 IT/安全治理、风险管理和合规活动的团队。最常见的是 n技术业务分析师 资源。	DevSecOps 工程
灰色+框	测试用例对 EUT 的内部设计结构了解有限。	连续交付架构，DevOps 测试工程
GUI 测试	测试的目的是 确定图形用户界面是否按预期运行。	连续交付架构，DevOps 测试工程
行会	一个"兴趣社区"团体，欢迎任何人，通常 跨越整个组织。 类似于实践 社区。	DevOps 基金会，DevOps 领导者
手关	将特定任务的责任从一个个人或团队转移到另一个个人或团队的过程。	DevOps 基金会，DevOps 领导者
硬化	通过删除或禁用不必要的软件、更新到已知良好的操作系统版本、将网络级访问限制为仅所需访问、配置日志记录以捕获警报、进行适当的访问管理和安装适当的安全工具，保护服务器或基础结构环境的安全。	DevSecOps 工程
头盔图表注册表	头盔图表是描述相关库伯内特资源的方式。工件和代码新鲜 支持用于维护赫尔 姆图表的主记录的注册表。	站点可靠性工程
遗产可靠性工程师（HRE）	将 SRE 的原则和实践应用于旧版应用程序和环境。	站点可靠性工程
高信任文化	具有高度信任文化的组织鼓励良好的信息流、跨职能协作、分担责任、从失败中学习和新想法。	DevOps 基金会

水平缩放	计算资源 可扩大规模，以增加处理量。例如，添加更多计算机并并行运行更多任务。	连续交付架构，DevOps 测试工程
幂等	CM 工具（例如，Puppet、Chef、Ansible 和 Salt）声称它们"幂等"，允许将服务器所需的状态定义为代码或声明，并自动执行必要的步骤，以一致实现定义的状态时间— 之后after-时间。	连续交付架构
身份	数字系统识别的人 、设备或两者的组合的唯一名称。也称为"帐户"或"用户"。	DevSecOps 工程
身份和访问管理 （IAM）	确保合适的人有权获得技术资源的政策 、程序和工具。	DevSecOps 工程
服务身份（IDAAS）	通过云或 订阅提供的身份和访问管理 服务。	DevSecOps 工程
基于图像的测试选择 方法	生成映像是预先分配的测试用例。通过匹配生成生成的映像更改，为生成选择测试用例。	连续交付架构，DevOps 测试工程
沉浸式学习	一种学习方法， 指导团队进行辅导和实践，帮助他们以新的方式学习工作。	DevOps 领导者
变	不可变对象是在创建其状态后无法修改其状态的对象。反义词是一个可变对象，可以在创建后 修改。	连续交付架构
不可更改 的基础架构	而不是实例化实例化实例（服务器、容器等），错误-容易，耗时-耗时的补丁和升级（即突变），将其替换为另一个实例，以引入更改或确保正确的行为。	连续交付架构，现场可靠性工程
障碍	任何阻止团队成员尽可能高效地完成工作的事情。	经过认证的敏捷流程所有者 、经过认证的敏捷服务经理、DevOps 基金会
障碍（Scrum）	任何阻止团队成员尽可能高效地完成工作的事情。	敏捷服务管理，DevOps基金会

正在 测试的实现	EUT 是一个 软件实现。例如，正在测试嵌入式程序。	连续交付架构，DevOps 测试工程
改进卡塔	一种结构化的方式，创造一种不断学习和改进的文化。（在日本企业中，Kata 的想法是以"正确"的方式做事。组织的文化可以通过其一致的角色建模、教学和辅导来将其描述为其 Kata。	DevOps 基金会
激励模型	旨在激励员工完成任务 以实现目标的系统。系统可能会对动机产生积极或消极的影响。	DevSecOps 工程
事件	IT 服务的任何计划外中断或 IT 服务质量降低。包括中断或可能中断服务的事件。（ITIL 定义）	DevOps 基金会，DevSecOps 工程
事件 管理	尽快恢复正常服务操作的过程，以最大限度地减少业务影响， 并确保保持商定的服务质量级别。（ITIL 定义）。涉及捕获服务事件的谁、什么、何时发生，以及继续使用此数据以确保服务级别目标得到满足。	DevOps 基金会，DevSecOps Engineering ， 现场可靠性工程
事件响应	处理和管理安全漏洞或攻击的后果（也称为事件）的有组织的方法。目标是以限制损害和减少恢复时间和 成本的方式处理这种情况。	DevSecOps 工程，现场可靠性工程
增量	可能可交付完成的工作是冲刺 （Sprint） 的结果。	经过认证的敏捷服务经理，DevOps 基金会
增量推出	增量部署意味着对服务部署许多小型的渐进式更改，而不是一些大型更改。用户将逐渐移动到服务的新版本，直到最终所有用户都跨移动。有时由 colored 环境（例如蓝色/绿色部署）引用。	站点可靠性工程

基础 设施	开发、测试、交付、监视和控制或支持 IT 服务所需的所有硬件、软件、网络、设施等。术语 IT 基础结构包括所有信息技术，但不包括相关人员、流程和文档。（ITIL 定义）	DevOps 基金会，DevSecOps 工程
基础架构作为代码	使用代码（脚本）来配置 和管理基础结构的做法。	DevOps 基金会，DevSecOps 工程
基础设施测试	测试的目的是验证 EUT 操作的框架。例如，验证目标环境中的特定操作系统 实用程序功能。	连续交付架构，DevOps 测试工程
基础设施=作为+服务-（IaaS）	按需访问可配置计算资源的共享池。	连续交付架构，DevOps 测试工程
集成开发环境 （IDE）	集成开发环境 （IDE） 是一个软件套件，它整合了开发人员编写和测试软件所需的基本工具。通常，IDE 包含代码编辑器、编译器或解释器以及开发人员通过单个图形用户界面 （GUI） 访问的调试器。IDE 可能是独立应用程序，也可以包含在一个或多个现有和兼容的应用程序中。（技术目标）	DevSecOps 工程
集成开发环境 （IDE） "lint" 检查	Linting 是运行一个程序的过程，该程序将分析代码的潜在错误（例如，格式差异、不遵守编码标准和约定、逻辑错误）。	DevSecOps 工程
物联网	通过基于 Web 的无线服务连接到 Internet 并可能相互的物理设备网络。	DevOps 基金会，DevSecOps 工程
内部自动化	脚本 和自动化作为服务的一部分提供，旨在减少辛劳。	站点可靠性工程
投资	Bill Wake 创建了一个半音，以提醒高质量的用户故事的特征。	认证敏捷服务经理
ISO 31000	提供风险管理原则和通用准则的一系列标准。	DevSecOps 工程

ISO/IEC 20000	IT 服务管理的国际标准。ISO/IEC 20000 用于审核和认证服务 管理功能。	DevOps 基金会
问题管理	在整个软件开发生命周期中捕获、跟踪和解决 Bug 和问题的过程。	DevSecOps 工程
IT 基础设施库 （ITIL）	用于 IT 服务管理的最佳实践出版物集。出版于一系列五本核心书籍，代表 IT 服务生命周期的各个阶段，这些阶段包括：服务策略、服务设计、服务转型、服务运营和持续服务改进。	经过认证的敏捷流程所有者
IT 服务	从 IT 组织提供给客户的服务。	DevOps 基金会
IT 服务管理 （ITSM）	实施和管理满足业务需求的高质量 IT 服务。（ITIL 定义）	经过认证的敏捷流程所有者，站点可靠性工程
itest	由 Spirent 通信公司许可用于创建自动测试用例的工具。	DevOps 测试工程
Itil	用于 IT 服务管理的最佳实践出版物 集。以一系列五本核心书籍出版，这些核心书籍代表了 IT 服务生命周期的各个阶段，包括：服务策略、服务设计、服务转型、服务运营和持续服务改进。	经过认证的敏捷服务经理，DevOps基金会，站点可靠性工程
詹金斯	詹金斯是一个免费软件工具。它是最流行的主自动化框架工具，特别是对于连续集成任务自动化。Jenkins 任务自动化围绕时分 流程进行。许多测试工具和其他工具提供插件，以简化与 Jenkins 的集成。	连续交付架构，DevOps 测试工程
改善	持续改进的实践。	DevOps 基金会
看板	以可 管理的速度将工作流程拉向流程的工作方法。	经过认证的敏捷服务经理，DevOps 基金会
看板	帮助团队组织、可视化和管理工作的工具。	DevOps 基金会

卡普曼戏剧 三角	戏剧 三角是人类互动的社会模式。三角形映射了冲突中的人之间可能发生的破坏互类型。	DevOps 领导者
关键指标	用于帮助管理流程、IT 服务或活动进行测量 和报告。	DevOps 基金会, DevOps 领导者
关键性能指标	用于衡量关键成功因素成就的关键指标。KPI 是关键成功因素的基础，以百分比衡量。	经过认证 的敏捷流程所有者，认证敏捷服务经理
关键性能指标 （KPI）	用于衡量关键成功因素成就的关键指标。KPI 是关键成功因素的基础，以百分比衡量。（ITIL 定义）	DevOps 基金会认证 Ag ile 服务经理
关键字=基于	测试用例使用引用可用于测试的程序的预定义名称创建。	连续交付架构, DevOps 测试工程
知识 管理	确保正确信息 在正确的时间传递到正确地点或人员的过程，以便做出明智的决策。	德沃普 基金会， DevSecOps 工程
已知错误	记录的根本原因和解决方法有问题。（ITIL 定义）	DevOps基金会, DevSecOps 工程
科尔布的学习风格	大卫·科尔布于1984年出版了他的学习风格模型;他的体验式学习理论 在两个层次上起作用：一个 四个阶段 的学习周期和四个独立的学习 风格。	DevOps 领导者
科特的双操作系统	John Kotter 描述了对双操作系统需求，该系统将网络的创业能力与传统 层次结构 的组织效率相结合。	DevOps 领导者
库贝内德斯	Kubernetes 是一个用于自动化应用程序部署、扩展和管理的开源容器编排系统。它最初由 谷歌设计， 现在由云本机计算基金会维护。	站点可靠性 工程师g
库布勒-罗斯变化曲线	描述和预测个人和组织对重大变化的反应阶段。	DevOps 基金会

