

ICS 35.240

L70

# 团 体 标 准

T/IQA 15-2022

## 软件研发效能度量规范

Measurement Specification of

Software Development Efficiency, Effectiveness and Excellence

(发布稿)

2022-09-07 发布

2022-09-08 实施

中关村智联软件服务业质量创新联盟 发布

## 目 次

前 言 .....	V
1 范围 .....	1
2 规范性引用文件 .....	1
3 术语和定义 .....	1
3.1 组件 component .....	1
3.2 数据 data .....	1
3.3 信息需要 information need .....	1
3.4 软件产品 software product .....	2
3.5 软件服务 software service .....	2
3.6 软件组织 software organization .....	2
3.7 软件绩效改进 software performance improvement .....	2
4 软件研发效能度量框架 .....	2
4.1 软件研发效能 .....	2
4.1.1 效率 .....	2
4.1.2 效果 .....	2
4.1.3 卓越能力 .....	2
4.2 度量框架 .....	2
4.2.1 认知域 .....	3
4.2.2 改进域 .....	3
4.3 度量方法 .....	3
4.3.1 确立目标和范围 .....	3
4.3.2 选取度量指标 .....	4
4.3.3 实施度量和改进 .....	4
4.4 度量原则 .....	5
4.4.1 适用性 .....	5
4.4.2 系统性 .....	5
4.4.3 可靠性 .....	5
4.4.4 持续性 .....	5
5 软件研发效能度量指标 .....	5
5.1 度量指标模型 .....	5
5.1.1 数据集 .....	5
5.1.2 指标 .....	5
5.1.3 度量 .....	5
5.1.4 视图 .....	6
5.1.5 模型定义 .....	6
5.1.6 指标定义 .....	6
5.2 度量指标集 .....	7

6 软件研发效能视图应用 .....	11
6.1 视图模型 .....	12
6.2 视图卡片 .....	12
附录 A.....	15
指标定义 .....	15
A.1 交付价值 .....	15
表 A.1.1 产品价值达成率 .....	15
表 A.1.2 客户满意率.....	16
表 A.1.3 客户问题响应时长.....	17
表 A.1.4 客户问题解决时长.....	18
表 A.1.5 用户增长率.....	19
表 A.1.6 市场占有率.....	20
表 A.1.7 营收增长率.....	21
A.2 交付速率.....	21
表 A.2.1 需求交付周期.....	21
表 A.2.2 开发交付周期.....	22
表 A.2.3 里程碑偏差.....	23
表 A.2.4 需求吞吐量（流速率） .....	23
表 A.2.5 流效率.....	24
表 A.2.6 需求颗粒度.....	24
表 A.2.7 需求按时交付率.....	25
表 A.2.8 需求变更率.....	26
表 A.2.9 组件按时交付率.....	26
表 A.2.10 组件复用率.....	27
表 A.2.11 接口变更率.....	27
表 A.2.12 代码开发当量.....	27
表 A.2.13 代码提交频率.....	28
表 A.2.14 测试一次通过率.....	29
A.3 交付质量 .....	29
表 A.3.1 演示频率.....	29
表 A.3.2 需求评审缺陷密度.....	29
表 A.3.3 需求评审通过率.....	30
表 A.3.4 设计评审缺陷密度.....	30
表 A.3.5 设计评审通过率.....	30
表 A.3.6 代码重复率.....	31
表 A.3.7 圈复杂度.....	31
表 A.3.8 静态扫描缺陷密度.....	32
表 A.3.9 代码走查缺陷密度.....	32
表 A.3.10 代码评审轮数.....	32
表 A.3.11 提测成功率.....	33
表 A.3.12 用例评审缺陷密度.....	33
表 A.3.13 用例评审通过率.....	33
表 A.3.14 测试覆盖率.....	34

表 A. 3. 15 测试缺陷密度.....	34
表 A. 3. 16 缺陷逃逸率.....	35
表 A. 3. 17 发布成功率.....	35
表 A. 3. 18 系统可用性.....	35
表 A. 3. 19 线上故障数.....	36
表 A. 3. 20 平均故障间隔时长.....	36
表 A. 3. 21 平均故障修复时长.....	36
A. 4 交付成本 .....	37
表 A. 4. 1 挣值.....	37
表 A. 4. 2 预算执行率.....	38
表 A. 4. 3 返工率.....	38
表 A. 4. 4 人力成本.....	39
表 A. 4. 5 非人力成本.....	40
表 A. 4. 6 工作量分布.....	40
表 A. 4. 7 技能指数.....	41
表 A. 4. 8 人员流动率.....	41
A. 5 交付能力 .....	42
表 A. 5. 1 流负载.....	42
表 A. 5. 2 构建频率.....	42
表 A. 5. 3 构建时长.....	43
表 A. 5. 4 构建成功率.....	43
表 A. 5. 5 缺陷重开率.....	43
表 A. 5. 6 测试自动化率.....	44
表 A. 5. 7 环境整备时长.....	44
表 A. 5. 8 部署频率.....	44
表 A. 5. 9 部署时长.....	45
表 A. 5. 10 部署成功率.....	45
表 A. 5. 11 回滚成功率.....	46
表 A. 5. 12 发布频率.....	46
表 A. 5. 13 发布时长.....	47
A. 6 持续改进 .....	47
表 A. 6. 1 改进效果评价.....	47
表 A. 6. 2 专项改进完成率.....	47
表 A. 6. 3 审计问题关闭率.....	48
表 A. 6. 4 审计频率.....	48
表 A. 6. 5 过程符合度.....	49
参 考 文 献 .....	51

## 前 言

本标准按照 GB/T 1.1—2020《标准化工作导则第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本标准的某些内容可能涉及专利。本标准的发布机构不承担识别专利的责任。

本标准由中关村智联软件服务业质量创新联盟提出并归口。

本标准起草单位：中关村智联软件服务业质量创新联盟、京东科技控股股份有限公司、横琴人寿保险有限公司、北京昊诚亚旭科技有限公司、北京思码逸科技有限公司、腾讯云计算（北京）有限公司、云加速（北京）科技有限公司、浙江中控技术股份有限公司、中国人寿财产保险股份有限公司、中电万维信息技术有限责任公司、金邦达有限公司北京分公司、紫光软件系统有限公司、北京四方继保自动化股份有限公司、中兴通讯股份有限公司、中国光大银行股份有限公司、株洲中车时代电气股份有限公司、平安银行股份有限公司、河北远东通信系统工程有限公司、网易（杭州）网络有限公司、中电海康集团有限公司、北京百度网讯科技有限公司、望海康信（北京）科技股份公司、京东方科技集团股份有限公司、潍柴动力股份有限公司、新华三技术有限公司、北京经纬恒润科技股份有限公司、华为技术有限公司、广联达科技股份有限公司、北京轩宇信息技术有限公司、科大讯飞股份有限公司、北京字节跳动网络技术有限公司、天翼云科技有限公司、中国民航信息网络股份有限公司、北京奕斯伟计算技术有限公司、中核控制系统工程有限公司、北京奇艺世纪科技有限公司、中国东方航空股份有限公司、黑龙江邮政易通信息网络有限责任公司、中国太平洋人寿保险股份有限公司、云智慧（北京）科技有限公司。

本标准主要起草人：胡延军、任晶磊、唐洪山、茹炳晨、马骏、郑伟娜、刘建军、陈敏华、张立明、葛田子、吕婧、胡晓天、高杨、徐延明、詹庆才、叶文华、陈晨、曾嵘、郭炜、付鸿飞、唐斐、钱峰、马国伟、张伟军、廖君仪、张茜、刘兴义、杨兰、刘毅、孟玉旺、徐毅、李强、吕芳、孙春艳、钟欣、董璐、林一鸣、孙静、宋晓旭、王冬、兴连博、陈连生、张瑜瑾、李景哲、高驰涛、蔡斌哲、关钦杰、李隽。

# 软件开发效能度量规范

## 1 范围

本标准提供了软件开发效能的基本框架、度量指标模型、基本度量指标集和视图应用。本标准所列的度量并非一个完整集合，使用者可以从本标准中选取合适的度量，或使用本标准未包含的其他度量。本标准适用于各类软件开发过程和组织，但并非每种度量均适用。

本标准适用于软件开发组织的以下活动：

- (1) 打造并展现软件研发和交付水平；
- (2) 通过持续改进提升软件研发和交付水平；
- (3) 寻求软件研发提供商并要求其确保软件交付水平；
- (4) 建设软件研发效能数据和分析平台；
- (5) 进行软件研发效能度量培训及审计。

本标准适用于以下使用者：

- (1) 软件研发组织管理者；
- (2) 软件研发组织中的产品经理、项目经理、技术经理、测试经理等；
- (3) 软件产品和服务的开发者；
- (4) 软件研发效能平台的开发者；
- (5) 软件研发效能分析师、咨询师、审计师、培训师等。

## 2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，仅注日期的版本（包括勘误的内容）适用于本标准。凡不注日期的引用文件，其最新版本（包括所有的修改单）适用于本标准。

GB/T 5271.1 信息技术 词汇 第1部分：基本术语  
GB/T 5271.20 信息技术 词汇 第20部分：系统开发  
GB/T 11457 信息技术 软件工程术语

## 3 术语和定义

下列术语和定义适用于本标准。

### 3.1 组件 component

在一个特定的分析层次上考虑的系统中带有分立结构的实体。诸如一个组合或软件模块。

[GB/T 18492—2001，定义 3.1]

### 3.2 数据 data

信息的可再解释的形式化表示，以适用于通信、解释或处理。

[ISO/IEC 25012:2008，定义 4.2]

### 3.3 信息需要 information need

管理目的、目标、风险和问题所需的洞察。

[ISO/IEC 15939:2007，定义 2.12]

### 3.4 软件产品 software product

一组计算机程序、规程以及可能的相关文档和数据。

[GB/T 8566—2007, 定义 3.27]

### 3.5 软件服务 software service

实施与软件产品有关的活动、工作或义务，比如软件开发、维护和运作。

[GB/T 8566—2007, 定义 3.28]

### 3.6 软件组织 software organization

以软件研发、交付和运营为主要业务的组织，包括：项目型组织、产品型组织等，通常以产品中心、研发中心、开发中心、交付中心、运营中心等形式存在，涉及应用软件、系统软件、平台软件、嵌入式软件的研发、交付和运营业务等。

### 3.7 软件绩效改进 software performance improvement

绩效改进能力是软件组织的关键能力，通常以 EPG、PMO、产品管理中心、质量管理中心等职能运行，采取强矩阵、弱矩阵、平衡矩阵等形式履行管理和改进职责，并针对人员能力、工具/平台、过程/子过程的进行持续改进以提升软件研发、交付和运营的卓越能力。

## 4 软件研发效能度量框架

软件研发效能度量框架（以下简称度量框架），规定软件研发效能度量的基本构成及其相互关系，用于指导对研发效能度量的全面理解。

### 4.1 软件研发效能

软件研发效能是持续快速交付高质量有价值的软件产品和服务的能力，包括效率（efficiency）、效果（effectiveness）和卓越能力（excellence）三个方面。

#### 4.1.1 效率

效率要求软件研发过程能够多、快、好、省地交付产品和服务，即价值高、速度快、质量好、成本低。

#### 4.1.2 效果

效果要求软件研发过程交付的产品和服务能够达成用户和业务的目标。

#### 4.1.3 卓越能力

卓越能力要求软件研发过程以健康的、可持续的方式交付产品和服务。

### 4.2 度量框架

软件研发效能度量框架由研发效能目标（E<sup>3</sup>CI objectives）、认知域（cognition area）和改进域（improvement area）三部分构成。研发效能目标，即研发效能定义的效率（efficiency）、效果（effectiveness）和卓越能力（excellence）（参见 4.1 节），统称为“E<sup>3</sup>”；认知域是认知研发效能主要方面的度量领域；改进域是对改进本身进行度量的领域，三个部分共同构成 E<sup>3</sup>CI 软件研发效能度量框架，如图 1 所示。

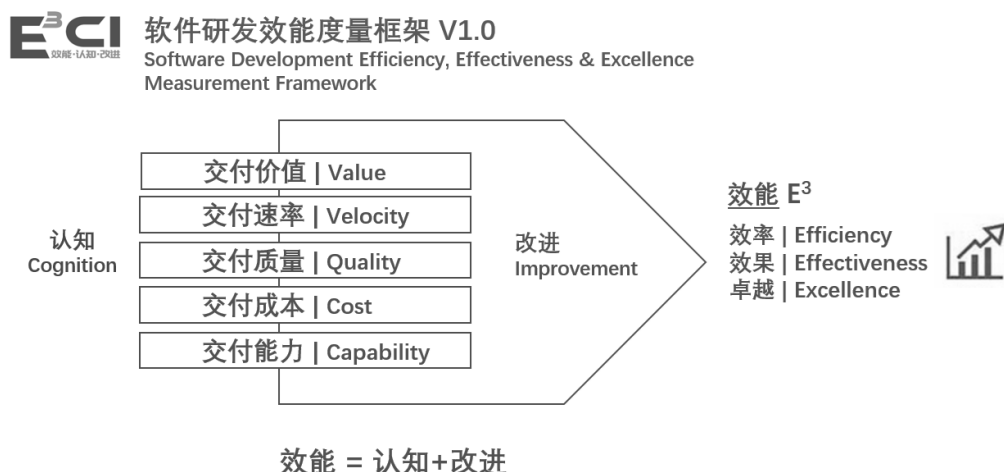


图 1 软件研发效能度量框架概要图

#### 4.2.1 认知域

##### 4.2.1.1 交付价值

交付价值 (delivery value) 认知域反映软件研发过程满足用户或业务目标的程度。

##### 4.2.1.2 交付速率

交付速率 (delivery velocity) 认知域反映软件研发过程交付产品和服务的快慢程度。

##### 4.2.1.3 交付质量

交付质量 (delivery quality) 认知域反映软件研发过程交付的产品和服务满足用户和业务需求的程度。

##### 4.2.1.4 交付成本

交付成本 (delivery cost) 认知域反映软件研发过程交付产品和服务的开销。

##### 4.2.1.5 交付能力

交付能力 (delivery capability) 认知域反映软件研发过程交付产品和服务的可持续性。

#### 4.2.2 改进域

改进域主要侧重持续改进 (continuous improvement)，反映在认知的基础上进行研发效能改进过程的成效。改进过程宜针对交付价值、交付速率、交付质量、交付成本、交付能力等领域进行系统或专项提升，以达成效率、效果和卓越能力等效能目标。

#### 4.3 度量方法

软件研发度量应按照确立目标和范围、选取度量指标、实施度量和改进三个步骤进行。

##### 4.3.1 确立目标和范围

软件研发效能度量应首先明确度量的目标，围绕目标提出问题，并进一步展开度量，防止贪大求全。目标的设立可从研发组织面临的挑战入手，宜在研发组织主要管理者或领导者间达成共识。度量目标应聚焦，可根据其重要性、迫切性和可行性对目标进行排序。度量的范围由目标针对的团队、项目或周期



确定，分别是人、事、时间维度上界定度量的范围。

#### 4.3.2 选取度量指标

软件研发组织可根据度量目标选择认知域，并从认知域中选择相应指标；如果目标是研发效能改进本身，应选择相应的认知域和改进域及相应指标。

软件研发效能的提升，依赖于组织认知的提升，只有对组织现状及其与目标的差距有充分的认知，才能够实施改进。确定认知域/改进域后，应从目标需要回答的问题或信息需要出发，选取能够回答当前问题或者满足当前信息需要的指标；在选取度量指标时，应充分考虑软件组织的现状和约束，可对度量指标进行优先级排序或基于权重进行计算，适当的变通是贴合实际和有效运用的要点。

图2展示研发效能度量框架展开后各认知域/改进域中的主要指标，并对应于或者横跨研发的需求、设计、开发、测试、发布和运营等各个阶段，各指标归属于其主认知域或改进域，同时也反映其他域的必要信息。

