

DevOps 导论 MOOC 习题答案

§ DevOps 概述

1. 下列描述中，不属于典型软件发展三大阶段的是： C
 - A. 网络化和服务化
 - B. 软硬件一体化阶段
 - C. 软件作坊
 - D. 软件成为独立产品
2. “Measure twice, Cut once” 是哪个阶段的典型开发特征？ D
 - A. 软件作坊阶段
 - B. 网络化阶段
 - C. 软件成为独立产品阶段
 - D. 软硬件一体化阶段
3. 关于软件过程管理，以下哪一种说法是比较贴切的： A
 - A. 软件过程管理关注的是企业软件过程能力的稳定输出和提升。
 - B. 软件过程管理是软件企业发展到较高层次才需要关心的话题。
 - C. 软件过程管理主要关注软件成本和质量目标的达成。
 - D. 进入互联网时代，软件过程管理是过于老套的话题。
4. 软件开发的本质难题中哪一个与软件发展阶段没有直接关系？ B
 - A. 复杂性
 - B. 不可见性
 - C. 可变性
 - D. 一致性
5. “Code and Fix” 是软件发展哪个阶段的典型开发特征？ C
 - A. 互联网时代
 - B. 网络化和服务化
 - C. 软硬件一体化
 - D. 软件作为独立产品
6. 以下哪个因素促成了软件成为独立的产品？ D
 - A. 个人电脑的出现
 - B. 互联网的出现
 - C. 高级程序设计语言的出现
 - D. 操作系统的出现
7. 软件危机和软件工程这两个概念提出时间是？ B
 - A. 上世纪五十年代
 - B. 上世纪六十年代
 - C. 上世纪八十年代
 - D. 上世纪七十年代
8. 以下描述中，哪几种是网络化和服务化这个阶段的典型软件应用特征？ ABC
 - A. 快速演化、需求不确定
 - B. 通过 SaaS 等方式来发布软件系统
 - C. 用户数量急剧增加
 - D. 通过 CD 和 DVD 等方式支持大容量和快速分发软件拷贝
9. 关于形式化方法的描述当中，不正确的有哪些？ BC
 - A. 这种方法对开发人员技能有较高的要求
 - B. 这种方法的主要目的是解决软件开发的效率问题
 - C. 这种方法是网络化和服务化阶段用来应对软件开发本质四大难题而提出来的
 - D. 这种方法应用范围有限，例如：不适合跟客户讨论需求。
10. 关于迭代式方法的说法哪些是比较恰当的？ AB
 - A. 迭代式方法是指一类具有类似特征的方法
 - B. 迭代式方法主要特征在于将软件开发过程视作一个逐步学习和交流的过程
 - C. 迭代式方法是上世纪九十年代中后期才出现的一种方法

- D. 迭代式方法主要是为了解决软件开发的质量问题
11. DevOps 方法的出现具有一定的必然性，与以下哪些软件应用特征相匹配？ABCD
- A. 软件系统部署环境越来越错综复杂
 - B. 软件定义世界，软件随处可见
 - C. 用户需求多变所带来的软件系统的快速演化的要求
 - D. 软件在社会生活当中扮演了越来越关键的角色
12. DevOps 的哪些特点可以有效支撑当前社会对软件系统的期望？ABCD
- A. 微服务架构设计
 - B. 敏捷开发、精益思想以及看板方法，支持快速开发、交付、迭代和演化
 - C. 工具链支持高效率的自动化
 - D. 虚拟机技术的大量应用
13. 在 DevOps 化的 three ways 当中，关注质量问题是第二个阶段才需要考虑的。 (X)
14. DevOps 中的 XaaS 特指 SaaS、PaaS 以及 IaaS 这三种。 (X)
15. DevOps 化的 Three ways 当中，建立反馈机制是二阶段应该实现的目标。 (✓)

