

CODING DevOps

产品简介



腾讯云

【 版权声明 】

©2013–2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分內容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

文档目录

产品简介

产品概述

子产品简介

代码托管

项目管理

测试管理

持续集成

制品库

持续部署

效能洞察

产品优势

应用场景

产品简介

产品概述

最近更新时间：2023-09-11 16:05:11

CODING DevOps 简介

腾讯云 CODING DevOps 是面向软件研发团队的一站式研发协作管理平台，提供从需求到设计、开发、构建、测试、发布到部署的全流程协同及研发工具支撑。CODING 解决方案可助力企业实现代码的统一安全管控，并快速实践敏捷开发与 DevOps，提升软件交付质量与速度，降低企业研发成本，实现研发效能升级。

子产品简介

子产品	产品描述	适用场景
代码托管	提供企业级的 Git/SVN 代码管理服务，支持精细化权限管控、多分支并行开发、多版本管理等功能。	企业代码管理、代码评审、分支管理等。
项目管理	软件开发团队可自由选择适合的研发管理模式，支持多项目管理、敏捷迭代管理、需求管理、缺陷跟踪、多维度报表数据等功能。	敏捷研发管理、需求管理、缺陷定位等。
测试管理	提供面向敏捷团队的一站式云端测试平台，支持可视化的测试规划和多维度的测试报告，满足敏捷团队对测试过程的多样化需求。	测试协作、测试用例管理等。
持续集成	提供基于云端的自动化代码构建、测试、分析和部署 workflow 服务，支持通过模板快速创建构建任务并进行可视化编排，极大提高软件开发团队的构建效率。	自动构建、自动化测试等。
制品库	提供云端构建产物管理服务，支持云端构建和本地构建推送，可快速索引存档构建物、进行版本控制。	依赖包存储、制品库质量管控、应用分发等。
持续部署	提供全自动化软件部署，可持续、可控地把软件制品在线发布到服务集群中，支持蓝绿发布、灰度发布（金丝雀发布）等多种发布策略。	灰度发布、蓝绿发布、发布审批等。
效能洞察	专注于研发效能的度量平台，通过专业的度量方法对 DevOps 全流程领域的研发效能数据进行可视化、场景化呈现，帮助管理者、研发人员、项目经理和研效专家等各类角色分析、理解和洞察整体效能情况，持续完成效能改进。	组织级研效评估、研发团队效能度量、项目交付价值流分析、个人效能改进等。

子产品简介

代码托管

最近更新时间：2023-09-11 16:05:11

产品概述

CODING 代码托管是 CODING DevOps 的子产品，为开发者打造的云端便捷研发协作工具，旨在为更多的开发者带去便捷、高效的开发体验，全面支持 Git、SVN 两种常见版本控制工具，云端高可用，无忧数据保障。用户可以通过本产品进行代码托管、代码评审和分支管理。

产品优势

CODING 代码托管是腾讯云为用户打造的云端便捷研发协作工具，其产品优势如下。

高可用仓库

CODING 代码托管提供高性能远端代码仓库，支持 Git、SVN 两种版本管理方式，分布式计算和存储，提供无限制的云端计算资源。给用户提供云端高可用环境，保障用户数据安全。本地化服务提供更快的速度体验，支持无限速的上传和下载。

代码评审

当开发工程师完成代码的开发，并创建一个合并请求，可以邀请其他成员对其代码进行评审，支持对代码逐行评论，提供良好的代码阅读与代码审查体验。帮助团队建立代码评审机制，可以有效减少代码错误数量，提升代码品质。

发布管理

版本管理基于 Git Tag，可以帮助您在目标节点将代码打包发布，每一个版本对应一个标签。您可以选择某个分支的最新代码发布一个版本，自动生成 ChangLog，规范化版本号，关联到任务里程碑，使得源码发布和交付标准高效。

代码浏览

CODING 代码托管提供良好的代码浏览工具，浏览代码时支持上下及左右对比模式，能清晰看到每一行代码修改前后的对比，显示有多少处（行）代码被修改，并支持对每一行被修改的代码进行评论。每一次代码的修改都会完整呈现，清晰可见。

分支保护

CODING 代码托管支持代码分支粒度管理，可以根据团队内成员权限进行精细化权限管理，设定分支管理员，使得有权限的成员才能修改，将特定分支保护起来，防止被破坏，让团队协作规范有序。

应用场景

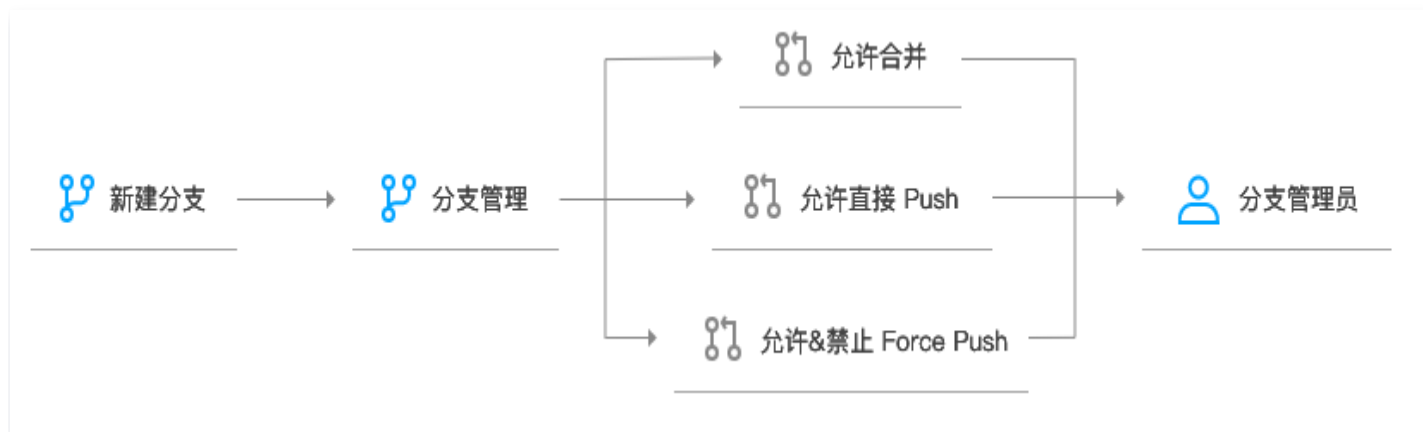
代码管理

CODING 代码托管支持 Git、SVN 两种模式进行代码管理，方便研发团队高效地管理代码资源。其中 Git 模式具备查看提交历史、代码对比、保护分支、部署公钥等功能，SVN 模式目前暂只支持代码浏览功能。