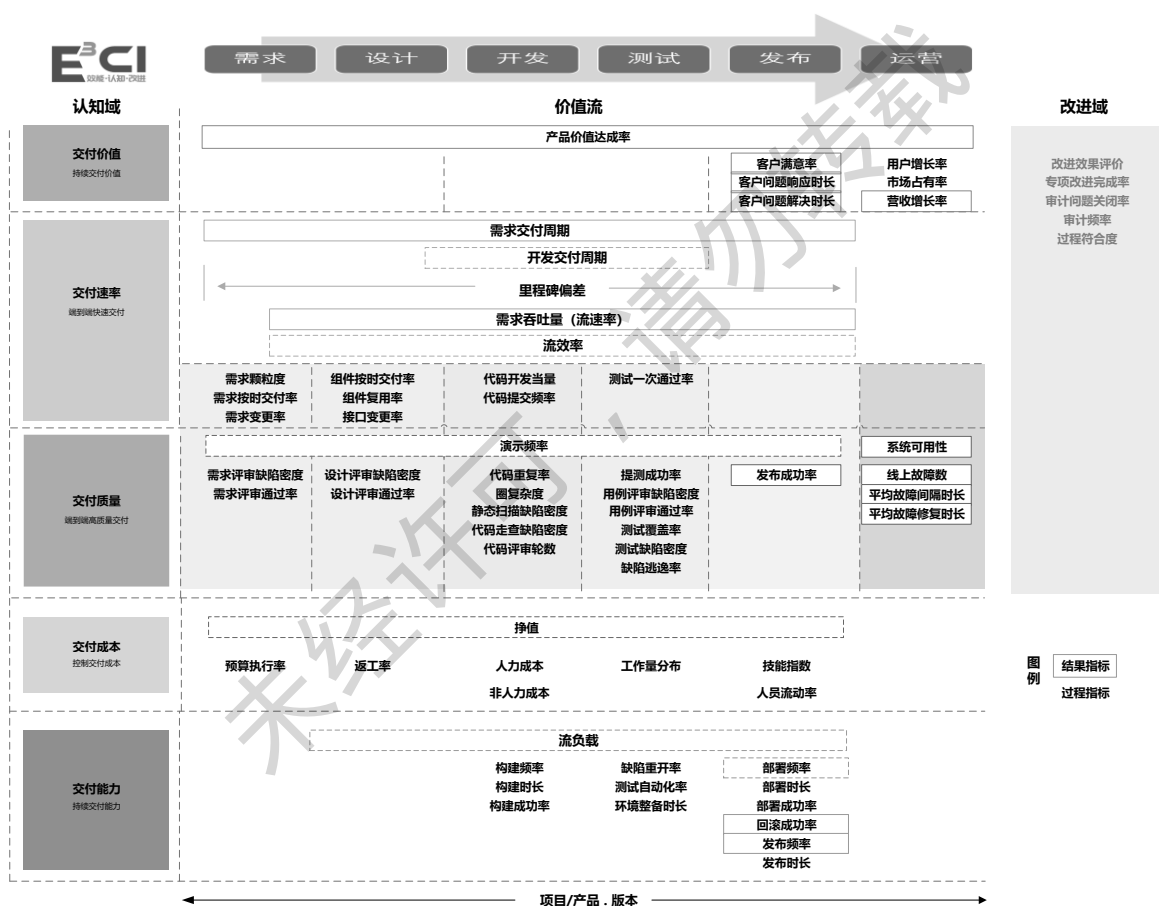


图2 软件研发效能度量框架展开图

#### 4.3.3 实施度量和改进

软件组织在确定度量指标和范围后，进入度量实施阶段，并可针对性地进行改进。

根据指标的数据来源、适用范围、适宜频率和计算方法，软件组织应进行数据收集、数据质量校验、数据分析，并合理展示度量指标。相对于人工输入，度量数据宜更多采用代码或工具平台中存留的数据，使之更具客观性，提升数据质量并降低度量成本。

度量指标分析可采用基本的分析图表，如：散点图、直方图、箱线图、饼状图、柱状图、趋势图、雷达图、累积图等；也可进一步采用统计技术，如描述统计、相关分析、方差分析、回归分析、控制图、

能力分析、假设检验等。

本规范对基本度量指标进行了定义和说明，筛选并包括了影响软件研发的主要因子。软件组织应根据自身所处发展阶段、现状和约束，针对性地选择度量指标，应用基本或高级的数据分析技术，进行数据分析和挖掘，以满足基本分析决策，或通过建模仿真和优化以实现量化的业务目标。

#### 4.4 度量原则

软件研发效能度量应遵循适用性、系统性、可靠性和持续性原则。

##### 4.4.1 适用性

软件研发效能度量应根据软件研发组织的目标和现状选择度量的范围和方法。适用性包括两个方面：一是软件研发组织应根据改进的目标，选择适用的度量范围和指标，而不应在目标不明确的情况下，盲目过度实施度量；二是软件研发组织应根据软件交付模式、软件周期模型、软件组织特点等，选择适用的度量范围和指标。

##### 4.4.2 系统性

软件研发效能度量应根据度量的目标，选取全面反映认知并且相互关联的指标。系统性包括两个方面：一是指标能够反映度量目标的主要方面，满足认知或改进需要；二是指标之间应能够相互印证或制约。度量的系统性可有效防止研发组织面向单一指标优化，造成局部最优而整体不优，从而影响研发效能整体提升。

##### 4.4.3 可靠性

软件研发效能度量的可靠性源于数据的及时性、真实性和客观性。可靠性包括三个方面：一是数据是否及时采集、及时分析，并支持及时采取措施；二是数据是否是真实数据，人工填报数据，往往真实性会打折扣；三是数据的客观性是否可通过多视角相互校验和排查予以保证。

##### 4.4.4 持续性

研发效能改进通常无法一蹴而就，宜基于系统视角、采用迭代的方式组合实施。每个迭代应聚焦少数重点目标，通过根因分析寻找并确定改进方案、实施专项改进、度量和验证改进效果，并沉淀最佳实践。如果改进效果达到预期，下一轮迭代可选取其他目标；如果改进效果未达预期，应改度量以反映更本质的问题，并支持下一轮迭代的目标设定和改进。

### 5 软件研发效能度量指标

#### 5.1 度量指标模型

##### 5.1.1 数据集

数据集是来自不同数据源的数据集合，完整的数据集合亦称为数据湖。

##### 5.1.2 指标

指标是用于解决研发实践问题、获得研发效能认知、产生研发效能价值而定义的可量化的概念和相应的计算方法。一个指标的数值可依赖其他指标的数值进行计算，依赖其他指标的指标称为衍生指标，无依赖的指标称为基础指标。

##### 5.1.3 度量

度量是将指标应用于特定研发数据集的过程。度量计算出的数值结果，称为度量值。

对于研发数据集  $D$ ，指标的度量  $m(D)$  为

$$m(D) = f(D, M(D)),$$

其中,  $D$  为数据集,  $M$  为其依赖的其他度量值的集合,  $f$  为指标的计算方法;  
对应基础指标的  $M$  为空集。

#### 5.1.4 视图

视图是基于研发数据集, 将指标的度量值应用于实际场景的展示。

#### 5.1.5 模型定义

软件研发效能度量指标模型是基于研发数据集、选择指标、计算并展示结果, 支持分析和决策的最佳实践范式。

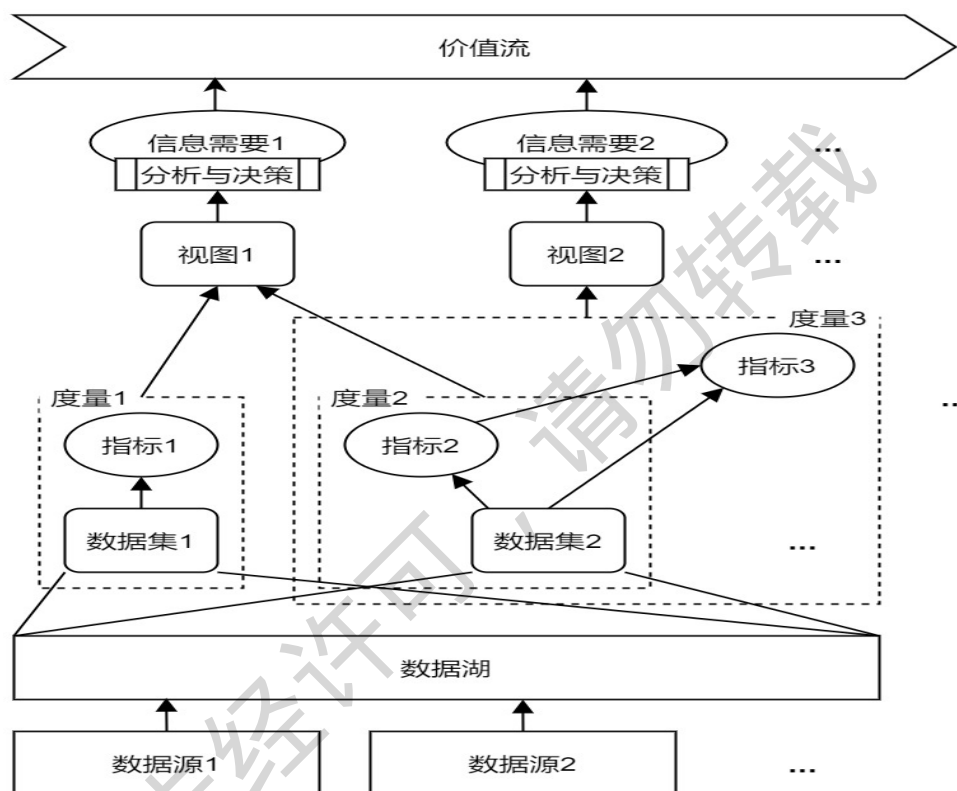


图3 研发效能度量指标模型

端到端的研发价值流（value stream）是效能度量的基础和出发点。基于价值流认知，用户提出不同的信息需要，从而选择指标和实施度量，并使用视图整合强相关的度量值进行分析与决策，以促进系统思考、平衡和决策。

#### 5.1.6 指标定义

一个度量指标的定义和说明通常包括如下部分：

表1 度量指标定义

指标名称	指标概念的简要名称。起名应遵从言简意赅、避免歧义、尊重习惯等原则。
概念描述	对指标含义、目的和作用的描述。
计算方法	指标度量值如何进行计算。
计量单位	指标度量值计算时需要的计量。

分类	度量主要反映的认知域或改进域，包括交付价值、交付速率、交付质量、交付成本、交付能力和持续改进；以及指标涉及的典型生命周期阶段，包括需求、设计、开发、测试、发布、运营中的一个或多个等。
补充说明	需要补充说明部分，利于对指标的深入理解和应用，可包含数据展现图表示例，以及在实施中应注意或考虑的问题等。

## 5.2 度量指标集

度量指标集是基于 E<sup>3</sup>CI 研发效能度量框架，按认知域和改进域汇总的度量指标集合，包括指标名称、基本定义和应用说明等基本信息。软件组织可根据需要，快速查找适用的度量指标，特别是基于系统思维，选择多个相关或约束的度量指标，以利于综合分析和支持决策。选择的度量指标的定义和说明，可通过查阅附件 A 中相应的指标定义获取以供参考。

表 2 度量指标集

认知域/ 改进域	指标名称	单位	基本定义	应用说明
交付 价值	产品价值 达成率	%	产品对用户的需求的满足度。	指标反映了是否在用户规定的时间内完整、正确的完成了用户提出的需求。
	客户满意率	%	产品交付满足客户/用户需要和期望的程度。	指标反映了客户/用户的综合体验，通常是关键业务指标之一。
	客户问题 响应时长	时长	客户问题响应时间与客户问题提出时间之差。	指标反映了响应客户的问题的及时性，是 SLA 的关键指标，也是客户体验的关键指标之一。
	客户问题 解决时长	时长	客户问题解决时间与客户问题首次响应时间之差。	指标反映了解决客户问题的及时性，是 SLA 的关键指标，也是客户体验的关键指标之一。
	用户增长率	%	统计范围内，增加的用户数量相对于用户总量的比例。	指标反映了企业的产品能力和运营能力。
	市场占有率	%	统计范围内，企业销售或运行的产品台套数占市场整体的比例。	指标反映了企业在市场上的产品竞争力和销售能力。
	营收增长率	%	统计范围内，增加的营收相对于总营收的比例。	指标反映了企业销售和经营的能力。
交付 速率	需求 交付周期	天	从需求创建到发布的时长。	反映整个团队（包括业务、产品、设计、开发、测试、运营等各职能）针对客户问题或业务机会的交付速率，也是团队协作能力的体现。
	开发 交付周期	天	需求被研发团队确认，到完成开发、测试、业务验收，达到可上线状态的时长	指标反映了研发团队的交付速率，依赖需求拆分和版本规划以及研发、测试和业务团队的分工协作。

	里程碑偏差	天	在统计范围内，里程碑实际完成时间与计划完成时间之差。	指标反映了项目进度以及里程碑的计划和控制能力。
	需求吞吐量 (流速率)	个	在统计范围内完成交付的需求数量。	指标反映了需求交付的产能或产量。
	流效率	%	研发活动时间占研发活动时间与等待时间之和的比例。	指标反映了研发活动协同流转的效率，通过持续改进，可减少等待、提高效率，是关键改进指标之一。
	需求颗粒度	功能点/故事点	单个需求的大小或规模。	指标反映了需求进行设计开发的复杂度。通过需求颗粒度的估算和核算，逐步提高对需求粒度的把握，有利于提高计划可行性、资源利用率、交付速率等。
	需求按时交付率	%	在统计范围内，需求按时交付的数量占交付需求总数的比例。	指标反映了需求按期交付能力，也一定程度反映了计划能力。
	需求变更率	%	在统计范围内，需求确认后发生变更的数量占已确认需求总数的比例。	指标反映了需求理解及变更控制能力，经确认后发生的需求变更尽管难以完全避免，但其变更对项目进度、质量、成本等往往带来不利影响。
	组件按时交付率	%	在统计范围内，组件按时交付的数量占交付组件总数的比例。	指标反映了组件按期交付能力，也一定程度反映了计划能力。
	组件复用率	%	在统计范围内，复用组件的数量占组件总数的比例。	指标反映了组件的功能、性能、质量和可扩展性，也一定程度上反映了设计能力。
	接口变更率	%	在统计范围内，接口确认后发生变更的比例。	指标反映了接口设计及变更控制能力。约定的接口如果发生变更通常会对其其他组件产生影响，并带来额外的成本或返工。
	代码开发当量	#	对每次代码提交所做的逻辑修改的量化。	指标反映了代码修改及其逻辑复杂度，不受源代码级噪音（如代码风格、重复、自动生成、字面修改等）的影响。
	代码提交频率	次/时长	在单位时间内，代码版本控制系统记录的代码变更次数。	指标反映了代码开发的活跃度，同时可用于提倡小步提交。
	测试一次通过率	%	在统计范围内，由开发提交测试的版本一次性通过测试的比例。	指标反映了开发质量，测试一次通过率的提高有利于减少返工和提高交付速率。
交付质量	演示频率	次/时长或版本	单位时间或版本演示的次数。	指标反映了与需求方进行持续沟通并获取反馈的程度，有助于拥抱变化，降低风险。
	需求评审缺陷密度	个/需求颗粒度	在需求评审中发现的单位需求规模的缺陷数。	指标反映了需求评审的效果。

需求评审 通过率	%	在统计范围内，评审通过的需求数占提交评审的需求总数的比例。	指标反映了需求定义和需求理解的质量，利于减少需求返工和提高交付速率。
设计评审 缺陷密度	个/设计规模	在设计评审中发现的单位设计规模的缺陷数。	指标反映了设计评审的效果。
设计评审 通过率	%	在统计范围内，评审通过的组件数占提交设计评审的组件总数的比例。	指标反映了组件及接口定义和设计的质量，利于减少返工和提高交付速率。
代码重复率	%	在统计范围内，重复的代码量占代码总量的比例。	指标反映了代码的可维护性。重复代码宜进行抽象和复用，以利于减少缺陷，更快地进行交付。
圈复杂度	#	软件代码中独立线性路径的数量。	指标反映了系统中控制或信息流的复杂度，有利于降低测试和维护的难度。
静态扫描 缺陷密度	个/代码规模	静态扫描发现的单位代码规模的缺陷数。	指标反映了编程规范的符合度和代码质量。
代码走查 缺陷密度	个/代码规模	代码走查发现的单位代码规模的缺陷数。	指标反映了代码质量及走查效果，有利于及早发现缺陷，减少返工。
代码 评审轮数	#	一次代码合并或提测经过的代码评审的轮数。	指标反映了代码合并或提测的质量。控制代码评审轮数有利于提高代码开发者的责任心和质量意识。
提测成功率	%	在统计范围内，由开发提交的满足测试准入条件的版本的版本的比例。	指标反映了开发质量和开发过程的规范性，提高提测成功率高利于减少返工、减少测试周期和提高交付速率。
用例评审 缺陷密度	个/用例数	在统计范围内，测试用例评审发现的缺陷与测试用例数的比例。	指标反映了测试用例的质量，可按缺陷类型、严重程度进行统计，宜注意和软件缺陷进行区分。
用例评审 通过率	%	在统计范围内，评审通过的测试用例数占提交评审的测试用例总数的比例。	指标反映了编写测试用例和理解需求的质量，有利于减少返工和提高交付速率。
测试覆盖率	%	被测试覆盖的条目占总的需要测试条目的比例。	指标反映了测试对需求或代码的覆盖程度，利于保证测试质量。条目如果是代码行，就是测试代码覆盖率；条目如果是需求，就是测试需求覆盖率。
测试 缺陷密度	个/代码规模	测试时发现的单位代码规模的缺陷数。	指标反映了测试的结果、用于评估开发的质量。