§ 个体软件过程单元测试

1. 下述各个度量项中，哪一个不是 PSP 的基本度量项？B
- A. 缺陷
 - B. 风险
 - C. 规模
 - D. 时间
2. 关于面向用户的质量观，我们应该关注如下哪些问题：ACD
- A. 用户期望是否有优先级？
 - B. 界面和可操作性是首要的，因为这是用户能直接感受到的。
 - C. 真实用户是谁？
 - D. 用户期望的优先级对软件开发的影响？
3. PSP 当中为什么用缺陷管理替代质量管理？下述说法中正确的是：BD
- A. 因为缺陷管理和质量管理其实是一回事。
 - B. 因为单纯质量管理很难操作。
 - C. 因为缺陷管理相关的活动（例如，测试等）本来就是软件开发中必须要开展的活动。
 - D. 因为缺陷往往对应了面向用户质量观中的首要用户期望。
4. 关于 PROBE 估算法，下述各种说法中，不正确的有哪些？ABD
- A. PROBE 估算结果带着小数，肯定不准确，因而，不应该在项目估算的时候使用。
 - B. PROBE 不能给出精确估算，因而适合用来跟用户讨论需求和规模。
 - C. PROBE 方法不能用来估算质量。
 - D. PROBE 方法不需要历史数据。
5. 关于质量路径 (Quality Journey)，下列说法中哪些不恰当。AD
- A. 质量路径中所列举的方法都是提升开发质量的有效手段，可以随意选择使用。
 - B. 高质量软件产品最终还是需要依赖测试来确保。
 - C. 进入测试之前的高质量，是获得测试之后高质量软件系统的前提条件。
 - D. 质量路径与个体软件工程师无关，是团队层面的集体努力。

6. 关于评审检查表，下述说法中不恰当的是：CD

- A. 评审检查表应该是个性化的
- B. 评审检查表应该定期更新
- C. 项目团队所有人应该共用一份评审检查表，体现统一性
- D. 评审检查表应该保持稳定，确保缺陷不会被遗漏

7. 关于 PQI，下述说法中不恰当的是：BD

- A. PQI 可以用来辅助判断模块开发的质量
- B. PQI 五个分指标都可以超过 1.0，比如，设计时间多于编码时间的时候，该分指标就超过 1.0 了
- C. PQI 可以为过程改进提供依据
- D. PQI 越高越好，最好达到 1.0

8. 关于评审，下述说法中不恰当是：CD

- A. 代码的个人评审也应该通过评审检查表来进行。
- B. 如果安排了代码的小组评审，那么代码个人评审就可以不用做。
- C. 代码的个人评审最好交叉进行，因为阅读自己代码容易产生思维定式，不利于缺陷发现。
- D. 代码的个人评审应该安排在单元测试之后，确保评审对象有着较高的质量，提升评审价值。

9. 关于质量的各种定义当中，下述哪些质量属性属于内部属性？AD

- A. 可扩展性
- B. 安全性
- C. 可靠性
- D. 可移植性

10. PSP 鼓励使用瀑布型生命周期模型。 (X)

11. 对于初学者来说，代码评审速度可以控制到每小时不超过 400 行。 (X)

12. “高质量的软件开发是计划出来的” (✓)

§ 敏捷软件开发

1. 根据敏捷宣言，以下哪项描述了更多的价值？B

- A. 可工作的软件、个体交互、响应变化、相近的文档
- B. 个体和交互、可工作的软件、客户协作、响应变化
- C. 客户协作、遵循计划、可工作的软件、个体交互
- D. 响应变化、个体和交互、流程和工作、客户协作

2. 下列哪一项更好地描述了敏捷宣言？B

- A. 它包含了许多敏捷团队使用的实践
- C. 它定义了构建软件的规则
- B. 它包含了建立敏捷思维方式的价值观
- D. 它概述了构建软件的最有效方法

3. 你是一家社交媒体公司的开发人员，正在开发一个项目，项目需要一个为企业客户创建私有网站的新功能。您需要与公司的网络工程师一起确定部署策略，并提出一组工程师可以用于管理站点的服务和工具。网络工程师希望在你的网络内部部署所有服务，但您和您的团队成员不同意，并且认为服务应该部署在客户的网络上。为了达成一个协议，该项目的工作已经停止。哪种敏捷价值最适合这种情况？A

- A. 客户合作高于合同谈判
- C. 个体和互动高于流程和工具
- B. 响应变化高于遵循计划
- D. 工作的软件高于详尽的文档

4. 你是一个软件团队的开发人员。一个用户向你的团队询问有关构建新功能的信息，并以规范的形式提供了需求。她非常确定这个功能要如何工作，并承诺不会有任何变化。哪种敏捷价值最适用于这种情况？B