实验室=作为=a- 服务 （LaaS）	云计算服务类别， 提供实验室 ， 使客户无需构建和维护实验室基础设施的复杂性即可测试应用程序。	连续交付架构， DevOps 测试工程
拉卢克斯 （文化模型）	弗雷德里克 ·拉卢克斯创造了一个理解组织文化的模型。	DevSecOps 工程
延迟	延迟是通信消息时发生的延迟， 时间 消息在接收初始请求（例如服务器接收请求）和客户端接收响应 之间 "按线"进行。	站点可靠性工 程
系统思维定律	彼得 ·森格在他的著作《第五条学科》中概述了11条法律将有助于理解商业系统， 并确定解决复杂业务问题的行为。	DevOps 领导者
精 益	生产理念， 专注于减少浪费和改善流程流程， 提高整体客户价值。	DevOps 领导者
精益（形容词）	备用的， 经济实惠的。缺乏丰富或 富足。	DevOps 基金会， DevSecOps 工程
精益（生产）	生产理念， 专注于减少浪费和改善流程流程， 提高整体客户价值。	DevOps 基金会， DevSecOps 工程
精益画布	精益画布是一个 1 页的商业计划模板。	DevOps 领导者
精益企业	战略性地将精益生产背后的关键理念应用于整个企业的组织。	DevOps 基金会， DevSecOps 工程
精益	将精益 生产背后的关键理念应用于 IT 产品和服务的开发和管理。	DevOps 基金会， DevSecOps 工程
精益制造	精益生产理念主要源自丰田生产系统。	DevOps基金会， DevSecOps 工程
精益 产品开发	精益产品开发（LPD）利用精益原则来迎接产品开发的挑战。	DevOps 领导者
精益六西格玛	通过消除"浪费"和减少 "缺陷"， 将精益制造和六西格玛的概念相结合的管理方法。	经过认证的敏捷流程所 有者

精益创业	用于以最有效的方式开发业务或产品的系统，以降低故障风险。	DevOps 领导者
精益思维	精益思维的目标是以更少的资源、更少的浪费为客户创造更多的价值。废物被视为任何不增加工艺价值的活动。	认证敏捷服务经理
许可证扫描	使用 Blackduck 和 概要等工具，检查依赖项的许可证是否与您的应用程序兼容， 并 批准或将其列入黑名单。	站点可靠性工程
小定律	约翰·利蒂的一个定理指出，固定系统中客户的长期平均L L数等于长期平均有效到达速率= 乘以客户在系统中花费的平均 W 时间。	DevOps 领导者
负载运行器	用于测试应用、测量系统行为和负载下 性能的工具。由惠普 授权。	连续交付架构, DevOps 测试工程
日志	详细情况的序列化报告，如测试活动和 EUT 控制台 日志。	连续交付架构, DevOps 测试工程
日志管理	用于管理和便利生成、传输、分析、存储、存档和最终处置在信息系统 内创建的大量日志数据的集体流程和政策。	DevSecOps 工程
测 井	捕获、聚合和存储与系统性能相关的所有日志，包括但不限于进程调用、事件、用户数据、响应、错误或状态代码。洛斯塔什和纳吉奥是受欢迎的例子。	站点可靠性工程
逻辑炸弹（渣码）	满足编程条件时用于对系统造成伤害的 一串恶意代码 。	DevSecOps 工程
长寿测试	测试的目的是确定整个系统是否长时间按预期运行	连续交付架构, DevOps 测试工程
机器学习	使用从数据中学习的算法的 数据分析。	DevOps 基金会

恶意 软件	一种程序，旨在访问计算机系统，通常为某些第三方的利益，未经用户许可	DevSecOps 工程
多重 身份验证	使用至少 2 个因素进行身份验证的做法。这两个因素可以是同一类的。	DevSecOps 工程
部署之间的平均时间	用于测量部署频率。	DevOps 基金会，DevSecOps 工程
平均故障之间的时间（MTBF）	CI 或 IT 服务可以不间断地执行其商定功能的平均时间。通常用于测量可靠性。从 CI 或服务开始工作到失败（正常运行时间）的测量。（ITIL 定义）	DevOps 基金会，DevsecOps 工程
检测缺陷平均时间（MTTD）	检测故障组件或设备所需的平均时间。	连续交付架构、DevOps 基础、DevSecOps 工程、现场可靠性工程
平均发现时间	漏洞或软件错误/缺陷在识别之前存在多长时间。	DevSecOps 工程
平均修补时间	发现漏洞后，将修补程序应用于环境需要多长时间。	DevSecOps 工程
平均维修 时间（MTTR）	修复故障组件或设备所需的平均时间。MTTR 不包括恢复或恢复服务所需的时间。	DevOps 基金会，DevSecOps 工程
平均分辨率时间（MTTRe）	生产影响问题 需要多久才能解决。	DevSecOps 工程，现场可靠性工程
平均恢复服务时间（MTRS）	用于测量从 CI 或 IT 服务发生故障到完全还原并交付其正常功能（停机时间）的时间。通常用于测量可维护性。（ITIL 定义）。	DevOps 基金会，DevSecOps 工程， 现场可靠性工程
心理模型	心理模型是解释某人在现实世界中如何思考某事的思维过程。	DevOps 领导者
合并	将软件集成在一起的操作将一起转换为软件版本管理系统。	连续交付架构，DevOps 测试工程

度量	用于帮助管理 流程、IT 服务或活动进行测量和报告。	DevOps 基金会, DevSecOps 工程
指标	这是一类与用于监视产品或基础结构运行状况的测量相关的术语。	连续交付 架构, DevOps 测试工程
微服务	一种软件体系结构，由通过 API 交互的小型模块组成，可以在不影响整个系统的情况下进行更新。	DevOps 基金会
心态	一个人通常的态度 或精神状态就是他们的心态。	DevOps 领导者
最低关键活动	必须执行的活动，以提供符合给定流程的证据。	经过认证的敏捷流程所有者
最低可行产品	大多数 最简码的产品版本 ， 可以发布，仍然提供足够的价值，人们愿意使用它 。	经过认证的敏捷服务经理, DevOps 基金会，DevOps 领导者
模拟对象	模拟是一种以受控方式模拟真实方法 /对象行为的方法/对象。模拟对象用于单元测试。通常，测试下的方法调用其中的其他外部服务或方法。这些称为依赖项。	连续交付架构, DevOps 测试工程
模型	系统、流程、IT 服务、CI 等的表示，用于帮助理解或预测未来行为。在流程上下文中，模型表示用于处理特定类型的事务的预定义步骤。	DevSecOps 工程
模型=基于	测试用例自动派生自受测试实体的模型。示例工具： 三centus	连续交付架构, DevOps 测试工程
监测	使用硬件或软件组件来监视系统 资源和计算机服务的性能。	站点可靠性工程
监控工具	允许 IT 组织识别特定发布的特定问题并了解对最终用户的影响的工具。	DevOps 领导者

整体	如果软件系统具有单一的体系结构，则称为"整体"，其中功能上可区分的方面（例如数据输入和输出、数据处理、错误处理和用户界面）都是交织在一起的，而不是包含体系结构上独立的组件的 than。	连续交付架构
多重 身份验证	使用 2 个或更多因素进行身份验证 的做法。常与 2 因素身份验证同义。	DevSecOps 工程
多云-cloud	多云DevOps 解决方案提供- 对开发和测试环境的按需多租户访问。	连续交付架构
网络可靠性工程师（NRE）	应用可靠性工程 方法测量和自动化网络可靠性的人。	站点可靠性工程
神经可塑性	描述大脑形成和重组突触连接的能力，特别是在对学习、经验或 受伤后的反应。	DevOps 领导者
神经	大脑和神经系统的研究。	DevOps 领导者
非功能 性要求	指定可用于判断系统操作的标准的要求，而不是用于判断 特定行为或功能（例如可用性、可靠性、可维护性、可支持性）;系统的质量。	DevOps 基金会
非功能测试	定义为一种服务测试，用于检查软件服务的性能 、可用性和可靠性等非功能性方面。	站点可靠性工程
正在测试的对象 （OUT）	EUT 是一个软件对象或对象类。	连续交付架构，DevOps 测试工程
目的	过程的目标或 目标。	经过认证的敏捷流程所有者
可观	可观察性侧重于尽可能多地外化有关整个服务的数据，使我们能够推断该服务的当前状态。	站点可靠性工程

通话中	即用电话意味着有人在设定的时间段内 有空， 并准备在适当紧迫的时间内对生产事件做出响应。	站点可靠性工程
开源	软件及其 源代码一起分发，以便最终用户组织和供应商可以出于自己的目的对其进行修改。	DevOps 基金会，DevSecOps 工程
运营级别协议	IT 服务提供商与同一组织的另一部分 之间的协议。（ITIL 定义）	经过认证的敏捷流程所有者
操作（操作）	参与部署和管理质量保证分析师、发布经理、系统和网络管理员 、信息安全干事、IT 运营专家和服务台分析员等系统和服务所需的日常业务活动的个人。	连续交付架构
运营 管理	执行在商定的级别交付和支持 IT 服务和支持 IT 基础结构所需的日常 活动的功能。 （ITIL）	DevSecOps 工程
老年 退休金 计划	参与部署和管理质量保证分析师、发布经理、系统和网络 管理员、信息安全干事、IT 运营专家和服务台分析员等系统和服务所需的日常业务活动的个人。	DevOps 基金会，DevSecOps 工程
编排	一种构建自动化的方法， 它将多个工具连接或"协调"在一起，形成工具链。	DevOps 基金会，DevSecOps 工程
组织文化	共同的价值观、假设、信念和规范体系，将组织成员团结在一起。	DevOps 领导者
组织模型	对于 DevOps，一种为 Spotify 的小队方法进行模型化的方法，用于组织 IT。	DevOps 领导者
组织 变革	努力调整组织内人类的行为，以满足新的结构、流程或 要求。	DevOps 基金会，DevSecOps 工程
操作系统虚拟化	一种将服务器拆分为多个分区（称为"容器"或"虚拟环境"）的方法，以防止应用程序相互干扰。	DevOps 基金会

结果	预期结果或实际结果。	DevOps 基金会, DevSecOps 工程
输出	流程活动（如信息、计划、文档、记录、报告等）产生的交付成果。	经过认证的 敏捷流程所有者
包注册表	软件包、工件及其相应元数据的存储库。可以存储组织本身或第三方二进制文件生成的文件。文物和连结是最流行的。	站点可靠性工程
页面	作为 CI/CD 管道的一部分自动创建支持 网页的内容。	站点可靠性工程
补丁	旨在解决（缓解/补救）错误或弱点的软件更新。	DevSecOps 工程
修补程序管理	识别和实施修补程序的过程。	DevSecOps 工程
病理文化	病理文化往往把信息视为个人资源，用于政治权力斗争(（Westrum）。).	DevOps 领导者，站点可靠性工程
渗透测试	对计算机系统进行的授权模拟攻击，该攻击可寻找安全漏洞，从而可能访问系统的功能和数据。	DevSecOps 工程
人员变动	专注于改变员工的态度、行为、技能或绩效。	DevOps 领导者
性能测试	测试的目的是确定 EUT 满足其系统性能标准，或确定 系统的性能能力。	连续交付架构, DevOps 测试工程
计划	正式的、经过批准的文档，用于描述实现结果所需的功能和资源。	经过认证的敏捷流程所有者
计划-执行检查-行为	流程管理和改进的四阶段周期由 W. Edwards Deming 负责。有时称为德明周期或 PDCA。	经过认证的敏捷流程所有者，认证敏捷服务 DevOps 基础经理, DevSecOps 工程
平台=作为+a服务（PaaS）	云计算服务类别，提供一个平台，使客户能够开发、运行和管理应用程序，而无需构建和维护基础设施的复杂性。	连续交付架构, DevOps 测试工程

