华为端到端DevOps概览



HUAWEI

前言

- 本章先从软件产业的现状、发展趋势及挑战，说明敏捷和DevOps必将成为业界主流的研发模式。接着介绍和阐述了敏捷和DevOps的关系，相关理念、价值和主要工程方法。最后重点介绍了华为端到端DevOps的实施框架及其主要内容。

HUAWEI

目标

- 学完本课程后，您将能够：
  - 了解软件产业发展趋势、挑战及软件研发模式演进方向
  - 理解敏捷、DevOps相关理念、价值和常用工程方法
  - 初步掌握华为端到端DevOps框架及其主要服务

HUAWEI

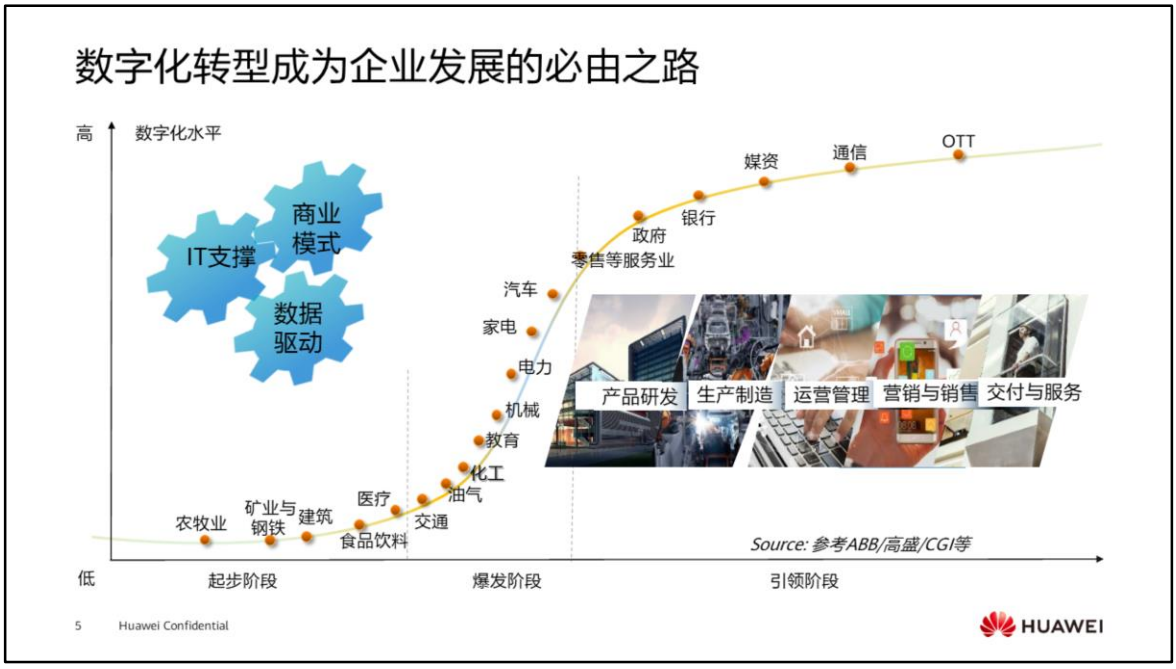
目录

1. **软件产业和交付模式发展趋势**
   - 软件产业发展趋势
   - 新形势下企业面临的挑战
   - 敏捷和DevOps逐渐成为主流交付模式
2. 新兴软件技术及交付模式简介
3. 华为云DevCloud HE2E DevOps框架及其主要服务

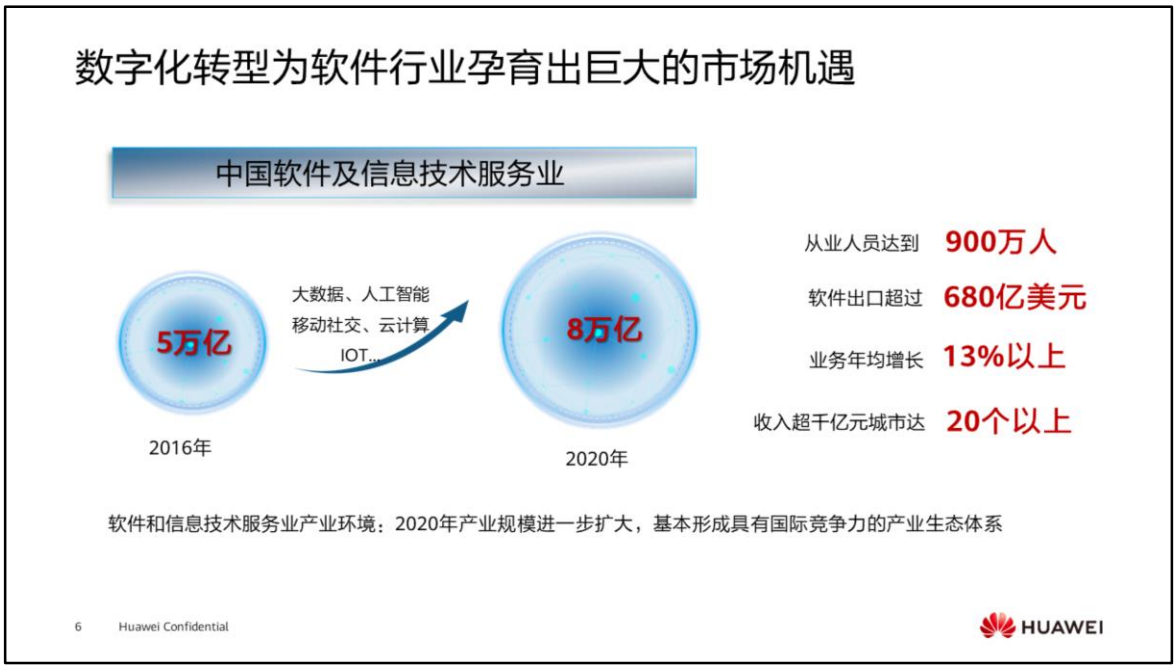
HUAWEI



- 随着云计算，IoT，AI等技术的蓬勃发展，我们正在进入一个万物感知、万物联接、万物智能的世界。云、大、物、智
- 企业在数字化转型的过程中不断收集和积累数据用于智能分析和精准控制以提升企业运行效率。
- 有两个实际例子：飞机引擎中的传感器每次飞行过程中平均产生超过1TB的数据，这些数据被用来诊断引擎的健康度，供引擎企业进行可预测性维护，降低维护成本，提升安全度。另外一个例子就是一个农场通过给奶牛带上项圈，读取奶牛的生物特征，识别出发情时间，提升奶牛配种的成功率和产奶效率。这两个处于不同的行业，生产不同的产品的企业，但是都在积极通过数字化来获取商业利益。

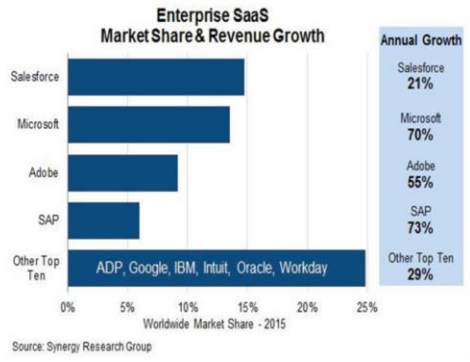


- 企业要数字化转型，才能具备“创新和敏捷”的业务能力，才能应对技术、需求和商业模式的变化，这是所有行业的企业发展的必经之路。
- 只是不同行业的数字化转型处于不同阶段，这里根据业界咨询公司的报告汇总了各行业数字化现状，互联网和电信运营商等信息服务提供商处于引领阶段， 紧随其后的是金融/政府等社会服务单位；汽车等机电企业处于爆发阶段，如车联网和自动驾驶；其他企业刚好处于起步阶段。
- OTT 是“Over The Top”的缩写，即互联网公司越过运营商，发展基于开放互联网的各种视频及数据服务业务，强调服务与物理网络的无关性。



- 企业在数字化转型过程中，必将面临海量的数据，如何有效地提取有效数据分析和处理，是大家面临的一个重要课题，针对这个课题，唯一解决方案就是软件，从中国软件行业的成长和发展趋势来看，数字化转型为软件行业孕育除了巨大的市场机遇，中国软件及信息技术服务业总产值将从2016年的5万亿增长到2020年的8万亿，年增长13%以上。就业人员将达到900万人，软件出口超过680亿美元。

全球软件产业正在向服务化转型



Salesforce：
- 自1999年起，Salesforce 一直致力于打造按需定制的全新SaaS商业模式，提倡"No Software"，是SaaS模式的鼻祖；
- 全球SaaS公司营收体量最大的公司,是云计算应用端的代表。2020财年营收171亿美元，增速 29%。

微软 Office：
- Office 2019 作为一个一次性的购买出售
- Office 365 是一种订阅服务，确保用户始终拥有最新的 Microsoft 生产力工具

据艾瑞咨询预测：
- 中国SaaS服务市场规模2021年将增长至654.2亿元
- 2018 年中国SaaS企业软件市场规模为243.5亿元，同步增长47.9%。

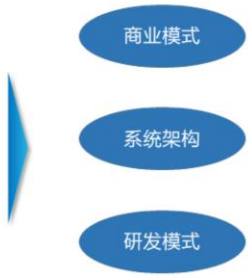
软件服务化需要软件企业关注长尾效应、专注业务升级、降低开发与维护成本，要求面向服务的方式构建软件，"多、快、好、省"地持续推出新服务或者升级现有服务

HUAWEI

- 软件产业面临巨大的市场机遇，它的发展方向和趋势是服务化。
- 据艾瑞咨询预测：到2021年，中国SaaS服务市场规模将增长至654.2亿元。而2018年中国SaaS企业软件市场规模为243.5亿元，同步增长47.9%。
- 软件产业向服务化转型趋势在两家典型的软件公司也得到了验证：
  - Salesforce，SaaS模式的鼻祖和云计算应用端的代表，2020财年其营收已达到171亿美元，增速 29%。
  - Salesforce作为CRM行业的服务提供商，拥有完善的云计算技术和经过专家设计的客户管理软件。
  - 再来看看微软的典型产品office，Office 也新推出在线订阅服务Office 365，并作为后续的主推方向。

云成为软件的普遍承载方式

- 来自Gartner的相关观点：
  - 到2020年，云计算将成为最主流的IT形态
  - 云计算已经成为增长最快的科技领域，整体增长速度为25%，而整体IT市场的增长率仅为1.1%
  - 新一代的硅谷明星企业，比如Airbnb、Pinterest、Lfyt等则无一不生长在云上



国家《云计算发展三年行动计划（2017－2019年）》指出：支持软件企业向云计算转型。支持软件和信息技术服务企业基于开发测试平台发展产品、服务和解决方案，加速向云计算转型，丰富完善办公、生产管理、财务管理、营销管理、人力资源管理等企业级SaaS服务，发展面向个人信息存储、家居生活、学习娱乐的云服务，培育信息消费新热点。

HUAWEI

- 服务化对软件的可靠性带来了更高的要求。以华为云为例，承诺的可靠性为5个9，即99.999%，也就是服务全年服务不可用或者服务性能下降的时间不超过5.256分钟。所以，IT形态云化也就成了必然的趋势，企业业务“服务化”、“云化”的趋势不可逆，公有云大势所趋，“多云”是主要方式。
- 从第三方咨询报告中，传递出来以下信息：
  - CSP的数据中心数量规模从增速和规模来看，比企业私有的DC越来越大。反映出越来越多的业务和数据都集中到了云中。
  - 云计算越来越成为云基础设施了，云的能力价值逐渐被企业所认可，成了企业业务的首选架构。
  - 云最大的安全疑虑也通过技术的成熟逐步的消除。
  - 软件服务化、云化对企业的商业模式、系统架构、研发模式都提出了新的要求和挑战。

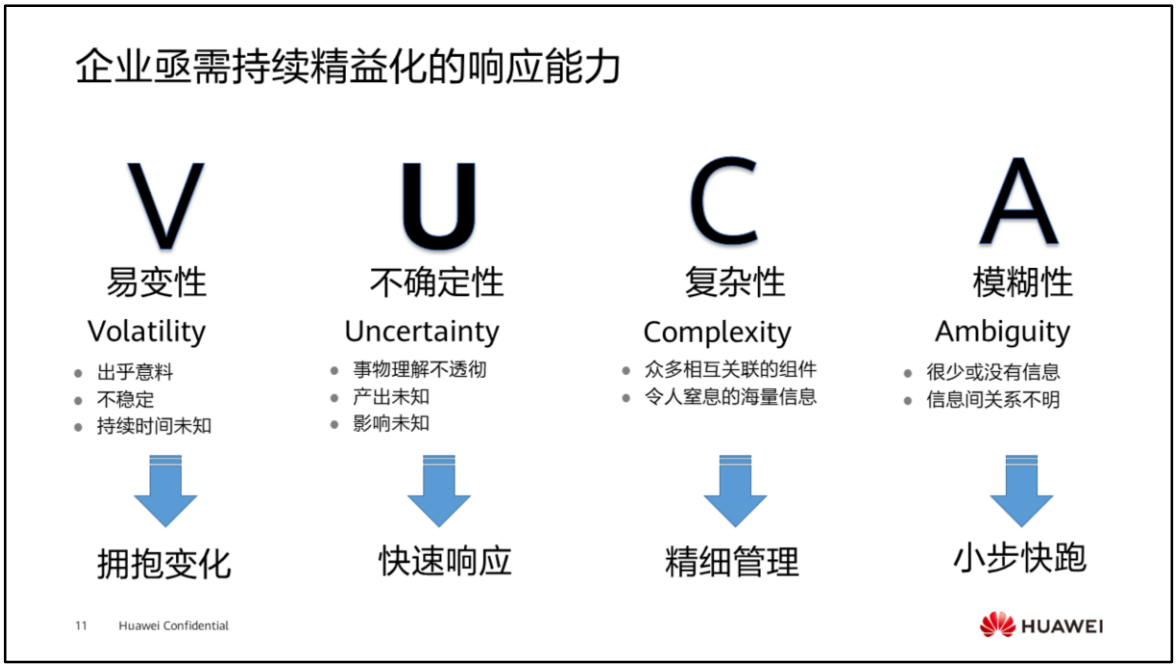
目录

1. **软件产业和交付模式发展趋势**
   - 软件产业发展趋势
   - **新形势下企业面临的挑战**
   - 敏捷和DevOps逐渐成为主流交付模式
2. 新兴软件技术及交付模式简介
3. 华为云DevCloud HE2E DevOps框架及其主要服务

HUAWEI



- 软件服务化和云化后，企业面对的客户群体的差异化需求呈现出爆炸式增长。
- 随着移动、社交、云计算、大数据、AI的快速发展和深度应用，颠覆式创新和跨界竞争不断加剧。对企业来说，既是机遇又是挑战，不管是颠覆别人还是被颠覆，企业都面临交付、协同、安全等多方面挑战、都要具备快速持续的创新和交付能力。
- 新形势下，企业必须快速高效地交付价值产品，才能抢占蓝海，获取最佳的商业回报，其次，软件系统复杂度不断增加、跨地域高效协作和多环境部署也是企业需要解决的重要问题，软件服务化对可靠性和安全性也提出了更高的要求，特别是核心数据更是企业的重要资产，不容闪失。
- MVP（最小可行产品）

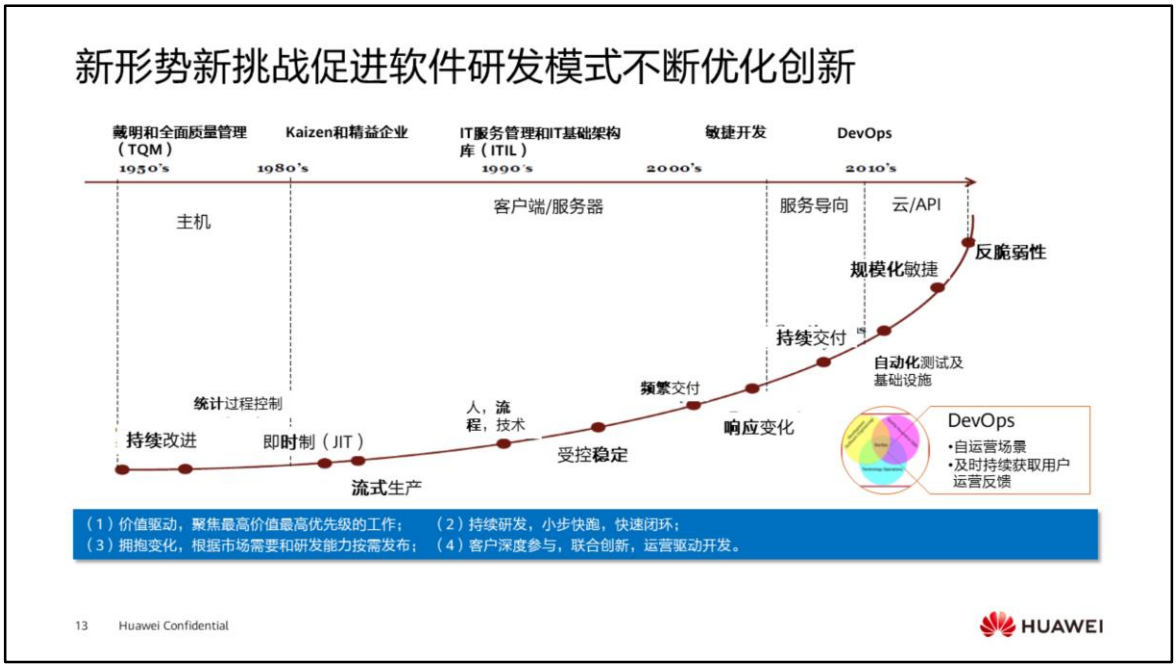


- 可以借用VUCA来归纳企业面临的挑战：
- VUCA的概念最早是美军在20世纪90年代，引用来描述冷战结束后越发不稳定的、不确定的、复杂、模棱两可的多边世界，后来，“VUCA”被战略性商业领袖用来描述混乱的和快速变化的商业环境。
- Volatility（易变性）是变化的本质和动力，也是由变化驱使和催化产生的。
- Uncertainty（不确定性）缺少预见性,缺乏对意外的预期和对事情的理解和意识。
- Complexity（复杂性）企业为各种力量，各种因素，各种事情所困扰。
- Ambiguity（模糊性）对现实的模糊，是误解的根源，各种条件和因果关系的混杂。
- 我们发现在复杂的系统中工作通常无法从局部行为来解释系统行为同样的事情做两次结果未必相同失败在复杂系统中会经常发生故障是不可避免的我们要做的不是问责而是快速转向学习型组织软件企业需要通过拥抱变化快速响应精细化管理和小步快跑来应对易变 不确定复杂和模糊的商业环境。

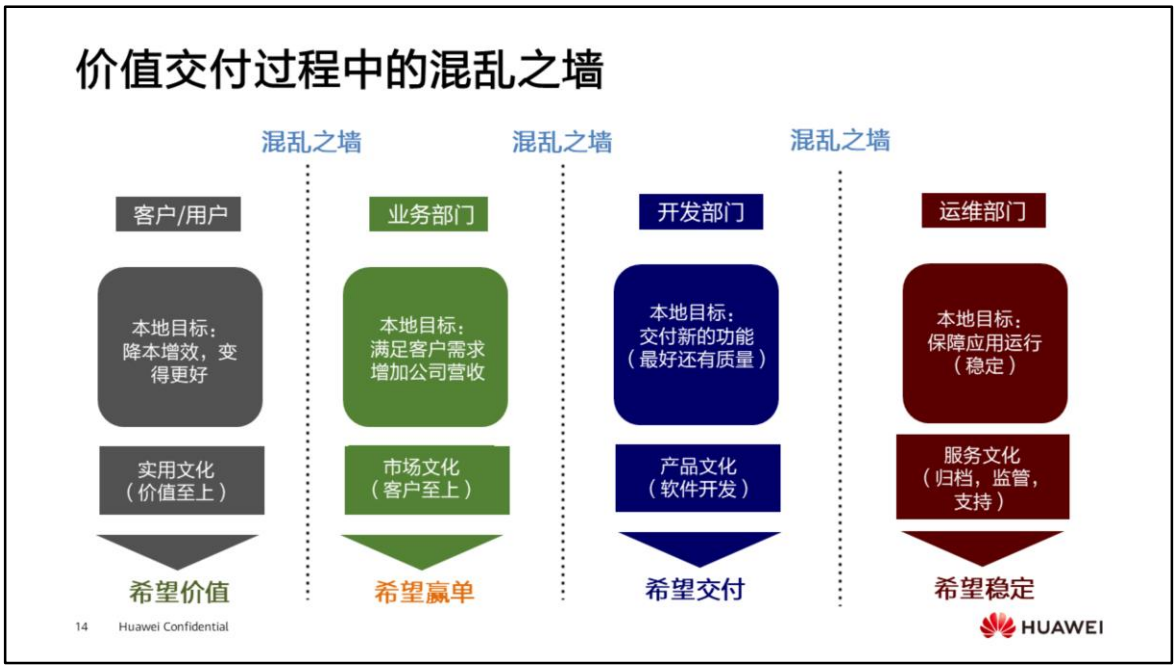
目录

1. **软件产业和交付模式发展趋势**
   - 软件产业发展趋势
   - 新形势下企业面临的挑战
   - **敏捷和DevOps逐渐成为主流交付模式**
2. 新兴软件技术及交付模式简介
3. 华为云DevCloud HE2E DevOps框架及其主要服务

HUAWEI

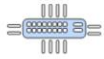


- 为了更好地应对新形势新挑战，软件企业的组织流程、研发模式不是一成不变，而是不断优化创新的。
- 如图是对组织流程优化历程的一个概览，我们可以看到在客户端/服务器年代的末期，敏捷开发开始出现，它主张频繁交付、响应变化，背景则是消费者市场的快速崛起。随后技术趋势向服务导向转变，业界开始主张更加持续的交付，以便满足消费者不断变化的需求和持续创新的需要。从2010年起，云计算走上舞台，进入了云和API技术主导的时代，DevOps也成为标示性的理念，强调研发和运维打通部门墙，更紧密地协作。与此同时，敏捷则开始走出只适用于小团队、小公司的迷思，为更大规模的企业所青睐，而能够指导大型企业实践敏捷的规模化敏捷方法论逐渐浮现并得到采纳。



- 从传统组织部门的职责和目标也能发现敏捷和DevOps产生的根本原因，那就是价值交付过程中的混乱之墙
- 首先，客户最关心的是获取价值，并且成本越低越好，业务部门希望能够满足客户的诉求，能够拿单来获取更大的营收，内部的目标就是能够尽快尽量多地交付新的功能或者特性，并且在现网上线，从而体现自己的价值，对运维部门来说，最大的目标就是保障应用运行稳定，希望越少变化越好，可以看出，各部门不仅存在如何保证价值有效传递的问题，也要解决目标和价值存在差异，如何有效协同的问题，敏捷和DevOps就是为解决这种混乱之墙而产生的。敏捷主要解决客户、业务部门、开发部门之间的混乱之墙，DevOps进一步延伸，解决了开发部门和运维部门之间的混乱之墙。

技术进步为敏捷和DevOps提供有力支撑

Application Infrastructure	Application Architecture	Deployment & Packaging	Development Process
 Datacenter	 Monolithic	 Physical Servers	 waterfall
 Hosted	 N-Tier	 Virtual Servers	 Agile
 Cloud	 Microservices	 Containers	 DevOps

HUAWEI

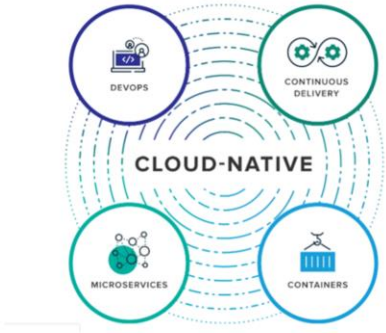
目录

1. 软件产业和交付模式发展趋势
2. **新兴软件技术及交付模式简介**
   - **云原生简介**
   - 微服务简介
   - 敏捷简介
   - DevOps简介
   - 敏捷和DevOps关系
3. 华为云DevCloud HE2E DevOps框架及其主要服务

HUAWEI

什么是云原生?

- 在**容器技术**、**可持续交付**、**编排系统**等开源社区的推动下，以及**微服务**等开发理念的带动下，应用上云已经是不可逆转的趋势。随着云化技术的不断进展，云原生的概念也应运而生。

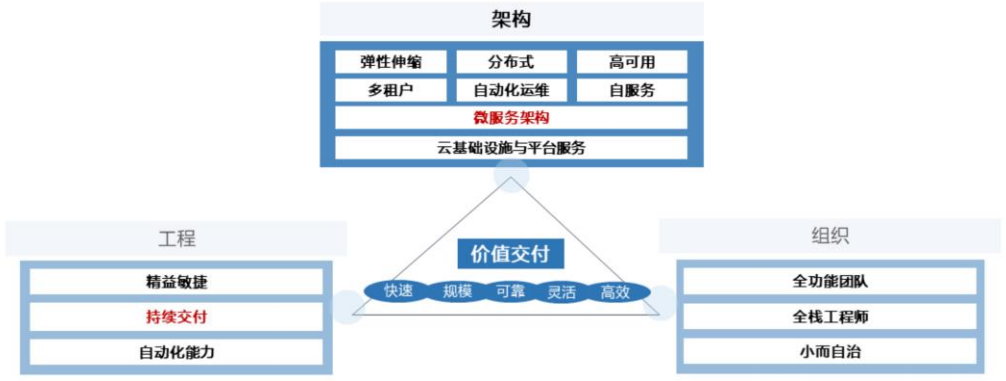


HUAWEI

- 云原生技术，大家可能只闻其声，不明其意。但是云原生背后典型的技术产品的名称可能大家经常听见：容器，docker，编排K8s，微服务 ，服务网格，敏捷，持续集成-交付-部署，DevOps等等，其本质是一些新兴技术和企业组织的集合。
- 云原生是方法和实践，云原生应用是云原生的结果，通过云原生方法实践的成果。
- 首先我们先来看一下云原生。过去数十年企业IT架构经历了单机、分布式和云计算三个阶段的技术演进。云计算时代的第一个十年，大多数互联网公司的应用从诞生之初就生长在云端。如今金融、制造、医疗以及政务等领域的企业和机构，也逐渐将业务迁移至云上。

云原生定义 - Cloud Native关键特征

- Cloud Native是指在云环境下构建、运行、管理软件的新的系统实践范式，充分利用云基础设施与平台服务，适应云环境，具备（微）服务化、弹性伸缩、分布式、高可用、多租户、自动化等关键特征的架构实践；建立全功能团队、发展全栈工程师并高度协作的组织，采用DevOps研发模式、自动化工具，实现微服务持续交付，是支撑Cloud Native架构的最佳组织和工程实践。



HUAWEI

- 企业的云化，不仅仅是基础设施和平台的升级，应用也需要摒弃传统的设计方法，从架构设计、开发方式到部署维护，整个软件生命周期都基于云的特点设计，从而构建原生为云而设计的应用，云平台的弹性以及分布式优势云原生就是这样一套全新的理念，背后涵盖了一系列全新的技术，例如容器、微服务、服务网格等，目前不仅互联网行业在使用云原生、制造 地产、生命科学、政府等各行各业也都在拥抱云原生技术，作为诞生于云计算时代的新技术理念，云原生拥有传统IT无法比拟的优势，从技术理念、核心架构、最佳实践等方面，帮助企业IT平滑、快速、渐进式地落地上云之路。

云原生应用

- 云原生应用程序是**专为云模型构建**的。这些应用程序**由小型专用功能团队**快速构建和部署到一个平台，可提供轻松的**横向扩展和硬件解耦** - 为组织提供跨云环境的更高**灵活性，弹性和可移植性**。**---Pivotal** https://pivotal.io/de/cloud-native

- 云原生应用是**独立的小规模松散耦合服务**的集合，旨在提供备受认可的业务价值，例如快速融合用户反馈以**实现持续改进**。简而言之，通过云原生应用开发，可以加速构建新应用，优化现有应用并将这些应用全部组合在一起。其目标是以企业需要的速度**满足应用用户的需求**。**---RedHat** https://www.redhat.com/en/topics/cloud-native-apps

HUAWEI

- 云原生应用，顾名思义，云原生应用是天然适合云特点的应用，云原生应用系统需要与操作系统等基础设施分离，不应该依赖linux或windows等底层平台或依赖某个云平台，也就是说应用从开始就设计为运行在云中，无论私有云或公有云，其次该应用必须满足扩展性需求，垂直扩展或水平扩展。云原生计算基金会CNCF给出了云原生应用的三大特征：
  - 第一容器化封装，以容器为基础，提高整体开发水平，形成代码和组件重用，简化云原生应用程序的维护，在容器中运行应用程序和进程，并作为应用程序部署的独立单元，实现高水平资源隔离。
  - 第二，动态管理，通过集中式的编排调度系统来动态地管理和调度。
  - 第三，面向微服务，明确服务间的依赖，相互解耦。



- 第一基准代码，一份代码库对应多份部署，所有部署的基准代码相同，但每份部署可以使用不同的版本。
- 第二，依赖显示声明依赖关系，通过依赖清单确切的声明所有依赖项，这一做法会统一应用到生产和开发环境。
- 第三，配置 在环境中存储配置、推荐，将应用的配置存储于环境变量中，环境变量可以非常方便地在不同的部署间做修改，却不动一行代码。与配置文件不同，不小心把它们迁入代码库的概率微乎其微，与一些传统的解决配置问题的机制，比如Java的属性配置文件相比，环境变量、语言和统计无关。
- 第四，后端服务把后端服务当作附加资源，每个不同的后端服务是一份资源，例如一个mysql数据库是一个资源，两个mysql数据库被当做两个不同的资源，云原生应用将这些数据库都视作附加资源，这些资源和他们附属的部署保持松耦合。
- 第五，构建发布运行云原生应用，需严格区分构建发布运行、这三个步骤。举例来说，直接修改处于运行状态的代码是非常不可取的做法，因为这些修改很难再同步回构建步骤。

目录

1. 软件产业和交付模式发展趋势
2. **新兴软件技术及交付模式简介**
   - 云原生简介
   - **微服务简介**
   - 敏捷简介
   - DevOps简介
   - 敏捷和DevOps关系
3. 华为云DevCloud HE2E DevOps框架及其主要服务

HUAWEI

微服务产生的背景

- 微服务架构的出现是因为**单体架构（Monolithic）**在其内部一个小改动，都会影响其他模块。特别是在云上发布，任何一个小的改动，都需要统一编译和发布。 对某个模块进扩展，也需要整体扩展。所以出现通过一系列的微服务来构建应用，各个微服务之间可以独立部署、独立扩展以及提供模块化的边界，还可以使用不同的语言进行开发。



HUAWEI

- 单体架构顾名思义就是一个归档包，如rar包或者jar包，包含了所有功能的应用程序，这是一种比较传统的架构风格。在软件开发的早期，因为单体架构部署简单，技术单一、用人成本低等特点，大家基本都采用这一架构模式。但是随着互联网时代的到来，随着业务需求复杂度的提高以及交付频率的不断加快，传统的单体架构，因为下面这些缺陷越来越难以满足开发人员的要求。
- 第一，复杂性高，单体架构项目包含的模块非常多，模块的边界模糊，依赖关系不清晰，代码质量参差不齐，整个项目非常复杂，每次修改代码都心惊胆战，甚至添加一个简单的功能或者修改一个bug，都会造成隐含的缺陷。
- 第二，技术债务逐渐上升，随着时间的推移，需求变更和人员更迭会逐渐形成应用程序的技术债务，并且越积越多，已使用的系统设计或代码难以修改，因为应用程序的其他模块可能会以意料之外的方式使用它。
- 第三，部署速度逐渐变慢，随着代码的增加，构建和部署的时间也会增加，而在单体应用中每次功能的变更或缺陷的修复，都会导致我们需要重新部署整个应用，全量部署的方式耗时长影响范围大、风险高，这使得单体应用项目上线部署的频率较低，出错概率高
- 第四，扩展能力受限，无法按需伸缩，单体应用只能作为一个整体进行扩展，无法结合业务模块的特点进行伸缩。
- 第五 阻碍技术创新、单体应用，往往使用统一的技术平台或方案解决所有问题，团队的每个成员都必须使用相同的开发语言和架构，想要引入新的框架或技术平台非常困难。

微服务架构的提出

- Martin Fowler总结亚马逊、NetFlix等互联网架构实践，提出了 Microservice（微服务）

In short, the microservice architectural style is an approach to developing a single application as a **suite of small services**, each **running in its own process** and communicating with lightweight mechanisms, often an HTTP resource API. These services are **built around business capabilities** and **independently deployable** by fully automated deployment machinery. There is a **bare minimum of centralized management** of these services, which may be written in different programming languages and use different data storage technologies.