	缺陷逃逸率	%	统计范围内，线上发现的缺陷数量与检出缺陷总数量的比值。	指标反映了缺陷检出的效果，利于需求、设计、开发、测试团队进行针对性改进。
	发布成功率	%	在统计范围内，成功发布次数占发布总次数的比例。	指标反映了发布过程和脚本质量及生产环境的一致性、稳定性。
	系统可用性	%	在统计范围内，系统可用时长占系统全部运行时间的比例。	指标反映了系统的质量和可靠性，有利于衡量设计、开发和运维的质量。
	线上故障数	#	系统运行时出现的故障数量。	指标反映了系统质量水平。
	平均故障间隔时长	时长	在正常运行过程中，系统两次相邻故障之间的平均时长。	指标反映了线上系统的可靠性，有助于预测系统在下次计划外故障发生之前可以运行多长时间。
	平均故障修复时长	时长	修复系统故障并将其恢复到完整功能所需的平均时长。	指标反映了线上故障恢复能力，通过缺陷快速修复或回滚降低平均故障修复时长是改进的重点之一。
交付成本	挣值	分值	已完成工作的成本和预算的比值，基于组件的WBS。	指标反映了成本策划和控制能力，利于及早识别成本超支和进行调整。
	预算执行率	%	在统计范围内，实际支出资金与实际到位资金的百分比	指标反映了项目资金使用效率和进程。
	返工率	%	花费在返工上的工作量占总工作量的比例。	指标反映了由于缺陷修复、需求理解偏差等原因而产生的额外工作量投入。
	人力成本	货币	软件研发各活动的工作量投入按人员费率折算的总成本。	指标反映了软件研发活动的人力投入。
	非人力成本	货币	直接用于项目的设备、培训、差旅等及分摊费用的总和。	指标反映了软件研发活动的非人力投入。
	工作量分布	%	工作量按选定维度的统计值和分布比例。	指标反映了企业交付过程中不同维度下的工作投入比例，可支持优化不同维度的工作分配。
	技能指数	分值	对员工技能的分值。	指标反映了员工的知识、经验、能力等级，对人员工作分配和成本有显著影响。
	人员流动率	%	统计范围内，离职、调入或调出员工与员工总数的百分比。	该指标反映企业组织与员工队伍的稳定性。除了整体计算，宜对骨干员工和新员工进行分类统计。
交付能力	流负载	#	研发过程中正处于设计、开发、测试、发布中的需求数量。	指标反映了设计、开发、测试、发布各环节的平衡状态，分析流负载利于识别瓶颈和积压。

	构建频率	次/时长	单位时间内软件进行统一编译构建的次数。	指标反映了持续构建的能力，有助于及早发现和修复缺陷。
	构建时长	时长	软件单次统一编译构建的时长。	指标反映了系统复杂度和基础设施等状况，过长的时间应考虑编译构建的优化。
	构建成功率	%	在统计范围内，统一编译构建成功次数占统一编译构建总次数的百分比。	指标反映了代码质量、编译环境的稳定性。
	缺陷重开率	%	在统计范围内，重新打开的缺陷占总缺陷的比率	指标反映了缺陷修复的效果，重开率高应触发针对性的改进。
	测试自动化率	%	自动化测试用例占测试用例总数的百分比。	指标反映了自动化测试的能力，有助于提高测试效率和快速交付能力。
	环境整備时长	时长	环境整備所需的时长。	指标反映了环境的复杂度和基础设施等状况，过长的时间应考虑环境的优化。
	部署频率	次/时长	在单位时间内，软件进行统一部署的次数。	指标反映了持续部署的能力，有助于及早发现和修复缺陷以及环境的不一致性。
	部署时长	时长	软件单次统一部署的时长。	指标反映了系统复杂度和基础设施等状况，过长的时间应考虑软件部署的优化。
	部署成功率	%	在统计范围内，统一部署成功次数占统一部署总次数的百分比。	指标反映了部署脚本质量和部署环境的一致性、稳定性。
	回滚成功率	%	在统计范围内，回滚成功次数占回滚总次数的百分比。	指标反映了回滚脚本质量和基础设施的能力，回滚成功率是降低持续交付风险的关键指标。
	发布频率	次/时长	在单位时间内，软件发布的次数。	指标反映了软硬件基础设施、交付过程、研发团队质量和效率等多种能力。
	发布时长	时长	软件单次发布的时长。	指标反映了系统复杂度和基础设施等状况，过长的时间应考虑软件发布过程的优化。
持续改进	改进效果评价	%	改进前后选定度量数据的变化比例。	指标反映了专项整改的量化的效果。
	专项改进完成率	%	统计范围内，专项改进项目的完成比例。	指标反映了专项改进在公司级、部门级、产品线的实施情况。
	审计问题关闭率	%	统计范围内，审计发现问题已关闭的比例。	指标反映了及时处理并解决审计所发现问题的能力。
	审计频率	次/时长	在单位时间内，内外部审计的次数。	指标反映了外部合规和内部管控的力度。
	过程符合度	%	统计范围内，体系和成果物审计合规的比例。	指标反映了制度和规范的落实情况。

## 6 软件研发效能视图应用



## 6.1 视图模型

在实际应用中，软件研发效能度量指标通常需要组合使用。使用 E<sup>3</sup>CI 视图模型（如下图所示）将有助于系统地分析度量数据，以支持业务决策，主要步骤如下：

1. 确定使用角色，考虑使用者和汇报对象，可包括管理者、产品负责人、技术/测试负责人、项目负责人等；
2. 确定关注视角，可包括业务、产品、项目、工程等；
3. 确定视图卡片的数量及其关联关系，以利于系统分析；
4. 针对每一个视图卡片，明确其意图、所聚焦的问题、关键指标、相应图表及分析方法；
5. 进行系统分析，以支持业务决策。

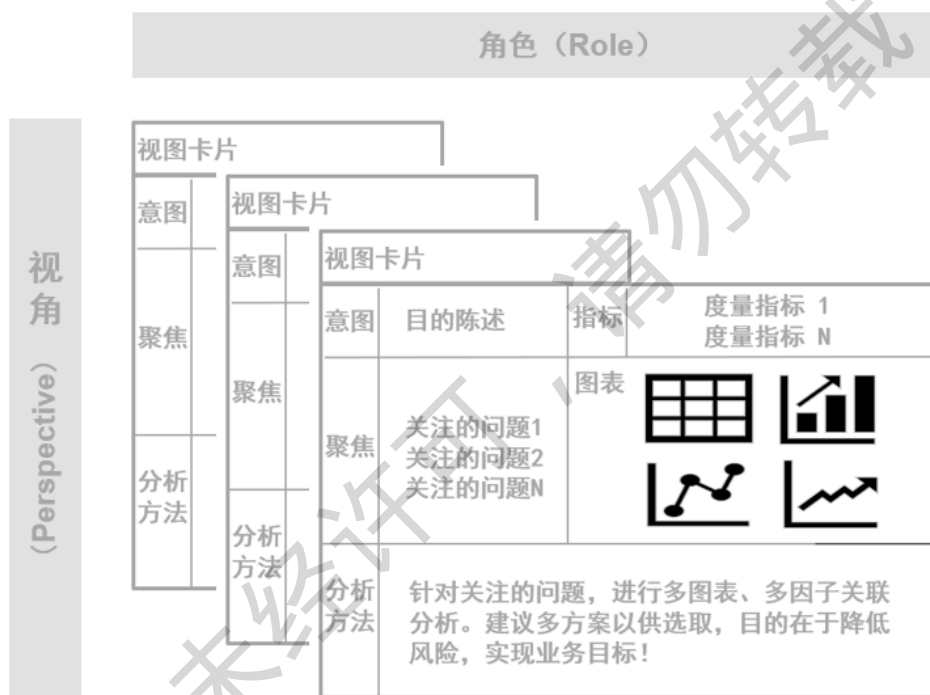


图 4 度量视图模型

## 6.2 视图卡片

视图卡片是视图模型在实践中的具体应用。根据业务实际需要，可将相关度量指标组合成一个或多个视图卡片，通过多维度分析回答遇到的问题，以支持决策和采取措施。相应的视图卡片可嵌入 IT 系统、分析报告或 PPT 中，支持向相关角色汇报。

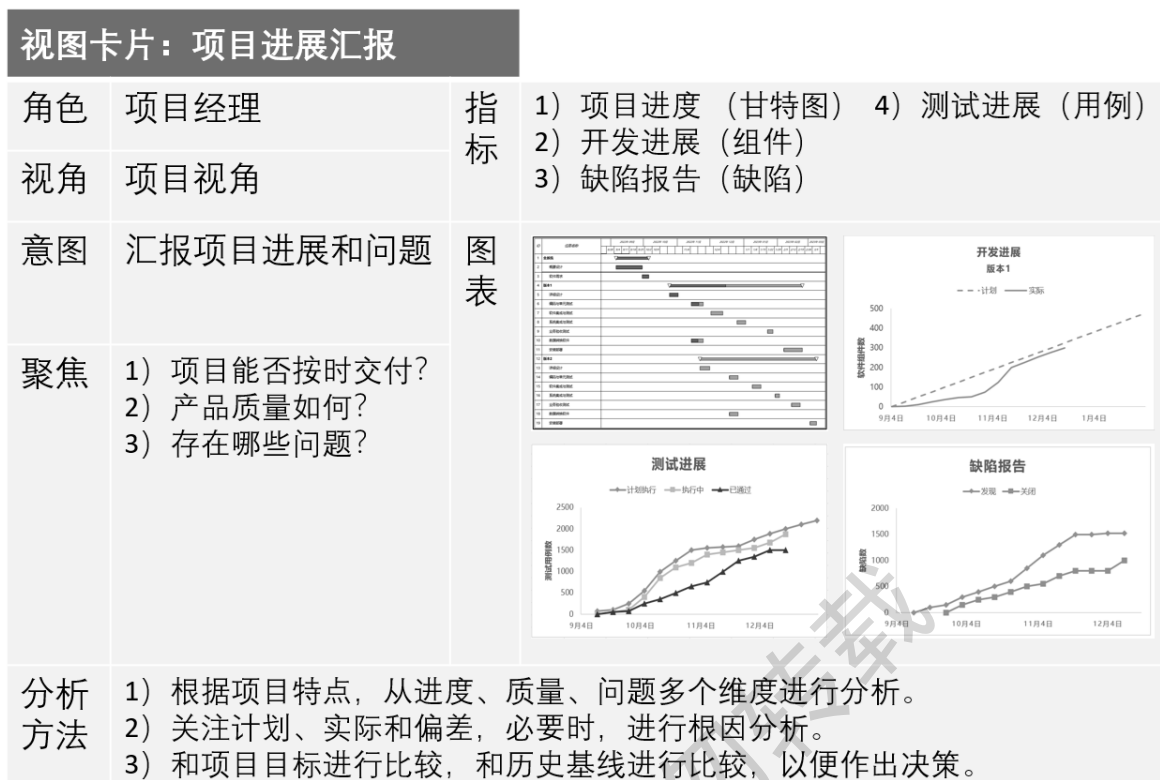


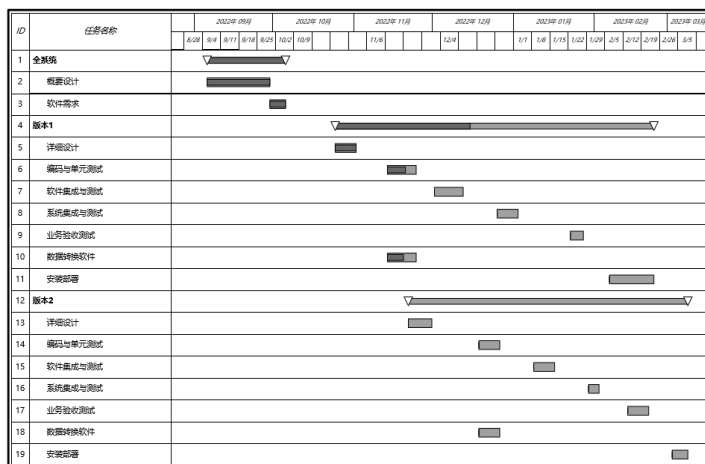
图 5 视图卡片应用示例

以项目经理站在项目视角, 进行项目进展汇报为例, 视图卡片应用示例如上:

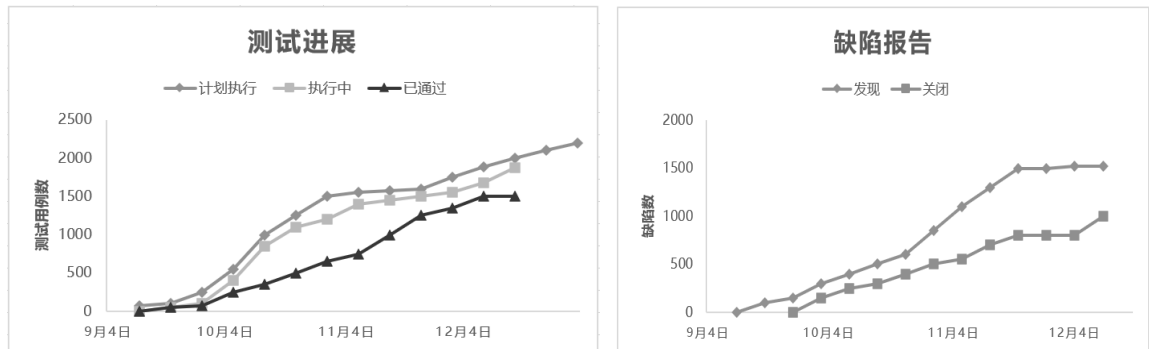
- ◆ 主要意图: 汇报项目进展和问题, 以支持项目级决策和采取纠正措施;
- ◆ 聚焦问题: 1) 项目能否按时交付? 2) 产品质量如何? 3) 存在哪些问题?
- ◆ 主要指标涉及: 1) 项目进度; 2) 开发进展; 3) 测试进展; 4) 缺陷修复进展;
- ◆ 分析方法: 1) 根据项目特点, 从进度、质量、缺陷多维度进行分析; 2) 关注计划、时间和偏差, 必要时, 进行根因分析; 3) 和项目目标进行比较, 和历史基线进行比较, 以便作出决策;

数据分析及结论:

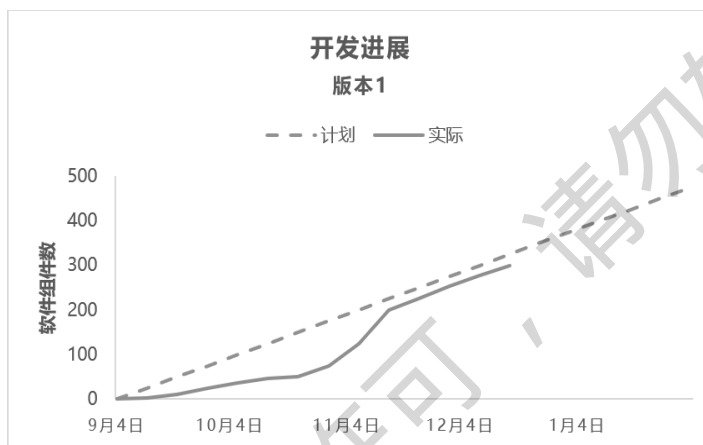
- ◆ 版本 1 进度延迟, 问题主要集中在编码和单元测试, 同时委外定制的数据转换软件进度滞后, 项目进度 (甘特图) 如下图所示:



