



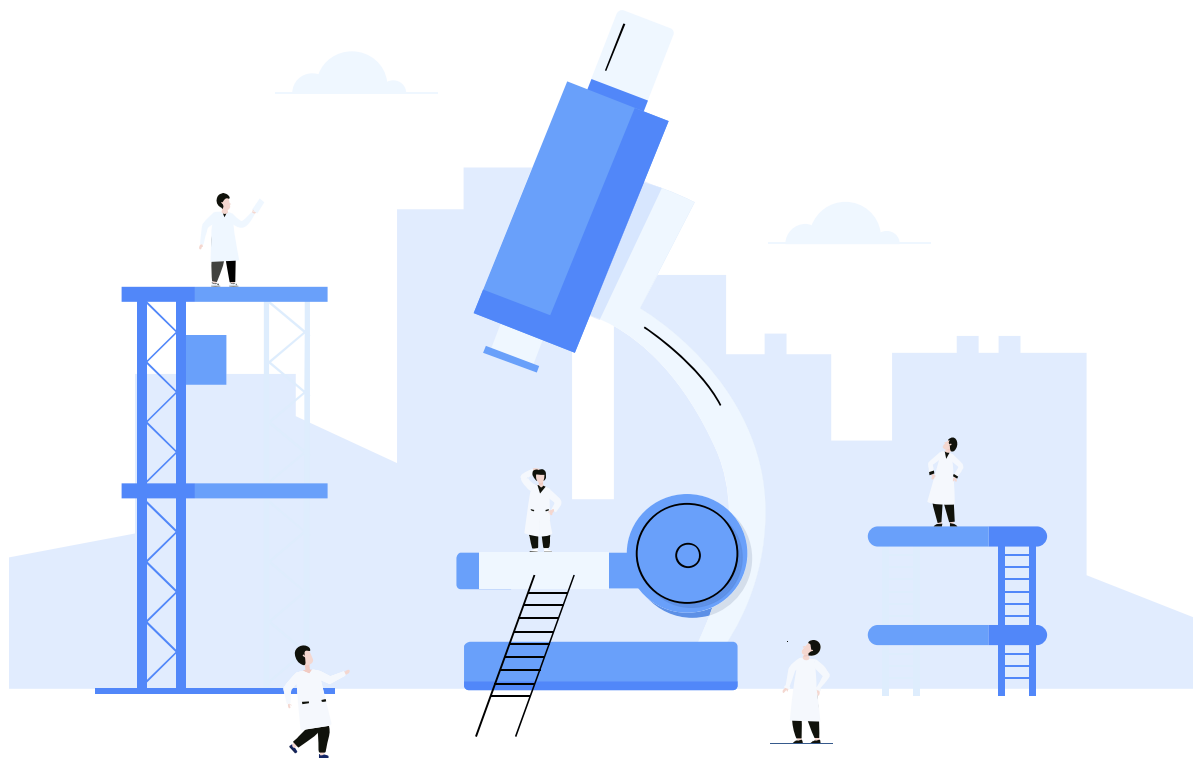
InfoQ 极客传媒

# 中国企业 研发高效能白皮书

## —CI/CD 篇



# CREATIVE



## 中国企业 研发高效能白皮书

- 🔑 近年来中国企业研发正在从粗放型走向精益型，研发工作的“高效能”成为几乎每个研发团队共同的追求。
- 🔑 中国软件服务产业也在近5-10年中得到了飞速发展，技术服务的边界不断拓展，赋能研发高效的产品层出不穷，适合中国研发环境的技术服务体系在不断完善。从结果上看，中国企业正在高效能研发的路径上快速前进。
- 🔑 本报告以中国高效能研发企业为研究对象，尝试解读市场中具有代表性的高效能研发解决方案。本次报告由五个篇章组成，如CI/CD、企业级架构、价值流管理等主题。研究小组期待可以通过研究，帮助中国企业研发团队获得高效能研发新知。



# 目录

# CONTENTS



## CI/CD概念和背景介绍

CI/CD的概念——关于CI/CD  
CI/CD发展的阶段——历经三阶段，平台型产品站上历史舞台  
CI/CD的价值——CI/CD赋能研发的四大抓手  
CI/CD的流程结构——CI/CD流程结构创新赋能研发效率