-- James Lewis and Martin Fowler (2014)

微服务架构样式是一种将单个应用程序开发为**一组小型服务的方法**，每个小型服务都**在自己的进程中运行**并通过轻量级机制（通常是 HTTP 资源 API）进行通信。这些服务**围绕业务功能构建**，并且可以由全自动部署机制来**独立部署**。这些服务通过**最低限度的集中管理**，可以用不同的编程语言和不同的数据存储技术。

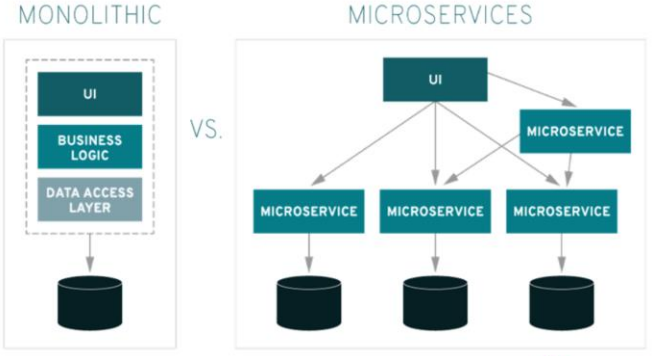
-- James Lewis and Martin Fowler (2014)

HUAWEI

- https://martinfowler.com/articles/microservices.html
- 定义翻译：微服务架构样式是一种将单个应用程序开发为一组小型服务的方法，每个小型服务都在自己的进程中运行并通过轻量级机制（通常是 HTTP 资源 API）进行通信。这些服务围绕业务功能构建，并且可以由全自动部署机制来独立部署。这些服务通过最低限度的集中管理，可以用不同的编程语言和不同的数据存储技术。-- James Lewis and Martin Fowler (2014)

微服务架构的优势

- 微服务是一种用于构建应用的架构方案。微服务架构有别于更为传统的单体式方案，可将应用拆分成多个核心功能。每个功能都被称为一项服务，可以单独构建和部署，这意味着各项服务在工作（和出现故障）时不会相互影响。
- 微服务架构优势
  - 缩短TTM时间
  - 高度可扩展
  - 弹性伸缩
  - 易于部署
  - 易于访问
  - 更加开放



HUAWEI

- 微服务架构的优势
  - 微服务可通过分布式部署，大幅提升您的团队和日常工作效率。还可以并行开发多个微服务。这意味着更多开发人员可以同时开发同一个应用，进而缩短开发所需的时间。
- 加速做好面市准备
  - 由于开发周期缩短，微服务架构有助于实现更加敏捷的部署和更新。
- 高度可扩展
  - 随着某些服务的不断扩展，您可以跨多个服务器和基础架构进行部署，充分满足自身需求。
- 出色的弹性
  - 只要确保正确构建，这些独立的服务就不会彼此影响。这意味着，一个服务出现故障不会导致整个应用下线，这一点与单体式应用模型不同。

目录

1. 软件产业和交付模式发展趋势
2. **新兴软件技术及交付模式简介**
   - 云原生简介
   - 微服务简介
   - **敏捷简介**
   - DevOps简介
   - 敏捷和DevOps关系
3. 华为云DevCloud HE2E DevOps框架及其主要服务

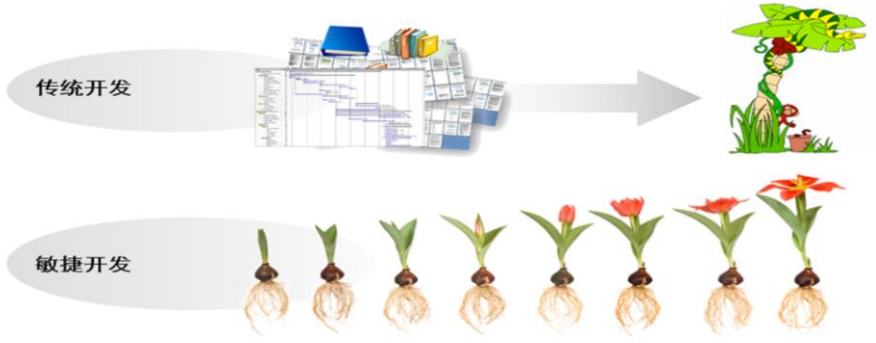
HUAWEI



- “敏捷软件开发宣言”。它是2001年，17位当时叫轻型方法论家，现在肯定就叫敏捷方法论家的大牛，在美国犹他州的雪鸟滑雪场讨论了3天之后达成的共识。包括四项价值观和十二条原则，其中值得我们注意的是第一条和第七条原则。
- 最重要的目标，是通过持续不断地及早交付有价值的软件使客户满意：这意味着，得有办法知道客户是否满意、能够判断软件功能是否有价值、能够尽可能提早交付，但还得是可持续地，比如加班、不做测试等办法，都会欠债，后续会拖慢速度。
- 可工作的软件是进度的首要度量标准：瀑布模式下，是按照某个阶段已完成、文档已完成来判断进度，但敏捷认为，只有某个软件功能已经可以工作甚至是已经上线运行才算是有进度。

可工作的软件是进度的首要度量标准

- 软件更像一个活着的植物，软件开发是自底向上逐步有序的生长过程，类似于植物自然生长
- 敏捷开发遵循软件客观规律，不断的进行迭代增量开发，最终交付符合客户价值的产品

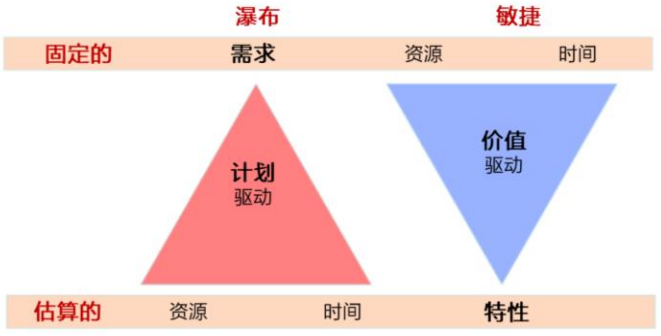


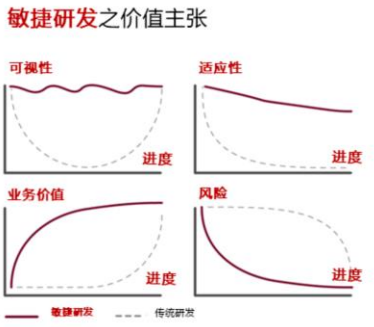
HUAWEI

- 上图为传统软件开发，在最终软件交付前，我们看到的是一堆中间件，文档、代码，不能评估其最终呈现，最后交付出来，很有可能是一个大bomb。客户本来希望一棵花草，但最终得到的却是奇形怪状的一个丑八怪。
- 与之相对的敏捷开发，通过迭代增量开发，不断向用户呈现持续增长的花草，不断地与客户交互纠偏，确保最终交付一棵美丽的绿色花草。

价值驱动 - 敏捷与传统瀑布型模式的最大区别

- 敏捷基于这种方式，可以实现研发过程的持续高可视性、高可适应性，更早且持续产出业务价值，更早发现和解决风险。

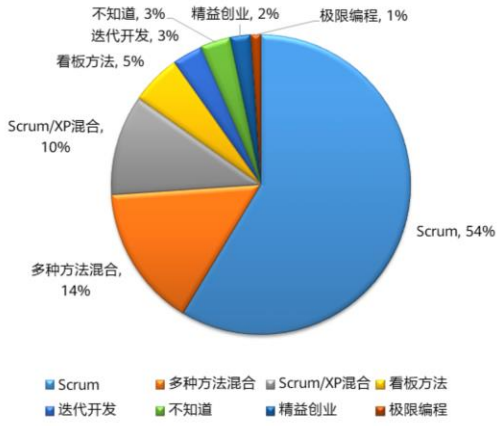




HUAWEI

- 先看左图，传统瀑布模式是计划驱动的，利用估算的资源在估算的时间内，按照计划去达成假定固定的需求。
- 敏捷则是追求价值驱动的，它更强调在我们固定资源投入和固定时间投入的情况下，通过调节特性的选择，追求最大化的价值产出。
- 敏捷主打的这种价值驱动模式，有什么好处呢？右图是对敏捷价值主张的一个简要呈现：
- 敏捷能够带来全程持续的高可视性，了解现场到底是什么情况；
- 敏捷通过短迭代的模式，在每个短迭代均可以进行需求调整，带来高适应性；
- 通过聚焦价值来排序特性，敏捷可以从一开始就产出高业务价值，更可以在边际效益递减时决策减少投入；
- 在敏捷方式下，能够更早地发现和解决风险，避免风险恶化；

敏捷常用的工程方法



根据最新的第13届VersionOne版年度敏捷行业状态报告（如左图），以Scrum为基础的方法论（包括**Scrum**、**Scrum/XP混合**等）体系仍然居于主流地位，使用率最高。其他还有：**看板方法**、**精益创业**、**极限编程**等。

除此之外，还有**DSDM**、**FDD**、**RAD**等一些符合敏捷精神的小众方法论，以及一些规模化敏捷的方法论，在此未一一列出。

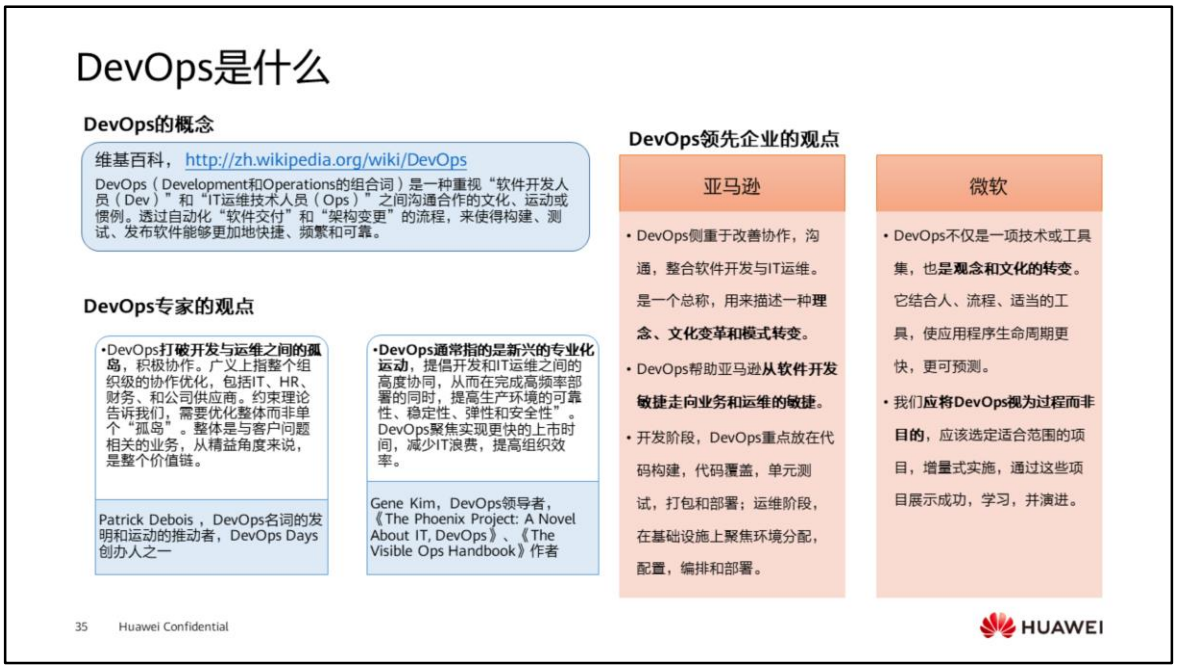
HUAWEI

- 敏捷宣言记载的是敏捷的精神，17位大牛并没有在非常具体的层面达成一致，也是因为前面我们说到的敏捷的真正难处，而适配不同环境的不同方法论很难在更具象的层面达成一致。所以，相比CMMI这种有唯一权威定义的时代或方法论，敏捷并没有一个唯一的官方权威的方法论，而是有很多符合敏捷精神的方法论可供选用。
- 根据业界权威的一份行业现状调查报告，由VersionOne公司每年度发起，今年刚刚发布了第13届报告。在众多方法论当中，Scrum以及Scrum/XP混合的方式最为普遍，其他方法论还有：Scrumban（糅合Scrum和看板方法的一种方法）、看板方法、迭代开发、精益创业、极限编程（XP），还有一些没有上榜的小众方法论，比如FDD、DSDM等。
- Scrum是迭代式增量软件开发过程。

目录

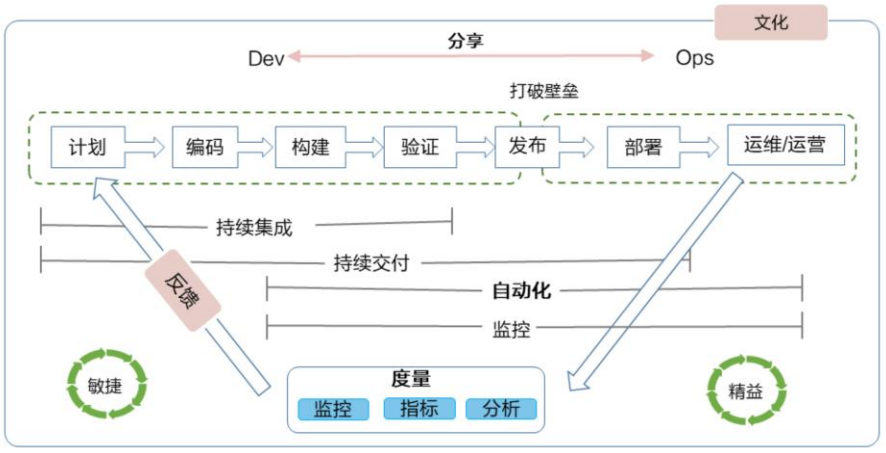
1. 软件产业和交付模式发展趋势
2. **新兴软件技术及交付模式简介**
   - 云原生简介
   - 微服务简介
   - 敏捷简介
   - DevOps简介
   - 敏捷和DevOps关系
3. 华为云DevCloud HE2E DevOps框架及其主要服务

HUAWEI



- DevOps这个词，我们要注意它承载了很多不同的含义。一方面它本身代表了一种运动，世界范围、行业范围内去推动这种文化和工作方式，另一方面也是代表了强调打破部门墙，开发和运维通力协作的文化，以及具体如何协作、更快交付业务价值的交付方式、协作方式。
- 关于工作方式层面的含义，我们可以参考维基百科的说法。
- DevOps，Development and Operations，是一组过程、方法与系统的统称，用于促进软件开发、运维和质量保障部门之间的沟通、协作与整合。它的出现是由于软件行业日益清晰地认识到：为了按时交付软件产品和服务，开发和运维工作必须紧密合作。

DevOps生命周期过程

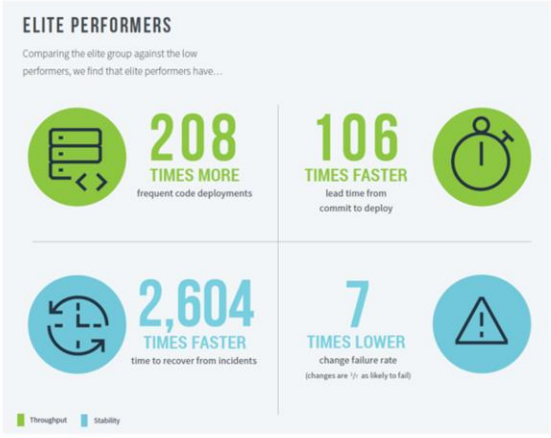


HUAWEI

- DevOps它不仅仅体现在流程上打通了软件开发的整个生命周期，它还包括以下五个要素：CALMS
  首先是文化：DevOps与传统职能型团队不同，它的前提是建立一体化的全功能团队，打破开发(Dev)与技术运营(Ops)隔阂，形成DevOps的协同合作的文化氛围
- 第二是自动化：自动化一切可以自动化的，通过自动化的工具或脚本实现软件工程从构建到运维过程的自动化流水线作业。
- 第三是精益：以精益的方式小步快跑，持续改善。
- 第四是度量：建立有效的监控与度量手段快速获得反馈，推动产品和团队的持续改进，测试驱动开发、增量式开发这些实践都是为了获得有效反馈并作用于下一次迭代周期中。
- 第五是分享：不同职能、不同产品之间经验分享能够促进DevOps的文化沉淀，促进产品迭代和更新。
- DevOps是敏捷理念从开发领域向运维领域的延伸。
- 计划阶段：运维人员为开发人员提供需求，并制定发布计划。
- 编码/构建/验证阶段：Scrum、极限编程和精益生产，持续集成、自动化测试等。
- 部署阶段：开发团队负责部署、监控部署过程，以及部署后的运行。

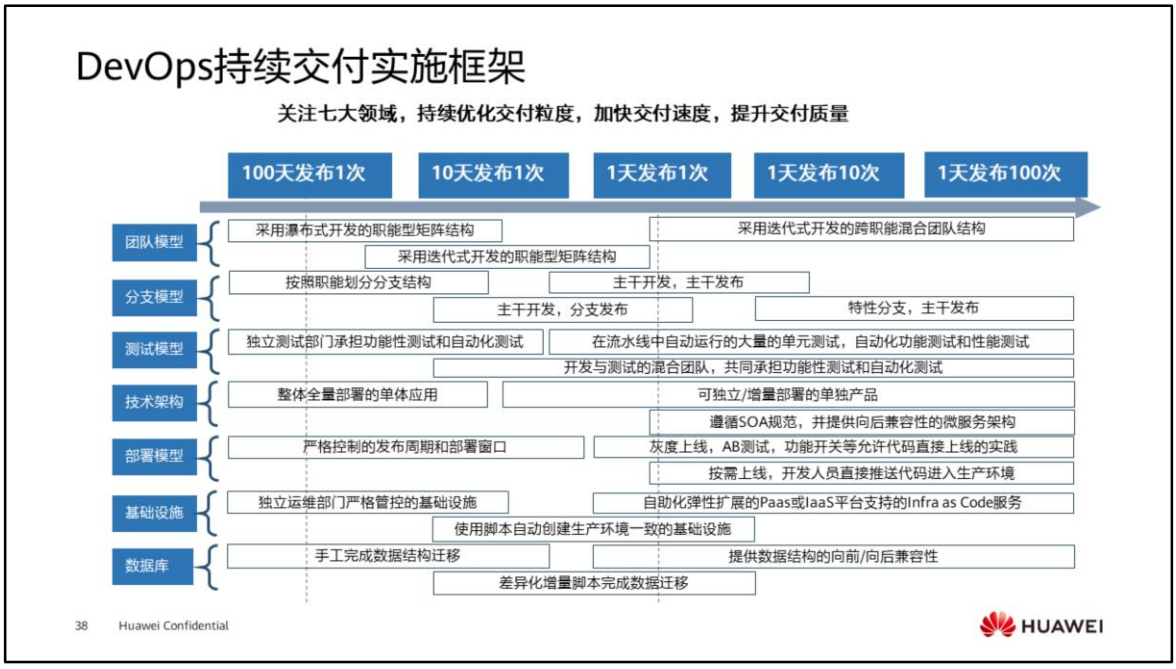
DevOps收益与价值

- 《2019 DevOps 状态报告》
- DevOps Eilite组织 VS DevOps Low组织
  - **208** 倍的代码部署频率
  - **106** 倍的代码到部署时间效率
  - **1/2604** 的故障恢复时间
  - **1/7** 的变更失败率



HUAWEI

- 实施了DevOps的企业与没有实施DevOps的企业，在交付速度和交付质量上都有了本质的提升。
- 我们看到，实施DevOps的企业，部署频率提高了几十倍。从代码提交到部署上线的时间，更是因为自动化有了翻天覆地的提高，甚至提高了几千倍。使得企业的交付速度和交付效率得到明显的提高。
- 下面两个指标反映了企业的交付质量，变更失败率指软件发布后不可用或者服务质量出现不可接受的性能下降；故障恢复时间指从现网发现问题，到故障全部恢复的时间。可以看到，这两个指标特别影响客户的感知和满意度，对企业影响特别大。我们看到，实施DevOps之后，企业的交付质量得到了质的提升。

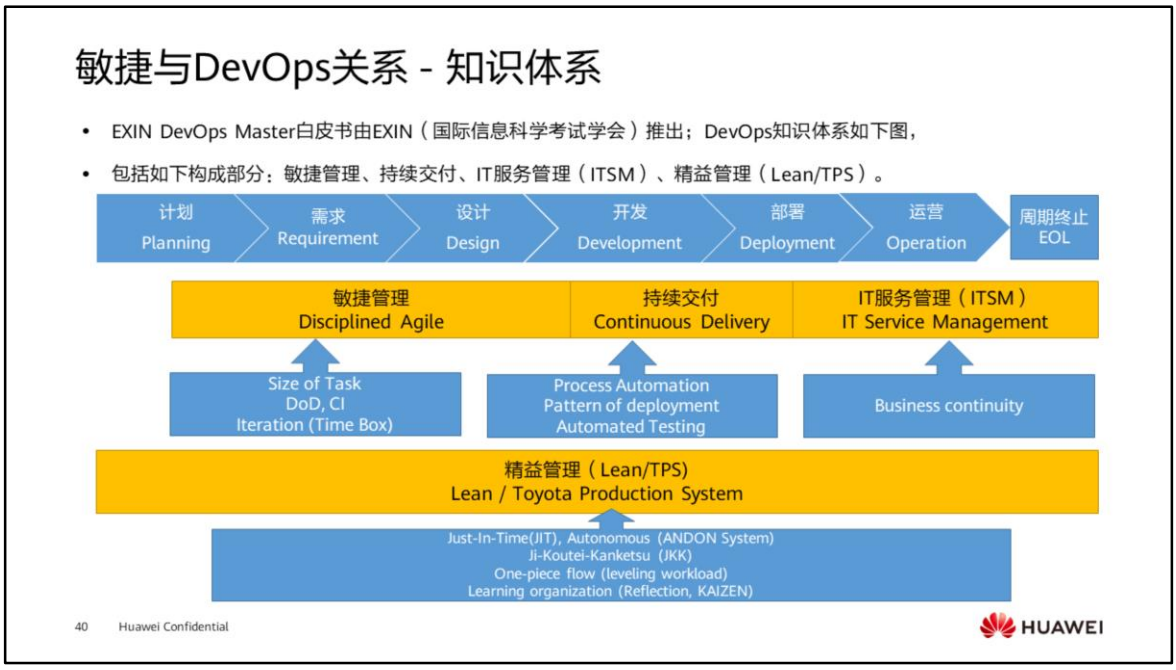


- 如图展示的是叫做“持续交付实施框架”的方法论，分为左边这七个领域，去改进。看中间主体部分，它告诉我们发布频率不同，工作方式或者具体实践是很不同的，如果要从100天发布一次的水平提升到一天发布100次的水平，在这七个领域都有很多改进动作需要落实才能做到。

目录

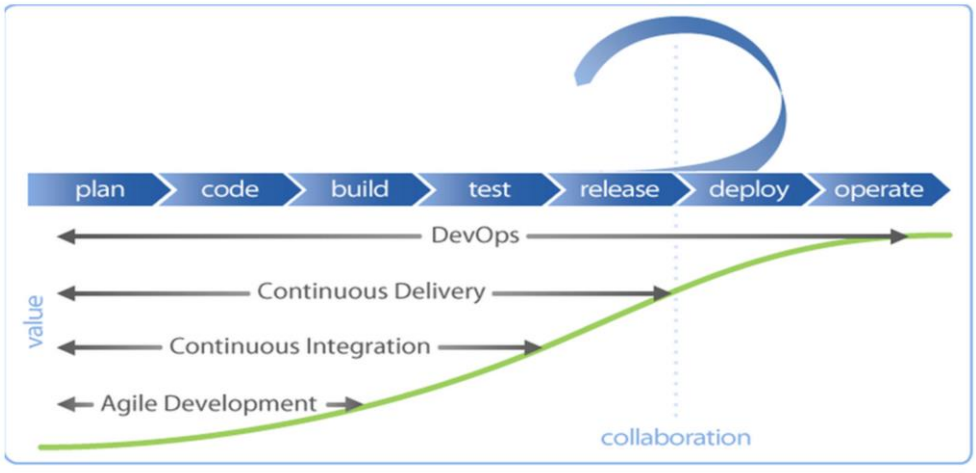
1. 软件产业和交付模式发展趋势
2. **新兴软件技术及交付模式简介**
   - 云原生简介
   - 微服务简介
   - 敏捷简介
   - DevOps简介
   - **敏捷和DevOps关系**
3. 华为云DevCloud HE2E DevOps框架及其主要服务

HUAWEI



- EXIN（国际信息科学考试学会）的DevOps Master白皮书，左图是白皮书中的DevOps知识体系，从中可以看到它将DevOps诠释为如下四个部分：
- 敏捷管理：包含计划、需求、设计、开发。
- 持续交付：包括开发、部署、运营。
- IT服务管理：包括运营、周期终止。
- 精益管理：从计划到周期终止的全程。
- 从中我们可以看出，DevOps不是对敏捷的否定，而是融合了敏捷和精益的思想和方法，并在其基础上的进一步发展。

敏捷与DevOps关系 - DevOps覆盖端到端交付周期



HUAWEI

- 从交付全流程的覆盖范围看，也经历了敏捷开发、持续集成、持续交付、DevOps四个重要阶段。
- 敏捷让开发团队拥抱变化、快速迭代，覆盖了计划、编码、软件生成阶段。
- 持续集成让开发团队提交代码后、立刻进行构建和单元测试，快速验证提交代码的正确性。
- 持续发布则在持续集成的基础上，将集成后的代码部署到更贴近真实运行环境验证，让团队可不断发布可用的软件版本。
- DevOps则覆盖全流程，加入了运维环节，用于促进开发、运维和质量保障部门之间的沟通、协作与整合，实现工程效率最大化。

- 持续集成：
  - 强调开发人员提交了新代码之后，立刻自动的进行构建、（单元）测试。根据测试结果，确定新代码和原有代码能否正确地集成在一起;
  - 重视自动化测试验证结果，对可能出现的一些问题进行预警，以保障最终合并的代码没有问题;
- 持续交付
  - 在持续集成的基础上，将集成后的代码部署到更贴近真实运行环境的（如类生产环境）中。交付给质量团队或者用户，以供评审。如果评审通过，代码就进入生产阶段。
  - 持续交付并不是指软件每一个改动都要尽快部署到产品环境中，它指的是任何代码修改都可以在任何时候实施部署。手动部署,有部署的能力，但不一定部署。
  - 持续部署则是部署活动是自动的，是持续交付的最高阶段。

思考题

1. 在某公司有一个新组建的敏捷团队，初次迭代时团队承诺交付8个story，但最终仅仅完成了5个，你作为Scrum Master应该怎么做?

   A. 将迭代时间延长

   B. 给团队增加更多的人力资源

   C. 将未完成的3个story放到待开发的列表中

   D. 加班完成剩下未完成的3个story

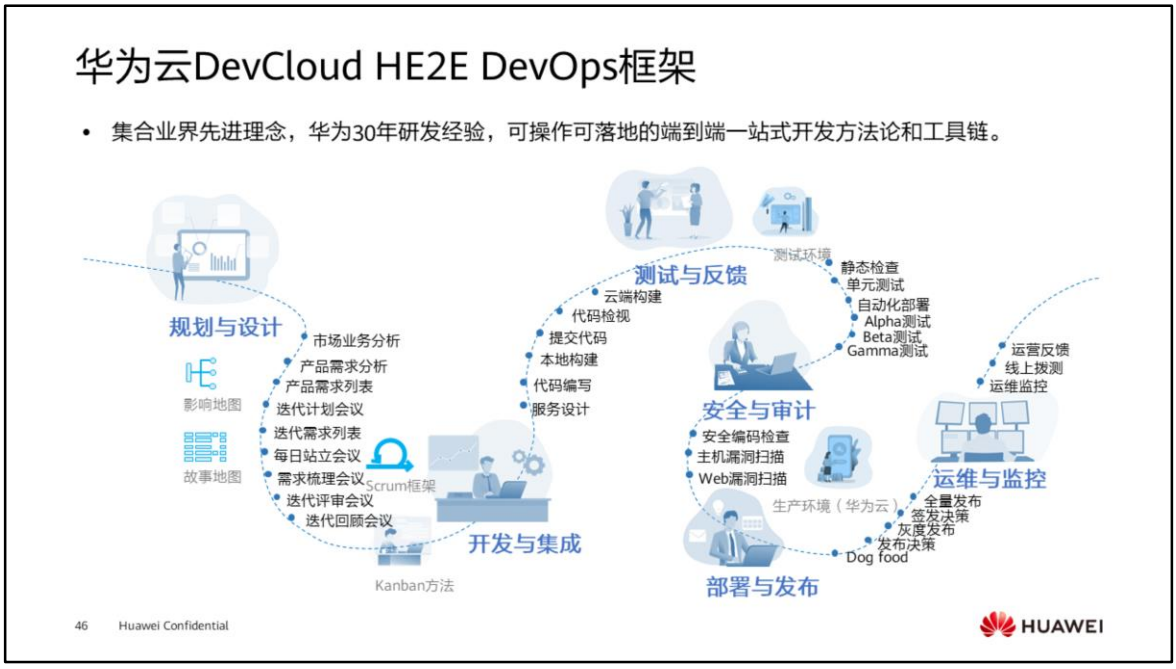
HUAWEI

- 参考答案：C

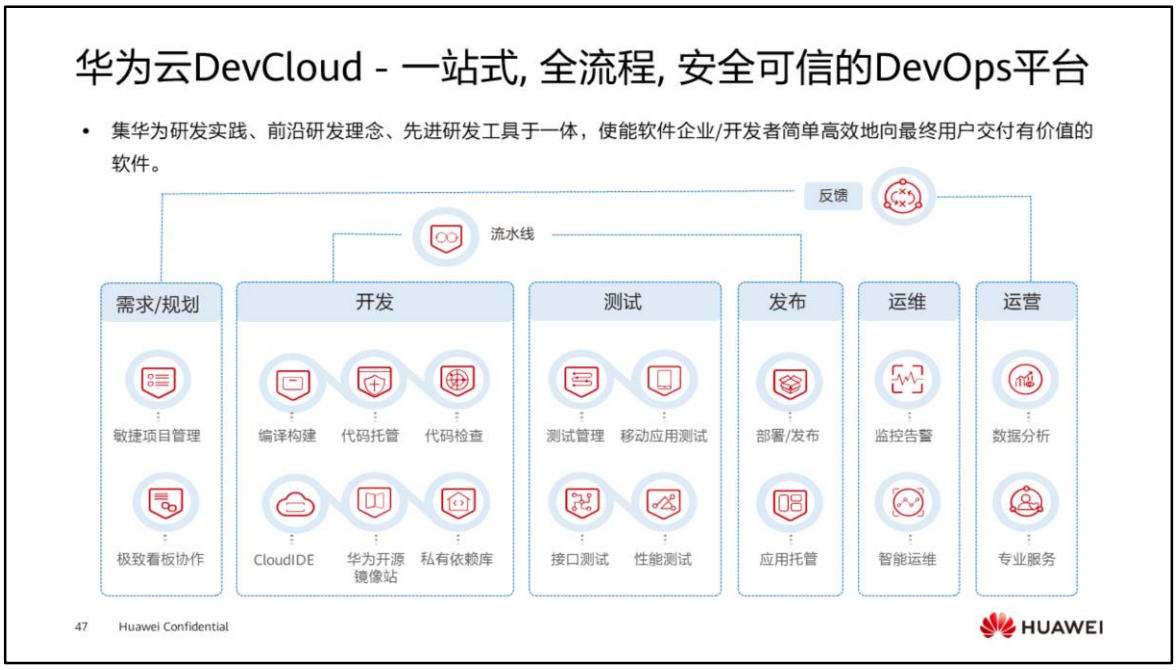
目录

1. 软件产业和交付模式发展趋势
2. 新兴软件技术及交付模式简介
3. **华为云DevCloud HE2E DevOps框架及其主要服务**
   - 华为云DevCloud HE2E DevOps框架
   - 华为云DevCloud HE2E DevOps主要服务

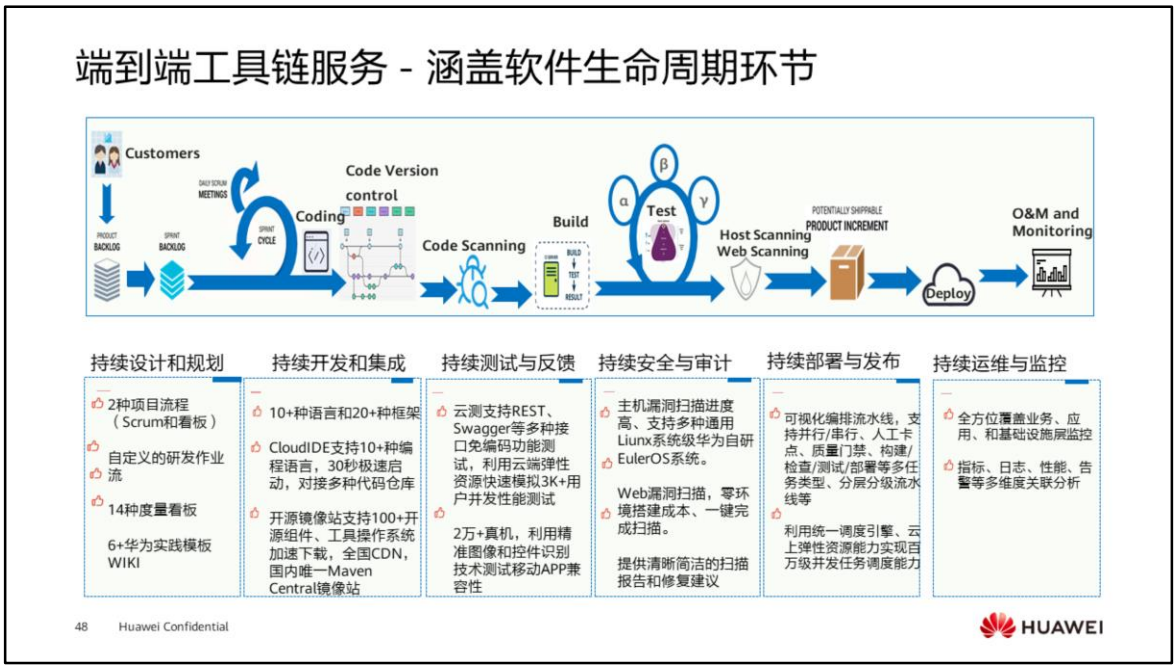
HUAWEI



- 企业进行DevOps变革，离不开方法与框架的指导。
- 在业界Lean、Agile、Kanban、CI/CD、DevOps等理念的基础上，结合华为30年研发经验与实践，华为云DevCloud形成了HE2E DevOps实施框架，为企业提供一套可操作可落地的敏捷开发方法论，并在华为云DevCloud工具链上进行承载。
- HE2E DevOps框架以产品研发生命周期划分为四个大的阶段：规划设计，迭代开发，持续测试，持续交付。
- 规划设计：基于对市场和行业的洞察，引用影响地图和用户故事地图进需求分析，基于Scrum和Kanban进行需求梳理以及迭代计划、评审、回顾等会议。
- 持续开发：服务设计与编码之后，进入到持续集成环节，代码进行本地或云端的构建，代码进行自查和互检，并基于分支模型进行代码提交。
- 持续测试：在各级环节进行分层分级的测试验证，从开发环境的静态检查、单元测试，通过自动化部署到alphha/beta/gamma环境，根据各级质量门禁进行质量检查。
- 持续交付：测试转生产之前，会在自用环境吃狗粮，再基于发布决策，进行灰度环境验证，通过的进入签发决策，进行全网的发布，随后持续进行运维监控，以及线上各类测试，通过运营及反馈进行完整的闭环。



