**职业技术技能标准**

标准编码：

**研发效能(DevOps)技术工程师**

**（2023年版）**

**中国电子教育学会 制定**

**说明**

为贯彻落实《关于深化人才发展体制机制改革的意见》，推动实施人才强国战略，促进专业技术人员提升职业素养、补充新知识新技能，实现人力资源深度开发，推动经济社会全面发展，根据《中华人民共和国劳动法》有关规定，中国电子教育学会组织有关专家，制定了《研发效能(DevOps)技术工程师职业技术技能标准（2022年版）》（以下简称《标准》）。

一、本《标准》按照《国家职业技能标准编制技术规程》有关要求，坚持“以职业活动为导向、以专业能力为核心”的指导思想，在充分考虑科技进步、社会经济发展和产业结构变化对研发效能(DevOps)技术工程师专业要求的基础上，以客观反映研发效能(DevOps)技术发展水平及其对从业人员的专业能力要求为目标，对研发效能(DevOps)技术工程师的专业活动内容进行规范细致描述，明确了各等级专业技术人员的工作领域、工作内容以及知识水平、专业能力要求。

二、本《标准》为首次制定，依据有关规定将本职业分为初级、中级、高级三个等级，包括职业概况、基本要求、工作要求、权重表和附录等五个方面内容。

三、本《标准》的编制工作在工业和信息化部教育与考试中心的指导下，由盛科天冠（北京）信息技术有限公司具体组织实施。

四、本《标准》主要起草单位有：工业和信息化部教育与考试中心、盛科天冠（北京）信息技术有限公司、中国科学院上海高等研究院、北京大学、华南理工大学、厦门理工学院、华为云技术有限公司、英捷创软（北京）科技有限公司、京东科技信息技术有限公司、腾讯科技（深圳）有限公司、中信银行股份有限公司、民生银行股份有限公司、中国农业银行股份有限公司、中国电子系统技术有限公司、中国石油天然气股份有限公司、新华三技术有限公司、金蝶软件（中国）有限公司、杭州飞致云信息科技有限公司、上海驻云科技有限公司、北京月朋科技有限公司、北京速云科技有限公司、印孚瑟斯技术有限公司、广东网联天下科技有限公司、北京中育培智教育发展中心等。

五、本《标准》主要审定人员有：左仁贵、杜志海、罗洪元、咸汝平、王立杰、姚冬、徐磊、宁德军、庄俊乾、薄涛、蒋华锋、赵宪林、彭晓楠等。

六、本《标准》主要编写人员有：许舟平、杜伟忠、李强、赵卫、赵舜东、陈晓鹏、陈计节、李红涛、胡稳安、姚元庆、王艳、冷大鲲、刘晓玲、胡振宇、徐浩、陈文峰、吴非、许华荣、施慧斌、张怀涛、黄隽、冀利斌、陈一梦、钟聪睿等。

七、本《标准》经中国电子教育学会批准，自颁布之日起施行。

目录

1.职业概况 5

1.1职业名称 5

1.2标准编码 5

1.3职业定义 5

1.4职业技术等级 5

1.5职业环境条件 5

1.6职业能力特征 5

1.7普通受教育程度 5

1.8职业培训要求 5

1.8.1培训时间 5

1.8.2培训教师 5

1.8.3培训场所设备 5

1.9专业技术考核要求 5

1.9.1申报条件 5

1.9.2考核方式 6

1.9.3监考人员、考评人员与考生配比 6

1.9.4考核时间 6

1.9.5考核场所设备 6

2.基本要求 6

2.1职业道德 6

2.1.1职业道德基本知识 6

2.1.2职业守则 6

2.2基础知识 7

2.2.1基础理论知识 7

2.2.2技术基础知识 7

2.2.3相关法律、法规知识 7

2.2.4其他相关知识 7

3.工作要求 7

3.1各级工作内容映射 8

3.2初级 10

3.3中级 19

3.4高级 29

4.权重表 41

4.1理论知识要求权重表 41

4.2实践能力要求权重表 41

5.附录 41

5.1参考文献 41

5.2中英文术语对照表 41

**研发效能（DevOps）技术工程师**

**职业技术技能标准**

（2023年版）

# 1.职业概况

## 1.1职业名称

研发效能（DevOps）技术工程师

## 1.2标准编码

（待补充）

## 1.3职业定义

从事研发效能（DevOps）管理与技术研究的工程技术人员，包括产品、架构、开发、测试、运维、运营等角色。研发效能（DevOps）技术工程师，能力要求为组织与协作管理、产品与运营、开发与交付、测试与安全、运维与监控等过程的技术实践，例如企业及团队的技术卓越建设、持续集成、持续测试、持续部署、持续发布以及自动化运维等；以及完成工作所必须的流程及管理能力要求。

## 1.4职业技术等级

本职业共设三个等级，分别为初级、中级、高级。

## 1.5职业环境条件

室内，常温。

## 1.6职业能力特征

具有较强的学习能力、计算能力、表达能力及分析、推理和判断能力。

## 1.7普通受教育程度

大学专科学历（或高等职业学校毕业）。

## 1.8职业培训要求

### 1.8.1培训时间

研发效能（DevOps）技术工程师需按照本《标准》的职业要求参加有关课程培训。完成规定学时，取得学时证明。初级 32 标准学时，中级 64 标准学时，高级 128 标准学时。

### 1.8.2培训教师

承担初级、中级理论知识或专业能力培训任务的人员，应具有相关职业中级及以上专业技术等级或相关专业中级及以上职称。

承担高级理论知识或专业能力培训任务的人员，应具有相关职业高级专业技术等级或相关专业高级职称。

### 1.8.3培训场所设备

理论知识培训在标准教室或线上平台进行；专业能力培训在具有相应软、硬件条件的培训场所进行。

## 1.9专业技术考核要求

### 1.9.1申报条件

——取得初级培训学时证明，并具备以下条件之一者，可申报初级专业技术等级：

（1）取得技术员职称。

（2）具备相关专业大学本科及以上学历（含在读的应届毕业生）。

（3）具备相关专业大学专科学历，从事本职业技术工作满 1 年。

（4）技工院校毕业生按国家有关规定申报。

——取得中级培训学时证明，并具备以下条件之一者，可申报中级专业技术等级：

（1）取得助理工程师职称后，从事本职业技术工作满 2 年。

（2）具备大学本科学历，或学士学位，或大学专科学历，从事本职业技术工作满 2年。

（3）具备硕士学位或第二学士学位，从事本职业技术工作满 1年。

（4）具备相关专业博士学位。

（5）技工院校毕业生按国家有关规定申报。

——取得高级培训学时证明，取得中级专业技术等级，并具备以下条件之一者，可申报高级专业技术等级：

（1）取得工程师职称后，从事本职业技术工作满 3 年。

（2）具备大学本科学历，或学士学位，或大学专科学历，从事本职业技术工作满 3 年。

（3）具备硕士学位或第二学士学位，从事本职业技术工作满 2 年。

（3）具备博士学位，从事本职业技术工作满 1 年。

（4）技工院校毕业生按国家有关规定申报。

### 1.9.2考核方式

分为理论知识考试以及实践能力考核。理论知识考试、实践能力考核均实行百分制，成绩皆达60分（含）以上者为合格，考核合格者获得相应专业技术等级证书。

理论知识考试以闭卷笔试、机考等方式为主，主要考核从业人员从事本职业应掌握的基本要求和相关知识要求；实践能力考核以实操考核、论文设计等方式为主，主要考核从事本职业应具备的技术水平。

### 1.9.3监考人员、考评人员与考生配比

理论知识考试中的监考人员与考生配比不低于1:15，且每个考场不少于1名监考人员；专业能力考核中的考评人员与考生配比不低于1:15，且考评人员为2人（含）以上数。

### 1.9.4考核时间

理论知识考试时间不少于90分钟，实践能力考核时间不少于150分钟。

### 1.9.5考核场所设备

理论知识考试统一在线上考试系统进行；实践能力考核在具备软、硬件及网络环境的教室进行。

# 2.基本要求

## 2.1职业道德

### 2.1.1职业道德基本知识

### 2.1.2职业守则

（1）遵纪守法，爱岗敬业。

（2）诚实守信，恪守职责。

（3）精益求精，勇于创新。

（4）遵守规程，安全操作。

（5）团结协作，忠于职守。

## 2.2基础知识

### 2.2.1基础理论知识

（1）软件工程基础知识。

（2）组织与协作管理基础知识。

（3）产品规划与运营基础知识。

（4）应用开发与交付基础知识。

（5）应用软件测试基础知识。

（6）应用安全基础知识。

（7）应用运维与监控基础知识。

### 2.2.2技术基础知识

（1）精益敏捷、DevOps、研发效能等软件工程知识。

（2）团队管理与组织知识。

（3）计划管理与效能度量知识。

（4）产品洞察设计与需求管理知识。

（5）应用技术设计、协同开发、持续集成、持续部署、按需发布、开发环境管理等知识。

（6）测试组织管理、测试生命周期管理、测试关键域管理等知识。

（7）IT服务管理、应用运行管理、应用监控管理、业务连续性管理等知识。

（8）安全防护与安全测试知识。

（9）产品运营知识。

### 2.2.3相关法律、法规知识

（1）《中华人民共和国劳动法》相关知识。

（2）《中华人民共和国安全生产法》相关知识。

（3）《中华人民共和国网络安全法》相关知识。

（4）《中华人民共和国个人信息保护法》相关知识。

（5）《全国人民代表大会常务委员会关于加强网络信息保护的决定》相关知识。

### 2.2.4其他相关知识

（1）环境保护知识。

（2）文明生产知识。

（3）劳动保护知识。

（4）资料保管保密知识。

# 3.工作要求

本标准从端到端的软件研发生命周期的维度，包括组织与协作管理、产品与运营、开发与交付、测试与安全、运维与监控等职业功能对研发效能（DevOps）工程师进行定义。

研发效能（DevOps）技术工程师侧重于如持续集成、持续测试、持续部署、持续发布以及自动化运维等技术实践，以及完成工作所必须的流程及管理能力要求。

本标准对初级、中级、高级的专业能力要求和相关知识要求依次递进，高级别涵盖低级别的要求。

## 3.1各级工作内容映射

在各职业功能及工作内容上，对初级、中级、高级的专业能力要求如下表所示。

（图示说明：

- 本级别对本项专业能力和相关知识无相应要求

● 本级别对本项专业能力和相关知识有相应要求

○ 本级别对本项专业能力和相关知识无相关要求，涵盖低级别要求）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **职业功能** | **工作内容** | **工作内容** | **初级** | **中级** | **高级** |
| 1.组织与协作 | 1.1构建个体能力 | 1.1.1推广敏捷DevOps理念 | ● | ● | ● |
| 1.1.2 构建研发效能转型方案 | ● | ● | ● |
| 1.1.3 推动研发效能转型 | - | ● | ● |
| 1.1.4 建立领导力 | - | - | - |
| 1.2管理团队协作 | 1.2.1组建团队，确定协作流程及工作环境 | ● | ● | ● |
| 1.2.2制定交付计划 | ● | ● | ● |
| 1.2.3跟踪执行计划 | ● | ● | ● |
| 1.2.4优化工作方式 | ● | ● | ● |
| 1.3建设学习型组织 | 1.3.1 营造组织持续学习环境 | - | ● | ○ |
| 1.3.2 建设生态型组织 | - | ● | ● |
| 1.3.3 建立组织的持续改进机制 | - | - | ● |
| 1.4度量研发效能 | 1.4.1 度量个体效能 | ● | ● | ● |
| 1.4.2 度量团队效能 | ● | ● | ● |
| 1.4.3 度量组织效能 | - | - | ● |
| 2.设计与运营 | 2.1探索产品 | 2.1.1洞察机会 | - | - | - |
| 2.1.2识别与验证解决方案 | - | - | - |
| 2.2 设计产品 | 2.2.1管理产品需求 | ● | ● | ● |
| 2.2.2设计产品架构 | - | - | ● |
| 2.2.3设计产品界面与用户体验 | ● | ● | ● |
| 2.3 运营产品 | 2.3.1制定运营策略 | ● | ● | ○ |
| 2.3.2监测激活指标 | ● | ● | ○ |
| 2.3.3 提升用户价值 | - | ● | ○ |
| 2.3.4 搭建规模化增长体系 | - | ● | ○ |
| 3.开发与交付 | 3.1 协同开发应用 | 3.1.1 设计解决方案 | - | ● | ● |
| 3.1.2 持续改进开发流程 | ● | ● | ● |
| 3.1.3 管理源代码和优化工具 | ● | ● | ● |
| 3.2 持续集成应用 | 3.2.1 搭建与实施流水线 | ● | ● | ● |
| 3.2.2 设计与集成质量门禁 | ● | ● | ● |
| 3.2.3 集成与应用代码质量保证 | ● | ● | ● |
| 3.2.4 设计与应用单元测试 | ● | ● | ● |
| 3.3 持续部署应用 | 3.3.1 装配与管理环境 | ● | ● | ○ |
| 3.3.2 部署与更新版本 | ● | ● | ○ |
| 3.4 按需发布应用 | 3.4.1 设计生产发布策略 | ● | ● | ● |
| 3.4.2 持续发布生产 | ● | ● | ● |
| 3.4.3 设计交付工具链及流程 | ● | ● | ● |
| 3.4.4 预发布与运营拟真 | ● | ● | ● |
| 3.5 管理开发环境 | 3.5.1 优化开发调试环境 | ● | ● | ● |
| 3.5.2 管理开发环境安全 | ● | ● | ● |
| 4.测试与安全 | 4.1 管理测试组织 | 4.1.1 采纳测试方法论 | ● | ● | ● |
| 4.1.2 管理测试团队 | - | ● | ● |
| 4.1.3 采纳与管理测试技术与平台工具 | ● | ● | ● |
| 4.2 管理测试生命周期 | 4.2.1 制定测试策略与计划 | ● | ● | ● |
| 4.2.2 管理测试需求 | ● | ● | ● |
| 4.2.3 设计测试用例 | ● | ● | ● |
| 4.2.4 执行与报告测试 | ● | ● | ● |
| 4.2.5 管理与预防缺陷 | ● | ● | ● |
| 4.3 管理测试关键域 | 4.3.1 执行测试估算 | ● | ● | ● |
| 4.3.2 制定测试度量 | ● | ● | ● |
| 4.3.3 使用测试技术 | ● | ● | ○ |
| 4.3.4 执行测试阶段 | ● | ● | ○ |
| 4.3.5 执行自动化功能测试 | ● | ● | ● |
| 4.3.6 执行非功能性测试 | ● | ● | ○ |
| 4.3.7 执行性能测试 | ● | ● | ● |
| 4.3.8 管理测试数据 | ● | ● | ● |
| 4.3.9 管理测试环境 | ● | ● | ● |
| 4.3.10 执行敏捷测试 | ● | ● | ● |
| 4.4 安全防护与安全测试 | 4.4.1 管理应用安全生命周期 | ● | ● | ● |
| 4.4.2 设计应用安全测试方案 | ● | ● | ● |
| 4.4.3 执行应用安全测试方案 | ● | ● | ○ |
| 4.4.4 执行云安全防护 | ● | ● | ● |
| 5. 运维与监控 | 5.1 管理IT服务 | 5.1.1 制定IT服务管理体系 | ● | ● | ● |
| 5.1.2 管理事件与问题 | ● | ● | ● |
| 5.1.3 管理变更与配置 | ● | ● | ● |
| 5.1.4 管理容量与成本 | ● | ● | ● |
| 5.2 管理应用运行 | 5.2.1 管理应用运行方案 | ● | ● | ● |
| 5.2.2 管理基础设施平台 | ● | ● | ● |
| 5.2.3 管理应用运行环境 | ● | ● | ● |
| 5.3 管理应用监控 | 5.3.1 设计与建设监控系统 | ● | ● | ● |
| 5.3.2 管理指标 | ● | ● | ● |
| 5.3.3 管理日志 | ● | ● | ● |
| 5.3.4 管理调用链 | ● | ● | ● |
| 5.3.5 进行智能监控 | ● | ● | ● |
| 5.4 管理业务连续性 | 5.4.1 确定服务质量目标 | ● | ● | ● |
| 5.4.2 响应与回顾事故 | ● | ● | ● |
| 5.4.3 进行灾备管理 | ● | ● | ● |
| 5.4.4 应用混沌工程 | ● | ● | ● |