## CI/CD行业发展概况

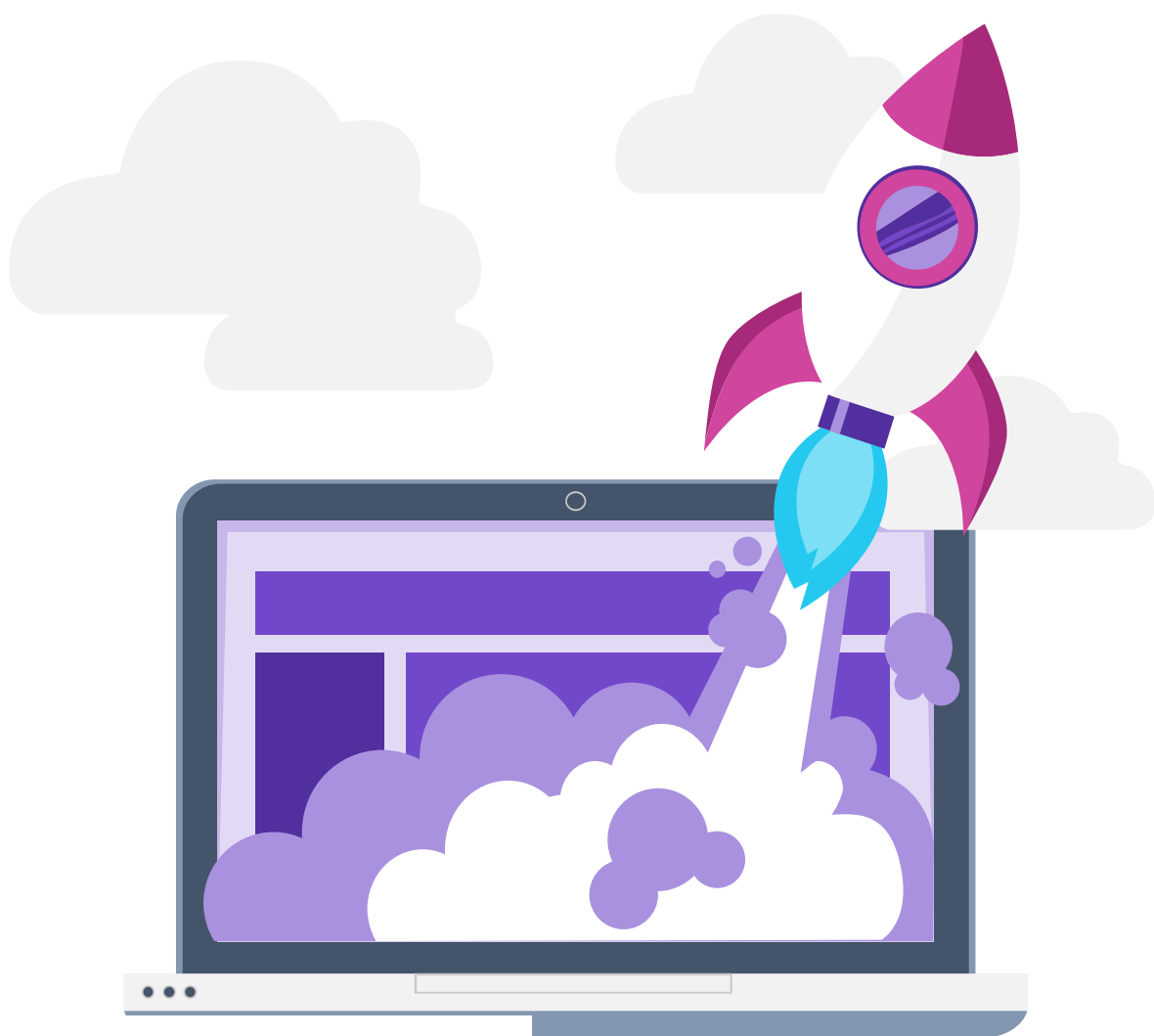
CI/CD市场基础数据——CI/CD市场蓬勃发展  
CI/CD市场厂商分布——CI/CD行业厂商矩阵逐步形成  
CI/CD市场特征洞察——行业发展推动市场需求呈现新特征



## 极狐GitLab CI/CD：带您开启新世界大门

极狐GitLab CI/CD简介——GitLab CI/CD使用前和工作流  
极狐GitLab CI/CD特征——极狐GitLab CI/CD四大核心优势  
极狐GitLab CI/CD真实用户使用反馈

# CI/CD 概念和背景介绍




# 关于CI/CD

研发团队提到降本增效，最直接想到的解决方案就是将一些繁琐、易错的工作，改造为持续自动化交付流程。而在研发团队传统交付过程中，“代码集成（Continuous Integration即为CI）”和“代码交付（Continuous Delivery即为CD）”可以说是很多团队最为头痛的问题。

通常研发团队中会设置一位同事或者一个团队专门负责代码集成和交付工作。但往往因为项目规模、构建环境、大量细微错误积累等复杂因素，项目的集成和交付经常会成为压垮开发团队的最后一根稻草。

有研究表明在软件开发早期，手动测试过多是导致开发团队开发效率低的主要原因之一。自动测试比较少，研发过程缺乏构建和部署自动化，导致了研发团队手动测试容易出现大量问题。统计数据表明，62%的研发团队被这样的问题困扰。



“所以，软件集成是一件高风险的、不确定的事情，国外甚至有个专门的说法，叫作‘集成地狱’。也正因为如此，人们就更倾向于不做集成，这就导致开发末端的集成环节变得更加困难，从而形成了一个恶性循环。”——开发者

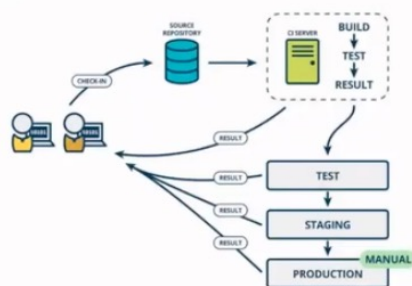
# 关于CI/CD

CI为 Continuous Integration 的英文缩写，意为持续集成。CD最早为 Continuous Delivery的英文缩写，意为持续交付；后来也被认为可以代表Continuous Deploy，意为持续部署。如今，CI/CD一般为持续集成、交付和部署的简称。



- “CI (Continuous Integration) 源于肯特·贝克 (Kent Beck) 在1996年提出的极限编程方法 (Extreme Programming)。顾名思义，极限编程是一种软件开发方法，作为敏捷开发的方法之一，目的在于通过缩短开发周期，提高发布频率来提升软件质量，改善用户需求响应速度。”