- 华为云DevCloud是我们匠心打造的一站式、全流程、安全可信的DevOps云平台，集华为30年优秀研发实践、业界前沿研发理念、先进研发工具于一体，旨在使能软件企业或开发者简单高效地向最终用户交付有价值的软件。目前，DevCloud已提供13个服务，支撑需求与计划、设计与开发、测试、部署与发布等软件生命周期相关环节的研发活动。



- 华为云DevCloud是一个端到端工具链服务，涵盖软件生命周期。
- 在需求与计划环节：
- DevCloud支持Scrum和看板等2种项目流程；支持以思维导图，图形化的方式辅助团队进行需求分析、分解、讨论；
- 提供软件全生命周期管理，包括需求规划、迭代交付、缺陷跟踪、wiki/文档协作、统计回顾、实时分析等特性。
- 支持基于编排引擎的作业流定制，提供基于Widget技术的14种度量看板，此外，Wiki百科支持6种华为优秀研发实践模板。
- 在开发环节，华为云DevCloud提供代码开发、代码托管、代码检查、编译构建等服务。
- 基于CloudIDE的代码开发，提供全云化的开发体验，云端资源一键式获取和配置；
- 代码托管，提供安全高效，提供基于Git的安全代码托管服务，支持基于特性分支的Code Review开发模式；提供海量代码模板，让开发工作始于80%。
- 代码检查，提供可协作的一站式深度代码检查服务，及时深度发现代码缺陷，持续看护代码质量。
- 编译构建，基于云端大规模并发加速，提供高速、低成本、配置简单的混合语言编译构建能力。

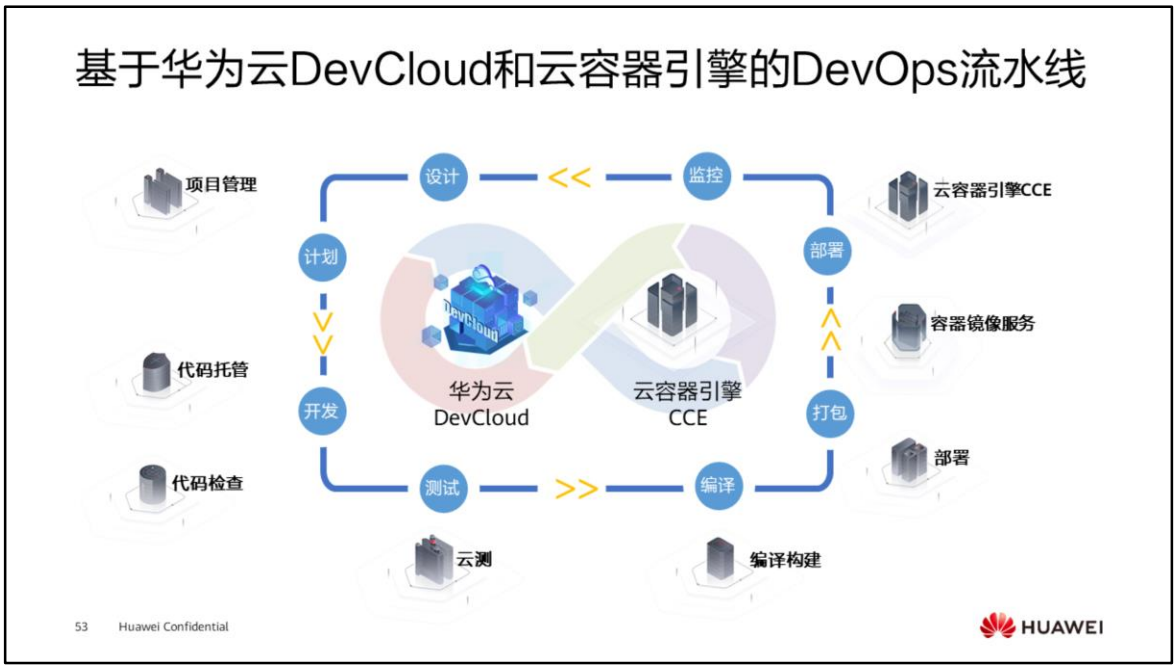


- 项目管理。
- 敏捷开发团队提供简单高效的团队协作服务，包含多项目管理、敏捷迭代、看板协作、需求管理、缺陷跟踪、文档管理、Wiki在线协作、仪表盘自定制报表等功能，确保团队可以按照用户故事进行高效的工作。
- 云测
- 面向软件开发者提供的一站式云端测试平台，覆盖测试管理、接口测试、性能测试，融入DevOps敏捷测试理念，帮助您高效管理测试活动，保障产品高质量交付

目录

1. 软件产业和交付模式发展趋势
2. 新兴软件技术及交付模式简介
3. **华为云DevCloud HE2E DevOps框架及其主要服务**
   - 华为云DevCloud HE2E DevOps框架
   - 华为云DevCloud HE2E DevOps主要服务

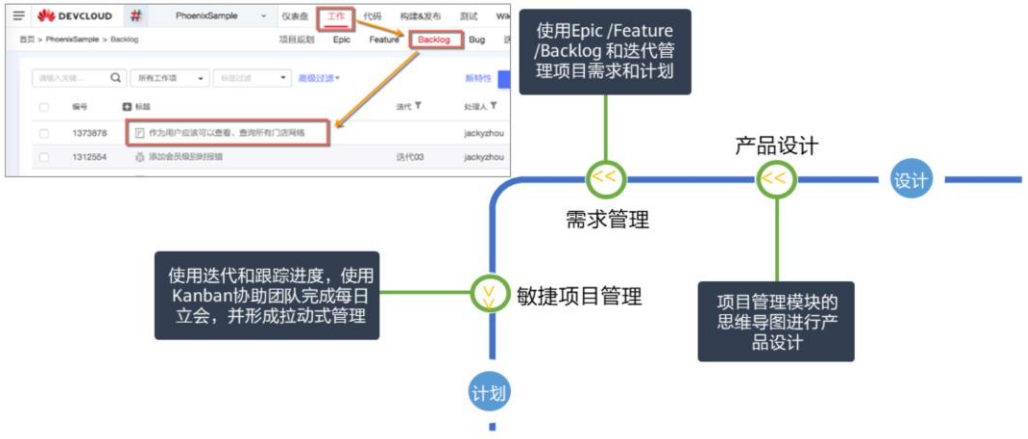
HUAWEI



- 项目管理：敏捷开发团队提供简单高效的团队协作服务，包含多项目管理、敏捷迭代、看板协作、需求管理、缺陷跟踪、文档管理、Wiki在线协作、仪表盘自定制报表等功能。
- 代码托管：为软件开发者提供基于Git的在线代码托管服务，包括代码克隆/下载/提交/推送/比较/合并/分支/Code Review等功能。
- 代码检查：基于云端实现代码质量管理的服务，软件开发者可在编码完成后执行多语言的代码静态检查和安全检查，获取全面的质量报告，并提供缺陷的改进建议和趋势分析，有效管控代码质量，降低解决成本。
- 云测：面向软件开发者提供的一站式云端测试平台，覆盖测试管理、接口测试、性能测试，融入DevOps敏捷测试理念，帮助您高效管理测试活动，保障产品高质量交付。
- 编译构建：基于云端大规模并发加速，为客户提供高速、低成本、配置简单的混合语言构建能力，帮助客户缩短构建时间，提升构建效率。
- 部署：提供可视化、一键式部署服务，支持并行部署和流水线无缝集成，支持脚本部署，容器部署等部署类型，支持java、node.js、python等多种技术栈，实现部署环境标准化和部署过程自动化。
- 容器镜像服务:支持容器镜像全生命周期管理的服务， 提供简单易用、安全可靠的镜像管理功能，帮助用户快速部署容器化服务。
- 云容器引擎CCE:高可靠高性能的企业级容器应用管理服务，支持Kubernetes社区原生应用和工具，简化云上自动化容器运行环境搭建。

设计与计划阶段

- 使用华为云DevCloud完成规划设计和敏捷项目管理



HUAWEI

敏捷项目管理

丢需求

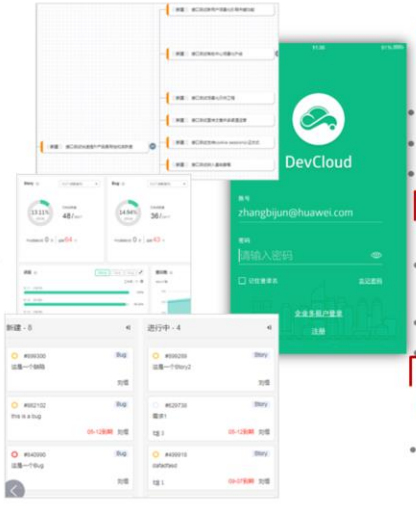
延期

抓瞎



重复犯错

阻塞



01 专业

- **敏捷管理**：迭代计划和时间线，及时准确掌控项目进展
- **思维导图规划**：产品全景规划，便捷需求拆分和路标计划
- **兼顾标准与轻量**：提供标准Scrum和轻量看板两种模板

02 连接数据、人、经验

- **管理者仪表盘**：自定制统计报表，掌握企业和项目宏观进展
- **项目文档管理**：超大容量文档托管，知识资产有效传承
- **Wiki协作**：团队知识在线分享，及时沉淀项目经验

03 随时随地

- **移动App**：随时查看和处理，不阻塞等待

透明

可控

可跟踪



继承

固化到工具

顺畅

HUAWEI

开发与测试阶段

- 使用华为云DevCloud协助团队完成编码开发，代码质量检查和质量验证



HUAWEI

代码托管

注重安全

- 支持IP白名单设置，确保接入安全
- 支持访问日志，方便审计
- 支持加密存储，防范代码泄露
- 支持Git-Crypt，扩展加密存储能力
- 安全防护墙，阻止用户上传敏感信息

高效协作

- 支持Git-Flow工作流
- 增强型Code Review：权限控制，Code Owners，评分机制，协作提醒，问题闭环
- 自动关联流水线，确保提交质量
- 独创Coding in Web，方便在线编辑代码

贡献可视化

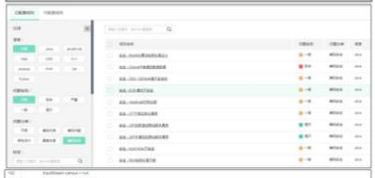
- 代码语言类型统计
- 代码增长趋势图
- 小组贡献量排名
- 提交活跃度分布
- 提交质量统计



HUAWEI

代码检查 - 持续看护代码质量, 防止代码腐化







全 面

- 覆盖了Top10主流编程语言的其中7种
- 从7个维度全面评价代码质量：编码安全、编码缺陷、编码风格、架构、依赖包安全、圈复杂度，代码重复

专 业

- 华为30年经验积累，提供专业级的预置规则集
- 提供专业的缺陷修复建议，指导代码质量改进

精 准

- 提供自定义规则集，可定制企业级质量标准，精准击中企业关心问题
- 缺陷精确定位到代码行

HUAWEI

云测 - 一站式云端测试管理和自动化测试平台

- 一站式云端测试平台
  - 覆盖测试管理、接口测试、性能测试，高效协同，一站式开展用例设计、测试执行、缺陷提交、生成报告，提高测试效率
- 云端高效自动化测试
  - 快速编排测试用例，一键性能测试，集成流水线，支持微服务测试、分层自动化测试等多种测试场景
- 全生命周期追溯和可视化
  - 需求-用例-缺陷双向追溯，测试有的放矢，多角色高效协同；多维度产品质量看板，全方位评估产品质量，保障产品高效验收



HUAWEI

- 简单:免编码，学习门槛低，快速上手；
- 0编码开发成本，乐高式编排脚本；
- 0环境搭建成本，聚焦测试和质量；
- 1步性能测试，1分钟获取性能报告；
- 2步接口测试，快速验证接口质量；
- 专业:基于业界先进理念和华为优秀实践；
- HTTP等4种API协议；高级测试场景编排；7类30+预置检查点；多环境测试参数管理；数千并发用户加压；
- 3种加压策略；8种性能指标；流水线持续自动化测试；
- 高效:融入DevOps敏捷理念和云端算力；
- 实时反馈测试结果；
- 多环境流水线自动化测试；
- 云端弹性资源池；
- 测试资产批量复用；

持续集成和持续部署

- 使用华为云DevCloud内置的CI/CD能力，持续交付价值



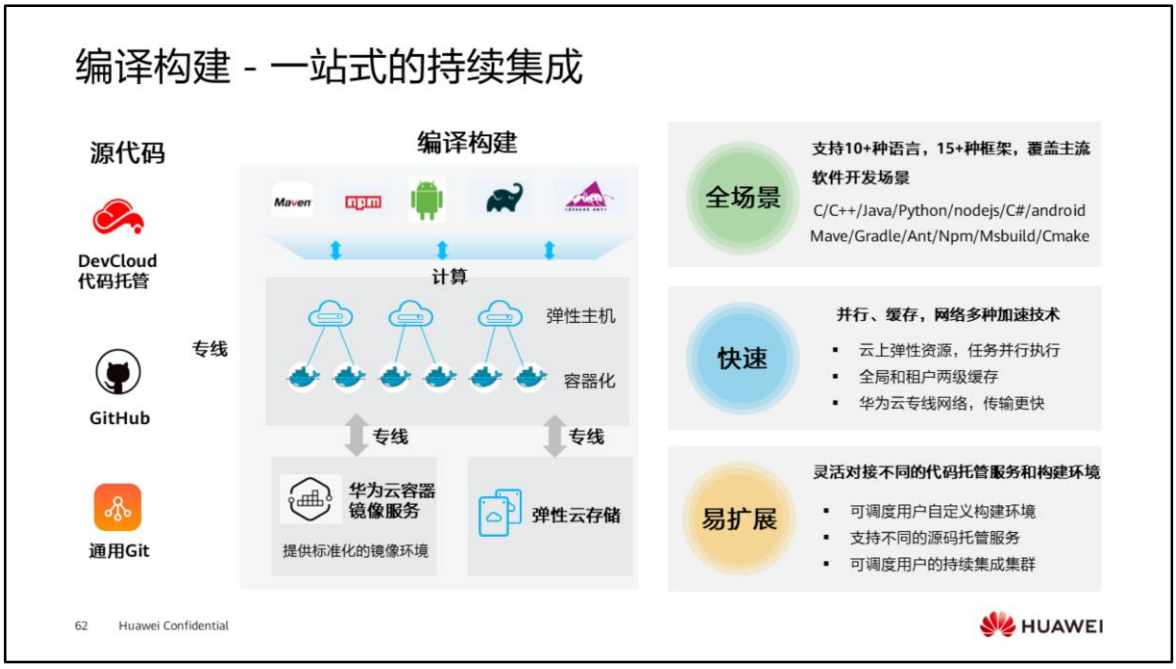
HUAWEI

打造快速, 可靠, 可重复的流水线

可视化编排	纳管任务	触发方式	管控模式	阻塞点精准识别	移动性
Stage阶段 Job任务 并行或串行 子流水线	代码检查 编译构建 接口测试 部署	代码提交 时间计划 人工触发	质量门禁 人工介入	可视化 任务健康度	移动App

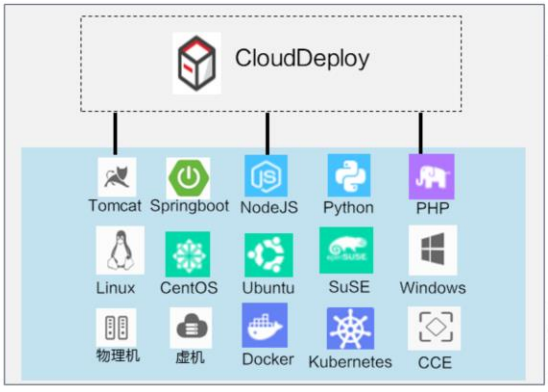


HUAWEI


编译构建 - 一站式的持续集成
源代码
DevCloud
代码托管
专线
GitHub
通用Git
编译构建
Maven
npm
计算
弹性主机
容器化
专线
专线
华为云容器
镜像服务
提供标准化的镜像环境
弹性云存储
全场景
支持10+种语言，15+种框架，覆盖主流软件开发场景
C/C++/Java/Python/nodejs/C#/android
Mave/Gradle/Ant/Npm/Msbuild/Cmake
快速
并行、缓存，网络多种加速技术
云上弹性资源，任务并行执行
全局和租户两级缓存
华为云专线网络，传输更快
易扩展
灵活对接不同的代码托管服务和构建环境
可调度用户自定义构建环境
支持不同的源码托管服务
可调度用户的持续集成集群
62
Huawei Confidential
HUAWEI

- 图片来源：华为云

部署 - 一键自动化部署到物理机, 虚机, 容器



部署应用到物理机、虚机、容器

- 支持将应用部署到物理机，虚拟机，容器。
- 支持主机有Agent或无Agent模式提供应用监控数据。
- 支持以代理机的方式进行部署。
- 支持部署到应用管理平台，纳管容器云集群或客户的私有K8S的集群。

支持多种技术栈应用的部署

- 支持Tomcat,Springboot,Nodejs等多种技术栈。
- 支持SpringCloud，ServiceComb，Dubbo等微服务应用部署和治理能力。
- 提供通用模板，且支持自定义模板，提供25+原子步骤组装成部署任务。

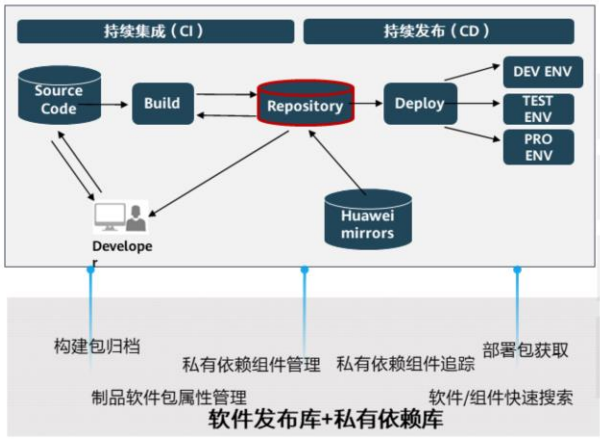
支持与流水线，运维管理无缝集成

- 在流水线能关联部署任务。
- 通过流水线关联构建，代码检查，测试等服务，实现端到端Devops流程。
- 应用部署后的可运维管理。

HUAWEI

- 图片来源：华为云

发布 - 软件包资产的可视化管理和追溯



构建包归档　私有依赖组件管理　私有依赖组件追踪　部署包获取

制品软件包属性管理　软件/组件快速搜索

软件发布库+私有依赖库

高效便捷的软件制品库

构建包自动归档	• 编译构建产物自动归档到发布库 • 编译构建属性自动关联软件包
多格式私有库	• 支持maven、docker私有组件管理 • 支持编译构建一键发布私有组件 • 私有组件变化追踪并及时知会
部署无缝集成	• 支持部署服务从发布库快速取包部署
多视图追溯	• 提供目录/构建两种视图支持用户进行软件包生命周期追溯

HUAWEI

华为开源镜像站

主流语言开发组件镜像

支持6种主流开发语言的开源组件镜像适合各类开发者的开发场景使用

maven、npm、python、docker、php 、nuget、rubygems

常用开发工具镜像

提供30种常用开发者工具镜像，配套各种开发场景的需要

nginx、apache、jdk、git、jenkins、mysql、ctan、zabbix、.....

热门开源操作系统镜像

提供22种热门开源操作系统的镜像供开发者按需选用并提供快捷下载方式

centos、ubuntu、epel、archlinux、debian、Opensuse、fedora、.....



更快

CDN加速，配置一键便捷生成

注册用户享受CDN专属加速

镜像通过香港代理出口及时同步

提供交流论坛，团队迅速响应问题

更全

一站式镜像站，适配各类开发场景

提供近60种常用开发镜像，免费使用

镜像覆盖开发组件、工具、操作系统

使用问题可在论坛中迅速获得帮助

更安全

官方社区合作，组件来源更可信

国内唯一的maven central镜像库

6种官方认证操作系统镜像

所有镜像来自官方社区并保持一致

HUAWEI

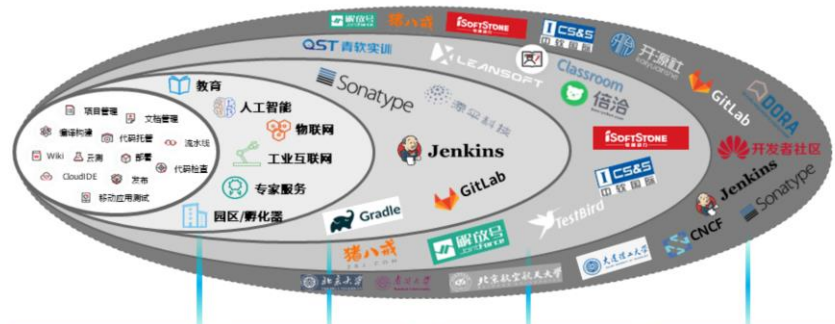
容器化云环境CCE

- 使用华为云提供的云容器引擎CCE确保应用的稳定高效运行和监控



HUAWEI

统一丰富的开发者生态



完整多维的解决方案 | 深入多层次的技术合作 | 全方位多领域的商业合作 | 开放丰富的开发者生态

生态开放

方案伙伴　高校　开发者社区开
专家服务　园区　源合作

平台开放

- DevOps、开源社区合作
- 源于开源，强于开源，回馈开源
- 开放API，分层解耦

HUAWEI

思考题

1. （单选）以下哪项不属于“敏捷软件开发宣言”提出的四个核心价值?

A. 流程和工具高于个体和交互
B. 工作的软件高于详尽的文档
C. 客户合作高于合同谈判
D. 响应变化高于遵循计划

HUAWEI

- A

本章总结

- 通过本章学习，我们对软件产业发展趋势、挑战及软件交付模式的演进有了一定了解，并掌握了敏捷、DevOps相关理念、价值和工程方法，对华为端到端DevOps框架、主要服务及其功能有了初步了解，为后续进一步学习打下坚实基础。

HUAWEI

更多信息

- 华为云官网：
  - https://www.huaweicloud.com/
- 华为云软件开发平台DevCloud：
  - https://www.huaweicloud.com/devcloud/
- 华为云帮助中心：
  - https://support.huaweicloud.com/
- 华为云学院：
  - https://edu.huaweicloud.com/

HUAWEI

学习推荐

- 《Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale》 -- Jennifer Davis&Katherine Daniels
- 《The DevOps Handbook: How to Create World-Class Agility, Reliability, and Securityin Technology》 -- Gene Kim & Patrick Debois & John Willis & Jez Humble
- 《Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation》 -- Jez Humble & David Farley
- 《Scrum 敏捷软件开发》-- Mike Cohn
- 《Scrum敏捷项目管理实战》-- Ken Schwaber

HUAWEI

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and organization for a fully connected, intelligent world.

Copyright©2020 Huawei Technologies Co., Ltd.
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.



持续规划与设计



HUAWEI

前言

- 本章主要讲述敏捷项目管理理念与方法实践，主要介绍规划与设计中的影响地图、用户故事地图，以及敏捷需求管理；并以华为敏捷项目管理企业实践，阐述了敏捷项目管理的理论、方法与模型。

HUAWEI

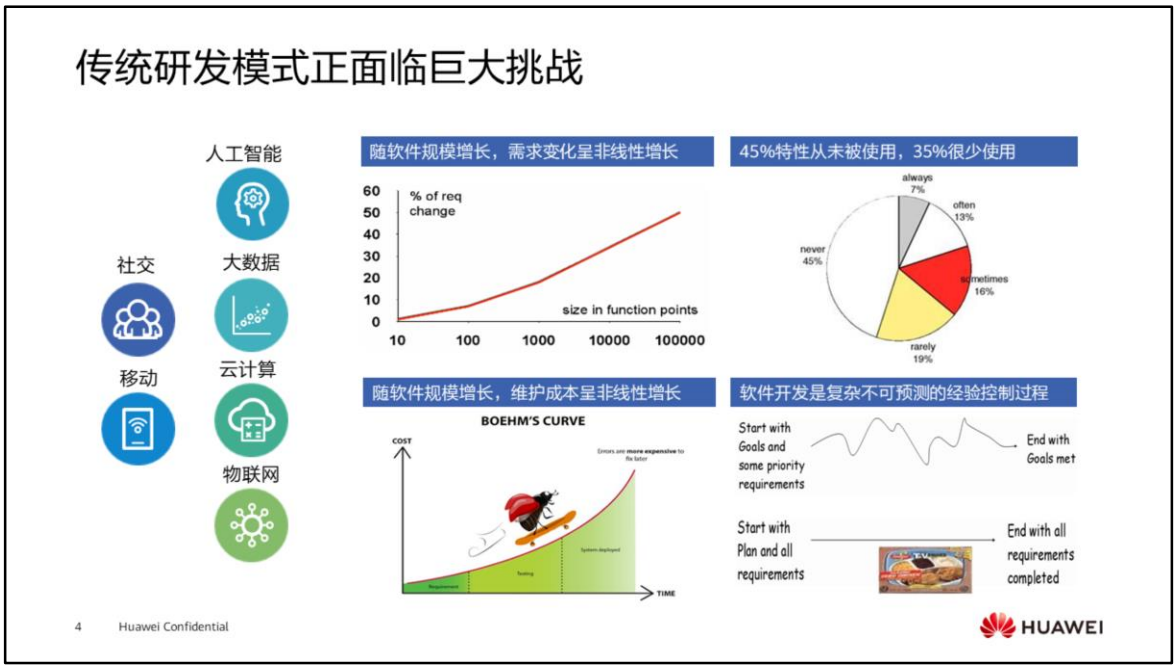
目标

- 了解敏捷项目管理理念
- 掌握影响地图、用户故事地图的实施方法
- 掌握敏捷需求管理

目录

1. **敏捷项目管理理念与方法实践**
2. 规划与设计
3. 敏捷项目管理的方法、模型
4. 敏捷需求管理
5. 团队与协作
6. 华为敏捷项目管理企业实践

HUAWEI



- 根据调查交付的软件中有45%的特性从未被使用，35%很少使用，一直使用加经常被使用，仅占20%。并且随着软件规模增长，需求变化和维护成本都呈非线性趋势增长，由此可见和烹饪食物盖高楼相比，软件开发是一个复杂的不可预测的过程。由于软件开发的复杂和不可预测，所以软件开发过程不再是传统的一次分析完所有的需求做出一个最终的产品交付，而是先做出产品的雏形，根据客户和市场的反馈逐步完善的过程。

敏捷更符合软件开发规律

- 软件更像一个活着的植物，软件开发是自底向上逐步有序的生长过程，类似于植物自然生长
- 敏捷开发遵循软件客观规律，不断的进行迭代增量开发，最终交付符合客户价值的产品

传统开发



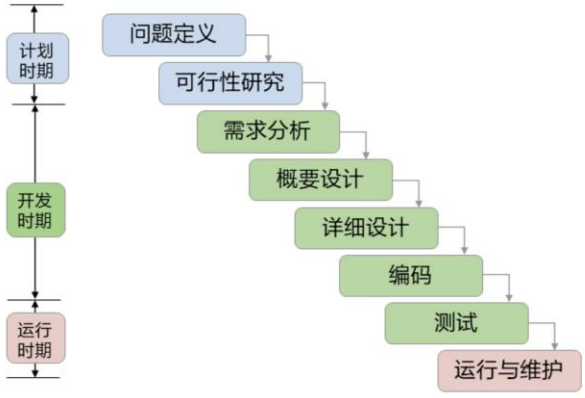
敏捷开发



HUAWEI

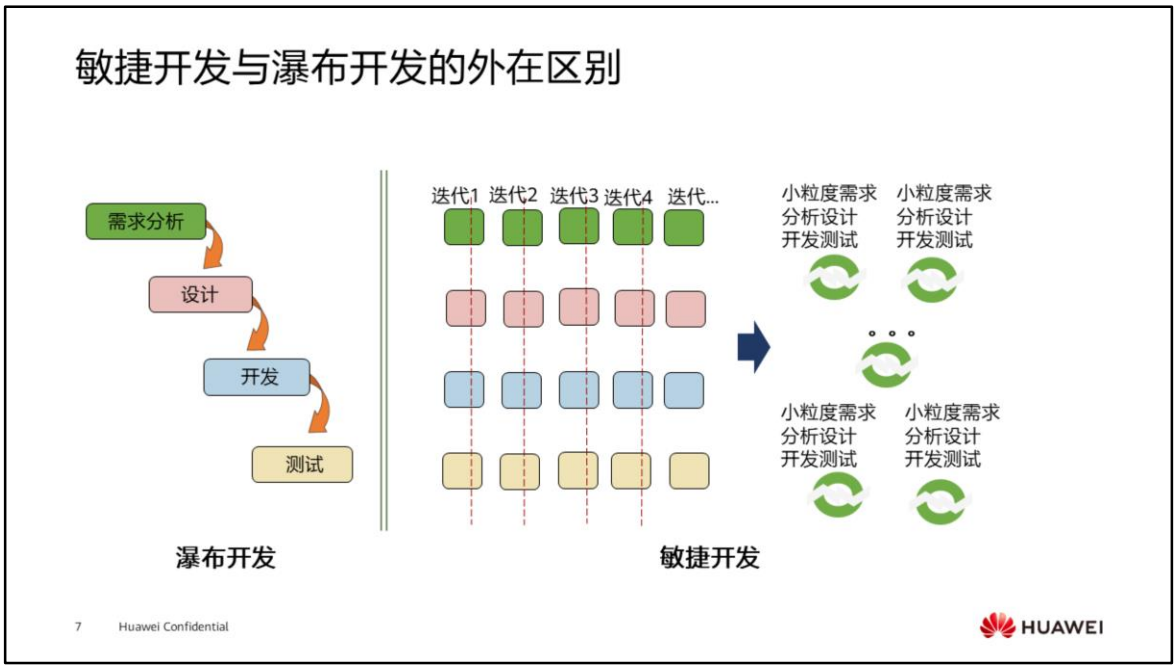
传统开发模型：瀑布模型

- 1970年温斯顿·罗伊斯(Winston Royce)提出了著名的“瀑布模型”，直到80年代早期，它一直是唯一被广泛采用的软件开发模型。罗伊斯提出的是一个两次瀑布的“迭代”模型(Do it Twice)。



HUAWEI

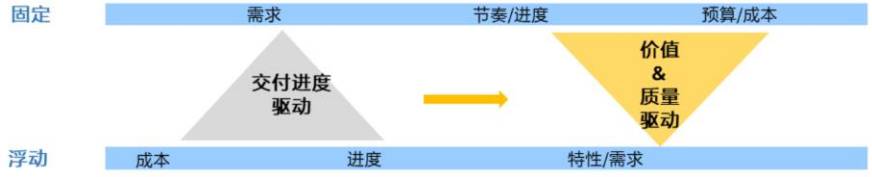
- 1970年温斯顿.罗伊斯在管理大型软件系统开发当中，提出了著名的瀑布模型，罗伊斯提出的是一个两次瀑布的迭代模型，两次瀑布的话是指将文档化的东西先试着预先走一遍，看看能否成为最终的产品。所以几十年来被大家所公认的瀑布之父，其实一直倡导的是迭代、递增和演进式的开发。他在文章当中所描述的瀑布模型，其实只是一种最简单的情况，并不是普遍适用的。



- 敏捷开发和瀑布开发的外在区别。瀑布开发是做完所有的需求分析之后进行设计、开发、测试，然后一次性交付。敏捷开发是分迭代进行，将需求拆分成小颗粒，每个迭代只交付一个。

敏捷开发与瀑布开发内在区别

- 敏捷开发的一个核心思维模式的转换便是：从瀑布式开发所代表的“Fix Scope， Flex time”（固定范围，弹性时间）转向“Fix time，Flex Scope”（固定时间，弹性范围）。在市场变化和技术变化的背景之下，既然市场需求和产品定义所代表的“范围”无法实现固化，因而无法确定应该投入多少资源来完成，那不妨固定好已有资源的，以资源为约束，实现“范围”的最大化实现。因为从“计划驱动”转向为“价值驱动”。



HUAWEI

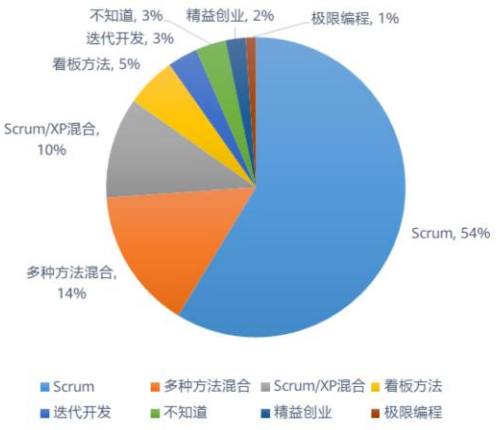
- 在瀑布开发当中，资源和时间是可变的，而需求是不可变，也就是固定范围要完全按照最初的计划来进行执行。也就是说在瀑布开发当中是以计划为驱动的，敏捷方法，尝试固定资源和时间，通过调整功能来达到在这些约束条件下产生更高优先级和更高质量的产品，也就是以价值为启动。



- placeholder：不是人名，可解释为“为项目作出贡献的无名氏” 。
- 敏捷宣言遵循的原则
  - 最重要的目标，是通过持续不断地及早交付有价值的软件使客户满意。
  - 欣然面对需求变化，即使在开发后期也一样。为了客户的竞争优势，敏捷过程掌控变化。
  - 经常地交付可工作的软件，相隔几星期或一两个月，倾向于采取较短的周期。
  - 业务人员和开发人员必须相互合作，项目中的每一天都不例外。
  - 激发个体的斗志，以他们为核心搭建项目。提供所需的环境和支援，辅以信任，从而达成目标。
  - 不论团队内外，传递信息效果最好效率也最高的方式是面对面的交谈。
  - 可工作的软件是进度的首要度量标准。
  - 敏捷过程倡导可持续开发。责任人、开发人员和用户要能够共同维持其步调稳定延续。
  - 坚持不懈地追求技术卓越和良好设计，敏捷能力由此增强。
  - 以简洁为本，它是极力减少不必要工作量的艺术。
  - 最好的架构、需求和设计出自自组织团队。
  - 团队定期地反思如何能提高成效，并依此调整自身的举止表现。

