

ICS

点击此处添加中国标准文献分类号

备案号：

中华人民共和国

行业标准

XX/T XXXXX—XXXX

研发运营一体化（DevOps）能力成熟度模型

第4部分：技术运营管理

The capability maturity model of DevOps

Part 4: Technology Operation management

点击此处添加与国际标准一致性程度的标识

（征求意见稿）

— XX — XX 发布

XXXX — XX —)

发 布

目 次

前言.....II

研发运营一体化（DevOps）能力成熟度模型 第 4 部分：技术运营管理1

 1 范围.....1

 2 规范性引用文件1

 3 术语.....1

 4 缩略语1

 5 技术运营2

 6 监控管理3

 7 事件管理6

 8 变更管理8

 9 容量和性能管理11

 10 成本管理14

 11 连续性和可用性服务17

 12 用户体验管理21

 13 运营一体化平台24

附 录 A （规范性附录） 容量与性能管理指标举例28

附 录 B （规范性附录） 容量与性能管理指标举例29

A31

A31

参考文献.....32

前 言

研发运营一体化是指在IT软件及相关服务的研发及交付过程中，将应用的需求、开发、测试、部署和运营统一起来，基于整个组织的协作和应用架构的优化，实现敏捷开发、持续交付和应用运营的无缝集成。帮助企业提升IT效能，在保证稳定的同时，快速交付高质量的软件及服务，灵活应对快速变化的业务需求和市场环境。

本标准是“研发运营一体化（DevOps）能力成熟度模型”系列标准的第 4 部分 技术运营管理，该系列标准的结构和名称如下：

第1部分：总体架构

第2部分：敏捷开发管理

第3部分：持续交付

第4部分：技术运营

第5部分：应用架构

第6部分：安全风险管

第7部分：组织结构

本标准按照GB/T 1.1-2009给出的规则起草。

本标准由中国通信标准化协会提出并归口。

本标准起草单位：待完善

本标准主要起草人：待完善

研发运营一体化（DevOps）能力成熟度模型 第4部分：技术运营管理

1 范围

本标准规定了研发运营一体化（DevOps）能力成熟度模型下技术运营管理的能力成熟度要求和评价方法。

本标准适用于具备IT软件研发交付运营能力的组织实施IT软件开发和服务过程的能力进行评价和指导；可供其他相关行业或组织进行参考；也可作为第三方权威评估机构衡量软件开发交付成熟的标准依据。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

- [1] GB/T 32400-2015 信息技术 云计算 概览与词汇
- [2] GB/T 32399-2016 信息技术 云计算 参考架构
- [3] YD/2441-2013 互联网数据中心技术及分级分类标准
- [4] GB/T 33136-2016 信息技术服务数据中心服务能力成熟度模型

3 术语

下列术语和定义适用于本文件。

3.1 配置项 configuration item

即纳入配置管理范畴的工作成果，是保存系统和项目的相关配置。

3.2 制品 artifact

即构建过程的输出物，包括软件包，测试报告，应用配置文件等。

3.3 代码复杂度 code complexity

主要度量指标为圈复杂度，即代码中线性独立路径的数量。

3.4 部署流水线 deployment pipeline

指软件从版本控制库到用户手中这一过程的自动化表现形式。

4 缩略语

下列缩略语适用于本文件。

APaaS	Application Platform as a Service	应用部署和运行平台
API	Application Programming Interface	应用程序编程接口
BA	Business Analyst	业务分析师
BIA	Business Impact Analysis	业务影响分析
CI	Continuous Integration	持续集成
CD	Continuous Delivery	持续交付
CDN	Content Delivery Network	内容分发网络
ETL	Extract Transform Load	数据仓库技术
ESB	Enterprise Service Bus	企业服务总线
SNMP	Simple Network Management Protocol	简单网络管理协议
SDK	Software Development Kit	软件开发工具包
TPS	TransactionsPerSecond	事务数/秒
IPaaS	integration platform as a service	集成平台
MTTR	Mean time to recovery	平均修复时间
KPI	Key Performance Indicator	关键绩效指标
RPO	Recovery Point Objective	恢复点目标
RTO	Recovery Time Objective	恢复时间目标
UI	User Interface	用户界面
SMART	Specific Measurable Attainable Relevant Time	具体的、可度量度、可实现的、相关性和时效性原则

5 技术运营

技术运营管理过程是技术运营能力建设的一个过程，它以业务为中心，交付稳定、安全、高效的技术运营服务，构建业界领先的技术运营能力，支撑企业的持续发展和战略成功。技术运营不仅关注“稳定”、“安全”、“可靠”，更要关注“体验”、“效率”、“效益”。

技术运营管理过程分为：监控管理、事件管理、变更管理、容量与性能管理、成本管理、连续性管理、用户体验管理、运营一体化平台等，如表1所示。

表1 持续交付分级技术要求

技术运营管理过程							
监控管理	事件管理	变更管理	容量与性能管理	成本管理	连续性与可用性管理	用户体验管理	运营一体化平台
指标采集	事件发现	计划内普通变更	容量指标识别	预算管理	连续性管理	业务认知能力	运营一体化平台整体框架
监控数据处理	事件处理	计划外紧急变更	容量与性能管理活动	成本优化	可用性管理	数据管理能力	IaaS统一管控能力

异常识别	事件回顾		容量架构设计		应急事件管理	体验优化能力	原子平台能力
监控可视化及通知			日常容量管理				PaaS平台能力
			特殊容量管理				运维场景能力
			容量管理等级				

6 监控管理

监控管理是对研发运营过程中的对象进行状态数据采集、数据处理分析和存储、异常识别和通知及对象状态可视化呈现。其成熟度决定了技术运营工作的立体性、及时性和有效性。

监控管理主要包括 4 个部分：指标采集，监控数据处理，异常识别，监控可视化及通知。

6.1 指标采集

指标采集是指通过主动采集或被动收集方式获取监控数据。指标采集的能力指标包括覆盖度、采集方法、准确性、及时性和采集性能等，如表 2 所示。

表2 指标采集

级别	覆盖度	采集方法	准确性	及时性
1	覆盖基础的主机操作系统性能监控。	通过操作系统自身工具采集数据。	依靠操作系统工具采集，准确性保持与操作系统一致。	具有分钟粒度的准实时性。
2	覆盖主机操作系统、业务应用的监控。	编写采集插件，使用定时任务框架调度插件采集数据。	业务应用的监控数据误差小于 5%，其余数据与观察一致。	同上
3	覆盖主机、网络、主机操作系统、业务应用的监控。	采集插件支持操作系统工具、自定义处理程序和 SNMP 等协议采集数据。	同上	同上
4	覆盖主机、网络、主机操作系统、中间件、业务应用的监控。	具有完善主动采集插件和任务调度框架，有匹配业务需求的监控数据上报 agent。	业务应用的监控数据误差小于 1%，其余数据与观察一致。	具备秒级上报的实时性。
5	覆盖主机、网络、主机操作系统、中间件、应用系统、终端访问（如终端）的监控。	具有完善主动采集插件和任务框架，有匹配业务需求的监控数据上报 agent 和 SDK。	业务应用的监控数据无误差，终端的监控数据仅有少量误差，其余数据与观察一致。	同上

6.2 监控数据处理

监控数据处理是指对数据进行过滤、转换、提取、聚合和存储等操作，是监控核心能力。监控数据处理能力包括ETL功能丰富度、处理能力、存储能力和适配能力，如表3所示。

表3 监控数据处理

级别	ETL 功能	处理能力	存储能力	适配能力
1	从采集信息中提取出监控对象、指标或事件用于发送异常告警。	通过简单程序进行少量的数据加工处理。	使用小规模通用关系数据库存储监控数据。	应用于特定监控场景，提供数据的直入直出能力。
2	对采集信息进行过滤后，提取监控对象、指标或事件，按时间粒度统计聚合形成监控指标。	在单机上部署少量程序对少量的数据加工处理。	对监控数据建立关系模型存储在通用关系数据库中。	应用于特定领域的监控场景，提供该场景下的指标查询和异常检测能力。
3	对采集信息进行清洗、翻译、转换后，提取监控对象、指标或事件，按时间粒度统计聚合形成监控指标。	在小型集群上部署数据处理程序对数据加工处理。	抽象监控数据模型，使用存储集群存储监控数据，提供高效数据查询接口。	应用于复杂的领域监控场景，提取和转换计算出该场景下的多个指标，提供查询和异常检测能力。
4	对采集信息进行清洗、翻译、转换后，提取监控对象、指标或事件，按时间粒度和维度统计聚合形成多维度指标。	在中等规模集群上部署分布式服务对数据进行加工处理，集群规模和处理能力可平行扩展。	抽象多维数据模型，使用可扩展的存储集群存储多维时间序列监控数据，提供高效数据查询接口。	应用于复杂领域的精细化监控场景，提取和转换计算出多维度监控指标，提供查询和分析能力。
5	通过自定义配置方式实现多种类型采集信息的清洗、翻译、转换和统计，形成多维度指标。	在通用的分布式流处理集群对多种类型的海量数据进行加工处理。	使用高性能可扩展的多维时间序列数据存储集群存储多个业务的监控数据，提供高效数据查询接口。	应用于复杂领域的精细化监控场景，支持多种业务数据接入和监控，提供查询、分析和关联整合能力。