- CD (Continuous Delivery) 最早出现于2010年，Jez Humble和David Farley在《持续交付》一书中提及和定义了CD：持续交付是“一系列的原则与实践的集合；通过这个集合，团队能够在低成本、短时间及低风险的状态下以增量方式将软件变更交到用户手上。”持续交付部分由版本控制、制品管理、变更管理、构建、持续集成、自动化测试、部署管理、流水线、持续交付工具等方面组成。



# 关于CI/CD



- ▶ CD经常被认为也同时代表持续部署即Continuous Deploy。持续部署更进一步。一旦构建在staging环境中通过了自动测试，持续部署系统将会自动将它部署到生产服务器上。换言之，每个通过测试的构建都是实时的，可供用户及早反馈。这使得团队可以不断发布新特性和修复bug，并以其测试流程提供的保证为后盾。所以，很多时候CI/CD是指代通过连续方法进行软件开发的工程方案。
- ▶ CI/CD这套软件工程思路和原则诞生后经常被同时提及，是更为优秀的软件开发方法，能够大大提高团队的工作效率和交付水平。持续集成（CI）是构建软件、完成初始测试等一系列工作，是通过小步快跑的方式，不断优化集成方式的过程。持续部署（CD）是将代码与基础设施相结合的过程，确保完成所有测试并遵循策略，然后将代码部署到预期环境中。

# 强痛点，推动CI/CD发展

If it hurts, do it more often, and bring the pain forward.

——Martin Fowler（马丁·福勒，软件开发方面的著作者和国际知名演说家）

集成、部署和交付的强痛点，推动了CI/CD近十年的快速发展。Martin Fowler认为如果痛点过于明确，那么就请频繁地去实践，并且将痛苦前置，以便我们更好地解决痛点，达到更接近预期的研发效果。

“该项目属于维护性质（在已有项目基础上开发），特征比较明显，客户也比较随（you）和（qian）。项目要求在两周内发布一些新特性。要上线的时候我才发现，团队里面居然没有人做过部署，做过的也只是通过WinSCP的工具进行拖拽。（团队没有经验的后果就是）部署10次有9次跑不起来（剩下的一次是我部署的）。”

“由于客户业务发展的需求和相关安全性考虑，决定半年后升级现有Java版本和中间件。除了升级中间件，还需要通过重构改善现有系统的性能，最后的测试时间只有一个月。‘编码一时爽，重构火葬场’，做过开发的同学应该都有深刻的体会，没有测试的重构，想想都头皮发麻，修Bug的味道我知道。”

——开发者





# 历经三阶段， 平台型产品站上历史舞台

从理念诞生开始，全球主要的研发团队即开始探索如何将CI/CD更好地在实际研发场景中帮助团队降低研发成本和提升研发效能。目前，CI/CD的实践分为三个阶段：

## 01 开源软件

大部分研发团队对于CI/CD的初体验多是使用开源软件/开源项目。开源项目一般解决的是行业某一个通用的痛点，行业共性需求很强。在不复杂的开发环境中，解决一些通用问题，开源软件的效果还是很容易看到。但是在复杂和具有企业个性的开发环境中，开源软件就会展现出来一些劣势。

“为了项目的顺利进行，我用某开源软件部署了人生中的第一条CI流水线，整个CI流水线只有两个功能，打包和部署环境。花费的时间从原来的1人日/部署，缩短到了10分钟/部署，我尝到了CI的甜头，真香啊~”


——某金融研发团队Leader



## 02 开源软件+其他软件协同


随着研发团队交付复杂程度和交付品质要求的不断提升，通用型解决方案逐渐开始面临挑战。面对个性化开发环境需求和具体的客户需求，研发团队通常无法通过单纯的人工审核流程进行满足。






（接上页）与此同时，人工审核流程还会消耗大量的团队耐心和时间，为团队整体研发成本增加负担。此时，能够连接开源项目的其他协同软件成为降本增效不错的选项。如通过协同静态扫描集成帮助研发团队把更多的精力投入到业务层面的确认，显著地提升整体研发质量。

## 03 平台型产品



随着中国企业数字化进程的不断推进，企业在使用CI/CD工具时的场景逐渐趋于复杂。行业的多样性也不断展现出来。因此，研发团队日常工作开始承接难度更高的跨功能、跨团队、跨行业的需求。团队在研发过程中各种插件的兼容性和适配性为团队的稳定交付带来不确定性挑战。

一体化平台产品的诞生满足了这种市场需求，大大降低了插件间适配度产生的时间成本和适配过程中出现事故的风险成本。



经过三阶段的发展。目前，具有解决方案平台能力的产品正在不断替代此前两阶段的CI/CD成为头部企业、行业引领者的共同选择。

# CI/CD赋能研发的 四大抓手

- ▶ CI/CD通过工作流程更新，将大量复杂的工作切分为可以控制或者可以修复的小单元工作。
- ▶ 在没有实践CI/CD的团队中，通常需要花费大量时间，由专门的高水平开发者或者团队去统一集成和部署。导致研发团队在人力成本和时间成本上都承担了更高的成本和压力。CI/CD通过增加集成和部署频率，降低单次集成和部署的复杂程度，从而提升了单次集成和部署动作的准确性。

## 持续性 01

- ▶ CI/CD通过程序帮助研发团队解放专职的集成和部署人员。开发人员提交新代码之后，程序立刻自动进行构建、测试、安全扫描等多项工作。因为这些新提交在最终合并到主线之前，都需要通过编译和自动化测试流程进行验证。这样做是基于之前持续集成过程中很重视自动化测试验证结果，以保障所有的提交在合并主线之后的质量，对可能出现的一些问题进行预警。