敏捷开发主张的方法论

- 根据最新的第13届VersionOne版年度敏捷行业状态报告（如左图），以Scrum为基础的方法论（包括**Scrum、Scrum/XP混合**等）体系仍然居于主流地位，使用率最高。其他还有：**看板方法、精益创业、极限编程**等。
- 除此之外，还有**DSDM、FDD、RAD**等一些符合敏捷精神的小众方法论，以及一些规模化敏捷的方法论，在此未一一列出。



HUAWEI

- 根据最新的第13届VersionOne版年度敏捷行业状态报告，其中Scrum方法的使用占了54%，Scrum/XP的混合占10%，多种方法混合占14%，看板方法占5%，其他方法占1%~3%不等。此外还有DSDM FDD RAD等一些符合敏捷精神的小众方法论，以及一些规模化的敏捷方法论，在此并未一一列出。

敏捷开发管理方法 - Scrum

- 1986年，竹内弘高和野中郁次郎发表文章《新新产品开发游戏》，阐述了一种新的整体性的方法，该方法能够提高商业新产品开发的速度和灵活性。
- 1995年，在奥斯汀举办的OOPSLA95上，Sutherland和Schwaber联合发表论文首次提出Scrum概念。Schwaber和Sutherland在接下的几年里合作，将上述的文章、他们的经验、以及业界的最佳实践融合起来，形成我们现在所知的每日Scrum。



HUAWEI

- 1986年竹内弘高、和野中郁次郎、在哈佛商业评论当中发表了文章，新新产品开发游戏，这篇文章描述了像本田、佳能、富士施乐这样的公司是如何通过可伸缩、基于团队的蜂拥式开发世界一流的产品，在文章当中强调了授权和自组织团队的重要性。95年Schwaber和Sutherland联合发表了论文，首次提出了Scrum概念。接下来的几年，二人合作将上述的文章、他们的经验以及业界的最佳实践融合起来，形成了现在我们所知的Scrum体系和框架。结合下面这张图，我们来大概了解一下，在Scrum当中首先将需求形成一个产品待办列表，然后在迭代的计划会议当中，从产品待办列表选取适当的需求条目，进入到sprint待办列表，接下来进入到2~4周的迭代开发过程，每天会进行每日站会，在站会上团队成员会共享进度资源和风险，迭代完成之后，会提交一个潜在的可交付的产品增量给客户进行评审，这就是一个Scrum的大体框架。
- ScrumAlliance：国际Scrum联盟认证https://www.scrumalliance.org/

敏捷开发管理方法 - 看板

- 看板源于精益生产实践（丰田生产），敏捷开发将其背后的可视化管理理念借鉴过来。看板使得项目管理最大的可视化，但是看板更可以将研发的过程进行管理，记录下用户故事研发过程中的细节和历程。

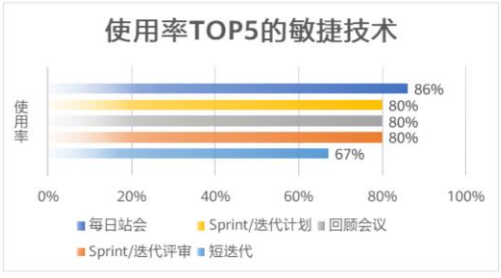


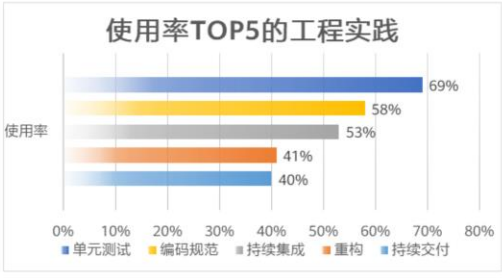
HUAWEI

- 看板是源于精益生产实践，也就是丰田生产。敏捷开发，将它背后的可视化的管理理念给借鉴过来，看板使得项目管理最大的可视化，但是看板更可以将研发的过程进行管理，记录下用户故事研发过程当中的细节和历程。看板的核心机制其实是限制在制品的数量的拉动疏通，通过它来暴露系统运作流程当中的问题，并激发协作以改进系统。

不同场景下敏捷开发实践

- 不同方法论包含一组敏捷实践的特定组合，企业也可以根据自身的需要混合多种方法论及实践，形成自己的独特组合。
  - 敏捷实践可以细分为管理实践、工程与技术实践等不同细类；具体的敏捷实践清单，可以参考如下一些权威来源：
  - 敏捷联盟的敏捷实践地铁图：https://www.agilealliance.org/agile101/subway-map-to-agile-practices
- VersionOne的年度敏捷行业状态报告
- 第13届敏捷行业状态报告中，敏捷技术和工程实践的使用率数据如下：





HUAWEI

敏捷开发实践举例

- 用户故事地图
  - 用户故事地图（User Story Mapping）是一门在需求拆分过程中保持全景图的技术。它可以使我们专注于用户和用户体验，产生更好的沟通效果，最终做出更好的产品。
- 影响地图
  - 影响地图（Impact Mapping）由高级技术和业务人员协作创建，它可视化了产品范围（需求）及其背后的假设。
- ATDD流程
  - 接收/验收测试驱动开发（Acceptance Test Driven Development）。
- 实例化需求
  - 实例化需求说明（Specification by Example）是一组过程模式，它帮助团队构建正确的软件产品。

HUAWEI

- 推荐学习材料：
- 用户故事地图：https://book.douban.com/subject/26760348/
- 影响地图：https://www.ituring.com.cn/book/1395
- ATDD流程：http://testobsessed.com/wp-content/uploads/2011/04/atddexample.pdf
- 实例化需求：https://book.douban.com/subject/11611022/
- 用户故事地图
  - 从左到右：讲述用户经历的重要步骤
  - 自顶向下：拆分上述每个步骤的细节，常见方式是按照用户活动、用户任务、用户故事三个层级进行细拆
- 影响地图
  - 它是在通过回答以下四个问题的讨论中，完善的一种思维导图。可以简单理解为是一种四级结构的需求实践
  - - Why：为什么做?
  - - Who：谁有助于做到?
  - - How：他们怎么帮助?
  - - What：用什么功能来做?

敏捷开发实践举例

- 用户故事地图
  - 用户故事地图（User Story Mapping）是一门在需求拆分过程中保持全景图的技术。它可以使我们专注于用户和用户体验，产生更好的沟通效果，最终做出更好的产品
- 影响地图
  - 影响地图（Impact Mapping）由高级技术和业务人员协作创建，它可视化了产品范围（需求）及其背后的假设
- ATDD流程
  - 接收/验收测试驱动开发（Acceptance Test Driven Development）
- 实例化需求
  - 实例化需求说明（Specification by Example）是一组过程模式，它帮助团队构建正确的软件产品

HUAWEI

- ATDD
  - 分三个主要过程：
  - 1，产品负责人与团队一起探讨需求内容及如何验证达成共识
  - 2，团队（开发&测试等）进行开发，期间频繁集成
  - 3，每完成一个需求特性，即可找产品负责人确认并交付

- 实例化需求
  - 使用实例化需求说明，团队编写的文档数量恰到好处，在短迭代或基于流的开发中可以有效地协助变更。
  - 核心理念是通过需求说明、实例、测试用例的结构，以及高度自动化，实现“活文档”

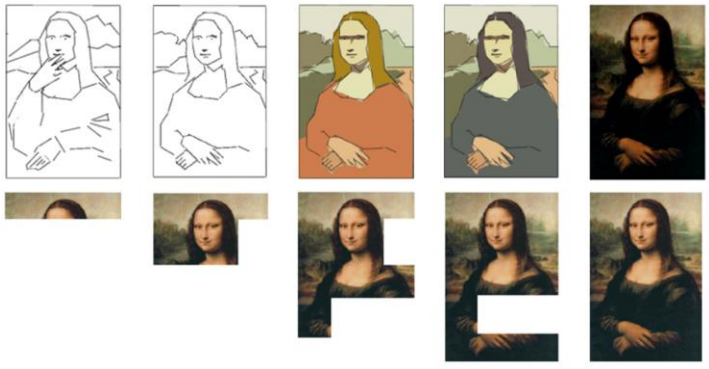
目录

1. 敏捷项目管理理念与方法实践
2. **规划与设计**
   - **增量式交付**
   - 影响地图
   - 用户故事地图
3. 敏捷项目管理的方法、模型
4. 敏捷需求管理
5. 团队与协作
6. 华为敏捷项目管理企业实践

HUAWEI

如果你是达芬奇?

- 在创作蒙娜丽莎的微笑这幅时，是先绘制整体轮廓，再逐渐填充细节和颜色；还是先绘制完成油画的局部，再逐渐绘制剩余的局部?



HUAWEI

- 第1组图当中的每一幅图都在前一幅图的基础上进行了调整，让整个图逐渐趋于最终的结果。第2组图的每一幅画，都是最后作品的局部实现。这两个创作过程哪一个是迭代增量的方式。首先是起步于MVP，细节功能不完善，但是可以验证用户的整体使用场景是可以独立交付的。想一想前面的两组绘画过程，第1组就满足了这样的一个特点。其次是每增加一点功能都能够让用户体会到价值的提升。产品的整个实现过程是一个持续改进的过程，是与用户一同探寻最终的产品，用户通过交付的增量能够更加理解后面的需求。

什么是增量式交付

- 起步于粗糙的产品原型/框架
  - MVP – minimum viable product (MVP) 最小可用产品
- 细节功能不完善，但可以验证用户的整体使用场景
  - 行走的骨骼 – 没有人形，但是具备了人的所有基础框架
- 每增加一点功能，用户都能体会到价值的提升
  - 持续改进 – 一次把糖果都给小朋友，每次给一颗的区别
- 与用户一同找寻最终的产品
  - 用户能从早期的增量，更加理解后面的需求

敏捷开发的基石是“增量式交付”。增量交付是指为及时反馈和接纳，频繁向客户交付连续改善的工作产品。

HUAWEI

- 增量交付是指为及时反馈和接纳频繁向客户交付连续改善的工作产品。

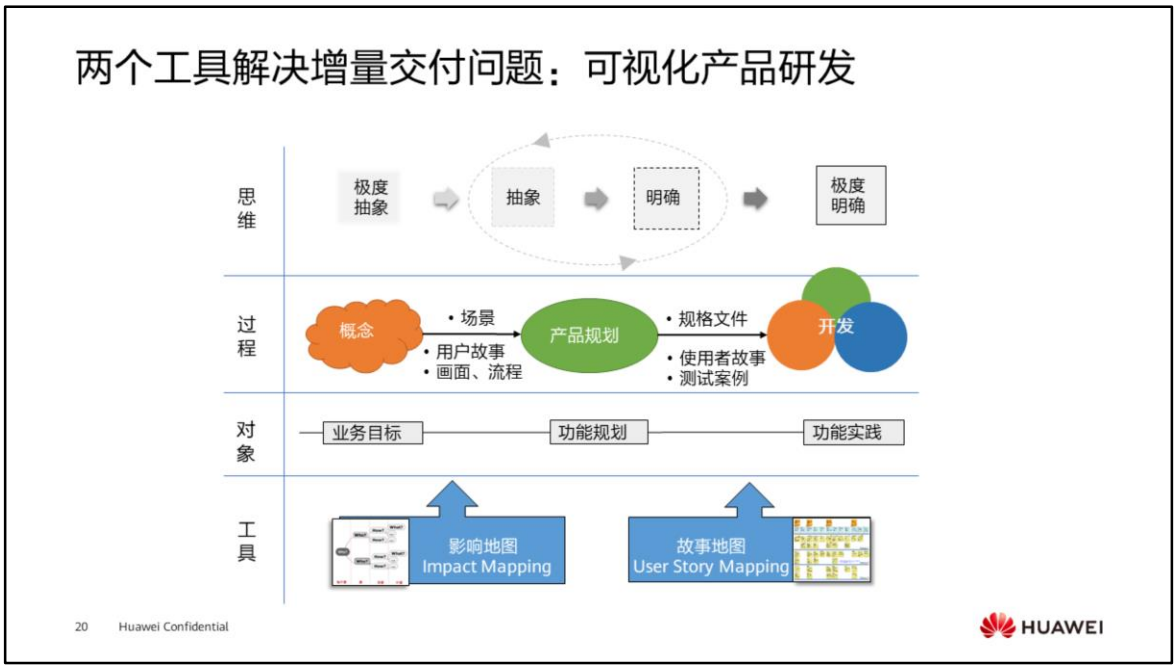
增量式交付 vs 瀑布式交付

- 如果交付的目标是确定的，增量式交付与瀑布式交付的周期和资源投入是一致的。
- 软件的目标总是变化的，因此经常出现增量式需要更长的周期和更多的投入，最终交付了不同的目标。
- 因此，我们需要措施，确保我们按照用户可用的场景去交付。

增量式交付更好?



HUAWEI



- 增量是交付式交付和瀑布式交付，在瀑布式交付当中一个产品经过了所有流程之后和客户见面，而增量式交付是每个阶段给出一个可交付的成果，根据用户的反馈不断调整，最终交付客户真正想要的真实可用的产品，在两种交付的方式当中，增量式交付更好吗？如果交付的目标是相同的，增量是交付和瀑布交付的周期和资源投入都是一致的。瀑布式交付是用于目标不变的场景，而增量式是用于目标不确定复杂变化的场景，软件的目标总是会变化，我们需要措施，确保我们按照用户可用的场景去交付，这就需要解决研发过程可视化的问题，将产品功能和用户目标联系在一起，我们引入两个工具，影响地图和故事地图，影响地图是实现从业务目标到功能规划，将极度抽象的概念，通过场景和用户故事，抽现为产品规划的过程。故事地图是实现从功能规划到功能实现，让抽象的功能规划变成明确和极度明确的开发任务。这样通过两个工具就实现了业务目标和产品功能的映射，确保我们可以按照用户可用的场景去进行交付。

目录

1. 敏捷项目管理理念与方法实践
2. **规划与设计**
   - 增量式交付
   - **影响地图**
   - 用户故事地图
3. 敏捷项目管理的方法、模型
4. 敏捷需求管理
5. 团队与协作
6. 华为敏捷项目管理企业实践

HUAWEI

影响地图 - 一个价值导向的实践方法

- 使用可视化的方式，建立商业目标与产品功能的联系，并将联系背后的假设可视化出来。



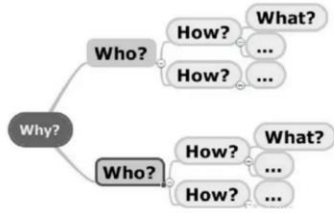


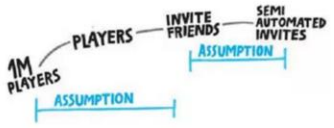
HUAWEI

- 二维码链接：https://www.impactmapping.org。
- 影响地图是一个以价值为导向的实践方法，团队都面临着这样的一个问题，业务目标和功能之间的联系很模糊或者不一致，业务人员和开发人员存在沟通的隔阂，导致大多数研发出来的产品都是失败的。
- 影响地图可以通过可视化的方式，建立商业目标和产品功能的关联，并将联系背后的假设，可视化出来，实现将概念进行可视化，它是通过四个问题达成统一的认知。

如何实践影响地图 - 通过四个问题达成统一认知

- 为什么 (Why):
  - 地图的中心描述了我们为什么做这些? 也就是我们 试图达成的目标是什么
- 谁 (Who):
  - 地图的第一层描述了谁能产生所需要的效果? 谁会阻碍它? 谁是我们产品的消费者或用户? 谁会被他影响? 也就是那些角色会影响结果
- 怎样 (How):
  - 地图的第二层描述了角色的行为是怎样被改变? 他们怎样说明我们达成目标? 他们怎样帮助或妨碍我们取得成功? 也就是我们 想要产生什么影响
- 什么 (What):
  - 地图的第三层描述了我们 可以做什么, 来支持第二层中所列出的影响的实现

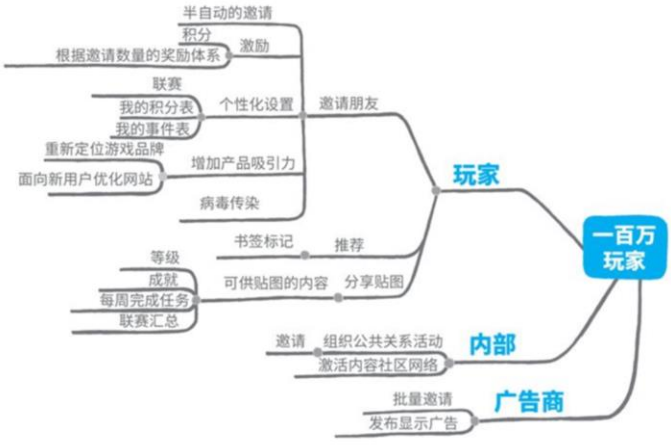




HUAWEI

如何实践影响地图 - 对复杂任务关系进行分析拆解

- 可以做什么?
- 目标是什么?



HUAWEI

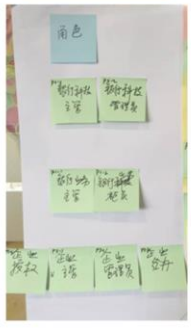
- 首先是why找准目标，可以运用5why的工具，找到的目标要符合smart原则。比如100万玩家，这就是一个明确的目标。第二步是who找准角色，所有和目标有关联的角色都要进行考虑。想实现100万玩家的目标，玩家就是一个和目标有关联的角色。第三步是，how，为了达成目标，我们根据用户角色需要用怎么样的方式去完成，玩家可以通过邀请朋友的方式来帮助实现100万玩家的目标，基于上面的方式，我们如何去做，玩家邀请朋友，可以通过半自动化的形式来进行邀请，这样就很清晰的指导，为了实现最终的目标，根据角色和用户不同，我们对应的方式和做法，将最后的what和开始的why建立了一个关联。整个过程也是基于一个假设的过程，通过一个完整的例子来看一下，目标是100万玩家，这是我们的why和实现100万玩家目标相关的角色，有玩家内部和广告商，这是who下一步是根据这些用户和角色，他们是以什么样的方式去影响到我们的目标，就是how,玩家可以邀请朋友推荐分享，内部可以组建活动，广告商可以发广告，最后就是what要想实现，how，我们需要做什么？邀请朋友可以通过半自动邀请，可以通过激励个性化设置增加产品吸引力，病毒传染，推荐可以是书签分享，可以是可供贴图的内容，内部组织活动可以进行邀请。这些都属于what。具体的做法在what中需要划分优先级，不是所有的what都要去实现。这样就将可以做什么和目标是什么？之间建立了联系，也就是将产品目标和功能之间建立了联系。相对于我们通常关心的第4层what，我们更应该把注意力放在前三层，目标、角色和影响上，尤其是角色和影响上关注点如此，优先级排序也是如此，先不要关注在what即自己要做什么样的事情上，这往往会让我们陷入执行的细节，埋头做事，而忽略了事情的初衷。影响地图的会议需要有严格的时间盒 当关键的想法已经出现在地图上，团队已经达成目标，并且确定了最小和最快的路径，暂时想不出更好的替代方案时，影响地图会议就可以结束，通过影响地图解决了业务和技术之间无法对话的问题，打通了业务和开发部门之间的墙，增量式交付。

如何实践影响地图 - 建立产品目标和功能之间的关系

目标是什么?



可以做什么?

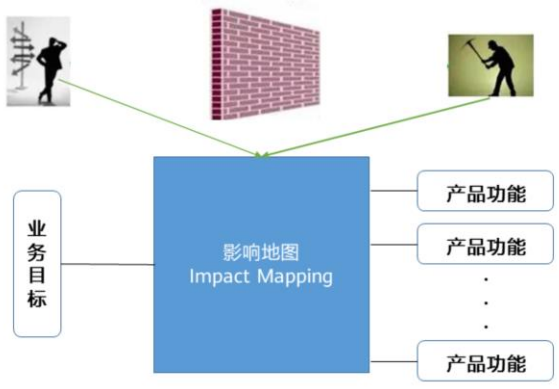




HUAWEI

解决业务和技术无法对话的问题

1. 业务部门及开发部门之间的理解、沟通、协调及隔阂
2. 目标到功能间联系的模糊和不一致



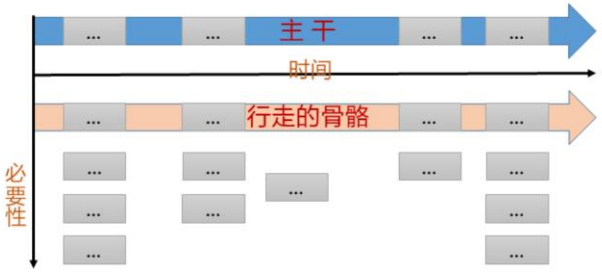
HUAWEI



- 增量式交付，还有一个问题，就是只见树木不见森林，待办列表的各种重要条目，让你无法看到全貌，这就带来了这样的几个问题：
- 一不能明显地聚焦于用户的需求，
- 二、不能方便地了解系统提供的功能的完整性。
- 三、不能方便地利用递增和迭代的方式去确定发布计划以及发布目标。
- 由此我们需要引入另外一个工具即用户故事地图。

什么是用户故事地图

- 用户故事地图：透过可视化的方式，建立用户场景与技术规格之间的联系，并辅助团队有效沟通。
- 让用户在产品还没有做出来之前就可以体验到产品所提供的功能，并以二维结构呈现产品的主线功能和辅助功能，帮助设计者决策。





第一个用户故事地图

HUAWEI

- 在用户故事地图二维的结构图当中，不仅指出了需求的优先级，也指出了他们彼此之间的关系和用户更高层目标，故事地图可以，帮助客户选择每个发布包含的产品特性并且进行分组根据产品特性的重要性，分为主干，行走的骨骼和额外的特性，这三类必须的功能组成了系统的主干，主干级别的特性，不需要进行优先级排序，包含在系统中就可以了。行走的骨骼是可能工作的最小系统，它是基于主干功能的额外的产品特性，在行走的骨骼下面，并且是基于优先级进行排序的。通过这种可视化的形式，让用户在产品还没有做出来之前，就可以体验到产品所提供的功能，并帮助客户进行选择。
- 第一个用户故事地图：https://www.oreilly.com/library/view/user-story-mapping/9781491904893/ch01.html

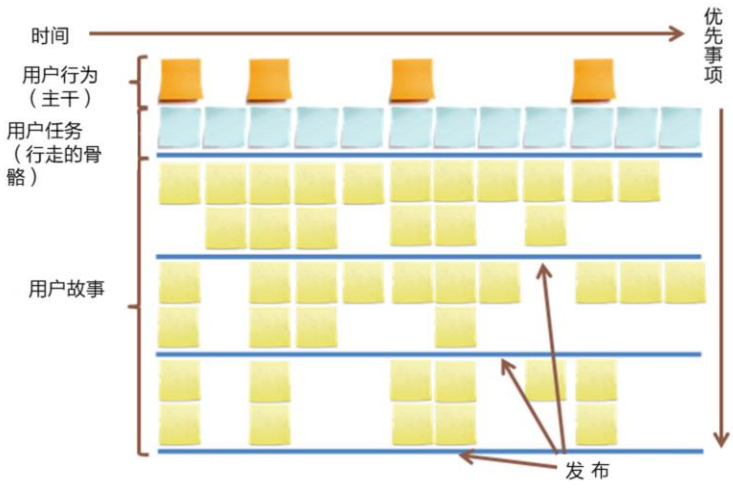
团队演练 - 典型的一天, 从起床到出门

- 请小组根据“起床到出门”的场景梳理出自己的第一张故事地图
- 时间：30分钟

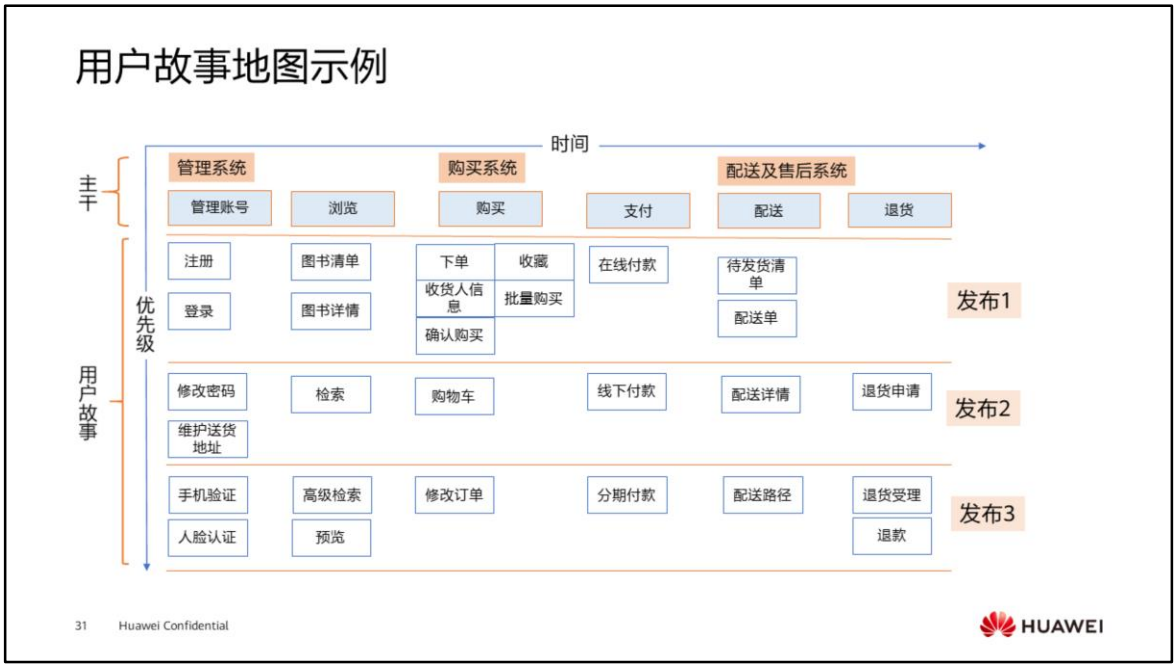


HUAWEI

用户地图的形态



HUAWEI



- 这是一个在线的购书网站，首先根据时间顺序识别出用户的关键行为，从管理、账号、浏览、购买、支付、配送和退货构成了产品的主干，然后对每个步骤进行细化和拆分，并且排列了优先级，得到了产品的用户故事。最后是综合用户的优先级和团队的速率进行版本规划，确定MVP，这就是从项目到落地计划的整个过程，总结一下，我们得出用户故事地图使用的三个步骤，第一步是广度优先，从左到右，讲述用户经历的重要步骤，要体现交互最好。第二步是深度优先，针对每个主流程进行细化。第三步就是版本规划确认MVP，这是一个需求讨论会的案例，通过使用影响地图和用户故事地图，最后产出了用户故事和功能点。

一个真实的需求讨论会案例：影响地图 + 用户故事地图



产出用户故事+功能点

US1：新用户注册





HUAWEI

- 优点：
  - 团队很聚焦，讨论效率很高。
  - 业务和技术之间产生了统一的理解，搞清楚了关键流程。
  - 讨论的输入明确，交付物很快产生，而且内容清晰，便于进行下一步的原型设计。
- 需要改进：
  - 缺少相对完整的故事列表，需要PO进行一些准备工作，以便明确这2周的工作量，确保最终完成阶段目标（完成所有故事的梳理）。
  - 会议需要更加明确的角色分工，主持人，需求记录人，问题记录人等。

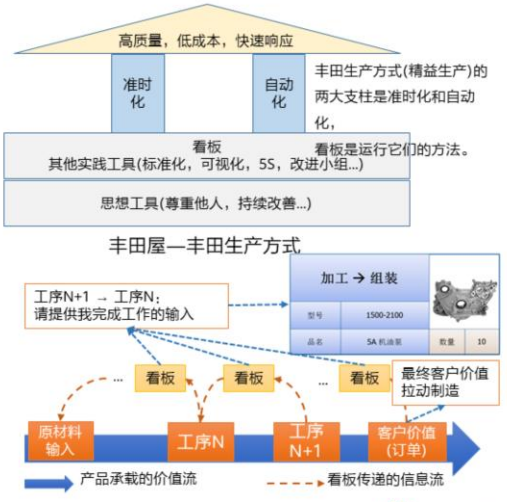
目录

1. 敏捷项目管理理念与方法实践
2. 规划与设计
3. **敏捷项目管理的理论、方法、模型**
   - Kanban 方法
   - Scrum 方法
4. 敏捷需求管理
5. 团队与协作
6. 华为敏捷项目管理企业实践

HUAWEI

Kanban 看板的含义

- 看板源自精益制造
  - 精益制造是上世纪50年代起，从丰田公司实践中演化出来的，又被称为“丰田生产方式”。直至今日，精益制造仍然是最先进的制造方式，是制造业共同追求的目标。
- 看板（kanban）一词来自日文，指可视化卡片
  - 看板工具的实质是：后道工序在需要时，通过看板向前道工序发出信号——请给我需要数量的输入，前道工序只有得到看板后，才按需生产。看板信号由下游向上游传递，拉动上游的生产活动，使产品向下游流动。拉动的源头是最下游的客户价值，也就是客户订单或需求。



HUAWEI

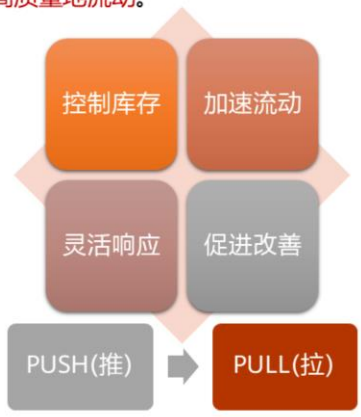
- 看板源自精益制造：
  - 精益制造是上世纪50年代起，从丰田公司实践中演化出来的，又被称为“丰田生产方式”。1990年麻省理工的James P. Womack等几位教授提炼 总结了丰田的实践，出版《精益生产方式——改变世界的机器》一书，精益制造的概念开始为世人所认识和效仿，直至今日，它仍然是最先进的制造方式，是制造业共同追求的目标。
- 看板（kanban）一词来自日文，指可视化卡片：
  - 看板工具的实质是：后道工序在需要时，通过看板向前道工序发出信号——请给我需要数量的输入，前道工序只有得到看板后，才按需生产。看板信号由下游向上游传递，拉动上游的生产活动，使产品向下游流动。拉动的源头是最下游的客户价值，也就是客户订单或需求。

拉动式生产的收益

- 精益制造体系通过看板形成拉动系统，带来控制库存、加速流通、灵活响应和促进改善等好处，最终让用户价值顺畅高质量地流动。

控制库存：下游需要时上游才开始生产，有效控制了库存。库存控制水平是工厂管理的核心指标。

加速流动：进入生产环节的物料和半成品，很快被拉入下一环节，直至变成成品，实现了保证安全库存的前提下物料最快地流动，提高了工厂的运转效



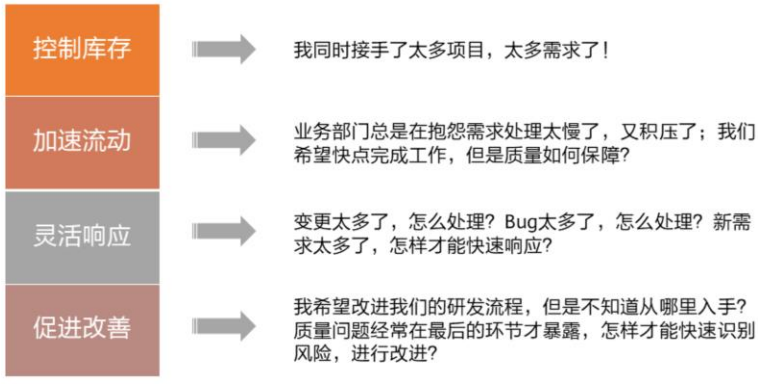
灵活响应：用户需求的变化通过看板形成的信息流快速传递至各个环节，系统做出最快的响应。同时低库存水平降低了负载，让响应更加迅捷和低成本。

促进改善：库存的降低和流动的加速，让生产环节的问题可以在第一时间暴露，这为现场现地解决问题和发现问题背后的根本原因提供了便利，也提升了系统改进的动力。

HUAWEI

思考题

- 拉动式生产是否能解决软件产品开发中的问题?