- ◆ 测试基本按计划执行，但测试通过率低，11月初表现尤为明显；目前缺陷修复速率远低于缺陷检出速率，如测试进展（按用例）报告和缺陷报告所示：



- ◆ 延迟主要集中在组件编码和测试阶段，在11月初表现明显。目前项目组进度赶上来了，但无暇快速修复发现的缺陷，如开发进展（按组件）报告所示。



- ◆ 需要进一步进行根因分析，可能的原因包括计划安排不合理、人员能力不足、缺乏自动化工具支持等，以便针对性地采取纠正措施。

附录 A  
(资料性)  
指标定义

A.1 交付价值

表 A.1.1 产品价值达成率

指标名称	产品价值达成率																														
概念描述	<p>产品价值达成率是产品对用户需求满足的程度。指标反映了是否在用户规定的时间内完整、正确的完成了用户提出的需求。</p> <p>产品价值达成率是由客户或业务代表进行打分的方法进行计算得出。应在策划/需求澄清环节和产品交付/演示环节分别评分。</p>																														
计算方法	<p>产品价值达成率 = <math>\Sigma</math> (产品价值评分*权重) / <math>\Sigma</math> (需求价值评分*权重)</p> <p>其中需求价值评分通常在策划/需求澄清环节, 产品价值评分通常在产品交付/演示环节, 分别由客户或业务打分。</p> <table border="1" data-bbox="392 835 1107 1184"> <thead> <tr> <th>需求概述</th><th>策划/澄清 (1-10)</th><th>交付/演示 (1-10)</th><th>权重</th></tr> </thead> <tbody> <tr> <td>需求 1</td><td>8</td><td>5</td><td>10.0</td></tr> <tr> <td>需求 2</td><td>9</td><td>10</td><td>3.0</td></tr> <tr> <td>...</td><td>...</td><td>...</td><td>...</td></tr> <tr> <td>需求 N</td><td>10</td><td>6</td><td>1.0</td></tr> <tr> <td>加权分值</td><td>117</td><td>86</td><td></td></tr> <tr> <td colspan="2">达成率</td><td>73.5%</td><td></td></tr> </tbody> </table>			需求概述	策划/澄清 (1-10)	交付/演示 (1-10)	权重	需求 1	8	5	10.0	需求 2	9	10	3.0	...	...	...	...	需求 N	10	6	1.0	加权分值	117	86		达成率		73.5%	
需求概述	策划/澄清 (1-10)	交付/演示 (1-10)	权重																												
需求 1	8	5	10.0																												
需求 2	9	10	3.0																												
...	...	...	...																												
需求 N	10	6	1.0																												
加权分值	117	86																													
达成率		73.5%																													
计量单位	%																														
分类	认知域/ 改进域	交付价值																													
	阶段	需求、发布																													

补充说明

随着多版本的持续交付，产品价值达成率指标体现了团队交付的产品或服务的时序变化。如果与目标值/基线值比较，则能突出产品价值的变化，利于业务决策。

**产品价值达成率 vs. 版本**

	版本 1	版本 2	版本 3	版本 4	版本 5	版本 6	版本 7	版本 8	版本 9
产品价值达成率	0.67	0.82	0.93	0.9	0.56	0.77	0.81	0.93	0.87
目标值	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8

坐标轴标题

—— 产品价值达成率    - - - 目标值

产品价值达成率，可以按项目、产品、产品线、部门进行统计和分析，目标值设定可根据实际业务需要设定，其变化趋势有助于支持决策和改进。

表 A.1.2 客户满意率

指标名称	客户满意率																
概念描述	客户满意率是产品交付满足客户/用户需要和期望的程度。指标反映了客户/用户综合体验，通常是关键业务指标之一。																
计算方法	<p>客户满意率 = <math>\Sigma(\text{分项得分} \times \text{权重}) / (\text{分项总分} \times \text{权重})</math></p> <p>客户满意度分值，通常包括很满意、满意、基本满意、不满意，亦可按对应取值或取值范围进行打分，如下表所示：</p> <table border="1" data-bbox="395 1435 1142 1664"> <thead> <tr> <th>定序取值</th><th>对应取值</th><th>取值范围</th></tr> </thead> <tbody> <tr> <td>很满意</td><td>100</td><td>90-99</td></tr> <tr> <td>满意</td><td>80</td><td>80-89</td></tr> <tr> <td>基本满意</td><td>60</td><td>60-79</td></tr> <tr> <td>不满意</td><td>0</td><td>0-59</td></tr> </tbody> </table> <p>客户满意度调查，应设定几个分项，针对各分项应设定相应权重。分项的设立可以按认知域定义，也可以用需求实现、产品质量、进度控制、服务能力、管控能力等更通俗的用词定义。</p>		定序取值	对应取值	取值范围	很满意	100	90-99	满意	80	80-89	基本满意	60	60-79	不满意	0	0-59
定序取值	对应取值	取值范围															
很满意	100	90-99															
满意	80	80-89															
基本满意	60	60-79															
不满意	0	0-59															
计量单位	%																
分类	认知域/ 改进域	交付价值															

	阶段	发布、运营																																																																		
补充说明	<p>为了充分利用客户满意度数据，应收集必要的属性，如所属：部门、产品线、产品、项目、版本、满意度调查时间等，以便进行分析和改进，示例如下：</p> <p>示例 1：客户满意率 vs. 交付版本，展现了随着各个版本的交付，客户满意度调查的结果的变化。</p> <div><h3>客户满意率 vs. 交付版本</h3><table><tr><th>交付版本</th><th>V1.0</th><th>V1.1</th><th>V1.2</th><th>V1.3</th><th>V2.0</th><th>V2.1</th><th>V2.2</th><th>V2.3</th><th>V3.0</th><th>V3.1</th><th>V3.2</th></tr><tr><td>客户满意率</td><td>81%</td><td>56%</td><td>81%</td><td>60%</td><td>90%</td><td>55%</td><td>77%</td><td>81%</td><td>84%</td><td>85%</td><td>75%</td></tr></table></div> <p>示例 2：客户满意率 vs. 交付质量得分。使用时序分析，针对总分（客户满意率）和分项（如：交付质量得分），进行比较，显示交付质量得分的变化对客户满意率的影响。</p> <div><h3>客户满意率 vs. 交付质量</h3><table><tr><th>交付质量</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th></tr><tr><td>交付质量</td><td>80</td><td>80</td><td>100</td><td>60</td><td>100</td><td>60</td><td>80</td><td>80</td><td>80</td><td>100</td><td>80</td><td></td><td>1</td></tr><tr><td>客户满意率</td><td>81%</td><td>56%</td><td>81%</td><td>60%</td><td>90%</td><td>55%</td><td>77%</td><td>81%</td><td>84%</td><td>85%</td><td>75%</td><td></td><td></td></tr></table></div> <p>在客户满意度调查中，应遵循及时有效原则，如：在发布后及时推送并获取客户满意度反馈。</p>		交付版本	V1.0	V1.1	V1.2	V1.3	V2.0	V2.1	V2.2	V2.3	V3.0	V3.1	V3.2	客户满意率	81%	56%	81%	60%	90%	55%	77%	81%	84%	85%	75%	交付质量	1	2	3	4	5	6	7	8	9	10	11	12	13	交付质量	80	80	100	60	100	60	80	80	80	100	80		1	客户满意率	81%	56%	81%	60%	90%	55%	77%	81%	84%	85%	75%		
	交付版本	V1.0	V1.1	V1.2	V1.3	V2.0	V2.1	V2.2	V2.3	V3.0	V3.1	V3.2																																																								
	客户满意率	81%	56%	81%	60%	90%	55%	77%	81%	84%	85%	75%																																																								
	交付质量	1	2	3	4	5	6	7	8	9	10	11	12	13																																																						
	交付质量	80	80	100	60	100	60	80	80	80	100	80		1																																																						
客户满意率	81%	56%	81%	60%	90%	55%	77%	81%	84%	85%	75%																																																									

表 A. 1. 3 客户问题响应时长

指标名称	客户问题响应时长
概念描述	客户问题响应时长是客户问题响应时间与客户问题提出时间之差。指标反映了响应客户的问题的及时性，是 SLA 的关键指标，也是客户体验的关键指标之一。
计算方法	<p>客户问题响应时长 = 客户问题响应时间 - 客户问题提出时间</p> <p>其中，客户问题响应时间和客户问题提出时间应根据企业业务实际情况和客户服务方式自行定义。</p>
计量单位	时长

分类	认知域/ 改进域	交付价值																																							
	阶段	需求、测试、发布、运营																																							
补充说明	<p>客户问题响应时长，宜进一步按关注的维度进行统计，并和目标值进行对比，以利于及时进行分析并采取纠正措施。客户问题响应时长的实际值与目标值或标准响应时长的对比分析，有利于促成快速响应的针对性改进。</p> <p>常用的关注维度包括：1）时间维度，如：天、周、月等；2）问题维度，如：按需求、缺陷、变更等；3）级别维度，如：高、中、低等级；4）组织维度，如：项目、产品线、部门等。</p> <p style="text-align: center;"><b>平均响应时长（小时）</b></p> <table border="1"> <caption>平均响应时长数据表</caption> <thead> <tr> <th>周次</th> <th>平均响应时间 (小时)</th> <th>目标值 (小时)</th> </tr> </thead> <tbody> <tr><td>W1</td><td>12</td><td>24</td></tr> <tr><td>W2</td><td>10</td><td>24</td></tr> <tr><td>W3</td><td>28</td><td>24</td></tr> <tr><td>W4</td><td>31</td><td>24</td></tr> <tr><td>W5</td><td>20</td><td>24</td></tr> <tr><td>W6</td><td>10</td><td>24</td></tr> <tr><td>W7</td><td>10</td><td>24</td></tr> <tr><td>W8</td><td>15</td><td>24</td></tr> <tr><td>W9</td><td>18</td><td>24</td></tr> <tr><td>W10</td><td>10</td><td>24</td></tr> <tr><td>W11</td><td>8</td><td>24</td></tr> <tr><td>W12</td><td>11</td><td>24</td></tr> </tbody> </table> <p>客户问题提出时间和客户问题的响应时间应关联相应的关注维度，宜以 IT 系统支持。</p>		周次	平均响应时间 (小时)	目标值 (小时)	W1	12	24	W2	10	24	W3	28	24	W4	31	24	W5	20	24	W6	10	24	W7	10	24	W8	15	24	W9	18	24	W10	10	24	W11	8	24	W12	11	24
周次	平均响应时间 (小时)	目标值 (小时)																																							
W1	12	24																																							
W2	10	24																																							
W3	28	24																																							
W4	31	24																																							
W5	20	24																																							
W6	10	24																																							
W7	10	24																																							
W8	15	24																																							
W9	18	24																																							
W10	10	24																																							
W11	8	24																																							
W12	11	24																																							

表 A.1.4 客户问题解决时长

指标名称	客户问题解决时长	
概念描述	客户问题解决时长是客户问题解决时间与客户问题首次响应时间之差。指标反映了解决客户问题的及时性，是 SLA 的关键指标，也是客户体验的关键指标之一。	
计算方法	$\text{客户问题解决时长} = \text{客户问题解决时间} - \text{客户问题首次响应时间}$ 客户问题解决时间和客户问题首次响应时间应根据企业业务实际情况和客户服务方式自行定义。	
计量单位	时长	
分类	认知域/ 改进域	交付价值
	阶段	需求、测试、发布、运营

补充说明	<p>客户问题解决时长，可进一步按关注的维度进行统计，并和目标值进行对比，以利于及时进行分析并采取纠正措施。客户问题解决时长的实际值与目标值的对比分析，有利于促成快速响应的针对性改进。</p> <p>常用的关注维度包括：1）时间维度，如：天、周、月等；2）问题维度，如：按需求、缺陷、变更等；3）级别维度，如：高、中、低等级；4）组织维度，如：项目、产品线、部门等。</p>																																							
	<div><div>平均问题解决时长-缺陷（小时）</div><table><thead><tr><th>周次</th><th>平均解决时长 (小时)</th><th>目标值 (小时)</th></tr></thead><tbody><tr><td>W1</td><td>20</td><td>75</td></tr><tr><td>W2</td><td>20</td><td>75</td></tr><tr><td>W3</td><td>55</td><td>75</td></tr><tr><td>W4</td><td>45</td><td>75</td></tr><tr><td>W5</td><td>25</td><td>75</td></tr><tr><td>W6</td><td>55</td><td>75</td></tr><tr><td>W7</td><td>125</td><td>75</td></tr><tr><td>W8</td><td>90</td><td>75</td></tr><tr><td>W9</td><td>65</td><td>75</td></tr><tr><td>W10</td><td>50</td><td>75</td></tr><tr><td>W11</td><td>55</td><td>75</td></tr><tr><td>W12</td><td>65</td><td>75</td></tr></tbody></table></div>	周次	平均解决时长 (小时)	目标值 (小时)	W1	20	75	W2	20	75	W3	55	75	W4	45	75	W5	25	75	W6	55	75	W7	125	75	W8	90	75	W9	65	75	W10	50	75	W11	55	75	W12	65	75
	周次	平均解决时长 (小时)	目标值 (小时)																																					
W1	20	75																																						
W2	20	75																																						
W3	55	75																																						
W4	45	75																																						
W5	25	75																																						
W6	55	75																																						
W7	125	75																																						
W8	90	75																																						
W9	65	75																																						
W10	50	75																																						
W11	55	75																																						
W12	65	75																																						
<p>客户问题首次响应的时间和客户问题的解决时间应关联相应的关注维度，宜以 IT 系统支持。记录客户问题首次响应时间是一个关键数据，通常由于内部流转次数过多而导致反复沟通会对客户体验产生较大影响。</p>																																								

表 A.1.5 用户增长率

指标名称	用户增长率	
概念描述	<p>用户增长率是统计范围内增加的用户数量相对于用户总量的比例。指标反映了企业的产品能力和运营能力。</p> <p>对于特定行业，用户增长率是营收增长的先行指标，可按业务线、产品线进行统计，并与营收增加额进行关联分析。</p>	
计算方法	<p>用户增长率 = 增加的用户数/用户总数</p> <p>增加的用户数可能是负数，以反映用户数量的减少</p>	
计量单位	%	
分类	认知域/ 改进域	交付价值
	阶段	运营



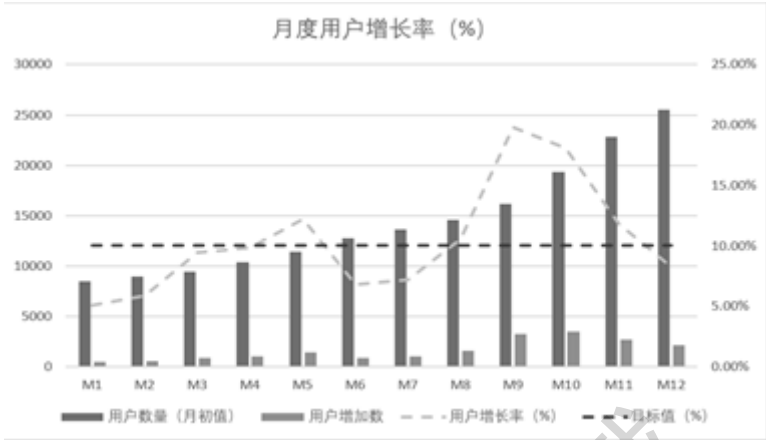
补充说明	<p>按照设定的统计范围，统计增加的用户数量（+/-），计算用户增长率，并和目标值进行对比，分析用户增长趋势，并识别异常原因，有利于支持决策和专项改进，示例如下：</p>  <p>进一步的数据分析，可进行环比和同比分析，并考虑周期性模式（如：季节性/节假日）的影响，以建立对数据的深入理解，并支持业务决策。</p>
------	---

表 A.1.6 市场占有率

指标名称	市场占有率	
概念描述	<p>市场占有率是统计范围内企业销售或运行的产品台套数占市场整体的比例。指标反映了企业在市场上的产品竞争力和销售能力。</p> <p>对于特定行业，市场占有率会有可靠数据来源。在产品的生命周期的不同阶段，由于采用策略不同，市场占有率也会呈现不同的比例。</p>	
计算方法	市场占有率 = 销售或运行的企业产品台套数/市场上销售或运行的同类产品台套总数	
计量单位	%	
分类	认知域/ 改进域	交付价值
	阶段	运营