## 3.2初级

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **职业功能** | **工作内容** | **工作内容** | **专业能力要求** | **相关知识要求** |
| 1.组织与协作 | 1.1构建个体能力 | 1.1.1推广敏捷DevOps理念 | 1.1.1.1 能够在团队内运用DevOps理念  1.1.1.2 能够在团队内运用狭义与广义的DevOps  1.1.1.3 能够在团队内运用DevOps价值观与原则  1.1.1.4 能够在团队内运用精益敏捷思想 | 1.1.1.1 敏捷DevOps历史，DevOps三步工作法，CALMR/CALMS  1.1.1.2 狭义与广义的DevOps，敏捷组织，精益企业  1.1.1.3 研发效能（DevOps）价值观与原则  1.1.1.4 精益与敏捷思想 |
| 1.1.2 构建研发效能转型方案 | 1.1.2.1 能够运用研发效能状况评估  1.1.2.2 能够运用研发效能转型方案  1.1.2.3 能够运用研发效能提升方案 | 1.1.2.1 研发效能转型的必要性、收益以及目标的制定；  1.1.2.2 研发效能方案设计，包括组织、流程等； |
| 1.2管理团队协作 | 1.2.1组建团队，确定协作流程及工作环境 | 1.2.1.1 能够遵守和执行工作流程与协议  1.2.1.2 能够遵守和执行沟通协调机制 | 1.2.1.1 Scrum、Kanban等敏捷开发方法  1.2.1.2 研发协同知识 |
| 1.2.2制定交付计划 | 1.2.2.1 能够遵循并执行发布计划  1.2.2.2 能够遵循并执行交付计划 | 1.2.2.1 发布计划与流程 1.2.2.2 交付计划与流程 |
| 1.2.3跟踪执行计划 | 1.2.3.1 能够更新任务进展  1.2.3.2 能够暴露问题和风险  1.2.3.3 能够执行计划调整 | 1.2.3.1每日站会、任务管理知识  1.2.3.2 问题和风险知识  1.2.3.3 站会知识 |
| 1.2.4优化工作方式 | 1.2.4.1 能够暴露改进点  1.2.4.2 能够落实工作方式的优化  1.2.4.3 能够落实改进项 | 1.2.4.1每日站会、演示会议、敏捷度量与反馈知识  1.2.4.2回顾会议  1.2.4.3行动计划与跟踪、SMART原则 |
| 1.4度量研发效能 | 1.4.1度量个体效能 | 1.4.1.1能够更新度量数据  1.4.1.2能够跟踪个人度量数据趋势  1.4.1.3能够提出优化建议 | 1.4.1.1价值流交付模型（需求提出到发布端到端价值流）  1.4.1.2任务计划工时，任务填报工时，任务剩余工时，计划达成率  1.4.1.3背景-问题-改进方案，STAR原则 |
| 1.4.2 度量团队效能 | 1.4.2.1 能够提供团队产能数据  1.4.2.2 能够提供交付效率数据  1.4.2.3 能够提供产品质量数据  1.4.2.4 能够提供团队效能数据 | 1.4.2.1 迭代速度，需求吞吐量  1.4.2.2 需求交付周期，代码变更前置时间，发布频率  1.4.2.3 缺陷库存数，线上缺陷新增数，缺陷关闭率，缺陷解决时长，缺陷关闭时长，千行代码缺陷率，故障平均恢复时间，故障平均响应时间  1.4.2.4 趋势图，雷达图，数据分析，相关性分析 |
| 2.设计与运营 | 2.2 设计产品 | 2.2.1管理产品需求 | 2.2.1.1能够与干系人确定需求细节；  2.2.1.2能够进行需求条目化拆分；  2.2.1.3能够提供信息支持需求优先级排定 | 2.2.1.1需求管理知识，如用户故事、用户角色、产品待办列表、需求就绪标准（DoR）、需求验收标准AC等；  2.2.1.2需求价值与拆分知识，如故事点、用户故事分层、需求拆分原则、用户故事拆分方法等；  2.2.1.3 优先级排定知识，如MoSoCoW、Kano、WSJF等； |
| 2.2.3 设计产品界面与用户体验 | 2.2.3.1 能够完成部分产品模块的原型图绘制  2.2.3.2 能够根据产品需求设计部分产品模块的交互效果 | 2.2.3.1设计产品原型知识，如以用户为导向的设计、面向场景设计、扩展性设计、原型图设计工具等；  2.2.3.2 交互效果知识，如基于UI的交互体验设计、高保真原型等； |
| 2.3 运营产品 | 2.3.1 制定运营策略 | 2.3.1.1能够整理并输出产品功能模块或功能点的核心目标用户、应用场景、用户需求，并能够评估使用产品后带给用户的改善效果；  2.3.1.2能够对产品功能模块或功能点制定增长指标，包含行为结果指标、过程指标与用户质量指标。 | 2.3.1.1卖点挖掘与价值提炼等知识，如用户痛点分析、用户痛点整理、产品卖点提炼、产品价值包装等；  2.3.1.2运营指标定义与设计知识，如事件定义、指标定义、AARRR海盗模型等； |
| 2.3.2监测激活指标 | 2.3.2.1 具备用户行为数据分析能力，能够依托于现有工具开展留存矩阵分析、行为路径分析、用户质量分析，提出产品迭代优化建议；  2.3.2.2 深刻掌握激活指标定义，能够根据激活数据指标，建立激活指标数据看板。 | 2.3.2.1 数据获取，如数据埋点、数据挖掘、数据清洗、数据整理等；  2.3.2.2 数据应用，如分析模型搭建、数据看板搭建、数据指标监测等； |
| 3.开发与交付 | 3.1 协同开发应用 | 3.1.2 持续改进开发流程 | 3.1.2.1 能掌握各类开发流程的实施要点并执行相应的开发流程  3.1.2.2 能掌握解敏捷开发思路，并且在技术模块设计和工作拆分中正确使用 | 3.1.2.1 各类XP实践、BDD，ATDD等方法  3.1.2.2 模块拆分原则匹配用户故事场景 |
| 3.1.3 管理源代码流程和优化工具 | 3.1.3.1 能够基于代码分支模型、代码审查流程持续进行小批量代码提交及解决冲突  3.1.3.2 能够基于代码分支模型、测试流程规范地实施测试执行  3.1.3.3 能够基于分支模型设置分支保护及Webhook集成流水线，实现代码提交、合并自动触发流水线 | 3.1.3.1 源代码管理工具使用、代码审查、小批量提交代码的方法、规范提交代码的方法  3.1.3.2 测试执行及门禁设置  3.1.3.3 分支保护设置方法、Webhook等 |
| 3.2 持续集成应用 | 3.2.1 搭建与实施流水线 | 3.2.1.1 能够基于统一的流水线模板使用流水线工具建立完整端到端流水线  3.2.1.2 能够支持团队规范使用流水线  3.2.1.3 能够查看流水线监控、日志快速分析处理问题  3.2.1.4 能够增加流水线环节以满足变化的管理需求 | 3.2.1.1 基于工具搭建流水线的方法  3.2.1.2 流水线使用基础知识  3.2.1.3 基于流水线监控、日志分析处理问题的知识  3.2.1.4 流水线中集成插件方法 |
| 3.2.2 设计与集成质量门禁 | 3.2.2.1 能够基于质量门禁报告规范研发流程执行  3.2.2.2 能够根据质量门化方案集成自动化测试、代码扫描、手工测试插件  3.2.2.3 能够根据质量门禁优化方案实施质量门禁优化 | 3.2.2.1 质量门禁使用  3.2.2.2 质量门禁中集成自动化工具的方法、自动化测试及扫描的工具知识  3.2.2.3 质量门禁指标定义 |
| 3.2.3 集成与应用代码质量保证 | 3.2.3.1 能够支持团队使用代码质量保证工具并修复问题  3.2.3.2 能够支持团队在IDE中配置静态代码扫描插件 | 3.2.3.1 代码质量保证工具使用  3.2.3.2 代码质量保证工具配置 |
| 3.2.4 设计与应用单元测试 | 3.2.4.1 能够基于常用的单元测试框架指导团队编写测试用例  3.2.4.2 能够支持团队利用单元测试用例发现程序中问题并修复 | 3.2.4.1 单元测试用例编写方法  3.2.4.2 单元测试问题解决方法 |
| 3.3 持续部署应用 | 3.3.1 装配与管理环境 | 3.3.1.1 能将应用部署到服务器和容器集群平台  3.3.1.2 能使用自动化流水线和脚本任务批量管理软件  3.3.1.3 能根据功能设计和发布规划为应用快速创建新部署环境 | 3.3.1.1 服务器操作系统和容器集群的使用知识  3.3.1.2 脚本编程基础知识 3.3.1.3 自动化流水线和脚本任务管理软件的使用知识 |
| 3.3.2 部署与更新版本 | 3.3.2.1 能编写脚本为应用程序生成启停脚本和容器镜像  3.3.2.2 能操作部署流水线将应用部署到服务器或容器集群 | 3.3.2.1 服务器操作系统和容器集群的使用知识  3.3.2.2 脚本编程基础知识  3.3.2.3 自动化流水线和脚本任务管理软件的使用知识 |
| 3.4 按需发布应用 | 3.4.1 设计生产发布策略 | 3.4.1.1能够根据应用场景制定生产发布计划  3.4.1.2能够根据应用场景设计生产发布步骤  3.4.1.3能够根据应用场景开发部署脚本、回退脚本。 | 3.4.1.1 生产发布计划、应用场景知识  3.4.1.2 生产发布步骤  3.4.1.3 掌握网络协议基本原理及常用的网络配置命令、开发部署脚本、回退脚本 |
| 3.4.2 持续发布生产环境 | 3.4.2.1 能够根据版本发布需要，实施负载均衡或流量网关软件的策略规则  3.4.2.2 能够使用API拨测、端口探测等工具，根据应用的生产发布验证方案进行发布验证  3.4.2.3 能够根据发布需要，更新功能开关配置 | 3.4.2.1 掌握流量网关和负载均衡软件的使用方法  3.4.2.2 掌握API 拨测工具和端口探测工具的使用方法  3.4.2.3 掌握集中式配置管理软件的使用方法 |
| 3.4.3 设计交付工具链及流程 | 3.4.3.1能够根据拆分的可独立交付需求，创建对应特性分支进行开发，并制定版本发布、并版、追版的流程和策略。  3.4.3.2 能够根据交付计划，规划测试环境，制定测试环境发版计划。  3.4.3.3 能够配置和使用自动化发布工具，减少高频发布的工作量。 | 3.4.3.1 熟悉常用的版本管理、制品库、流水线工具  3.4.3.2 掌握常见的分支管理策略及适用场景  3.4.3.3 自动化发布工具 |
| 3.4.4 预发布与运营拟真 | 3.4.4.1 能编制流量网关策略、配置功能开关并验证预发布演练的业务影响  3.4.4.2 能为应用、中间件和基础环境配置日志分析、监控和告警规则  3.4.4.3 能查询日志和监控工具并验证发布变更演练的系统影响 | 3.4.4.1 流量网关、负载均衡和配置管理软件的使用知识  3.4.4.2 API 拨测工具和端口探测工具的使用知识  3.4.4.3 日志分析和监控告警工具的使用知识 |
| 3.5 管理开发环境 | 3.5.1 优化开发调试环境 | 3.5.1.1 能够按照既定的设计搭建高效的开发调试环境，包括开发人员本地开发环境和团队共享环境  3.5.1.2 能够使用各类对开发调试有帮助的各类工具和实践，通过技术手段解决开发人员/团队的日常技术问题 | 3.5.1.1 开发调测环境知识  3.5.1.2 各类源代码管理、开发技术栈以及相关工具链的搭建和使用 |
| 3.5.2 管理开发环境安全 | 3.5.2.1 能够制定安全保障机制  3.5.2.1 能够按照具体的安全保障机制完成对应的工具配置 | 3.5.2.1 安全编码，例如：缓冲区溢出、整数算法错误、XSS/CSRF、SQL注入、弱加密  3.5.2.2 安全测试：安全测试和黑盒测试的区别、风险评估、安全测试方法  3.6.2.3 隐私与敏感数据：敏感数据类型、风险评估、隐私开发和测试的最佳实践 |
| 4.测试与安全 | 4.1 管理测试组织 | 4.1.1 采纳测试方法论 | 4.1.1.1能够识读测试计划的范围、阶段划分、测试进度等，掌握测试策略与目标  4.1.1.2能够根据测试计划，利用测试平台搭建测试环境 | 4.1.1.1 测试标准知识，测试计划，测试进度，测试策略，测试目标；  4.1.1.2 测试平台与工具，测试环境管理； |
| 4.1.3 采纳测试技术与管理平台工具 | 4.1.3.1 能够使用测试平台与工具，根据流程规划完成指定测试任务  4.1.3.2 能够通过测试流程发现平台工具缺陷，并分析问题 | 4.1.3.1 测试自动化工具知识  4.1.3.2 测试平台使用与维护知识 |
| 4.2 管理测试生命周期 | 4.2.1 制定测试策略与计划 | 4.2.1.1能够分析测试风险  4.2.1.2能够分析测试策略  4.2.1.3能够编写测试方案与测试计划  4.2.1.4能够分析软件测试带给干系人的价值 | 4.2.1.1 测试风险管理知识  4.2.1.2 测试方针和目标知识  4.2.1.3 测试策略知识  4.2.1.4 测试方案或测试计划知识 |
| 4.2.2 管理测试需求 | 4.2.2.1能够进行需求评审，并提出合理建议  4.2.2.2能够根据软件需求分析出测试的具体需求,形成测试点  4.2.2.3能够根据跟踪规则维护测试需求和测试用例的跟踪关系  4.2.2.4能够根据变更的测试需求,更新测试用例 | 4.2.2.1 测试评审技能知识  4.2.2.2 测试分析技能知识  4.2.2.3 需求管理知识  4.2.2.4 需求变更管理知识 |
| 4.2.3 设计测试用例 | 4.2.3.1能够运用不同的测试用例设计方法（如等价类、边界值、因果图等）进行测试用例的设计  4.2.3.2能够根据制定的测试用例优先级规则,分配测试用例优先级  4.2.3.3能够基于测试点分析，评审测试用例 | 4.2.3.1 测试用例设计方法（等价类、边界值、因果图等）  4.2.3.2 测试优先级知识  4.2.3.3 测试评审知识 |