- A. 响应变化高于遵循计划
B. 工作的软件高于详尽的文档
C. 客户合作高于合同谈判
D. 个体和互动高于流程和工具
5. Sean 是一个正在构建财务软件的团队的开发人员。他的团队被要求开发一个新的交易系统。他和他的团队召开会议来提出他们正在使用的工作流的图景。然后，他们将流程放在白板上，流程中的每个步骤都有一列。经过对团队在白板上工作项目进行了几周观察，他们注意到这个过程中有几个步骤似乎过载了。对于他们来说，下一步应该做什么？ D
- A. 专注于完成看板上的工作
B. 在较慢的步骤中使用更多的人力
C. 与团队合作，在工作进展缓慢的阶段更好地完成工作
D. 对过载步骤中正在进行的工作项目的数量进行限制
6. 下列哪一个不是精益原则？ A
- A. 实施反馈循环
B. 尽可能晚的做决定
C. 消除浪费
D. 识别所有的步骤
7. 下列哪一个更好地描述了如何使用看板？ C
- A. 帮助团队自我组织，并了解工作流程中的瓶颈所在
B. 跟踪缺陷和问题，并创建解决产品问题的最快途径
C. 观察特征如何流过程，以便团队可以确定如何限制 WIP 并通过工作流程中的步骤确定最均匀的工作流程
D. 跟踪 WIP 限制和当前任务状态，以便团队知道他们还有多少工作要做
8. 以下不是经常出现在 Kanban 上记事贴中的内容 B
- A. 完成时间
B. 团队名词
C. 谁在处理这个工作项
D. 工作项描述
9. 一个公司内，各个团队的 Kanban 列设置应当一致，便于公司管理。 (X)
10. 在制品规模越小越好，因为这样可以优化前置时间，并且团队的效率会变高。 (X)
11. 在 DevOps 中，可以使用 Kanban 方法，也可以使用 Scrum 等其他敏捷方法。 (✓)

§ 软件架构演化

1. 下面关于软件架构的描述哪个是不正确的？ B
- A. 软件架构包含一系列重要决策，包括软件组织、构成系统的结构要素等。
B. 软件架构是一组特定的架构元素，包括处理元素、数据元素和上下文元素。
C. 软件架构包括系统组件、连接件和约束的集合。
D. 软件架构即一系列重要的设计决策。
2. 在应用分层架构的软件系统中，最先处理外部请求的是： B
- A. 数据层
B. 表现层
C. 应用层
D. 业务层
3. 以下哪个关于面向服务架构的描述是错误的？ B
- A. 面向服务架构包含服务提供者组件和服务消费者组件
B. 面向服务架构是一个集中式组件的集合
C. 在 SOA 中，服务消费者消费其他组件提供的服务不需要知道其具体的实现细节
D. SOA 依赖企业服务总线为服务间的相互调用提供支持环境
4. 以下对于微服务优点的描述中，哪一个是错误的？ C

- A. 不同的微服务可以使用不同的语言进行开发 C. 微服务系统测试变得非常简单
B. 微服务可以使用 RPC 进行服务间通信 D. 单个微服务很简单，只关注一个业务功能
5. 在微服务架构中，ZooKeeper 的主要作用是? B
A. 开发服务 B. 注册服务 C. 封装服务 D. 调用服务
6. 除 Spring Boot 之外，主流的微服务开发框架还有什么? A
A. Apache Dubbo B. Django C. MyBaits D. Kubernetes
7. 在组成派看来，软件架构是指? ACD
A. 软件架构包括系统组件、连接件和约束的集合。
B. 软件架构是一系列重要决策的集合，包括构成系统的结构要素及其接口的选择。
C. 软件架构由软件元素、这些元素的外部可见属性，以及元素之间的关系组成。
D. 软件架构将系统定义为计算组件及组件间的交互。
8. 分层架构将软件系统的组件分成多个互不重叠的层，包括: BC
A. 物理层 B. 持久层 C. 数据层 D. 应用层
9. 分层架构模式的缺点包括: ABCD
A. 由于层间依赖关系，软件系统的可扩展性差 C. 软件升级需要暂停整个服务
B. 代码调整通常比较麻烦 D. 不易于持续发布和部署
10. 以下哪几个不是面向服务架构强调的实现原则? BC
A. 服务解耦 B. 服务去中心化 C. 服务简单 D. 服务自治
11. 以下选项中，哪些属于微服务架构的特点? ABCD
A. 围绕业务能力组织 B. 基础设施自动化 C. 通过服务组件化 D. 内聚和解耦
12. 以下选项中，API 网关模式的优点有哪些? ABC
A. 将从客户端调用多项服务的逻辑转换为从 API 网关处调用，以简化整个客户端
B. 为每套客户端提供最优 API
C. 确保客户端不受服务实例位置的影响
D. 增加请求往返次数
13. 与面向服务架构相关的 Web 服务标准包括: ABC
A. SOAP B. UDDI C. WSDL D. UML
14. 单体应用的所有功能都被集成在一起作为一个单一的单元。 (✓)
15. 单体架构更多地作为应用的部署架构，单体应用只运行在一个进程中。 (✗)
16. 微服务架构风格是一种将一个单一应用程序开发为一个小型服务的方法。 (✗)
17. 本质上，微服务架构是 SOA 的一种扩展。 (✓)
18. 核心模式即针对采用微服务系统在通用场景下的所有问题，所使用的成熟的架构解决方案集合。(✗)