## 02 自动化

# CI/CD赋能研发的四大抓手

## 可追溯 03

- ▶ CI/CD小单元的持续集成、部署和交付，将原本超出控制、高概率不可修复的流程前置，将更多可能出现的问题提前解决。并且所有开发人员均可以追溯开发流程，更精准地帮助项目内容的完成。

## 04 高效迭代

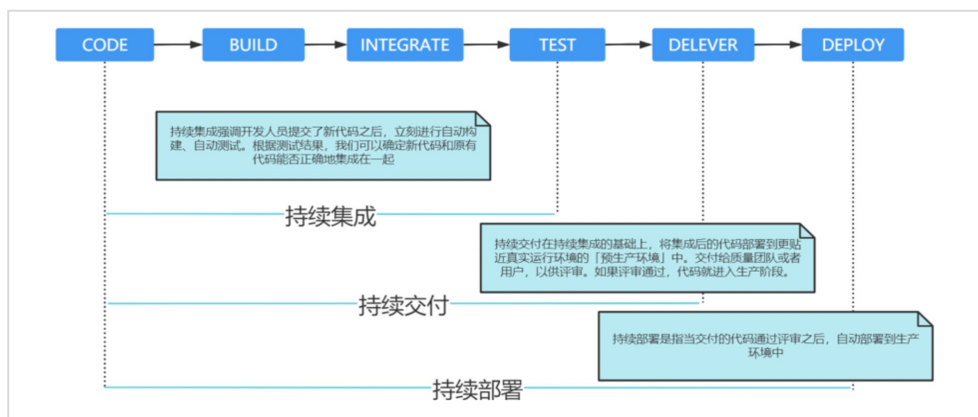
- ▶ CI/CD通过持续性、自动化、可追溯的实践，在可控范围内不断降低项目研发的机会成本，在宏观层面帮助项目研发更高效迭代。