| 4.2.4 执行测试与报告结果 | 4.2.4.1能够根据已提供的测试说明解释和执行测试用例  4.2.4.2能够记录并报告测试结果  4.2.4.3能够汇总测试结果,形成测试报告并对产品完成测试后的遗留风险进行评估 | 4.2.4.1执行测试用例技能知识（静态测试、动态测试）  4.2.4.2 测试暂停和退出技能知识  4.2.4.3 测试报告的相关技能知识 |
| 4.2.5 管理与预防缺陷 | 4.2.5.1 能够发现缺陷,并将缺陷录入相关测试管理工具或平台中  4.2.5.2 能够编写和清晰沟通缺陷报告  4.2.5.3 能够验证缺陷修复,并报告修复结果 | 4.2.5.1 缺陷管理技能  4.2.5.2 缺陷报告技能知识  4.2.5.3 缺陷验证与报告知识 |
| 4.3 测试关键域管理 | 4.3.1 执行测试估算 | 4.3.1.1 能够估算测试工作量  4.3.1.2 能够编写测试计划 | 4.3.1.1 常用的测试估算方法  4.3.1.2 制定测试估算计划的能力 |
| 4.3.2 制定测试度量 | 4.3.2.1 能够根据测试度量指标的数据要求完成数据收集  4.3.2.2 能够根据数据收集结果生成度量报告 | 4.3.2.1 缺陷密度、漏测率、测试覆盖率等测试度量指标知识  4.3.2.2统计公式和图表等知识 |
| 4.3.3 使用测试技术 | 4.3.3.1 能够运用基本的测试技术，如白盒与黑盒测试  4.3.3.2 能够使用基本的工具执行测试，如白盒与黑盒测试工具 | 4.3.3.1 白盒测试、黑盒测试等基本知识  4.3.3.2 白盒与黑盒测试工具 |
| 4.3.4 执行测试阶段 | 4.3.4.1 能够完成系统测试用例设计及测试执行  4.3.4.2 能够完成系统集成测试用例设计及测试执行  4.3.4.3 能够完成验收测试用例设计及测试执行  4.3.4.4 能够完成生产测试用例设计及测试执行 | 4.3.4.1 系统测试知识  4.3.4.2 系统集成测试知识  4.3.4.3 验收测试知识，Alpha测试、Beta测试  4.3.4.4 生产测试知识，线上测试、日常巡检、混沌测试、AB测试 |
| 4.3.5 执行自动化功能测试 | 4.3.5.1 能够按需搭建自动化测试环境  4.3.5.2 能够运行自动化测试脚本 | 4.3.5.1 自动化测试环境搭建知识  4.3.5.2 自动化测试执行知识 |
| 4.3.6 执行非功能性测试 | 4.3.6.1 能够完成设计安装/反安装测试场景及测试用例，并且执行测试  4.3.6.2 能够完成设计可移植性测试场景及测试用例，并且执行测试 | 4.3.6.1 安装/反安装测试知识  4.3.6.2 可移植性测试知识 |
| 4.3.7 执行性能测试 | 4.3.7.1 能够搭建性能测试环境  4.3.7.2 能够执行性能测试脚本  4.3.7.3 能够收集和记录性能测试结果并且生成报告 | 4.3.7.1 性能测试环境搭建知识  4.3.7.2 性能测试脚本执行知识  4.3.7.3 性能测试报告知识 |
| 4.3.8 管理测试数据 | 4.3.8.1 能够根据设计好的测试数据策略，进行测试数据维护和管理  4.3.8.2 能够支持测试数据分析及脱敏 | 4.3.8.1 进行测试用例编写与测试数据维护的知识  4.3.8.2 数据脱敏规则，数据变形，敏感隐私数据可靠保护 |
| 4.3.9 管理测试环境 | 4.3.9.1 能够根据测试环境生命周期管理的基本要素进行测试环境管理  4.3.9.2 能够支持构建完整的测试环境，包括但不限于物理机、虚机、容器等 | 4.3.9.1测试环境全生命周期管理的基本知识  4.3.9.2测试虚拟化的基本知识  4.3.9.3云计算中IaaS、PaaS和容器技术快速构建测试环境的知识 |
| 4.3.10 执行敏捷测试 | 4.3.10.1 能够基于敏捷测试四象限并根据制定好的敏捷测试流程执行测试活动  4.3.10.2 能够根据敏捷测试执行结果生成测试报告 | 4.3.10.1 敏捷测试四象限知识  4.3.10.2 测试执行与测试报告知识 |
| 4.4 安全防护与安全测试 | 4.4.1 管理应用安全生命周期 | 4.4.1.1 能够识读应用安全生命周期管理的范围、阶段划分等，掌握测试策略与目标  4.4.1.2 能够利用测试平台搭建安全测试环境  4.4.1.3 能够根据应用安全生命周期管理的基本流程进行执行 | 4.4.1.1 应用安全生命周期管理的基本知识  4.4.1.2 DevSecOps理念与实践  4.4.1.3 应用安全生命周期各阶级的定义  4.4.1.4 应用生命周期各阶级的安全实践 |
| 4.4.2 设计应用安全测试方案 | 4.4.2.1 能够进行应用安全生命周期管理某一阶段的测试方案设计  4.4.2.2 能够选择应用安全生命周期管理某一阶段的测试技术 | 4.4.2.1安全相关模型，如IPDRR、CIA、STRIDE威胁模型等  4.4.2.2安全工具金字塔  4.4.2.3应用生命周期各阶级的安全实践 |
| 4.4.3 执行应用安全测试方案 | 4.4.3.1 能够运用SAST静态应用安全测试工具进行执行  4.4.3.2 能够运用DAST动态应用安全测试工具进行执行  4.4.3.3 能够运用WAF、IDS/IPS工具进行安全规则防护  4.4.3.4 能够根据安全测试的结果进行分析并支持修复 | 4.4.3.1 SAST静态安全测试工具与静态安全检查知识  4.4.3.2 DAST动态安全测试工具与动态安全检查知识  4.4.3.3 WAF、IDS/IPS工具与安全规则防护知识  4.4.3.4 安全测试结果分析及修复知识 |
| 4.4.4 执行云安全防护 | 4.4.4.1 能够执行安全流水线的基本操作  4.4.4.2 能够执行容器安全检查的基本操作  4.4.4.3 能够搭建基础的云安全和安全防护 | 4.4.4.1 安全流水线各阶段的知识，如预提交检查、提交时检查、构建时检查、测试时检查、部署时检查等  4.4.4.2 容器与编排安全知识  4.4.4.3 安全加固与云安全的基础知识 |
| 5. 运维与监控 | 5.1 管理IT服务 | 5.1.1 制定IT服务管理体系 | 5.1.1.1 能够根据IT服务管理流程及标准作业程序，编制各类文档及模板。  5.1.1.2 能够根据IT服务管理要求，通过IT服务管理平台落实执行管理事项。  5.1.1.3 能够贯彻IT服务管理体系与流程执行，并进行跟踪监控、指导支持。 | 5.1.1.1 IT服务管理常规流程，如事件管理、问题管理、变更管理等知识  5.1.1.2 IT服务管理管理平台工具，如ITSM、CMDB、监控平台、自动化平台、云管平台等知识  5.1.1.3 IT服务管理管理平台常规流程使用，如事件管理、问题管理、变更管理等，及跟踪监控、指导支持等知识 |
| 5.1.2 管理事件与问题 | 5.1.2.1 能够根据事件与问题管理流程及标准作业程序，编制各类文档及模板。  5.1.2.2 能够贯彻落实事件与问题管理流程，并进行跟踪监控、指导支持，推动流程运转。  5.1.2.3 能够总结事件、问题临时解决方案，并推动推动根因分析，彻底解决故障。 | 5.1.2.1 IT服务管理常规流程，如事件管理、问题管理等知识  5.1.2.2 IT服务管理工具，如事件管理、问题管理等知识  5.1.2.3 IT服务管理问题管理流程知识 |
| 5.1.3 管理变更与配置 | 5.1.3.1 能够根据变更与配置管理流程及标准作业程序，编制各类文档及模板。  5.1.3.2 能够贯彻变更与配置管理流程落实，推动按时有效部署发布，有效管控风险。  5.1.3.3 能够跟踪监控变更与配置管理流程，并进行跟踪监控、指导支持，推动流程运转。 | 5.1.3.1 IT服务管理常规流程，如CMDB管理、变更管理等知识  5.1.3.2 IT服务管理工具，如CMDB管理、变更管理等知识  5.1.3.3 IT服务管理变更与管理流程知识 |
| 5.1.4 管理容量与成本 | 5.1.4.1 能够根据容量与成本管理流程及标准作业程序，编制各类文档及模板，贯彻容量与成本管理流程落实，推动、跟踪流程运转。  5.1.4.2 能够按照组织容量管理目标，测量并报告组织容量情况。  5.1.4.3 能够响应、监控、识别组织业务的容量变化及要求，诊断和解决与性能和容量有关的故障、问题、变更。 | 5.1.4.1 IT服务管理流程，如容量管理等知识  5.1.4.2 IT服务管理工具，如监控管理、容量管理等知识  5.1.4.3 容量管理知识与容量监控、容量管理技术 |
| 5.2 管理应用运行 | 5.2.1 管理应用运行方案 | 5.2.1.1能够根据应用运行方案，支持应用运行方案的落地实施  5.2.1.2能够在应用运维管理过程中，按照方案确定的管理策略和运维流程，引入并配置相关的运行支撑工具 | 5.2.1.1基础设施、中间件、数据库知识  5.2.1.2负载均衡、数据管理、应用运维知识 |
| 5.2.2 管理基础设施平台 | 5.2.2.1 能够根据应用运行方案，管理应用运行所依赖的基础设施平台，为应用的运行提供基础设施保障  5.2.2.2 能够落实各类基础设施平台的管理与使用，包括但不限于物理机、虚拟化、私有云、公有云、超融合  5.2.2.3 能够根据应用运行的规划要求，针对不同的网络与存储资源，完成相应的配置、验证、管理与日常运维工作 | 5.2.2.1 计算、存储、网络、操作系统知识及操作运维知识  5.2.2.2 虚拟化、超融合、云平台、云服务、容器知识  5.2.2.3 VPC/子网/安全组/防火墙/WAF/端口/NAT/弹性IP知识 |
| 5.2.3 管理应用运行环境 | 5.2.3.1 能够根据应用运行方案，部署、配置与管理应用运行所依赖中间件、数据库、PaaS服务等  5.2.3.2 能够根据应用的部署方案与高可用方案，部署与配置应用运行的高可用运维与监控架构，能够进行应用的部署、更新、故障排除与结果反馈  5.2.3.3 能够根据开发与交付的工程实践，支持团队实施各种不同类型的持续发布策略、负载均衡策略、熔断策略等 | 5.2.3.1 数据库/中间件/云PaaS服务知识  5.2.3.2 集群与高可用知识、缺陷知识  5.2.3.3 代码管理、应用发布部署、服务负载均衡、熔断策略等知识 |
| 5.3 设计管理应用监控 | 5.3.1设计与建设监控系统 | 5.3.1.1 能够根据简单场景设计出对应的监控架构以及技术选型。  5.3.1.2 能够设计常用的可视化图表类型以及应用场景。  5.3.1.3 能够配置监控告警规则  5.3.1.4 能够根据发布需求，对发布的应用进行运行状态及流量进行监控  5.3.1.5 能够对生产发布环境的操作系统运行状态如CPU、内存、磁盘空间、应用日志进行监控 | 5.3.1.1 开源监控软件历史分类及特点  5.3.1.2 监控架构原则方法论（推/拉）  5.3.1.3 3-5种监控软件安装与使用  5.3.1.4 折线图/柱状图等图表适用场景  5.3.1.5 熟悉操作系统基本原理  5.3.1.6 熟悉批量运维脚本管理和执行工具的使用方法 |
| 5.3.2 管理监控指标 | 5.3.2.1能够根据设计文档部署指标监控工具  5.3.2.2能够根据设计文档配置告警规则，调整监控阈值  5.3.2.3能够根据设计文档定制监控仪表盘，展示指标数据 | 5.3.2.1时序性数据库原理，实现  5.3.2.2操作系统与网络设备常用监控指标  5.3.2.3简单的统计学知识  5.3.2.4 OpenTelemetry标准 |
| 5.3.3 管理收集日志 | 5.3.3.1能够根据设计文档部署日志采集工具  5.3.3.2能够根据设计文档配置日志解析规则  5.3.3.3能够根据设计文档配置基于日志数据的仪表盘 | 5.3.3.1 日志采集工具技术原理  5.3.3.2常见日志打印组件、结构化日志、日志存储技术等知识  5.3.3.3常见设备的日志格式（linux windows），监控仪表盘知识 |
| 5.3.4 规划调用链监控 | 5.3.4.1能够根据设计文档部署调用链采集工具  5.3.4.2能够根据设计文档配置调用链监控规则  5.3.4.3能够根据设计文档配置基于调用链数据的仪表盘 | 5.3.4.1 调用链采集工具技术原理  5.3.4.2 OpenTelemetry、OpenTracing标准等知识  5.3.4.3编程基础，调用链仪表盘知识 |
| 5.3.5 规划智能监控 | 5.3.5.1能够通过统计学方法实现简单的异常检测  5.3.5.2能够通过统计学方法实现简单的告警压缩  5.3.5.3 能够使用可观测性工具进行关联监控 | 5.3.5.1 统计学方法，异常检测技术  5.3.5.2统计学方法，告警压缩技术  5.3.5.3 指标、日志、调用链组合使用技能 |
| 5.4 管理业务连续性 | 5.4.1 确定服务质量目标 | 5.4.1.1 能够支持服务风险度量方案。  5.4.1.1 能够采用多种方式进行指标的收集。 | 5.4.1.1 服务风险度量方案知识，SRE基础知识  5.4.1.2 SLO的定义和SLI的知识  5.4.1.3 不同指标采集方法 |
| 5.4.2 响应与回顾事故 | 5.4.2.1 能够掌握不同层级的故障排查手段（计算、网络、存储、应用、数据库）  5.4.2.2 能够进行On-Call值班活动 | 5.4.2.1 故障排查相关技能。  5.4.2.2 On-Call的工作方式和技能。 |
| 5.4.3 进行灾备管理 | 5.4.3.1 根据灾备管理目标，支持进行方案执行  5.4.3.2 能够支持进行数据备份和恢复 | 5.4.3.1 灾备管理，如RTO、RPO等知识  5.4.3.2 数据备份和恢复等级知识 |
| 5.4.4 应用混沌工程 | 5.4.4.1 能够掌握并执行混沌工程  5.4.4.2 能够反馈混沌工程执行结果 | 5.4.4.1 混沌工程知识  5.4.4.2 混沌工程执行 |