HUAWEI

建立和运作看板的五大实践

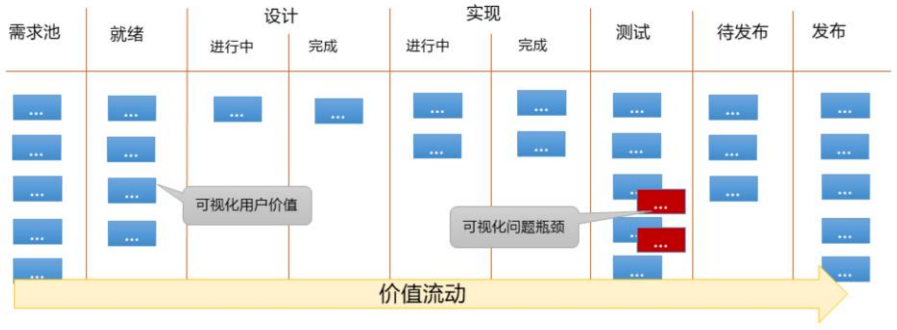


HUAWEI

- 2006 年在 Don Reinertsen 的启发和鼓励下，David J. Anderson 最早在软件开发中借鉴和应用看板实践，并总结成为完整的方法体系 ——“看板方法”

可视化价值流动

- 可视化价值流动是看板方法中最基础的实践，它涉及可视化用户价值、价值的流动过程，以及价值流动过程中的问题和瓶颈等方面。

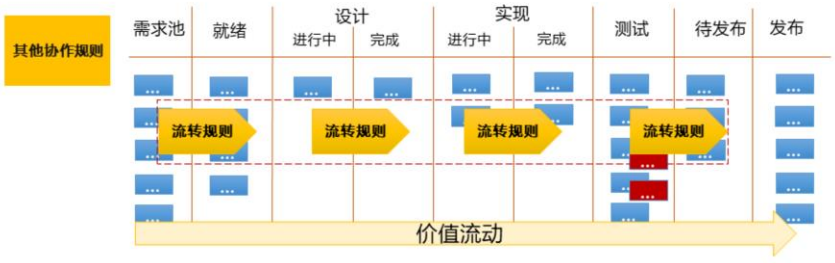


HUAWEI

- 用户价值：产品开发的目标是交付用户价值，工作也应该从用户的视角来组织。图中每一个蓝色的卡片代表一个用户的价值，典型的是一个可验证、可交付的用户需求。
- 用户价值端到端的流动过程：所谓端到端是指价值提出到价值交付的整个过程。它通常由顺序的环节构成，这其中既包括工作环节，也包括等待环节。用户价值流经这些列，直至交付给用户。
- 问题和瓶颈：问题是指那些阻碍用户价值流动的因素，如需求不明确、技术障碍、外部依赖等；瓶颈指价值流动不畅的环节，瓶颈处积压形成长工作队列。

显式化流程规则

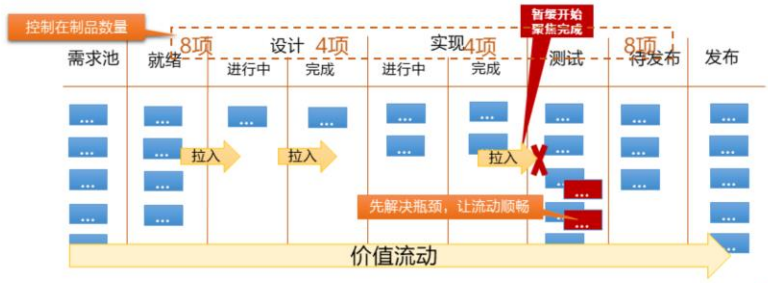
- 显式化流程规则是指明确价值流转和团队协作的规则并达成共识。显式化的流程规则是团队协作的依据，更是团队改进的基线。
  - 流转规则：在什么条件下卡片可以进入下一环节。
  - 分类规则：何种类型的工作使用何种颜色的卡片，进入哪条泳道，采用什么样的优先级。
  - 工作节奏：团队以怎样的节奏接受新的工作，怎样的节奏更新看板，怎样的节奏对外发布。



HUAWEI

控制在制品数量

- 控制在制品数量让环节内并行工作减少，单个工作项的完成加等待时间缩短，工作项从进入看板系统到完成交付的时间随之缩短。因此，控制在制品数量加速了用户价值的流动，对产品开发的敏捷性至关重要。
- 在制品是指特定环节内所有的工作项：包括进行中和等待的；橙色的数字是在制品数目的限制，环节内在制品数目小于这个数字时，可以从上一环节拉入新的工作，否则不允许拉入新的工作。
- 如某个工作长时间受阻成为瓶颈，影响到上游环节，团队应该聚焦于完成已经开始的工作，即时处理出现的问题。

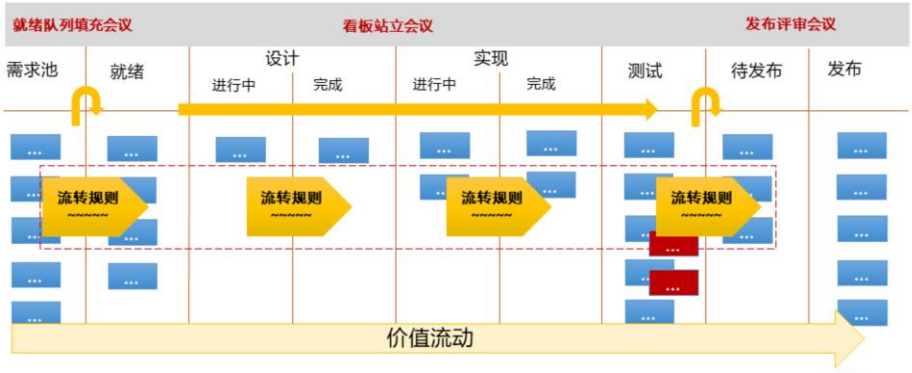


HUAWEI

- 如某个工作长时间受阻成为瓶颈，影响到上游环节，团队应该聚焦于完成已经开始的工作，即时处理出现的问题；如测试环节的在制品数量达到了上限，再拉入新工作就被禁止了。

管理工作项流动

- 管理工作流是为了让用户价值顺畅和高质量地流动，它包含三个分项实践：就绪队列填充活动、看板站会、发布评审，分别对应着用户价值的输入、中间过程和输出。

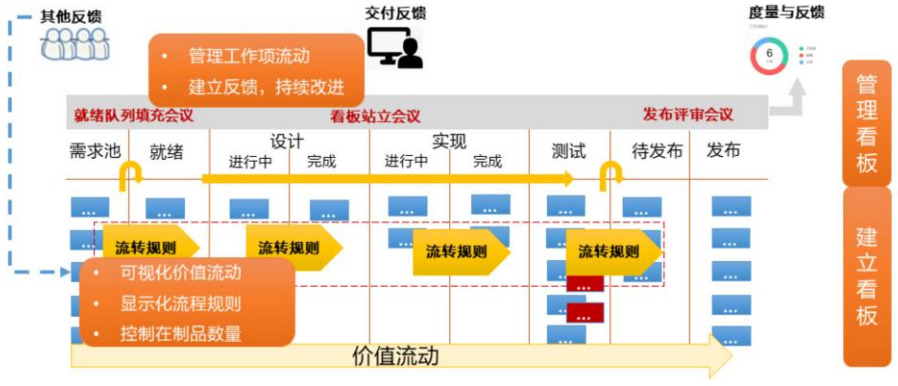


HUAWEI

- 就绪队列填充活动：就绪队列是看板系统的输入环节和价值流动的源头，管理好就绪队列的填充对价值的顺畅流动和质量非常重要。
- 看板站会：站会是管理价值流动过程的活动，一个典型的看板站会发生在每个工作日、同一时间、同一地点（看板墙前），团队成员从右至左走读看板墙上的卡片，重点关注价值流动过程中问题和阻碍，处理这些问题或提出跟踪方案。
- 发布评审：发布评审是需求发布前的计划活动，决定上线或发布哪些需求以及相关发布策略等。发布评审是一个可选的活动，如在持续交付的模式下，它很可能被例行机制所替代。

建立反馈, 持续改进

- 流动是否顺畅的反馈，如：阻碍问题分类，影响和原因分析。
- 质量问题的反馈，如开发环节和测试环节遗漏缺陷的分析等。



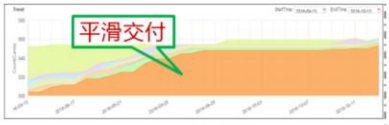
HUAWEI

不同角色关注看板的重点

PM及以上人员关注点



1、每天需求交付计划和交付数量
2、总体产能和吞吐量



1、交付是否平滑 ★★★
2、控制需求导入节奏

SL关注点



1、WIP 限制 ★★★
2、关注流动



1、关注阻塞和瓶颈 ★★★
2、聚焦解决

团队人员关注点



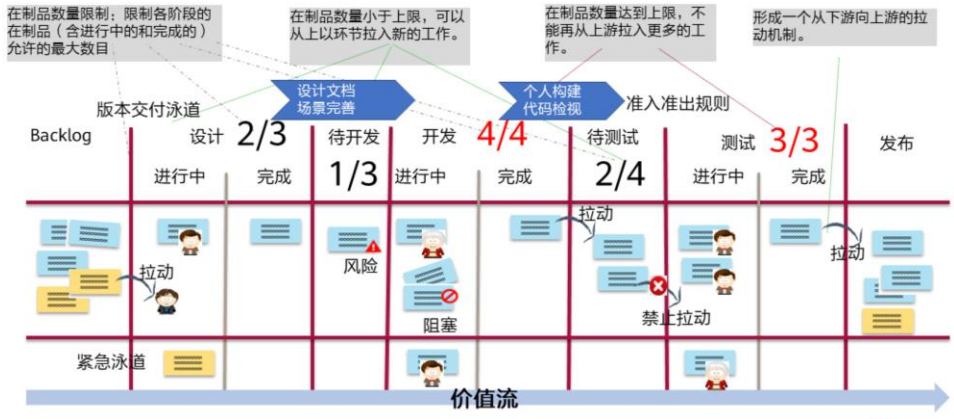
1、制定计划 ★★★
2、及时拖动



1、标识阻塞和风险 ★★★

看板展示核心元素

- 看板中，通常存在分层、泳道、列、价值流、 在制品(WIP)、风险&瓶颈、拉动式开发等核心元素，如下图：

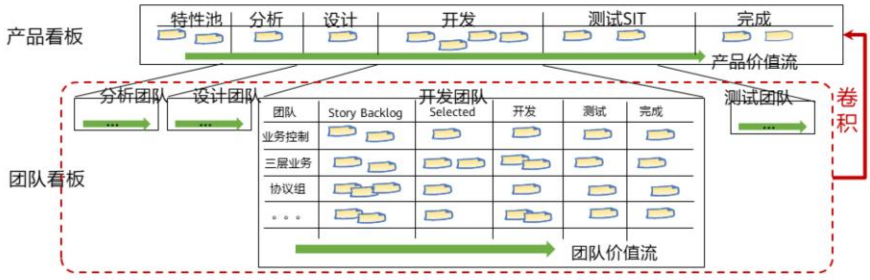


HUAWEI

- 在特定的客户市场，通过对竞争对手对于客户和市场特征需求的满足程度和产品组合竞争力的评估，形成竞争对手竞争力关键特性，通过与我司产品组合的竞争力对比，给出我司不满足的特性，推导出解决方案竞争力的需求。

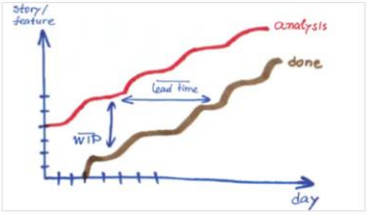
看板分层架构

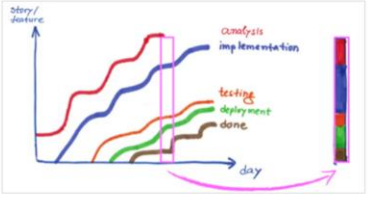
- 基于产品研发等不同视角的价值流，看板可以划分为多层视图，更好的管理产品的价值的流动。
  - 产品级看板：基于产品视角看到的研发价值流，这是每个项目开展精益看板时，首先要分析和建立的看板系统。管理产品特性的流动。
  - 团队级看板：基于设计团队、开发团队、SIT测试团队的价值流视图，这是团队开展和改进的视图。精细化管理需求在设计阶段、Story开发、需求测试的流动。



看板度量指标和方法

- 看板度量主要指标：
  - 前置时间(Lead Time)：又称为交付时间(Delivery time)
  - 吞吐量(Throughput)
  - FE流动效率
  - 准时交付率
  - 流动性&波动性
- 看板度量主要方法：
  - 价值流图
  - 累积流图
  - 控制图
  - 直方图(weibull分布图)

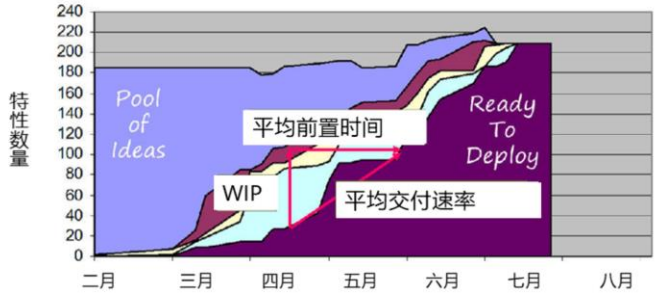




HUAWEI

累积流图 - Cumulative Flow Diagram

- 累积流程图(CFD)是可以快速概述项目或产品工作中发生的情况的图表之一；CFD中找到有关工作状态的典型信息：已完成的工作量，正在进行的工作以及未完成的工作，进度如何等



HUAWEI

- 在制品数量和平均前置时间之间存在相关性，而且是线性相关。
- 根据利特尔法则：平均交付速率 = 在制品数量(WIP) / 平均前置时间。
- 为了快速交付，最有效的方法是降低在制品数量。
- 前置时间和质量之间存在相关性，因此，提高质量的管理杠杆点在于减少在制品数量。

目录

1. 敏捷项目管理理念与方法实践
2. 规划与设计
3. **敏捷项目管理的理论、方法、模型**
   - Kanban 方法
   - Scrum 方法
4. 敏捷需求管理
5. 团队与协作
6. 华为敏捷项目管理企业实践

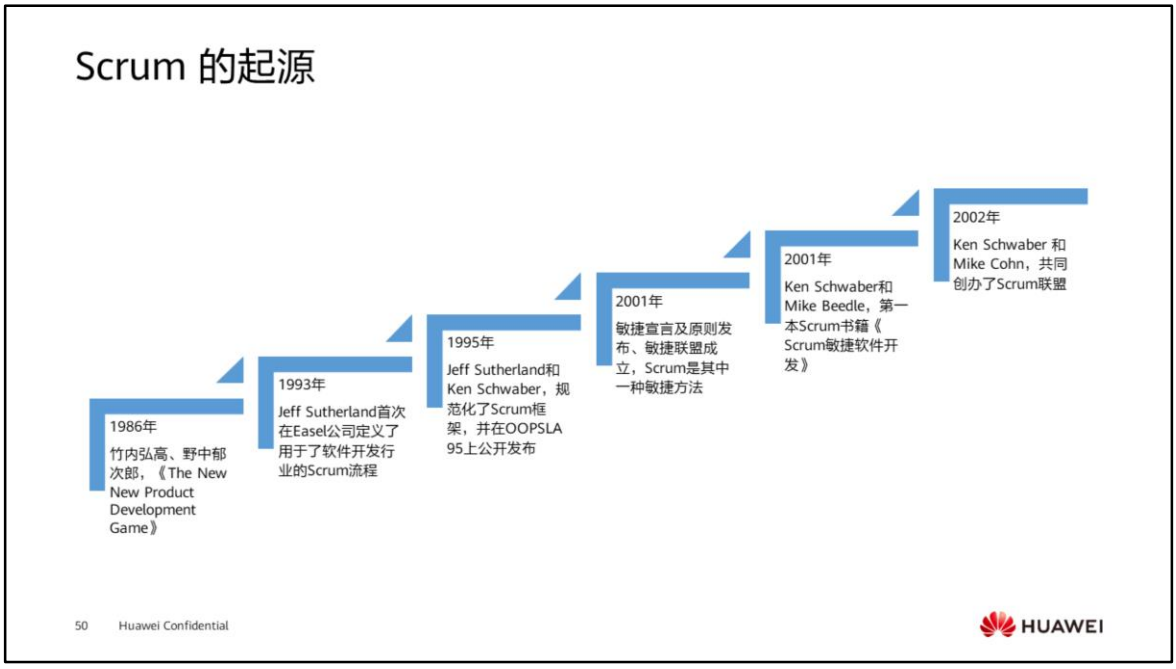
HUAWEI

Scrum 是什么?

- Scrum的英文意思是橄榄球运动的一个专业术语，表示“争球”的动作。
- 1986年，竹内弘高和野中郁次郎在《The New New Product Development Game》文章提到将Scrum应用于产品开发。
  - 他们指出：传统的“接力跑”产品开发模式已经不能满足快速灵活的市场需求。
  - 而另一种如同橄榄球赛的团队合作方式：团队作为一个整体前进，在团队的内部传球并保持前进；这也许可以更好的满足当前激烈的市场竞争。



HUAWEI



- 最早是由日本的竹内弘高、和野中郁次郎、在1986年发表的一篇新新产品开发游戏文章当中提出来的。文章里边指出说，传统的接力跑的模式已经不能满足灵活的市场需求，像橄榄球比赛的团队合作的方式，也许可以更好地满足竞争激烈的市场需求。从第一次提出到scrum联盟的创建，经历了6个重要阶段。1986年竹内弘高、和野中郁次郎、发表了文章，提出了scrum的概念。1993年Jeff Sutherland首次定义了用于软件行业开发用的scrum流程。1995年Jeff Sutherland和Ken Schwaber规范了scrum框架，并且公开进行了发布 2001年敏捷宣言及。原则发布敏捷联盟成立scrum是其中一种敏捷方法。2001年Ken Schwaber和Mike Beedle，联合出版了第一个scrum书籍，scrum敏捷软件开发，2002年，Ken Schwaber和Mike Cohn共同创建了scrum联盟，这就是scrum的起源。

Scrum 三大特点

- “可能性的”艺术
- 团队自组织，自管理
- 面对面沟通



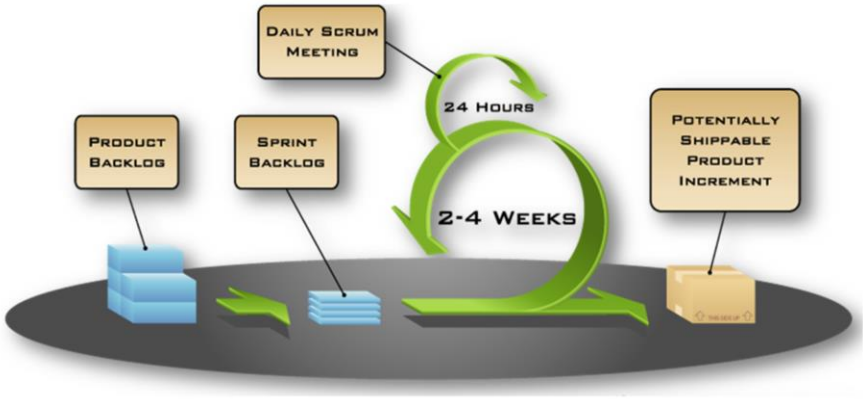




HUAWEI

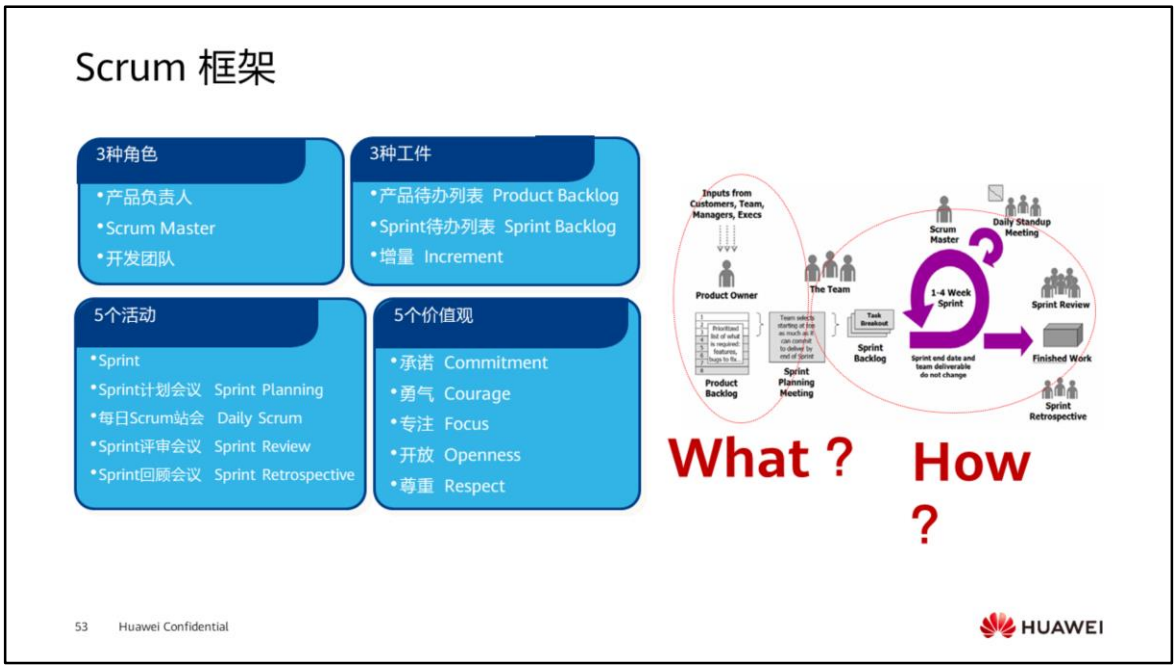
全面视角的 Scrum 框架

- Scrum是一个轻量级的项目管理的框架，它的核心在于迭代



HUAWEI

- 通过迭代去实现最终的产品。在整个框架当中，首先要有产品待办列表，在计划会议上 团队从产品代表列表当中，选择合适的条目进入到迭代的待办列表，然后团队开始2~4周的迭代开发，在迭代开发的每一天团队都会进行每日站会，迭代结束后会提交一个潜在的可交付的增量，给用户评审，迭代最后团队会共同开展一次回顾会议，针对本次迭代的内容进行回顾。



- Scrum是一个包括了一系列实践和预定义角色的过程框架。任何的软件开发过程框架都可以由最基本的三个要素组成：角色（人）、活动及其输入输出的工件，Scrum 框架主要包括以下内容：
- 角色
  - 产品负责人（Product Owner）
  - Scrum主管（ScrumMaster）
  - 团队成员
- 活动
  - 冲刺规划会议（Sprint Plan Meeting）
  - 每日站立会议（Scrum Daily Meeting）
  - 冲刺复审会议（Sprint Review Meeting）
  - 冲刺回顾会议（Sprint Retrospective Meeting）
- 工件
  - 产品订单（Product Backlog）
  - 冲刺订单（Sprint Backlog）
  - 燃尽图（Burndown Chart）
  - 新的功能增量

Scrum 团队模型（三种角色）

Scrum Master	Product Owner	团队
• 保证Scrum团队可以遵守Scrum的价值，实践和规范 • 帮助Scrum团队和组织采用Scrum模式进行项目流程组织 • 指导并带领团队变得更加高效，实现更高质量 • 保护团队不要受到外界因素的干扰 • 保证各个不同角色之间的良好写作，消除障碍 • 帮助PO更好地利用团队的能力 • **不要管理**团队	• PO 是一个人并只能由一个人来担任 • 负责管理产品待办事项表（Product Backlog）并保证其对于客户和团队保持透明度 • 对产品代办事项表进行优先级排序 • 与团队一起来进行工作量估算 • 对于项目的成功负责并保证投资回报率 （ROI）	• 最佳团队大小：5-9 人 • 多功能团队：程序员，测试人员，设计师，数据库管理员和架构师 • 保证团队成员全职参与开发 • 自我管理，没有头衔之分，不组建子团队 • 成员更替只能在迭代之间进行，最佳方式是在发布之间进行

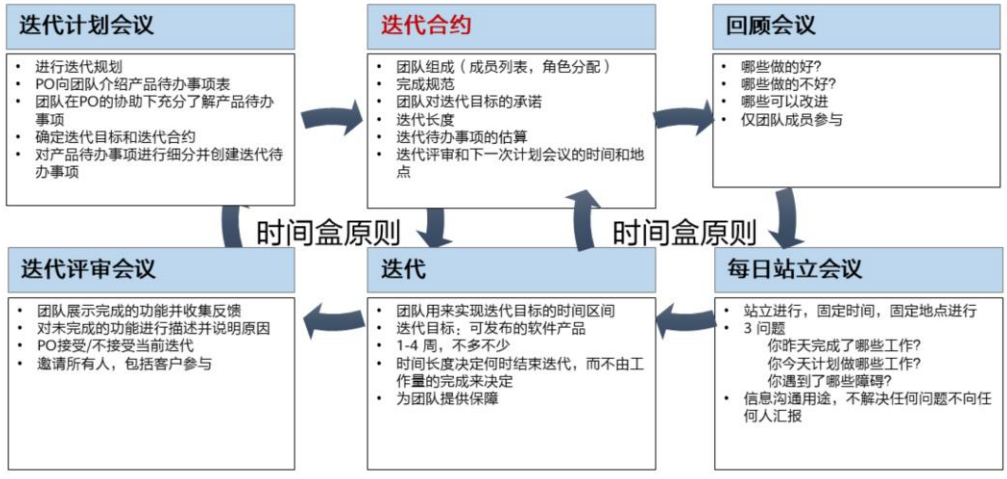
HUAWEI

Scrum 三种工件

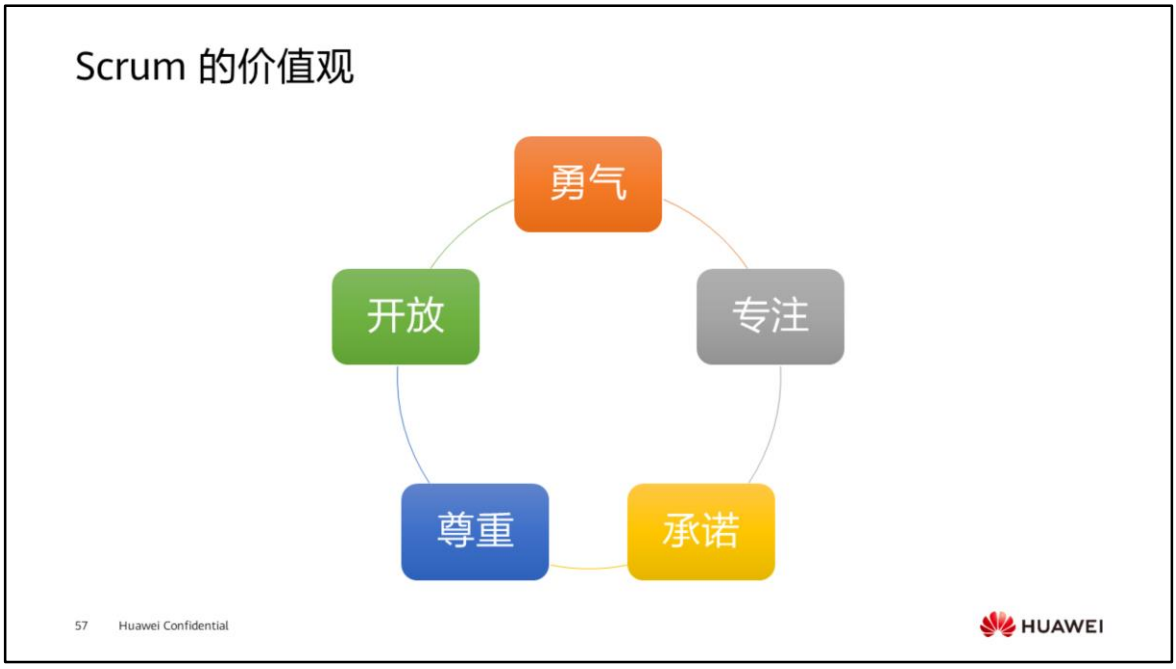
产品Backlog	Sprint Backlog	产品增量
• 产品需求变动的唯一来源 • 动态，永不完整，持续更新 • 有序，排序越高越清晰具体；排序越低， 细节越少 • 每个产品一个，与团队数量无关 • 产品负责人负责管理其内容，可用性和排序	• 包含产品待办事项列表中当前 Sprint 的子集 • 包含完成 Sprint 目标所需的任务细节 • 开发团队可视情况增加或移除任务	• 当前 Sprint 完成的产品待办事项列表，以及之前所产生增量的总和 • 必须达到"完成"的标准 • 无论是否发布，必须是可用的

HUAWEI

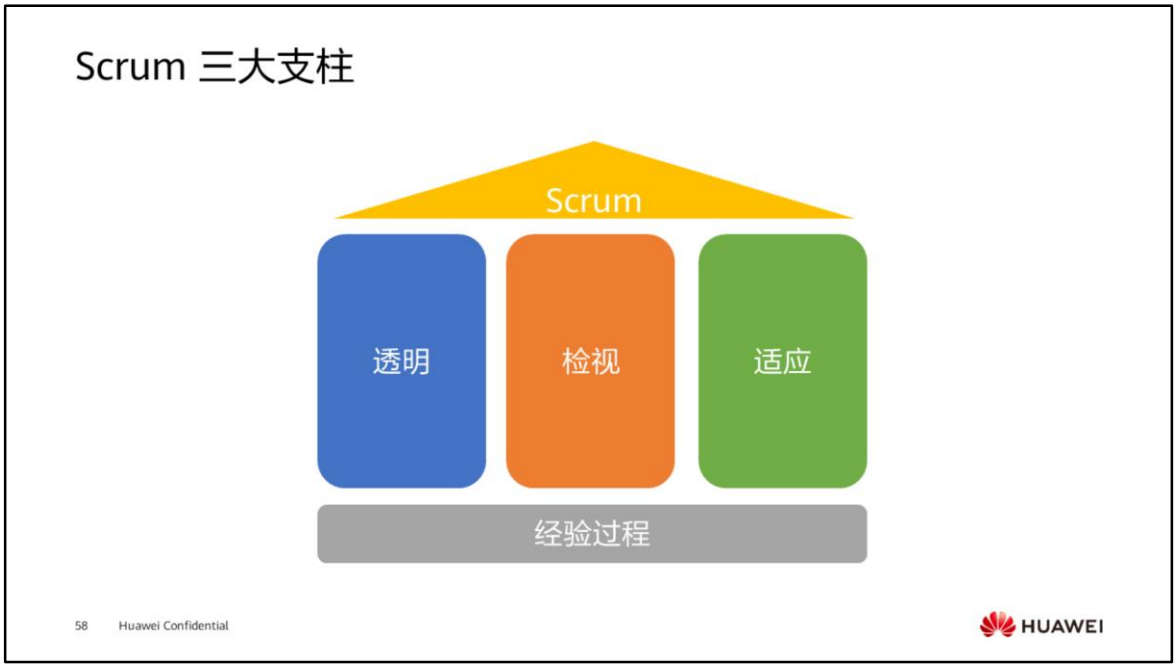
Scrum 过程模型 （5个活动+1个合约）



HUAWEI



- scrum的价值观 承诺是愿意对目标作出承诺，专注是把你的心思和能力都用到你承诺的工作上去，勇气是要有勇气作出承诺，并且要履行承诺，接受别人的尊重。开放是scrum让把项目当中的一切都开放给每个人看，尊重是每个人都有他独特的背景和经验，我们都要给予尊重。



- 在软件开发过程当中，常用的控制理论有两种，预定义控制和经验过程控制，预定义过程控制类似于瀑布开发模型，而经验过程控制理论是敏捷的开发模式。scrum是一个敏捷开发框架，它的基础也是基于经验过程，所以经验过程的透明、检视和适应就是它的三大支柱。透明就是通过任务板的形式，把项目中的任务和资源等进行可视化，在每日站会评审和回顾等环节都是进行解释的环节。在检视过程当中发现了偏差，就要进行调整，以适应当前的情况。

练习 - 组建Scrum团队

- 内容：
  - 组内讨论，在提供的三个项目中，确定本组所选项目
  - 组内分配人员角色，组建Scrum团队
  - 各角色在小组内阐述自己的角色职责
- 目的：帮助学员了解Scrum团队的组织架构，并帮助学员理解Scrum各角色的职责
- 时间：10分钟



HUAWEI

目录

1. 敏捷项目管理理念与方法实践
2. 规划与设计
3. 敏捷项目管理的方法、模型
4. **敏捷需求管理**
   - **用户故事**
   - 敏捷估算
5. 团队与协作
6. 华为敏捷项目管理企业实践

HUAWEI

什么是用户故事?

- 用户故事描述了对用户，系统或软件购买者有价值的功能，用户故事由以下三方面组成：
  - 一份书面的故事描述卡片，用来做计划和作为提示。
  - 有关故事的交流，用于具体化故事细节。
  - 测试，用于记录和确认故事细节且可用于确定故事何时完成。
- 用户故事是敏捷开发方法中用来定义系统需要提供的功能和做为需求管理条目化的基础。



卡片(Card)



交流(Conversation)



确认(Confirmation)

HUAWEI

描述用户故事

- 作为 X [什么用户角色]，
- 我希望 Y [系统提供什么功能] ，
- 以便 Z [达到什么目的或实现什么价值]

或者



- 为了 X [达到什么目的或实现什么价值]，
- 作为 Y [什么用户角色]，
- 我希望 Z [系统提供什么功能]

- 客户最终需要的并不是文档
- 而是通过软件帮助其完成业务价值
- 短小精悍的故事可以帮助我们推进沟通，挖掘客户的真实需求。

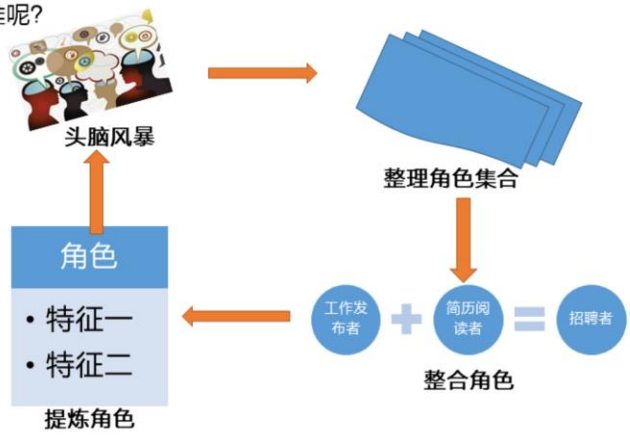
相对于编写好的用户故事，
产生和讨论的过程更加重要！

HUAWEI

- 写到一张卡片上的里面包含故事的具体信息，场景化。如果我们用电子产品的话，就可以不使用这种实体卡，直接录入到电子产品里。交流是指相比于传统的需求，靠纯文档传递，用户故事来自交流，所以在用户故事卡片上要记载交流的细节，帮助开发团队更好地理解需求。测试 用户故事最终是要通过测试的，所以在用户故事里就要写上检测点 这些检测点往往是在。写用户故事以及交流的过程中就写出来的检测点，只有通过这几个检测点才能说该故事测试通过了，这是用户故事描述的核心。AS who， I want to what？ so that why。

用户故事 - 角色类型的创建

- 我们将要开始开发一个招聘网站项目，这类网站会有很多不同类型的用户。当我们编写用户故事的时候，我们所说的用户是谁呢?