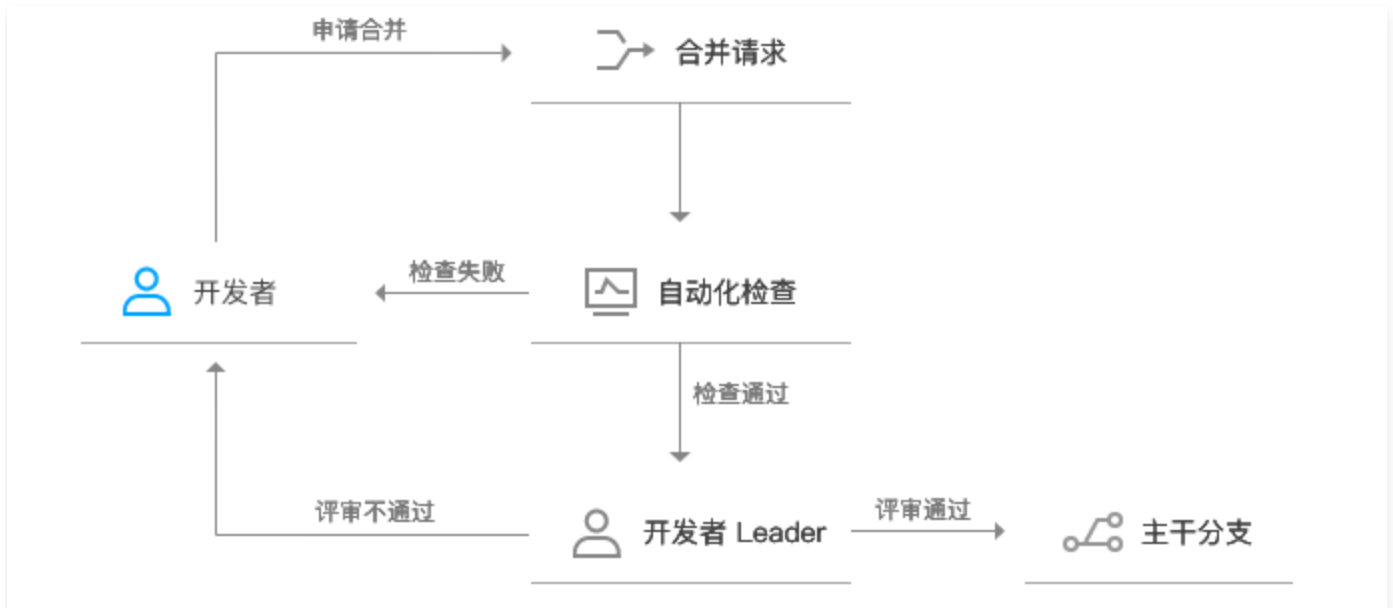
分支管理

在多人开发的情况下，CODING 代码托管能清晰地查看各分支权限，针对团队成员设置分支操作权限，进行精确有效的管理，提高安全性和协作流畅度。



代码评审

开发完毕的代码在合并入主干分支前可以进行组员评审，审批等流程，并可以通过配置持续集成等自动化工具，检测代码风格并对代码质量进行辅助评审。



项目管理

最近更新时间：2023-09-11 16:05:11

产品概述

CODING 项目管理是 CODING DevOps 的子产品。项目是实践 CODING DevOps 的最小单位，您可以将它当做最基础的项目进度可视化协调工具，将合并请求关联至事项，分配缺陷至相关责任人。CODING 项目协同功能模块是协调各个事项的调度中心，我们为团队提供 **Scrum 敏捷项目管理模式**和**经典项目管理模式**两种方案，团队可以根据管理偏好选择合适的模式。CODING 让所有协作发生在云端，通过合理的任务分配与处理机制实现无间协作，帮助团队攻克难关，高效生产，提高交付效率。

功能概览

迭代

团队可以根据研发效能，将开发工作分为2周 - 3周一轮的迭代，并使用规划功能将需求或缺陷规划进迭代中。开始迭代后，可以通过迭代概览查询迭代的详情和进度，并通过事务列表查看当前迭代内的需求、任务和缺陷。

需求管理

需求是指解决某一问题或达到某一目标所需的软件功能，需求决定了软件研发的方向与结果。需求管理包括需求创建、需求分解、需求处理和任务分解等功能，当创建一个需求时，可以设置优先级、截止日期和需求分类等基本信息，并指定相关人员处理。较大粒度的需求则可以分解为较小的子需求，通过在需求下新建或关联任务实现开发任务的分解和分配。

任务管理

任务是指为实现某个需求所进行的具体活动。任务管理主要包括任务创建、任务列表和任务处理等功能，当团队在项目协同中创建一个任务时，可以设置优先级、截止日期和任务描述等基本信息，并指定相关人员处理。

缺陷管理

缺陷是指不符合最初定义的业务需求。缺陷管理主要包括报告缺陷、缺陷列表、缺陷处理和缺陷报告等功能，当创建一个缺陷时，可以设置优先级、截止日期和缺陷分类等基本信息，并指定相关人员处理。处理状态是指缺陷在生命周期中所处的阶段，用于组织和跟踪缺陷。

自定义属性和 workflow

团队可以通过自定义属性和 workflow，配置项目内需求、任务和缺陷的属性和 workflow，满足不同团队的定制化属性和 workflow 需求。通过定义全局属性和状态，团队能够实现属性和状态定义的一致性，进而提高跨项目和跨部门协作的效率。

Wiki 和网盘

Wiki 管理：Wiki 支持 Markdown 格式，拥有版本控制、新旧版本对比、多层级拓展等功能。可以让项目成员更好地进行文档书写及协作，帮助团队进行知识沉淀。同时还支持公开文档链接和一键生成静态网站等功能。

文件管理：项目文件中心支持对文件进行上传、下载、协同编辑、在线预览、公开分享和删除等操作。每个文件都拥有动态和历史版本，可以查看下载旧版本文件。支持在线查看 Excel、PPT、Word 和 PDF 文件，可对 TXT 或 Markdown 文档进行编辑操作。

产品优势

CODING 项目管理适用于研发团队进行项目管理或敏捷开发实践，其产品优势如下。

多个协作功能一体化

项目管理工具已具备迭代管理、需求管理、任务管理、缺陷管理等功能，实现团队内产品、研发、测试、运营等不同角色无缝协作。同时项目管理工具对需求任务进行分解、关联、预估工时和讨论，准确把控每个迭代中的事务进度，达到可持续性快速迭代。

可选研发模式

项目管理提供面向敏捷团队的 Scrum 模式和偏向传统项目管理下的经典模式，研发团队可以根据自身需求及实际的项目管理偏好选择项目协同模式。敏捷研发是涉及整个软件工程的理念与实践，它的核心是迭代和增量式软件开发方法，而传统项目管理的特点是强计划驱动，围绕着需求、资源和时间展开，需求固定下来后才可分配人员和时间。

自定义 workflow

项目管理功能提供自定义 workflow 功能，研发团队可根据不同的需求场景来设计团队内需求、任务的处理流转过程。帮助团队管理规范化，定制团队 workflow。

多元化权限管理

支持从企业成员管理以及项目成员管理的两个维度进行权限设置。项目管理工具支持对团队成员进行角色分组，分别赋予不同组别的权限，并对每个组别按功能权限进行精细定义，为团队打造一个高效的权限管理体系。