§ 云原生与容器技术

1. 下列哪项不是 Docker 容器的特点: C

- A. 创建速度很快
B. 可以共享操作系统的资源
C. 启动时间是分钟级
D. 资源使用较少
2. 下列哪项不是 Docker 的网络模式 B
A. Bridge 模式 B. 其他全是 C. None 模式 D. Host 模式
3. 以下哪些是 Docker 的存储驱动: A
A. 其他都是 B. verlayFS C. Device mapper D. AUFS
4. 以下哪个命令可以查看当前运行容器: B
A. docker top B. docker ps C. docker run D. docker logs
5. Kubernetes 集群将元数据保存在以下哪个组件: B
A. Kubelet B. Etd C. 以上都不是 D. Kube-apiserver
6. 以下哪些是 Kubernetes 的控制器: C
A. ReplicaSet C. Both ReplicaSet and Deployment
B. Deployment D. Rolling Updates
7. 以下哪些是 Kubernetes 的核心概念 A
A. 其他都是 B. Services C. Volumes D. Pods
8. Kubernetes 里面的 Replication 控制器的职责是: D
A. 当已存在的 Pod 异常退出后, 创建新的 Pod C. 删除或者更新多个 Pod
B. 帮助达到预期的状态 D. 其他都是
9. 如何通过命令行创建一个容器? C
A. docker create B. docker start C. docker run D. docker poll
10. Dockerfile 中的命令 RUN, CMD 和 ENTRYPOINT 几者有何区别? B
A. ENTRYPOINT 配置容器启动时运行的命令
B. 其他都是
C. RUN 执行命令并创建新的镜像层, RUN 经常用于安装软件包。
D. CMD 设置容器启动后默认执行的命令及其参数, 但 CMD 能够被 docker run 后面跟的命令行参数替换
11. 使用 Kubernetes 带来的好处有哪些? B
A. 横向扩展 B. 其他都是 C. 自动调度 D. 自动回滚
12. 以下哪项用于确保 pod 不会被调度到不适当的节点上? A
A. Taints 和 Tolerations C. TTaints
B. 以上都不是 D. Tolerations
13. Docker 容器的状态有: BD
A. Paused B. Running C. Restarting D. Exited
14. 关于 Kubernetes 的 namespace: 命名空间是在多个用户之间划分群集资源的方法 (✓)
15. 以下描述是否正确: 多步构建允许在 Dockerfile 中使用多个 FROM 指令。两个 FROM 指令之间的所有指令会生产一个中间镜像, 最后一个 FROM 指令之后的指令将生成最终镜像。中间镜像中的文件可以通过 COPY --from=<image-number>指令拷贝, 其中image-number为镜像编号, 0 为第一个基础镜像。没有被拷贝的文件都不会存在于最终生成的镜像, 这样可以减小镜像大小, 同时避免出现安全问题。(✓)