## 3.3中级

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **职业功能** | **工作内容** | **工作内容** | **专业能力要求** | **相关知识要求** |
| 1.组织与协作 | 1.1构建个体能力 | 1.1.1推广敏捷DevOps理念 | 1.1.1.1 能够在团队内推广DevOps理念  1.1.1.2 能够在团队内推广狭义与广义的DevOps  1.1.1.3 能够在团队内推广DevOps价值观与原则  1.1.1.4 能够在团队内推广精益敏捷思想 | 1.1.1.1 敏捷DevOps历史，DevOps三步工作法，CALMR/CALMS  1.1.1.2 狭义与广义的DevOps，敏捷组织，精益企业  1.1.1.3 研发效能（DevOps）价值观与原则  1.1.1.4 精益与敏捷思想 |
| 1.1.2 构建研发效能转型方案 | 1.1.2.1 能够进行转型方案设计  1.1.2.2 能够进行研发效能提升方案制定 | 1.1.2.1 研发效能转型的必要性、收益以及目标的制定；  1.1.2.2 研发效能方案设计，包括组织、流程等； |
| 1.1.3 推动研发效能转型 | 1.1.3.1 能够支持转型方案落地  1.1.3.2 能够通过度量反馈等方法，推进研发效能提升 | 1.1.3.1 敏捷项目管理方法论，PDCA  1.1.3.2 研发效能度量，回顾会议，鱼骨图，5Why等方法 |
| 1.2管理团队协作 | 1.2.1组建团队，确定协作流程及工作环境 | 1.2.1.1 能够确定团队组成结构、组建团队；  1.2.1.2 能够确定与定义团队工作流程；  1.2.1.3 能够制定有助于提升研发效能的沟通协调机制；  1.2.1.4 能够落实支持团队沟通协调的软硬件资源。 | 1.2.1.1 特性团队、组件团队；  1.2.1.2 Scrum of Scrums；  1.2.1.3 敏捷办公环境，沟通协同机制，远程协同办公；  1.2.1.4 协同办公软硬件资源； |
| 1.2.2制定交付计划 | 1.2.2.1 能够组织团队制定发布计划；  1.2.2.2 能够组织团队制定交付计划； | 1.2.2.1 发布计划与流程 1.2.2.2 交付计划与流程 |
| 1.2.3跟踪执行计划 | 1.2.3.1 能够跟踪团队进度  1.2.3.2 能够管理团队问题和风险  1.2.3.3 能够更新团队计划 | 1.2.3.1每日站会、任务管理知识  1.2.3.2 问题和风险知识  1.2.3.3 站会会后跟进会议 |
| 1.2.4优化工作方式 | 1.2.4.1 能够组织团队识别改进点；  1.2.4.2 能够组织团队制定改进计划；  1.2.4.3 能够跟踪团队改进计划 | 1.2.4.1每日站会、演示会议、敏捷度量与反馈知识  1.2.4.2回顾会议  1.2.4.3行动计划与跟踪、SMART原则 |
| 1.3建设学习型组织 | 1.3.1 营造组织持续学习环境 | 1.3.1.1 能够举办持续改进活动  1.3.1.2 能够建立组织改进资产库 | 1.3.1.1 CoP 、开放日、Game Day、猿猴军团、读书会、黑客马拉松等  1.3.1.2 知识管理、案例集、白皮书/实践指南 |
| 1.3.2 建设生态型组织 | 1.3.2.1 能够针对组织的商业生态与开发生态建设规划指定对应的实施计划，指导并推动团队落地这些规划  1.3.2.2 能够评估不同的内外部组件的能力、应用范围和安全性影响；指定并实施相关的管控策略  1.3.2.3 能够参与内外部生态环境和社区，并形成良性的互动关系  1.3.2.4 能够使用源代码管理工具、制品库和其他相关工具落实内外源管理控制测策略 | 1.3.2.1 内源与开源机制，各类开源协议，社区开源渠道，开源基金会运作方式  1.3.2.2 组件应用范围和安全性，组件管控机制  1.3.2.3 内外部生态与社区  1.3.2.4 业界常见源代码管理工具、各种开发技术栈所涉及的包管理工具以以及制品库的基本功能； |
| 1.4度量研发效能 | 1.4.1 度量个体效能 | 1.4.1.1 能够定义度量数据  1.4.1.2 能够分析个人度量数据趋势  1.4.1.3 能够提出优化建议 | 1.4.1.1价值流交付模型（需求提出到发布端到端价值流）  1.4.1.2任务计划工时，任务填报工时，任务剩余工时，计划达成率  1.4.1.3背景-问题-改进方案，STAR原则 |
| 1.4.2 度量团队效能 | 1.4.2.1 能够定义和跟踪团队产能  1.4.2.2 能够定义和跟踪交付效率  1.4.2.3 能够定义和跟踪产品质量  1.4.2.4 能够分析和优化团队效能 | 1.4.2.1 迭代速度，需求吞吐量  1.4.2.2 需求交付周期，代码变更前置时间，发布频率  1.4.2.3 缺陷库存数，线上缺陷新增数，缺陷关闭率，缺陷解决时长，缺陷关闭时长，千行代码缺陷率，故障平均恢复时间，故障平均响应时间  1.4.2.4 趋势图，雷达图，数据分析，相关性分析 |
| 2. 规划与运营 | 2.2 设计产品 | 2.2.1管理产品需求 | 2.2.1.1 能够根据需求梳理产品路线图  2.2.1.2 能够组织需求规模估算  2.2.1.3 能够根据业务价值，排定需求优先级 | 2.2.1.1 产品路线图，如影响地图、用户故事地图、产品版本规划、MVP等  2.2.1.2 需求规模估算方法，如故事点、T恤Size等；估算工具如计划扑克  2.2.1.3 排定优先级，如MoSoCoW、Kano、WSJF等 |
| 2.2.3 设计产品界面与用户体验 | 2.2.3.1 能够根据产品需求完成整个产品原型图绘制  2.2.3.2 能够根据产品需求设计整个产品的交互效果  2.2.3.3 能够指导产品设计和优化产品用户体验。 | 2.2.3.1设计产品原型知识，如以用户为导向的设计、面向场景设计、扩展性设计、原型图设计工具等；  2.2.3.2交互效果知识，如整体产品的交互体验设计、交互设计的指导原则等；常用交互设计理论，如米勒定律、席克定律、费茨定律、泰勒斯定律、奥卡姆剃刀定律、接近原则、防错原则等；  2.2.3.3 用户体验设计，如什么是用户体验设计、用户体验设计流程、用户体验设计标准等；用户体验设计框架：战略层、范围层、结构层、框架层、视觉层等； |
| 2.3 产品运营 | 2.3.1制定运营策略 | 2.3.1.1能够整理并输出产品功能模块或功能点的核心目标用户、应用场景、用户需求，并能够评估使用产品后带给用户的改善效果；  2.3.1.2能够制定对产品功能模块或功能点的增长指标，包含行为结果指标、过程指标与用户质量指标。 | 2.3.1.1卖点挖掘与价值提炼等知识，如用户痛点分析、用户痛点整理、产品卖点提炼、产品价值包装等；  2.3.1.2运营指标定义与设计知识，如事件定义、指标定义、AARRR海盗模型等； |
| 2.3.2监测激活指标 | 2.3.2.1 具备设计用户激活模型的能力，能够推进用户激活模型的落地，搭建监测系统看板。  2.3.2.2 能够通过用户激活监测模型，围绕具体业务流程开展用户留存分析、行为路径分析、用户质量分析、渠道质量分析，并诊断用户流失原因，提出产品迭代与优化策略。 | 2.3.2.1 数据获取，如数据埋点、数据挖掘、数据清洗、数据整理等；  2.3.2.2 数据应用，如分析模型搭建、数据看板搭建、数据指标监测等； |
| 3.开发与交付 | 3.1 协同开发应用 | 3.1.1 设计解决方案 | 3.1.1.1 能够掌握既定的解决方案，并完成相关的实施工作  3.1.1.2 能够根据项目推进情况，识别风险、问题，提供分析和对应的解决方案 | 3.1.1.1 团队组织、技术选型、基础设施需求、用户需求、业务需求、项目预算、项目时间表、安全和合规性需求  3.1.1.2 项目风险管理、问题识别、分析与解决方案等 |
| 3.1.2 持续改进开发流程 | 3.1.2.1 能够根据组织协作流程指导和执行相应的开发流程，指定计划推动实施  3.1.2.2 能够按照敏捷开发思路指导团队进行技术模块设计和工作拆分 | 3.1.2.1 各类XP实践、BDD，ATDD等方法  3.1.2.2 模块拆分原则匹配用户故事场景 |
| 3.1.3 管理和优化源代码流程及工具 | 3.1.3.1 能够指导团队代码分支模型、代码审查流程持续进行小批量代码提交及解决冲突  3.1.3.2 能够指导团队基于代码分支模型、测试流程规范地实施测试执行  3.1.3.3 能够根据分支模型，检查守护团队分支模型设置  3.1.3.4 能够指导团队成员掌握并使用源代码管理工具 | 3.1.3.1 代码审查、小批量提交代码的方法、规范提交代码的方法  3.1.3.2 测试执行及门禁设置  3.1.3.3 分支保护设置方法、Webhook等  3.1.3.4 源代码管理工具使用 |
| 3.2 应用持续集成 | 3.2.1 搭建与实施流水线 | 3.2.1.1 能够基于流水线模型配置统一的流水线模板，指导团队规范建立完整端到端流水线  3.2.1.2 能够基于流行的工具实施流水线监控方案，确保流水线稳定可用  3.2.1.3 能够基于工具实施三方依赖包管理策略、构建方案，确保自动构建稳定可用  3.2.1.4 能够基于流行的工具实施流水线日志方案，便于遇到问题快速定位分析  3.2.1.5 能够在流水线模板中设置多道质量门禁并指导流水线规范使用 | 3.2.1.1 流水线模板设计制作方法、基于工具搭建流水线的方法  3.2.1.2 流水线采集与监控设计方法  3.2.1.3 编绎原理、构建脚本编程基础知识、自动化构建基础知识、依赖包管理工具使用方法  3.2.1.4 流水线中集成日志系统的方法  3.2.1.5 质量门禁原理、质量门禁集成方法 |
| 3.2.2 设计与集成质量门禁 | 3.2.2.1 能够根据质量门禁设计方案指导团队实现质量门禁与研发管理流程融合  3.2.2.2 能够根据业务、技术发展状态优化质量门禁策略，提升团队效能  3.2.2.3 能够指导团队集成自动化测试、代码扫描、手工测试插件 | 3.2.2.1 质量门禁原理、质量门禁与研发流程融合方法  3.2.2.2 质量门禁指标定义  3.2.2.3 质量门禁中集成自动化工具的方法、自动化测试及扫描的工具知识、质量门禁指标定义 |
| 3.2.3 集成与应用代码质量保证 | 3.2.3.1 能够依据代码质量保证方案在质量门禁中集成代码质量保证工具，并输出报告  3.2.3.2 能够组织技术专家设计代码规约并配置于代码代码质量保证工具中  3.2.3.3 能够根据团队内外技术发展趋势，及时调整优化代码质量保证方案 | 3.2.3.1 代码质量保证工具集成方法  3.2.3.2 代码扫描规则配置方法  3.2.3.3 代码质量保证方案原理 |
| 3.2.4 设计与应用单元测试 | 3.2.4.1 能够实施单元测试环境、平台搭建  3.2.4.2 能够在质量门禁中集成单元测试自动化，并输出报告  3.2.4.3 能够根据业务、技术发展状态，及时调整自动化测试策略 | 3.2.4.1 单元测试平台搭建方法、单元测试环境部署方法  3.2.4.2 质量门禁中集成单元测试插件的方法、单元测试工具结果输出的方法  3.2.4.3 自动化测试策略调整方法 |
| 3.3 应用持续部署 | 3.3.1 装配与管理环境 | 3.3.1.1 能编写脚本申请和装配服务器操作系统与中间件软件  3.3.1.2 能搭建持续交付流水线和脚本任务批量管理体系 | 3.3.1.1 公有云、私有云和虚拟化平台使用方法与API调用知识  3.3.1.2 常见操作系统和中间件软件的自动化装配知识  3.3.1.3 部署流水线和脚本任务管理软件的搭建知识 |
| 3.3.2 部署与更新版本 | 3.3.2.1 能编写包含级联和多方集成的复杂应用部署与变更脚本  3.3.2.2 能将自动化脚本制作为应用部署与更新流水线和模板 | 3.3.2.1 服务器、数据库和中间件脚本编程知识  3.3.2.2 自动化流水线和脚本任务管理软件的搭建和使用知识 |
| 3.4 应用按需发布 | 3.4.1 设计生产发布策略 | 3.4.1.1能够根据不同的场景和需求设计匹配的发布策略，例如灰度/滚动发布、蓝绿发布、A/B 测试、金丝雀发布等；  3.4.1.2能够根据常见的发布策略，设计发布和回退脚本。 | 3.4.1.1 发布策略，例如灰度/滚动发布、蓝绿发布、A/B 测试、金丝雀发布等；  3.4.1.2 掌握常见中间件的安装和配置；  3.4.1.3 开发部署脚本、回退脚本等； |
| 3.4.2 持续发布生产 | 3.4.2.1 能根据常见应用发布策略，编写流量网关策略规则及自动化脚本  3.4.2.2 能根据常见应用发布策略，设计功能开关的配置脚本 | 3.4.2.1 熟悉网络协议的基础知识  3.4.2.2 掌握常见流量网关软件的策略编写方法 |
| 3.4.3 设计交付工具链及流程 | 3.4.3.1能够提升系统自动化回归测试能力，快速验证按需发布功能，保障交付质量  3.4.3.2能够运用容器技术实现测试环境的快速搭建和伸缩  3.4.3.3能够借助工具快速实现影响度分析，识别测试范围 | 3.4.3.1 掌握容器与编排工具的基本原理和安装配置  3.4.3.2 掌握常见的自动化测试工具  3.4.3.3 测试影响度分析，测试范围 |
| 3.4.4 预发布与运营拟真 | 3.4.4.1 能根据高可用要求及处置预案制定演练步骤  3.4.4.2 能为发布计划和演练步骤收集用于效果验证的关键遥测与预警指标数据 | 3.4.4.1 虚拟化、云和容器集群和常见操作系统与中间件软件的原理  3.4.4.2 数据统计分析知识  3.4.4.3 系统容量评估知识 |
| 3.5 开发环境管理 | 3.5.1 优化开发调试环境 | 3.5.1.1 能够指导和搭建高效的开发调试环境，制定计划推动实施，包括开发人员本地开发环境和团队共享环境  3.5.1.2 能够有效评估和引入对开发调试有帮助的各类工具和实践，通过管理和技术手段解决开发人员/团队的日常技术问题 | 3.5.1.1 开发调测环境知识  3.5.1.2 各类源代码管理、开发技术栈以及相关工具链的特点 |
| 3.5.2 管理开发环境安全 | 3.5.2.1 能够根据既定的管理策略设计具体的安全保障机制  3.5.2.1 能够根据安全保障机制进行工具选型 | 3.5.2.1 高级安全概念、可信用户界面设计、安全漏洞细节、自定义威胁缓解  3.5.2.1 安全工具 |
| 4.测试与安全 | 4.1 测试组织管理 | 4.1.1 采纳测试方法论 | 4.1.1.1 能够根据现有的国际和国内测试标准，分析应用测试需求，明确测试目标，编写测试计划文档  4.1.1.2 能够根据测试计划与测试制度，管控指导测试流程 | 4.1.1.1 测试标准知识，测试计需求，测试目标，测试计划；  4.1.1.2 需求分析技术 |
| 4.1.2 管理测试团队 | 4.1.2.1 能够根据测试计划，协调人力测试资源  4.1.2.2 能够根据测试计划，安排分配计划任务，确保计划执行效果  4.1.2.3 能够判断团队成员水平能否满足测试需求，并指导水平较低的成员 | 4.1.2.1 测试人力资源管理，测试计划  4.1.2.2 测试计划与跟踪，测试任务管理  4.1.2.3 团队项目管理知识 |
| 4.1.3 采纳测试技术与管理平台工具 | 4.1.3.1 能够根据现有测试计划方案，制定并改进测试平台功能与流程，选择测试工具  4.1.3.2 能够根据测试计划，选择软件测试模型与技术，安排并指导团队使用测试平台与工具  4.1.3.3 能够根据团队与产品测试需求，编写测试用例 | 4.1.3.1 测试平台功能与流程  4.1.3.2 测试平台与工具管理  4.1.3.3 测试用例编写 |
| 4.2 测试生命周期管理 | 4.2.1 制定测试策略与计划 | 4.2.1.1 能够组织并开展测试策略分析  4.2.1.2 能够分析测试任务，编制测试方案或测试计划  4.2.1.3 能够识别测试干系人 | 4.2.1.1 测试风险管理知识  4.2.1.2 测试方针和目标知识  4.2.1.3 测试策略知识  4.2.1.4 测试方案或测试计划知识 |
| 4.2.2 管理测试需求 | 4.2.2.1 能够组织测试点的分析  4.2.2.2 能够按照测试策略中的业务需求和技术需求结构化测试任务  4.2.2.3 能够制定并维护测试需求和测试用例的跟踪规则 | 4.2.2.1 测试评审技能知识  4.2.2.2 测试分析技能知识  4.2.2.3 需求管理知识  4.2.2.4 需求变更管理知识 |
| 4.2.3 设计测试用例 | 4.2.3.1 能够组织编写完成测试用例  4.2.3.2 能够组织测试用例评审 | 4.2.3.1 测试用例设计方法（等价类、边界值、因果图等）  4.2.3.2 测试优先级知识  4.2.3.3 测试评审知识 |