沉淀团队知识库

项目内支持 Word、Excel、PPT、视频等不同格式文件存储、在线预览、下载及分享，按文件夹进行整理归纳。Wiki 功能支持在线树形文档结构编辑存储。文件、Wiki 可保存每个历史版本，满足不同大小团队内各部门的协作、管理和文档资源追溯，沉淀团队知识库。

项目资源深度关联

项目需求、任务、缺陷之间可相互关联资源，也可关联相关的文件、Wiki，使每个事务之间关系可追踪、可定位，帮助团队快速理解上下文和事务背景。

可视化数据报告

提供事务状态甘特图、数据报表等可视化视图，随时监控迭代内事务进度和发展趋势，协助团队分析问题，给团队管理者提供决策依据。可以快速调整项目预期和计划，确保敏捷协作实践过程进度可控。

测试管理

最近更新时间：2023-09-11 16:05:12

产品概述

CODING 测试管理是 CODING DevOps 的子产品，为您提供井然有序的测试协同管理工具服务。从测试用例库管理、制定测试计划，到协作完成测试任务，为测试团队提供敏捷测试工作方式，提高测试与研发团队的协同效率。提供可视化的工作视图以及数据报告，随时把控测试进度和规划。

产品优势

CODING 测试管理为您提供井然有序的测试协同管理工具服务，其产品具有以下优势。

多角色实时协同

在 CODING 测试管理中，团队内的测试、产品、研发上下游均能协同参与测试计划，保持测试需求、测试任务、缺陷任务的实时同步。记录测试过程，便于产品和研发复现问题，快速定位需求和缺陷，实现各部门之间的无缝协作。

测试用例关联缺陷

与 CODING 缺陷管理功能深度关联，测试过程中发现功能未符合预期即马上关联缺陷，开发者可根据缺陷快速定位测试用例，测试结果更新也可同步到对应缺陷。

测试用例可复用

测试用例库支持新增、导入及编辑用例，且树形结构的用例库便于灵活组织测试计划，使得用例能重复使用于不同的测试计划中。避免重复编辑相同的用例，大大提升测试工作效率。

批量导入测试用例

支持批量导入 Excel 或 CSV 格式的测试用例到指定的项目中，不需重新编辑用例。帮助您快速脱离低效的表格管理，开启高效的测试协同管理方式。

可视化测试进度

提供可视化工作视图纵览项目测试概况，测试进展一目了然。直观看到每一个人的工作进度和任务量，准确把握团队工作量峰、谷值，轻松改善团队工作规划。

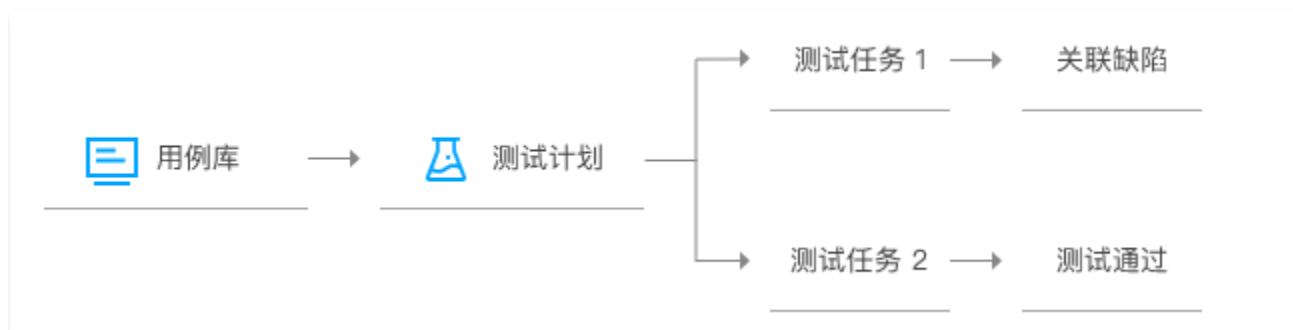
多维度数据报告

测试迭代结束后，可以创建定期数据报告模板，系统会在设定的时间自动生成测试结论、图表、工作分布、测试耗时等维度的测试报告，并定期发送给相应关注者。

应用场景

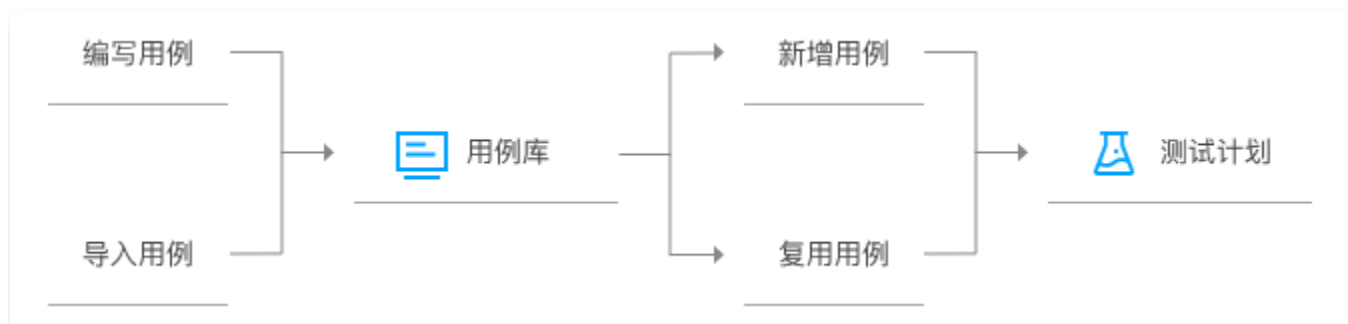
测试协作

CODING 测试管理帮助测试团队建立井然有序的测试协作流程，包含编写测试用例、规划测试计划、分发测试任务、记录测试结果，还可与研发人员协同，同步测试过程和结果。



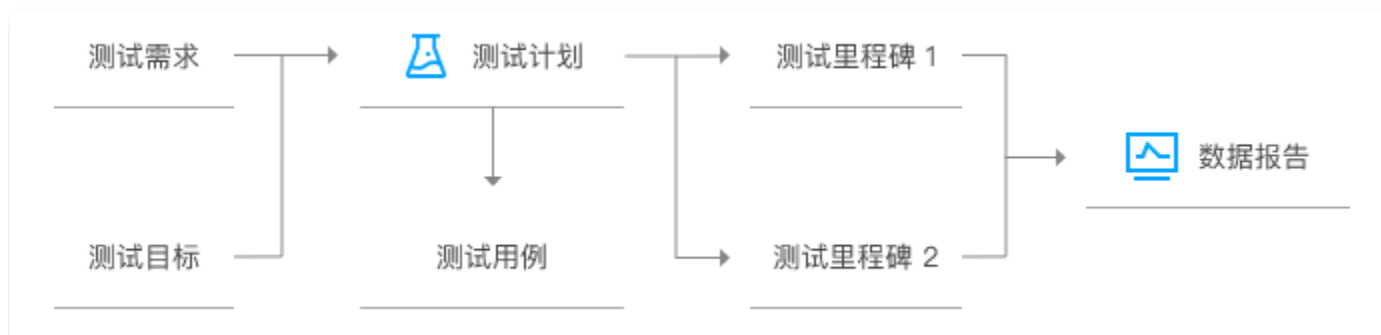
测试用例管理

用于建立团队的测试用例库，并管理每个功能或页面的用例，每个用例包括了本次测试的描述、步骤、预期结果。用例库的管理有利于实现用例复用，避免重复编写用例，提高工作效率。支持 Excel、CSV 格式导入用例，将测试人员从繁杂的表格编辑中解脱出来。