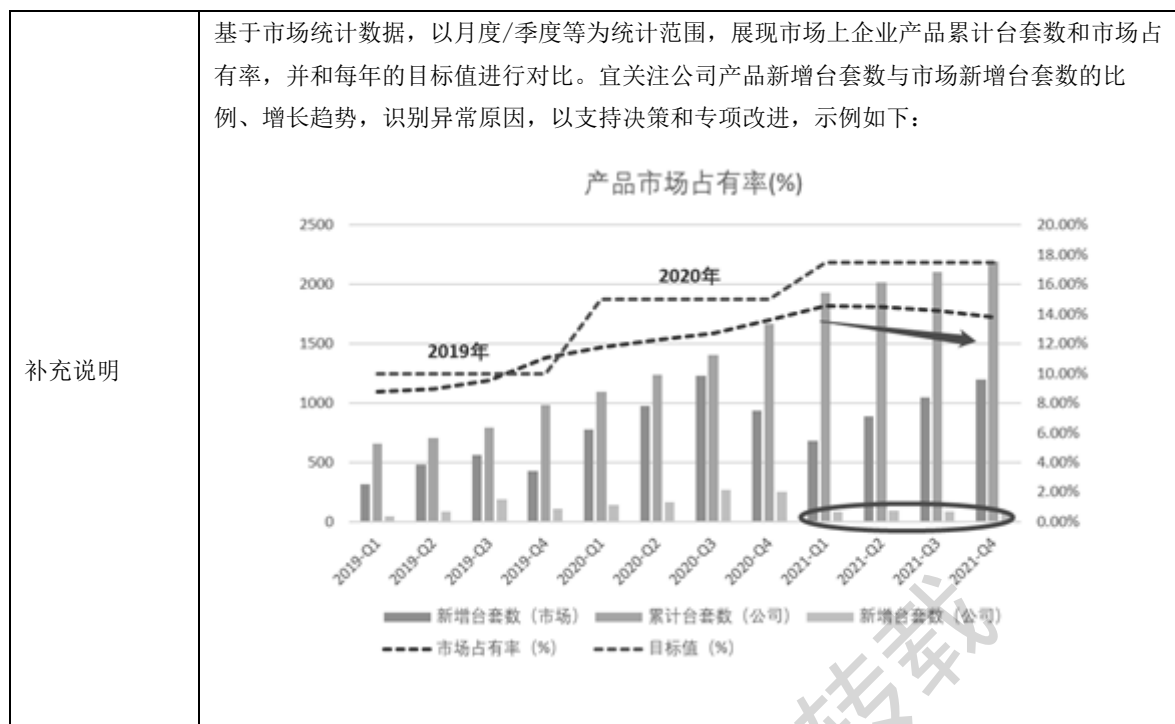


表 A. 1. 7 营收增长率

指标名称	营收增长率	
概念描述	<p>营收增长率是统计范围内增加的营收相对于总营收的比例。指标反映了企业销售和经营的能力。</p> <p>通常按业务线、产品线统计营收增长率，并进行同比和环比分析，以评判签约、回款和利润变化，支持决策分析。</p>	
计算方法	$\text{营收增长率} = \frac{\text{增加营收}}{\text{总营收}}$	
计量单位	%	
分类	认知域/ 改进域	交付价值
	阶段	运营
补充说明	<p>在实际统计分析中，应注意主营业务和增长业务数据的分析和判定，通常可扩展为签约、回款、利润等。尽管有的企业按季度进行考核，但需按月度进行监控，并注意内在模式的影响，如：一季度通常会受春节和规划影响、三季度往往是高峰、四季度往往回落等。部门经理、产品经理应关注业务和产品价值的财务数据，并与研发效能度量及提升相结合。</p>	

## A. 2 交付速率

表 A. 2. 1 需求交付周期

指标名称	需求交付周期
概念描述	需求交付周期是从需求创建到发布的时长。指标反映了整个团队（包括业务、产品、设计、开

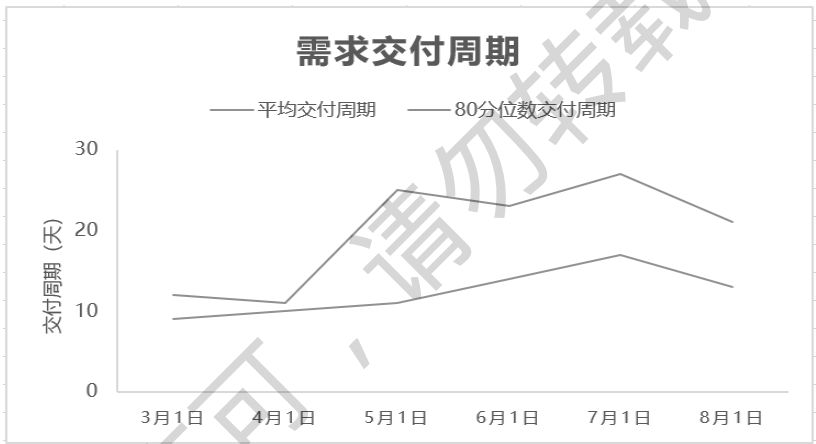
	发、测试、运营等各职能) 针对客户问题或业务机会的交付速率, 也是团队协作能力的体现。可按照需求状态, 统计软件生命周期各阶段的时间占比, 以支持分析和决策。	
计算方法	需求交付周期 = 需求发布时间 - 需求创建时间 可按照需求在各个状态中停留的时长, 将需求交付周期按照软件生命周期各个阶段细分为多个组成部分。企业可根据业务实际情况定义需求交付是指单个需求或批次需求, 也可使用业务需求评审或确认时间代替需求创建时间, 以便获取明确的时间节点。	
计量单位	天	
分类	认知域/ 改进域	交付速率
	阶段	需求、设计、开发、测试、发布
补充说明	<p>下图展示了几个月份中, 需求交付周期的均值和 80%分位值。</p>  <p>需求交付周期反映研发团队的快速响应能力。通常, 向客户交付价值的速率越快越好, 但必须综合考虑交付价值是否满足顾客期望、需求吞吐量和交付质量等其他方面。</p>	

表 A. 2. 2 开发交付周期

指标名称	开发交付周期	
概念描述	开发交付周期是需求被研发团队确认, 到完成开发、测试、业务验收, 达到可上线状态的时长。指标反映了研发团队的交付速率, 依赖需求拆分和版本规划以及研发、测试和业务团队的分工协作。	
计算方法	开发交付周期 = 需求验收时间 - 需求确认时间 需求验收时间是指通过业务验收, 达到可发布状态的时间, 需求确认时间是指研发确认需求并纳入版本计划的时间。	
计量单位	天	
分类	认知域/ 改进域	交付速率

	阶段	开发、测试、发布
补充说明	软件发布通常包括内部发布和外部发布。基于版本规划，根据市场或客户需要选择适宜的內部发布版本对市场或客户发布，也是建立业务弹性和降低风险能力的体现。	

表 A. 2. 3 里程碑偏差

指标名称	里程碑偏差	
概念描述	里程碑偏差是指在统计范围内，里程碑实际完成时间与计划完成时间之差。指标反映了项目进度，以及里程碑的计划和控制能力。	
计算方法	里程碑偏差 = 里程碑实际完成时间 - 里程碑计划完成时间	
计量单位	天	
分类	认知域/ 改进域	交付速率
	阶段	需求、设计、发布
补充说明	亦可使用里程碑偏差率（里程碑偏差/计划周期），反映偏差的比例。	

表 A. 2. 4 需求吞吐量（流速率）

指标名称	需求吞吐量（流速率）	
概念描述	需求吞吐量是在统计范围内交付的需求数量，也称流速率。指标反映了需求交付的产能或产量。如进行横向对比，应考虑需求颗粒度的统一，宜采用功能点、故事点等进行统计。	
计算方法	需求吞吐量 = 统计范围内交付的需求数	
计量单位	个	
分类	认知域/ 改进域	交付速率
	阶段	需求
补充说明	不同周期内需求个数展示需求交付和变化的趋势。例如，下图展示了各个月份交付的需求数量，即每月的需求吞吐量，可用于观察其变化趋势，并进一步进行根因分析。	

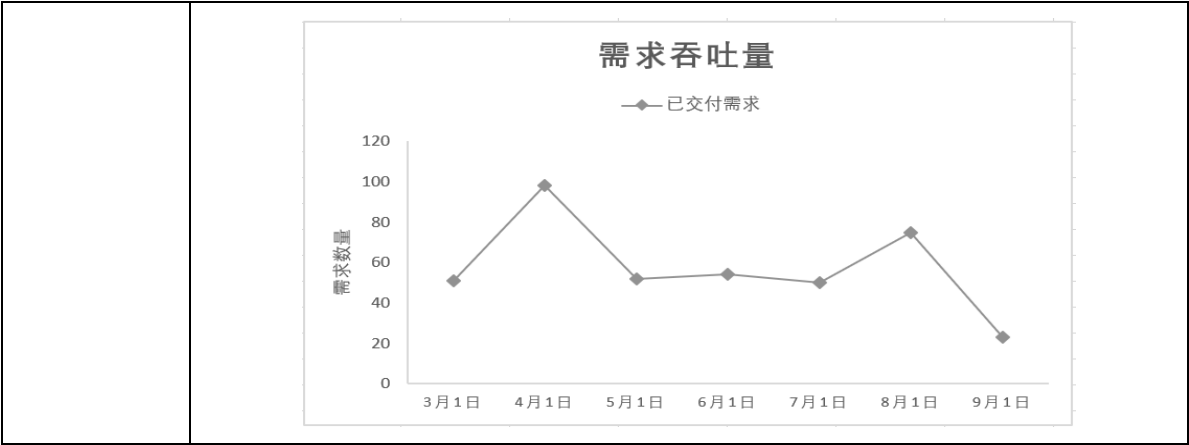


表 A. 2. 5 流效率

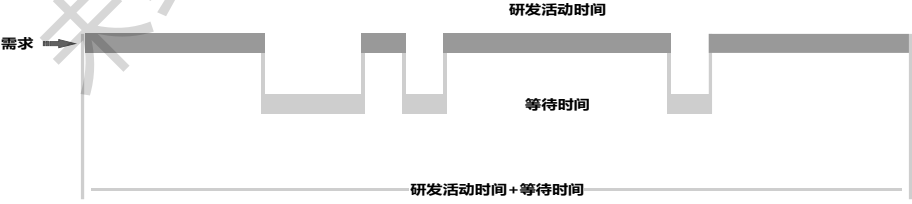
指标名称	流效率	
概念描述	流效率是研发活动时间占研发活动时间与等待时间之和的比例。指标反映了研发活动协同流转的效率，通过持续改进，可减少等待、提高效率，是关键改进指标之一。	
计算方法	流效率 = 研发活动时间 / （研发活动时间 + 等待时间）	
计量单位	%	
分类	认知域/ 改进域	交付速率
	阶段	需求、设计、开发、测试、发布
补充说明	<p>由于流程周转、协作不畅等导致的等待时间过长，一般是流效率低下的主要原因。通过降低等待时间以提升流效率，通常是专项改进的首要改进方向。</p> 	

表 A. 2. 6 需求颗粒度

指标名称	需求颗粒度
概念描述	需求颗粒度即单个需求的大小或规模。指标反映了需求进行设计开发的复杂度。通过需求颗粒度的估算和核算，逐步提高对需求粒度的把握，有利于提高计划可行性、资源利用率、交付速率等。

计算方法	需求颗粒度 = 需求的大小或规模 可提取各个需求的复杂度相关的属性，如故事点、功能点或对应代码提交的当量。																																																	
计量单位	功能点/故事点																																																	
分类	认知域/ 改进域	交付速率																																																
	阶段	需求																																																
补充说明	<p>需求颗粒度可根据企业业务实际情况采用故事点、功能点等进行估算。在需求开发之后，需求颗粒度可以通过其实际数据进行核算，例如采用对应代码的行数、代码当量等。通过估算和核算分析差异，逐步校准团队对需求颗粒度的把握，有利于项目管理中事务的合理安排和顺畅流转，提高资源利用率。</p> <p>下图给出了故事点数和代码当量对比分析的例子，如果故事点数与代码当量的折算率过高或过低，应进行讨论，提升团队对需求颗粒度的预估和把控。</p> <table border="1"> <thead> <tr> <th>需求编号</th><th>故事点数</th><th>代码当量</th></tr> </thead> <tbody> <tr><td>MESS-9917</td><td>8</td><td>1173</td></tr> <tr><td>MESS-8456</td><td>8</td><td>643</td></tr> <tr><td>MESS-9608</td><td>5</td><td>1992</td></tr> <tr><td>MESS-6765</td><td>5</td><td>1149</td></tr> <tr><td>MESS-9807</td><td>5</td><td>793</td></tr> <tr><td>MESS-9812</td><td>5</td><td>494</td></tr> <tr><td>MESS-9820</td><td>5</td><td>232</td></tr> <tr><td>MESS-7099</td><td>5</td><td>182</td></tr> <tr><td>MESS-9110</td><td>3</td><td>803</td></tr> <tr><td>MESS-5726</td><td>3</td><td>398</td></tr> <tr><td>MESS-9733</td><td>3</td><td>356</td></tr> <tr><td>MESS-9086</td><td>3</td><td>315</td></tr> <tr><td>MESS-7124</td><td>3</td><td>280</td></tr> <tr><td>MESS-9640</td><td>3</td><td>246</td></tr> <tr><td>MESS-9104</td><td>3</td><td>232</td></tr> </tbody> </table>		需求编号	故事点数	代码当量	MESS-9917	8	1173	MESS-8456	8	643	MESS-9608	5	1992	MESS-6765	5	1149	MESS-9807	5	793	MESS-9812	5	494	MESS-9820	5	232	MESS-7099	5	182	MESS-9110	3	803	MESS-5726	3	398	MESS-9733	3	356	MESS-9086	3	315	MESS-7124	3	280	MESS-9640	3	246	MESS-9104	3	232
需求编号	故事点数	代码当量																																																
MESS-9917	8	1173																																																
MESS-8456	8	643																																																
MESS-9608	5	1992																																																
MESS-6765	5	1149																																																
MESS-9807	5	793																																																
MESS-9812	5	494																																																
MESS-9820	5	232																																																
MESS-7099	5	182																																																
MESS-9110	3	803																																																
MESS-5726	3	398																																																
MESS-9733	3	356																																																
MESS-9086	3	315																																																
MESS-7124	3	280																																																
MESS-9640	3	246																																																
MESS-9104	3	232																																																

表 A. 2. 7 需求按时交付率

指标名称	需求按时交付率	
概念描述	需求按时交付率是在统计范围内需求按时交付的数量占交付需求总数的比例。指标反映了需求按期交付能力，也一定程度反映了计划能力。	
计算方法	需求按时交付率 = 按时交付的需求数 / 计划交付需求数	
计量单位	%	
分类	认知域/ 改进域	交付速率

	阶段	需求、交付
补充说明	无	

表 A. 2. 8 需求变更率

指标名称	需求变更率	
概念描述	需求变更率是在统计范围内，需求确认后发生变更的数量占已确认需求总数的比例。指标反映了需求理解及变更控制能力，经确认后发生的需求变更尽管难以完全避免，但其变更对项目进度、质量、成本等往往带来不利影响。	
计算方法	需求变更率 = 发生过变更的需求数 / 已确认需求数。 需求变更及其次数可从项目管理软件的记录中获得。	
计量单位	%	
分类	认知域/ 改进域	交付速率
	阶段	需求、设计、开发、测试、发布、运营
补充说明	需求变更率应首先基于客户视角进行统计，计算需求变化的比例，以利于沟通和协调。进一步的统计分析，可考虑需求颗粒度，如功能点、故事点等，和变更的工作量的影响，以利于对标和决策。	

表 A. 2. 9 组件按时交付率

指标名称	组件按时交付率	
概念描述	组件按时交付率是在统计范围内组件按时交付的数量占交付组件总数的比例。指标反映了组件按期交付能力，也一定程度反映了计划能力。	
计算方法	组件按时交付率 = 按时交付的组件数 / 计划交付组件数	
计量单位	%	
分类	认知域/ 改进域	交付速率
	阶段	设计、开发、测试
补充说明	在组件化开发模式下，组件的按期交付对产品集成、测试和发布有显著影响。当基于组件和平台进行定制化开发时，延期交付的组件，可能会迫使定制开发团队重复开发，造成资源浪费、质量低下和维护困难。	

表 A. 2. 10 组件复用率

指标名称	组件复用率	
概念描述	组件复用率是在统计范围内复用组件的数量占组件总数的比例。指标反映了组件的功能、性能、质量和可扩展性，也一定程度上反映了设计能力。组件复用率的提升，将显著提升交付速率和质量。	
计算方法	组件复用率 = 复用的组件数 / 组件总数	
计量单位	%	
分类	认知域/ 改进域	交付速率
	阶段	设计、开发
补充说明	在实际应用中，可进一步定义组件复用度（%），即组件复用程度的百分比，以增加可操作性和支持持续改进。	

表 A. 2. 11 接口变更率

指标名称	接口变更率	
概念描述	接口变更率是在统计范围内，接口确认后发生变更的比例。指标反映了接口设计及变更控制能力。约定的接口如果发生变更，通常会对其他组件产生影响，并带来额外的成本或返工。	
计算方法	接口变更率 = 变更的接口数量/接口总数	
计量单位	%	
分类	认知域/ 改进域	交付速率
	阶段	需求、设计、开发、测试
补充说明	接口变更率可按照变更发生的阶段进行分类统计，以利于分析和控制。 接口定义、确认和变更在开发流程中应有完整的记录。有的代码分析工具可用于统计接口变更率，包括：需要指明接口文件或函数；统计发生了增删改并通过测试的接口并可作为接口总数；进而，统计这些接口定义的代码发生的变更，视为接口定义的变更并计算变更的接口数量。	