插件	业务流程-工具和其他工具之间的预编程集成。例如，许多工具提供插件来与 Jenkins 集成。	连续交付 架构， DevOps 测试工程
政策	正式文档，根据组织作为其操作的一部分可能或可能不会做什么来定义边界。	DevOps 基金会， DevSecOps 工程
政策	描述 服务提供商的总体意图和方向的正式文件，如高级管理层所表达的那样。	经过认证的敏捷流程所有者
策略作为代码	安全原则和概念可以在代码（例如软件、配置管理、 自动化）中阐明，这一概念足以大大减少了对广泛传统策略框架的需要。标准和准则应在代码和配置中实施，在合规性、差异 或可疑违规行为方面自动实施和自动报告。	DevSecOps 工程
实施后审查 （PIR）	实施变更或项目后进行的审查，以评估更改是否成功和改进机会。	经过认证的敏捷服务经理， DevOps 基金会
可能可发货的产品	增加"完成"的工作，如果这样做有意义，可以被释放。	经过认证的敏捷服务经理， Devops 基金会
飞行前 +飞行	这是一类术语，它是指在集成到主干分支之前在 EUT 上执行的活动和进程的名称。	连续交付架构， DevOps 测试工程
优先	事件 、问题或变化的相对重要性;基于影响和紧迫性。（ITIL 定义）	DevOps 基金会， DevSecOps 工程
特权访问管理 （PAM ）	通过保护、管理和监视特权帐户和 访问权限，帮助组织提供对关键资产的安全特权访问并满足合规性要求的技术。（加特纳）	DevSecOps 工程
问题	一个或多个事件的根本原因。（ITIL 定义）	DevOps 基金会， DevSecOps 工程
程序	步骤-+by +步骤说明，说明如何在流程中执行活动。	认证敏捷服务经理

过程	旨在实现特定目标的结构化活动集。流程 需要输入并转换为定义的输出。采取特定投入并产生对客户有价值的特定输出的相关工作活动。	经过认证的敏捷服务经理，DevOps 基金会，DevSecOps 工程
过程B acklog	对于流程（包括当前和未来要求）需要设计或改进的所有内容的优先级列表。	认证敏捷服务经理
流程更改	专注于标准 IT 流程的更改，如软件开发 实践、ITIL 流程、变更管理、审批等。	DevOps 领导者
流程客户	进程输出的接收者。	认证敏捷服务经理
流程改进团队	设计或重新设计流程并确定 如何最好地在整个组织中实施新流程的个人团队。	经过认证的敏捷流程所有者
流程管理器	负责流程运营（日常）管理的个人。	经过认证的敏捷流程所有者
流程所有者	角色负责流程的整体质量。可以分配给执行流程管理器角色的同一个人，但两个角色在较大的组织中可能是分开的。（ITIL 定义）	DevOps 基金会，DevSecOps 工程，认证敏捷服务经理
流程所有者	负责流程整体质量的人和流程积压工作的所有者。	认证敏捷服务经理
流程规划会议	确定流程的目标、 目标、投入、成果、活动、利益相关者、工具和其他方面的高级别活动。此会议未进行时间框。	认证敏捷服务经理
流程供应商	流程输入的创建者。	认证敏捷服务 经理
处理时间	一个或多个输入通过制造或开发过程转化为成品的期间。（商业词典）	DevOps 领导者
产品积压工作	系统的功能和非功能 要求的优先级列表通常以用户情景表示。	经过认证的敏捷流程所有者、经过认证的敏捷服务经理、DevOps 基金会

产品积压工作优化	向积压工作项添加详细信息、估计和 顺序的持续过程。有时称为产品积压工作整理。	认证敏捷服务经理
产品所有者	负责最大化产品价值和管理产品积压工作的个人。优先级、整理和拥有积压 工作。给球队目标。	经过认证的敏捷流程所有者、经过认证的敏捷服务经理、DevOps 基金会、DevOps 领导者
编程=基于	测试用例是通过使用编程语言编写代码创建的。例如 JavaScript、 Python 、 TCL 、 Ruby	连续交付架构, DevOps 测试工程
项目	为创造独特的产品、服务或结果而进行的临时努力。	经过认证的敏捷流程所有者
预配平台	提供用于预配基础结构的平台 的工具（例如，木偶、厨师、盐）。	DevOps 领导者
心理安全	心理安全是团队对人际冒险安全的共同信念。	DevOps 领导者
QTP	快速测试专业人员是软件应用程序的功能和 回归测试自动化工具。由惠普授权。	DevOps 测试工程
质量管理	处理测试用例规划、测试执行、缺陷跟踪（通常为积压工作）、严重性和优先级分析的工具。CA 的敏捷中心	站点可靠性工程
拉基矩阵	将角色和责任映射到流程或项目的活动。	经过认证的敏捷流程所有者
拉诺雷克斯	用于测试桌面、基于 Web + 和移动应用程序的 GUI测试自动化框架。由拉诺雷克斯许可。	DevOps 测试工程
勒索软件	加密用户设备上的文件 网络的存储设备。恢复访问权限 对于加密文件，用户必须向网络罪犯支付"赎金"，通常通过 难以追踪的电子支付方式（如比特币）。	DevSecOps 工程
回归测试	测试的目的是确定 EUT 的新版本是否损坏了以前有效的东西。	连续交付 架构, DevOps 测试工程
法规合规性测试	测试的目的是确定 EUT 是否符合特定的法规要求。例如，验证 EUT 是否满足政府有关消费者信用卡处理 的规定。	连续交付架构, DevOps 测试工程

释放	在生产环境中构建、测试和部署的软件。	连续交付架构、DevOps 基础、DevSecOps 工程
发布验收标准	发布包的可测量属性，用于确定发布候选版本是否可以部署到客户。	连续交付架构, DevOps 测试工程
发布候选	已 为部署准备的发布包可能已通过，也可能未通过该版本。	连续交付架构, DevOps 测试工程
发布 治理	发布治理是所有关于控制和自动化（安全性、合规性或其他），以确保您的版本以可审核 和可跟踪的方式进行管理，以满足业务了解正在发生变化的需要。	站点可靠性工程
发布 管理	管理发布 并支撑持续交付和部署管道的过程。	DevOps 基金会, DevSecOps 工程
发布业务流程	通常 为部署管道，用于检测任何会导致生产问题的更改。 协调其他工具将识别性能、安全性或可用性问题。像詹金斯和Gitlab CI这样的工具可以"协调"发布。	站点可靠性工程
发布计划会议	时间盒事件，确定发布的目标 、风险、功能、功能、交付日期和成本。它还包括确定产品积压工作优先级。	经过认证的敏捷流程所有者，认证敏捷服务经理
关联	持续测试原则，强调在最重要的测试和测试结果上优先考虑 f ocus	连续交付架构, DevOps 测试工程
可靠性	衡量服务、组件或 CI 可以不间断地执行其商定功能的时间。通常以 MTBF 或 MTBSI 进行测量。（ITIL 定义）	DevOps 基金会, DevSecOps 工程， 现场可靠性工程
可靠性测试	测试的目的是确定整个系统在长时间内在紧张和加载的条件下 是否按预期运行。	连续交付架构, DevOps 测试工程
修复	解决 DevOps 进程中发现的问题的操作。例如，回滚EUT更改的更改，导致 CT 测试用例失败。	连续交付架构, DevOps 测试工程

补救计划	确定更改或释放失败后要执行的操作的计划。（ITIL 定义）	DevOps 基金会, DevSecOps 工程
更改请求 （RFC）	正式建议做出改变。术语 RFC 经常被误用为表示更改记录或更改本身。（ITIL 定义）	DevOps 基金会
需求 管理	工具比处理需求定义、可追溯性、层次结构和依赖关系。 通常还处理代码要求和测试用例的要求。	站点可靠性工程
弹性	构建对更改和事件具有容忍性的环境或组织。	DevSecOps 工程，现场可靠性 工程
响应时间	响应时间是用户发出请求直至收到响应时的总时间。	站点可靠性工程
休息	表示状态转移。世界软件架构风格+ 万维网。	连续交付架构, DevOps 测试工程
宁静的 API	网络上的表示状态传输 （REST） 或 RESTful 服务（如 HTTP）提供了可扩展的互操作性，用于请求系统快速可靠地访问和使用无状态操作（GET、POST、PUT、DELETE 等）对资源进行文本表示（XML、HTML、JSON）。	连续交付架构
RESTful 接口测试	测试的目的是确定 API 是否满足其设计 标准和 REST 体系结构的期望。	连续交付架构, DevOps 测试工程
投资回报 （ROI）	所获得的利益与实现该收益的成本之间的差额，以百分比 表示。	DevOps 基金会, DevSecOps 工程
查看应用	允许实时提交和启动代码 – 环境被启动，使开发人员能够查看其应用程序。	站点可靠性工程
返工	纠正缺陷 （浪费）所需的时间和精力。	DevOps 领导者

风险	可能导致伤害或损失或影响组织实现其目标的能力的事件。风险管理包括三项活动：识别 风险、分析风险和管理风险。未来 损失 的频率和可能幅度。涉及可能造成伤害或损失或影响组织执行或实现其目标的能力的事件。	DevOps 基金会, DevSecOps 工程
风险事件	可能导致伤害或损失或影响组织实现其目标的能力的事件。风险管理包括三项活动：识别风险、分析风险和降低风险。	DevOps 领导者
风险管理流程	"风险"的上下文化、评估和处理的过程。从 ISO 31000： 1） 建立上下文， 2） 评估风险， 3） 处理风险（补救、减少或接受）。	DevSecOps 工程
机器人框架	由谷歌创建和支持的 TDD 框架。	连续交付架构, DevOps 测试工程
作用	授予某人或团队的职责、活动和权限。角色由流程 定义。一个人或团队可能有多个角色。分配给用户或用户组的一组权限，允许用户在系统或应用程序中 执行操作 。	DevOps 基金会, DevSecOps 工程
基于角色的访问控制 （RBAC）	限制系统访问授权用户的方法。	DevSecOps 工程
回滚=返回	已集成的软件更改将从集成中删除。	连续交付架构, DevOps 测试工程
根本原因 分析 （RCA）	采取措施确定问题或事件的根本原因。	DevOps 基金会, DevSecOps 工程
坚固的 开发 （开发）	坚固开发 （DevOps） 是一种在连续交付管道中尽早包含安全实践的方法， 用于提高网络安全、速度和质量， 而这超出了 DevOps 实践单独产生的范围。	DevOps 基金会