测试计划制定

帮助测试团队管理者按不同阶段的测试目标制定测试计划，测试结束后自动生成对应的测试报告，分析测试成果情况，为下一次测试计划提供数据支持。



持续集成

最近更新时间：2023-10-07 10:44:21

产品概述

CODING 持续集成是 CODING DevOps 的子产品，其全面兼容 Jenkins 的持续集成服务，支持 Java、Python、Node.js 等主流语言，并且支持 Docker 镜像构建，图形化编排，高配集群多计划并行构建全面提升您的构建任务。支持主流的 Git 代码仓库，包括 CODING 代码托管、GitHub、GitLab 等等。在构建依赖拉取方面，使用专用网络优化包括 Maven，NPM 等主流镜像源，保证拉取速度，进一步提升构建速度。

产品优势

CODING 持续集成的产品优势如下：

全面的构建类型

除了支持 Docker 镜像构建，也支持 Jar，APK 等软件包的构建，并且预置了大量的构建环境镜像，例如 Java、Python、Node.js 等。

多构建计划并行构建

支持单项目并行构建，以满足重度持续集成用户的需求。后端的服务器集群可以根据用户需求实时调度相应的计算资源，保证用户构建任务的快速开始，减少排队时间。

缓存加速

持续集成构建过程中会用大量的时间重复下载依赖文件，致使构建时间过长，CODING 持续集成支持在不同的构建任务之间支持开启缓存，以提高反复构建的速度。开启缓存功能可以平均提高300%的构建速度。

图形化编排

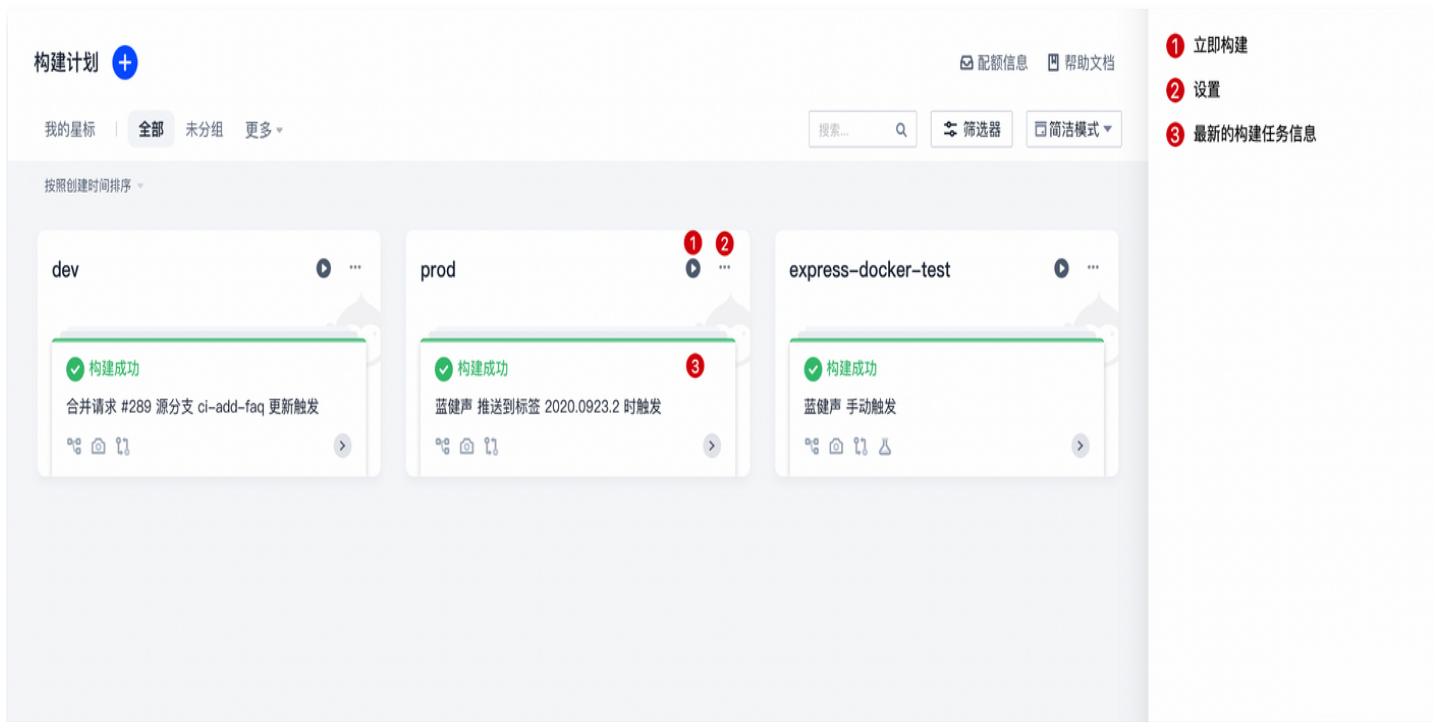
除了手工编辑构建脚本以外，CODING 持续集成还有完善的图形化编排能力，以降低使用门槛。针对构建的每一个步骤提供丰富的构建脚本模板供用户选择，实现了边写边看、所见即所得的直观编辑体验。