6.3 异常识别

异常识别是指对提取的监控指标或事件进行检测，识别出异常点用于告警和决策。异常识别主要包括识别方法、收敛策略、有效性和告警时延，如表4所示。

表4 异常识别

级别	识别方法	收敛策略	有效性	告警时延
1	由监控对象直接发	直接发送原始异常事件	能感知监控对象发生异	能实时获知异常事件

	送异常事件		常，但出现持续告警量	
2	通过监控对象的指标是否超过阈值识别异常点	对异常事件按时间、告警对象或接收人维度进行告警合并，减少告警发送量	能获知监控指标超出阈值区间的异常事件，但相比真实的异常情况存在误告警或漏告警情形	能准实时获知指标异常情况，告警延时在3分钟内
3	集中聚合多个待检测指标进行异常识别	除对异常事件按时间、告警对象或接收人维度汇聚收敛外，还按异常持续时间和告警频率收敛	通过综合多个指标识别异常点，提升识别准确率，但漏告警情形仍存在	能准实时获知指标异常情况，告警延时在3分钟内
4	采用指标分级方法对重点指标进行异常识别，实现告警分级	按告警级别分别采用对应的收敛策略，突出重要告警	通过告警分级方式一方面提升重要告警的准确率，另一方面避免重要告警被淹没	能准实时获知指标异常情况，重要告警延时在3分钟内
5	采用智能化方法识别监控对象指标的异常点	采用智能化方法对告警进行关联分析，按告警根源收敛	告警准确率得到有效提升，漏告警率显著减少。重要告警得已凸显。	能准实时获知指标异常情况，重要告警延时在3分钟内

6.4 监控可视化及通知

监控可视化是将监控数据通过图表方式进行呈现，监控通知是将异常事件通过邮件或短信等通信方式告知业务负责人。监控可视化及通知能力包括可视化呈现、通知方式、预处理手段和告警统计等，如表5所示。

表5 监控可视化及通知

级别	可视化呈现	通知方式	预处理手段	告警统计
1	告警、性能事件能够逐条呈现	能将告警、性能事件通过短信或邮件等方式通告	无预处理提示或方法	人工统计告警统计报表
2	对短周期内恢复的异常，能够自动消除告警、性能事件，但存在记录	同上	同上	能够按照告警项、告警级别、时间周期、告警频率维度进行统计和导出
3	告警、性能事件能区分级别、类型，并按需进行呈现	根据设置的级别、类型进行短信或邮件等方式通告	能记录告警处理经验，有自动化脚本收集告警、性能对应的当前环境信息	同上
4	告警、性能事件能进行合并或关联，将重复事件信息消重，将同一根本原因的关联事件进行合并	将重要事件生成工单跟踪。如一段时间未受理，有自动电话拨打提示	有预处理脚本或工具，能针对告警、性能进行处理，并邮件至运维负责人反馈处理结果	能够根据主机、网络、组件、应用等承载关系、告警关联关系进行告警信息统计

5	对于应用系统，能够展示业务链调用情况，展示各个环节异常情况	如受理不及时，能自动通知 B 角人员或进行责任上升通报	有预处理脚本或工具，能针对告警、性能进行处理，处理完毕后能自动消除告警	能够自动生成上述报表，按照设置的周期和发送方式，自动发送至相关干系人员
---	-------------------------------	-----------------------------	-------------------------------------	-------------------------------------

7 事件管理

事件是指计划外的服务中断、服务质量下降或还未影响服务的事态。事件管理流程是技术运营中非常关键的一个能力项，目的是快速响应用户事件，并快速恢复受事件影响的 IT 服务，力求使事件对用户的影响最小化。通过对事件整个生命周期的控制，包括：事件发现、事件分析及处理、事件回顾，实现事件的闭环管理。

随着事件管理的精细化，根据事件性质可以细化为生产事件、业务事件、安全事件和桌面事件，并制定不同的管理策略进行管理，以提高事件管理效率和用户感受。

生产事件是指影响或可能影响生产环境中业务应用、系统环境、网络通信、机器设备、机房设施的正常有效运行，导致或可能导致业务服务中断、交易成功率下降、交易差错或者金额损失等后果的情形。业务事件是指在软件运行中因为程序本身有错误，但是不会导致业务服务中断、交易成功率下降、交易差错或者金额损失等后果的情形。安全事件是指利用计算机信息系统技术脆弱性，使计算机信息系统特有的机密性、真实性、完整性、有效性、不可否认性遭受到不同程度的破坏，或因人为因素及自然灾害，对计算机信息系统或基础设施的正常使用造成不同程度的破坏和威胁的事件。桌面事件是指个人桌面相关故障或服务请求。

7.1 事件发现

事件发现主要指 IT 部门通过各种技术手段主动或被动发现和受理信息系统的异常或者例外事件，如表 6 所示。

表6 事件发现

级别	事件分类分级	工具	用户感知
1	被动受理和处理 IT 故障。	依赖用户报障，无主动事件发现工具。	用户主观感受 IT 系统不稳定，业务系统经常出现各种故障，阻碍业务运行。
2	建立服务台统一受理事件，并初步对事件进行分类，根据影响度和紧急度划分事件优先级，建立闭环管理机制，但仍属于被动响应。	建立基本的工具记录事件，建立专门的报障电话、邮箱。初步建立零散的监控工具或者脚本。	用户主观感受 IT 系统不稳定，业务系统偶尔出现重大事件。
3	先于用户发现事件，并将事件细化为服务请求、生产事件、安全事件和监控告警。 对不同类别和等级的事件设定对应的服务级别。	建立用户自助 IT 服务门户，并建立统一的监控系统。监控告警自动生成事件单，事件解决后自动消除告警。	IT 系统相对较稳定性，事件基本能在约定的时间内恢复。
4	深入分析各类事件的发生趋势	具备对历史事件和监控告警数	IT 系统稳定可靠，能弹性伸

	和原因，有效控制事件发生数量，尤其是重大影响的事件。	据的分析能力，辅助事件趋势分析和数量控制，提前预警重大事件发生征兆。	缩系统容量，以支撑业务各种重大营销活动。
5	实时收集用户感知和反馈，自动匹配事件分类和分级，并根据事件的处理进展中影响度和紧急度的变化，自动调整事件级别。	基于 IT 运维大数据分析和机器学习技术，自动定位事件根本原因和业务影响，基于场景和知识库实现故障自愈。	用户完全感知不到事件发生，而且IT主动引领业务创新。

7.2 事件处理

事件处理指 IT 部门各条线技术人员通过各种技术手段对事件进行分析、诊断、快速解决和恢复业务的过程，包括事件处理机制、事件处理角色与考核两个方面，如表 7 所示。

表7 事件管理

级别	事件处理机制	事件处理角色与考核
1	事件处理无时效性要求，无重大事件处理机制。	未区分不同事件处理角色，主要根据技术条线划分岗位。基本无 KPI 指标
2	统一定义各等级事件服务级别，基本保证达标。有建立重大事件处理子流程。	建立服务台，并设定一、二、三线运维支持团队。建立流程运行 KPI 指标，例如事件总数、解决率、及时解决率。
3	有效保证各等级事件处理时效性符合服务级别要求。重大事件应急处置机制运行顺畅，并与IT连续性管理办法一致。	明确各团队的 KPI 考核指标。KPI 指标偏向于如何提高系统稳定性，例如降低事件发生数量、一线解决率等。
4	各等级事件处理时效服务级别可控。逻辑较简单生产事件可实现自动化处理，例如故障实例自动重启、切换等。	KPI 指标偏向于自动化事件恢复比例。
5	绝大部分生产事件无需人工参与，可自动修复，并且业务完全无感知。	KPI 指标偏向于如何提高用户体验。