| 4.2.4 执行测试并报告 | 4.2.4.1 能够组织测试用例的执行  4.2.4.2 能够控制进度和跟踪任务  4.2.4.3 能够根据测试目标和策略,选择、获取和分派任务所需的足够资源  4.2.4.4 能够根据测试过程的暂停和退出规则，在项目中进行裁剪和应用，并决策测试执行的暂停和退出  4.2.4.5 能够评审测试报告内容 | 4.2.4.1执行测试用例技能知识（静态测试、动态测试）  4.2.4.2 测试暂停和退出技能知识  4.2.4.3 测试报告的相关技能知识  4.2.4.4 测试过程管理  4.2.4.5 测试进度与跟踪 |
| 4.2.5 管理与预防缺陷 | 4.2.5.1 能够根据测试过程数据和缺陷汇报结果  4.2.5.2 能够对于产品质量和风险进行评估 | 4.2.5.1 缺陷分析，产品质量  4.2.5.2 风险评估 |
| 4.3 测试关键域管理 | 4.3.1 执行测试估算 | 4.3.1.1 能够进行测试估算  4.3.1.2 能够基于估算结果设计测试计划 | 4.3.1.1 常用的测试估算方法  4.3.1.2 制定测试估算计划的能力 |
| 4.3.2 制定测试度量 | 4.3.2.1 能够根据组织级测试度量体系制定项目或团队相关的测试度量指标  4.3.2.2 能够根据测试度量的数据完成测试度量监控，并根据度量结果完成改进 | 4.3.2.1 缺陷密度、漏测率、测试覆盖率等测试度量指标知识  4.3.2.2 燃尽图、燃起图、速率、鱼骨图、帕累托图等知识 |
| 4.3.3 使用测试技术 | 4.3.3.1 能够使用白盒测试、黑盒测试、灰盒测试的基本方法进行测试设计  4.3.3.2 能够使用功能测试，性能测试，安全测试的基本方法进行测试设计  4.3.3.3 能够使用手动测试和自动化测试的基本方法进行测试设计 | 4.3.3.1 白盒测试中语句覆盖、条件覆盖、路径覆盖等测试基本方法；渗透测试、压力测试等黑盒测试的基本方法及相关工具  4.3.3.2 功能测试，性能测试，安全测试知识及相关工具  4.3.3.3手动测试和自动化测试知识及相关工具 |
| 4.3.4 执行测试阶段 | 4.3.4.1 能够完成单元测试用例设计及测试执行  4.3.4.2 能够完成集成测试用例设计及测试执行 | 4.3.4.1 单元测试知识  4.3.4.1 集成测试知识，自顶向下集成、自底向上集成、API测试、契约测试 |
| 4.3.5 执行自动化功能测试 | 4.3.5.1 能够设计与搭建自动化测试框架  4.3.5.2 能够开发自动化测试脚本 | 4.3.5.1 自动化测试框架知识  4.3.5.2 自动化脚本开发知识，自动化测试工具知识 |
| 4.3.6 执行非功能性测试 | 4.3.6.1 能够完成设计兼容性测试场景及测试用例，并且执行测试  4.3.6.2 能够完成设计可用性测试场景及测试用例，并且执行测试  4.3.6.3 能够完成设计稳定性测试场景及测试用例，并且执行测试 | 4.3.6.1 兼容性测试知识  4.3.6.2 可用性测试知识  4.3.6.3 稳定性测试知识 |
| 4.3.7 执行性能测试 | 4.3.7.1 能够根据业务场景设计性能负载模型  4.3.7.2 能够基于性能测试工具，开发性能测试脚本 | 4.3.7.1 性能负载设计知识  4.3.7.2 性能脚本开发知识，性能测试工具的使用 |
| 4.3.8 管理测试数据 | 4.3.8.1 能够根据掌握的测试数据构造基本方法进行设计  4.3.8.2 能够分析测试数据并完成数据脱敏 | 4.3.8.1 通过GUI、API、数据库操作构造测试数据的知识  4.3.8.2 通过脱敏规则进行数据的变形，实现敏感隐私数据的可靠保护的知识 |
| 4.3.9 管理测试环境 | 4.3.9.1 能够运用云计算/容器技术设计或创建完整的测试环境  4.3.9.2 能够运用设计并构建完整的测试环境，包括但不限于物理机、虚机、容器等技术 | 4.3.9.1 测试虚拟化的基本知识  4.3.9.2 云计算中IaaS、PaaS和容器技术快速构建测试环境的知识 |
| 4.3.10 执行敏捷测试 | 4.3.10.1 能够运用测试金字塔原理设计自动化测试策略  4.3.10.2 能够开展探索式测试 | 4.3.10.1 测试金字塔知识  4.3.10.2 探索式测试知识 |
| 4.4 安全防护与安全测试 | 4.4.1 管理应用安全生命周期 | 4.4.1.1 能够根据现有的国际和国内安全相关标准，分析应用安全测试需求，明确测试目标，编写测试计划文档  4.4.1.2 能够根据安全测试计划与合规要求，管控指导测试流程  4.4.1.3 能够主导进行某一类应用的安全生命周期管理流程设计 | 4.4.1.1 应用安全生命周期管理的基本知识  4.4.1.2 DevSecOps理念与实践  4.4.1.3 应用安全生命周期各阶级的定义  4.4.1.4 应用生命周期各阶级的安全实践 |
| 4.4.2 设计应用安全测试方案 | 4.4.2.1 能够主导应用安全生命周期管理中的各个阶段进行安全测试方案设计  4.4.2.2 能够选择应用安全生命周期管理中的各个阶段合适的测试技术 | 4.4.2.1 安全相关模型，如IPDRR、CIA、STRIDE威胁模型等  4.4.2.2 安全工具金字塔  4.4.2.3 应用生命周期各阶级的安全实践  4.4.2.4 安全策略知识，例如最小特权原则、最小泄露原则和多级安全策略等 |
| 4.4.3 执行应用安全测试方案 | 4.4.3.1 能够结合SAST静态安全测试工具进行静态安全检查策略设计  4.4.3.2 能够结合DAST动态安全测试工具进行动态安全检查策略设计  4.4.3.3 能够结合WAF、IDS/IPS工具进行安全规则防护策略设计  4.4.3.4 能够进行软件组成分析（SCA），设计开源和外来组件的安全治理策略  4.4.3.5 能够结合IAST交互式安全测试工具进行交互式安全检查策略设计并执行  4.4.3.6 能够结合RAST运行时应用自保护工具进行应用运行时应用自保护策略设计并执行  4.4.3.7 能够结合渗透测试工具进行渗透测试设计并执行  4.4.3.8 能够结合模糊测试（fuzz testing）工具进行模糊测试设计并执行 | 4.4.3.1 SAST静态安全测试工具与静态安全检查知识  4.4.3.2 DAST动态安全测试工具与动态安全检查知识  4.4.3.3 WAF、IDS/IPS工具与安全规则防护知识  4.4.3.4 软件组成分析（SCA），开源安全治理  4.4.3.5 IAST交互式安全测试工具与交互式安全检查知识  4.4.3.6 RAST运行时应用自保护工具与应用运行时应用自保护知识  4.4.3.7 渗透测试工具与渗透测试知识  4.4.3.8 模糊测试（fuzz testing）工具与模糊测试知识  4.4.3.9 安全测试结果分析及修复知识 |
| 4.4.4 执行云安全防护 | 4.4.4.1 能够运用工具设计安全流水线策略  4.4.4.2 能够运用工具设计容器安全检查策略  4.4.4.3 能够运用安全知识设计并构建基础的云安全防护机制 | 4.4.4.1 安全流水线各阶段的知识，如预提交检查、提交时检查、构建时检查、测试时检查、部署时检查等  4.4.4.2 容器与编排安全知识  4.4.4.3 安全加固与云安全的基础知识 |
| 5. 运维与监控 | 5.1 IT服务管理 | 5.1.1 制定IT服务管理体系 | 5.1.1.1 能够根据IT服务管理体系，编制各项IT服务管理流程及标准作业程序。  5.1.1.2 能够落实IT服务管理平台规划，完成落地实施与运营。  5.1.1.3 能够贯彻IT服务管理体系，监控执行绩效，并进行持续改进。  5.1.1.4 能够支持研发及服务提供商，高质量、高效端到端交付能力。 | 5.1.1.1 IT服务管理主要流程，如事件管理、问题管理、变更管理、容量管理、可用性管理、配置管理等知识  5.1.1.2 IT服务管理工具，如事件管理、问题管理、变更管理、配置管理、容量管理、可用性管理等知识  5.1.1.3 IT服务管理体系知识，绩效管理知识，PDCA  5.1.1.4 DevOps持续交付知识 |
| 5.1.2 管理事件与问题 | 5.1.2.1 能够根据事件与问题管理总体设计，编写事件与问题管理流程及标准作业程序。  5.1.2.2 能够贯彻事件与问题管理流程，监控执行绩效，并进行持续改进。  5.1.2.3 能够落实IT服务管理平台事件与问题管理流程，完成落地实施与管理。  5.1.2.4 能够建立研发及服务提供商的统一事件与问题处理与协调机制，监控执行绩效，并进行持续改进。 | 5.1.2.1 IT服务管理常规流程，如事件管理、问题管理等知识  5.1.2.2 IT服务管理体系知识，绩效管理知识，PDCA  5.1.2.3 IT服务管理工具，如事件管理、问题管理等知识  5.1.2.4 IT服务管理问题流程设计知识 |
| 5.1.3 管理变更与配置 | 5.1.3.1 能够根据变更与配置管理总体设计，编写变更与配置管理流程及标准作业程序。  5.1.3.2 能够贯彻变更与配置管理流程，监控执行绩效，并进行持续改进。  5.1.3.3 能够落实IT服务管理平台变更与配置管理流程及CMDB管理，完成落地实施与管理。  5.1.3.4 能够建立统一变更与配置管理的协调机制，监控执行绩效，并进行持续改进。 | 5.1.3.1 IT服务管理流程，如变更管理流程、配置与资产管理流程、发布管理流程等知识  5.1.3.2 IT服务管理体系知识，绩效管理知识，PDCA  5.1.3.3 IT服务管理工具，如变更与配置管理流程、CMDB等知识  5.1.3.4 IT服务管理流程设计知识 |
| 5.1.4 管理容量与成本 | 5.1.4.1 能够根据容量与成本管理总体设计，编写容量与成本管理流程及标准作业程序。  5.1.4.2 能够贯彻容量与成本管理流程，监控执行绩效，并进行持续改进。  5.1.4.3 能够根据组织业务情况，设计容量计划，并持续维护满足未来需求。  5.1.4.4 能够评估变更对容量计划、服务和资源的性能和容量带来的影响。就容量和性能相关问题，为业务和其他领域提供建议和指导。通过管理服务和资源的性能和容量，确保服务性能达到或超过约定的性能目标。实施主动测量来改进服务的性能。 | 5.1.4.1 IT服务管理流程设计知识  5.1.4.2 IT服务管理流程，如容量管理、可用性管理等知识  5.1.4.3 容量计划知识  5.1.4.4 IT服务管理流程，如容量管理、可用性管理、变更管理、测量工具与方法等知识 |
| 5.2 应用运行管理 | 5.2.1 管理应用运行方案 | 5.2.1.1 能够进行应用运行方案制定，确保其正常运行，并持续改进和优化。  5.2.1.2 能够根据应用架构设计，合理选择运行资源，制定应用运行方案  5.2.1.3 能够根据应用与业务规划，设计对应的应用部署方案（包括高可用方案） | 5.2.1.1 基础设施、中间件、数据库知识  5.2.1.2 应用架构知识、微服务知识、容器知识  5.2.1.3 灾备、高可用知识 |
| 5.2.2 管理基础设施平台 | 5.2.2.1 能够根据应用运行方案，制定基础设施平台整体架构  5.2.2.2 能够根据应用运行方案，设计基础设施平台的架构拓扑，为应用运行方案提供底层技术支撑  5.2.2.3 能够根据应用运行方案，设计网络数据流向与数据存储方案  5.2.2.4能够根据应用运行的部署方案和高可用方案，指导相关技术人员执行基础设施的整体交付与管理 | 5.2.2.1 计算、存储、网络、操作系统知识  5.2.2.2 VPC/子网/安全组/防火墙/WAF/端口/NAT/弹性IP知识  5.2.2.3 对象存储/文件存储/RAID知识、数据管理知识、网络管理知识  5.2.2.4 硬件负载均衡、软件负载均衡、云负载均衡知识 |
| 5.2.3 管理应用运行环境 | 5.2.3.1 能够根据应用运行方案，选择合适的自动化技术，设计自动化实施方案  能够设计并实现运行环境的批量自动化部署与交付方案，并在此基础上实现应用的自动化部署与更新  5.2.3.2 能够根据应用架构与业务特性需要，设计应用不同的持续发布策略、负载均衡策略、熔断策略，以技术手段实现并提供支持  5.2.3.3 能够根据应用的运行方案制定相应的灾备方案，指导团队进行灾备的技术实施与技术演练 | 5.2.3.1 应用运行环境自动化工具，开发语言知识  5.2.3.2 流水线、CI/CD、集群与高可用知识  5.2.3.3 灾备/容灾知识、负载均衡知识 |
| 5.3 设计管理应用监控 | 5.3.1 设计与建设监控系统 | 5.3.1.1 能够根据复杂场景设计出对应的监控架构以及技术选型  5.3.1.2 能够设计覆盖全部监控要素的全域监控体系，制定监控标准  5.3.1.3 能够设计常用的可视化图表类型以及应用场景  5.3.1.4 能够设计监控告警规则  5.3.1.5 能够根据发布应用产生的问题进行分析，并能通过流量控制、应用回滚、功能开关等方式进行应急  5.3.1.6 能够对生产发布环境的基础设施软件如数据库、中间件等运行状态进行监控  5.3.1.7 可进行发布软件的性能、安全性等非功能指标的监控分析 | 5.3.1.1 全域监控体系  5.3.1.2 监控对象指标  5.3.1.3 可视化图表  5.3.1.4 监控告警规则  5.3.1.5 流量网关和负载均衡软件的使用方法  5.3.1.6 常用的监控工具使用方法  5.3.1.7 性能、安全性等非功能指标监控分析 |
| 5.3.2 管理监控指标 | 5.3.2.1 能够编写指标监控工具设计文档  5.3.2.2 能够编写告警规则设计文档  5.3.2.3 能够编写监控仪表盘设计文档 | 5.3.2.1 指标监控工具  5.3.2.2 告警规则  5.3.2.3 指标采集工具部署，监控仪表盘 |
| 5.3.3 管理收集日志 | 5.3.3.1 能够编写部署日志采集工具设计文档  5.3.3.2 能够编写日志解析规则设计文档  5.3.3.3 能够编写基于日志数据的仪表盘设计文档  5.3.3.4 能够设计简单结构化的日志监控工具，覆盖收集、整理、存储等 | 5.3.3.1 日志采集工具知识  5.3.3.2 日志解析原理  5.3.3.3 日志数据仪表盘  5.3.3.4 日志代理采集原理 |
| 5.3.4 规划调用链监控 | 5.3.4.1 能够编写调用链采集工具部署设计文档  5.3.4.2 能够编写调用链监控规则配置设计文档  5.3.4.3 能够编写基于调用链数据仪表盘配置设计文档 | 5.3.4.1 调用链采集工具技术原理  5.3.4.2 调用链监控规则配置  5.3.4.3 调用链数据仪表盘配置  5.3.4.4 应用性能监控软件的技术原理以及实现方式  5.3.4.5 OpenTracing字段标准 |
| 5.3.5 规划智能监控 | 5.3.5.1 能够利用智能运维算法实现单指标的异常检测  5.3.5.2 能够通过智能运维算法实现基于时间维度和相似文本的告警压缩  5.3.5.3 能够使用可观测性工具进行趋势关联分析 | 5.3.5.1 智能运维相关算法及算法所适用的场景，如随机森林、关联分析、异常检测  5.3.5.2 算法所适用的场景  5.3.5.3 趋势关联算法 |
| 5.4 业务连续性管理 | 5.4.1 确定服务质量目标 | 5.4.1.1 能够根据应用系统的架构，使用不同的服务风险度量方案。  5.4.1.2 能够根据应用的SLO服务级别目标进行SLI服务级别指标的拆分。 | 5.4.1.1 服务风险度量方案知识，SRE基础知识  5.4.1.2 SLO的定义和SLI的知识 |
| 5.4.2 响应与回顾事故 | 5.4.2.1 能够组织开展故障复盘会议，进行故障回顾  5.4.2.2 能够针对应用系统设计限流、降级等治理方案。  5.4.2.3 能够承担On-Call值班活动 | 5.4.2.1 故障排查相关技能。  5.4.2.2 On-Call的工作方式和技能。  5.4.2.3 故障回顾技术，如故障复盘会议。  5.4.2.4 应用系统应急预案知识。 |
| 5.4.3 进行灾备管理 | 5.4.3.1 能够进行灾难恢复能力等级的划分和评定  5.4.3.2 能够进行不同层级的数据备份和恢复  5.4.3.3 根据根据不同的灾备等级进行方案执行 | 5.4.3.1 灾难恢复能力等级划分和评定  5.4.3.2 数据备份和恢复等级  5.4.3.3 灾备等级方案 |
| 5.4.4 应用混沌工程 | 5.4.4.1 能够根据混沌工程的实验结果反馈应用和架构优化建议。  5.4.4.2 能够进行混沌工程方案规划和设计。 | 5.4.4.1 混沌工程知识  5.4.4.2 混沌工程执行  5.4.4.3 公有云和容器化环境下的混沌工程实践 |