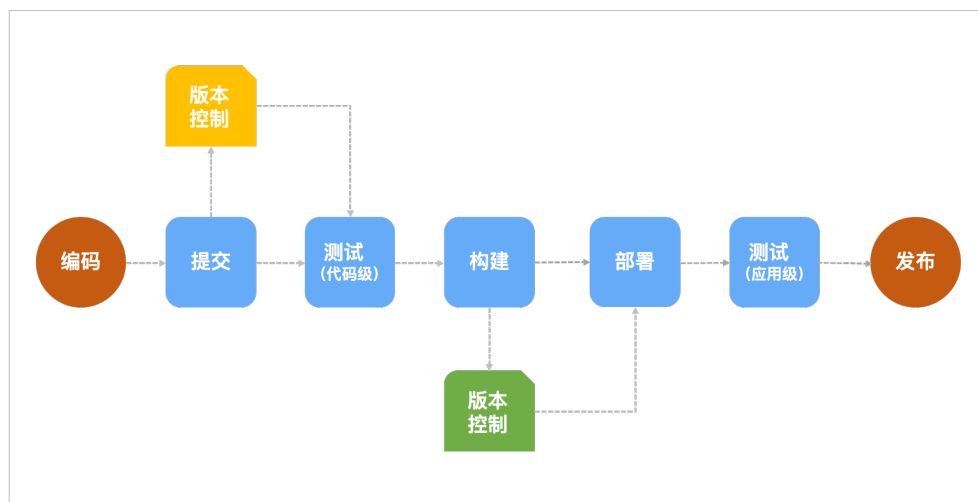
# CI/CD流程结构

## 创新赋能研发效率

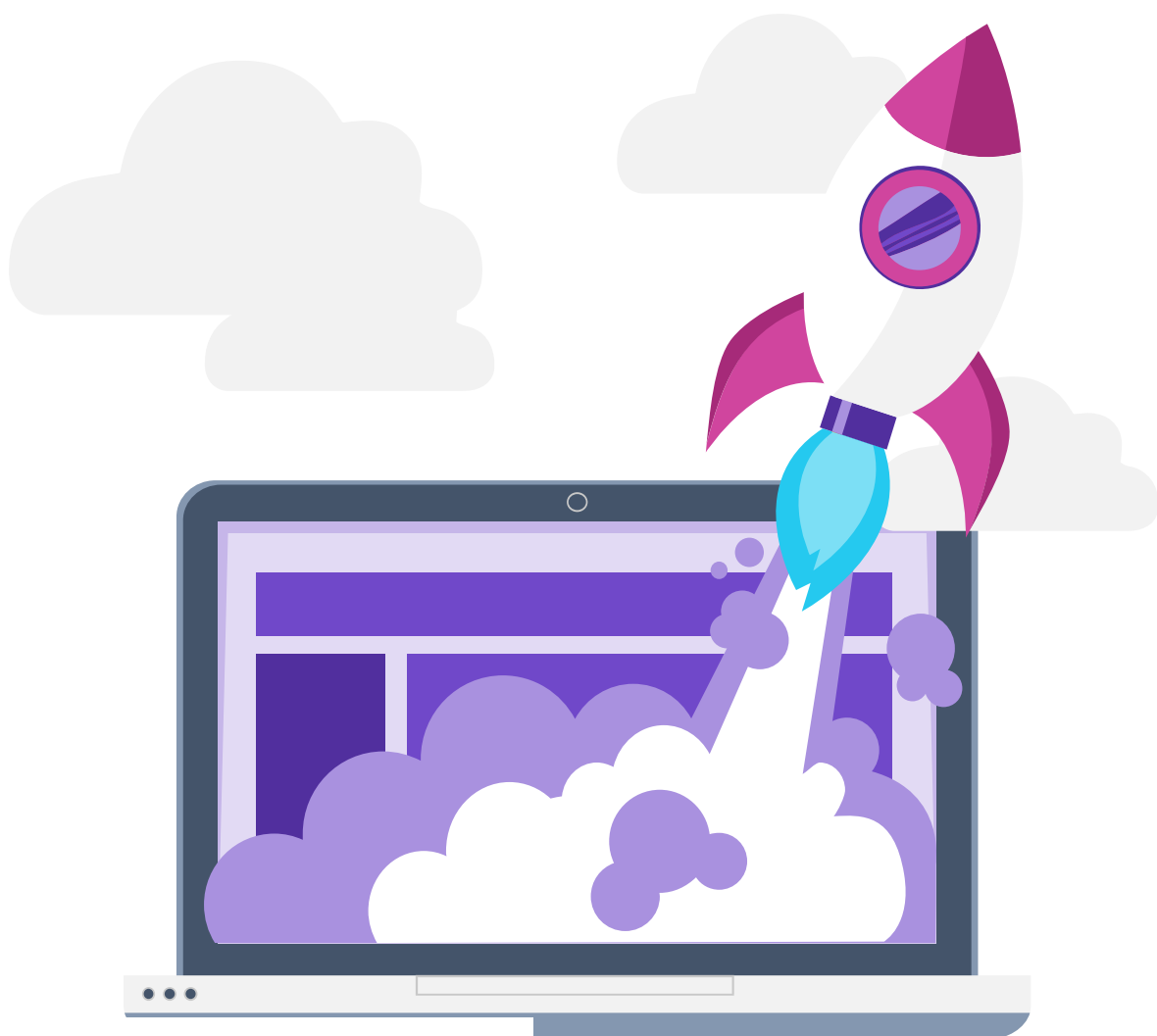
CI/CD的核心流程主要包含编码、提交、版本控制、测试（代码级）、构建、制品中心、部署、测试（应用级）和发布等。CI（持续集成）主要包括编码、构建、集成和测试；CD（持续交付和持续部署）在CI的基础上将代码部署到更为接近真实运行环境的预生产环境中，并在代码审核通过之后，自动部署到生产环境中。



在整体的CI/CD过程中测试步骤也可以前置到合适的位置。如在极狐GitLab中即将测试分为代码级测试和应用级测试，代码级测试被前置于代码提交后和构建之前，进一步确保整体交付和部署的准确性和持续性。



# CI/CD 行业发展概览



# CI/CD市场蓬勃发展

## 13 亿美元规模

Market Growth Reports 2022年的统计数据，2017年CI/CD市场规模已经达到4.02亿美元，2023年市场规模有望达到13亿美元。

3.2倍规模增长

## 85.16%

国内调查显示，持续集成、自动构建、单元测试和持续部署四种敏捷工程实践，占比分别为85.16%、81.61%、81.53%和80.66%。

认可度高

## 90% 使用持续集成

19.38%的企业拥有独立的持续集成团队维护持续集成系统；25.74%的企业实现了持续集成平台的自服务化，另有24.80%的企业通过自服务化持续集成平台实现组织级交付能力的提升，持续优化和改进团队的持续集成服务。

普遍使用

## 3.78% 手工部署

软件质量被企业持续关注，集成问题普遍在1天内完成修复。如今仅有3.78%的企业手工完成所有环境的部署。

更自动化

# CI/CD行业厂商矩阵 逐步形成

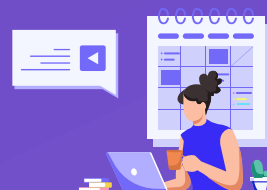


CI/CD市场需求的不断提升，离不开国内外众多CI/CD厂商的持续创新和市场推广。目前，CI/CD已经成为DevOps最具规模的技术应用之一。目前国内外已经形成以极狐GitLab CI、Jenkins、Bamboo CI、Circle CI、TeamCity等为代表的厂商矩阵。市场中的主要厂商业务可以分为三种类型：