表 A. 2. 12 代码开发当量

指标名称	代码开发当量
------	--------



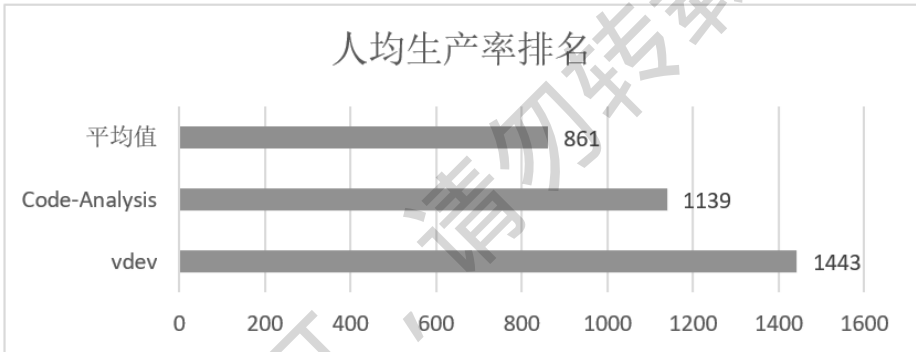
概念描述	代码开发当量是对每次代码提交所做的逻辑修改的量化，简称为代码当量或开发当量。指标反映了代码修改及其逻辑复杂度，不受源代码级噪音（如代码风格、重复、自动生成、字面修改等）的影响。									
计算方法	代码开发当量 = 代码抽象语法树加权最小编辑距离 每次代码提交前后可视为代码的两个切面状态。每个状态的代码都可以通过编译器生成抽象语法树，两个切面对应抽象语法树之间的最小编辑距离，为该次代码提交的当量。可按抽象语法树节点类型和编辑操作类型进行加权，且进一步应用编译技术优化抽象语法树。									
计量单位	#									
分类	认知域/ 改进域	交付速率								
	阶段	开发								
补充说明	<p>例如，下图展示了两个项目人均代码当量及其均值。</p> <div><p>人均生产率排名</p><table><thead><tr><th>项目</th><th>人均生产率</th></tr></thead><tbody><tr><td>平均值</td><td>861</td></tr><tr><td>Code-Analysis</td><td>1139</td></tr><tr><td>vdev</td><td>1443</td></tr></tbody></table></div>		项目	人均生产率	平均值	861	Code-Analysis	1139	vdev	1443
项目	人均生产率									
平均值	861									
Code-Analysis	1139									
vdev	1443									

表 A. 2. 13 代码提交频率

指标名称	代码提交频率	
概念描述	代码提交频率是在单位时间内，代码版本控制系统记录的代码变更次数。指标反映了代码开发的活跃度，同时可用于提倡小步提交。 可按项目、团队（个人）、版本等维度统计提交频率，展示一定时间段（如最近一周或一天）的代码开发活动。	
计算方法	代码提交频率 = 代码提交次数 / 时长	
计量单位	次/时长	
分类	认知域/ 改进域	交付速率
	阶段	开发

补充说明	代码变更宜小步提交，有利于代码评审，并尽早将变更暴露给其他协作的开发者，降低代码合并和缺陷修复成本。
------	--

表 A. 2. 14 测试一次通过率

指标名称	测试一次通过率	
概念描述	测试一次通过率是在统计范围内，由开发提交测试的版本一次性通过测试的比例。该指标反映开发质量，测试一次通过率的提高有利于减少返工和提高交付速率。 如果开发能够通过单元测试或自测等机制，尽量保证提测的质量，有利于减少开发和测试整体的工作量，并提高交付速率。	
计算方法	测试一次通过率 = 一次性通过测试的版本数 / 测试的总版本数	
计量单位	%	
分类	认知域/ 改进域	交付速率
	阶段	开发、测试
补充说明	统计维度可包括产品、项目、团队、周期等。企业宜关注测试一次通过率，避免多次反复提测，浪费资源、延迟交付。	

## A. 3 交付质量

表 A. 3. 1 演示频率

指标名称	演示频率	
概念描述	演示频率是单位时间或版本演示的次数。指标反映了与需求方进行持续沟通并获取反馈的程度，有助于拥抱变化，降低风险。	
计算方法	演示频率 = 演示次数/时长或版本数	
计量单位	次/时长或版本	
分类	认知域/ 改进域	交付质量
	阶段	需求、设计、开发、测试、发布
补充说明	演示包括产品演示、原型演示、样品/样机/样件演示，系统演示/成品演示等，是有效沟通、获取反馈、提升客户体验的有效手段。	

表 A. 3. 2 需求评审缺陷密度

指标名称	需求评审缺陷密度
概念描述	需求评审缺陷密度是在需求评审中发现的单位需求规模的缺陷数。指标反映了需求评审的效

	果。	
计算方法	需求评审缺陷密度 = 需求评审检出缺陷数 / 需求规模	
计量单位	个/需求规模	
分类	认知域/ 改进域	交付质量
	阶段	需求
补充说明	需求规模可按需求数、功能点、故事点等进行统计。需求评审缺陷可按缺陷类别、级别进一步分析，以利于针对性地改进。	

表 A. 3. 3 需求评审通过率

指标名称	需求评审通过率	
概念描述	需求评审通过率是在统计范围内，评审通过的需求数占提交评审的需求总数的比例。指标反映了需求定义和需求理解的质量，利于减少返工和提高交付速率。	
计算方法	需求评审通过率 = 通过评审的需求数 / 提交评审的需求总数	
计量单位	%	
分类	认知域/ 改进域	交付质量
	阶段	需求
补充说明	基于企业业务实际，可进一步计算需求评审的一次通过率，有利于提升需求定义和需求理解的质量。	

表 A. 3. 4 设计评审缺陷密度

指标名称	设计评审缺陷密度	
概念描述	设计评审缺陷密度是在设计评审中发现的单位设计规模的缺陷数。指标反映了设计评审的效果。	
计算方法	设计评审缺陷密度 = 设计评审检出缺陷数 / 设计规模	
计量单位	个/设计规模	
分类	认知域/ 改进域	交付质量
	阶段	设计
补充说明	设计规模可按需求数、功能点、故事点、组件数等进行统计。设计评审缺陷可按缺陷类别、级别进一步分析，以利于针对性地改进。	

表 A. 3. 5 设计评审通过率

指标名称	设计评审通过率	
概念描述	设计评审通过率是在统计范围内，评审通过的组件数占提交评审的组件总数的比例。指标反	

	映了组件及接口定义和设计的质量，利于减少返工和提高交付速率。	
计算方法	设计评审通过率 = 通过评审的组件数 / 提交评审的组件总数	
计量单位	%	
分类	认知域/ 改进域	交付质量
	阶段	设计
补充说明	基于企业业务实际情况，可进一步计算设计评审的一次通过率，有利于提升组件及接口定义和设计的质量。	

表 A.3.6 代码重复率

指标名称	代码重复率	
概念描述	代码重复率是在统计范围内，重复的代码量占代码总量的比例。指标反映了代码的可维护性。重复代码宜进行抽象和复用，以利于减少缺陷，更快地进行交付。表达相同或相似语义的代码称为重复代码。	
计算方法	代码重复率 = 重复代码的代码量/代码的代码总量	
计量单位	%	
分类	认知域/ 改进域	交付速率
	阶段	开发
补充说明	<p>重复代码短期便利了交付，但长期可维护性差，导致修改的工作量成倍增加，甚至导致缺陷。相同或相似逻辑的代码宜进行抽象和复用，方式包括函数、类库或服务。</p> <p>代码重复的判断依赖克隆检测技术，可只在源代码级别按字面判断，也可通过深度代码分析进行语法语义层面的判断。代码量的统计口径可采用代码行数，也可采用代码当量等。</p>	

表 A.3.7 圈复杂度

指标名称	圈复杂度
概念描述	圈复杂度是软件代码中独立线性路径的数量。指标反映了系统中控制或信息流的复杂度，有利于降低测试和维护的难度。
计算方法	<p>方法一，点边算法，计算公式为：  <math>V(G) = E - N + 2</math>            其中，E 表示控制流图中边（Edge）的数量，N 表示控制流图中节点（Node）的数量。</p> <p>方法二：节点判定法，计算公式为：  <math>V(G) = \text{判定节点数} + 1</math>            判定节点指包括条件的节点。            判定节点的数量再加上 1 即控制流图的区域数。</p> <p>方法三：区域数法，计算公式为：</p>

	$V(G)=R$ 其中 R 代表平面被控制流图划分成的区域数。	
计量单位	#	
分类	认知域/ 改进域	分类
	阶段	开发
补充说明	可以用来评估一个系统中控制或信息流的复杂度，很多工具可自动测量。高复杂度通常是高缺陷率的领先指标。具有高复杂性的组件通常需要额外的评审，增加测试或重构。	

表 A. 3. 8 静态扫描缺陷密度

指标名称	静态扫描缺陷密度	
概念描述	静态扫描缺陷密度是静态扫描发现的单位代码规模的缺陷数。指标反映了编程规范的符合度和代码质量。	
计算方法	静态扫描缺陷密度 = 静态扫描检出缺陷数 / 代码规模	
计量单位	个/代码规模	
分类	认知域/ 改进域	交付质量
	阶段	开发、测试
补充说明	静态扫描缺陷可按缺陷类别、级别、趋势进一步分析，并通过改进活动支持代码质量提升。	

表 A. 3. 9 代码走查缺陷密度

指标名称	代码走查缺陷密度	
概念描述	代码走查缺陷密度是代码走查发现的单位代码规模的缺陷数。指标反映了代码质量及走查效果，有利于及早发现缺陷，减少返工。	
计算方法	代码走查缺陷密度 = 代码走查检出的缺陷数 / 代码规模	
计量单位	个/代码规模	
分类	认知域/ 改进域	交付质量
	阶段	开发、测试
补充说明	代码走查主要由团队执行。发现的缺陷可按缺陷类型和严重程度等统计。	

表 A. 3. 10 代码评审轮数

指标名称	代码评审轮数
------	--------

概念描述	代码评审轮数是一次代码合并或提测经过的代码评审的轮数。指标反映了代码合并或提测的质量。控制代码评审轮数有利于提高代码开发者的责任心和质量意识。	
计算方法	代码评审轮数	
计量单位	#	
分类	认知域/ 改进域	交付质量
	阶段	开发
补充说明	一轮评审指评审人根据代码做出接受的判断或者提出需要修改的问题。每次修改、补充提交并再次进行评审，轮数加一，直至代码被合并或拒绝。	

表 A. 3. 11 提测成功率

指标名称	提测成功率	
概念描述	提测成功率是在统计范围内，由开发提交的满足测试准入条件的版本的比例。指标反映了开发质量和开发过程的规范性，提高提测成功率利于减少返工、减少测试周期和提高交付速率。	
计算方法	$\text{提测成功率} = \text{提测成功次数} / \text{提测总次数}$	
计量单位	%	
分类	认知域/ 改进域	交付质量
	阶段	开发、测试
补充说明	无。	

表 A. 3. 12 用例评审缺陷密度

指标名称	用例评审缺陷密度	
概念描述	用例评审缺陷密度是在统计范围内测试用例评审发现的缺陷与测试用例数的比例。指标反映了测试用例的质量，可按缺陷类型、严重程度进行统计，宜注意和软件缺陷进行区分。	
计算方法	$\text{测试用例评审缺陷密度} = \text{测试用例评审检出缺陷数} / \text{测试用例数}$	
计量单位	个/用例数	
分类	认知域/ 改进域	交付质量
	阶段	需求、开发、测试
补充说明	无。	

表 A. 3. 13 用例评审通过率

指标名称	用例评审通过率
------	---------

概念描述	用例评审通过率是在统计范围内，评审通过的测试用例数占提交评审的测试用例总数的比例。指标反映了编写测试用例和理解需求的质量，有利于减少返工和提高交付速率。	
计算方法	用例评审通过率 = 通过评审的测试用例数 / 提交评审的测试用例总数	
计量单位	%	
分类	认知域/ 改进域	交付质量
	阶段	需求、开发、测试
补充说明	基于企业业务实际情况，可进一步计算测试用例评审的一次通过率，有利于提升用例编写和理解需求的质量。	

表 A. 3. 14 测试覆盖率

指标名称	测试覆盖率	
概念描述	测试覆盖率是被测试覆盖的条目占总的需要测试条目的比例。指标反映了测试对需求或代码的覆盖程度，利于保证测试质量。条目如果是代码行，就是测试代码覆盖率；条目如果是需求，就是测试需求覆盖率。	
计算方法	测试覆盖率 = 测试覆盖的条目数量 / 需要测试条目的总数量	
计量单位	%	
分类	认知域/ 改进域	交付质量
	阶段	需求、开发、测试
补充说明	测试覆盖率是指对测试人员完成用例的度量和评估。它通常依据某种覆盖准则来对测试用例执行情况进行衡量，以判断测试执行得是否充分。	

表 A. 3. 15 测试缺陷密度

指标名称	测试缺陷密度	
概念描述	测试缺陷密度是测试时发现的单位代码规模的缺陷数。指标反映了测试的结果、用于评估开发的质量。 通常，测试缺陷可按缺陷类别、级别进一步分析。	
计算方法	测试缺陷密度 = 测试检出缺陷数量 / 代码规模	
计量单位	个/代码规模	
分类	认知域/ 改进域	交付质量
	阶段	测试
补充说明	基于企业业务实际情况，可将测试缺陷密度进一步扩展到组件测试、集成测试、验收测试等。	

表 A. 3. 16 缺陷逃逸率

指标名称	缺陷逃逸率	
概念描述	缺陷逃逸率是统计范围内线上发现的缺陷数量与检出缺陷总数量的比值。指标反映了缺陷检出的效果，利于需求、设计、开发、测试团队进行针对性改进。	
计算方法	缺陷逃逸率 = 线上发现的缺陷数量 / 缺陷总数量	
计量单位	%	
分类	认知域/ 改进域	交付质量
	阶段	需求、设计、开发、测试、发布、运营
补充说明	缺陷总数量包括需求、设计、开发、测试、发布、运营各阶段发现的缺陷总数，宜进行分类统计，并针对性地进行改进。	

表 A. 3. 17 发布成功率

指标名称	发布成功率	
概念描述	发布成功率是在统计范围内成功发布次数占发布总次数的比例。指标反映了发布过程和脚本质量及生产环境的一致性、稳定性。	
计算方法	发布成功率 = 成功发布次数 / 发布总次数	
计量单位	%	
分类	认知域/ 改进域	交付质量
	阶段	发布
补充说明	基于企业业务实际情况，可进一步计算一次发布成功率，有利于提升发布流程和发布脚本的质量和效率。	

表 A. 3. 18 系统可用性

指标名称	系统可用性	
概念描述	系统可用性是在统计范围内系统可用时长占系统全部运行时间的比例。指标反映了系统的质量和可靠性，有利于衡量设计、开发和运维的质量。	
计算方法	系统可用性 = 系统可用时长 / 系统全部运行时长	
计量单位	%	
分类	认知域/ 改进域	交付质量
	阶段	运营
补充说明	无。	



表 A. 3. 19 线上故障数

指标名称	线上故障数	
概念描述	线上故障数是系统运行时出现的故障数量。指标反映了系统质量水平。 线上故障宜按故障类别、级别进一步分析和改进。	
计算方法	线上故障数量	
计量单位	#	
分类	认知域/ 改进域	交付质量
	阶段	运营
补充说明	基于行业特征，线上故障一般分为软件故障与硬件故障等。	

表 A. 3. 20 平均故障间隔时长

指标名称	平均故障间隔时长	
概念描述	平均故障间隔时长（MTBF）是在正常运行过程中，系统两次相邻故障之间的平均时长。指标反映了线上系统的可靠性，有助于预测系统在下一计划外故障发生之前可以运行多长时间。	
计算方法	<p>平均故障间隔时长（MTBF）= <math>\Sigma (T1+T2+T3) / N</math>（故障次数）</p>  <p>MTBF: 平均故障间隔时长= <math>\Sigma (T1+T2+T3) / N</math> (N=故障次数)</p>	
计量单位	时长	
分类	认知域/ 改进域	交付质量
	阶段	运营
补充说明	无。	