## 3.4高级

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **职业功能** | **工作内容** | **工作内容** | **专业能力要求** | **相关知识要求** |
| 1.组织与协作 | 1.1构建个体能力 | 1.1.1推广敏捷DevOps理念 | 1.1.1.1 能够在组织级别推广DevOps理念  1.1.1.2 能够在组织级别推广狭义与广义的DevOps  1.1.1.3 能够在组织级别推广DevOps价值观与原则  1.1.1.4 能够在组织级别推广精益敏捷思想 | 1.1.1.1 敏捷DevOps历史，DevOps三步工作法，CALMR/CALMS  1.1.1.2 狭义与广义的DevOps，敏捷组织，精益企业  1.1.1.3 研发效能（DevOps）价值观与原则  1.1.1.4 精益与敏捷思想 |
| 1.1.2 构建研发效能转型方案 | 1.1.2.1 能够主导评估当前研发效能状况  1.1.2.2 能够主导进行研发效能转型方案设计  1.1.2.3 能够主导制定研发效能提升方案 | 1.1.2.1 研发效能转型的必要性、收益以及目标的制定；  1.1.2.2 研发效能方案设计，包括组织、流程等； |
| 1.1.3 推动研发效能转型 | 1.1.3.1 能够主导制定转型路线，制定相关举措及时间表，包括组织、人员、实践等层面  1.1.3.2 能够结合转型路线，进行平台工具选型方案的制定 | 1.1.3.1 在组织、人员、实践等层面的转型路径、相关举措及时间表  1.1.3.2 DevOps平台工具方案评估 |
| 1.2管理团队协作 | 1.2.1组建团队，确定协作流程及工作环境 | 1.2.1.1 能够主导制定多团队组织结构并帮助组建团队；  1.2.1.2 能够围绕价值流，定义并建立跨团队的工作流程；  1.2.1.3 能够建立跨团队沟通、协调和协同机制； | 1.2.1.1 Scrum、Kanban等敏捷开发方法  1.2.1.2 研发协同知识 |
| 1.2.2制定交付计划 | 1.2.2.1 能够围绕价值流，策划并制定跨团队发布计划；  1.2.2.2 能够围绕价值流，策划并制定跨团队交付计划； | 1.2.2.1 发布计划与流程 1.2.2.2 交付计划与流程 |
| 1.2.3跟踪执行计划 | 1.2.3.1 能够管理多团队间依赖  1.2.3.2 能够处理价值流问题和风险，更新价值流计划 | 1.2.3.1每日站会、任务管理知识  1.2.3.2 问题和风险知识  1.2.3.3 站会会后跟进会议 |
| 1.2.4优化工作方式 | 1.2.4.1 能够组织优化价值流  1.2.4.2 能够组织制定价值流优化计划  1.2.4.3 能够组织跟踪价值流优化措施 | 1.2.4.1每日站会、演示会议、敏捷度量与反馈知识  1.2.4.2回顾会议  1.2.4.3行动计划与跟踪、SMART原则 |
| 1.3建设学习型组织 | 1.3.2 建设生态型组织 | 1.3.2.1 能够有效的识别，评估和引入外部/开源生态的相关技术、组件和工具，建立内源机制，辅助团队提升研发效率，同时控制风险  1.3.2.2 能够有效的设计、指导和建立企业内部技术生态环境和社区，并与外部生态和社区形成良性的互动关系 | 1.3.2.1 内源与开源机制，各类开源协议，社区开源渠道，开源基金会运作方式  1.3.2.2 内外部生态与社区 |
| 1.3.3 建立组织的持续改进机制 | 1.3.3.1 能够建立过程改进团队  1.3.3.2 能够制定过程持续改进策略与机制 | 1.3.3.1 利益相关者识别、过程改进团队人员构成  1.3.3.2 渐进式/激进式等策略、SWOT、TOC、过程改进团队运行机制 |
| 1.4度量研发效能 | 1.4.1 度量个体效能 | 1.4.1.1 能够设计度量体系  1.4.1.2 能够洞察个人度量数据趋势  1.4.1.3 能够在组织层面推广优化建议 | 1.4.1.1价值流交付模型（需求提出到发布端到端价值流）  1.4.1.2任务计划工时，任务填报工时，任务剩余工时，计划达成率  1.4.1.3背景-问题-改进方案，STAR原则 |
| 1.4.2 度量团队效能 | 1.4.2.1 能够基于组织效能，指导团队产能提升  1.4.2.2 能够基于组织效能，指导交付效率提升  1.4.2.3 能够基于组织效能，指导产品质量提升  1.4.2.4 能够基于组织效能，指导团队效能提升 | 1.4.2.1 迭代速度，需求吞吐量  1.4.2.2 需求交付周期，代码变更前置时间，发布频率  1.4.2.3 缺陷库存数，线上缺陷新增数，缺陷关闭率，缺陷解决时长，缺陷关闭时长，千行代码缺陷率，故障平均恢复时间，故障平均响应时间  1.4.2.4 趋势图，雷达图，数据分析，相关性分析 |
| 1.4.3度量组织效能 | 1.4.3.1 能够定义和跟踪产品维度度量模型  1.4.3.2 能够定义和跟踪DevOps成熟度模型  1.4.3.3 能够定义和跟踪工时投入模型  1.4.3.4 能够定义和跟踪投入回报模型  1.4.3.5 能够分析和优化组织效能 | 1.4.3.1 产品维度产能、效率、质量  1.4.3.2 团队敏捷DevOps成熟度  1.4.3.3 工时填报率，工时投入分布  1.4.3.4 需求收益达成率，项目收益达成率，OKR达成度  1.4.3.5 价值流映射分析及度量，因果回路图 |
| 2.设计与运营 | 2.2 设计产品 | 2.2.1管理产品需求管 | 2.2.1.1 能够基于行业发展构建产品路线图，能够完成长期规划和多个版本迭代（如用户量等非功能因素、本地化、国际化等）  2.2.1.2 能够识别潜在风险，并进行管理（如信息泄露风险、法律风险、地缘政治等） | 2.2.1.1 产品路线图，如影响地图、用户故事地图、产品版本规划、MVP等  2.2.1.2 风险管理，风险识别与分级，风险应对方案 |
| 2.2.2 产品架构设计 | 2.2.2.1 能够拆分使用场景，并绘制流程图；  2.2.2.2 能够拆分设计产品功能模块，划分模块边界，设计松耦合的架构  2.2.2.3 能够通过数据漏斗视角指导运营策略，优化产品设计； | 2.2.2.1 使用场景及流程，使用场景分析，流程图等  2.2.2.2 功能模块，如模块的定义、模块边界及拓展性、模块关联关系，是否是低耦合等  2.2.2.3 数据漏斗模型 |
| 2.2.3 设计产品界面与用户体验 | 2.2.3.1 能够结合产品定位，整体设计产品的交互体验以及交互的指导原则  2.2.3.2 能够基于运营以及销售目标，设定全局业务的交互设计，策略，形象等 | 2.2.3.1设计产品原型知识，如以用户为导向的设计、面向场景设计、扩展性设计、原型图设计工具等；  2.2.3.2交互效果知识，如整体产品的交互体验设计、交互设计的指导原则等；常用交互设计理论，如米勒定律、席克定律、费茨定律、泰勒斯定律、奥卡姆剃刀定律、接近原则、防错原则等；  2.2.3.3 用户体验设计，如什么是用户体验设计、用户体验设计流程、用户体验设计标准等；用户体验设计框架：战略层、范围层、结构层、框架层、视觉层等； |
| 3.开发与交付 | 3.1协同开发应用 | 3.1.1 设计解决方案 | 3.1.1.1 能够根据产品和项目需求，完成对应的解决方案设计  3.1.1.2 能够持续跟踪项目运作情况，根据市场和团队反馈，评估和引入对应的技术、实践和工具  3.1.1.3 能够对业界最新的技术趋势持续跟进和学习，评估这些新技术的价值并完成技术引入的规划设计 | 3.1.1.1 团队组织、技术选型、基础设施需求、用户需求、业务需求、项目预算、项目时间表、安全和合规性需求等；  3.1.1.2 云原生、微服务、架构解耦、康威定律、容器化、无服务器、可观测性、可测试性、安全、发布策略等；  3.1.1.3 注册、发现、熔断、降级、弹性、LB等；  3.1.1.4 传统架构、分布式架构； |
| 3.1.2 持续改进开发流程 | 3.1.2.1 能够根据组织协作流程设计，指导和执行相应的开发流程  3.1.2.2 能够按照敏捷开发思路进行技术模块设计和工作拆分  3.1.2.3 能够识别个人和团队日常开发过程中的问题，给出建议，制定方案和落地执行改进方案 | 3.1.2.1 各类XP实践、BDD，ATDD等方法  3.1.2.2 模块拆分原则匹配用户故事场景  3.1.2.3 项目风险管理、问题识别、分析与解决方案等； |
| 3.1.3 管理和优化源代码流程及工具 | 3.1.3.1 能够有效的识别，评估和引入适合当前团队现状的源代码管理工具和管理流程  3.1.3.2 掌握业界常见和流行的各类源代码管理工具的主体功能，并可以根据不同工具的特性设计适合当前团队使用的策略  3.1.3.3 掌握不同源代码管理工具和流程对日常编码开发过程的影响，能够根据不同的团队协作模式、开发技术栈、外部依赖以及上下游环节的影响设计、熟知和实施源代码管理与开发调试环境的对接模式，能够根据团队当前的协作模式设计、指导和实施源代码管理策略 | 3.1.3.1 分支管理知识，源码管理流程；分支策略，代码审查流程设计方法、代码规范制定方法  3.1.3.2 源代码管理工具，分支管理策略；  3.1.3.3 团队协作模式、开发技术栈、外部依赖以及上下游环节的影响设计，源代码管理与开发调试环境的对接模式，测试生命周期管理；上下游协作、项目管理、任务管理、流水线、测试与安全、运维与监控等基本知识； |
| 3.2 应用持续集成 | 3.2.1 搭建与实施流水线 | 3.2.1.1 能够基于业务现状、分支模型设计统一的流水线模型并指导流水线模板配置  3.2.1.2 能够基于流行的工具设计流水线监控方案并指导方案实施  3.2.1.3 能够基于技术架构设计三方依赖包管理策略、构建方案并指导方案实施  3.2.1.4 能够基于流行的工具设计流水线日志方案并指导方案实施  3.3.1.5 能够设计流水线集成多道质量门禁并指导实施推广 | 3.2.1.1 流水线模型设计制作方法  3.2.1.2 流水线采集与监控设计方法  3.2.1.3 编绎原理、构建脚本编程基础知识、自动化构建基础知识、依赖包管理工具使用方法  3.2.1.4 日志中心的设计方法  3.2.1.5 流水线集成质量门禁原理、设计方法 |
| 3.2.2 设计与集成质量门禁 | 3.2.2.1 能够设计质量门禁方案并指导实施  3.2.2.2 能够指导团队将质量门禁与研发管理流程融合集成，便于团队高效便利使用 | 3.2.2.1 质量门禁原理、设计方法、质量门禁指标定义  3.2.2.2 质量门禁与研发流程融合 |
| 3.2.3 集成与应用代码质量保证 | 3.2.3.1 能够根据团队技术栈、分支模型、编程语言设计代码质量保证方案  3.2.3.2 能够组织技术专家设计代码规约并指导实施  3.2.3.3 能够组织技术团队开发静态代码扫描插件集成于IDE中，从IDE开始提升代码质量  3.2.3.4 能够根据团队内外技术发展趋势，组织优化代码质量保证方案 | 3.2.3.1 分支模型、代码质量保证工具、研发管理流程  3.2.3.2 代码规约设计方法  3.2.3.3 扫描插件开发、扫描方案设计  3.2.3.4 技术发展趋势，代码质量保证方案 |
| 3.2.4 设计与应用单元测试 | 3.2.4.1 能够根据业务现状、测试技术、测试能力规划单元测试方案  3.2.4.2 能够指导单元测试环境、平台搭建并集成于质量门禁中 | 3.2.4.1 测试金字塔、单元测试技术、单元测试流行工具、单元测试平台设计方法  3.2.4.2 单元测试插件集成方法 |
| 3.3 应用持续部署 | 3.3.1 装配与管理环境 | 3.3.1.1 能搭建和维护具备脚本和API操作能力的虚拟化或私有云平台  3.3.1.2 能搭建和维护容器集群、流量治理与服务观测等相关组件 | 3.3.1.1 虚拟化平台或私有云软件的搭建知识  3.3.1.2 容器集群、流量治理和服务观测组件的装配和维护知识 |
| 3.3.2 部署与更新版本 | 3.3.2.1 能设计应用在服务器或容器集群上的部署架构  3.3.2.2 能设计应用或子模块版本变更的准备、执行、验证和回退步骤  3.3.2.3 能评估应用或子模块版本变更的风险并提出对应处置预案 | 3.3.2.1 虚拟化平台、云系统、容器集群的原理知识  3.3.2.2 应用或子模块版本变更步骤  3.3.2.3 应用或子模块版本变更处置预案 |
| 3.4 应用按需发布 | 3.4.1 设计生产发布策略 | 3.4.1.1能够根据业务场景和环境进行生产发布流水线设计  3.4.1.2能够根据不同的发布策略进行发布脚本模板设计  3.4.1.3能够根据规划制定生产发布流程指引和发布规范 | 3.4.1.1 业务场景与环境知识，发布流水线知识  3.4.1.2 发布策略，发布脚本模板只是  3.4.1.3 熟悉应用安全、性能分析常用方法与工具 |
| 3.4.2 持续发布生产 | 3.4.2.1 能根据发布应用的特点和业务场景，规划流量控制及应用开关控制工具  3.4.2.2 根据业务特点和系统状态，规划生产环境按需发布的流程和工具体系 | 3.4.2.1 熟悉持续发布工具链的规划和设计  3.4.2.2 生产环境流程与工具体系 |
| 3.4.3 设计交付工具链及流程 | 3.4.3.1能从业务需求中识别独立的交付功能范围，并提炼功能及子模块间的业务与技术依赖  3.4.3.2能够根据规划，实现从需求、开发、测试到投产的端到端的流程设计能力  3.4.3.3能够按需进行发布工具体系规划，可系统性地指导工具体系的建设方向 | 3.4.3.1 熟悉业务规划，具有产品的规划能力  3.4.3.2 掌握交付全流程的总体架构  3.4.3.3 按需发布工具体系及特性 |
| 3.4.4 预发布与运营拟真 | 3.4.4.1 能根据功能特点和用户情况制定发布计划  3.4.4.2 能为应用设计各场景高可用要求及降级处置预案 | 3.4.4.1 业务价值分析知识  3.4.4.2 系统可靠性设计知识 |
| 3.5 开发环境管理 | 3.5.1 优化开发调试环境 | 3.5.1.1 能够设计，指导和搭建高效的开发调试环境，包括开发人员本地开发环境和团队共享环境  3.5.1.2 能够有效的识别，评估和引入对开发调试有帮助的各类工具和实践，通过管理和技术手段解决开发人员/团队的日常技术问题 | 3.5.1.1 开发调测环境知识  3.5.1.2 各类源代码管理、开发技术栈以及相关工具链的特点 |
| 3.5.2 管理开发环境安全 | 3.5.2.1 能够在设计开发调试过程中充分考虑安全性要求并提出对应管理策略  3.5.2.2 能够在需求分析阶段提出安全性要求 | 3.5.2.1 安全设计：包括减小攻击面、深度防御、最小权限原则、服务器安全配置  3.5.2.2 威胁建模：概述、设计意义、基于威胁建模的编码约束 |
| 4.测试与安全 | 4.1 测试组织管理 | 4.1.1 采纳测试方法论 | 4.1.1.1 能够根据应用测试需求，沟通协调测试流程与整体开发流程  4.1.1.2 能够根据技术发展方向，调研行业成熟与先进方法，制定或改进测试流程体系  4.1.1.3 能够根据现有的国际和国内测试标准，明确测试方针，制定或改进测试制度 | 4.1.1.1 测试标准知识，测试计需求，测试目标，测试计划；  4.1.1.2 需求流程体系，技术发展趋势，行业成熟与先进方法；  4.1.1.3 流程优化知识； |
| 4.1.2 管理测试团队 | 4.1.2.1能够结合技术发展方向，调研系统测试与安全防护先进技术，组建团队并进行测试团队建设  4.1.2.2 能够根据企业以及团队实际情况，制定并改进团队激励与追责机制，明确成员工作规范 | 4.1.2.1 测试技术趋势知识，测试与安全防护知识  4.1.2.2 团队制度建设知识，测试工作规范制定知识 |
| 4.1.3 采纳测试技术并管理平台工具 | 4.1.3.1 能够根据团队与应用测试需求，选择技术路线，整合搭建测试平台  4.1.3.2 能够跨团队部门协作，系统性搭建或改进组织级的测试平台与工具 | 4.1.3.1 平台搭建知识  4.1.3.2 跨部门协作管理知识 |
| 4.2 测试生命周期管理 | 4.2.1 制定测试策略与计划 | 4.2.1.1 能够组织测试风险分析  4.2.1.2 能够制定测试方针和测试目标 | 4.2.1.1 测试风险管理知识  4.2.1.2 测试方针和目标知识  4.2.1.3 测试策略知识  4.2.1.4 测试方案或测试计划知识 |
| 4.2.2 管理测试需求 | 4.2.2.1 能够组织测试需求的评审  4.2.2.2 能够评估系统需求以确定有效测试范围  4.2.2.3 能够分析测试需求变更的必要性和合理性，确定是否实施测试需求变更 | 4.2.2.1 测试评审技能知识  4.2.2.2 测试分析技能知识  4.2.2.3 需求管理知识  4.2.2.4 需求变更管理知识 |