全面兼容 Jenkins

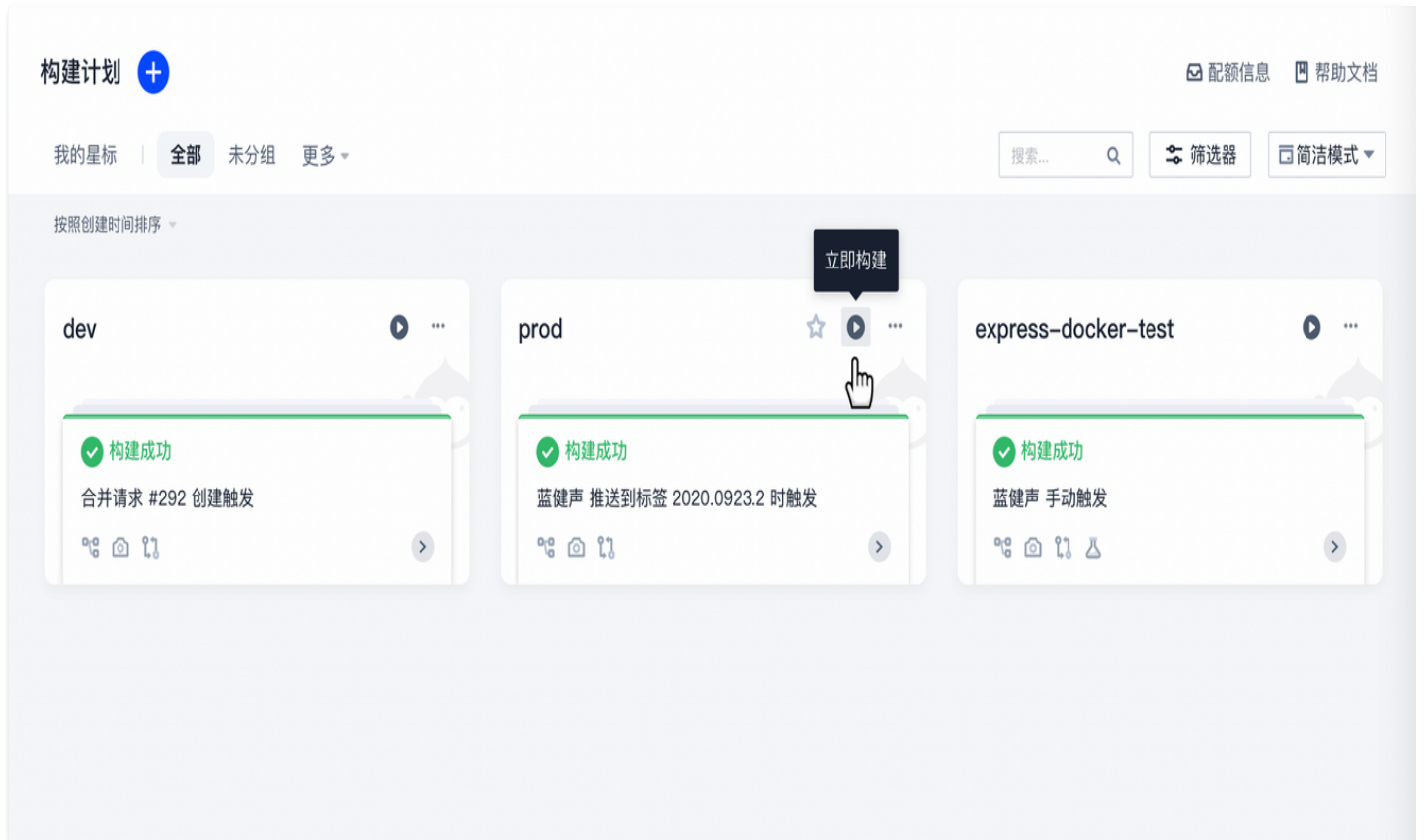
Jenkins 是目前广泛应用的持续集成工具，CODING 持续集成的构建脚本在语法上全面兼容 Jenkins。用户可以无缝迁移 Jenkins 的构建至 CODING。

功能指引

持续集成的核心组成是构建计划，我们采用卡片化的设计形式进行展示，功能内相关按钮的说明参见下图：



将鼠标移至图标上将会悬浮展示该按钮的功能说明。



在构建记录页面中，用户还可以更方便地设置构建计划和筛选构建记录。

构建计划 +

dev | CODING 中国上海 1

状态徽标 定时触发 缓存 设置 立即触发

只显示我触发的 筛选: 全部 2

全部构建状态	触发信息	持续时长	开始时间	快速查看	操作
✓ 构建成功	合并请求 #291 源分支 mr/... #601 -> 57cb55a	38 秒	32 分钟前		...
✓ 构建成功	蓝键声 推送到分支 master... #600 ma... -> 1e5c096	46 秒	15 小时前		...
✓ 构建成功	合并请求 #292 创建触发 #599 -> 5d7023e	40 秒	17 小时前		...
✓ 构建成功	合并请求 #291 创建触发 #598 -> e027190	38 秒	18 小时前		...
✗ 增量检查 git commit / 构建失败	合并请求 #290 源分支 mr... #597 -> 32fc33c	33 秒	18 小时前		...
✗ 检查代码规范 / 构建失败	合并请求 #290 创建触发 #596 -> c281b6b	27 秒	19 小时前		...
✓ 构建成功	合并请求 #289 源分支 ci-... #595 -> 24fd3d3	49 秒	2 天前		...
✓ 构建成功	合并请求 #289 创建触发 #594 -> 889317e	39 秒	2 天前		...

1-15 个, 共 601 个

每页显示行数 15 | 1 2 3 4 5 6 ... 41 >

1 代码源和构建节点
2 筛选基于特定标签/分支/修订版本构建的记录

在单条构建记录中，可以通过链接快速定位到构建分支和修订版本，在快速查看区域，您也可以通过链接查看构建过程、构建快照和改动记录。

构建计划 +
dev | CODING | 中国上海
状态徽标 | 定时触发 | 缓存 | 设置
立即触发

我的星标 | 全部 | 未分组 | 更多

按照创建时间排序

只显示我触发的
筛选: 全部

全部构建状态	触发信息	持续时长	开始时间	快速查看	操作
✔ 构建成功	合并请求 #291 源分支 mr/... #601 57cb55a	38 秒	35 分钟前		...
✔ 构建成功	蓝健声 推送到分支 master... #600 ma... 1e5c096	46 秒	15 小时前		...
✔ 构建成功	合并请求 #292 创建触发 #599 5d7023e	40 秒	18 小时前		...
✔ 构建成功	合并请求 #291 创建触发 #598 e027190	38 秒	18 小时前		...
✘ 增量检查 git commit / 构建失败	合并请求 #290 源分支 mr... #597 32fc33c	33 秒	18 小时前		...
✘ 检查代码规范 / 构建失败	合并请求 #290 创建触发 #596 c281b6b	27 秒	19 小时前		...
✔ 构建成功	合并请求 #289 源分支 ci-... #595 24fd3	49 秒	2 天前		...
✔ 构建成功	合并请求 #289 创建触发 #594 889317e	39 秒	2 天前		...

1-15 个, 共 601 个
每页显示行数 15 | 1 2 3 4 5 6 ... 41 >

构建计划

构建计划（Job）是持续集成的基本单元，您可以在构建计划中详细配置构建计划的代码源、构建流程、触发规则、环境变量、通知提醒等信息。在后续使用过程中，按照既定的规则触发该计划，从而实现自动化的流水线构建。

构建任务

构建计划在配置完成后，每一次构建执行，都产生一个具体的构建任务。您可以查看每个构建任务的构建过程、改动记录、测试报告、构建产物、构建快照等执行信息。

Jenkinsfile

Jenkinsfile 定义了持续集成中的工作流水线（pipeline），其不仅实现了对步骤的流式化封装和管理，也是持续集成中的基本功能单位。流水线可以顺序执行，也可以并行执行。

制品库

最近更新时间：2023-09-11 16:05:12

产品概述

CODING 制品库是 CODING DevOps 的子产品，用来管理源代码编译后的构建产物，支持 Docker、Maven、Helm、npm 包等常见制品库类型，制品库可以跟源代码协同进行版本化控制，可以与本地各构建工具和云上的持续集成，持续部署无缝结合，并支持漏洞扫描等特性。为研发团队提供优质高效的构建物管理服务，把控构建物质量。