§ DevOps 工具链

1. 下列哪项不属于 DevOps 工具生态圈? C
A. 编译 B. 持续集成 C. 持续部署 D. 监控
2. 下列哪项不属于协同开发工具? A
A. Confluence B. Kanban C. JIRA D. Rally
3. 下列哪种持续集成工具是目前 DevOps 领域使用最广泛的? C
A. TeamCity B. Travis CI C. Jenkins D. VSTS
4. 下列哪种版本管理工具不是同宗同源的? A
A. Subversion B. Git C. GitLab D. GitHub
5. 以下哪项不是 Git 的文件目录? D
A. .git 目录 B. 加载区 C. 工作目录 D. Documents
6. 下列哪种编译工具无法编译 JAVA 语言? A
A. MSBuild B. Maven C. Gradle D. Ant
7. 下列哪种工具无法实现对远程服务器的配置操作? A
A. JIRA B. Ansible C. Chef D. Puppet
8. 下列不属于测试的是: A
A. SIT 部署 B. UI 测试 C. 单元测试 D. API 测试
9. 使用 _____ 工具完成 DevOps 持续交付流水线编排配置? C
A. Java B. SonarQube C. Jenkins D. JUnit
10. Git 是何种工具? D
A. 监控工具 B. 单元测试工具 C. 持续集成工具 D. 版本管理工具
11. 以下哪种工具是开源工具? A
A. JUnit B. TeamCity C. Zabbix D. JIRA
12. 下列哪种工具能模拟市场上主流浏览器的操作? D
A. JUnit B. Jenkins C. FitNesse D. Selenium
13. JIRA Software 不支持极限编程这种敏捷开发方法。 (✓)
14. Jenkins 支持工作流即代码 (pipeline-as-code)。 (✓)
15. Git 使用副本方式存储文件版本。 (✗)
16. Selenium 能实现自动化单元测试。 (✗)
17. Nagios 不属于监控工具。 (✗)
18. Zabbix 有两种工作模式。 (✓)
19. SonarQube 能完成持续交付流水线编排配置 (✗)
20. Ansible 只需要在 Server 端安装就能实现对远程服务器的配置管理? (✓)

§ 简答题

1. 请描述一个简单的持续交付流水线所包含的基本步骤。

解析：开发 → 版本控制 → 代码检查 → 构建 → 自动化测试 → 打包 → 远程仓库发布 → 自动化部署

2. 请列举出敏捷软件开发中常见的 3 种开发方法。

解析：

1. 极限编程（简称 XP），是一种近螺旋式的开发方法，它将复杂的开发过程分解为一个个相对比较简单的小周期；通过积极的交流、反馈以及其它一系列的方法，开发人员和客户可以非常清楚开发进度、变化、待解决的问题和潜在的困难等，并根据实际情况及时地调整开发过程。
2. 精益开发，思想起源于丰田公司，旨在创造价值的目标下，通过改良流程不断地消除浪费。这种方法现已被广泛用于生产制造管理，对于 IT 系统建设，精益开发的常用工具模型是价值流模型。
3. Scrum 是一个用于开发和维护复杂产品的框架，是一个增量的、迭代的开发过程。Scrum 以经验性过程控制理论（经验主义）做为理论基础的过程。经验主义主张知识源于经验，以及基于已知的东西做决定。Scrum 采用迭代、增量的方法来优化可预见性并控制风险。

3. 请描述敏捷软件开发宣言内容？

解析：

- 个体和互动胜过程序和工具
- 客户合作胜过合同谈判
- 可以工作的软件胜过详尽的文档
- 响应变化胜过遵循计划
- 也就是说，尽管右项有其价值，我们更重视左项的价值

4. 请解释一下什么是 Kanban 方法中的 WIP？为什么要限制 WIP？

解析：WIP 全称是 work in progress，表示在途工作量，即同时进行中的工作数量。

限制 WIP 可以让团队成员更佳专注与手边的工作，减少工作切换所造成的浪费，因而可以加速工作完成的时间。此外，当工作卡住而无法完成的时候，开发团队不能以此为借口忽略这些被卡住的工作（因为有 WIP 限制，不能无限制的一直拿工作而不完成它），此时就有可能会有人闲置下来，而影响工作流程与产能。鼓励团队成员一起解决问题，排除阻碍。可以更均衡的工作产出，避免前期开发后期集中或者压缩测试时间，能够更稳健的长久的进行敏捷实践。