以 极狐GitLab CI /CD 为代表的DevOps平台型工具产品厂商

以 Jenkins、Bamboo CI 为代表的单一工具链产品厂商

以 AWS CodePipeline 为代表的企业级厂商（与自身产品绑定）



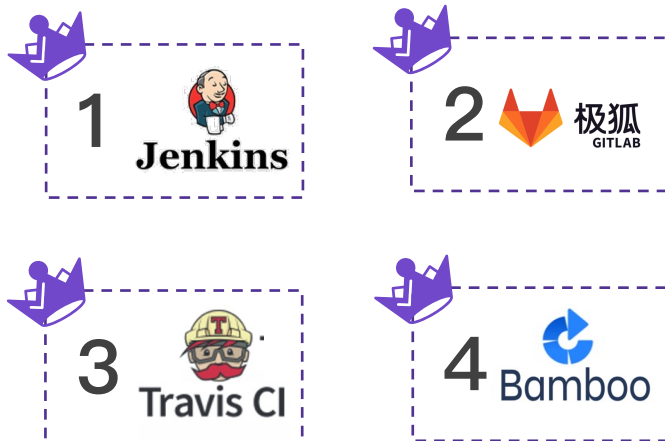


# CI/CD中国市场格局

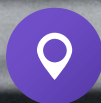
DATA\_

## Jenkins、极狐GitLab CI/CD 呈现领先厂商姿态

### 中国持续集成与流水线 企业使用率



数据来源：《中国信息通信研究院》



将精英组织和低效组织相对比，  
精英组织使用CI/CD更多，并具有以下特性：

## 精英效能

**973X**

更加频繁  
部署代码

**6570X**

更短的前置时间  
从代码提交到部署

**3X**

低于1/3的  
变更失败

**6570X**

更快的故障修  
复时间

# 行业发展推动市场需求 呈现新特征



## 开箱即用

随着国内数字化水平的不断推进，研发团队不仅仅出现在互联网和科技企业中，也正在成为更多行业中的中流砥柱。随之而来的，越来越多行业中的企业和机构（如金融、医疗、交通等）也正在使用CI/CD工具帮助其提升研发团队效率。对于新行业中的新企业，研发团队的技术能力较传统互联网和科技企业较低，对研发工具的易用性要求更高。同时，因为更多企业对研发工具易用性的投入，也推动了CI/CD工具进入更多的行业和企业研发团队中。开箱即用的产品一般需要具备兼容性高、学习成本低、可追溯等特征。

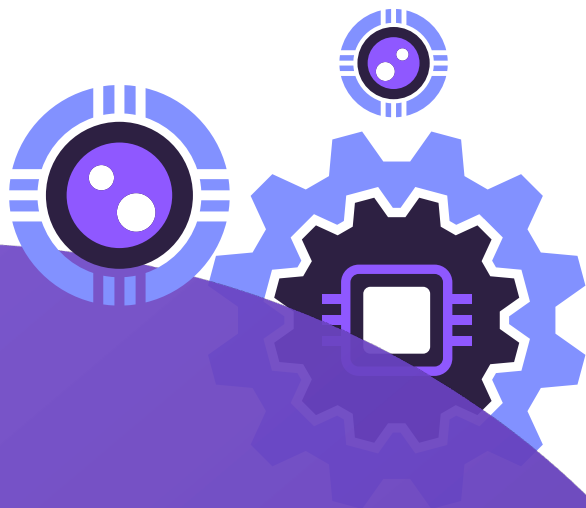


## 安全性能

一方面随着行业的拓展，研发团队的稳定性和安全性被赋予了更高的要求。如金融企业天然需要研发过程运行速度更快，软件稳定性更强。

另一方面，研发工作中随着代码规模、并行需求的不断增加，研发的复杂性进一步被提高。因此CI/CD的安全性要求在整体工具选择中的权重不断被变大。工具是否能够帮助企业保障研发安全，降低出现重大错误的隐患的需求不断被放大，成为众多企业选择CI/CD工具时最重要的因素之一。

# 行业发展推动市场需求 呈现新特征



## 云原生转移

全球软件正在经历云原生的洗礼，国内主要的企业也逐渐在进行上云和云原生的动作。大部分互联网企业和科技企业，以及相对比较领先的传统大型企业的云原生环境基本构建完成，云原生软件在研发工作中的使用比例快速升高。CI/CD工具在被选择时，能够与主要云原生软件平顺连接，成为考虑TOC的重要因素。



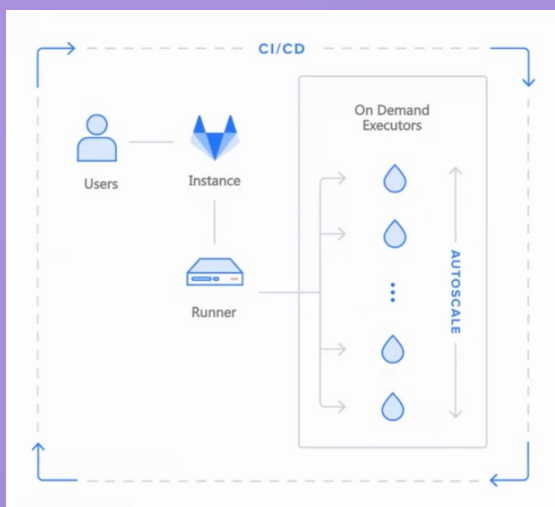
## 业务洞察

随着软件科技的不断演进，更具有业务思维的研发软件逐渐被市场所青睐。在CI/CD工具中，为满足市场不断变化的需求，降低技术专业人士的学习和工作成本，以及满足更多业务侧的非技术专业人士在研发领域的工作需求。具有业务视角的可视化观察和统计工具成为新的产品功能形态。