产品优势

CODING 制品库具有如下优势：

支持多种制品类型

支持包括 Docker、Maven、npm、Generic、Pypi、Helm 等多种制品库类型，满足单项目多个制品库的诉求，可实现同一个项目中既支持 Docker 又支持 Maven。

无缝衔接常见构建工具

CODING 制品库兼容所有常见的制品格式标准，您可以不用更换任何构建工具、安装任何其他本地软件或者插件，即可无缝使用。

支持极速分发

支持公开仓库和私有仓库极速分发，依托腾讯云强大的 CDN 能力，您的团队可以在全球范围内安全地，极速畅享制品库上传和下载。

支持漏洞扫描

存放在制品库的制品产物可以使用预先提供的镜像安全扫描功能或自定义的安全扫描策略进行质检，提高项目安全审查力度。

整合上下游

不管是与上游的代码仓库版本匹配，还是与持续部署和运维系统的接口兼容，CODING 制品库都提供了良好的适配接口，使得 DevOps 可以上下游一体化。

应用场景

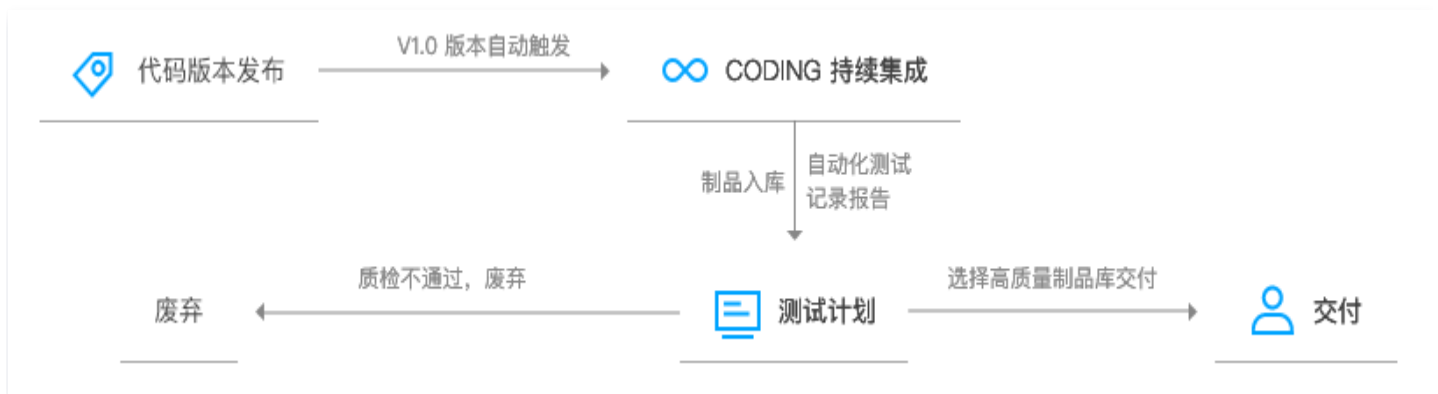
依赖包存储

CODING 制品库可以用于执行存储软件开发过程中的私有部分组件或中间产物，实现开发部门和小组之间高效的代码组件协作。



制品库质量管控

CODING 制品库支持对部分格式的构建产物进行漏洞扫描，并且可以跟测试管理、缺陷管理等进行结合，为构建产物记录缺陷情况，把控质量。



应用分发

适用于应用分发，不论是桌面软件，还是容器镜像，用户都可以通过预先搭建好的极速全球连通网络进行快速下载。



持续部署

最近更新时间：2023-09-11 16:05:12

产品概述

CODING 持续部署是 CODING DevOps 的子产品。持续部署指在软件开发过程中，以自动化方式，频繁而且持续性的将软件部署到生产环境，使软件产品能够快速的交付使用。作为持续集成的延伸，持续部署以 CODING 上下游产品优势为根基，是实现 DevOps 闭环的核心流程，实现全流程管控。

CODING 持续部署用于把控构建之后的项目发布与部署交付流程。能够无缝对接上游 Git 仓库、下游制品仓库以实现全自动化部署。同时还支持 Webhook 等外部对接能力，高效集成各种开发、运维工具。在稳定的技术架构、运维工具等基础上，具备蓝绿发布，灰度发布（金丝雀发布），滚动发布，快速回滚等能力。

产品优势

发布多种构建产物

不论是 Docker 镜像、War 包、Helm 包，还是软件源代码，都可以配置在 CODING 持续部署中作为待发布制品，可无缝对接 CODING 持续集成和制品库从而实现全流程 CI/CD。

支持多种运行环境

借助 Spinnaker 对基础设施完善的抽象机制，CODING 持续部署可以对接各类常见的运维系统和云原生环境，例如 CVM 云服务器、SCF 云函数、TKE 容器服务、Kubernetes 集群等。

灵活的发布流程编排

发布流程将持续部署流水线化，可以方便的组织起具备多个微服务的应用在不同集群环境（例如：生产、测试、预发布）之间的发布流程控制。并具备过程预编排、错误回退等能力。

强大的审批系统

预编排的审批流程可以让发布过程更为稳定可靠，用户可以根据发布流程的级别差异自动适配不同审批流程。测试、产品经理和技术主管等角色都可以根据需求加入发布的审批环节中，配合上自动化过程和通知机制，使发布过程更加高效。

静态网站服务

这是 CODING 持续部署的特色功能，支持将代码仓库中的代码或项目文件部署成一个静态网站，支持绑定自定义域名，自动续期 SSL/TLS 证书。可以用来搭建个人博客，企业官网等简易网站。