| 4.2.3 测试用例 | 4.2.3.1 能够掌握测试用例的设计方法，掌握不同类型的测试用例的区别（如功能/性能/单元等多种不同类型的测试用例）  4.2.3.2 能够根据测试目标和策略,确定测试用例的优先级 | 4.2.3.1 测试用例设计方法（等价类、边界值、因果图等）  4.2.3.2 测试优先级知识  4.2.3.3 测试评审知识 |
| 4.2.4 执行测试并报告 | 4.2.4.1 能够组织测试团队成员之间、测试团队和其它项目干系人之间的沟通  4.2.4.2 能够组织测试报告评审和审批  4.2.4.3 能够制定测试过程的暂停和退出规则 | 4.2.4.1执行测试用例技能知识（静态测试、动态测试）  4.2.4.2 测试暂停和退出技能知识  4.2.4.3 测试报告的相关技能知识  4.2.4.4 测试过程管理  4.2.4.5 测试进度与跟踪 |
| 4.2.5 管理与预防缺陷 | 4.2.5.1 能够对测试问题收敛趋势进行分析  4.2.5.2 能够基于测试度量指标制定组织基线，并对产品质量实行质量控制 | 4.2.5.1 缺陷预测技能，如Gompertz模型  4.2.5.2 质量控制技能，如蒙特卡洛分析 |
| 4.3 测试关键域管理 | 4.3.1 执行测试估算 | 4.3.1.1 能够规划完整的测试估算计划，并控制估算误差  4.3.1.2 能够设计并建立组织级的测试估算指导和规则 | 4.3.1.1 常用的测试估算方法  4.3.1.2 制定测试估算计划的能力 |
| 4.3.2 制定测试度量 | 4.3.2.1 能够规划和设计组织级的测试度量指标体系  4.3.2.2 能够根据项目级度量实施结果优化组织级度量指标体系 | 4.3.2.1 缺陷密度、漏测率、测试覆盖率等测试度量指标知识  4.3.2.2 根源分析方法、鱼骨图分析法、帕累托分析法、5why分析法等 |
| 4.3.3 使用测试技术 | 4.3.3.1 能够运用白盒测试、黑盒测试技术完成复杂系统测试规划  4.3.3.2 能够运用功能测试，性能测试，安全测试完成复杂系统测试规划  4.3.3.3 能够运用手动测试和自动化测试完成复杂系统测试规划 | 4.3.3.1 白盒测试中语句覆盖、条件覆盖、路径覆盖等测试基本方法；渗透测试、压力测试等黑盒测试的基本方法及相关工具  4.3.3.2 功能测试，性能测试，安全测试知识及相关工具  4.3.3.3手动测试和自动化测试知识及相关工具 |
| 4.3.4 执行测试阶段 | 4.3.4.1 能够完成单元测试用例设计及测试执行  4.3.4.2 能够完成集成测试用例设计及测试执行 | 4.3.4.1 单元测试知识  4.3.4.1 集成测试知识，自顶向下集成、自底向上集成、API测试、契约测试 |
| 4.3.5 执行自动化功能测试 | 4.3.5.1 能够根据不同阶段规划不同的自动化测试策略  4.3.5.2 能够根据不同技术要求选择不同的自动化测试工具  4.3.5.3 能够规划管理自动化测试用例 | 4.3.5.1 架构设计知识、自动化测试策略知识  4.3.5.2 自动化测试工具知识 |
| 4.3.6 执行非功能性测试 | 4.3.6.1 能够完成设计兼容性测试场景及测试用例，并且执行测试  4.3.6.2 能够完成设计可用性测试场景及测试用例，并且执行测试  4.3.6.3 能够完成设计稳定性测试场景及测试用例，并且执行测试 | 4.3.6.1 兼容性测试知识  4.3.6.2 可用性测试知识  4.3.6.3 稳定性测试知识 |
| 4.3.7 执行性能测试 | 4.3.7.1 能够根据业务目标规划与制定性能测试目标  4.3.7.2 能够辅助开发人员进行性能瓶颈定位及调优  4.3.7.3 能够进行全链路压测设计及执行 | 4.3.7.1 性能测试目标知识  4.3.7.2 性能调优知识  4.3.7.3 全链路压测知识、数据染色知识 |
| 4.3.8 管理测试数据 | 4.3.8.1 能够设计测试数据生命周期管理策略  4.3.8.2 能够设计测试数据分析及脱敏策略 | 4.3.8.1 测试数据全生命周期管理（测试用例与需求、任务关联，测试计划与迭代关联，形成测试流程闭环）的知识  4.3.8.2 数据脱敏规则，数据变形，敏感隐私数据可靠保护 |
| 4.3.9 管理测试环境 | 4.3.9.1 能够规划和设计测试环境管理策略  4.3.9.2 能够规划和设计测试环境管理流程 | 4.3.9.1测试环境全生命周期管理的基本知识  4.3.9.2测试虚拟化的基本知识  4.3.9.3云计算中IaaS、PaaS和容器技术快速构建测试环境的知识 |
| 4.3.10 执行敏捷测试 | 4.3.10.1 能够设计或制定敏捷测试流程和规则  4.3.10.2 能够进行测试与持续集成的结合 | 4.3.10.1 敏捷Scrum知识、流程设计知识  4.3.10.2 持续集成知识 |
| 4.4 安全防护与安全测试 | 4.4.1 管理应用安全生命周期 | 4.4.1.1 能够根据应用安全测试需求，沟通协调测试流程与整体开发流程  4.4.1.2 能够根据技术发展方向，调研行业成熟与先进方法，制定或改进安全测试流程体系  4.4.1.3 能够根据现有的国际和国内安全相关标准，明确安全测试方针，制定或改进安全规则制度  4.4.1.4 能够跨部门规划和设计企业级应用安全生命周期流程 | 4.4.1.1 应用安全生命周期管理的基本知识  4.4.1.2 DevSecOps理念与实践  4.4.1.3 应用安全生命周期各阶级的定义  4.4.1.4 应用生命周期各阶级的安全实践  4.4.1.5 应用安全体系的知识：iso27001、csa-star |
| 4.4.2 设计应用安全测试方案 | 4.4.2.1 能够规划和设计企业级应用安全生命周期的安全测试方案设计  4.4.2.2 能够规划和选择企业级应用安全生命周期的安全测试技术 | 4.4.2.1 安全相关模型，如IPDRR、CIA、STRIDE威胁模型等  4.4.2.2 安全工具金字塔  4.4.2.3 应用生命周期各阶级的安全实践  4.4.2.4 安全策略知识，例如最小特权原则、最小泄露原则和多级安全策略等 |
| 4.4.3 执行应用安全测试方案 | 4.4.3.1 能够结合SAST静态安全测试工具组织进行静态安全检查策略设计  4.4.3.2 能够结合DAST动态安全测试工具组织进行动态安全检查策略设计  4.4.3.3 能够结合WAF、IDS/IPS工具组织组织进行安全规则防护策略设计  4.4.3.4 能够组织进行软件组成分析（SCA），设计开源和外来组件的安全治理策略  4.4.3.5 能够结合IAST交互式安全测试工具组织进行交互式安全检查策略设计  4.4.3.6 能够结合RAST运行时应用自保护工具组织进行应用运行时应用自保护策略设计  4.4.3.7 能够结合渗透测试工具组织进行渗透测试设计  4.4.3.8 能够结合模糊测试（fuzz testing）工具组织进行模糊测试设计 | 4.4.3.1 SAST静态安全测试工具与静态安全检查知识  4.4.3.2 DAST动态安全测试工具与动态安全检查知识  4.4.3.3 WAF、IDS/IPS工具与安全规则防护知识  4.4.3.4 软件组成分析（SCA），开源安全治理  4.4.3.5 IAST交互式安全测试工具与交互式安全检查知识  4.4.3.6 RAST运行时应用自保护工具与应用运行时应用自保护知识  4.4.3.7 渗透测试工具与渗透测试知识  4.4.3.8 模糊测试（fuzz testing）工具与模糊测试知识 |
| 4.4.4 执行云安全防护 | 4.4.4.1 能够结合安全流水线为应用安全生命周期提供全面安全防护的规划方案  4.4.4.2 能够结合容器安全为应用安全生命周期提供全面安全防护的规划方案  4.4.4.3 能够运用防火墙、堡垒机、身份管理与访问控制等规划多层云防护体系  4.4.4.4 能够运规划设计数据安全与隐私保护体系 | 4.4.4.1 安全流水线各阶段的知识，如预提交检查、提交时检查、构建时检查、测试时检查、部署时检查等  4.4.4.2 容器与编排安全知识  4.4.4.3 安全加固与云安全的基础知识  4.4.4.4 DDoS 防护、Web应用防火墙（WAF）、SSL证书配置等知识  4.4.4.5 云安全中心（态势感知）、云防火墙、安全堡垒机、身份管理与访问控制等知识  4.4.4.6 数据安全与敏感数据保护，数据库审计、加密服务、密钥管理等知识 |
| 5. 运维与监控 | 5.1 IT服务管理 | 5.1.1 制定IT服务管理体系 | 5.1.1.1 能够主导制定IT服务管理体系及相关标准规范（价值流的梳理与沉淀）。  5.1.1.2 能够主导整体服务绩效KPI的设定与评价。  5.1.1.3 能够主导IT服务管理平台的规划、落地实施。  5.1.1.4 能够主导IT服务管理领域行业标准、最佳实践的应用研究与理念导入； | 5.1.1.1 IT服务管理主要流程，如事件管理、问题管理、变更管理、容量管理、可用性管理、配置管理等知识  5.1.1.2 IT服务管理工具，如ITSM、CMDB、监控平台、自动化平台、云管平台等  5.1.1.3 价值流梳理、IT服务岗位角色设置、职责分工界定、能力矩阵维护、胜任力评估、康威模型、平衡积分卡等知识  5.1.1.4 行业标准，组织变革方法等知识 |
| 5.1.2 管理事件与问题 | 5.1.2.1 能够按照IT服务管理体系，主导事件与问题管理流程的梳理与总体设计。  5.1.2.2 能够主导事件与问题管理流程绩效KPI的设定与评价。  5.1.2.3 能够主导IT服务管理平台事件与问题管理流程的规划设计与落地实施。 | 5.1.2.1 IT服务管理常规流程，如事件管理、问题管理等知识  5.1.2.2 IT服务管理体系知识，绩效管理知识，PDCA  5.1.2.3 IT服务管理工具，如事件管理、问题管理等知识 |
| 5.1.3 管理变更与配置 | 5.1.3.1 能够按照IT服务管理体系，主导变更与配置管理流程的梳理与总体设计。  5.1.3.2 能够主导变更与配置管理流程绩效KPI的设定与评价。  5.1.3.3 能够主导IT服务管理平台变更与配置管理流程与CMDB的规划设计与落地实施。  5.1.3.4 能够引入新型技术手段，实现变更、配置管理的自动化、智能化，满足业务持续交付要求。 | 5.1.3.1 IT服务管理流程，如变更管理流程、配置与资产管理流程、发布管理流程等知识  5.1.3.2 IT服务管理体系知识，绩效管理知识，PDCA  5.1.3.3 IT服务管理工具，如变更与配置管理流程、CMDB等知识  5.1.3.4 IT服务管理流程设计、自动化、监控与自动发现等知识 |
| 5.1.4 管理容量与成本 | 5.1.4.1 能够按照IT服务管理体系，主导容量与成本管理流程的梳理与总体设计。  5.1.4.2 能够主导容量与成本管理流程绩效KPI的设定与评价。  5.1.4.3 能够主导IT服务管理平台容量与成本管理流程的规划设计与落地实施。  5.1.4.4 能够引入技术手段，推动容量管理的自动化、智能化，支撑运营与业务持续交付。 | 5.1.4.1 IT服务管理流程设计知识  5.1.4.2 IT服务管理体系知识，绩效管理知识，PDCA  5.1.4.3 IT服务管理工具，如监控管理、容量管理、可用性管理等知识  5.1.3.4 自动化、监控与智能化等知识 |
| 5.2 应用运行管理 | 5.2.1 管理应用运行方案 | 5.2.1.1 能够根据应用架构设计，组织制定应用运行方案，进行应用运行方案的长期规划  5.2.1.2 能够根据应用的迭代演进，适时引入新技术，领导制定应用的长期运行方案  5.2.1.3 能综合应用架构、业务规划和行业特性等多种因素，参考相关的行业运行实践，组织并指导制定应用的容量规划与成本方案 | 5.2.1.1 基础设施、中间件、数据库知识  5.2.1.2 云原生知识、微服务知识、容器知识  5.2.1.3 容量估算知识、成本估算知识、TPMC算法 |
| 5.2.2 管理基础设施平台 | 5.2.2.1 能够根据应用运行方案，组织规划基础设施平台整体架构，确保具备足够的扩展性与容错性，满足应用的持续增长与不间断服务  5.2.2.2 在容量规划与预算编制上具备前瞻性，有针对性地追踪新技术与行业趋势，洞察应用架构的技术迭代与演进，确保在基础设施层面能及时满足应用对新技术与新基础设施平台的需求 确保基础设施平台具备前瞻性，及时满足技术与业务发展需求 | 5.2.2.1 集群知识、基础设施架构知识、高可用知识  5.2.2.2 容量规划知识、预算知识、容器平台知识、编排平台知识 |
| 5.2.3 管理应用运行环境 | 5.2.3.1 能够根据应用架构设计应用运行方案，规划应用多地多中心运行及灾备的技术方案；  能够根据应用架构的演进，规划新技术的引进，制定基于新技术的运行环境管理方案；  5.2.3.2 能够组织并指导团队完成应用资源与运行环境的端到端部署、管理与自动化交付； | 5.2.3.1 分布式架构、多地多中心部署、灾备知识  5.2.3.2 流水线知识、项目管理知识、质量管理知识 |
| 5.3 设计管理应用监控 | 5.3.1设计与建设监控系统 | 5.3.1.1 能够根据复杂场景在大访问量的场景组织设计监控架构以及技术选型。  5.3.1.2 能够组织设计覆盖全部监控要素的全域监控体系，制定监控标准  5.3.1.3 能够组织设计常用的可视化图表类型以及应用场景。  5.3.1.4 能够组织设计监控告警规则，确保较高的告警有效率  5.3.1.5 能够结合发布策略和场景，组织设计应急处理的流程和自动化脚本  5.3.1.6 能够对常见的中间件、数据库，网络的监控数据有一定的分析和处理能力 | 5.3.1.1 监控实现方法  5.3.1.2 监控工具架构  5.3.1.3 可视化图表  5.3.1.4 监控告警规则  5.3.1.5 应急管理、故障处理知识  5.3.1.6 常用中间件、数据库的运维知识 |
| 5.3.2 管理监控指标 | 5.3.2.1 能够组织设计指标监控工具，充分考虑指标数据的采集、存储、计算和高可用等。  5.3.2.2 能够组织设计告警规则，能够根据不同场景和情况设定合理的阈值  5.3.2.3 能够组织设计监控仪表盘，设计复杂的联合指标，与CMDB集成 | 5.3.2.1 指标监控工具，指标存储方案  5.3.2.2 告警规则，告警阈值  5.3.2.3 指标采集工具部署，监控仪表盘，可视化仪表盘 |
| 5.3.3 管理收集日志 | 5.3.3.1 能够组织设计复杂非结构化的日志监控工具，覆盖收集、整理、存储等。  5.3.3.2 能够组织干系人采集和管理各类应用和系统日志  5.3.3.3 能够组织设计日志数据的仪表盘 | 5.3.3.1 日志监控工具知识  5.3.3.2 日志监控工具架构  5.3.3.3 日志数据仪表盘配置 |
| 5.3.4 规划调用链监控 | 5.3.4.1 能够组织设计复杂调用链监控工具，覆盖收集、整理、存储等  5.3.4.2 能够组织设计调用链监控规则  5.3.4.3 能够组织设计复杂的调用链监控仪表盘 | 5.3.4.1 调用链监控工具知识  5.3.4.2 调用链监控规则设计  5.3.4.3 调用链监控仪表盘配置  5.3.4.4 OpenTelemetry技术原理  5.3.4.5 调用链数据存储方案 |
| 5.3.5 规划智能监控 | 5.3.5.1 能够利用智能运维算法实现多指标的异常检测  5.3.5.2 能够通过智能运维算法实现基于空间维度的告警压缩  5.3.5.3 能够使用可观测性工具进行应用性能的根因分析 | 5.3.5.1 多指标异常检测算法  5.3.5.2 智能运维算法，告警压缩  5.3.5.3 根因分析关系算法 |
| 5.4 业务连续性管理 | 5.4.1 确定服务质量目标 | 5.4.1.1 能根据应用系统的架构，使用不同的服务风险度量方案。  5.4.1.2 跟根据应用系统的架构，设置对应的SLO服务级别目标。 | 5.4.1.1 SRE基础知识  5.4.1.2 SLO的定义和SLI的拆分知识  5.4.1.3 不同指标采集方法 |
| 5.4.2 响应与回顾事故 | 5.4.2.1 能够进行应用系统应急预案的制定  5.4.2.2 能够培训工程师加入On-Call | 5.4.2.1 故障排查相关技能。  5.4.2.2 On-Call的工作方式和技能。  5.4.2.3 故障回顾技术，如故障复盘会议。  5.4.2.4 应用系统应急预案知识。 |
| 5.4.3 进行灾备管理 | 5.4.3.1 能够根据灾备管理标准，制定不同层级的灾备方案（包括基础设施、中间件、数据库、应用系统等。）  5.4.3.2 能够设计不同等级的灾难恢复能力等级  5.4.3.3 能够设计定期灾备和切换演练策略 | 5.4.3.1 灾备管理标准，灾备方案  5.4.3.2 灾难恢复能力等级知识  5.4.3.3 定期灾备和切换演练策略 |
| 5.4.4 应用混沌工程 | 5.4.4.1 能够根据应用架构规划和设计混沌工程实验方案。  5.4.4.2 能够在公有云模式下规划和设计混沌工程方案。  5.4.4.3 能够在容器化模式下规划和设计混沌工程方案。 | 5.4.4.1 混沌工程知识  5.4.4.2 混沌工程执行  5.4.4.3 公有云和容器化环境下的混沌工程实践 |