坚固的 DevOps	坚固耐用的 DevOps 是一种在连续交付管道中尽早包含安全实践的方法，用于 提高网络安全、速度和质量，超出 DevOps 实践单独产生的范围。	连续交付架构, DevOps 测试工程
运行手册	服务顺利运行所需的过程集合。以前手动的性质，他们现在通常自动化的工具，如安西布尔。	站点可靠性工 程
运行时应用程序自我保护 （RASP）	在利用漏洞之前，主动 moni tor 并阻止生产环境中的威胁的工具。	站点可靠性工 程
理智测试	一组非常基本的测试，用于确定软件是否正常工作。	连续交付 架构, DevOps 测试工程
可 伸缩 性	可伸缩性是服务的特点， 它描述了在增加或扩展负载下应对和执行任务 的能力。	站点可靠性工 程
扩展敏捷框架 （ SAFE）	一个经过验证的公开框架，用于在企业级应用精益敏捷原则和实践。	DevOps 基金会
围巾模型	神经科学中关于人们社交方式的重要发现摘要。	DevOps 领导者
调度	计划：将变更发布到生产中的计划流程。	DevOps 领导者
Scrum	在复杂项目上进行有效团队协作的简单框架。Scrum 提供了一组小规则，这些规则创建"足够"的结构，使团队能够专注于创新，解决本来是 无法逾越的问题 挑战。（Scrum.org）	经过认证的敏捷服务经理, DevOps 基金会
Scrum 项目	产品积压、冲刺积压、燃尽 图、产品增量	经过认证的敏捷流程所 有者
Scrum 组件	Scrum 的角色、事件、工件和将它们绑定在一起的规则。	认证敏捷服务经理
Scrum 事件	发布计划会议（可选）、冲刺计划 会议、冲刺、每日 Scrum、 冲刺审核、冲刺回顾	经过认证的敏捷流程所 有者
Scrum 指南	Scrum 概念和实践的定义，由肯·施 瓦伯和杰夫·萨瑟兰编写。	认证敏捷服务经理

Scrum 支柱	维护 Scrum 框架的支柱，包括：透明度、检查和适应。	经过认证的敏捷流程所有者
Scrum 角色	产品所有者、开发团队（团队）和 Scrum 主管。另请参阅敏捷服务管理角色。	经过认证的敏捷流程所有者
Scrum 团队	一个自我组织、跨职能的团队，使用 Scrum 框架以迭代和增量方式交付产品。Scrum 团队由产品所有者、 开发团队和 Scrum 主管组成 。	DevOps 基金会
Scrum 值	一套基本价值观和品质支撑着Scrum框架：承诺、专注、开放、尊重和勇气。	经过认证的敏捷流程所有者，认证敏捷服务经理
ScrumMaster	为 Scrum 提供流程领导（即确保了解和遵循 Scrum 实践）的个人，以及通过消除障碍支持 Scrum 团队的个人。	DevOps 基金会
秘密检测	秘密检测旨在 防止该敏感信息（如密码、身份验证令牌和私钥）作为存储库内容的一部分无意中泄露。	站点可靠性工程
秘密 管理	机密管理是指用于管理 数字身份验证凭据（机密）的工具和方法，包括密码、密钥、API 和令牌，用于应用程序、服务、特权帐户和 IT 生态系统的其他敏感部分。	站点可靠性工程
安全自动化	安全自动化通过保护整个交付管道中使用的 工具，消除了人为错误（和故意破坏）的可能性。	站点可靠性工程
安全（信息安全）	旨在保护计算机系统数据的 机密性、完整性和可用性免受恶意意图者攻击的做法。	DevOps 基金会，DevSecOps 工程
安全性为代码	将安全性自动化并构建到 DevOps 工具和实践中， 使其成为工具链和工作流的重要组成部分。	DevOps 基金会，DevSecOps 工程

安全测试	测试的目的是确定 EUT 是否满足其安全要求。例如，确定 EUT 是否正确处理 登录凭据的测试。	连续交付架构， DevOps 测试工程
硒	适用于软件测试GUI 和 Web 应用程序的热门开源工具。	连续交付架构， DevOps 测试工程
自我修复	自我修复意味着服务和底层环境自动检测和解决问题的能力。它消除了人工干预的需要。	站点可靠性工 程
自组织团队	管理原则，其中团队选择如何最好地完成他们的工作，而不是由团队外部的其他人指导。自我组织发生在边界内和针对给定目标（即做什么）。	经过认证的敏捷流程所 有者
自我+组织	团队自主组织工作的管理原则。自我组织 发生在边界内和针对给定目标。团队选择如何最好地完成他们的工作，而不是由团队外的其他人 指导。	认证敏捷服务经理
无服务器	代码执行范例不需要底层基础结构或依赖项，此外，一段代码由负责创建执行环境的服务提供商（通常是 云）执行。AWS 和 Azure 函数中的 Lambda 函数就是示例。	站点可靠性工 程
服务	通过促进客户希望实现的结果，而无需承担特定成本和风险， 为客户提供价值的方法。	DevOps 基金会， DevSecOps 工程
服务目录	服务组合的子集，由实时或可用于部署的服务组成。有两个方面：业务/客户服务目录 （客户可见）和技术/支持服务目录。（ITIL 定义）	DevOps 基金会
服务设计	ITIL 核心出版物之一和服务生命周期的一个阶段。	DevOps 基金会

服务台	服务提供商和用户之间的单点联系。"现在服务"等工具用于管理服务的生命周期以及内部和外部利益相关者的参与。	DevOps 基金会
服务级别协议 （SLA）	IT 服务提供商与其客户之间的书面协议，确定双方的关键服务目标和职责。SLA 可能涵盖多个服务或客户。（ITIL 定义）	经过认证的敏捷流程所有者、DevOps 基金会、站点再责任工程
服务级别指示器 （SLI）	SLI 用于传达有关服务的定量数据，通常用于测量服务对 SLO 的性能。	站点可靠性工程
服务级别 管理	确保所有 当前和计划好的 IT 服务都交付到商定的可实现目标的过程。（ITIL 定义）	经过认证的敏捷流程所有者
服务级别目标 （SLO）	SLO 是产品或服务应运行良好的目标。SLO 的设置 基于组织对服务的期望。	站点可靠性工程
服务生命周期	ITIL 核心指南的结构。	DevOps 基金会
服务 管理	一套专门的组织能力，以 服务的形式为客户提供价值。（ITIL 定义）	DevOps 基金会
服务管理办公室	协调整个生命周期中管理服务提供商服务的所有流程和函数的功能。流程所有者 可以直接或通过"虚线"报告行向 SMO 报告。	经过认证的敏捷流程所有者
服务操作	ITIL 核心出版物之一和服务生命周期的一个阶段。	DevOps 基金会
服务提供商	向一 个或多个内部或外部客户提供服务的组织。（ITIL 定义）	DevOps 基金会
服务请求	用户从 IT 服务提供商处请求标准服务。（ITIL 定义）	DevOps 基金会
服务策略	ITIL 核心出版物 之一和服务生命周期的一个阶段。	DevOps 基金会
服务转换	ITIL 核心出版物之一和服务生命周期的一个阶段。	DevOps 基金会

DevOps 的七大支柱	七个独特的"支柱" 为 DevOps 系统奠定了基础，这些系统包括协作文化、DevOps 设计、持续集成、持续测试、持续交付和部署、持续监控和弹性基础架构和工具。	连续交付架构
向左移动	一种通过尽早且经常地将测试纳入软件开发过程而努力将质量纳入软件开发过程的方法。这个概念延伸到安全架构、强化映像、应用程序安全性、甚至其他方面。	DevOps 基金会，DevSecOps 工程
丝绸测试	企业应用程序的自动化功能和回归测试。由博兰许可。	DevOps 测试工程
西米安军	西米安军队是由 Netflix 设计的一套导致故障的工具。最著名的例子是混沌猴子，它随机终止生产中的服务，作为混沌工程方法的一部分。	站点可靠性工程
站点可靠性工程 （SRE）	该 学科包含软件工程的各个方面，并将其应用于基础结构和操作问题。主要目标是创建可扩展且高度可靠的软件系统。	站点可靠性工程
六西格玛	严格的数据驱动方法，通过测量标准偏差来减少缺陷。	经过认证的敏捷流程所有者
智能目标	具体、可衡量、可实现、相关和有时限的目标。	德沃普 基金会
烟雾测试	一组基本功能测试，在构建软件组件后立即运行。与 CI 回归测试相同。	连续交付架构，DevOps 测试工程
快照	特定生成通过/未执行 结果的报告。	连续交付架构，DevOps 测试工程
片段	存储和共享代码片段，允许围绕特定代码段进行协作。还允许在其他代码库中使用的 代码片段。比特桶和 Gitlab allow 这个。	站点可靠性工程
肥皂	简单对象访问协议 （SOAP） 是基于 XML 的消息传递协议，用于在计算机之间交换信息。	连续交付架构，DevOps 测试工程

软件 组成 分析	一种工具，用于检查源代码中具有已知漏洞的库或函数。	DevSecOps 工程
软件定义网络 （SDN）	软件定义网络 （SDN） 是一种网络 架构方法，它使网络能够使用软件应用程序进行智能和集中控制，或"编程"。	站点可靠性工程
软件交付生命周期（SDLC）	用于设计、开发和测试高质量软件的过程。	DevOps 领导者，站点可靠性工程
软件版本管理系统	用于管理软件更改的存储库工具。例如：Azure DevOps, BitBucket, Git, GitHub, GitLab， VSTS.	连续交付架构, DevOps 测试 工程
软件=作为+服务-（SaaS）	以订阅为许可软件的云计算服务类别。	DevOps 基础、连续交付架构、DevOps 测试工程
源代码工具	用于控制关键资产（应用程序和基础结构）的源代码的存储库作为单一的真理来源。	DevOps 基金会, DevOps 领导者
Spotify 小队模型	一种组织模型，可帮助大型组织中的团队像 初创公司一 样运行， 并且具有灵活性。	DevOps 基金会, DevOps 领导者
冲刺	2 ~-4 周的期间，在此期间，产品工作增加完成。	经过认证的敏捷流程所有者、经过认证的敏捷服务管理器、持续交付架构
冲刺（Scrum）	工作的时间盒迭代，在此期间实现产品功能增量。	DevOps 基金会
冲刺积压工作	表示实现冲刺目标必须完成的工作的积压工作的子集。	经过认证的敏捷流程所有者，DevOps 基金会
冲刺目标	Sprint 的目的和目标，通常表示为要解决的业务问题。	经过认证的敏捷流程所有者，认证敏捷服务经理
冲刺计划 会议	定义 Sprint 目标、将在冲刺 （sprint） 期间完成的产品积压工作增量以及完成该任务 4 到 8 小时的时盒事件。	经过认证的敏捷流程所有者，认证敏捷服务经理