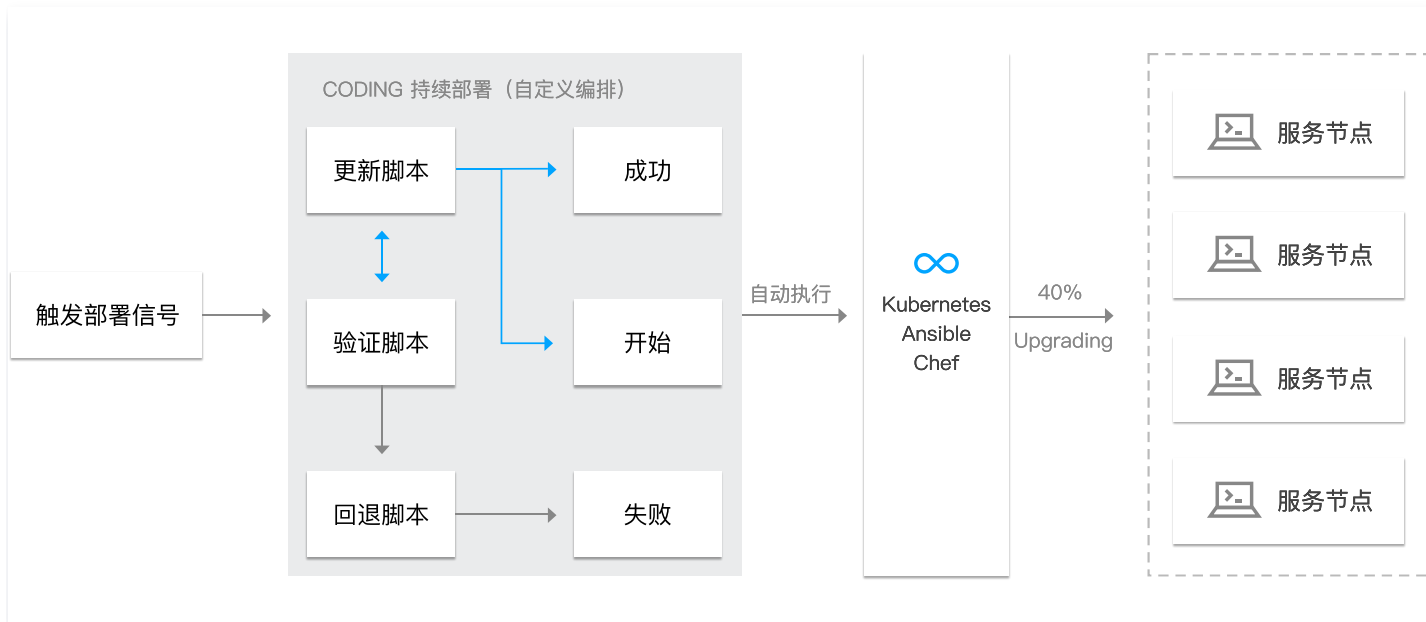
完善的扩展生态

CODING 部署控制台基于持续交付基金会旗下开源项目 Spinnaker 实现，在其国际化的开源生态的基础上做了大量优化和改造，使控制台能够流畅地对接 CODING 中的上下游研发流程，提供开箱即用的开发体验。

应用场景

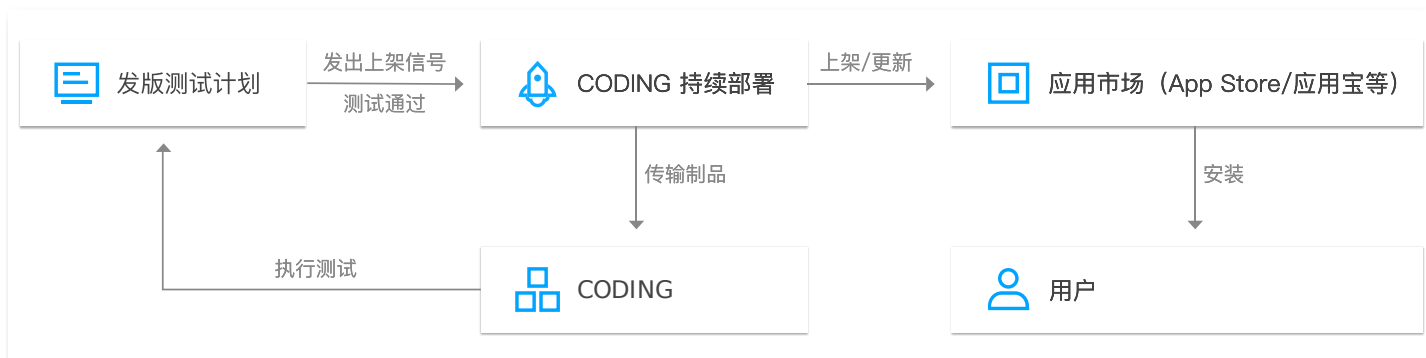
灰度发布

CODING 持续部署支持配置发布策略，配合负载均衡、监控等运维工具实现金丝雀试验、灰度发布、蓝绿发布等能力。



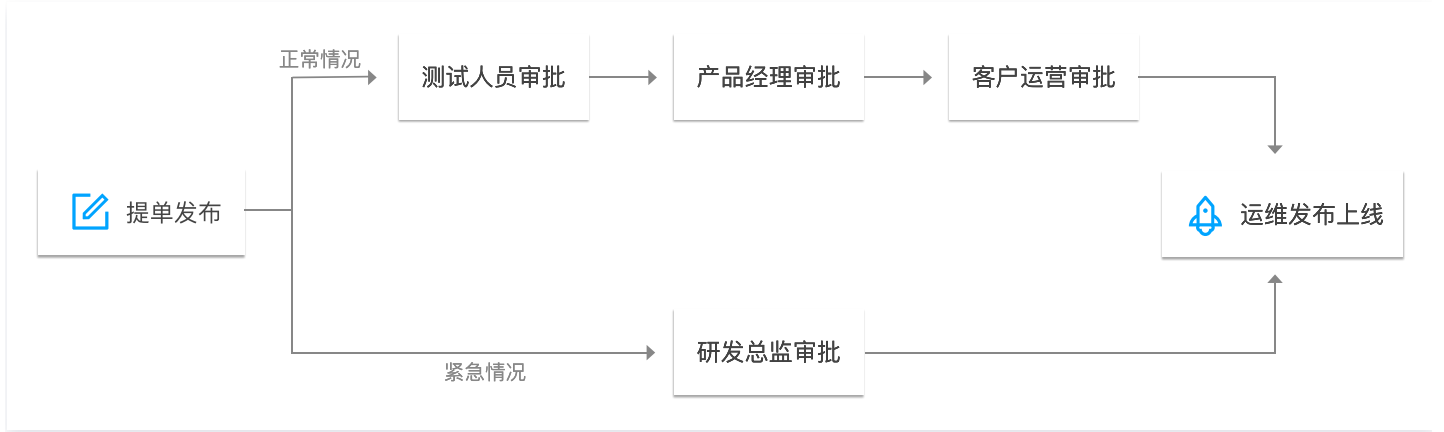
应用上架

CODING 持续部署可直接把应用的构建产物交付到应用市场，实现全自动上架。



发布审批

CODING 持续部署支持在发布流程开始前，实现多场景的人工和自动化审批。



效能洞察

最近更新时间：2023-09-11 16:05:12

产品概述

效能洞察是 CODING DevOps 的子产品，包含模板视图、模板图表、图表数据下钻等功能，用以统计分析团队成员在一段时间内的事项分布、事项概览、代码分布等数据，由此度量团队成员在周期内完成的工作量与工作动态。好比服务器中的探针程序，运行后便能够收集工作数据，最后汇总至可视化视图中以供团队管理员进行分析。

同时效能洞察还具备多维度、跨项目的筛选条件查询项，能够全方位覆盖交付效率，为团队建立更专业的研发度量指标体系。

产品优势

开箱即用的度量模板

提供开箱即用的团队度量、项目度量、个人度量及质量分析、需求分析、效率分析、价值与成本分析等多个专家实践模板视图，帮助您快速开始效能分析与洞察。

丰富的度量指标

提供50+统计度量指标，包括需求交付周期、缺陷修复周期、代码提交趋势、构建频率与部署成功率等，为企业及团队提供覆盖软件开发全生命周期的统计视角。

灵活的自定义能力

每个度量指标均可自由配置呈现样式以及数据筛选条件、时间范围、对比时间维度等多维度信息，灵活满足不同场景的度量分析需求。

数据自动采集

CODING 效能洞察集成于 CODING DevOps 工具链，无需人工干预，既可以自动完成从需求、代码、构建、测试到部署各环节的数据采集，让数据客观、如实反映研发现状。