# 极狐GitLab CI/CD: 带您开启研发高效能



# 使用极狐GitLab前



极狐GitLab CI 运行前，需要提前安装Instance、Runner和Executor。

## Instance

提供代码托管、任务队列、制品存储等服务



## 极狐GitLab Runner

由Go语言编写，支持在多种平台运行



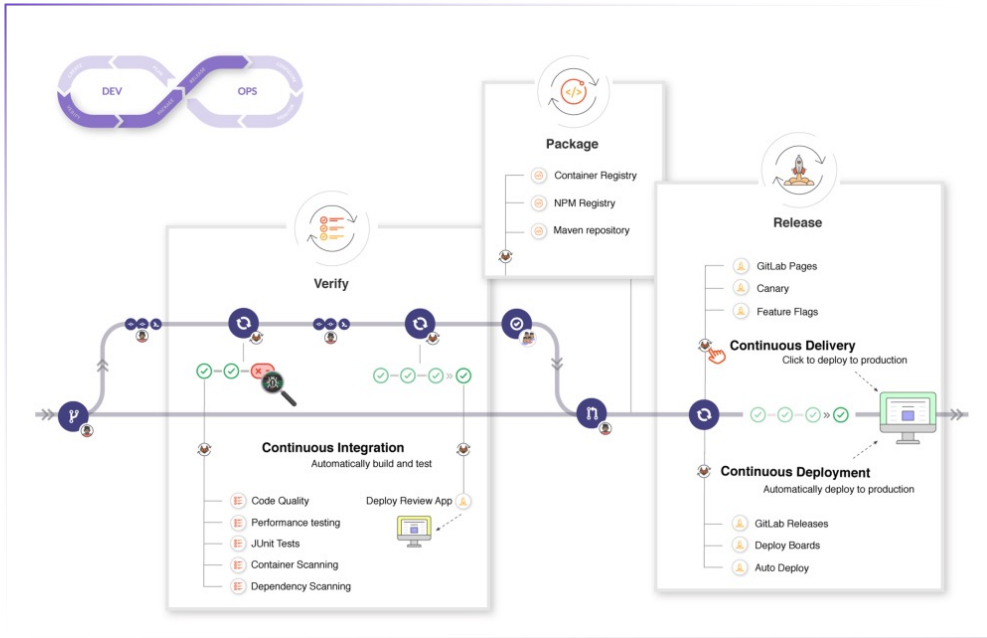
## Executor

CI/CD流水线中具体任务的执行器



- 操作系统：Linux、Windows、macOS、FreeBSD
- 底层架构：x86、AMD64、ARM64、ARM、s390x、ppc64le
- 安装方式：容器化、二进制、软件包

# 极狐GitLab CI/CD



# 极狐GitLab CI/CD 四大核心优势



## 1

### 一体化平台，一站式体验

极狐GitLab CI/CD属于极狐GitLab一体化DevOps平台的一部分，无需安装运维第三方工具链，节约运营成本，而且还可以和其他功能诸如项目管理、SCM、DevSecOps等打通，加速软件的持续交付。

### 简单易用，便捷高效

极狐GitLab CI/CD无需安装配置第三方工具，只需要配置 .gitlab-ci.yml 文件即可，节约了众多工具链的学习、运营及时间成本。

## 2

## 3

### 数据可视，监控优化

极狐GitLab CI/CD在赋能高研发效率和高安全性的同时，在可视化方面也下足功夫。在极狐GitLab中不仅可以做到CI/CD Pipeline构建过程的可视化，还可以做到Runner的可视化。

### 安全构建，安全交付

极狐GitLab DevSecOps的七大功能能够直接集成到CI/CD中，为应用程序提供构建到上线的全流程安全保障能力，同时能够做到安全报告的统一展示透明、安全漏洞的修复管理透明。

## 4



# 一体化平台，一站式体验



## 数据反馈，研发闭环

极狐GitLab CI/CD的构建结果以及报告（测试、安全等）直接反馈到MR（Merge Request）中，打通了各个环节的数据流，避免了数据孤岛的形成，同时为研发、代码审核人员提供代码是否能够合并的决策能力，从源头保证代码质量以及应用程序安全。



## 场景无关，通用性强

极狐GitLab CI/CD与环境（云原生非云原生）、研发模式（敏捷、瀑布）、架构方式（微服务、单体服务）无关，能够无差别地适用各种场景。



## 持续迭代，功能增强

月度发版机制，每个月28号发布新版本，每个版本都会有大量与CI/CD相关的功能特性通过不断地持续迭代，增强CI/CD的功能。



## 2 简单易用，便捷高效

### 01

#### YAML配置，快速上手

所有流程通过YAML配置即可，内置多种关键字，研发、运维均可快速上手。



### 02

#### 模版沉淀，提高复用

经过多个版本的迭代，沉淀出了众多模版，可以在创建 `.gitlab-ci.yml` 文件的时候直接选择对应的模版，也可以选择通过include语法来实现模版的引用，实现模版的即插即用和提高复用。



# 简单易用，便捷高效

## 03

### 功能众多，高效敏捷

极狐GitLab CI/CD提供多种流水线（Pipeline），不同类型的Pipeline可以适用于不同规模的团队、不同的研发模式、不同的应用架构。满足研发团队从基础流水线到各类复杂和独特的流水线需求。极狐GitLab CI/CD流水线（Pipeline）包括：

#### 基础流水线（Basic Pipelines）

流水线步骤顺序执行，步骤内任务并行执行

#### 多项目流水线

##### （Multi-project Pipeline）

跨项目流水线触发与关联

#### 合并请求流水线

##### （Merge Requests Pipeline）

针对合并请求触发和运行的流水线

#### 合并列车（Merge Trains）

使用合并结果流水线触发合并请求

#### 有向无环图流水线

##### （Directed Acyclic Graph Pipeline）

流水线步骤顺序执行，步骤内任务可跨步骤串行

#### 父子流水线（Parent-Child Pipeline）

在同一个项目中根据触发规则配置多条不同流水线

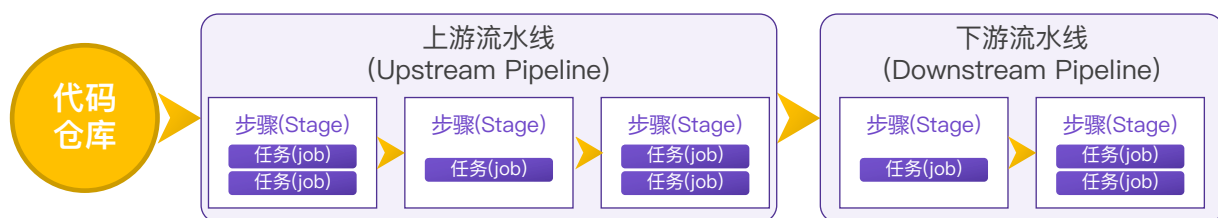
#### 合并结果流水线

##### （Merged Result Pipeline）

针对合并请求完成后触发和运行的流水线



## 简单易用，便捷高效



### 03 功能众多，高效敏捷

通过使用CI/CD策略确保安全合并更改和绿色主机，企业就无需太担心由于基础设施故障和代码冲突导致的意外停机。有了更高效的流水线，开发人员能够减少花在维护上的时间，将更多的时间和空间集中在创造富有想法和创新力的工作上，从而提升代码质量，改善公司氛围并提升整体士气。