冲刺回顾	1.5 到 3 小时时箱事件，团队在其中查看上次冲刺（Sprint），并确定下一个冲刺（Sprint）的改进并确定其优先级。	经过认证的敏捷流程所有者，认证敏捷服务经理
冲刺审核	4 小时或更短的时间框事件，团队和利益干系人检查冲刺（Sprint）导致的工作并更新产品积压工作。	经过认证的敏捷流程所有者，认证敏捷服务经理
间谍 软件	在用户不知情 的情况下安装在计算机中的软件，并将有关用户计算机活动的信息传回威胁代理。	DevSecOps 工程师
队	跨职能、同地、自主、自主的团队。	DevOps 领导者
利益 相关 者	对组织、项目或 IT 服务感兴趣的人员。利益相关者可能包括客户、用户和供应商。（ITIL 定义）。	DevOps 基金会，DevSecOps 工程
稳定性	服务接受更改 的敏感性以及系统更改可能造成的负面影响。服务可能具有可靠性，因为如果功能长时间， 但 可能不容易更改，因此没有稳定性。	站点可靠性工程
站立ard 更改	预先批准的低风险变更，遵循程序或工作指导。（ITIL 定义）	DevOps 基金会，DevSecOps 工程
静态应用程序安全测试（SAST）	一种测试，用于检查源代码中的错误和 弱点。	DevSecOps 工程
静态代码分析	测试的目的是检测源代码逻辑错误和遗漏，如内存泄漏、未使用的变量、未使用的指针。	连续交付架构，DevOps 测试 工程
状态页	能够轻松地将服务状态传达给客户和用户的服务页面。	站点可靠性工程
棒	消极的激励，劝阻或惩罚不受欢迎的行为。	DevSecOps 工程
存储安全	一个特殊的安全领域，用于保护数据存储系统和生态系统以及驻留在这些系统上的数据。	站点可靠性工程
风暴堆	基于事件触发器而不是基于 时间的商业业务流程工具。	DevOps 测试工程

斯托斯塔基	这代表停止、开始和 保持：这是 一 个针对过去事件的交互式时盒 练习。	DevOps 领导者
战略冲刺	2至4 周时间盒的 Sprint， 在此期间完成流程规划会议期间定义的战略元素，以便团队能够继续设计流程的活动。	经过认证的敏捷流程所有者，认证敏捷服务经理
结构变化	权力层次、目标、结构特征、行政程序和管理制度的变化。	DevOps 领导者
供应商	负责提供 IT 服务所需的商品或服务的外部（第三方）供应商、制造商或供应商。	DevOps 基金会
综合监控	综合监视（也称为活动监视或语义监视）定期对系统运行应用程序的自动测试的子 集。结果将推送到监视服务中，在发生故障时触发警报。	连续交付架构
记录系统	记录系统是数据元素或数据实体的权威数据源。	DevOps 基金会，DevSecOps 工程
系统测试	测试的目的是确定整个系统在预期配置中是否按预期执行。	连续交付架构，DevOps 测试工程
正在测试的系统 （SUT ）	EUT 是一个完整的系统。银行出纳机正在测试中。	连续交付架构，DevOps 测试工程
标记=基于测试选择方法	测试和代码模块是预先分配的标记。为生成匹配的预分配标记选择-测试。	连续交付架构，DevOps 测试工程
目标操作模型	所需状态的说明 组织的运营 模式。	DevOps 领导者
蒂尔组织	一种新兴的组织范 式，它提倡意识水平，包括组织运作中所有以前的 世界观点。	DevOps 领导者
团队动态	衡量团队如何协同工作。包括团队文化、沟通风格 、决策能力、成员之间的信任以及团队变革的意愿。	DevOps 领导者

技术经济 范式转变	技术经济范式转变是卡洛塔 佩雷斯所设想的一般、 基于创新的经济和社会发展理论的核心。	DevOps 领导者
遥测	遥测是收集远程或无法访问点的测量或其他数据，并自动传输至接收设备进行监测。	站点可靠性工程
测试架构师	负责为 EUT 定义总体结束to测试策略的人。	连续交付架构, DevOps 测试工程
测试项目存储库	用于测试的文件 数据库。	连续交付架构, DevOps 测试工程
测试活动	测试活动可能包括一个或多个测试会话。	连续交付架构, DevOps 测试工程
测试用例	测试步骤集以及数据和 配置信息。测试用例具有测试 EUT 至少一个属性的特定用途。	连续交付架构, DevOps 测试工程
测试创建方法	这是一类测试术语，它指的是 用于创建测试用例的方法。	连续交付架构, DevOps 测试工程
测试驱动开发 （TDD）	测试驱动开发 （TDD） 是一个软件开发过程, 开发人员在编写代码 之前编写测试。然后，他们遵循以程： 1. 编写 测试 2. 运行测试和任何其他相关 测试, 并看到它们 失败 3. 编写 代码 4. 运行测试 5. 如果需要，重构 代码 6. 重复 单元级测试和/或应用程序测试是在 要测试的代码之前创建的。	持续交付架构、DevOps 基础、DevOps 测试工程
测试持续时间	运行测试需要的时间。例如，每个测试的 # 小时数	连续交付架构, DevOps 测试工程

测试环境	测试环境是指执行测试的操作系统（例如 Linus、Windows 版本等）、软件配置（例如参数选项）、动态条件（例如 CPU 和内存利用率）以及物理环境（例如电源、 冷却）。	连续交付架构, DevOps 测试工程
测试速度快	一种CT原则，指的是加速测试。	连续交付架构, DevOps 测试工程
测试框架	一组 流程、过程、抽象概念和环境，用于设计和实施自动化测试。	连续交付架构, DevOps 测试工程
测试线束	一种使测试自动化 的工具。它是指执行 测试所需的系统测试驱动程序和其他支持工具。它提供与所测试软件交互的小程序的存根和驱动程序 。	连续交付架构, DevOps 测试工程
测试 层次结构	这是一类术语，描述了将测试组织成组。	连续交付架构, DevOps 测试工程
测试方法	此类术语标识测试使用的一般方法。例如白 盒、黑盒	连续交付架构, DevOps 测试工程
测试结果存储库	测试结果数据库。	连续交付架构, DevOps 测试工程
测试结果趋势= 基于	相关因子的矩阵 根据测试结果（判断）关联测试用例和代码模块。	连续交付架构, DevOps 测试工程
测试角色	此类术语标识与测试相关的人员的一般角色和责任。	连续交付架构, DevOps 测试工程
测试脚本	自动测试用例。单个测试脚本可以根据数据实现一个或多个测试用例。	连续交付架构, DevOps 测试工程
测试 选择方法	此类术语是指用于选择要在 EUT 版本上执行的测试的方法。	连续交付架构, DevOps 测试工程
测试会话	一组在特定时间在单个生成中一 起运行的一个或多个测试套件。	连续交付架构, DevOps 测试工程

测试套件	在特定时间在单个生成上一起运行的测试用例集。	连续交付架构, DevOps 测试 工程
测试趋势	判决的历史。	连续交付架构, DevOps 测试工程
测试类型	指示测试用途的类。	连续交付架构, DevOps 测试工程
测试版本	用于测试特定生成的文件的版本。	连续交付架构, DevOps 测试工程
测试	负责测试系统或服务的个人。	连续交付架构, DevOps 测试工程
测试工具	在传递生成之前验证代码质量的工具。	DevOps 领导者
咨询流程	任何决定的人必须向受决策影响有意义的每个人和在该事项方面具有专业知识的人寻求建议。收到的建议 必须考虑到，但不必接受或遵循。咨询过程的目的不是形成共识，而是通知决策者，以便他们能够做出可能的最佳决策。不遵守建议会破坏信任， 不必要地给业务带来风险。	DevSecOps 工程
复选框陷阱	以审计为中心的视角只关注法规遵从要求的"框框"，而不考虑 总体安全目标的情况。	DevSecOps 工程
TED 的力量	TED* 的力量提供了一个替代 卡普曼戏剧 三角，其角色受害者，迫害者，和救援者。授权动态 （TED） 提供创造者、挑战者和教练的解药作用，以及更积极的应对生活挑战的方法。	DevOps 领导者
三种方式	DevOps 的关键原则 – 流程、反馈、持续实验和学习。	DevOps基金会, DevSecOps 工程， 现场可靠性工程

约束理论	确定阻碍实现目标，然后系统地改进限制，直到它不再是限制因素的最重要限制因素（即约束）的方法。	DevOps 基金会, DevSecOps 工程
托马斯·基尔曼库存（TKI）	衡量某人在某些冲突情况下的行为选择。	德沃普 基金会
威胁代理	行为人或自动执行者，对系统采取行动，意图损害或危害该系统。有时也称为"威胁参与者"。	DevSecOps 工程
威胁检测	是指检测、报告和支持响应攻击的能力。入侵检测系统和拒绝服务系统允许进行for一定程度的威胁检测和预防。	
威胁情报	与威胁的性质或威胁 所采取的行动有关的信息可能已知是所为。还可能包括与给定威胁的行动相关的"妥协指标"，以及描述如何补救给定威胁行动的"行动方案"。	DevSecOps 工程
威胁 建模	一种对潜在威胁进行排名和模型的方法，以便可以在潜在威胁所涉及到的应用程序价值上下文中理解和缓解风险。	DevSecOps 工程
上市时间	从构思想法到 客户可获得想法之间的时间段。	DevOps 领导者
价值时间	衡量企业从功能或服务实现价值所用的时间。	DevOps 基金会, DevSecOps 工程
时间跟踪	允许针对 单个问题或其他工作或项目类型的跟踪时间的工具。	站点可靠性工程
时间框	Scrum 事件的最大持续时间。	经过认证的敏捷流程所有者，认证敏捷服务经理
辛劳	一种与 运行生产服务相关的工作，往往手动、重复、自动化、战术性、缺乏持久价值。	站点可靠性工程