# 4.权重表

## 4.1理论知识要求权重表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 专业技术等级 | | 初级（%） | 中级（%） | 高级（%） |
| 基本要求 | 职业道德 | 5 | 5 | 5 |
| 基础知识 | 10 | 10 | 10 |
| 相关知识要求 | 组织与协作 | 10 | 10 | 15 |
| 规划与运营 | 5 | 10 | 15 |
| 开发与交付 | 20 | 25 | 20 |
| 测试与安全 | 30 | 25 | 20 |
| 运维与监控 | 20 | 15 | 15 |
| 合计 | | 100 | 100 | 100 |

## 4.2实践能力要求权重表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 专业技术等级 | | 初级（%） | 中级（%） | 高级（%） |
| 相关知识要求 | 组织与协作 | - | 10 | 15 |
| 规划与运营 | - | 10 | 15 |
| 开发与交付 | - | 30 | 25 |
| 测试与安全 | - | 30 | 30 |
| 运维与监控 | - | 20 | 15 |
| 合计 | | - | 100 | 100 |

# 5.附录

## 5.1参考文献

[1]工业和信息化部.《系统与软件工程 开发运维一体化 能力成熟度模型》（工信部信标委 20194204-T-469号）

[2]工业和信息化部.《系统与软件工程 功能规模测量方法》等（工信部信标委 20194201-T469号）

[3]中国国家标准化管理委员会.《软件工程 软件开发成本度量规范》（工信部信标委 GB/T 36964）

[4]工业和信息化部.《软件研发成本度量规范》（工信部信标委 SJ/T 11463）

[5]工业和信息化部.《软件工程 功能规模测量NESMA方法》（工信部信标委SJ/T 11619）

## 5.2中英文术语对照表

|  |  |  |
| --- | --- | --- |
| 序号 | 英文 | 中文 |
| 1 | 4P营销理论 | 产品 ( Product )、价格( Price )、推广( Promotion)、渠道（Place ） |
| 2 | 5W2H | 七问分析法，分别指的是Who、Where、When、Why、What、How以及How much |
| 3 | AB测试 | 一种测试方法，用于验证用户体验、市场推广等是否正确 |
| 4 | Alpha测试 | Alpha测试是由一个用户在开发环境下进行的测试 |
| 5 | API, Application Programming Interface | 应用程序编程接口 |
| 6 | ATDD | 验收测试驱动开发 |
| 7 | ASO | App store Optimization 应用商店优化 |
| 8 | BDD | 业务驱动开发 |
| 9 | Beta测试 | 一种验收测试 |
| 10 | CALMR/S | 文化，自动化，精益，度量，恢复/分享 |
| 11 | CIA | 信息安全三要素(CIA)：保密性(Confidentiality)、完整性(Integrity)、可用性(Availability) |
| 12 | CMDB | 配置管理数据库 |
| 13 | CoP | Community of Practice，实践社区 |
| 14 | DAST | 动态安全测试工具 |
| 15 | DevOps | 开发运维一体化 |
| 16 | DevSecOps | 开发安全运维一体化 |
| 17 | fuzz testing | 模糊测试 |
| 18 | GameDay | 比赛日，一种DevOps实践 |
| 19 | Gompertz模型 | 种群增长模型 |
| 20 | IaaS | IaaS是（Infrastructure as a Service）的缩写，是指基础设施即服务，是云计算三种服务模式之一 |
| 21 | IAST | 交互式安全测试工具 |
| 22 | IDE | 集成开发环境 |
| 23 | IDS | 入侵检测系统，英文“Intrusion Detection Systems”的缩写 |
| 24 | IPDRR | IPDRR能力框架模型包括风险识别(Identify)、安全防御(Protect)、安全检测(Detect)、安全响应(Response)和安全恢复(Recovery)五大能力 |
| 25 | IPS | 互联网协议群(Internet Protocol Suite )的简写 |
| 26 | IT | 信息技术（Information Technology） |
| 27 | ITIL | 信息技术基础架构库 |
| 28 | ITSM | IT服务管理 |
| 29 | Kanban方法 | 一种敏捷开发方法 |
| 30 | Kano模型 | 狩野纪昭(Noriaki Kano)发明的对用户需求分类和优先排序的有用工具 |
| 31 | LB | Load Balance，负载均衡 |
| 32 | LeSS | 一种规模化敏捷方法 |
| 33 | MoSoCoW | 莫斯科(MoSoCoW)优先级评估法：必须有(MUST)，应该有(SHOULD)，可以有(COULD)，不会有(WONT’T) |
| 34 | MVP | 最小可行产品 |
| 35 | NAT | NAT(Network Address Translation)，是指网络地址转换 |
| 36 | OKR | OKR(Objectives and Key Results)即目标与关键成果法 |
| 37 | On-Call | 值班 |
| 38 | OpenAPI, Open Application Programming Interface | 开放应用程序编程接口 |
| 39 | OpenTelemetry | OpenTelemetry Protocol，简称OTLP，云原生可观测性的标准协议 |
| 40 | PaaS | PaaS是（Platform as a Service）的缩写，是指平台即服务，是云计算三种服务模式之一 |
| 41 | PDCA | “Plan, Do, Check, Act” 的缩写，意思是“计划、执行、检查和行动” |
| 42 | RAST | 运行时应用自保护工具 |
| 43 | RPO | Recovery Point Objective，数据恢复点目标 |
| 44 | RTO | Recovery Time Objective，恢复时间目标 |
| 45 | SAFe | 一种规模化敏捷方法 |
| 46 | SAST | 静态安全测试工具 |
| 47 | SCA | 软件组成分析 |
| 48 | Scrum | 一种敏捷开发方法 |
| 49 | Scrum@Scale | 一种规模化敏捷方法 |
| 50 | Scrum of Scrums | 一种规模化敏捷方法 |
| 51 | SEM | Search Engine Marketing搜索引擎营销 |
| 52 | SEO | Search Engine Optimizaion 搜索引擎优化 |
| 53 | SLI | Service Level Indicator，服务等级指标 |
| 54 | SLO | Sevice Level Objective，服务等级目标 |
| 55 | SMART原则 | 即Specific（明确性）、Measurable（可衡量性）、Attainable（可实现性）、Relevant（相关性）、Time-bound（时限性  ）等英文单词的首字母组合 |
| 56 | SRE | Site Reliability Engineer ，站点可靠性工程师 |
| 57 | STAR原则 | 即Situation（情景）、Task（任务）、Action（行动）和Result（结果）等英文单词的首字母组合 |
| 58 | STRIDE | STRIDE，代表六种安全威胁：身份假冒（Spoofing）、篡改（Tampering）、抵赖（Repudiation）、信息泄露（Information Disclosure）、拒绝服务（Denial of Service）、特权提升（Elevation of Privilege） |
| 59 | SWOT | 英文Strengths、Weaknesses、Opportunities和Threats的缩写,分别代表优势、劣势、机会和威胁 |
| 60 | TOC | 约束理论 |
| 61 | TPMC | transactions-per-minute-C(tpmC), 也就是每分钟执行的transaction(数量) |
| 62 | UI | User Interface，用户界面 |
| 63 | VPC | 虚拟私有云(Virtual Private Cloud)的缩写 |
| 64 | WAF | Web应用防护系统（Web Application Firewall） |
| 65 | Webhook | WebHook是网页开发中的一种通过自定义回调函数来增加或更改网页表现的方法 |
| 66 | WSJF | 加权最短作业优先 |
| 67 | XP | 极限编程 |
|  |  |  |