表 A. 3. 21 平均故障修复时长

指标名称	平均故障修复时长	
概念描述	平均故障修复时长（MTTR）是修复系统故障并将其恢复到完整功能所需的平均时长。指标反映了线上故障恢复能力，通过缺陷快速修复或回滚降低平均故障修复时长是改进的重点之一。	
计算方法	平均故障恢复时长（MTTR）= $\Sigma (T2+T3) / N$ （故障次数）	

	<p><b>MTBF: 平均故障修复时长 = <math>\sum(T2+T3)/N</math> (N=故障次数)</b></p>	
计量单位	时长	
分类	认知域/ 改进域	交付质量
	阶段	运营
补充说明	<p>平均故障修复时长 (Mean Time To Repair) 指标越低, 故障恢复的越快, 故障持续的时间越短, 意味着服务更稳定。</p> <p>这个指标的提示: 在已经发生故障的情况下, 如何才能让服务更快的恢复?</p>	

#### A. 4 交付成本

表 A. 4. 1 挣值

指标名称	挣值	
概念描述	挣值是已完成工作的成本和预算的比值。指标反映了成本策划和控制能力, 利于及早识别成本超支和进行调整。	
计算方法	<p>CV (Cost Variance) = BCWP-ACWP, 成本差异, 比较实际完成的任务所对应的预算成本和实际完成的任务所对应的实际成本之差。</p> <p>SV (Schedule Variance) = BCWP-BCWS, 进度差异, 比较实际完成的任务所对应的预算成本和计划完成的任务所对应的预算成本之差。</p> <p>CPI (Cost Performance Index) = BCWP/ACWP, 成本绩效指数, 即实际完成的任务所对应的预算成本和实际完成的任务所对应的实际成本之比。</p> <p>SPI (Schedule Performance Index) = BCWP/BCWS, 进度绩效指数, 即实际完成的任务所对应的预算成本和计划完成的任务所对应的预算成本之比。</p> <p>BCWS (Budgeted Cost of Work Scheduled), 即计划完成的任务所对应的预算成本;</p> <p>BCWP (Budgeted Cost of Work Performed), 即实际完成的任务所对应的预算成本, 也称为挣值 (EV: Earned Value);</p> <p>ACWP (Actual Cost of Work Performed), 即实际完成的任务所对应的实际成本。</p>	
计量单位	分值	
分类	认知域/ 改进域	交付成本
	阶段	需求、设计、开发、测试、发布
补充说明	<p>挣值方法主要支持项目绩效管理, 可基于项目和组件的 WBS 进行统计和分析, 其目的在于比较项目实际与计划的差异, 关注的是实际中的各个项目任务, 在进度、质量、成本等方面与计划的差异, 可根据这些差异, 对项目剩余的任务进行预测、调整和控制, 示例如下:</p>	

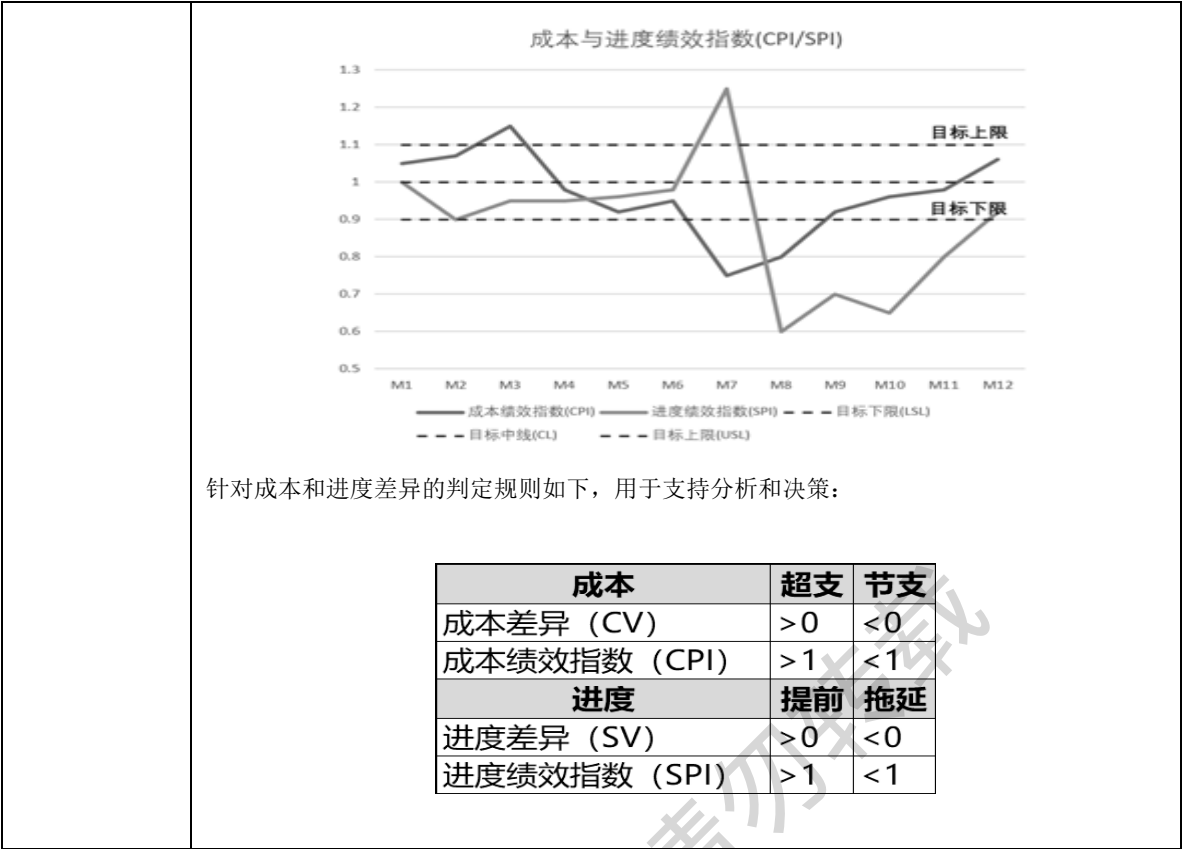


表 A. 4. 2 预算执行率

指标名称	预算执行率	
概念描述	预算执行率是在统计范围内，实际支出资金与实际到位资金的百分比。指标反映了项目资金使用效率和进程。	
计算方法	预算执行率 = 实际支出资金 / 实际到位资金	
计量单位	%	
分类	认知域/ 改进域	交付成本
	阶段	需求、设计、开发、测试、发布、运营
补充说明	实际支出资金指一定时期（本年度或项目期）内项目实际拨付的资金。实际到位资金指一定时期（本年度或项目期）内落实到具体项目的资金。宜在项目计划阶段，建立预算执行计划，并在执行过程中计算预算执行率。预算执行率越接近 100%，显示计划执行越佳；反之，则越差。	

表 A. 4. 3 返工率

指标名称	返工率
概念描述	返工率是花费在返工上的工作量占总工作量的比例。指标反映了由于缺陷修复、需求理解偏差等原因而产生的额外工作量投入。 企业应尽可能减少返工。返工率过高会导致交付周期延长和人力成本的上升。

计算方法	返工率 = 返工工作量/总工作量。	
计量单位	%	
分类	认知域/ 改进域	交付成本
	阶段	测试、发布
补充说明	<p>计算返工率需首先需对返工进行明确定义，通常将在测试和发布阶段针对发现的缺陷的修复定义为返工。</p> <p>通过计算各阶段返工率，有助于发现薄弱环节，如：组件测试和集成测试没有及时发现缺陷而泄露到系统测试阶段才发现，导致返工的显著增加，需针对性地进行改进，示例如下：</p>	
	<p style="text-align: center;"><b>缺陷修复工时(人天) vs. 返工率(%)</b></p> <p style="text-align: center;"><b>返工率(%) vs. 版本</b></p>	

表 A. 4. 4 人力成本

指标名称	人力成本
概念描述	人力成本是软件研发各活动的工作量投入按人员费率折算的总成本。指标反映了软件研发活动的人力投入，取决于不同技能级别的人员费率和工时投入。

	有利于企业管理者从人力成本方面对项目支出进行掌握。																																											
计算方法	人力成本 = $\Sigma$ (工时投入*人员费率)																																											
计量单位	货币																																											
分类	认知域/ 改进域	交付成本																																										
	阶段	需求、设计、开发、测试、发布、运营																																										
补充说明	人力成本宜按产品和组件（或应用）进行统计，可分配到相应的阶段，以便进一步分析和比较，支持项目策划、控制和改进，示例如下：																																											
	<table><tr><th>产品/组件</th><th>需求</th><th>设计</th><th>开发</th><th>测试</th><th>发布</th><th>合计</th></tr><tr><td>产品A</td><td>¥ 240,000.00</td><td>¥ 360,000.00</td><td></td><td>¥ 80,000.00</td><td>¥ 50,000.00</td><td>¥ 730,000.00</td></tr><tr><td>组件1</td><td></td><td></td><td>¥ 120,000.00</td><td>¥ 20,000.00</td><td></td><td>¥ 140,000.00</td></tr><tr><td>组件2</td><td></td><td></td><td>¥ 180,000.00</td><td>¥ 12,000.00</td><td></td><td>¥ 192,000.00</td></tr><tr><td>组件N</td><td></td><td></td><td>¥ 210,000.00</td><td>¥ 21,000.00</td><td></td><td>¥ 231,000.00</td></tr><tr><td>合计</td><td>¥ 240,000.00</td><td>¥ 360,000.00</td><td>¥ 510,000.00</td><td>¥ 133,000.00</td><td>¥ 50,000.00</td><td>¥ 1,293,000.00</td></tr></table>		产品/组件	需求	设计	开发	测试	发布	合计	产品A	¥ 240,000.00	¥ 360,000.00		¥ 80,000.00	¥ 50,000.00	¥ 730,000.00	组件1			¥ 120,000.00	¥ 20,000.00		¥ 140,000.00	组件2			¥ 180,000.00	¥ 12,000.00		¥ 192,000.00	组件N			¥ 210,000.00	¥ 21,000.00		¥ 231,000.00	合计	¥ 240,000.00	¥ 360,000.00	¥ 510,000.00	¥ 133,000.00	¥ 50,000.00	¥ 1,293,000.00
	产品/组件	需求	设计	开发	测试	发布	合计																																					
	产品A	¥ 240,000.00	¥ 360,000.00		¥ 80,000.00	¥ 50,000.00	¥ 730,000.00																																					
	组件1			¥ 120,000.00	¥ 20,000.00		¥ 140,000.00																																					
	组件2			¥ 180,000.00	¥ 12,000.00		¥ 192,000.00																																					
	组件N			¥ 210,000.00	¥ 21,000.00		¥ 231,000.00																																					
	合计	¥ 240,000.00	¥ 360,000.00	¥ 510,000.00	¥ 133,000.00	¥ 50,000.00	¥ 1,293,000.00																																					
组织级通常按人员级别明确相应费率，如：需求工程师：3.5 万元/人月，开发工程师 2.5 万元/人月等，以利于计算人力成本。																																												

表 A. 4. 5 非人力成本

指标名称	非人力成本	
概念描述	非人力成本是直接用于项目的设备、培训、差旅等及分摊费用的总和。指标反映软件研发活动的非人力投入。	
计算方法	非人力成本 = $\Sigma$ (各类支出费用)	
计量单位	货币	
分类	认知域/ 改进域	交付成本
	阶段	无
补充说明	非人力成本一般按项目、产品线、部门等进行核算和分摊。	

表 A. 4. 6 工作量分布

指标名称	工作量分布	
概念描述	工作量分布是工作量按选定维度的统计值和分布比例。该指标反映企业交付过程中不同维度下的工作投入比例，可支持优化不同维度的工作分配。 维度包括生命周期阶段、需求、组件、版本等。	
计算方法	工作量按不维度分段进行统计。	
计量单位	%	
分类	认知域/ 改进域	交付成本
	阶段	需求、设计、开发、测试、发布、运营

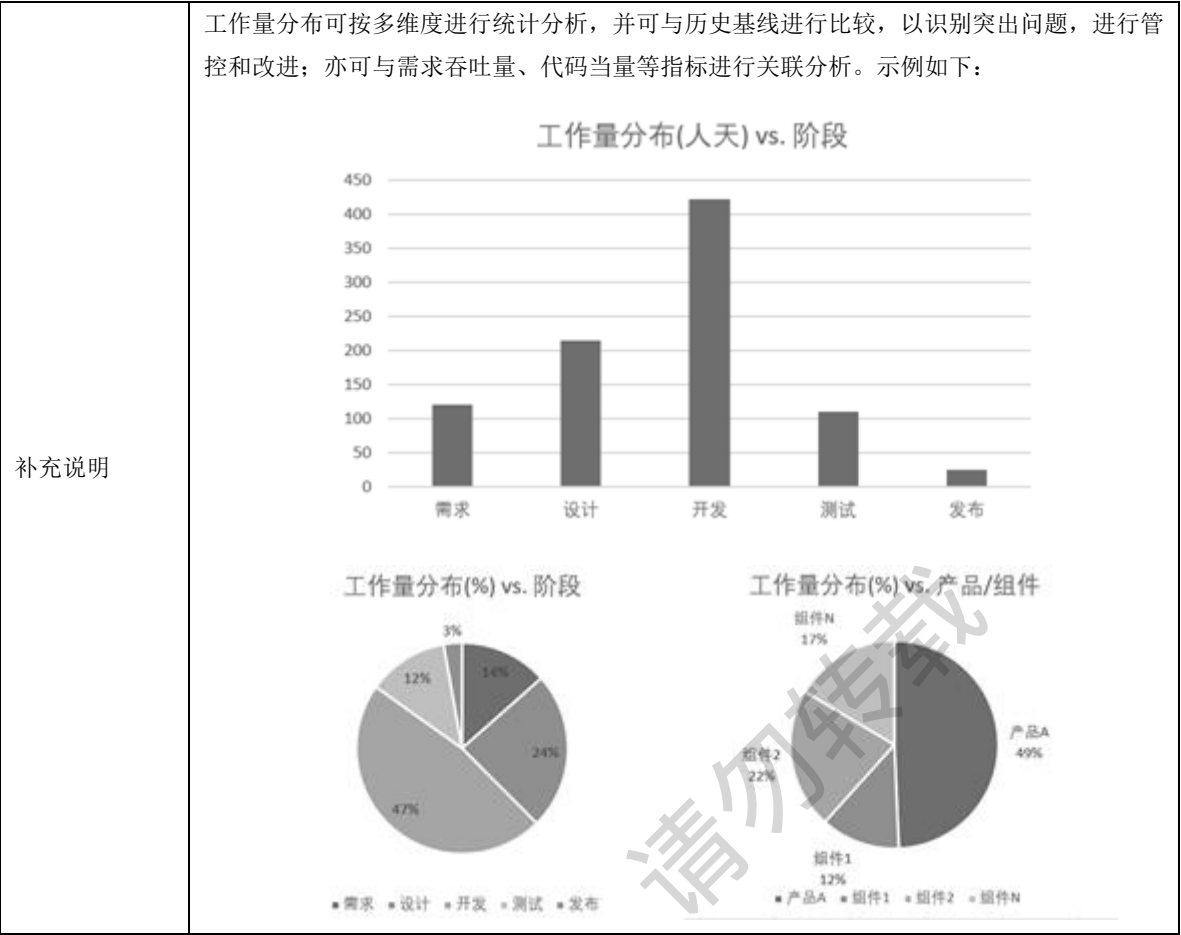
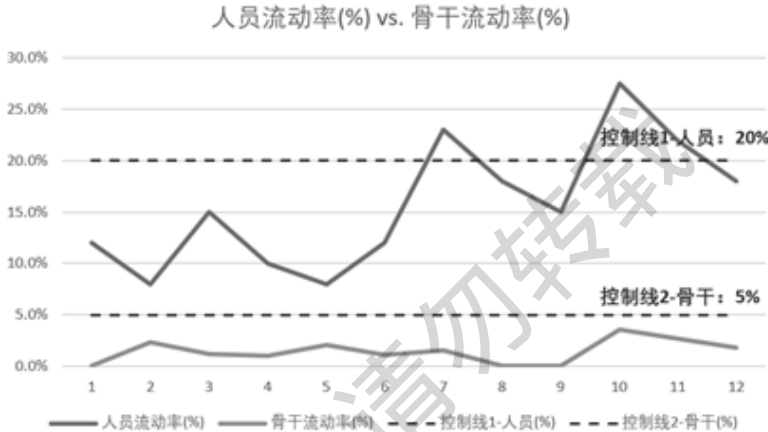