7.3 事件回顾

事件回顾指IT部门定期对周期内发生的所有事件进行总结，包括生成相关报表、服务级别达成情况和典型案例经验分享等，旨在持续提高事件管理能效，事件回顾包括事件数量趋势、系统可用性影响和知识管理三方面，如表7所示。

表8 事件回顾

级别	事件数量趋势	系统可用性影响	知识管理
1	随机发生，随着IT信息系统规模扩张而指数上升。	不可预测	存在零乱知识积累行为
2	随机发生，随着IT基础设施规模和复杂度上升而上升，开始对重复事件进行管理。	建立可用性服务级别，但是存在违约情况。	将重大、典型或重复故障生成问题进行处理，并形成知识，但分散在不同的位置存储。

3	事件数量可控，不随着IT信息系统规模扩张而上升，并有效控制重复性事件。	建立可用性服务级别，且能保证达标。	建立专业的知识录入和分享工具，并在整个IT部门高效运转。
4	事件数量稳中有降，主动预防重大影响事件。	通过云平台、容器等高可用技术，保证可用性不受影响。	业务部门和IT部门一起形成知识的分享和使用圈。
5	基于弹性和健壮的IT基础架构，杜绝对业务有影响的生生产事件。	通过资源弹性伸缩和容灾建设，能够无缝支撑业务的增长以及突发的业务活动。	形成的知识库，能录入自动化工具，实现故障自动解决恢复。

8 变更管理

变更就是增加，修改，删除或任何对正常服务有影响的操作。变更管理的目标就是控制所有变更的生命周期，确保在最小化IT服务中断的情况下实现有益的变更。在DevOps环境下，生产环境的变化非常频繁，变更管理面临许多挑战。组织需要了解变更的内容、识别变更的影响范围、识别变更的风险，以便对采取有效的管控措施，既要高效率实施变更，又要最小化业务风险。

任何团队或个人均可提出变更请求，变更管理团队将受理、评估、批准或拒绝这些变更、协调变更的实施并验收变更、持续分析和总结所有的变更活动，以持续改进变更管理活动。

对所有DevOps关键基础运营环境，包括软件、架构、数据库、基础设施等变更，也包括对IT服务和其他配置项的变化、可能会影响用户体验和业务正常运行的变化，都应纳入到变更管理的控制范围内。变更管理分为计划内变更和计划外变更两个部分。

8.1 计划内普通变更

计划内普通变更是指按照技术组织预定义好的计划实施的高频的、常规的、操作规范的、标准的变更。DevOps计划内的变更管理成熟度可以从变更管理流程、变更管理人员、变更管理工具、变更报告、变更指标等5个维度来描述，如表9所示。

表9 计划内普通变更

级别	变更管理流程	变更管理人员	变更管理工具	变更管理报告与通知	紧急变更指标
1	有零散的、非强制执行的变更管理实践。	a) 设立了变更管理岗位。 b) 变更的实施，除了项目组外，运维需要全程值守和参与变更实施	手工实施变更	变更活动仅通知到技术团队内部。	a) 首次变更失败率小于等于5% b) 回退失败率大于等于3%
2	a) 有正规的、端到端打通的、需强制执行变更管理	a) 同上 b) 变更管理组织增加了变更管理委员会。	a) 有变更管理系统，是一套基于工作流的审批系统。 b) 手工实施加	a) 同上 b) 异常情况能够升级到技术团队的管理层。	a) 首次变更失败率小于3% b) 回退失败率小于1%

	流程； b) 有特定的变更窗口 c) 定期召开变更会议，审批变更。		上工具/脚本辅助的变更。	a) 变更管理定期出具变更管理报告。	
3	a) 同上 b) 变更管理流程与发布、部署管理流程、配置管理流程集成 c) 变更管理会议上能对此前变更进行回顾和总结。	a) 同上 b) 变更实施主要由项目组或研发团队实施变更，运维无需干预。 c) 需要运维现场值守。	a) 变更管理系统与发布和部署管理流程、配置管理流程集成。 b) 变更管理系统支持多种灰度模式自动变更。	b) 同上 c) 变更活动会通知到技术组织和受影响的内部业务部门。	a) 首次变更失败率小于1% b) 回退失败率小于0.5%
4	a) 同上 b) 变更管理流程全面支持DevOps其它相关流程。 c) 变更管理流程能够实现全面自动化实施。	a) 同上 b) 运维只需要远程值守。	a) 全面的自动化变更管理系统，一键自动变更。 b) 支持业务无感知的自动化的变更故障自愈和自动回滚。	a) 变更活动能够自动通知到技术组织和企业内外受影响的所有干系人。 b) 不同的技术或业务组织出具不同的变更管理报告。	a) 首次变更失败率小于0.5% b) 回退失败率小于0.1%
5	变更管理流程能够全面实现智能化实施。	实现无人值守变更	a) 同上 b) 人工智能的变更管理平台，具有大数据采集和人工智能分析与自动决策能力	a) 同上 b) 该报告具备多样化的定制内容要求和自动推送能力，满足不同干系人的阅读要求，并有良好的用户体验。	a) 首次变更失败率小于0.1% b) 回退失败率小于0.05%

8.2 计划外紧急变更

计划外变更是指遇到紧急和突发事件，需尽快在生产环境中进行的变更，以满足业务急迫的需要，比如发生了故障或特别紧急的需求，需要紧急变更。

DevOps计划外的紧急变更管理成熟度可以从变更管理流程、变更管理人员、变更管理工具、变更报告、变更指标等5个维度来描述，如表10所示。

表10 计划外紧急变更

级别	变更管理流程	变更管理人员	变更管理工具	变更管理报告与通知	紧急变更指标
1	有零散的、非强制执行的紧急变更管理实践	a) 设立了紧急变更管理岗位。 b) 紧急变更的实施，除了项目组外，运维需要全程值守和参与变更实施。	手工实施紧急变更	紧急变更活动仅通知到技术团队内部。	a) 首次变更失败率小于等于5%。 b) 回退失败率小于等于3%。
2	a) 有正规的、端到端打通的、需强制执行的紧急变更管理流程； b) 能够及时召开紧急变更会议，审批紧急变更。	a) 同上 b) 紧急变更管理组织增加了紧急变更管理委员会。	a) 有变更管理系统，是一套基于工作流的审批系统，支持紧急变更管理流程。 b) 手工实施加上工具/脚本辅助的变更。	a) 同上 b) 异常情况能够升级到技术团队的管理层。 c) 定期出具紧急变更管理报告。	a) 首次变更失败率小于3%。 b) 回退失败率小于1%。
3	a) 同上 b) 紧急变更管理流程、发布与部署管理流程、配置管理流程集成。 c) 例行的变更管理会议上能对此前的紧急变更进行回顾和	a) 同上 b) 紧急变更实施主要由项目组或研发团队实施紧急变更，运维无需干预。 c) 需要运维现场值守。	a) 变更管理系统与发布和部署管理流程、配置管理流程集成。 b) 变更管理系统支持多种灰度模式自动变更。	a) 同上 b) 紧急变更活动会通知到技术组织和受影响的内部业务部门。	a) 首次变更失败率小于1%。 b) 回退失败率小于0.5%。

	总结。				
4	a) 同上 b) 紧急变更管理流程全面支持DevOps 其它相关流程。 c) 紧急变更管理流程能够实现全面自动化实施。	a) 同上 b) 运维只需要远程值守紧急变更。	a) 全面的自动化变更管理系统，一键自动变更。 b) 支持业务无感知的自动化的紧急变更故障自愈和自动回滚。	a) 紧急变更活动能够自动通知到技术组织和企业内外受影响的所有干系人。 b) 能够为不同的技术或业务组织出具不同的变更管理报告。	a) 首次变更失败率小于0.5%。 b) 回退失败率小于0.1%。
5	a) 紧急变更管理流程能够实现全面智能化实施。	a) 实现无人值守的紧急变更	a) 同上 b) 人工智能的变更管理平台，具有大数据采集和人工智能分析与自动决策能力	a) 同上 b) 该报告支持多需求定制内容和自动推送能力，满足不同干系人的阅读要求，并有良好的用户体验	a) 首次变更失败率小于0.1%。 b) 回退失败率小于0.05%。

9 容量和性能管理

容量与性能管理 (Capacity Management) 是对容量/性能进行评估、规划、分析、调整和优化过程，它结合了业务、服务和资源容量需求，以保证对资源的最优利用，满足与用户之间所约定的性能等级要求，在公司业务快速增长以及有大量的个人用户端业务时，则必须进行容量与性能管理。