工具	此类介绍协调、自动化、模拟和监视 EUT 和基础结构的工具。	连续交付架构, DevOps 测试工程
工具链	一种理念，涉及使用一组集成的免费任务特定工具来自动化端到端流程（与单供应商 解决方案相比）。	DevOps 基金会
触摸时间	在精益生产 the 系统中，"接触时间"是实际使用产品的时间，并且正在增加价值。	DevOps 领导者
跟踪	跟踪提供对已部署应用程序的 性能和运行状况的洞察，跟踪处理给定请求的每个函数或微服务。	站点可靠性工程
流量	服务访问者发送和接收的数据量（例如网站或 API）。	站点可靠性工程
从房间后面开始训练	使用 4C 教学设计"地图"（连接、概念、具体实践、结论）符合敏捷价值观和原则的加速学习模型。	
转型 领导力	领导者通过吸引员工的价值观和目标感，促进大规模的组织变革，激励和激励追随者实现更高的绩效（DevOps 报告， 2017 年）。	DevOps 领导者
部落铅	高级技术领导者，在所有球队的技术领域拥有广泛而深厚的技术专业知识。一群在公共功能集、产品或服务上协同工作的小队是 Spotify 的定义中的部落。	DevOps 领导者
部落	具有长期任务的小队集合，在相关业务能力中工作。	DevOps 领导者
特洛伊木马（特洛伊木马）	恶意软件在所需的操作（如玩在线 游戏）下执行恶意操作。特洛伊木马与病毒不同，因为特洛伊木马程序将自己绑定到非可执行文件，如图像文件、音频文件，而病毒需要可执行文件才能运行。	DevSecOps 工程

树干	软件产品的主要源代码集成存储库。	连续交付架构, DevOps 测试工程
单元测试	测试的目的是验证代码逻辑。	连续交付架构, DevOps 测试工程
可用性测试	测试的目的是确定人类在使用 EUT 时是否拥 有令人满意的体验。	连续交付架构, DevOps 测试工程
用户	IT 服务消费者。或者, 在身份验证 期间声明的身份（也称为用户名）。	DevOps 基金会, DevSecOps 工程
用户和实体行为分析 （UEBA）	一种机器学习技术, 用于分析正常和" 异常 "用户行为, 以防止后者出现。	站点可靠性工 程
用户案例	从用户的业务角度编写的声明, 描述用户如何从产品的功能实现目标。用户情景在产品积压工作（或进程积压工作）中捕获。	认证Agile 流程所有者, 认证敏捷服务经理
增值时间	在创造价值的活动（例如开发、测试）上花费的时间量。	DevOps 领导者
价值效率	能够以最少的时间和资源产生 价值。	DevOps 领导者
价值流	从客户请求到交付的产品或服务的所有活动。	DevOps 基金会
价值流映射	精益工具, 描述信息、材料和工作在功能孤岛之间的流动, 重点是量化浪费, 包括时间和质量。	DevOps 基金会
价值流 管理	能够可视化整个 DevOps 生命周期的价值交付流程。Gitlab CI 和 Jenkins 扩展（来自云蜜蜂） DevOptics 可以提供此可视化效果。	站点可靠性工 程
价值流所有者	个人对高级管理层负责, 以提高 给定产品或服务的非价值比率。	经过认证的敏捷流程所 有者
变速 IT	传统流程和数字流程在组织内共存, 同时以自己的速度移动的方法。	DevOps 基金会

速度	在预定义间隔内完成的工作量度。 个人或团队在给定时间上可以完成的工作量。	DevOps 基金会, DevSecOps 工程， 现场可靠性工程
判决	测试结果分为"失败"、"通过"或"无结论"。	连续交付 架构, DevOps 测试工程
版本控制工具	确保"单一的真相来源"，并启用所有生产工件的变更控制和跟踪。	DevOps 基金会
垂直缩放	计算资源的扩展速度 更高，以提高处理速度，例如使用更快的计算机更快地运行更多任务。	DevOps 测试工程
病毒（计算机）	附加到文件的恶意可执行代码，当受感染的文件从系统传递到系统时，该代码可能无害 （但令人讨厌）， 或者可以修改或删除数据。	DevSecOps 工程
客户之声 （VOC ）	捕获和分析客户需求和反馈以了解客户需求的过程。	DevOps 基金会
漏洞	设计、系统或应用程序中可能被攻击者利用的弱点。	DevSecOps 工程
漏洞 智能	描述已知漏洞的信息，包括按版本影响的软件、漏洞的相对严重性（例如，它会导致用户角色的特权升级，还是导致拒绝服务）、漏洞的利用（漏洞是多么容易/难以利用）， 以及有时在野外的当前 exploi tation 速率（是被积极利用还是只是理论上的）。此信息还通常包括有关已知哪些软件版本已修复所述漏洞的指导。	DevSecOps 工程
漏洞 管理	识别和修复漏洞的过程。	DevSecOps 工程
等待时间	等待工作浪费的时间量（例如，等待开发和测试基础结构、等待资源、等待管理层 批准）。	DevOps 领导者

废物（精益 制造）	任何不为流程、产品或服务增加价值的活动。	经过认证的敏捷流程所有者、经过认证的敏捷服务经理、DevOps 基金会、DevOps 领导者
水+Scrum+秋天	结合瀑布和 Scrum 开发的混合应用程序生命周期管理方法可以在给定时间内完成。	连续交付架构
瀑布（项目管理）	线性和 顺序方法管理软件设计和开发项目，其中进度被视为稳定（和顺序）向动（如瀑布）。	经过认证的敏捷服务经理、持续交付架构、DevOps 基础
弱点	攻击者可以利用软件中的错误来破坏应用程序、系统或其中包含的数据。也称为漏洞。	DevSecOps 工程
Web 应用防火墙（WAF）	检查发送到应用程序的 流量并可以阻止任何看起来恶意的工具。	站点可靠性工程
Web IDE	具有 Web 客户端集成开发环境的工具。无需使用本地开发工具即可提高开发人员 的工作效率。	站点可靠性工程
韦斯特鲁姆 （组织类型）	Ron Westrum 发展了组织文化类型，其中包括三种类型的组织：病理学（以权力为导向）、官僚主义（以规则为导向）和生成（以绩效为导向）。	DevSecOps 工程，现场可靠性工程
白色+盒式测试（或透明、玻璃、透明盒测试或结构测试）	测试用例使用广泛的知识，对应用程序的内部设计结构或工作，而不是其功能（即黑盒测试）。	连续交付架构，DevOps 测试工程
白名单	应用程序白名单是指定允许在计算机系统上存在 并处于活动状态的已批准软件应用程序的索引的做法。	连续交付架构
邪恶的问题	邪恶的问题被用来揭露塑造我们行动和选择的假设。它们是 阐明我们关于问题、问题或背景的根深蒂固的、常常是相互矛盾的假设的问题。	DevOps 领导者

维基	通过使用创建丰富 Wiki 内容的工具，可以启用知识 共享	站点可靠性工程
威尔伯的象限	一种 识别人类 一般方法模式的模型。使用两个轴：在一个轴上，人们倾向于个性或集体。	DevOps 领导者
正在进行中的工作 （WIP）	已启动但尚未完成的任何工作。	DevOps 基金会
解决 方案	减少或消除事件或问题影响的临时方法。可能记录为已知错误数据库中的已知错误。（ITIL 定义）。	DevOps基金会，DevSecOps 工程
世界咖啡厅	是知识共享的结构化对话过程，其中一组人在几个表中讨论一个主题，个人定期切换表，然后由"表主机"介绍到新表中 的上一个讨论。	DevOps 领导者
蠕虫（计算机）	蠕虫通过将自己附加到不同的文件并在计算机之间查找路径来在系统上复制自身。它们通常减慢网络速度，并且可以自己运行（病毒需要主机程序才能运行）。	DevSecOps 工程

SRE 基础课程：增值资源

本文档提供与站点可靠性工程相关的文章和视频的链接(SRE)课程来自 DevOps 研究所。提供此信息是为了提高您对 SRE 相关概念和术语的理解，不可检查。当然，有是网上其他视频、博客和案例研究的财富。我们欢迎增加建议。

课程中的视频

Module	Title & Description	Link
1: SRE 准则& 实践	‘DevOps和SRE有什么不同?’ Seth Vargo 和 Liz Fong- Jones of Google (05:10)	https://youtu.be/uTEL8Ff1Zvk
2: 服务级别目标与错误预算	‘风险和错误预算’ Seth Vargo 和 Liz Fong-Jones of Google (06:17)	https://youtu.be/y2ILKr8kCJU
3: 减少琐事	‘实用自动化’ with Max Luebbe of GCP (04:45)	https://www.youtube.com/watch?v=oDcjAcFTFC0&t=0m56s
4: 监控和服务级别指示器	‘SLI和可靠性深度解析’ David N. Blank-Edelman of Microsoft (08:35)	https://www.youtube.com/watch?v=1iMo3SkdQqQ
5: SRE 工具& 自动化	‘自动化的讽刺：三部分的喜剧’ Tanner Lund of Microsoft (18:32)	https://www.youtube.com/watch?v=U3ubcoNzx9k
6: 防脆弱 – 从失败中学习	‘懒惰，引发网络故障的工具’ Preetha Appan of Indeed.com (04:45)	https://www.usenix.org/conference/srecon17americas/program/presentation/appan
7: SRE 的组织影响	‘Uber的SRE之路’ Rick Boone of Uber (06:24)	https://www.youtube.com/watch?v=qJnS-EfIlIE
8: SRE，其他框架和未来	‘ITIL4 & SRE’ Jayne Groll of DevOps Institute (11:25)	https://dev.tube/video/vFyPXIsUEhE

SRE报告

报告名称	作者/出版商	链接
2019 SRE报告	Catchpoint	http://pages.catchpoint.com/SRE-Report-2019.html
什么是SRE?	Kurt Andersen & Craig Sebenik from O'Reilly Media	https://www.oreilly.com/library/view/what-is-sre/9781492054429/

SRE文章

文章标题和作者	相关模块	链接
'哪些因素影响软件项目维护成本更多?' by Sayed Mehdi Hejazi Dehaghani and Nafiseh Hajrahimi	1: SRE 准则& 实践	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3610582/
'衡量和评估服务级别目标（SLO）' by Serhat Can	1: SRE 准则& 实践	https://medium.com/@serhatcan/measuring-and-evaluating-service-level-objectives-slos-84b0dc740a0a
'彭博社在SRE上下了大赌注' by Michael Rembetsy	1: SRE 准则& 实践	https://www.techatbloomberg.com/blog/bloomberg-bets-big-on-sres/
'SRE在彭博社的实践' by Stig Sorensen	1: SRE 准则& 实践	https://player.fm/series/devops-chat/site-reliability-engineering-sre-bloomberg-w-stig-Sorenson
'成为一名SRE意味着什么?' by Molly Struve	1: SRE 准则& 实践	https://dev.to/molly_struve/what-it-means-to-be-a-site-reliability-engineer-32ki
'错误预算-实际实施' by Yaroslav Molochko	2: 服务级别目标与错误预算	https://www.slideshare.net/yaroslavmolochko/implementing-error-budgets-125400822
'如何避免5个SRE实现陷阱，即使是最好的团队也会被困住' by Lyon Wong	2: 服务级别目标与错误预算	https://thenewstack.io/how-to-avoid-the-5-sre-implementation-traps-that-catch-even-the-best-teams/