表 A. 4. 7 技能指数

指标名称	技能指数	
概念描述	技能指数是对员工技能评价的分值。指标反映了员工的知识、经验、能力等级，对人员工作分配和成本有显著影响。 宜针对人员所在岗位或领域，如项目管理、需求分析、设计开发、测试等，进行评价。企业可设置评价权重和取值范围。	
计算方法	技能指数 = $\sum$ (技能分值*权重)	
计量单位	分值	
分类	认知域/ 改进域	交付成本
	阶段	需求、设计、开发、测试、发布、运营
补充说明	员工技能指数的统计和分析有助于建立更合理的人员梯队和恰当的分配工作。	

表 A. 4. 8 人员流动率

指标名称	人员流动率
概念描述	人员流动率是统计范围内，离职、调入或调出员工与员工总数的百分比。指标反映了企业组

	织与员工队伍的稳定性。除了整体计算，宜对骨干员工和新员工进行分类统计。																																								
计算方法	人员流动率 = 流动员工数/员工总数																																								
计量单位	%																																								
分类	认知域/ 改进域	交付成本																																							
	阶段	运营																																							
补充说明	<p>控制人员流动率，特别是将骨干流动率控制在合理水平至关重要，对进度、质量、成本等有显著影响。人员流动率可在项目、产品线、部门各维度进行统计分析，示例如下：</p>  <p>人员流动率(%) vs. 骨干流动率(%)</p> <table border="1"> <caption>人员流动率(%) vs. 骨干流动率(%) 数据表</caption> <thead> <tr> <th>周期</th> <th>人员流动率(%)</th> <th>骨干流动率(%)</th> </tr> </thead> <tbody> <tr><td>1</td><td>12.0</td><td>1.0</td></tr> <tr><td>2</td><td>8.0</td><td>2.0</td></tr> <tr><td>3</td><td>15.0</td><td>1.0</td></tr> <tr><td>4</td><td>10.0</td><td>1.0</td></tr> <tr><td>5</td><td>8.0</td><td>2.0</td></tr> <tr><td>6</td><td>12.0</td><td>1.0</td></tr> <tr><td>7</td><td>23.0</td><td>2.0</td></tr> <tr><td>8</td><td>18.0</td><td>1.0</td></tr> <tr><td>9</td><td>15.0</td><td>1.0</td></tr> <tr><td>10</td><td>28.0</td><td>4.0</td></tr> <tr><td>11</td><td>20.0</td><td>3.0</td></tr> <tr><td>12</td><td>18.0</td><td>2.0</td></tr> </tbody> </table>		周期	人员流动率(%)	骨干流动率(%)	1	12.0	1.0	2	8.0	2.0	3	15.0	1.0	4	10.0	1.0	5	8.0	2.0	6	12.0	1.0	7	23.0	2.0	8	18.0	1.0	9	15.0	1.0	10	28.0	4.0	11	20.0	3.0	12	18.0	2.0
周期	人员流动率(%)	骨干流动率(%)																																							
1	12.0	1.0																																							
2	8.0	2.0																																							
3	15.0	1.0																																							
4	10.0	1.0																																							
5	8.0	2.0																																							
6	12.0	1.0																																							
7	23.0	2.0																																							
8	18.0	1.0																																							
9	15.0	1.0																																							
10	28.0	4.0																																							
11	20.0	3.0																																							
12	18.0	2.0																																							

## A.5 交付能力

表 A.5.1 流负载

指标名称	流负载	
概念描述	流负载指的是研发过程中正在处于设计、开发、测试、发布中的需求数量。指标反映了设计、开发、测试、发布各环节的平衡状态，分析流负载利于识别瓶颈和积压。	
计算方法	流负载 = $\Sigma$ （已开始但未交付的需求数）	
计量单位	#	
分类	认知域/ 改进域	交付能力
	阶段	设计、开发、测试、发布
补充说明	控制在制品数量有助于缩短交付时间，及时暴露价值交付过程中的问题和瓶颈。	

表 A.5.2 构建频率

指标名称	构建频率
概念描述	构建频率是单位时间内软件进行统一编译构建的次数。指标反映了持续构建的能力，有助于

	及早发现和修复缺陷。	
计算方法	构建频率 = 构建次数/单位时间	
计量单位	次/时长	
分类	认知域/ 改进域	交付能力
	阶段	开发
补充说明	无	

表 A. 5. 3 构建时长

指标名称	构建时长	
概念描述	构建时长是软件单次编译构建的时长。指标反映了系统复杂度和基础设施等状况，过长的时间应考虑编译构建的优化。	
计算方法	构建时长 = 构建完成时间 - 构建开始时间	
计量单位	时长	
分类	认知域/ 改进域	交付能力
	阶段	开发
补充说明	无	

表 A. 5. 4 构建成功率

指标名称	构建成功率	
概念描述	构建成功率是在统计范围内统一编译构建成功次数占统一编译总构建次数的百分比。指标反映了代码质量、编译环境的稳定性。	
计算方法	构建成功率 = 统一编译构建成功次数/统一编译构建总次数	
计量单位	%	
分类	认知域/ 改进域	交付能力
	阶段	开发
补充说明	构建成功率低可能意味着过程规范性、代码质量、工具链、基础设施等持续交付能力需进行改进。	

表 A. 5. 5 缺陷重开率

指标名称	缺陷重开率	
概念描述	缺陷重开率是在统计范围内，重新打开的缺陷占总缺陷的比率。指标反映了缺陷修复的效果，重开率高应触发针对性的改进。	



计算方法	重新打开缺陷数/总缺陷数	
计量单位	%	
分类	认知域/ 改进域	交付能力
	阶段	测试、发布、运维
补充说明	无	

表 A. 5. 6 测试自动化率

指标名称	测试自动化率	
概念描述	测试自动化率是自动化测试用例占测试用例总数的百分比。指标反映了自动化测试的能力，有助于提高测试效率和快速交付能力。	
计算方法	测试自动化率 = 自动化测试用例数/总测试用例数	
计量单位	%	
分类	认知域/ 改进域	交付能力
	阶段	开发、测试
补充说明	通常按功能测试、性能测试、安全测试等，以及单元测试、组件测试、集成测试、系统测试等维度进行统一，特别应关注回归测试自动化比例。	

表 A. 5. 7 环境整备时长

指标名称	环境整备时长	
概念描述	环境整备时长是环境整备所需的时长。指标反映了环境的复杂度和基础设施等状况，过长的时间应考虑环境的优化。	
计算方法	环境整备时长 = 环境整备完成时间 - 环境整备开始时间	
计量单位	时长	
分类	认知域/ 改进域	交付能力
	阶段	开发、测试、发布、运维
补充说明	在软件研发过程中，开发环境、系统测试环境、验收测试环境的整备时长和一致性，对快速持续交付能力有显著影响。	

表 A. 5. 8 部署频率

指标名称	部署频率
------	------

概念描述	部署频率是在单位时间内，软件进行部署的次数。指标反映了持续部署的能力，有助于及早发现和修复缺陷以及环境的不一致性。	
计算方法	部署频率 = 部署次数/单位时间	
计量单位	次/时长	
分类	认知域/ 改进域	交付能力
	阶段	开发、测试、发布
补充说明	宜考虑针对不同环境的部署时长，如开发环境、测试环境、准生产环境、生产环境等，有利于针对性的进行改进。	

表 A. 5. 9 部署时长

指标名称	部署时长	
概念描述	部署时长是软件单次部署的时长。指标反映了系统复杂度和基础设施等状况，过长的时间应考虑软件部署的优化。	
计算方法	部署时长 = 部署完成时间 - 部署开始时间	
计量单位	时长	
分类	认知域/ 改进域	交付能力
	阶段	开发、测试、发布
补充说明	无	

表 A. 5. 10 部署成功率

指标名称	部署成功率	
概念描述	部署成功率在统计范围内统一部署成功次数占统一部署总次数的百分比。指标反映了部署脚本质量和部署环境的一致性、稳定性。	
计算方法	部署成功率 = 统一部署成功次数/统一部署总次数	
计量单位	%	

分类	认知域/ 改进域	交付能力
	阶段	开发、测试、发布
补充说明	宜考虑针对不同环境的部署成功率，如开发环境、测试环境、准生产环境、生产环境等，有利于针对性的进行改进。	

表 A. 5. 11 回滚成功率

指标名称	回滚成功率	
概念描述	回滚成功率是在统计范围内回滚成功次数占回滚总次数的百分比。指标反映了回滚脚本质量和基础设施的能力，回滚成功率是降低持续交付风险的关键指标。	
计算方法	回滚成功率 = 回滚成功次数 / 回滚总次数	
计量单位	%	
分类	认知域/ 改进域	交付能力
	阶段	开发、测试、发布
补充说明	宜考虑针对不同环境的回滚成功率，如开发环境、测试环境、准生产环境、生产环境等，有利于针对性的进行改进。	

表 A. 5. 12 发布频率

指标名称	发布频率	
概念描述	在单位时间内，软件发布的次数。该指标反映了软硬件基础设施、交付过程、研发团队质量和效率等多种能力。	
计算方法	发布次数 / 单位时间	
计量单位	次/时长	
分类	认知域/ 改进域	交付能力
	阶段	发布
补充说明	控制合理的发布频率可以满足快速交付和获得客户及时反馈的需要。	

表 A. 5. 13 发布时长

指标名称	发布时长	
概念描述	发布时长是软件单次发布的时长。指标反映了系统复杂度和基础设施等状况，过长的时间应考虑软件发布过程的优化。	
计算方法	发布时长 = 发布完成时间 - 发布开始时间	
计量单位	时长	
分类	认知域/ 改进域	交付能力
	阶段	发布
补充说明	无	

## A. 6 持续改进

表 A. 6. 1 改进效果评价

指标名称	改进效果评价	
概念描述	改进效果评价是改进前后选定度量数据的变化比例。指标反映了专项整改的量化的效果。	
计算方法	改进效果评价 = 改进后数据/改进前数据	
计量单位	%	
分类	认知域/ 改进域	持续改进
	阶段	需求、设计、开发、测试、发布、运营
补充说明	例如，改进前项目返工率均值为 35%，改进后返工率均值降到 17.5%，则项目返工率均值下降了 50%。改进效果评价还可使用其他统计方法，如假设检验（检验均值、标准差、中位数的变化等），以建立对过程改进效果的深入理解，并支持决策分析和风险评估。	

表 A. 6. 2 专项改进完成率

指标名称	专项改进完成率
概念描述	专项改进完成率是统计范围内专项改进项目的完成比例。指标反映了专项改进在公司级、部门级、产品线的实施情况。
计算方法	专项改进完成率 = 按时完成的专项改进数/计划完成的专项改进数

计量单位	%	
分类	认知域/ 改进域	持续改进
	阶段	需求、设计、开发、测试、发布、运营
补充说明	宜按照年度、季度、月度统计改进项目数、状态、成功率等，以确保相应的持续改进能力。	

表 A. 6. 3 审计问题关闭率

指标名称	审计问题关闭率	
概念描述	审计问题关闭率是统计范围内审计发现问题已关闭的比例。指标反映了及时处理并解决审计所发现问题的能力。	
计算方法	审计问题关闭率 = 已关闭的审计问题数/审计问题总数	
计量单位	%	
分类	认知域/ 改进域	持续改进
	阶段	需求、设计、开发、测试、发布、运营
补充说明	宜按类型、严重程度等分类统计，以及时处理和解决内外审计中的关键不符合项。	

表 A. 6. 4 审计频率

指标名称	审计频率	
概念描述	审计频率是在单位时间内内外部审计的次数。指标反映了外部合规和内部管控的力度。内外部审计的频率宜按类别适度控制，为体系落地，及早发现问题，及时改进提供支持。	
计算方法	审计频率 = (内部审计次数+外部审计次数)/时长	
计量单位	次/时长	
分类	认知域/ 改进域	持续改进
	阶段	需求、设计、开发、测试、发布、运营

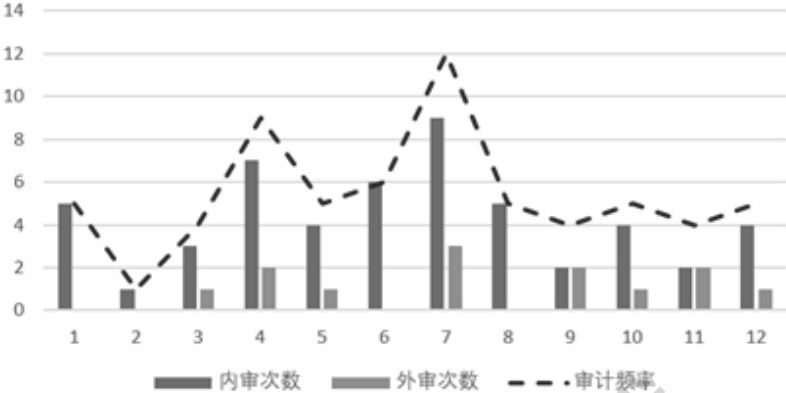
补充说明	<p>为了保证质量和合规，特定行业需要频繁进行审计。合理的审计频率，可以保证过程持续置于监管之下，并保证规章制度得以贯彻执行，示例如下：</p> <p style="text-align: center;"><b>内/外部审计 vs. 频率</b></p>  <p>审计频率的统计宜按类别进行统计，如：外部审计、内部审计、需求审计、设计审计、开发审计、测试审计、发布审计、质量审计等。内部审计宜按季度策划和执行，确保项目、产品线、部分和体系的覆盖率和持续受控。</p>
------	--

表 A. 6. 5 过程符合度

指标名称	过程符合度	
概念描述	<p>过程符合度是统计范围内体系和成果物审计合规的比例。指标反映了制度和规范的落实情况。</p> <p>应以质量审计和配置审计为依据，关注项目中体系实施的必要性和充分性，并计算其符合度。</p>	
计算方法	<p>过程符合度 = <math>(\sum (G)*5 + \sum (Y)*3) / (\sum (G, Y, R)*5)</math></p> <p>G/Y/R 是审计评价结果，其中，G:大部分满足，Y:部分满足，R:不满足，NA:不适用，NY:时机未到。</p>	
计量单位	%	
分类	认知域/ 改进域	持续改进
	阶段	需求、设计、开发、测试、发布、运营

补充说明

适用的过程可根据企业业务实际确定，包括但不限于需求开发、需求管理、版本规划、设计规范、开发规范、测试规范、评审规范、验收规范、上线规范、运营规范、项目立项、项目估算、项目策划、项目监控、项目结项、风险管理、问题管理、缺陷管理、事件管理、度量分析、决策分析、根因分析、配置管理、质量追溯、质量保证等制度和规范及其检查单等，示例如下：

适用过程/规范/指南	产品线1				产品线N					
	项目1	项目2	项目3	项目4	项目5	项目6	项目7	项目8	项目9	项目10
工程管理制度	G	G	G	G	G	G	G	G	G	G
需求分析指南	G	Y	G	G	Y	Y	G	G	Y	G
设计规范	G	G	G	G	G	G	Y	G	G	G
集成及测试指南	Y	G	Y	G	G	G	G	G	Y	Y
测试验证规范	G	G	G	Y	G	Y	G	Y	G	G
技术评审规范	G	Y	G	G	G	Y	Y	G	G	G
代码评审规范	Y	Y	R	Y	Y	G	G	G	Y	Y
项目实施流程	G	NA	NA	NY	NA	NA	G	NY	NA	NA
项目管理程序	G	G	G	Y	G	G	G	G	G	G
项目估算指南	Y	Y	Y	G	Y	Y	Y	G	Y	G
风险与机会管理指南	G	G	G	G	G	Y	Y	Y	G	G
项目度量分析指南	Y	Y	G	Y	G	NY	Y	G	Y	Y
决策分析指南	R	Y	Y	NY	Y	NY	G	R	G	NY
根因分析指南	R	R	R	NY	NY	NY	NY	R	NY	NY
项目配置管理程序	G	Y	G	G	G	G	G	G	G	G
项目质量保证程序	G	Y	G	G	G	Y	G	G	G	G
*****										
质量审计得分	78%	72%	79%	88%	89%	80%	87%	81%	86%	91%

术语定义	缩写语	分值
大部分满足	G	5
部分满足	Y	3
不满足	R	0
不适用	NA	
时机未到	NY	

过程符合度数据可进一步汇总到产品线 and 部门级，并通过时序分析，展示和分析过程符合度的变化，与目标值或基线进行对比分析，支持过程改进及其效果评价。

## 参 考 文 献

- [1] GB/T 8566-2007 信息技术软件生存周期过程
  - [2] GB/T 32423-2015 系统与软件工程验证与确认
  - [3] Practical Software and System Measurement Version 4.0c
  - [4] Continuous Iterative Development Measurement Framework V2.1
- 

未经许可，请勿转载