HUAWEI

- 我们开发一个招聘网站，这类网站会有很多的用户，所以当我们在编写用户的时候，我们所说的用户究竟都有哪些呢！这里面来介绍一套方法，首先是头脑风暴，然后我们去整理角色，接下来去整合他们！这些角色，最后提炼出来这个角色，这一套方法具体该如何去做？接下来我们会一步步给。大家展示出来！首先是头脑风暴，在头脑风暴环节的话，我们只创建角色，不讨论角色！团队！所有人聚在一起，尽可能去想，跟这个网站相关的都有哪些角色？比如说有求职者，比如说有猎头，比如说有人上来看看简历！比如说有失业了的人上来寻找一些岗位，诸如此类。大家不要去想每个角色到底在这个网站上会去做什么，你不要去想这些角色之间有没有重复性，我们就是头脑风暴，尽可能去想，将所有的角色全都想出来。头脑风暴完毕后，我们已经列出了大量的用户角色，这个时候我们可以将这些角色放在一起进行分组，将那些有关系的角色的卡片进行重叠。
- 比如说大学毕业生和初次找工作者，我们发现这两类角色他们之间的关系性很大，所以它们的重叠度比较高。再比如说工作发布者和阅读者，这两个其实都是招聘者所有的属性，所以他们和招聘者之间有一定的关系。用这个方法我们将这些卡片进行了分组，分组之后我们进一步去整合浓缩这些角色。每一个卡片的作者再去详细说一下卡片？背后有什么意义这个角色究竟代表什么？以便于团队进一步去讨论它们之间的关系，然后决定如何处理这些有重叠关系的卡片。如果两个重叠的卡片对我们的意义是等同的，就可以选择保留其中一个，丢弃另外一个，或者合并成一个新的，比如说大学毕业生和初次找工作者，如果说我们认为这两个角色对我们来说，他们的区别可以忽略不计，那么我们就可以将它们整合成一张卡片。

- 再比如说对于招聘者来说，他没有必要区分成工作发布者还是简历阅读者？我们认为这两个角色都应该是招聘者所独有属性，这时候我们就可以将他们去进一步整合。整合之后，我们看到说大学毕业生已经被拿掉了，有了初次找工作者，而刚才的简历阅读者，我们已经删掉了，这个时候我们又进一步发现，招聘者其实应该分出内部招聘者和外部招聘者。因为招聘网站上可能有一些公司的猎头过来看，也有一些。第三方的猎头！专门给市场上 给其他单位去找这种推荐者，我其实有必要将它区分出来。
  所以在进一步整合之后，我们可以将卡片列出当前这种样式，这样的结果是我们可以很清楚的看出来它们之间的关系，可以对这些角色进一步的了解。接下来我们可以去提炼这些用户角色，将每一个用户角色我们去寻找他们的一些特征列出来。这些特征我们在写的时候可以考虑以下几个方面，比如说操作计算机的熟练程度，专业技术水平、外语能力等等，这些全是根据我们角色的实际情况去列出来。当我们将每一个角色的所有特征都列出来之后，我们就制作出了这一个角色的用户画像。这时候我们就创建好了用户模型，用户模型创建好之后，我们可以开始收集我们的用户故事。

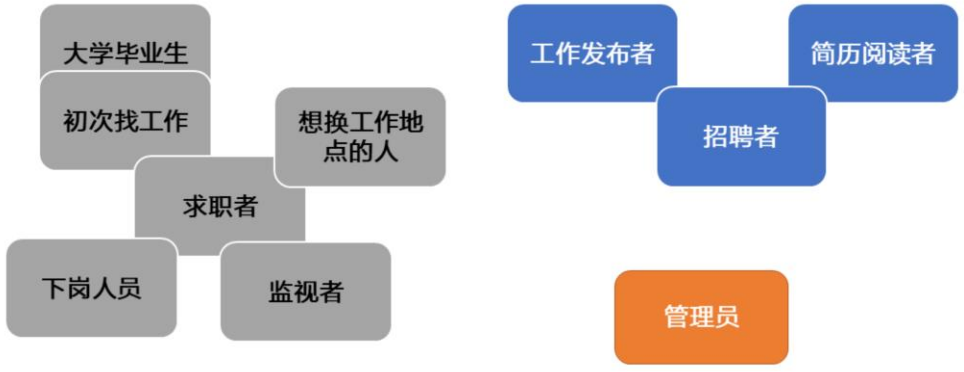
角色类型的创建 - 头脑风暴

- 只创造角色
- 不讨论角色



HUAWEI

角色类型的创建 - 整理角色类型集合



HUAWEI

角色类型的创建 - 整合角色类型



HUAWEI

角色类型的创建 - 提炼角色类型

- 角色特征考虑方面：
  - 操作计算机的熟练程度
  - 专业技术水平
  - ……

下岗人员	内部招聘者	管理员
• 特征1 • 特征2	• 特征1 • 特征2	• 特征1 • 特征2

HUAWEI

用户故事 - 搜集故事



HUAWEI

用户故事 - INVEST原则

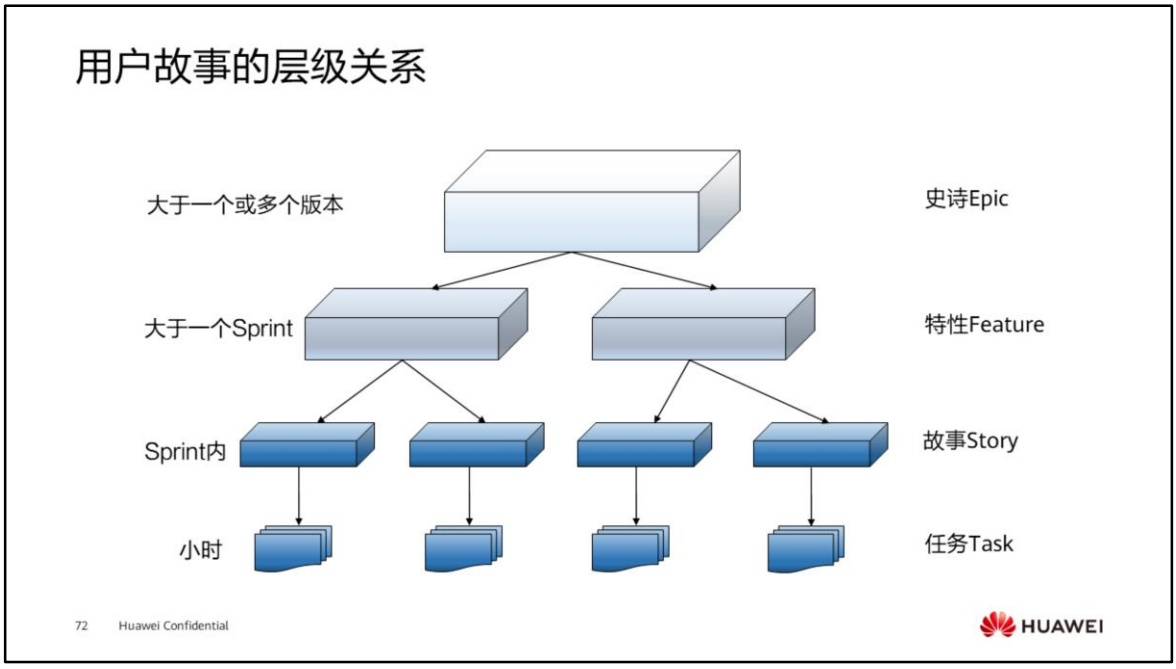


HUAWEI

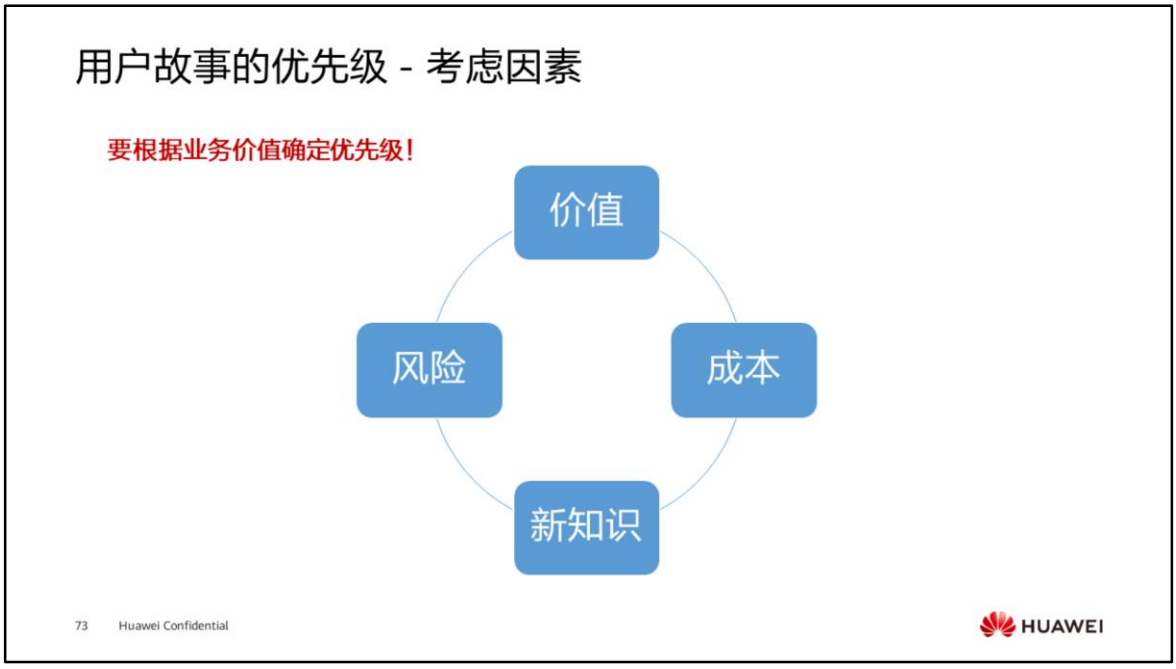
练习 - 用户故事

- 项目名称：二手房交易网站
- 内容：
  - 按照前面所讲的方法，组内确定出本项目中可能出现的角色。
  - 按照用户故事的格式，尝试描述出尽可能多的用户故事。
- 时间：20分钟

HUAWEI



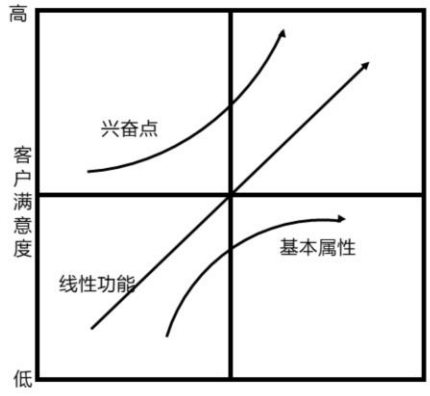
- 业界通用的是四个层级，史诗 特性、故事 任务 从上到下依次减小。史诗一般是大于一个或多个版本才能完成的。而特性一般都是在多个迭代完成。故事的话我们一般都是在一个迭代内完成，尽量不要让用故事跨迭代。最后的这个任务，它就是最小一级的，它都是按几个小时来计算的。



- 最主要的就是要根据业务价值来确定优先级，即用户故事能给客户带来多少经济上的收入，这就是最重要的。成本方面，实现该需求需要花费的成本。比如说客户建议增加一个功能，不考虑了它带来的好处，以及需要花费的成本之后，就会改变，优先级。举个例子，登录时 页面加载需要控制在一秒钟之内，但是这样需要一人周的工作量，用户考虑了一下，决定降低它的优先级，先去做别的需求。风险，故事不能预期完成的风险。根据现状，是否能在当前迭代里完成这个需求？如果完不成会带来什么问题？这就是我们考虑的风险因素 最后新知识会带来一些成本上的增加和风险上的增加，以及说假如这个迭代用时间去学习新知识的话，我们还要考虑在未来的几个迭代里面，是否会继续用到新知识？

用户故事的优先级 - 渴望度 Kano 模型

- 当我们购买一件产品时，有哪些功能是我们觉得理所应该有的?
- 哪些是我们觉得越多越好的?
- 哪些功能如果有的话是会领我们觉得兴奋的?



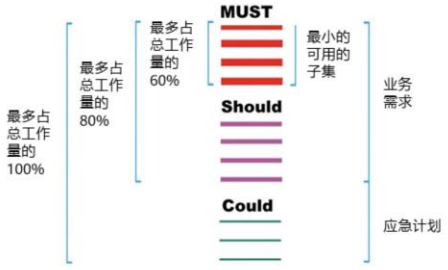
HUAWEI

- 比如冰箱制冷，有这个功能 客户觉得很正常，并不会很兴奋。但如果没有的话，那客户就会投诉你了。连最基本的功能都没有。线性属性是产品应当有的，并且产品的功能和客户满意度是成线性的。比如冰箱制冷时的噪音，噪音越小客户越满意。兴奋点属性，产品本身就没有，客户也没有去关注他没有这个预期。但如果你有了的话，客户会非常兴奋。比如说你的冰箱可以播放音乐，客户本身并没有预期这个功能。你没有的话他也觉得还可以。但如果你有了的话，他会非常满意。这就是kano模型，我们可以借助这个模型来帮助我们给予，不同的功能以不同的优先级。

用户故事的优先级 - MoSCoW原则

Must Have	Guaranteed	>20% effort	**M**inimum **U**seable **S**ubse**T**
Should Have	Expected	@ 20% effort	
Could Have	Possibly	@ 20% effort	
Won't have this time			





HUAWEI

- moscow原则主要分为四个方向。must should could have won't have this time。must have。基础功能是一定要有的。should have是很重要，但是短期内有替代方案。如果项目没有时间约束的话，这部分功能是认为应该有的。could have。如果没有时间就可以 在现在的项目中不予考虑。won't have客户希望拥有，但是我们一定要同时承认它 需要在后续发布中实现。这次发布中是不会有的。

练习 - 用户故事排序

- 内容
  - 按团队喜欢的方式，尽可能全面的考虑用户故事优先级的影响因素，队内讨论并记录这些因素。
  - 按照讨论的结果，为之前编写的用户故事进行优先级排序。
- 时间：20分钟

HUAWEI

用户故事的优势

- 强调口头沟通
- 便于理解
- 大小适中
- 适合迭代开发
- 鼓励延迟细节
- 适应变化
- 参与性设计
- 传播隐性知识

HUAWEI

用户故事出现问题的迹象



HUAWEI

目录

1. 敏捷项目管理理念与方法实践
2. 规划与设计
3. 敏捷项目管理的方法、模型
4. **敏捷需求管理**
   - 用户故事
   - **敏捷估算**
5. 团队与协作
6. 华为敏捷项目管理企业实践

HUAWEI

相对估算与故事点

- 什么是相对估算?
  - 当我们根据户型图来预估粉刷每个房间工作量时，是否能准确说出每个房间会耗费多少时间?
- 故事点：用于表达用户故事、特性或其他工作总体大小的度量单位。



HUAWEI

- 相对估算-对比参照物进行估算。故事点-参照物。

故事点可以引导对整个故事的思考

- 当我们用理想人天来估算整个故事的工作量时，很难避免团队中不同岗位的人员以自身的工作量进行分别估算



工作的前后依赖

各自的工作划分

HUAWEI

- 故事点与理想人天的比较。团队成员常常以自身岗位对整个故事进行估算。4开发人天+2测试人天+3PO人天对不同角色的强调会让团队成员偏离“我们是一个整体”。9人天这种说法更好。故事点以整个团队的角度出发，关注一个故事有多大。

故事点的优点

- 项目开始时，5 个故事点的故事耗费团队 3 天完成
- 项目尾声时，同样 5 个故事点的故事是什么情况?
- 如果此时用来估算工作量的是理想人天，那么又将发生什么情况?



这是一条5分钟的路!

我的理想人天不等于你的



这是一条8分钟的路!

HUAWEI

- 故事点-不变的。
- 理想人天可能改变-人员熟练。

理想人天的优点

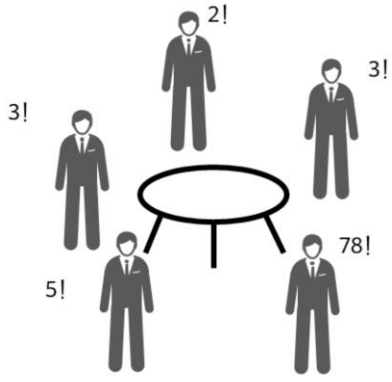
- 更容易跟团队外的人解释
- 估算更容易开始
- 便于预测速度



HUAWEI

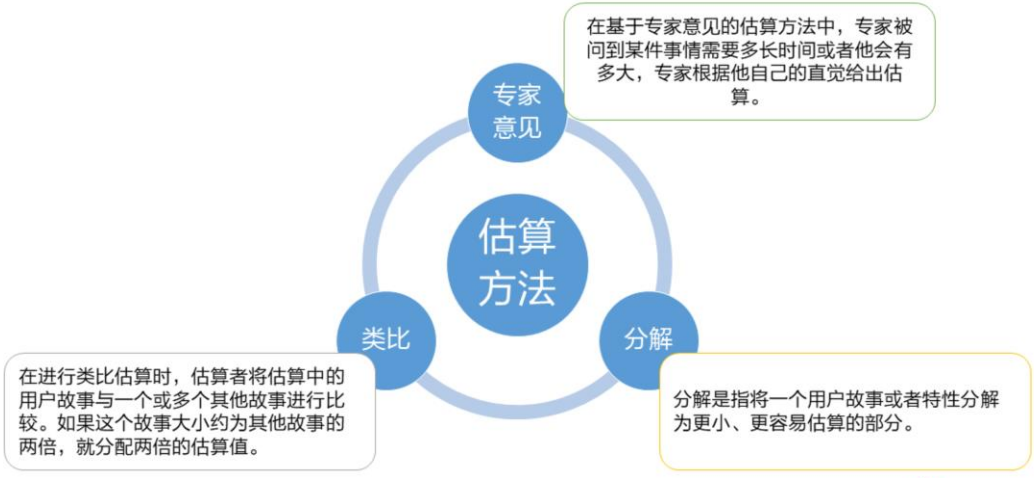
估算方法 - 原则

- 这个故事应该估算为几个故事点?



HUAWEI

估算方法 - 如何确定估算值

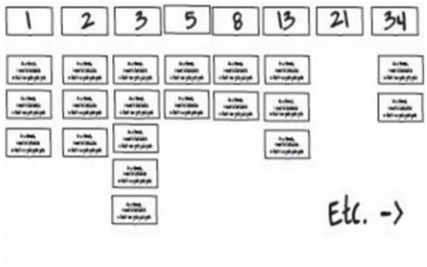


HUAWEI

估算扑克

- 用户故事的规模
- 受以下因素影响
  - 工作量
  - 复杂程度
  - 不确定性





* 斐波纳契数列 （Fibonacci Sequence）

0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100, ?



HUAWEI

- 估算扑克是一个辅助工具，具体玩法是在护送过程中每个人手持一套牌，估算完毕。所有人亮手中的牌，然后得出团队的估算值。这里的重点在扑克牌的面值，不是连续数字，而是斐波那契数列。为什么要这么做呢？因为用户故事的估算并不要求很精确，不推荐在这上面花太多的功夫。所以当你在纠结这个故事点是3还是4的时候，问题解决了。扑克牌里没有4，只有3和5，所以你很容易判断出来应该给几个故事点。

练习 - 使用估算扑克对用户故事进行估算

- 内容：
  - 由产品负责人牵头，组内逐一对故事的内容进行讨论，对于需求阐述不明确的地方，由产品负责人向小组成员阐明
  - 应用估算扑克对明确的用户故事进行估算
- 时间：40分钟



HUAWEI

- 敏捷估算2.0：如果要对估算扑克进行改进，有哪些可以提高效率的改进方向?
- 计划扑克是Scrum团队应用最广泛的敏捷估算方法，但是有时候计划扑克玩起来耗费比较多的时间，尤其是在十人左右的团队中。Ken Schwaber在他的书《Agile Project Management with Scrum》中指出，在进行迭代规划时，迭代计划环节应该控制为一个4小时的固定时间，但是从战术的角度看，如果一个会议持续4小时，大部分的参会者会有精疲力竭的感觉，并且很难保持持久的注意力。
- 为了解决这个问题，Brad Swanson 和 Björn Jensen在上海Scrum Gathering（2010/4/19）上介绍了Agile Estimating 2.0技术。这个新的估算技术同样基于的专家意见，类比和分解，同样适用Fibonacci数列，但是它可以显著地缩短会议时间。
- n 第一步是由Product Owner向团队介绍每一个用户故事，确保所有需求相关的问题都在做估算前得到解决。
- n 第二步是整个团队一起参与这个游戏。只有一个简单的游戏规则：一次仅由一个人将一个用户故事卡放在白板的合适位置上：规模小的故事在左，大的在右，一样大的竖向排成一列。整个团队轮流移动故事卡，直到整个团队都认同白板上的故事卡的排序为止。
- n 第三步，团队将故事点（Story Point）分配给每个用户故事（列）。最简单的做法是使用投票来决定每个用户故事分配到哪一个Fibonacci数字。
- n 最后一步：使用不同颜色来区分影响估算大小的不同方面，并且重新考虑是否需要修改估算值。例如，我们使用红色来表示那些无法被自动化测试脚本覆盖的用户故事，因此那些用户故事需要一个更大的数字来容纳手工回归测试的代价。
- 在国内一些企业多次实践Agile Estimating 2.0之后，反馈的结果还是令人兴奋。据称，团队对于估算的准确性更有信心了，并且只耗费原先的1/2时间。
- 使用Planning Poker进行故事点估算。
- 使用Agile Estimating2.0进行故事点估算。通常耗费的时间会缩短。

分解用户故事 - 何时分解

- 当我们成功编写了一个用户故事以后，由于各种原因，我们往往需要把一个用户故事分解成多个更小的部分



HUAWEI

分解用户故事的方法 - 数据边界分解

- 按照用户故事所支持的数据边界来分解大型用户故事

作为用户，我可以输入我的资产负债表信息：



HUAWEI

分解用户故事的方法 - 操作边界分解

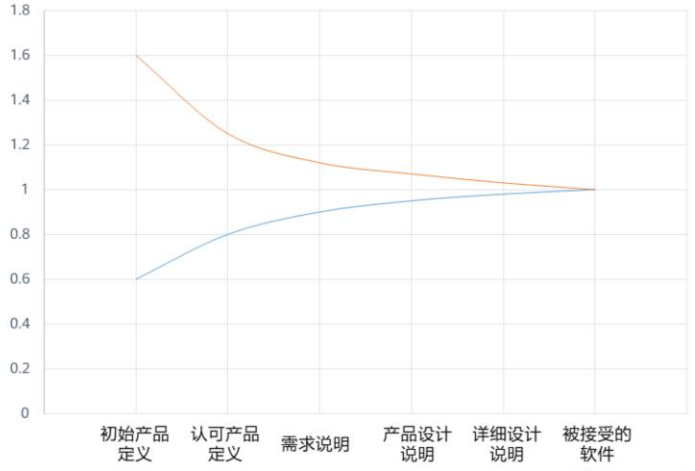
- 按照大型用户故事中进行的操作对齐进行分解

什么是你能想到的最简单的按操作边界分解的例子?

HUAWEI

估算速度 - 使用历史值

- 使用的技术是否一样?
- 针对的领域是否一样?
- 开发团队是否一样?
- 产品负责人是否一样?
- .....



HUAWEI

估算速度 - 预测



HUAWEI

DoR 与 DoD

DoR (Definition of Ready)

敏捷开发发展后，发现进入迭代开发应当满足一定条件，否则过于模糊的需求会导致迭代失败，在迭代内花费过多的时间去做需求澄清，因此给进入迭代设立门槛，称为Definition of Ready，简称为“DoR”。

常见的 DoR

用户故事澄清
用户故事的故事点估算
用户故事的验收条件

DoD (Definition of Done)

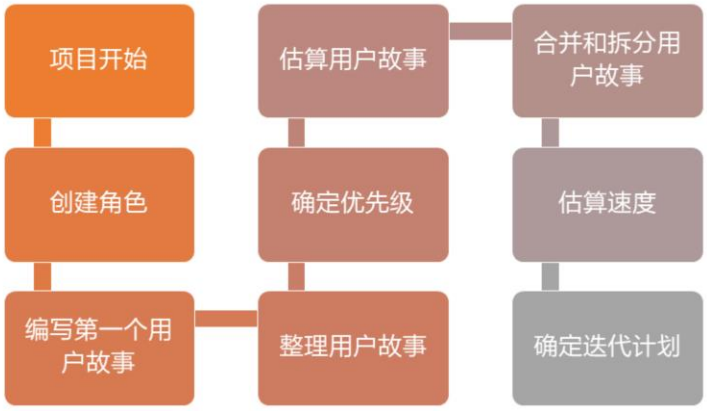
在敏捷软件开发中，常用 Definition of Done“完成的定义”来表示工作是否已完成，不同的活动有不同的完成定义。

DoD 类型及常见规则

DoD类型	常见规则
用户故事 DoD	用户故事最终的描述符合 INVEST；用户故事得到测试用例的对应覆盖；用户故事得到对应的自动化测试用例；用户故事得到 PO 试用并初步认可
迭代 DoD	所有代码通过静态检测，严重问题已修改；所有新增代码得到人工评审；测试用例都已执行
发布 DoD	完成发布规划所要求的重点需求；至少通过一次全量回归测试；修复所有等级为1、2的缺陷，3、4级缺陷不超过20个
版本 DoD	产品文档已全部更新；代码已部署到产品服务器上；运维在验收测试环境上冒烟通过；原始需求提交人对功能已经验收通过；对运维、市场、客服的新功能培训已完成
每日 DoD	下班前检入当天编写的代码；当天的代码在当天或者第2天邀请同伴进行代码评审；检入的功能代码要有对应的单元测试；每天晚上触发静态代码检查、自动化回归测试；当天持续集成、构建环境中的问题当天解决

HUAWEI

案例模拟 - 项目流程



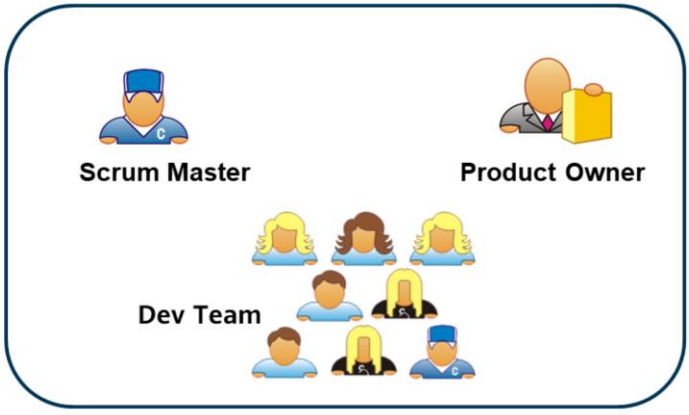
HUAWEI

目录

1. 敏捷项目管理理念与方法实践
2. 规划与设计
3. 敏捷项目管理的方法、模型
4. 敏捷需求管理
5. **团队与协作**
6. 华为敏捷项目管理企业实践

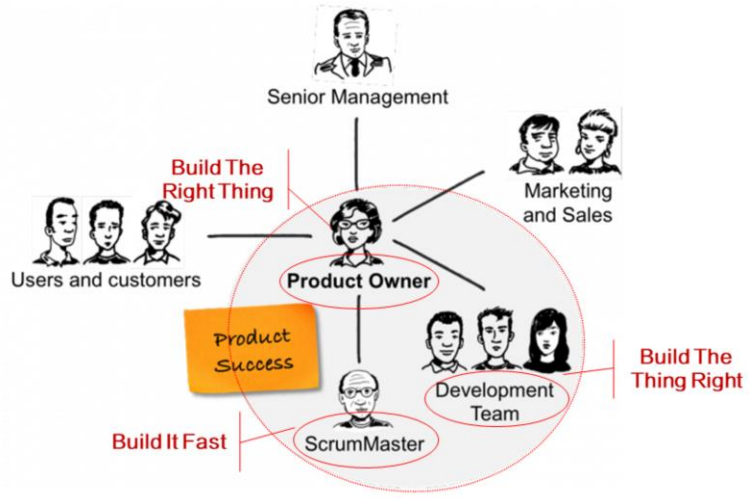
HUAWEI

Scrum中的角色



HUAWEI

Scrum 团队 - 角色分工概览



HUAWEI

Product Owner - 产品负责人

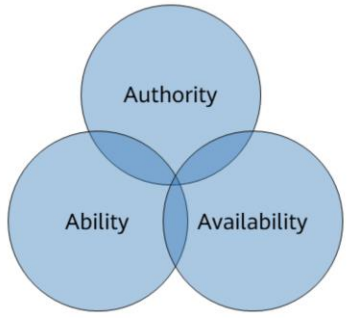
- 客户代表
- 定义所有产品功能
- 决定产品发布的内容以及日期
- 对产品的投入产出负责
- 根据市场变化对需要开发的功能排列优先顺序
- 合理的调整产品功能和迭代顺序
- 认同或者拒绝迭代的交付



HUAWEI

Product Owner - 特征

- “一”个人, 而非一个委员会
- 被授与产品（“做什么”）决策权
- 承诺投入到合作中并且在需要时被团队找到
- 企业家、系统思考者、创新者
- 驱动产品走向成功
- 提供产品领导力
- 和所有人合作
- 理想情况下是真正的用户来担任



HUAWEI

Product Owner - 职责

- PO是?
  - 产品经理? 愿景和方向，结果负责
  - 需求分析? 业务分析和需求分析
  - 需求管理? 维护、终止和变更
  - 项目管理? 优先级排序和项目状态跟踪
  - 质量保障? 检查产品结果
  - 客户代表? 产品体验/接受拒绝
  - 对外职责? 老板、客户和用户、营销和销售...

HUAWEI

Product Owner - 使命

- 建立产品愿景
- 负责最大化投资回报
- 从“为什么”开始
- 定义产品功能（“做什么”）
- 确保迭代输入准备好
- 根据反馈，为最好地实现业务目标将产品Backlog排定优先顺序和调整需求待办清单
- 决定版本发布日期和内容
- 参加各个Sprint事件
- 接受或退回工作成果

HUAWEI

Dev Team - 开发团队

- 经典团队拥有 5-9 人
- 跨职能团队
- 全职投入且长期存在
  - 特殊职能可以例外 (例如, 数据库管理员)
- 团队关系在一个迭代中应该是固定的，个人的职能可以在新迭代开始时发生调整
- 团队自我组织和管理
  - 没有管理者头衔
  - 全员负责制
  - 团队承担大部分微观管理工作并决定如何一起工作



HUAWEI

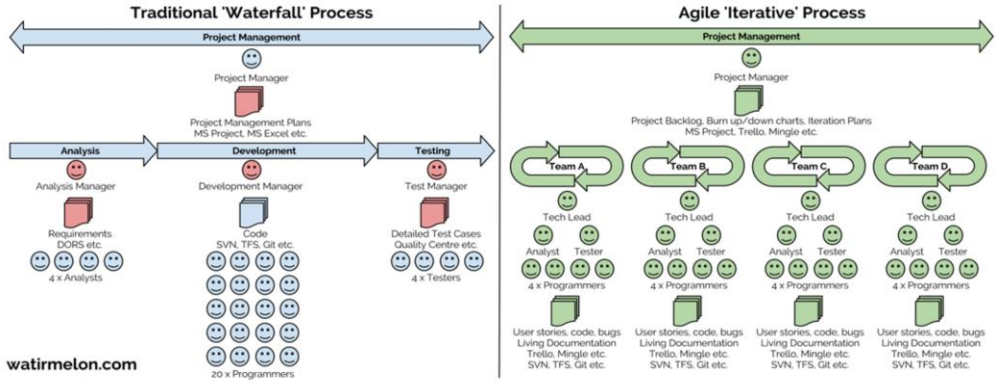
- 跨职能团队
  - 具备不同的职能
  - 端到端的特性团队
  - 没有子团队，没有头衔
  - T-Shape人才，持续学习的通用专才

Dev Team - 使命

- 与客户紧密工作在一起
- 决定迭代的工作容量
- 每个迭代交付产品增量
- 对“怎样做”和交付质量负责
- 管理Sprint Backlog并跟踪进度
- 找到团队内部合作的最佳方法
- 与其他团队及相关方协作
- 持续自我改进

HUAWEI

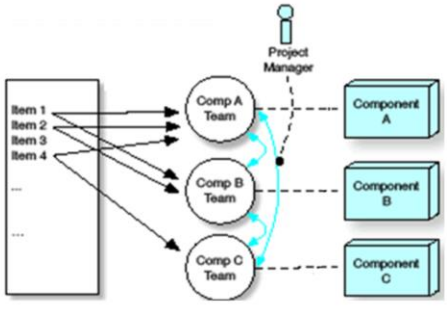
Dev Team 跨职能团队



HUAWEI

Dev Team 特性团队

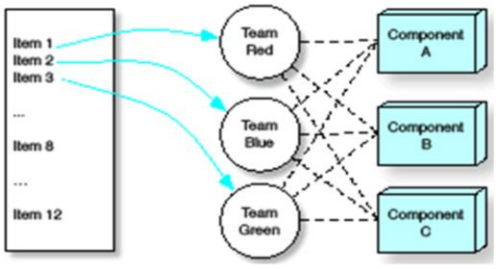
Component Team



Component Team的组织方式会导致瀑布式的开发流程，以便协调团队间的工作任务。

VS

Feature Team



Feature Team组织方式的成功依赖于团队能力，以及团队对于重构、持续集成、自动化测试等敏捷开发工具和实践的使用。

http://www.craiglarman.com/content/feature-teams/feature-teams.htm

HUAWEI

Dev Team 全栈工程师

瀑布式：
团队成员专司一职，难以获得横向拓展

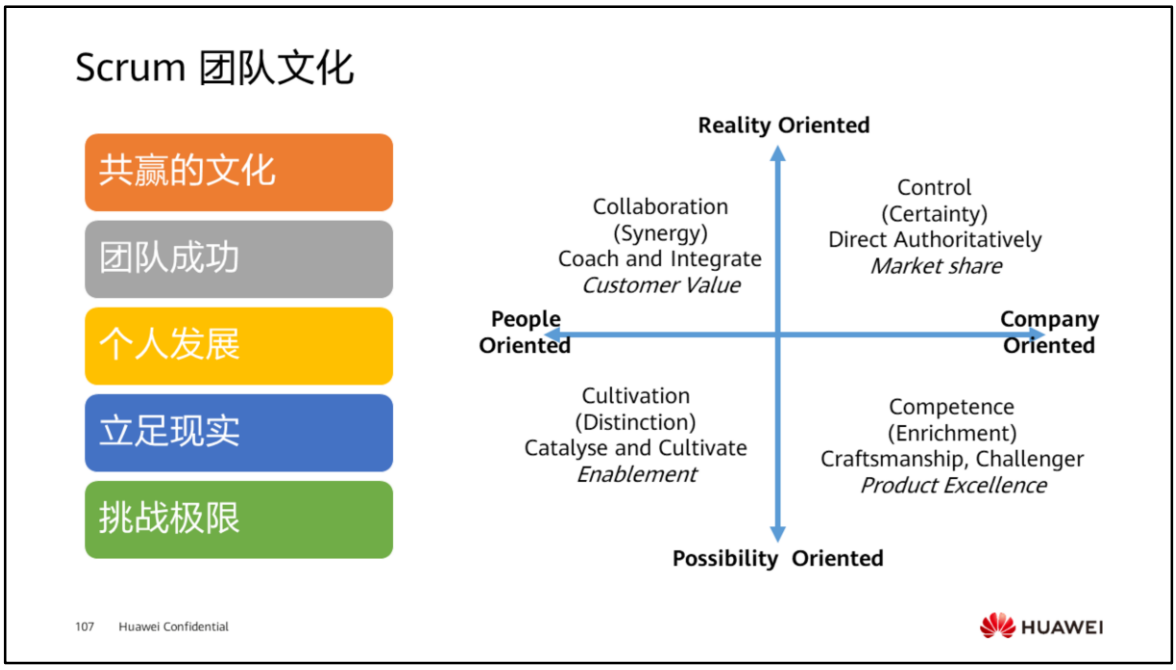


SCRUM：
人们可能主要技能不尽相同，如有人擅长开发而另外的人擅长测试，但是，在Scrum团队里，我们鼓励团队成员学习新的领域技能，尝试在新的领域工作，团队成员跨领域互相帮助完成一项产品特性。比如，一个“架构师”可能要写自动化测试代码，而一个“测试人员”则可能要分析业务。