不同类型的流水线可以有针对性进行灵活编排，  
从而提高研发团队工作效率。

- 针对大型代码库、针对文件变更触发流水线等具体研发情况，利用父子流水线可以帮助研发团队降低非关联任务依赖、动态按需触发子流水线以及提高流水线可视化程度。
- 如大型跨项目软件产品和微服务架构应用，利用多项目流水线可以帮助研发团队实现流水线上、下灵活定义、自动触发依赖流水线以及推动流水线全景可视化展示。



## 3 数据可视，监控优化



### 01

极狐GitLab CI/CD在赋能高研发效率和安全性的同时，在可视化方面也下足功夫。在极狐GitLab中不仅可以将CI/CD Pipeline构建过程可视化，还可以做到Runner的可视化。

### 02

多种维度的可视化，帮助项目执行人员、项目管理人员和项目协作人员均能在透明高效的环境中进行协作。可视化机制也为研发团队资源利用率的优化、构建时间的优化提供重要的决策依据。



## 4 安全构建，安全交付

01

### 实现“安全左移”

所有的安全扫描数据在经过整理后会嵌入到对应的MR中，方便研发在第一时间看到自己变更代码中的安全问题，及时修复，做到了真正的“安全左移”。

02

### 实现安全防护持续自动化

安全扫描嵌入CI/CD，每次变更都会执行安全扫描的自动执行，实现安全防护的持续自动化，提升效率。

极狐GitLab CI/CD

# 真实用户反馈

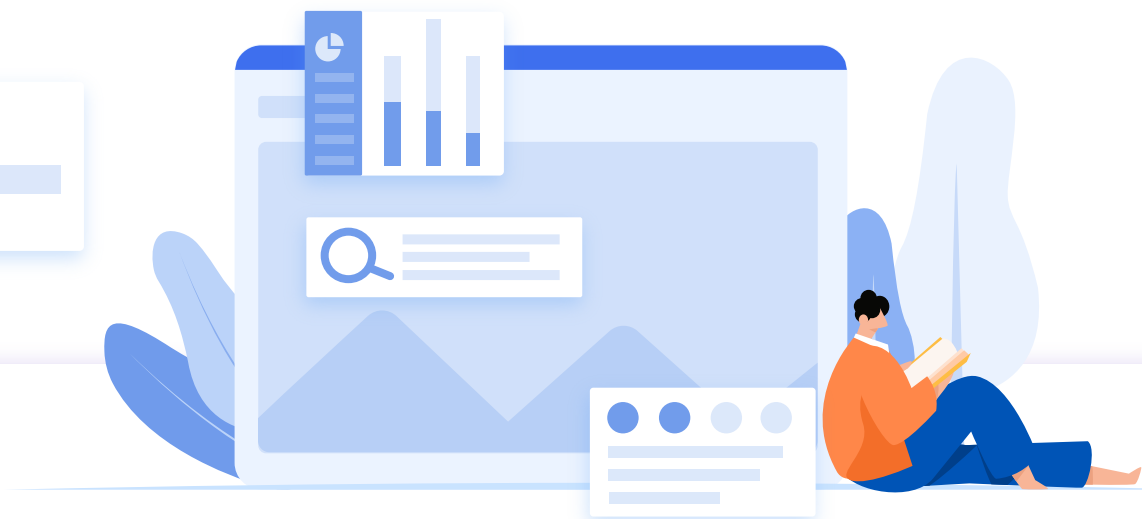
在极狐GitLab上，最初只配置了基础构建过程，一个团队成员发现Java Pipeline过程慢，就添加了依赖库的缓存配置，通过复用上一次CI过程中的依赖库，避免重复下载，实现提速。这个改动记录在 .gitlab-ci.yml 里，很快被发现并借鉴到其它项目中。即使内部没有做培训，但是团队成员都已经玩起来了，“只要有一个Java工程师掌握了这个能力，其他Java工程师就都掌握了这个能力”，“团队整体的端到端能力得到了很大的提升。”

——齐碳科技软件开发资深工程师李颂



极狐GitLab CI/CD

## 真实用户反馈



派拉之前大概有100+的CI/CD Pipeline，完全迁移到了极狐GitLab CI/CD并使用了template等功能对CI/CD Pipeline进行了优化。

极狐GitLab的CI/CD流水线保证了派拉产品能够及时构建和部署，在代码管理和合并审核管理上，集合CI/CD反馈的健康度指标，提升了产品代码的规范性和质量。

目前，极狐GitLab已经在派拉内部的所有产品线上使用，是保障派拉各产品线敏捷迭代、高效交付的重要支撑。“尤其是在今年上海疫情期间，我们被迫居家办公，基于与极狐GitLab的深度集成，我们解决了流程效率问题。结合派拉零信任产品，在疫情期间全员远程办公条件下，我们仍然保证了产品的双周更新迭代发布。”

——派拉软件研发总监茆正华



敬请期待 更多企业研发高效能白皮书篇章