5. 什么叫做面向用户的质量观？这种观点对软件开发有什么影响？

解析：面向用户的质量观是定义质量为满足用户需求的程度。这个定义中需要进一步明确：用户究竟是谁？用户需求的优先级是什么？这种用户的优先级对软件产品的开发过程产生什么样的影响？怎样来度量这种质量观下的质量水平？

在软件开发中，用缺陷管理来替代质量管理，高质量产品也就意味着要求组成软件产品的各个组件基本无缺陷。缺陷消除的平均代价随着开发过程的进展会显著增加，各个组件的高质量是通过高质量评审来实现的。

6. DevOps 有哪些常见的质量手段有助于确保最终软件服务的质量？

解析：

1. 持续集成。持续集成 (CI) 是一个开发过程，每天多次将代码集成到共享存储库中。借助自动化测试，CI 帮助允许团队及早识别错误、轻松定位问题，提高了软件质量并缩短了交付时间。
2. 持续部署。通过评估拉取请求并将它们组合到主分支，持续部署为开发人员提供了对流水线末端产品的关注。它允许企业快速部署、验证新功能，并在测试自动化完成后立即做出响应。有了持续部

署流水线，一旦客户提交质量问题，团队就可以轻松处理新版本的错误，因为每个版本都是小批量交付的。

3. 持续测试。在 CI/CD 工作流中，构建往往以小批量进行。因此，为每次构建，手动运行测试用例会非常耗时。持续测试借助自动化手段，尽早、逐步和充分地执行测试，发现问题解决问题。
4. 自动化。借助强大的部署自动化手段和标准化的环境管理来降低部署操作的成本，确保部署任务的可重复性，减少部署出错的可能性。
5. DevOps 致力于在整个开发过程中的每一个环节都引入 QA 和测试管理，使它们成为质量的推动者，并确保产品符合利益相关者和用户所设定的质量标准。QA 实际上被认为是 DevOps 中非常关键的组件，甚至于 DevOps 强调质量保证是每个人的责任。

7. 请谈谈微服务架构与面向服务的架构存在哪些异同？

解析：

1. 微服务是一种软件的架构风格，面向服务的架构（SOA）不是一种特定的技术，而是一种分布式计算的软件设计方法。
2. 微服务简单的说就是组合化，它的每部分需要实现的功能可以有不同的小程序单独构成，然后相互之间协同实现一个大的目标。这个角度上来说，两者是一脉相承的，但是面向服务的架构，没有微服务的分离度高，相互之间的关联度还是相对较高。
3. 微服务相比较来说，在各个组件上可以使用不一样的编程语言。
4. 微服务更加关注于解耦，不追求系统之间的相关性。
5. 微服务的系统发生改变只需要构建一个新的服务，简单快捷，但是 SOA 则需要对整个系统进行修改。
6. 微服务使用的协议一般都是轻量级的协议，就像 HTTP、Thrift API 等协议，但是 SOA 则是使用更为复杂多样的多种消息协议。微服务的容错性能会更好，即使一个微服务出现问题，其他的微服务也会正常工作。
7. 每个微服务都有单独的数据库，SOA 则是共享一个数据库。
8. 微服务的规模更小，SOA 则是一个较大的规模。SOA 可以是一个整体，也可以是多个微服务组成的。

8. 从你的理解出发，谈谈为什么微服务架构具有高可用性、灵活性等优点？

解析：微服务架构普遍被采用于云原生应用、无服务器计算、以及使用轻量级容器部署的应用等，根据 Fowler 的观点，由于服务数量众多（与单体应用实现相比），为了有效地开发、维护和运营这类应用，去中心化的持续交付和带有整体服务监控的 DevOps 是必要的。遵循这种方法的一个合理性结果是，单独的微服务可以单独扩展。在单体应用架构方法中，一个支持三个功能的应用，即使只有其中一个功能需要添加资源约束，也需要对其进行整体的扩展。微服务则不同，只需要对有资源约束需求的微服务进行扩展，这样就带来了资源和成本的优化。

微服务并不是单体应用中的一个层，相反，它是一个自成一体的业务功能，具有明确的接口，可以通过自己的内部组件实现分层架构。从策略的角度来看，微服务架构本质上遵循了 Unix 的“做一件事，做好一件事”的理念，改变应用程序的一小部分只需要重建和重新部署一个或少量的服务即可，使用可直达独立部署服务的精细化接口、业务驱动开发（如领域驱动设计）等原则。