成就全栈工程师

HUAWEI



- scrum团队的文化主要体现在五个方面分别是共赢的文化团队成功个人发展立足现实以及挑战极限这也和敏捷企业的文化思想线比较相似在这种文化中既包含横轴左侧所代表的就是个人的能力方向个人方向又包括协作和培养也包括横轴右侧的非个人方向包括控制以及能力 同时也有上半部分的现实方向它包括协作和控制又包括下半部分可能性的方向它包含培养和能力它是一个彼此对立又相互依存的多维文化。

Scrum Master 职责

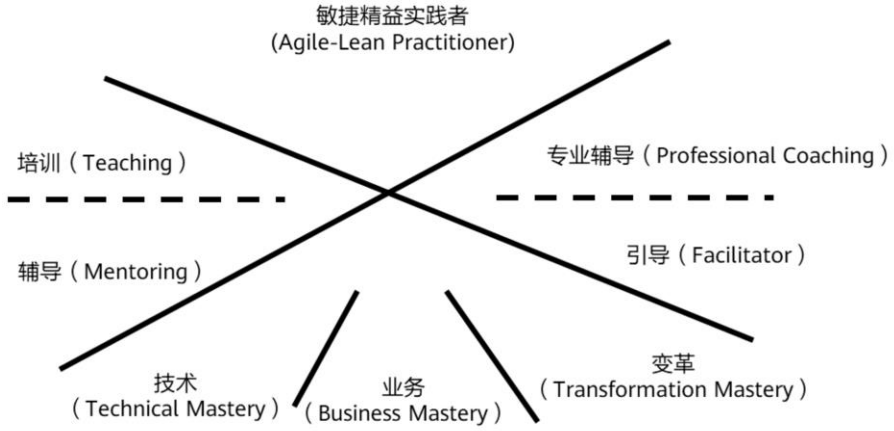


HUAWEI

- 并非传统的团队或项目经理，没有管理头衔和权力。
- 是一个具备影响力的仆人式Leader（非Manager）。
- 面向管理层时代表团队；面向团队代表管理层。
- 领导团队完成Scrum的实践以及体现其价值，不代表团队做出决定。
- 排除团队遇到的困难 。
- 确保团队的胜任其工作，并保持高效的生产率。
- 使得团队紧密合作，使得团队个人具有多方面职能的工作能力。
- 保护团队不受到外来无端影响，被授权的“牧羊犬”。
- 团队和组织变革的代理人。

- 听多于说，思维开放，敏捷者。
- 专注并细心，团队合作，善于解决问题。
- 灵活运用Scrum模式，准备好挑战他人并接受他人的挑战。
- 持续改进自我的愿望。
- 开放心态，积极探索，愿意分享和帮助他人。
- 经历过转型或至少了解组织政治生态，善用权力但不渴望权力。
- 中等偏上的技术和产品知识水平。
- 具备沟通能力和意愿包括影响力。
- 从性格像限看，友善或表现型偏多。

Scrum Master 核心能力



HUAWEI

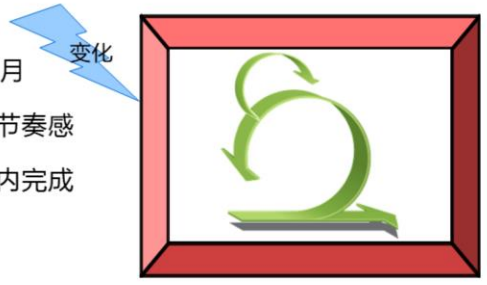
Scrum 事件

- Sprint
- Sprint 计划会议 Sprint Planning
- 每日 Scrum 站会 Daily Scrum
- Sprint 评审会议 Sprint Review
- Sprint 回顾会议 Sprint Retrospective

HUAWEI

Sprint

- Scrum 项目周期以一组迭代周期“Sprints”组成
  - 可以和极限开发的迭代周期类比
- 典型的迭代周期为 2-4 周或者最多一个自然月
- 一个固定的周期能够创造出项目的更优美的节奏感
- 产品的设计，开发，测试全部都在一个迭代内完成

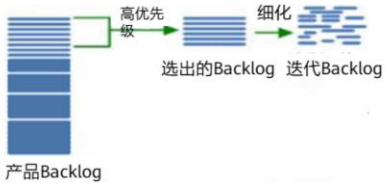


一个迭代周期的长短的设定取决于您能够保障多长时间需求变化不影响到产品开发

HUAWEI

Sprint 计划会议

- 什么是Sprint计划会议
  - 每轮迭代启动前，团队共同讨论本轮迭代详细开发计划的过程，输入是产品Backlog，输出是团队迭代Backlog
- 分成两个阶段
  - 第一个阶段：选取用户故事，确定迭代目标
    - PO与团队一起从Product Backlog中选择待完成的用户故事
  - 第二个阶段：拆分任务，创建迭代backlog
    - 团队拆分和确认任务，给出工作量估算



注：Sprint Backlog是团队协作的结果，不是只由SM和PO来决定。

HUAWEI

- Sprint计划会议的好处：
  - 通过充分讨论，使团队成员对任务和完成标准理解一致；
  - 团队共同参与，促进团队成员更认真对待自己的承诺。
- Sprint计划会议的关键要点：
  - 充分参与：Scrum Master确保PO和Team充分参与讨论，达成理解一致。
  - 相互承诺：Team承诺完成迭代Backlog中的需求并达到”完成标准“，PO承诺在短迭代周期不增加需求（2-4周）。
  - 确定内部任务：Team和PO协商把一些内部任务放入迭代中（例如重构、持续集成环境搭建等），由PO考虑并与其他外部需求一起排序 。

每日 Scrum 站会

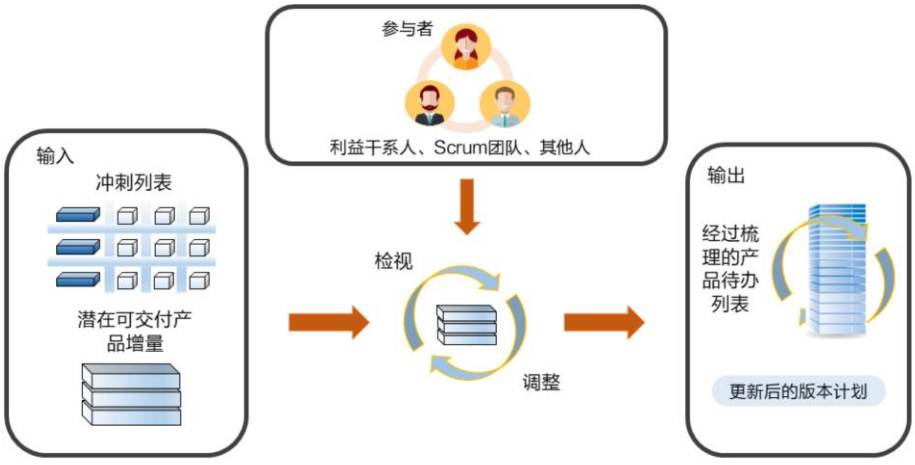
- 执行
  - 每天都会开
  - 15分钟结束
  - 站着开会
- 不是为了解决问题
  - 所有相关的人被邀请
  - 只有 Scrum master，团队成员能够在会上发言
- 避免无关的讨论
- 更新Sprint Backlog，包括增减任务项、更新任务进度和状态



HUAWEI

- 每日站立会议的好处：
  - 增加团队凝聚力，产生积极的工作氛围。
  - 及时暴露风险和问题。
  - 促进团队内成员的沟通和协调。
- 每日站立会议的关键要点：
  - 准时开始：按计划会议制定的时间地点开会，形成团队成员的自然习惯。
  - 高效会议：会议限时15分钟，每个人都保持站立，依次发言，不讨论与会议三个主题无关的事情（如技术解决方案等）。
  - 问题跟踪：Scrum Master应该记录下所有的问题并跟踪解决。
  - 每日站立会议促进团队沟通协调，及时暴露问题。

Sprint 评审会议 (1)



HUAWEI

Sprint 评审会议 (2)

- 什么是 Sprint 验收
  - 每次迭代开发结束时举行，通过演示可工作的软件检查需求是否满足客户要求；
  - 由Scrum Master 组织， PO 和用户代表（外部或内部利益相关人）负责验收、Team负责演示可工作软件。
- 非正式的
  - 2 小时的提前准备
  - 不需要正式演示文档
- 整个团队都需要参加
- 邀请所有关注产品的人参加



HUAWEI

- 团队按照Backlog中的问题，逐个地介绍这次Sprint的结果，和演示新功能。
- 如果产品负责人想要改变功能，添加一个新问题到产品Backlog中。
- 如果对功能有一个新的想法，添加一个新问题到产品Backlog中。
- 如果小组报告项目遇到阻碍现在还没能解决，把该障碍加入到障碍Backlog。
- 团队达成对这次Sprint的结果和整个产品的开发状态的共识。

展示“真实”的产品：Team 应在真实环境中展示可运行的软件，判断是否达到“完成”标准。

收集反馈：PO 根据验收情况及客户反馈意见，及时调整产品Backlog。

迭代验收的好处：

通过演示可工作的软件来确认项目的进度，具有真实性。

能尽早的获得用户对产品的反馈，使产品更加贴近客户需求。

Sprint 回顾会议 (1)

- 回顾会议
  - 在每个Sprint结束后，对于当前的迭代周期做一个阶段性的总结，包括好的方面和不好的方面，帮助团队在下一个迭代中扬长避短，对于一个团队的健康发展很有好处
- 主要指导原则
  - 不管我们现在发现了什么问题，我们必须懂得并坚信每个人通过他们当时所知的，他所拥有的技能和可得到的资源，在限定的环境下，都尽其所能做出了最好的成绩



—— 哪些工作做得好（应该继续保持）？

—— 哪些做得不好（应该停止）？

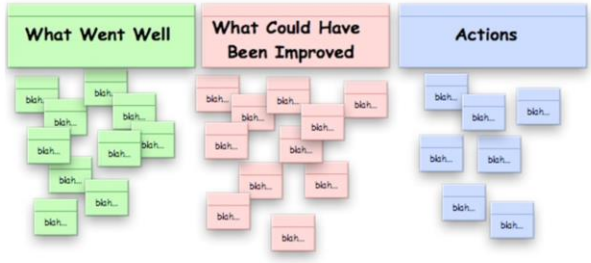
—— 哪些可以改进（就被按优先排序的改进的行动达成共识）？

HUAWEI

- 周期性的回顾，总结工作中的经验和教训。
- 一般 15–30 分钟。
- 在每个迭代结束时进行。
- 整个团队都需要参加。
  - Scrum Master
  - Product Owner
  - Dev Team
  - 可能还包括客户

Sprint 回顾会议 (2)

- 一般可以从以下方面来进行回顾：
  - 开发团队效率如何
  - 开发团队合作如何
  - 项目进展曲线是否平稳
  - 开发团队前端和后端的分工如何
  - 测试团队的缺陷报告率如何
  - 开发周期中有没有被严重Block的因素
  - 有没有需求方面的不明确导致Rework
  - 在任务分配方面有没有不均衡，导致个别人太忙或者太闲



迭代回顾会议是促进团队持续改进的最有效手段

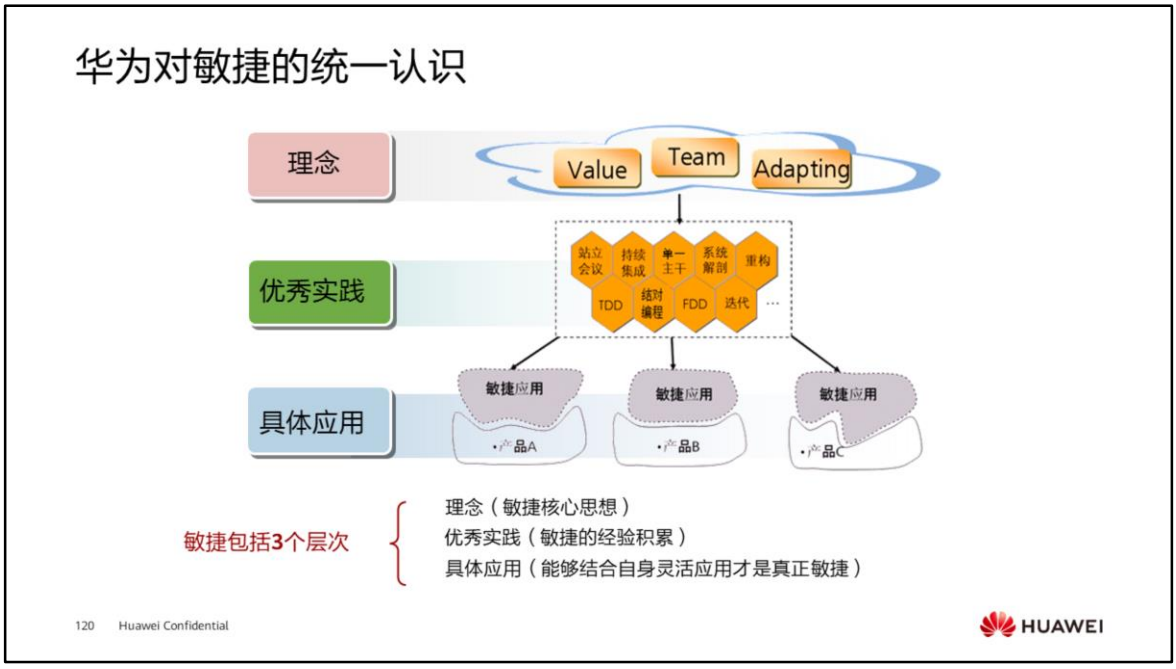
HUAWEI

- 迭代回顾会议的关键要点。
- 会议气氛：Team全员参加，气氛宽松自由，畅所欲言，头脑风暴发现问题，共同分析根因。
- 关注重点：Team共同讨论优先级，将精力放在最需要的地方（关注几个改进就够了）。
- 会议结论要跟踪闭环：可以放入迭代backlog中。
- 迭代回顾会议的好处
- 激励团队成员；
- 帮助团队挖掘优秀经验并继承；
- 避免团队犯重复的错误；
- 营造团队自主改进的氛围。

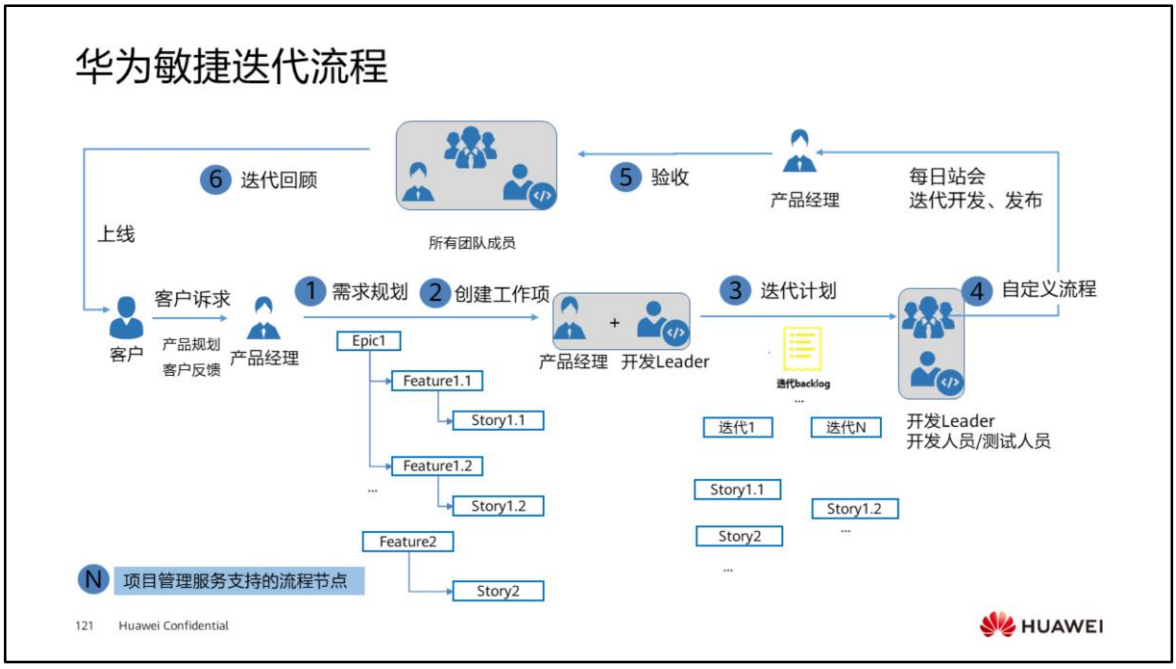
目录

1. 敏捷项目管理理念与方法实践
2. 规划与设计
3. 敏捷项目管理的方法、模型
4. 敏捷需求管理
5. 团队与协作
6. **华为敏捷项目管理企业实践**

HUAWEI



- Value：价值，为客户提供价值--产品商业成功为目标，聚焦客户价值、围绕价值流消除浪费。
- Team：团队--团队是价值真正创造者，加强团队协作、激发团队潜能；软件开发是一种团队活动，首先应做到提升沟通效率降低交流成本。
- Adapting：适应，适应变化--不断的根据经验调整，最终交付达到业务目标的产品



- 推荐的敏捷迭代流程是需求规划，迭代计划，开发（每日站会），并通过Alpha/Beta/Gama等环境发布上线，并由产品经理代为Showcase验收，并在演示后，进行迭代回顾和总结。

项目管理服务

- 推荐的典型作业流场景，以及项目管理服务对应的功能分解



HUAWEI

- 支持从创建项目到迭代回顾的端到端敏捷管理。

新建项目

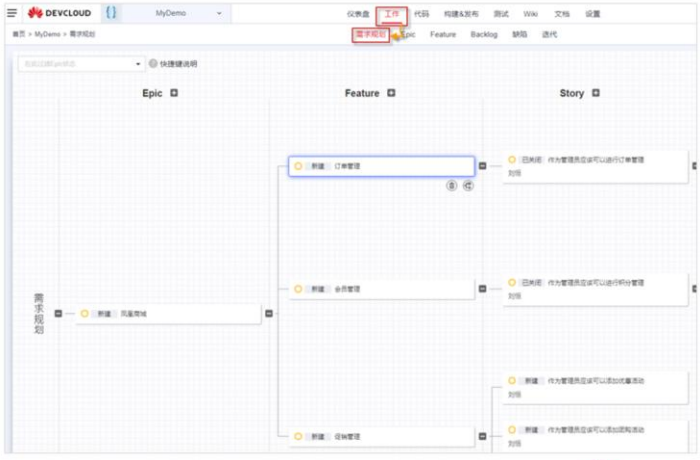
- Scrum：
  - 标准的Scrum敏捷管理流程和实践
- 看板：
  - 轻量级，事务管理
- DevOps全流程样例项目：
  - 预置工作项，代码仓，一键可发布的在线商城的样例项目



HUAWEI

需求规划

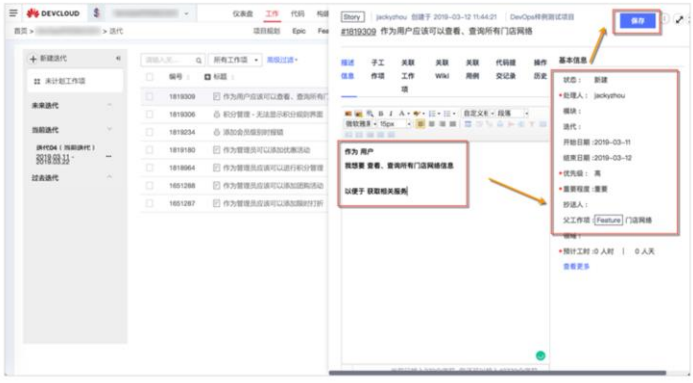
- 思维导图的形式
  - 结构化分解需求
- 图形化快速录入
- 图形化快速拖动
  - 调整需求父子关系



HUAWEI

工作项管理

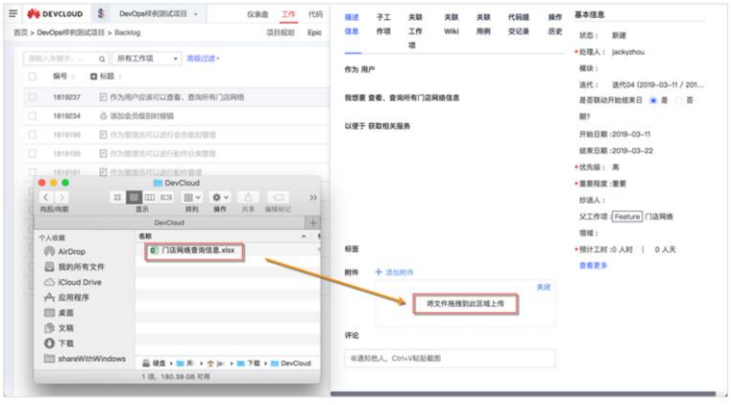
- 填写工作项描述
- 可指定模块，优先级，迭代，重要程度，抄送人等
- 可指定预计工时



HUAWEI

管理文档附件

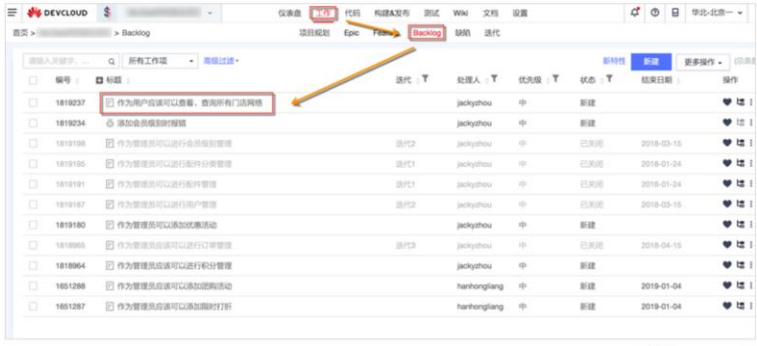
- 工作项关联文档
- 方便查看工作项的设计、分析文档



HUAWEI

Backlog 管理

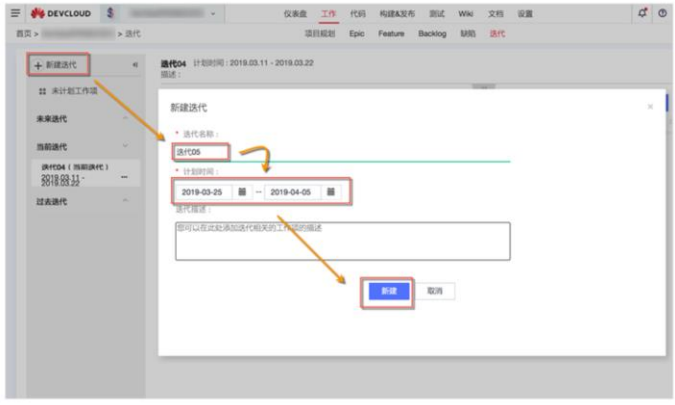
- 集中管理Story、缺陷和任务
- 表格式交互
- 快速创建子工作项



HUAWEI

新建迭代计划

- 管理所有迭代
- 为迭代指定开始结束时间
- 为迭代命名



HUAWEI

工作项指定迭代

- 可根据迭代导航，选择查看当前迭代，未来迭代和过去迭代
- 可查看不同迭代下的待交付的工作项
- 可拖动或者直接修改列表的迭代，为工作项指定迭代



HUAWEI

自定义工作项模板

- 自定义添加字段
- 可为字段指定不同的字段类型，灵活满足不同的场景



HUAWEI

自定义工作流

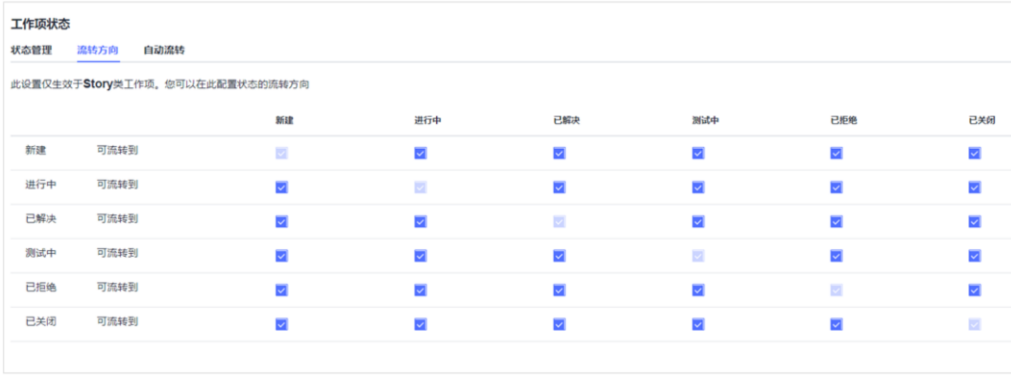
- 自定义配置不同工作项的状态
- 可自定义调整状态的前后顺序



HUAWEI

自定义流转方向

- 可自定义状态的流转方向
- 通过流转方向的控制，可以实现流程的走向控制



HUAWEI

自定义默认处理机制

- 为特定状态自定义默认的处理人
- 通过此设置，可实现指定状态默认走到特定的处理人，实现关键状态的审核集中处理
- 通过此设置，可实现特定状态必须填写评论，如拒绝Bug必须填写拒绝原因等



HUAWEI

自定义通知

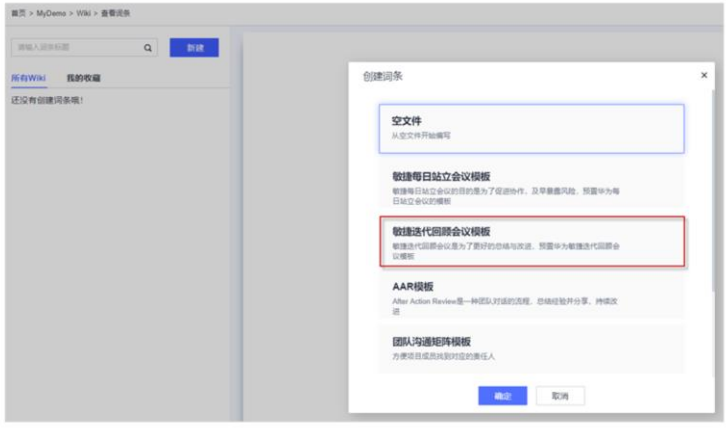
- 自定义通知，可配置特定条件下通知到特定的人员
- 支持站内信和邮件



HUAWEI

Wiki在线协作，预置迭代回顾会议模板

- Wiki轻量级的在线协作，可用于需求分析，设计文档，会议纪要，迭代回顾等
- 预置多种敏捷实践模板



HUAWEI

文档托管服务

- 云端的项目文档托管
- 支持同名文件的版本管理
- 可在线预览多种文件格式
- 支持和工作项管理，方便查看工作项的文档
- 方便迭代回顾的总结和分享



HUAWEI

迭代回顾

- 多达12种的仪表盘Widget
- 覆盖进度，质量，效率，成员工作项分布等，方便实时了解项目进展
- 也用于迭代回顾，识别下个迭代可以改进的方向



HUAWEI

思考题

1. （判断）需求排序是按照用户价值来排列的，并且与工作量相关?
2. （多选）Scrum 的框架中包含哪些工件?

A. 产品Backlog

B. Sprint Backlog

C. 产品增量

D. Sprint增量

HUAWEI

- 参考答案：1.False，2. ABC

思考题

3. （多选）项目小批量，频繁交付带来的好处有哪些?

A. 减少排队

B. 缩短交付周期

C. 增强质量

D. 加快反馈

4. （单选题）为了促进团队持续改进，使用哪种会议活动比较合适?

A. 迭代会议

B. 评审会议

C. 回顾会议

D. 每日站立会议

HUAWEI

- 参考答案：3. ABCD　4. C

本节小结

- 软件非常特殊，天生就具有复杂性，一致性，可变性，不可见性。
- 管理好软件的开发是一件非常有挑战的事情
- 敏捷是当前软件项目管理主流的理念和实践
- 在理解这些理念和实践的基础上，介绍了华为云推荐的敏捷管理作业流
- 介绍了项目管理服务支持敏捷管理的主要功能和基本操作

HUAWEI

本章总结

- 本章以华为敏捷项目管理企业实践为例，详细介绍了规划与设计中的影响地图、用户故事地图等方法，介绍了敏捷需求管理。对敏捷项目管理的理论、方法与模型进行了系统介绍。

HUAWEI

更多信息

- 华为云官网：
  - https://www.huaweicloud.com/
- 华为云软件开发平台 DevCloud：
  - https://www.huaweicloud.com/devcloud/
- 华为云帮助中心：
  - https://support.huaweicloud.com/
- 华为云学院：
  - https://edu.huaweicloud.com/

HUAWEI

Thank you.