9.1 容量与性能管理活动

容量管理由一系统活动组成，覆盖的活动如下：

1) 容量指标的识别

容量指标的识别指识别整个业务运行过程中涉及各领域层级的容量指标，领域涵盖业务领域、应用领域、基础架构领域，其中不同应用的业务领域与应用领域容量指标不同，基础架构领域相对稳定，但会因使用产品（开源/供应商）的不同指标有所变化，容量指标识别是容量架构设计、上线前/测试容量与性能管理、日常容量与性能管理、年度容量与性能管理、特殊日期/营销容量与性能管理这五部分活动开展的前提，也是容量与性能管理活动开展的关键。

容量指标识别的方法为，通过分析业务全链路的数据流经的所有节点，逐个分析每个节点涉及的容量指标。同时不仅要关注自身业务应用涉及的指标，也要关注外部系统的性能指标，比如外部连接机构、外部服务提供公司，如定位、地图、支付、征信等。容量与性能管理的指标举例详见附录A和附录B。

2) 容量架构设计

在应用架构设计时进行的容量性能的架构评估与设计，根据业务未来发展需求从应用架构与基础架构解耦、横向伸缩、自动弹性、服务化等角度设计应用架构。

3) 上线前/测试容量与性能管理

进行系统或设备上线前的测试调优和资源分配，上线前容量与性能管理容量主要为容量测试上线前的测试案例中需指定容量测试案例，容量测试案例覆盖单节点极限峰值，一般以 CPU 利用率、带宽到达峰值时的业务值为测试结果。测试案例覆盖主要容量指标（例如 CPU、业务 TPS、时延、业务处理时间、互联网带宽、网络带宽等）通过测试业务量增加时应用节点扩容的方式（纵向/横向、在线/离线、自动/手工）等，确认应用在应对容量扩容时的方案。

4) 日常容量与性能管理

主要内容为日常容量监控和容量分析/预测。

- 1、进行 IT 组件的容量资源监控、预警、告警。
- 2、各 IT 组件的容量指标不断完善，容量监控质量持续提高。
- 3、通过容量数据分析，预测容量发展趋势。
- 4、发现版本发布前后的容量变化。

5) 年度容量与性能管理

容量与性能管理进行年度容量评估和次年容量计划，将评估结果作用于第二年的资源扩充，

年度容量与性能管理的目标是：

- 1、对全年重点 IT 组件的容量使用进行总结。
- 2、分析容量短板，制定次年容量计划，并作为 IT 资源计划制定的依据。

结合容量指标监控数据、业务指标监控数据、节前容量数据对当年的组件容量进行评估分析，并结合次年业务目标数据确立容量短板或资源过盛的情况，提出专业领域容量调整计划。

6) 特殊日期/营销容量与性能管理

特殊日期容量与性能管理主要包括节假日（春节、五一、中秋/国庆、元旦等）、内外部营销活动（双11/12 等）的容量预估、容量压测、营销后数据分析等工作。

特殊日期容量与性能管理的目标是：

- 1、保证特殊日期主要 IT 组件的容量满足特殊日期业务需要。
- 2、持续优化业务量预测模型、业务-容量测算模型。
- 3、持续优化各 IT 组件容量指标体系。
- 4、及时提出扩容需求并进行扩容。

9.2 容量与性能管理

容量与性能管理是通过对容量相关活动与容量指标的管理，保障业务的可持续性运行，是业务发展阶段中必须开展的一项重要工作内容。提供用户优质的产品体验提升产品的技术竞争力。要求团队重视性能的各个阶段，建立体系化的管理包括工具、数据度量、控制流程。

9.2.1 容量管理

表11 容量管理

级别	管理范围与策略	应用架构	工具
1	没有容量与性能管理概念, 缺乏对容量与性能管理的理解, 无相关容量与性能管理活动, 被动受理和处理容量故障。	无容量指标考量, 架构设计中不考虑容量设计。	无容量数据的记录行为。
2	有基本的容量与性能管理活动, 包括容量测试及线上容量监控, 无容量指标概念, 无明确的容量与性能管理制度与要求。	有基本的容量设计考虑, 但是应用架构本身无法进行容量性能扩展。	有监控、测试等工具进行记录, 相对独立。
3	有明确的容量与性能管理制度要求, 以及较完整的覆盖架构设计、测试、日常、年度及特殊日期/营销容量与性能管理阶段与活动。	应用架构可以进行性能扩展, 但一般为纵向扩容, 受限于有限硬件性能, 且扩容需要进行应用变更进行。	有容量与性能管理每个阶段的工具支撑, 数据相关独立。
4	有明确的容量与性能管理制度及容量指标分析识别, 并应用在容量日常管理阶段与活动中。日常熔炼管理活动者能进行趋势与预测性分析。	应用架构可进行不受硬件架构限制的横向扩容, 但无形成与资源/业务指标结合的自动伸缩架构。	能通过每个容量活动阶段的数据统计进行容量数据分析及管理活动的开展, 并支撑趋势与预测性分析。
5	有完整的的数据与工具平台支撑容量活动的正常开展与分析优化。	应用架构可在线根据业务/资源指标进行弹性伸缩。	有完成的容量平台支撑容量相关阶段与活动的开展。 利用机器学习、算法训练等开展容器预测, 并能实现弹性的能力对接。

9.2.2 性能管理

表12 性能管理

级别	管理范围与策略	人员机制	工具
1	性能管理的各项工作没有开展。	没有相关的人员配置。	无相关工具。
2	团队对性能管理有局部的投入。 能够在基础性能指标建立起有效管理控制能力。 对于架构有初步的投入和管控。	根据数据排查对初级问题快速定位和解决。	通过平台, 对基础监控和度量能力进行管理。

3	基于应用服务、架构、用户体验有容量管理和性能度量。	团队具备主动性能管控要求，对于全链路的主要性能项有准入标准和异常发现能力。 建立基本的业务准入机制。	多维度的监控平台，能覆盖到应用级、架构级、用户体验等维度。
4	端到端的性能管理能力，并且能够建立关联分析、趋势分析，从解决问题到提前预防问题。有完善的性能解决方案。	人员具备多维度的专项分析能力。 定期的全链路压测。	立体化监控 端到端监控 专项测试工具、覆盖硬件
5	持续优化、智能化。 有明确的数据分析、训练的能力。	有数据/算法工程师进行智能工作开展。	有数据训练、算法平台。

10 成本管理

如今IT 服务对很多企业都是至关重要的，企业对IT的投入也是越来越大，随着产品用户数量、质量、功能的增加，包括IT服务系统本身的技术复杂性，都会导致IT服务成本的快速上升，在很多企业中IT成本是公司最大的资金开销。

10.1 IT 服务运营成本定义

运营成本包括不限于：机房机位、内外网带宽、交换机、服务器、软件购买、云服务购买等选择不同的资源模式成本管理有很大差异，企业常见的有四种：

IDC托管模式：该模式有第三方提供机房物力空间、机架、供电、制冷等基础设施，企业购置服务器、网络设备、带宽、软件等自建设部署；

托管服务模式：该模式下外包核心基础设施，如电能和网络连通性，企业无需再购买服务器和网络硬件由托管服务提供商提供租赁，并负责管理这些硬件系统和安装操作系统软件。

云模式：和托管服务模型类似，在这个模型中，企业以购买云服务的方式外包了基础设施和硬件，且企业并不是独享硬件资源，而是在需要云服务资源时动态分配的虚拟化资源。

云模式的意义在于，由于采用按需使用付费的原则，它改变了成本模型中CAPEX（Capital Expenditure 既资本性支出，一般是固定资产的投入，云模式降低了跟基础设施相关的运营成本，包括人员成本，硬件维护成本，电力费用等，企业可方便的根据业务需求进行增减资源。

10.2 运营成本管理

实施成本管理是企业从业务经营视角来对运营成本进行分摊核算、预测、规划、优化、考核、奖惩，保障资源尽可能的合理利用，并结合产品质量和效率，以达到平衡，是企业追求价值最大化的必然选择，但并不是单纯追求成本最低利润最大化，其中最重要环节是预算管理和成本优化。

10.3 预算管理

IT预算管理是指在公司战略目标的指导下，对未来经营周期内（通常指一年）IT系统为支持业务正常运转和发展所需要的资源进行充分、全面的预测和规划，通过预核算执行率监控，不断比对和分析实际完成情况与预算目标，从而及时改善和调整预算，预算不足时需要及时做预算滚动，预算过多需要控制使用，以帮助管理者更加有效地管理IT系统花费，并支持实现战略目标。