数据可视化

提供柱状图、饼状图、燃尽图、面积图等多种可视化图表指标，只需简单配置即可快速生成所见即所得的度量图表，并支持一键分享、复制、数据导出等便捷操作。

专业的度量方法

融合精益、敏捷、持续交付等先进工程理念和实践，打造精准有效的研发效能度量指标，以专业的效能度量方法驱动研效提升。

应用场景

组织级研效评估

从全局视角，系统性分析各类质量、效率、成本指标，以及涵盖软件交付全流程领域的过程指标。帮助组织快速了解内部研发效果的现状，定位全局效能瓶颈点，并基于评估结果进行优化。



研发团队效能度量

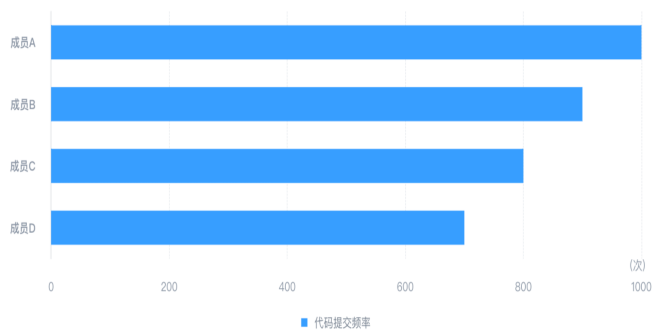
提供代码提交、构建执行、开发自测等研发过程的可量化研效数据，驱动团队实施过程改进分析、落地优秀工程实践、清理技术债务、优化资源配置以及客观评价团队成员的产出与成本。

团队产能分析

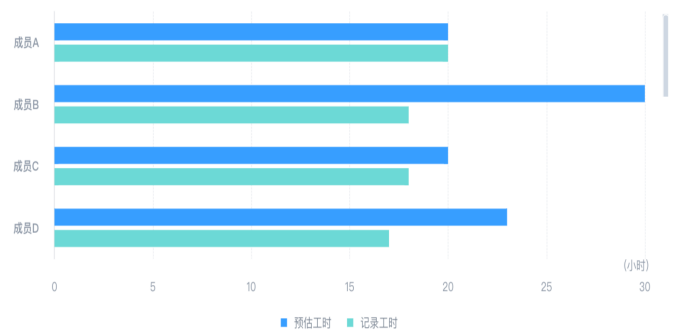
用户	提交总代码行(行)	新增需求(个)	已完成需求(个)	新增缺陷(个)	已修复缺陷(个)	记录工时(小时)	故事点(个)	部署频率(次)	构建频率(次)	自动化测试用例数(个)
成员A	3733	12	2	4	1	32.00	21.00	4	9	5
成员B	3733	12	2	4	1	32.00	21.00	4	9	5
成员C	2342	31	12	3	1	21.00	11.00	3	32	23
成员D	2342	42	31	312	124	212.00	21.00	3	22	2
成员E	2313	32	12	45	14	54.00	42.00	3	2	23

共 6 条 < 1 > 10 条/页

代码提交排名



工时排名



项目交付价值流分析

基于精益理论，通过需求吞吐量、交付周期等可视化指标帮助团队洞察项目交付过程中的瓶颈和风险，在迭代回顾时进行有效总结复盘，找到项目交付的瓶颈点，持续提升价值流动效率。



个人效能改进

通过个人维度的度量分析视图，帮助开发者全面、准确了解自己的开发效率、开发质量和价值产出，明确自身工作优势和待改进点，自驱提升。

个人工作概览

待处理需求

2个

环比 -33.33%

处理中需求

2个

环比 -33.33%

逾期需求

2个

环比 -33.33%

待处理缺陷

1个

环比 -50%

处理中缺陷

1个

环比 -50%

提交代码行

400行

环比 +33.33%

记录工时

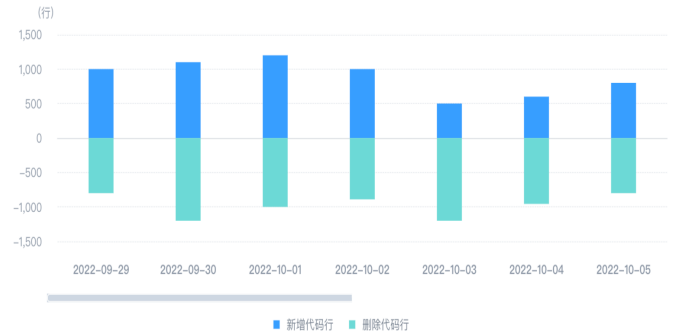
3小时

环比 -25%

代码提交频率



代码行变更趋势



产品优势

最近更新时间：2023-09-11 16:05:12

提升研发效能

CODING 与云端优势相结合，依托业界敏捷项目管理与 DevOps 体系方法融入到产品中，打通研发过程中的工具链孤岛及协作壁垒，覆盖敏捷开发全生命周期，帮助团队实现需求、迭代、开发、测试、持续集成、持续部署全方位研发管理，提升软件研发效能。

支持双态研发体系建设

CODING 适用于不同规模的开发团队以及不同类型的软件开发模式（如瀑布模型、敏捷模型），满足多业务场景的协作需求。

项目 workflow 和度量数据可视化

CODING 提供可视化看板，支持对代码、项目进度、人员工作量等不同维度输出详尽的数据报告，为团队管理者提供决策依据，调整项目计划和合理安排研发人力。

无缝集成第三方平台

CODING 支持无缝集成 GitHub、GitLab 等第三方代码库及各类常见的运维系统和云原生环境，让用户实现跨平台的无缝迁移。

应用场景

最近更新时间：2023-09-11 16:05:12

企业级 DevOps

通过腾讯云提供的一站式 CODING DevOps 工具链快速打通从需求到部署的端到端研发全流程，实现需求、代码、制品、CI/CD 之间的无缝衔接，助力跨职能之间高效协同，保证软件稳定、持续交付。

敏捷研发管理

腾讯云 CODING DevOps 深度结合敏捷研发理念。从用户故事开始，到需求池管理以及任务拆解、缺陷管理、测试管理，敏捷开发过程都有序建立，帮助企业以更低的预算、更快速的迭代方式验证产品版本投入市场的效果。

DevSecOps

将腾讯云 CODING DevOps 提供的自动化代码扫描、制品扫描、质量门禁等能力与 CI/CD 进行集成，打造从编码、构建、测试到部署的自动化安全检测链路，实现“质量+效率”双保障。

云原生实践

腾讯云 CODING DevOps 提供以应用为中心的云原生应用全生命周期管理工具，提供对开发者友好的应用建模、应用发布、应用观测等一站式应用管理能力，助力企业降低应用云原生化门槛，提升应用的交付效率和可靠性，打造系统化的应用运维管理能力。