‘SRE: DevOps 2.0’ by Saba Anees	2: 服务级别目标与错误预算	https://www.appdynamics.com/blog/engineering/site-reliability-engineering-devops-2-0/
‘SRE入门’ by Jennifer Petoff	2: 服务级别目标与错误预算	https://www.devops.talksplus.com/wp-content/themes/dotc/2019_Melbourne/presentations/Getting%20Started%20with%20Site%20Reliability%20Engineering%20(Jennifer%20Petoff%20DOTC%20Deck).pdf
‘多发明，少苦工’ by Betsy Beyer, Brendan Gleason, Dave O’connor and Vivek Rau	3: 减少琐事	https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45765.pdf
‘SRE的教训:不断优化以减少琐事’ by Damon Edwards	3: 减少琐事	https://www.rundeck.com/blog/sre-lessons-continuously-optimize-to-reduce-toil
‘辛劳: 终于有了一个我们都感受到的问题的名字’ by Damon Edwards	3: 减少琐事	https://www.rundeck.com/blog/toil-finally-a-name-for-a-problem
‘SRE的教训:不断优化以减少琐事’ by Damon Edwards	3: 减少琐事	https://www.rundeck.com/blog/sre-lessons-continuously-optimize-to-reduce-toil
‘Site Reliability Engineering (SRE): 简单概述’ by Mac Slocum	3: 减少琐事	https://www.oreilly.com/ideas/site-reliability-engineering-sre-a-simple-overview
‘什么是SRE?’ by Craig Sebenik & Kurt Andersen	3: 减少琐事	https://www.oreilly.com/library/view/what-is-sre/9781492054429/
‘值得花时间吗?’ by Xkcd	3: 减少琐事	https://imgs.xkcd.com/comics/is_it_worth_the_time.png
‘SLA、SLO和SLI的工程师指南’ by Ram Lyengar	4: 监控和服务级别指示器	https://plumbr.io/blog/monitoring/an-engineers-guide-to-sla-slo-and-sli
‘实践中的服务水平指标’ by Stephen Thorne	4: 监控和服务级别指示器	https://medium.com/@jerub/service-level-indicators-in-practice-6a1125e24bee
‘停止使用Nagios（这样它就可以平静地死去了）’ by Andy Sykes	4: 监控和服务级别指示器	http://www.slideshare.net/superdupersheep/stop-using-nagios-so-it-can-die-Peacefully

为什么我的监控系统很糟糕？' by Todd Palion	4: 监控和服务级别指示器	https://www.usenix.org/conference/srecon19asia/presentation/palino- monitoring

'可观测性——3年回顾' by Charity Majors	4: 监控和服务级别指示器	https://thenewstack.io/observability-a-3-year-retrospective/
'监控和可观察性-有什么区别，为什么重要？' by Peter Waterhouse	4: 监控和服务级别指示器	https://thenewstack.io/monitoring-and-observability-whats-the-difference-and-why-does-it-matter/
'降低监控中警报噪音的3种方法' by Christina DiSomma	4: 监控和服务级别指示器	https://www.metricly.com/3-ways-reduce-alert-noise/
'可观察性和对系统运行后果的理解' by Charity Majors	4: 监控和服务级别指示器	https://www.infoq.com/articles/charity-majors-observability-failure/
'运行服务水平指标(SLI)研讨会' BY GDS	4: 监控和服务级别指示器	https://gds-way.cloudapps.digital/standards/slis.html
'谷歌自动化的演变' by Niall Murphy	5: SRE 工具& 自动化	https://landing.google.com/sre/sre-book/chapters/automation-at-google/
'工作与养老金部的SRE' by various	5: SRE 工具& 自动化	https://dwpdigital.blog.gov.uk/category/site-reliability-engineering-sre/
'衡量和评估服务水平目标' by Serhat Can	5: SRE 工具& 自动化	https://www.atlassian.com/blog/opsgenie/measuring-and-evaluating-service-level-objectives
'2019年最佳NoSQL数据库'	5: SRE 工具& 自动化	https://www.improgrammer.net/most-popular-nosql-database/
'支持DevOps文化的on-call工具' by Dan Holloran	5: SRE 工具& 自动化	https://victorops.com/blog/devops-on-call-tools-to-support-culture
'很棒的网站可靠度工具' by Raghu Chinnannan	5: SRE 工具& 自动化	https://github.com/squadcastHQ/awesome-sre-tools
'安全性和合规性' by Ansible	5: SRE 工具& 自动化	https://www.ansible.com/use-cases/security-and-compliance

‘安全编码最佳实践 ’ by OWASP	5: SRE 工具& 自动化	https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf

‘安全的生产测试方式’ by Cindy Sridharan	5: SRE 工具& 自动化	https://medium.com/@copyconstruct/testing-in-production-the-safe-way-18ca102d0ef1
‘亚马逊安东线:它是什么以及如何反应’ by Velentin Bayard	5: SRE 工具& 自动化	https://blueboard.io/resources/amazon-andon-cord/
‘DevOps工具环境’ by GitLab	5: SRE 工具& 自动化	https://about.gitlab.com/devops-tools/
‘衡量效率、有效性和文化以优化开发转型’ by IT Revolution	6: 防脆弱 – 从失败中学习	http://devopsenterprise.io/media/DOES_forum_metrics_102015.pdf
‘跟踪每个版本’ by Mike Brittain	6: 防脆弱 – 从失败中学习	https://codeascraft.com/2010/12/08/track-every-release/
‘恢复点目标’ by Margaret Rouse	6: 防脆弱 – 从失败中学习	https://whatis.techtarget.com/definition/recovery-point-objective-RPO
‘学习型组织’ by Andrew Shafer	6: 防脆弱 – 从失败中学习	https://www.slideshare.net/littleidea/the-learning-organization-modev
‘三种方式: 支撑devops的原则’ by Gene Kim	6: 防脆弱 – 从失败中学习	https://itrevolution.com/the-three-ways-principles-underpinning-devops/
‘组织文化的类型 ’ by R Westrum	6: 防脆弱 – 从失败中学习	http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1765804/pdf/v013p0ii22.pdf
‘您是否希望云解决方案取得成功？从拥抱失败开始!’ by Arman Kamran	6: 防脆弱 – 从失败中学习	https://medium.com/@armankamran/do-you-want-your-cloud-solutions-to-succeed-start-with-embracing-failures-8f5f40b57a64
‘IT停机时间的成本’ by Michael Copeland	7: SRE 的组织影响	https://www.the20.com/blog/the-cost-of-it-downtime/
‘SRE团队是如何组织的，以及如何开始 ’ by Matt Brown	7: SRE 的组织影响	https://cloud.google.com/blog/products/devops-sre/how-sre-teams-are-organized-and-how-to-get-started

‘Kubernetes启动并运行’ by Brendan Burns, Joe Beda & Kelsey Hightower	7: SRE 的组织影响	https://clouddamcdnprodep.azureedge.net/gdc/gdckTIBtc/original

‘无可指责的事后总结和公正的文化’ by John Allspaw	7: SRE 的组织影响	https://codeascraft.com/2012/05/22/blameless-postmortems/
‘首要指令’ by Norm Kerth	7: SRE 的组织影响	https://retrospectivewiki.org/index.php?title=The_Prime_Directive
‘创建抗脆弱系统:企业的现场可靠性工程’ by Contino	7: SRE 的组织影响	https://www.contino.io/files/Enterprise-Site-Reliability-Engineering-Contino.pdf
‘扩展SRE组织:从一个团队到多个团队的旅程’ by Gustavo Franco	7: SRE 的组织影响	https://www.usenix.org/sites/default/files/conference/protected-files/sre19amer_slides_franco.pdf
‘Devop的集合’ by John Willis	8: SRE，其他框架和未来	http://itrevolution.com/the-convergence-of-devops/
‘SRE的角色和职责’ by Dan Holloran	8: SRE，其他框架和未来	https://victorops.com/blog/site-reliability-engineer-sre-roles-and-responsibilities
‘ITIL4和SRE如何与DevOps保持一致’ by Jayne Groll	8: SRE，其他框架和未来	https://techbeacon.com/enterprise-it/how-itil4-sre-align-devops
‘可靠性工程的未来’ by Michael Kehoe	8: SRE，其他框架和未来	https://michael-kehoe.io/tags/future-of-sre/
‘数据库可靠性简介’ by Mackenzie Clark	8: SRE，其他框架和未来	https://softwareengineeringdaily.com/2018/10/16/an-introduction-to-database-reliability/
‘停止争论:ITILv4、SRE和DevOps都是转型的帮手’	8: SRE，其他框架和未来	https://devopsinstitute.com/2019/11/05/stop-the-arguments-itil-v4-and-sre-and-devops-all-are-transformation-aids/

Websites

Title	Link
Usenix	https://www.usenix.org
Honeycomb	https://www.honeycomb.io/
Player FM – DevOps Chat	https://player.fm/series/devops-chat
SRE Weekly	https://sreweekly.com/