10.3.1 预算管理组织体系

预算管理组织体系由决策机构、工作机构、执行机构三个层面组成，承担着预算编制、审批、执行、控制、调整、监督、核算、分析、考评及奖惩等一系列预算管理活动。

10.3.1.1 决策机构

预算委员会是在公司董事会领导或授权下，预算管理事务的决策机构，职责是通过定期或不定期召开预算工作会议决定和处理全面预算管理的重大事宜。

表13 决策机构

级别	预算编制	预算控制	预算分析	人员机制	工具能力
1	管理层直接给出未来一年的预算金额。	无	无	管理层	Excel表格记录 电子邮件审批
2	重点是理顺流程，开始有合理性依据：编辑颗粒度细化到产品，不同产品业务运维根据历史消耗资源量，结合产品KPI进行资源量预测，然后预算管理员汇总，财务换算为金额。	颁布预算管理流程规范（申请、审批、使用）； 管理层对不同产品预算评审； 预算执行按年按部门。	预算评审时分析总成本、现金流和产品KPI增长是否吻合。	管理层 Ops 财务 预算管理员	Excel表格记录 电子邮件审批
3	同上外，重点是建立合理的资源量推导模型（不同资源类型容量瓶颈）和产品指标的预测（历史数据、生命周期、KPI）；增加预算滚动机制，预算不足时及时发起预算追加；	预算评审部门评审侧重于产品技术合理性，公司评审侧重战略、行业对标（产品策略和技术）。 预算执行颗粒度更细，按月按产品； 同一个部门不同产品有预算拆借账单	不同部门、产品预算执行率分析。	同上外， 评审委员会 DevOps	预算系统支持流程管理和预算执行数据记录。 例如：申请录入、审批修改、报表查询、采购数据等功能。 不同的产品录入后汇总为部门预算和公司预算供评审委员会和财务评审。
4	同上外，人工和机器学习双轨齐下优化资源模型，资源瓶颈	同上外，按不同产品或部门考核预算执行率，	同上外，分析产品指标预测与实际是否偏差。	同上外， QA AIOps	同上外，重点是系统收集预算编辑需要的数据（产品

	来自现网压测数据。带宽预算使用正向推导，而不是历史数据反推。	预算执行过程数据被有效记录。			指标、运维指标、资源量、容量等）尽可能减轻运维合理性推导工作量； 提供对外系统接口可以与容量管理、采购管理、活动项目对接； 自动统计预算考核数据。
5	同上外，预算编辑的业务指标历史数据采集与预测、实际使用容量现状统计、不合理现状可优化点自动测算，业务运维根据调整预测指标值、资源使用模型参数、成本优化人力投入，输出合理的预算结果。管理层根据战略、财务情况等来做最终决策。 系统根据产品实际指标增值情况提出预算滚动的建议。	全员都有良好的预算控制意识，自觉遵守预算控制流程，对未来工作有良好计划，避免盲目；预算控制流程、效率持续改善。	基于数据驱动，实时获取预算执行率，并自动趋势分析，预算偏差分析，并产生预警，相关人员主动进行纠偏，预算执行与预算计划偏差率低；预算分析流程、效率持续改善。	同上	同上外，建立自动建模能力，系统有效分析执行力偏差原因，重点模块实现智能预测下一年度预算。

10.4 成本优化

应用的部署成本和容量成本，除了与应用本身的特点有关，还与交付模式、应用架构密切相关，稳定、快速的持续交付能力，可以满足弹性交付的需要，并可节省部署资源，优秀的应用架构，决定了应用的扩展能力和部署模式，从应用管理视角出发，对资源管理，交付模式，应用架构，人员机制，工具能力这五方面进行合理安排，会有效的节省应用部署及运行成本，如表16所示。

表14 成本优化

级别	资源管理	交付模式	应用架构	人员机制	工具能力
1	无成本管理意识；	人工部署，手工创建、配置各资源，交付周期以周计；	传统三层应用架构，应用架构水平扩展能力不足；	无	无
2	建立应用资源使用标准规范，并建立对主机、网络、CDN、存储等	定义交付流程，OPS通过工具完成交付链上各环节资源准备，但	引入中间件层，应用层，数据层支持有限水平扩展能力；	OPS	有基本的工具支持，可支持主机

	资源使用的信息收集机制； 周期性评估资源使用情况；	仍需人工进行流程流转，交付周期以天计；			资源的自动化创建；
3	同上，结合业务目标需求，根据应用资源使用数据，性能数据，精确建立各业务应用的容量模型； 并事人工触发资源扩容/缩容操作；	打通持续交付流程，用户自助完成测试、生产资源申请及代码发布； 交付周期以小时计；	应用层进行有效垂直拆分，数据层利用缓存，读写分离，分库分表等 Sharding 技术，降低对单实力的配置要求，并实现水平扩展；	DevOps	工具或工具集支持大多数基础设施资源（计算实例、存储资源、网络资源、数据库实例、负载均衡实例）的创建、删除、管理；
4	同上，并结合应用容量管理数据，实时资源使用情况，进行动态编排，峰值自动扩容，低谷自动缩容；并根据应用运行数据，识别闲置资源，并自动回收；	按需交付，自动化工具部署；交付周期以分钟计；	面向SOA，服务化应用架构，服务自发现，自注册，各应用支持独立部署，水平扩展；但基于专用的中间件系统或组件，需申请VM独立部署；	DevOps AIOps	持续交付平台打通所有环节，完成应用的一站式端到端交付； 建立OPS数据仓库，有AI工具集支持；
5	同上，并依据应用运行大数据，对应用进行画像标示，调度系统实现应用亲和性部署；并充分利用 Online/Offline资源消耗特点，实现离线应用与在线应用的混合部署，提升资源利用率；	研发人员自助，基于容器镜像的应用即时端到端交付模式；交付周期以秒钟计；	应用架构微服务化，基于云原生设计，充分利用云平台服务例如：消息队列，大数据，存储等；	DevOps AIOps	建立运维一体化平台； 建立AIOps数据分析平台； AIOps平台为运维一体化平台提供决策引擎；

11 连续性和可用性服务

DevOps服务团队需要监控、评估和记录DevOps服务的连续性和可用性的需求，识别DevOps服务的连续性和可用性风险，制定连续性和可用性的计划，建立连续性和可用性的度量指标体系，实施计划并监控指标的达成情况，采取持续的主动改进措施，持续满足约定的业务及用户的连续性和可用性需求。

11.1 连续性管理

连续性服务是组织在计划和应对重大故障和灾难时的一种战略和战术能力。它致力在重大故障或灾难情况下，确保业务在一个预先定义的可接受的水平上能够持续的运作。DevOps连续性是IT连续性的组成部分，IT连续性业务连续性的一个组成部分。IT服务的连续性和高可用性事关业务的生死存亡。

组织需要按照业务和用户的要求保障DevOps服务的连续性或应对重大故障或灾难能及时恢复服务，以支持业务的持续发展，体现出DevOps服务的价值。

组织需要制定DevOps连续性服务的计划、流程，通过持续的软硬件容灾与备份设施设备的投入，确保所需的技术和服务设备（包括计算机系统、网络、应用、技术支持和服务）在预定的恢复点目标（RPO）和恢复时间目标（RTO）进行恢复。平时要通过演习来检验这套连续性机制是否健全、是否可靠，以确保业务和服务的连续性。

连续性服务可以从连续性活动、数据备份、容灾能力、演习、人员和文化等多个维度来衡量其成熟度。

表15 连续性管理

级别	连续性计划和流程	数据备份活动	演习演练	关键业务容灾能力	人员	文化
1	a)有 IT 连续性管理流程； b)有 IT 灾难恢复计划。	a) 完全数据备份至少每月一次； b) 备份数据在同一数据中心；	无演习演练，或只有不定期的桌面检查、走查。	冷备，核心业务 RPO>24 小时，RTO>72 小时	a) 有都由技术人员组成的灾难恢复团队	a) 技术组织自发进行临时性的连续性学习
2	同上，且有业务恢复计划	a) 完全数据备份至少每周一次； b) 数据同城备份；	不定期的桌面检查、走查、模拟演练	冷备，核心业务 RPO<24 小时，RTO<72 小时	a) 同上 b) 团队仅覆盖部分关键业务	a) 有组织的按季度的连续性服务宣传和培训
3	同上，且有业务连续计划。	a) 完全数据备份至少每天一次； b) 数据异地备份	不定期的桌面检查、走查、模拟演练	冷备，核心业务 RPO<12 小时，RTO<36 小时	a) 同上 b)有 IT 业务连续性组织 c) 团队覆盖全部关键业务。 c) 业务人员参与连续性组织。	a) 有组织的按月度的连续性服务宣传和培训。
4	同上，且有业务连续性管理。	a) 同上 b) 时时增量备份 C) 数据多于 2 份备份	定期（每季度或每月）桌面演练、沙盘演练、模拟演练、部分系统演习、全面演习	热备，核心业务 RPO<5 分钟，RTO<2 小时	a) 同上 c) 供应商人员参与连续性组织。 d) 公司高层担任连续性管理的最高领导。	a) 有组织的按月度或不定期的连续性服务宣传、培训和知识竞赛。 b) 开设连续性学习专栏。
5	同上，且以上流程、规范，能够持续改进。	a) 同上 b) 数据多于 3 份备份	定期（每季度或每月）桌面演练、沙盘演练、模拟演练、部分系统演习、全面演习	热备，核心业务 RPO<1 分钟，RTO<30 分钟	a) 同上 b) 供应商和上下游合作伙伴参与到连续性组织。	a) 同上

11.2 可用性管理

可用性是服务、组件或配置项在需要时，执行其约定功能的能力。DevOps服务团队可以通过业务分析（BA）和业务影响分析（BIA）、以及访谈、问卷调查等措施，了解业务和用户对可用性的需求，基于对需求理解，建立可用性管理组织，制定可用性计划，建立可用性度量指标体系，识别可用性风险，设计和构建DevOps高可用架构，通过日常监控系统采集和分析可用性指标的达成情况，主动开展可用性优化措施。

DevOps可用性服务可以从可用性架构、可用性管理、可用性组织、可用性总体指标和平均故障修复时间（MTTR）等维度来衡量其成熟度。

表16 可用性管理

级别	高可用服务架构	可用性管理	可用性组织	业务可用性指标	MTTR
1	a) 数据本地存储; b) 个别关键 IT 系统高可用架构设计。 c) 竖井式架构为主。	被动的可用管理活动	无可用性管理团队	a) 低可用性：可用性<99% b) 年度停机时间>87.6 小时	a) 持续集成失败平均故障修复时间>8 小时; b) 持续交付失败平均故障修复时间>8 小时; c) DevOps 平台平均故障修复时间>24 小时
2	a) 同城灾备中心; b) 部分关键 IT 系统高可用架构设计。 c) 竖井式架构为主，分布式架构为辅。	a) 实施简单的主动的可用性管理。 C) 实施非端到端的可用性监控。	a) 有可用性管理团队。 b) 可用性管理团队关注技术上的可用性。	a) 中可用性：可用性>99% b) 年度停机时间<87.6 小时	a) 持续集成失败平均故障修复时间<8 小时; b) 持续交付失败平均故障修复时间<8 小时; c) DevOps 平台平均故障修复时间<24 小时
3	a) 异地灾备中心; b) 大部分 IT 系统高可用架构设计 c) 个别关键 IT 系统容错架构设计 d) 竖井式架构为辅，分布式架构为主	a) 实施全面的主动的可用性管理。主动的可用性趋势分析，采取主动措施，消除可用性风险; C) 可用性监控是端到端的监控。	a) 有可用性管理团队。 b) 可用性管理团队关注技术、流程等多个因素，构建全面的可用性管理体系。	a) 较高可用性：可用性>99.9% b) 年度停机时间<8.8 小时	a) 持续集成失败平均故障修复时间<4 小时; b) 持续交付失败平均故障修复时间<4 小时; c) DevOps 平台平均故障修复时间<8 小时
4	a) 全部 IT 系统高可用架构设计;	a) 同上，且需求调查与可用性计划得	a) 有可用性管理团队	a) 较高可用性：可用性>99.9%	a) 持续集成失败平均故障修复时间<2

	b) 大部分关键 IT 系统容错设计。	到业务和用户的认可。 C) 所有系统实现端到端的可用性监控。 d) 全面支持自动化运维。	b) 可用性管理团队关注技术、流程、人员等多个因素，构建较全面的可用性管理体系。	b) 年度停机时间 <8.8 小时	小时； b) 持续交付失败平均故障修复时间 <2 小时； c) DevOps 平台平均故障修复时间 <4 小时
5	a) 分布式多活数据中心； b) 全部关键 IT 系统容错设计 c) 架构具备自我修复和自我优化的能力。	a) 同上，且全面支持智能化运维，故障自愈。	a) 有可用性管理团队。 b) 可用性管理团队关注技术、流程、人员和供应商等多个因素，构建全面的可用性管理体系。	a) 较高可用性：可用性 >99.9% b) 年度停机时间 <8.8 小时	a) 持续集成失败平均故障修复时间 <0.5 小时； b) 持续交付失败平均故障修复时间 <0.5 小时； c) DevOps 平台平均故障修复时间 <1 小时

11.3 应用事件管理

事件管理是当系统出现故障时，对故障进行的应急处理，快速恢复业务的过程管理。事件管理的过程主要包含事件发现与登记、事件分析与恢复、事件原因定位与根除，事件反馈与验证，事件总结与整改。日常运维，需要实现制定各种场景的事件应急预案并进行演练，一旦事件发生，采用事件应急预案来快速恢复业务是关键，避免沉浸在查找事件原因而造成对业务造成影响。

事件管理可以提高对突发事件和各类故障的应对处理能力，能更好的保障系统稳定可靠运行，提升业务连续性。事件管理涉及到维护和研发团队，要求维护和研发团队高效协同，以业务恢复和抢通为第一共同目标。

11.3.1 应急事件管理

表17 应急事件管理

级别	事件管理状态	事件管理过程的工具手段	可视化	人员
1	无监控，用户先于维护人员发现事件，没有事件响应团队，事件无登记，事件级别无定义，不通知用户，不汇报，运维人员自行处理事件。	事件处理过程没有工具。	事件过程和状态封闭，停留在具体处理事件的运维人员层面。	无监控响应团队，由开发团队负责处理问题。
2	有监控，用户先于维护人员发现事件，有事件响应团队，及时登记事件，有时通知用户和报告，故障处理有应急预案。	事件处理过程有数据可分析，有通知和报告的自动化工具。	事件过程和状态在团队内可见，会同研发团队一起处理事件。	有服务响应团队，响应时间为 5*8，开发人员，兼职模式。

3	有监控，先于用户发现故障，有事件级别定义与相关处理要求，及时通知用户，能利用应急预案优先恢复业务，并找到事件引发的原因。	事件处理过程有数据可关联分析，有协同工具进行事件会诊处理。	事件过程和状态在团队内可见，与研发团队有共同度量指标，会同研发团队，快速恢复业务。	有 5*8 小时的专业服务响应团队，负责主要业务的事件响应，事件的处理主要由研发团队支持。
4	事件能事先自动预警，有事件管理规范并执行，有各种场景的应急预案，一旦发生事件，可按照预案，快速恢复业务，减少对业务的影响。	有自动预警的手段，有事件处理过程的自动化工具。	事件过程和状态在团队内可见，并可对过程信息进行趋势分析等。	有 7*24 小时的专业服务响应团队，可进行部分事件的事件处理。
5	事件自动预警，自动诊断，自动修复，有事件管理规范并执行，有各种场景的应急预案，并按照预案自动恢复业务，对业务没有影响。	能根据历史监控数据进行预警、诊断和修复。能对事件的处理全过程进行管理和记录。	事件过程和状态在团队内可见，并可对过程信息进行分析和综合评价和优化。	有 7*24 小时的专业服务响应团队，有一键处理的智能化工具第一时间进行事件处理。

12 用户体验管理

用户体验（User Experience，简称 UE/UX）是用户在使用产品过程中建立起来的一种主观感受。一般用户体验是有关于产品设计方面的，不同的产品对用户体验的追求不同。本章节提到的用户体验指的是通过技术运营手段来提升用户使用产品直观感受。例如：提升流畅度、缩短响应时间等等。

用户价值是企业赖以生存的核心竞争力，诸多优秀公司也将对最终用户提供极致服务最优体验视作价值愿景。在技术运营过程中，用户体验管理通过技术运营手段，让用户使用产品更加顺畅，付出最小的成本达到用户目的。

用户价值是技术运营中的一项进阶能力要求，需要技术运营团队同时具备业务认知能力、数据管理能力、体验优化能力。本规范主要目的是指导技术运营团队在技术运营过程中更好的改善产品的用户体验。基于用户体验的服务能力，分解为业务认知能力、数据管理能力、体验优化能力。