SRE（现场可靠性工程）基础 v1.0
样本检验

 的 2020 年 6 月

1. 以下哪一项是 SRE 的最佳定义？

A. 当软件工程师负责执行操作时会发生什么
B. 一门融合了软件工程方面的学科，并将其应用于基础设施和运营问题
C. 处理后期制作歌剧工作的角色，如管理 SLA 和错误预算
D. 将软件工程的各个方面应用于服务后期生产管理的学科

2. 根据谷歌的说法， 以下哪一项不是 DevOps 的支柱？

A. 减少组织孤岛
B. 接受正常故障
C. 实施渐进式变革
D. 从右到左增加流量

3. SRE 中的服务"管理"了哪些？

A. 服务级别目标 （处）
B. 服务级别指示器
C. 错误预算
D. 服务级别 协议

4. 故障成本可以降低吗？

A. 改善平均维修时间 （MTTR）
B. 进行较小的更改
C. 金丝雀部署
D. 以上所有内容

5. 以下哪一项是跟踪最广泛的服务级别目标 （S处） ？

A. 更改故障率
B. 安全
C. 可用 性
D. 能力

6. 根据 Catchpoint 调查，"延迟"的定义是什么？

A. 与 停机相关的延迟
B. 用户发出请求到收到响应
C. 服务级别目标和错误预算的区别
D. 通信消息时发生的延迟

7. 团队每月的服务 级别目标为 99.9%。多少时间
是否为错误预算分配？

a. 45 分钟
b. 43 分钟
c. 25 分钟
d. 32 分钟

8. "T"在 SLO VALET 模型中代表什么？

a. 交通
b. 票
c. 测试
d. 维修 时间

9. 数百万客户使用的旅行网站会经历一个小时的中断，
超过商定的错误预算。 对业务可能产生什么影响？

a. 相关应用程序的失败
b. 员工分心
c. 错过的服务 级别指示器
d. 社交媒体反弹

10. 错误预算应每月刻录为零？

a. 真
b. 假

11. –家医疗保健公司 SRE 团队已经开始探索他们执行哪些操作任务可以自动。自动化任务的示例是什么？

A. 正常工作
B. 无可指责的验尸
C. 批准生产部署 的物理会议
D. 规划会议

12. 解决问题的实验是一种辛劳？

A. 真
B. 假

13. 一家有20年历史的软件公司很难保持与这么多新的初创公司 将新功能推向市场需要很长时间，发布时经常有错误。 什么实践可以帮助公司更快地提高质量？

A. 服务级别目标 （处）
B. 减少辛劳
C. 管理错误预算
D. 自动扩展基础结构

14. SRE 至少 50% 的时间应该花在工程工作上：

A. 改进服务
B. 遵守错误预算策略
C. 改进服务支持
D. 降低成本

15. 谷歌引入工程工作 50% 规则的原因之 一是什么？斯雷斯？

A. 为 SREs 提供向开发人员学习的时间
B. 确保一个团队或人员有足够的工程技能来帮助自动化劳作
C. 确保一个团队不会成为"行动"
D. 确保有充分的时间评估和 实施自动化

16. 以下哪一项描述了服务级别指示器？

a. 服务 的目标可靠性级别
b. 指示是否满足服务级别目标的数据
c. 定义服务级别的正式合同
d. 由客户定义的服务级别要求

17. 虽然许多数字可以作为服务 Level 指示器，但 Google 建议将 SLA 视为：

a. 公式
b. 自动算法
c. 比
d. 可观测数据

18. 从多个远程端收集和聚合数据的自动化过程点称为

a. 提醒
b. 可观
c. 遥测
d. 应用程序性能管理

19. 应用程序性能管理的目标是什么？

a. 远程数据的监视和聚合，用于应用程序方面，例如网络、服务器、应用软件
b. 监测和管理业绩和可用性软件应用程序
c. 使用硬件或软件组件来监视系统资源和性能
d. 异常检测

20. 以下哪一项不是监控方法的一部分？

a. 建立我是对的，什么是错的规则
b. 以适当的比例在时间范围中聚合
c. SLO 和关联 SLI 的仪表板和显示
d. 事件票证系统，以跟踪事件响应

21. 运营团队正在实施新的监控解决方案，并且必须就将生成警报的 CPU 阈值。这是一个例子？

A. 图形
B. 仪表板
C. 事件标识
D. 异常检测

22. 以下哪一项是遥测的定义？

A. 监测和管理业绩和可用性
   软件应用程序
B. 使用硬件或软件组件 来监视系统资源和
   计算机系统的性能
C. 工程师交流系统定量数据的方法
D. 进行测量的自动化通信流程
   以及在远程或不可访问点收集的其他数据，并传送至
   接收设备进行监控。

23. 以下是 SRE 主导服务自动化的以下哪一项

A. 监控和警报侧重于已知出错的事情，
   因此影响可靠性
B. 鼓励 越来越多的功能被推动生产，没有
   考虑影响
C. 环境必须预配为基础结构（或配置）作为代码
   支持一致性
D. 自动功能测试和非功能测试可立即进行
   在部署之前，以避免生产中的问题

24. SRES 应在生产中执行什么类型的测试？

A. 功能测试
B. 非功能测试
C. 用户要求测试
D. 功能测试和非功能测试

25. 基础架构作为代码的主要好处是什么？

A. 开发人员可以在生产环境中测试其代码， 因为基础结构代码是处理与应用程序代码相同
B. 相同的代码用于创建开发、测试、暂存和生产，因此所有代码环境是一致的
C. 它为容器化设置了舞台
D. 它允许 SREs 提高 可靠性，即使将更多功能推送到生产更快

26. 为什么必须制定检测设备，使服务在外部可见？

A. 确保返回正确的数据和服务级别指示器，正在生成和存储日志文件
B. 确保生产前测试和生产测试是相同的， 以便验证结果
C. 确保 DevOps 管道专注于通过更好的提高可靠性生产 测试
D. 确保开发人员和业务利益干系人可以访问该服务

27. 以下哪一项不是概述未来增长包络的好处？

A. 提前减少辛劳
B. 返工减少
C. 安全和审核事件集中
D. 降低了服务所有权的总成本

28. 以下哪一项不是将基础结构预配为代码的好处？

A. 在生产中测试和审核更改会更容易
B. 所有环境的预生产和 后期制作将一致
C. 开发人员需要了解环境才能编写基础结构代码
D. 在测试环境中识别错误会更容易

29. 如果一个组织正在努力采用从失败培养者中学习， 他们应该怎么做

A. 在发布之前执行更多测试
B. 做更多的回顾
C. 调查停机成本
D. 投资专业培训

30. 在 Westrum 研究中，是什么类型的组织导致对失败的调查？

A. 病理
B. 生成
C. 官僚
D. 学习

31. 定期进行消防演习的主要好处是什么？

A. 确保组织能够引入混沌工程
B. 确保业务连续性计划的资金充足
C. 确保业务在不可预见的事件或失败后继续工作
D. 在生产事件成为重大事件之前识别它们

32. 混沌工程的最佳定义是什么？

A. 随机关闭服务器或基础结构以在意外中断中实施恢复的纪律。
B. 经常进行消防演习以确保业务在发生重大中断时能够恢复的纪律
C. 经常进行消防演习以确保业务在发生重大中断时能够恢复的纪律
D. 将软件工程的各个方面应用于服务后期生产管理的学科

33. 以下哪一项描述了采用 SRE 的完整 SRE 方法？

A. SRE 的所有权由服务/交付团队拥有
B. SRE 与开发之间有共同的责任和共享的通话
   团队
C. 运营实践的完全所有权由 SRE 团队负责
D. 混合运营和工程团队

**34. 就地应叫程序对于确保满足服务级别目标至关重要。谁
可以参加电话计划吗？**

A. 每个人 都
B. SRE 团队和事件响应团队
C. 服务台和 SRE 团队
D. 运营和工程团队的代表

35. Google 倡导 SRES 的通话时间百分比是多少？

a. 25%
b. 50%
c. 30%
d. 10%

36. 无可指责的验尸结果之一是什么？

a. 缓解未来类似事件 的跟进行动列表
b. SRES 可以采取的任务列表，以减少辛劳，以减轻类似事件
c. 讨论避免未来类似事件所需的资源
d. 服务级别目标列表 及其受事件影响

37. ITIL/ITSM 的哪些方面可以特别支持 SRE？

a. 变革咨询委员会
b. 服务价值系统
c. 服务台
d. 工艺模型

38. 数据库可靠性引擎的主要责任是什么？

a. 帮助组织从官僚主义转向生成流程
信息
b. 通过频繁进行消防演习和数据完整性测试确保数据可靠性
c. 将混沌工程应用于数据库以确认 故障转移和恢复
反应
d. 确保数据得到充分备份和可恢复

39. 传统应用程序还可以受益于 SRE 原则和实践。什么服务
可以建立级别目标，以专注于遗留问题？

a. 旧 版应用程序的数量
b. COBOL 代码行
c. 大型机处理器速度
d. 批处理时间

40. 以下哪一项不是调整 SRE 和敏捷实践的关键优势？

A. 完成的定义将更清晰， 并有更多的端到端的视角
B. 客户将从工作软件中实现更多价值
C. 辛劳的积压 使工作可见，并可以帮助优先考虑工程和自动化故事
D. SR 将嵌入到敏捷或 Scrum 团队中

回答键

问题	正确答案	主题区域
1	B	1： SRE 原则与实践
2	D	1： SRE 原则与实践
3	A	1： SRE 原则与实践
4	D	1： SRE 原则与实践
5	C	2：服务级别目标和错误预算
6	D	2：服务级别目标和错误预算
7	B	2：服务级别目标和错误预算
8	B	2：服务级别目标和错误预算
9	D	2：服务级别目标和错误预算
10	A	2：服务级别目标和错误预算
11	C	3： 消除辛劳
12	B	3： 消除辛劳
13	B	3： 消除辛劳
14	A	3： 消除辛劳
15	C	3： 消除辛劳
16	B	4： 监控和服务级别指示器
17	C	4： 监控和服务级别指示器
18	C	4： 监控和服务级别指示器
19	B	4： 监控和服务级别指示器
20	D	4： 监控和服务级别指示器
21	D	4： 监控和服务级别指示器

22	D	4： 监控和服务级别指示器

23	C	5： SRE 工具和自动化
24	D	5： SRE 工具和自动化
25	B	5： SRE 工具和自动化
26	A	5： SRE 工具和自动化
27	C	5： SRE 工具和自动化
28	C	5： SRE 工具和自动化
29	C	6： 抗脆弱和从失败 中学习
30	B	6： 抗脆弱和从失败中学习
31	C	6： 抗脆弱和从失败中学习
32	C	6： 抗脆弱和从失败中学习
33	D	7： SRE 的组织影响
34	D	7： SRE 的组织影响
35	A	7： SRE 的组织影响
36	A	7： SRE 的组织影响
37	D	8： SRE， 其他框架， 趋势
38	C	8： SRE， 其他框架， 趋势
39	D	8： SRE， 其他框架， 趋势
40	D	8： SRE， 其他框架， 趋势

Your Path to

DevOps Success

DevOps Institute is dedicated to advancing the human elements of DevOps success. Our goal is to help advance careers and support emerging practices using a role-based approach to certification which focuses on the most modern competencies and hireable skills required by today's organizations adopting DevOps.

Take the next steps in your learning and certification journey to DevOps success.

Click on a certification or visit www.devopsinstitute.com/certifications to learn more.

Become a Member

Join the fastest growing global community of DevOps practitioners and professionals and gain access to invaluable learning content, the latest news, events, emerging practices, develop your network and advance your career.

You belong.

(www.devopsinstitute.com/become-a-community-member)