12.1 业务认知

业务认知指的是运营团队对产品在业务流程上的掌握能力、体验异常的发现能力，业务认知包括业务流程知识管理和业务知识管理两部分，如表20所示。

12.1.1 业务流程知识管理

业务流程知识管理，是指技术运营团队对所运营产品的业务流程，产品功能，使用方法的了解和掌握的能力。例如：注册流程、支付流程、购买流程，登录流程等业务的核心流程及功能。

12.1.2 业务知识管理

业务知识管理，对业务和行业的知识管理能力，团队具备持续的学习能力，建立在深度理解产品的基础上，提升服务能力快速处理问题异常。

表18 业务认知

级别	业务流程	业务知识	工具	人员
1	不了解业务流程。	不熟悉所负责产品和行业。	无用户体验工具。	无用户体验改善人员。
2	了解业务流程。	具备基本的培训，对产品有一定理解。	利用简单的监控类工具寻找用户体验问题。	无用户体验改善人员。
3	掌握核心业务流程。	有较为全面的业务理解，合理解决业务问题。	有较全面的监控工具，但是在用户体验方向上仍然比较简单。	较少的精力投入用户体验相关工作。
4	精通核心业务流程。	对业务有深入理解和学习，关注竞品。	有专用的用户体验工具，可以发现常见用户体验问题。	有专门的人力投入用户体验优化工作。
5	精通所有业务流程。	精通所处行业，能够通过技术优化促进业务提升。	有智能化的用户体验处理工具，可以定位大多数用户体验问题。	有专门的人力组建用户体验优化团队。

12.2 数据管理

数据管理是运营团队对产品的数据采集能力、数据监控能力、数据分析能力，如表21所示。

12.2.1 数据采集

数据采集能力，是指运营团队对数据的获取能力。是否有合法获取数据的手段，是否能够以合理的方式，合理的频率进行数据采集和上报。

12.2.2 数据监控

数据监控能力，是指运营团队对产品的各种采集数据具备监视、控制和告警能力。例如：对某一页面打开时间的监控数据。

12.2.3 数据分析

数据分析能力，是指运营团队对用户数据进行分析，挖掘更大能力。例如：通过用户行为数据分析，建立用户画像。

表19 数据管理

级别	数据采集	数据监控	数据分析	工具	人员
1	无数据采集能力	无数据监控	无数据分析能力	无数据管理工具	无数据管理人员
2	常规用户数据采集。	常规数据监控	简单的数据统计	无数据管理工具或者简单的开源工具进行管理。	有较少人力负责工具搭建、维护和数据统计。
3	多维度用户数据采集，没有明确目标和方向性。	典型用户场景数据监控，大量数据没有监控能力。	开始积累大数据，初步具备数据分析能力	有较完善的数据管理工具。	有较少人力负责工具搭建、维护和数据统计。
4	数据采集形成标	形成标准用户场景	具备数据分析能	有专业的大数据管	有专门的人负责数

	准，有明确的格式和采集目标。	数据监控，有针对性的进行告警。	力，能够用图表展示分析结果，通过分析结果可以指导数据采集的方向和优化方案。	理平台，具备大数据分析能力。	据平台的开发、优化、维护。
5	能够控制用户数据采集，仅针对目标用户做标准数据采集。	精准告警某一类目标用户的指定操作。	具备完善的分析能力，能够提供用户体验数据标签。	有专业的数据平台和数据挖掘平台具备 AI 算法能力。	有专门的人力和团队负责开发、优化、维护。

12.3 体验优化

体验优化，是指运营团队能够对体验问题进行定位、解决，甚至可根据数据分析做出决策的能力，如表22所示。

12.3.1 体验异常发现

体验异常发现是问题定位的能力，是指运营团队能够凭借数据、图表展示等方法快速定位体验问题的根本原因的能力。

12.3.2 体验异常定位

体验问题解决能力，是指运营团队通过技术优化、数据分析、流程改造、产品设计等手段解决体验问题的能力。

12.3.3 体验异常决策

体验问题决策能力，是指运营团队不需要根据经验进行决策，而是由体验管理系统的数据分析结论自动决策应该如何解决体验管理的问题。

12.3.4 体验异常修复

表20 体验优化

级别	体 验 异 常 发现	体验异常定位	体验异常决策	体验异常修复	工具	人员
1	没有区分异常场景和数据上报，无法发现问题。	一般运维技术能力 不能定位体验异常	无体验异常决策能力	无体验异常修复能力	无用户体验发现和修复工具	无用户体验改善人员
2	有基本的日志分类和异常记录，可以通过日志发现问题。	较好的系统运维技术，可以定位简单体验异常。	无体验异常决策能力，但是可以展示体验异常原因。	通过手工修复简单体验异常。	利用简单的监控类工具寻找用户体验问题。	同上
3	通过主动监控的建设能够及时发现	全面的技术运维能力，可以定位复杂的体验异常	无体验异常决策能力，但是能够给出建议方案。	通过标准化工具修复常见体验异常。	有较全面的监控工具，但是在用户体验方向上仍	较少的精力投入用户体验相关工作。

	通用问题并快速处理。	常。			然比较简单。	
4	对于异常有清晰全面的分类,监控提前发现并且有较高的准确率。	全面的技术运维能力和丰富的基础设施运维能力,可定位复杂环境下,系统内、系统外的体验问题。	有体验异常决策能力,提供决策参考,但是不具备执行能力。	手工启动自动化工具修复常见体验异常。	有专用的用户体验工具,可以发现常见用户体验问题。	有专门的人力投入用户体验优化工作。
5	通过大数据的采集能力以及智能算法的投入,能够预测问题并前置修复。	全面的技术运维能力和丰富的基础设施运维能力,可预见复杂环境下,系统内、系统外可能存在的体验问题。	能够对体验异常进行决策,并具备执行能力。	自动启动自动化的对体验异常进行修复。	有智能化的用户体验处理工具,可以定位大多数用户体验问题。	有专门的人力组建用户体验优化团队。

13 运营一体化平台

运营一体化平台是对企业内的IT资源整合,依托云计算的设计理念将技术运营领域中所涉及的底层通用能力抽象成平台,包括统一管控能力,资产配置的统一管理能力,运营数据的采集、传输、汇聚、计算以及存储分析的能力等,提供一整套完备的集成开发框架以及运行托管服务,实现对各类异构业务下技术运营场景层的监控、事件、变更、容量、成本、连续性服务以及用户体验的低成本快速构建。

13.1 运营一体化平台整理框架

通过运营一体化平台,实现了平台与场景的分离,避免了以往运营场景底层能力建设难度大、建设重复度高的弊病,助力技术运营人员在瞬息万变的业务场景中快速满足业务稳定运行以及提升运营价值,同时在面对敏捷迭代的业务版本以及企业组织架构调整等不可控状况下都能具有良好的适配能力和灵活可扩展性。

13.2 IaaS 统一管控能力

IaaS统一管控能力是整个技术运营一体化平台的底层支撑体系,是上层PaaS与底层IaaS的连接器,为上层提供指令、文件、数据的通道,支持直连、代理(Proxy)模式。统一管控能力需采用分布式C/S架构,主要包含智能代理(Agent)和提供文件分发传输、命令实时执行、大数据采集与传输服务能力的Server以及服务注册与发现、KV存储、关系型数据库等模块。

在技术运营一体化平台体系建设中,统一管控能力为各平台模块提供了人机交互的通道与能力,并提供三种类型的服务能力:文件分发传输能力、命令实时执行与反馈的能力、大数据采集与传输的能力,实现企业内IDC、各类公有云的独立管控以及混合云的统一管控。

表21 IaaS 统一管控能力

级别	跨云管理能力	文件传输类型	执行任务类型	数据采集服务
----	--------	--------	--------	--------

1	一定数量节点的私有云管控能力	仅支持文件的直传模式	支持Linux下的通用Bash命令执行，批量能力弱	支持固定字段的采集上报
2	一定数量节点的公有云管控能力	支持文件、文件目录的直传模式	批量能力较强的Bash命令执行能力	自定义数据采集
3	海量节点的公（或私）有云的管控能力	支持一定大小以上的文件（文件目录）采用BT模式	支持Linux上除Bash以外的其它脚本语言（如python、perl等）	自定义数据采集、采集器插件化支持
4	一定数量节点的混合云管控能力	根据文件（文件目录）智能判断采用直传或BT模式，同时支持正则匹配	除Linux外，还支持Windows下的BAT或PowerShell	自定义数据采集、采集器插件化支持、服务自助发现
5	海量节点的混合云管控能力	智能的BT模式以及正则匹配，区域链控制、跨区域穿透	同上，且支持AIX下的ksh	自定义数据采集、采集器插件化支持、实时数据快照、动态负载均衡

13.3 原子平台能力

原子平台能力为IaaS之上的管理与技术运营基础操作平台，具体包括：技术运营操作对象管理的CMDB模块，运维脚本执行和文件传输的作业功能模块，对基础设施、应用程序、中间件等各类环境的运行状态进行数据采集和分析的运维数据平台模块，对容器的镜像生产、管理、资源调度的容器管理模块，以及建设智能化运维为目标的功能模块，以及其它满足技术运营基础操作的独立统一功能模块。

表22 原子平台能力

级别	CMDB	作业功能	运维数据平台	容器管理	数据挖掘
1	基于业务层面的主机资源管理	支持一定数量级别的并发脚本、文件传输作业的执行	不适用	不适用	不适用
2	基于业务层面的主机资源管理、业务拓扑资源管理、业务管理	支持一定数量级别的并发脚本、文件传输作业的执行及Web化脚本管理	不适用	不适用	不适用
3	基于业务层面的主机资源管理、业务拓扑资源管理、业务管理、自定义属性管理、进程端口与配置文件管理、操作审计	支持一定数量级别的并发脚本、文件传输作业的执行，Web化脚本管理，作业编排	满足统一数据接入、可视化计算任务配置管理	应用仓库、容器编排和调度服务、多环境一致性管理	不适用
4	基于业务层面的主机资源管理、业务	支持一定数量级别的并发脚本、文件	满足统一数据接入、可视化计算任	应用仓库、容器编排和调度服务、多	拖拽式建模、交互式测试调优、自动

	拓扑资源管理、业务管理、自定义属性管理、进程端口与配置文件管理、操作审计、支持自定义模型&业务架构层级调整	传输（支持“多对多”，“多对一”模式）作业的执行，Web 化脚本管理，作业编排	务配置管理、运维数据存储查询	环境一致性管理、容器网络服务	化模型评估、模型训练运行管理
5	基于业务层面的主机资源管理、s 业务拓扑资源管理、业务管理、自定义属性管理、进程端口与配置文件管理、操作审计、支持自定义模型&业务架构层级调整、拓扑的实时发现	支持海量节点的并发脚本、文件传输（支持“多对多”、“多对一”模式）作业的执行，Web 化脚本管理，作业编排	满足统一数据接入、可视化计算任务配置管理、运维数据存储查询、可视化建模管理	满足应用仓库、容器编排和调度服务、多环境一致性管理、容器网络服务、容器安全服务、一体化监控和日志查询服务	拖拽式建模、交互式测试调优、自动化模型评估、模型训练运行管理、场景模型（公共的通用的模型）

13.4 PaaS 能力

PaaS是一个开放的平台，包含完善的前后台开发框架以及用于支持用户简单快速的创建、部署和托管应用的APaaS（Application Platform as a Service），以及提供了服务总线（ESB）、调度引擎、公共组件等模块的IPaaS（Integration Platform as a Service）。PaaS为每一个运营场景SaaS从创建到部署，再到后续的维护管理提供了完善的自助化和自动化服务，如日志查询、监控告警等。

表23 PaaS 能力

级别	IPaaS	APaaS
1	服务总线	支持通用开放的开发框架
2	服务总线、API 网关	支持通用开放的开发框架、基于虚拟机的 SaaS 免运维托管
3	服务总线、API 网关、调度引擎	支持通用开放的开发框架、基于 Docker 容器的 SaaS 免运维托管
4	服务总线、API 网关、调度引擎，同时支持频率控制、防爆破机制。	支持通用开放的开发框架、基于 Docker 容器的 SaaS 免运维托管、全流程自动化上线布署
5	服务总线、API 网关、调度引擎、通用的多类公共组件	支持通用开放的开发框架、基于 Docker 容器的 SaaS 免运维托管、全流程自动化上线布署、SaaS 运营数据可视化

13.5 运维场景能力

运营场景能力是基于PaaS能力之上快速构建的面向技术运营场景解决方案的载体。它通过对底层各平台能力原子功能的拼装，实现监控管理、事件管理、变更管理、容量管理、成本管理、连续性服务、用户体验等各类技术运营场景。

表24 运维场景能力

级别	监控能力	自动化运维能力	ITSM 管理能力	运营分析能力
1	基础指标（cpu、内存、硬盘容量、网卡流量等）的监控	基础运维（自动化发布、日志查看、数据提取）	简单的运维工单流程管理	不适用
2	基础指标+通用公共组件的监控	CI/CD(代码自动构建、版本自动部署、自动化测试、版本管理)	运维工单流程&运维权限申请等流程管理	不适用
3	基础指标+通用公共组件+业务特性(如业务进程、注册、在线等)的监控	基于调度编排的跨系统全流程调度自动化。	基于 ITIL 的事件管理、故障管理、变更管理、容量管理、成本管理等	基于普通业务数据的可视化展示
4	基础指标+通用公共组件+业务特性(如业务进程、注册、在线等)的监控+关联分析+根因分析	基于无人值守的全流程自动化(如自动扩缩容、自动开服)	基于 ITIL 的各流程管理与自动化运维操作相融合，由自动化操作节点创建工单、同时由自动化操作驱动流程。	详细的业务各指标维度展示，并能根据指标范围提供对应的改进建议等
5	基础指标+通用公共组件+业务特性(如业务进程、注册、在线等)的监控+关联分析+智能算法根因分析+故障自愈	基于人工智能决策的全流程无人值守自动化	基于 ITIL 的各流程管理与自动化运维操作相融合，由自动化操作节点创建工单、同时由自动化操作驱动流程。同时借助 AI 能力对现有流程进行优化并进行改进措施推荐。	基于业务全生命周期、用户每一体验节点全程运营数据节点多维度展示,同时借助 AI 能力智能推荐处理方式及智能回归总结。

附 录 A
(规范性附录)
容量与性能管理指标举例

服务器物理 CPU 个数	交易量
逻辑节点 CPU 使用率	作业执行时间
服务器物理内存大小	表空间使用率
逻辑节点内存使用率	应用文件系统使用率
逻辑节点 SWAP 使用率	共享内存使用
逻辑节点磁盘 busy	消息队列使用
逻辑节点服务网卡使用	队列等待处理的请求数量
逻辑节点文件系统使用率	队列的每秒处理数量
数据库表空间使用率	应用内部连接池活动连接数
数据库活动日志序列号	应用内部连接池可用连接数
数据库缓冲池命中率	应用内部连接池等待连接数
数据库连接数	应用内部连接池重连次数
存储空间使用率	应用内部连接池连接延迟时间
全局/应用用户 nofiles	应用工作线程池活动线程数量
apache 连接数	应用工作线程池可用线程数量
apache close wait 数量	应用自身处理的请求的 QPS
Web 容器 JVM 内存、使用率	应用自身处理的平均时间
Web 容器连接数	应用自身处理的超时率
gc/fd 频率	应用调用后端的平均返回时间
缓存命中率	设备连接数
缓存命中率	加密机连接数
缓存连接数	机房系统供电容量
缓存内存大小	机房空调系统制冷容量
互联网接入线路带宽	TPS
CDN 带宽	
CDN 连接数	
CDN 访问时延	

广域网接入网络带宽	
-----------	--

附 录 B
(规范性附录)
容量与性能管理指标举例

前端性能测试点	加载时长	dom 节点过多
		频繁改动 dom 结构;
		低效的 css
		执行时间过长的 js 逻辑
		js 执行后出现大量错误
		无缓存机制
		图片尺寸大
		数据未压缩
		http 连接未复用
		未使用分布加载机制
		图片未分域名
		无效资源
		动画效果
		返回包含 404 或者 302 跳转
		PC 和移动端未做相应页面映射
	流量	无缓存机制
		大图
		数据未压缩
		http 连接未复用
		静态资源使用 cookie
基础组件性能测试点	Web 服务器	第三方库未使用精简版(Jquery)
		硬件资源 (cpu, 内存消耗)
		网络 IO (进出口带宽, 数据分发, 吞吐量)
		请求连接数
		节点数
	应用	低效配置
		低效代码
		网络 IO (服务转发, 吞吐量)
		多线程
		节点数
		路由解析
		容器配置 (例如 jvm 初始化配置)

		微服务架构的总管理端（发现服务，转发节点，收集日志）
	数据库	弱查询语句
		主从服务器节点数
		读写未分离
		数据库 IO（db 事务吞吐量）
		数据库引擎类型
		整体数据量
	数据缓存服务器	节点数
		命中率
		数据量大

参 考 文 献