HUAWEI

持续开发与集成



HUAWEI

前言

- 本文对持续开发和集成在软件项目开发中的应用进行了研究，提出了在应用时遇到的关键问题并给出相应的解决思路。
- 讨论了应用持续开发与集成时可以采用的辅助工具及相应的使用方法，给出了工具软件的配置方案，并说明了工具软件之间的相互关系和协作方法。
- 说明了在项目开发中应该如何应用持续集成以保持项目整体开发的高速度和高效率问题。

HUAWEI

目标

- 掌握持续开发与集成的理念及基本原则
- 理解持续开发与集成对软件开发过程管理的重要意义
- 学会如何应用持续开发与集成保持项目整体开发的高速和高效问题

HUAWEI

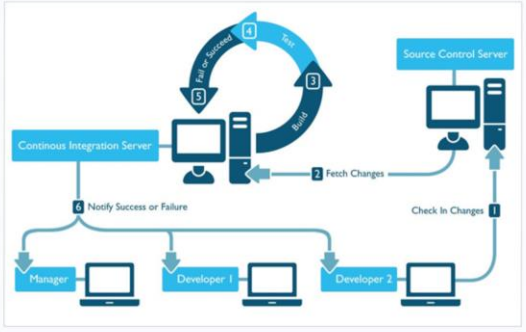
目录

1. **持续集成理念、方法与实践**
2. 代码托管与分支策略
3. Git基本概念及主要操作
4. 代码提交及代码评审
5. 华为云 DevCloud 代码托管服务及 CloudIDE
6. 静态代码检查

HUAWEI

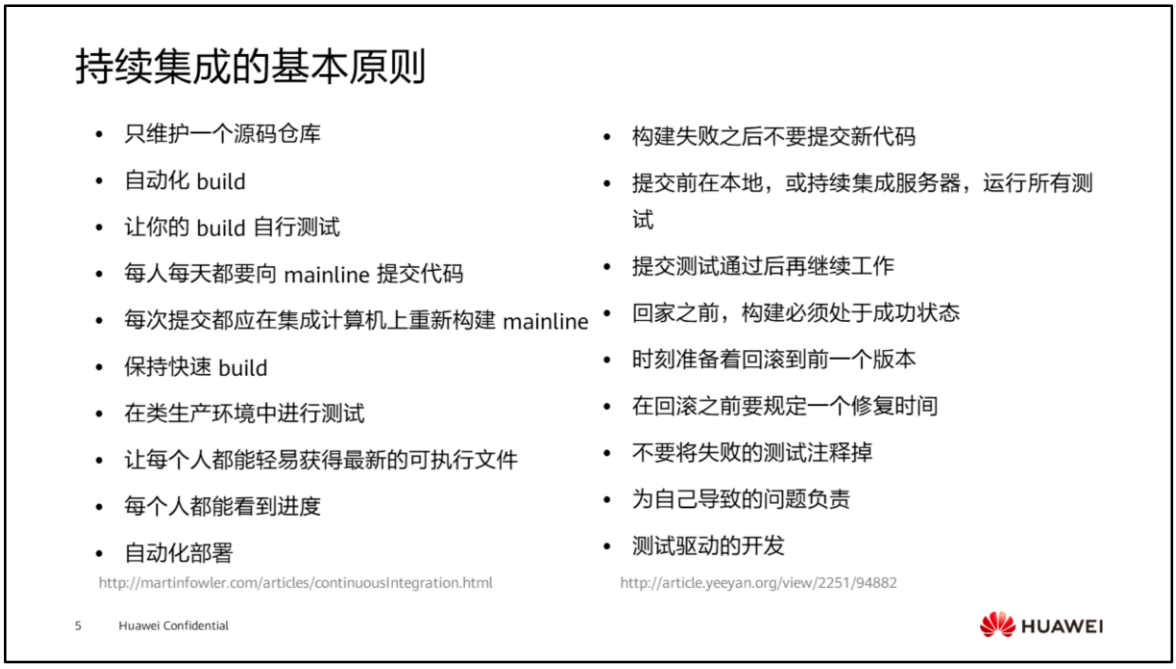
持续集成

- 持续集成是一种软件开发实践，即团队开发成员经常集成他们的工作，通常每个成员每天至少集成一次，也就意味着每天可能会发生多次集成。每次集成都通过自动化的构建（包括编译，发布，自动化测试)来验证，从而尽快地发现集成错误。（Martin Fowler）



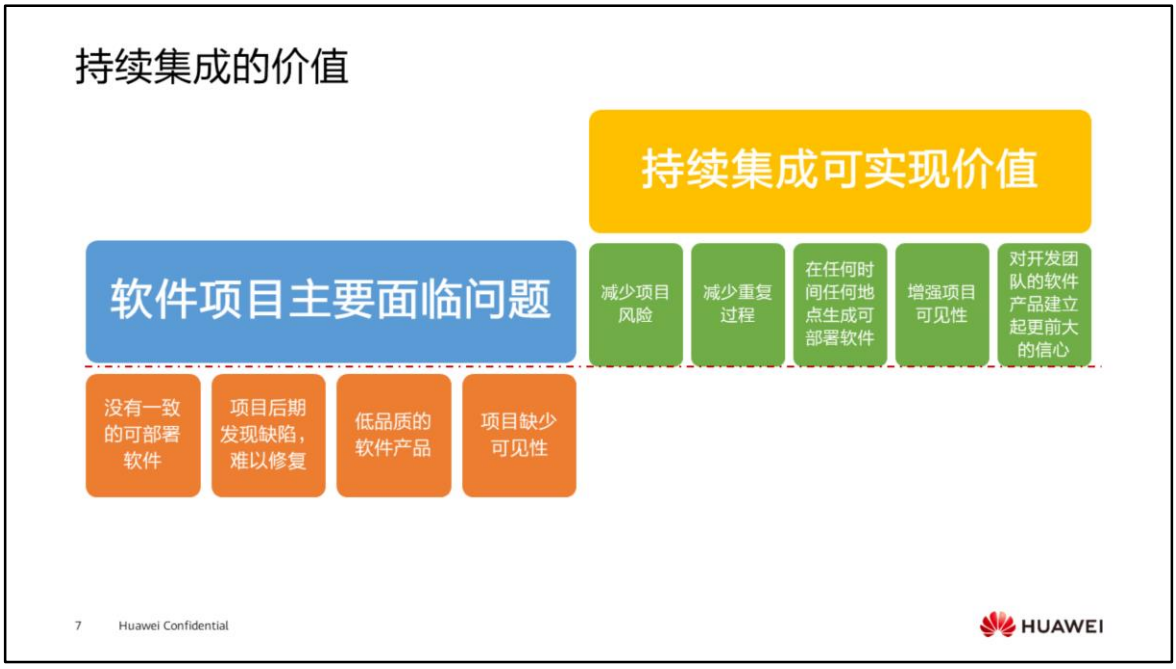
HUAWEI

- Martin Fowler对持续集成的定义四：持续集成是一种软件开发实践，即团队开发成员经常集成他们的工作，通常每个成员每天至少集成一次，也就意味着每天可能会发生多次集成。每次集成都通过自动化的构建（包括编译，发布，自动化测试)来验证，从而尽快地发现集成错误。
- 图中，每个研发工程师都会将自己的代码提交到代码服务器中，项目组中的持续集成服务器。
- 从代码服务器中拉取代码进行构建，并进行自动化测试。最后持续集成服务器会将结果通知项目组成员，如果测试失败，第一时间进行问题定位及代码修改，持续集成服务器可以手动启动或者自动化定时启动。
- 一般情况下，建议项目组在每个工作日结束之后定时启动构建，通过日编译版本把控研发质量。

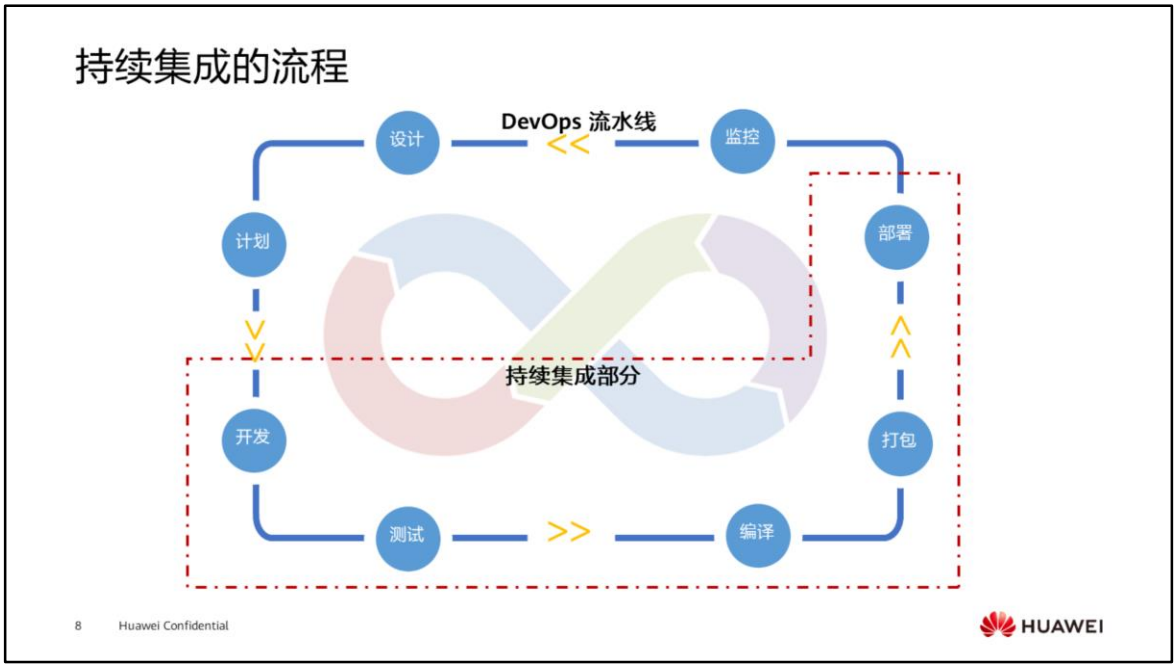


- 持续集成需要遵守一定的基本原则，左侧部分可以说是一些初级原则，右侧部分是对项目组更全面的原则要求，是完成持续构建必不可少的实践：
  - 构建失败之后不要提交新代码——持续集成的第一禁忌就是构建已经失败了，还向版本控制库中提交新代码。如果某人提交代码后构建失败了，他就是修复构建的最佳人选，应尽快找出失败的原因并修复构建。
  - 提交前在本地，或持续集成服务器，运行所有测试——不断的增量提交代码虽然是件轻量级的事儿，但确是一件严肃的事儿，我们需要在本地进行完善的自动化测试，尽量确保服务器构建的成功，不至于影响其他人的及时提交。
  - 提交测试通过后再继续工作——提交代码触发构建后，开发人员应该监视这个构建过程，直到构建成功，构建结束之前，不应该被开会或其他事儿干扰。
  - 回家之前，构建必须处于成功状态——这是我们协同研发应该具备的基本个人素质，不让自己的工作成为团队的阻塞。
  - 时刻准备着回滚到前一个版本——如果某次提交失败了，最重要的是尽快让一切再次正常运转，当无法快速修复问题，应该将它回滚到上一个版本。
  - 在回滚之前要规定一个修复时间——对于是否应该等待继续问题定位，还是回滚版本，项目组之间应该有个约定，20分钟或是更久，我们需要有个时间界定。

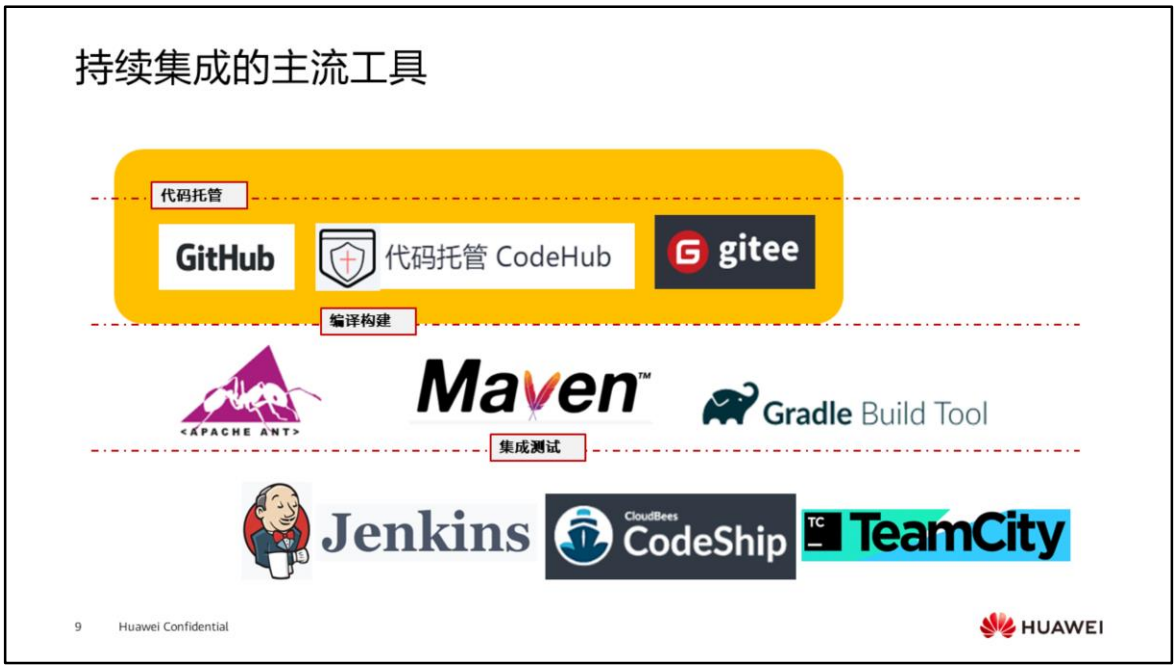
- 不要将失败的测试注释掉——导致测试失败的原因很多，为了找到真正的原因可能需要耗费很多时间，但这是值得的。
- 为自己导致的问题负责。
- 测试驱动的开发——当开发新的功能或修复缺陷时，开发人员先要写一个测试，既可以驱动应用程序的设计，也可以作为回归测试使用。



- 在采用持续集成之前，软件项目主要面临几个问题：第一：个人环境开发构建，软件包差异较大，缺少一致的可部署软件，第二：缺陷暴露较晚，在研发后期发现的问题，一般需要投入更大的人力成本去修复，这也导致了其他的问题发生，软件产品质量较差，项目缺少可见性。
- 当团队采用持续集成之后，团队能够经常合并较小的变更，他们将减少合并的复杂性与努力，将减少变更到生产环境的前置时间，可以在任何时间地点从服务器轻松的获取可以部署的软件版本进行下一步动作，提高项目的可见性。



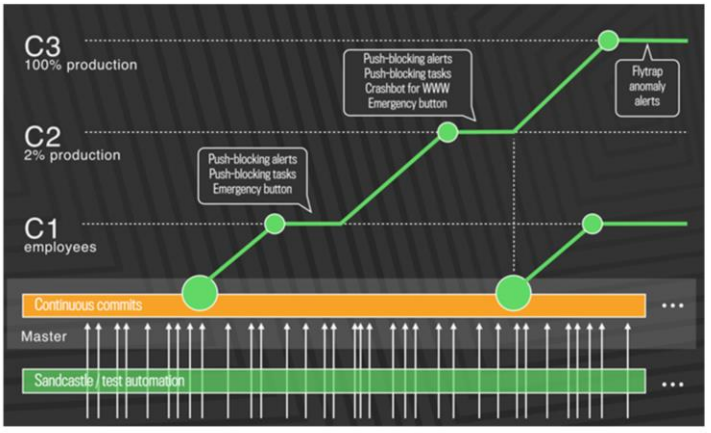
- 从本图中可以看到，持续集成涵盖了DevOps研发流程的大部分环节，是研发过程的重要实践。



- 基于Git的代码托管平台较多，有GitHub，华为云DevCloud软件开发平台中的CodeHub，码云gitee等等。
- 编译构建工具有Apache的Ant，以及后被广泛使用的Maven，由于可以自动管理java库和项目间的依赖，迅速占领了大型java项目，此外，还有Grade，MSBuild，这些工具的的核心功能，就是管理代码的依赖关系，能以正确的顺序执行一些列任务，构建出软件包。
- 持续集成工具有Jenkins，还有CodeShip，TeamCity等。
- 图片来自：
    - https://ant.apache.org/
    - https://maven.apache.org/
    - https://gradle.org/
    - https://www.jenkins.io/
    - https://gitee.com/
    - https://codeship.com/
    - https://www.jetbrains.com/teamcity/

企业实践

- 不再需要 hotfixes
- 支持全球化开发
- 用户体验更佳
- 倒推要求更高的工程能力



HUAWEI

- 早先的模式是从主干分支的提交中选择特定的变更放到发布分支上。发布分支每天将这些变更推送到生产环境。但是，逐渐遇到了问题，随着时间的推移，提交的数量和团队成员都有所增加，发布工作越来越繁重。后来，Facebook找到了自己的持续集成之路，先将变更经过一系列自动化测试后，从内部推送到 Facebook 员工，就是图中的C1，在这一阶段发现的任何问题，都会认为工作阻塞去高优先级解决。下一步涉及到“ canary deployment ”（金丝雀部署），只推送至生产环境的 2% ，是图中的C2，依靠连续的监测来发现问题。如果一切顺利，这些变更将 100% 部署到生产环境中。这个实践使Facebook获得了更好的用户体验。
- 图片来自：https://www.infoq.cn/article/2017/09/facebook-release-scale
- 更多参考：https://engineering.fb.com/developer-tools/rapid-release-at-massive-scale/

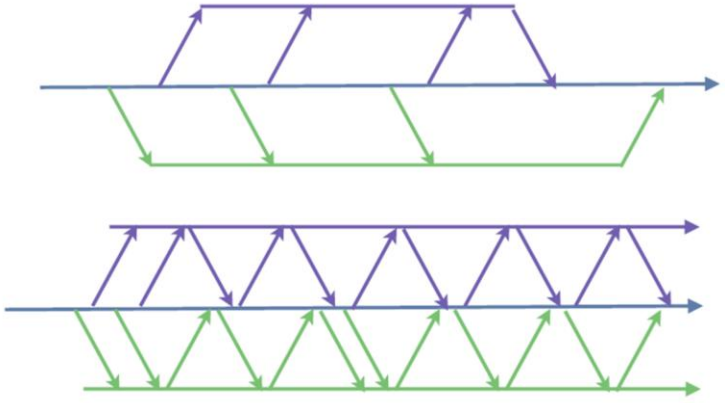
目录

1. 持续集成理念、方法与实践
2. **代码托管与分支策略**
3. Git基本概念及主要操作
4. 代码提交及代码评审
5. 华为云 DevCloud 代码托管服务及 CloudIDE
6. 静态代码检查

HUAWEI

版本管理

- 你喜欢哪一种?

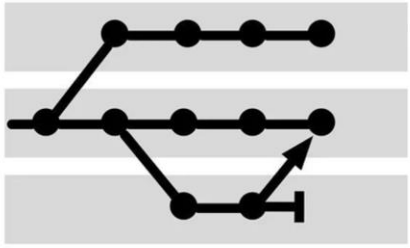


HUAWEI

- 第二幅图更符合持续集成的原则，更多的版本合入，促使提前发现问题。
- 强烈推荐DevOps团队通过“基于主干开发”（trunk-based development）方式践行持续集成，让开发者经常将代码合并到中央仓库，理想情况下是每天几次。当团队能够经常合并较小的变更，他们将减少合并的复杂性与努力。

版本规划

- 软件版本管理，不只是分支管理
  - Build Manager?
  - 配置管理员?
  - IT Support?
- 版本管理的目的
  - 版本控制，历史信息
  - 团队协作并行开发
  - 过程管理（变更、审批、流程）
  - 问题追踪



持续交付，从代码分支策略，看到的是软件的发布机制；发布的过程，是开发与运维的沟通。

HUAWEI

- 软件版本管理是什么？有人认为它就是分支管理，其实，不单单局限于这些，它广泛的涵盖了项目流程管理和团队协同开发范畴。在研发过程中，团队制定适合自己的分支策略，就是制定了相应的发布机制，是开发和运维沟通的桥梁。
- 在产品的整个生命周期中，一定会遇到多种渠道获得的各种各样的需求，包括部门内部由于运营需要而提出的需求，产品经理提出的需求，也有通过用户投诉或反馈得来的需求，以及通过研发人员对资深产品和竞品的体验分析得来的需求，如何将这些需求落地，做好版本规划非常关键。
- 软件版本管理不只是分支管理的范畴，还广泛地涵盖了敏捷项目管理和团队协同开发的范畴。对于产品经理而言，版本规划可以帮助他设计产品，制定发布的节奏，调整需求的优先级。
- 对于公司其他部门的同事以及外部的客户而言，发布计划可以让他们了解产品的进展情况，以便做好相应的安排。同时在开发项目关联需求的时候，版本规划可以做好提前的评估。对于持续交付来说，版本规划要提前做好代码的分支策略，从代码的分支策略看到的是软件的发布机制，发布的过程，使开发与运维的沟通。

工作流 (Git Flow) (1)

- 用于记录历史的分支
  - Git Flow 使用两个分支来记录项目开发的历史，而不是使用单一的 Master 分支。在 Git Flow 流程中，Master 只是用于保存官方的发布历史，而 Develop 分支才是用于集成各种功能开发的分支。



HUAWEI

- GitFlow工作流中有两个长期存在的分支，master和develop分支。
- master ：分支中包含的是可以部署到生产环境中的代码，只可以merge，不允许commit。
- develop：是从master分支中拉取的主开发分支，用来集成测试最新合入的开发成果，包含要发布到下一个release的代码。
- Git Flow工作流，Git Flow相对其他工作流来说，分支较为复杂。它非常适合管理大型项目的发布和维护。
- 与其他工作流相比，Git Flow流程，没有增加任何的新的概念或者命令。它的特色在于为不同的分支分配了非常明确的角色，并且定义了使用场景和用法。除了用于功能开发的分支，它还使用了独立的分支进行发布前的准备记录以及后期维护。
- Git Flow使用一个中央代码仓库，它是所有开发者的信息交流中心。这跟其他的工作流程是一样的。开发者在本地完成开发，然后再将分支代码推送到中央仓库。不同的工作流仅仅是分支定义和使用场景的不同。
- Git Flow工作流有两个长期存在的分支，Master分支和Develop分支，Git flow使用两个分支，来记录项目的开发历史，Master分支，包含了可以部署到生产环境的代码，一般只可以从其它的分支合并、变更，不允许直接提交。
- 在很多代码托管平台中，Master是要被设置成分支保护模式的，因为只有这样才能保证它的代码的稳定性，Develop分支从master分支拉取代码，用来集成测试，最新合入的开发成果，是研发团队新功能集合验证的分支，它包含了所有要发布到下一个版本的功能代码。

工作流 (Git Flow) (2)

- 用于功能开发的分支
  - 每一个新功能的开发都各自使用独立的分支。在创建新的功能开发分支时，父分支选择 Develop（而不是 Master）。当功能开发完成时，改动的代码被合并（merge）到 Develop 分支。

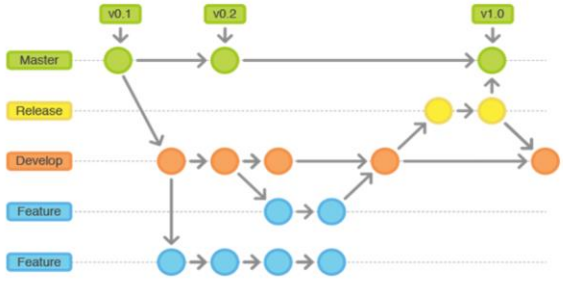


HUAWEI

- Git Flow工作流在开发新功能的开始，都会基于Develop分支创建一个独立的Feature分支。
- 开发人员在分支中完成开发及单元测试，最后将代码合入到Develop分支。在很多企业实践中，Feature向Develop分支合并的时候，会提交一个合并请求，有专门的技术负责人进行审核，把控代码质量。只有审核通过的代码才允许被合入到Develop分支，Feature分支它是一个临时分支，它的生命周期随着功能开发完成而结束。

工作流 (Git Flow) (3)

- 用于发布的分支
  - Release 分支基于 Develop 分支创建，我们可以在这个 Release 分支上测试，修改 bug 等。这个分支的创建意味着一个发布周期的开始，也意味着本次发布不会再增加新的功能。在一切准备就绪的时候，合并该分支到 Master 和 Develop，并且用版本号打上标签。



HUAWEI

- Git Flow工作流在进行发布时，会基于Develop分支创建一个Release分支。
- 分支的创建意味着一个发布周期的开始，任何新功能不可以再向Release分支合并，Release分支只接受bug修改代码的提交，Release分支它也是一个临时分支，当发布代码合并到Master分支时，打上标签，Release分支的生命周期就结束了。

工作流 (Git Flow) (4)

- 用于维护的分支
  - 发布后的维护工作或者紧急问题的快速修复也需要使用一个独立的分支。这是唯一一种可以直接基于 Master 创建的分支。一旦问题被修复了，所做的改动应该被合并入 Master 和 Develop 分支。在这之后，Master 上还要使用更新的版本号打好标签。

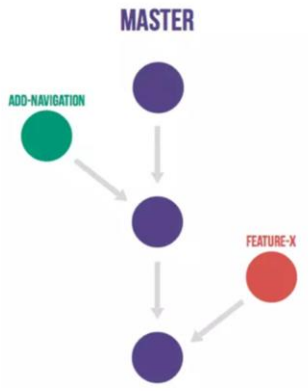


HUAWEI

- hotfix：补丁分支，生产环境发现新Bug时创建的临时分支，问题验证后，合并到Master和Develop分支，所以Hotfix的改动会进入下一个Release 。
- 当版本发布后，现网发现紧急问题时，快速修复也需要使用一个独立的分支，也就是图中的Hotfix分支，这是一个补丁分支，它也是唯一一个可以直接从Master创建的分支。当问题验证后，合并到Master和Develop分支，所以Hotfix的变更也会进入到下一个Release。

工作流 (Github Flow)

- Github Flow 工作流仅具有特性分支和主干分支，非常简单和干净。
- 将所有内容合并到主干分支和部署，最大限度地减少“库存”中的代码量。

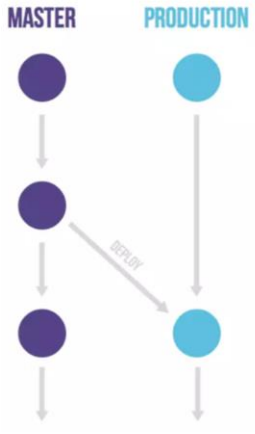


HUAWEI

- 轻量级的Github工作流，Github工作流仅具有特性分支和主干分支，非常简单和干净。
- 将所有内容合并到主干、分支和部署，最大限度地减少了库存中的代码量。它有一个基本原则，任何主干上的版本都是可发布的。
- 开发人员创建的新分支，无论是新增特性，还是修改bug，分支命名应该具有较好的描述性，这样其他人就可以看出分支的意图。
- 特性分支创建成功后，无论什么操作，新增、修改或是删除文件，都可以及时地提交到特性分支。这样的提交过程，便于跟踪开发速度。
- Github工作流的特点之一就是pull request，它可以为提交初始化一个讨论组，在开发过程中的任何时候都可以创建一个pull request，甚至没有写一行代码的时候，只是想寻求其他人的帮助，或者是功能开发完成，准备让其他人来检查代码。
- 一旦pull request被审核通过，新功能就进入了产品中等待验证。但是，Github工作流隐含一个问题，每次合并新功能，主分支的代码是应该及时发布的，但实际中常常不能做到这一点。

工作流 (GitLab Flow) (1)

- 带生产分支工作流
  - 创建一个反映已部署代码的 PRODUCTION 分支
  - 将 MASTER 合并到 PRODUCTION 分支来部署新版本
- 优点
  - 可以随时检出 PRODUCTION 分支，查看生产代码
  - 可以清晰查看提交合并记录
  - 可避免 Git Flow 中常见的发布，打标签和合并的开销

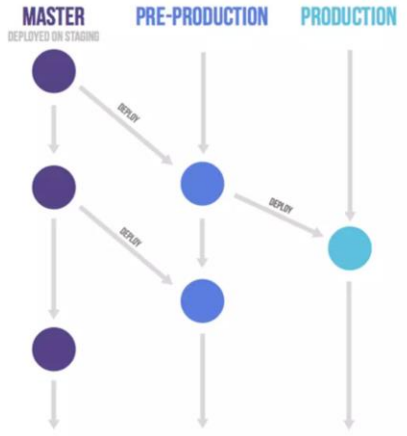


HUAWEI

- Gitlab Flow 的最大原则叫做"上游优先"（upsteam first），即只存在一个主分支master，它是所有其他分支的"上游"。只有上游分支采纳的代码变化，才能应用到其他分支。在这个前提下，我们创建一个反应已部署代码的production，master合并到production分支来部署新版本。

工作流 (GitLab Flow) (2)

- 带环境分支工作流
  - MASTER 分支部署测试环境
  - MASTER 分支向 PRE-PRODUCTION 分支合并，部署预生产环境
  - PRE-PRODUCTION 分支向 PRODUCTION 分支合并，部署生产环境
- 优点
  - 这种变更提交(commits)只向下游流动的工作流可确保所有变更在所有环境中都经过测试

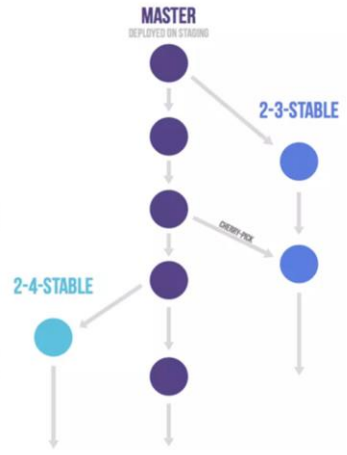


HUAWEI

- 可以随时检出production分支，查看生产代码，也可以清晰的查看提交合并记录，还可以避免Gitlab Flow中常见的发布打标签和合并引起的开销。
- Gitlab Flow创建了不同环境的多个分支，master分支部署测试环境，master分支向pre-production分支合并、部署预生产环境。pre-production分支向production分支合并，部署生产环境，代码的变更必须经由上游向下游发展。
- 当要部署预生产环境时，它会创建从测试分支到预生产分支的合并请求，而将预生产分支合并到生产分支则会触发代码的直接上线。这种变更的提交，优点是只向下游流动的工作流，可确保所有的变更在所有环境中都经过测试, 但是，在某些极端的情况下，可能需要执行更多的手工测试，这就导致无法执行上述的从测试环境到预生产环境，到生产环境的标准流程，那么我们可以直接从特性分支cherry pick一个合并请求，到其他的下游分支。

工作流 (GitLab Flow) (3)

- 带发布分支工作流
  - 只有一个 MASTER 分支
  - 每次需要发布时拉取发布分支，并对发布分支打标签
  - 发布分支以 MASTER 分支为起点，尽可能晚的创建
  - 在一个发布分支被公布后，仅有严重错误的修复能被包括在该分支中
- 优点
  - 分支数少，易于管理。发布分支明确，从项目管理维度可以明确的判断 bugfix 需要合入分支的策略



HUAWEI

- 关于“版本发布”的流程，建议的做法是每一个稳定版本，都要从master分支拉出一个分支，比如2-3-stable、2-4-stable等等。
- 并且此时要更新小版本号Gitlab Flow工作流。旨在向外界发布软件时才使用发布分支。此stable分支以master分支作为起点，并尽可能晚的创建。因为尽可能晚地创建分支，可以最大限度地减少bugfix，应用于多个分支所花费的时间。
- 在一个发布分支上线之后，仅有严重问题的代码修复才能被提交到该分支中。如果可能的话，这些bugfix、应首先合并到master中，然后再去cherry pick到发布分支，这样做是为了让我们不会忘记，将他们从发布分支反向cherry pick到master中，以避免在后续版本中遇到相同的bug。
- bug修复也可以独立地创建分支。建议以fix前缀命名，作为master的上游分支，它属于临时分支，bug解决后，可以删除该分支这些Git Flow工作流。
- 上游优先的实践，也是很多企业像谷歌、红帽实施的策略，每次在发布分支中，包含错误修复时，都会通过打新标签来设置补丁的版本，会向前更新一个小的版本号。GitLab Flow使用发布分支。这样做的优点是分支数少，易于管理。发布分支明确，从项目管理维度可以明确的判断，bugfix需要合入分支的策略，发布分支非常适用于APP、小程序等，有版本规划的项目。

企业实践

- **工作流程**
  - 开发者先创建文件的本地拷贝，这叫做"工作区"（workspace）。完成开发后，工作区的快照共享给其他开发者进行代码评审。只有通过了评审，代码才能合并到中央仓库。

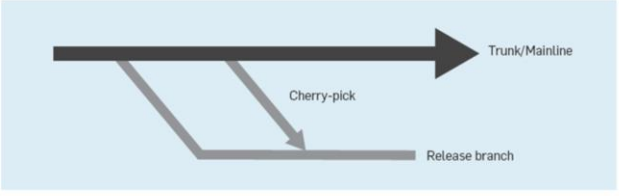


HUAWEI

- 由于不采用"分支开发"，谷歌引入新功能，一般在代码中使用开关控制。这避免了另起一个分支，也使得通过配置切换功能变得容易，一旦新功能发生故障，很容易切换回旧功能。等到新功能稳定，再彻底删除旧代码。

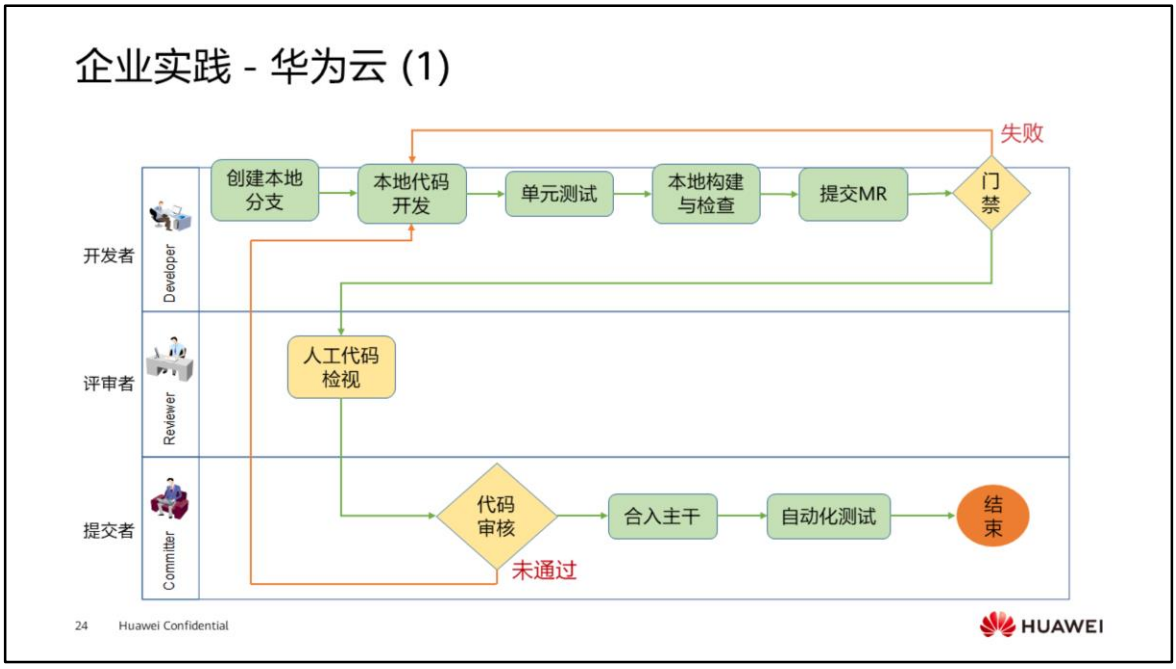
企业实践

- Google 采用"主干开发"(trunk-based development)。代码一般提交到主干的头部。这样保证了所有用户看到的都是同一份代码的最新版本。
- "主干开发"避免了合并分支时的麻烦。谷歌一般不采用分支开发，分支只用来发布。大多数时候，发布分支是主干某个时点的快照。以后的除错和功能增强，都是提交到主干，必要时 cherry-pick 到发布分支。



HUAWEI

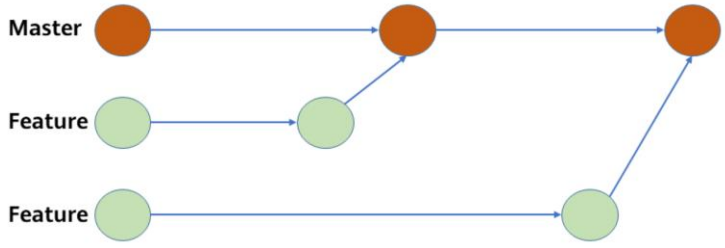
- 由于不采用"分支开发"，谷歌引入新功能，一般在代码中使用开关控制。这避免了另起一个分支，也使得通过配置切换功能变得容易，一旦新功能发生故障，很容易切换回旧功能。等到新功能稳定，再彻底删除旧代码。



- 华为云实践如图所示，三种不同角色的项目成员分工协作，保证最终合入主干分支的代码质量。Developer在完成代码开发和自验证后，发起代码入库申请，分别选择Reviewer和Committer。
- IT工具触发门禁检查，门禁检查成功后，通知Reviewer检视。
- Reviewer收到代码检视通知后，对代码进行检视并提出代码检视意见。
- Developer收到检视意见后，完善代码并闭环检视意见，更新代码入库申请。

企业实践 - 华为云 (2)

- 分支规则：Feature分支向Master分支合入
  - 从 Master 分支拉取 Feature分支
  - 根据版本节奏，Feature分支向 Master 合入发布版本代码



HUAWEI

目录

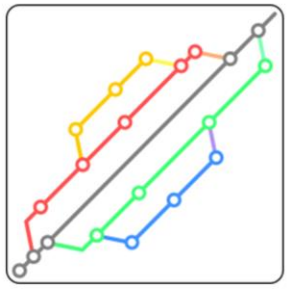
1. 持续集成理念、方法与实践
2. 代码托管与分支策略
3. **Git 基本概念及主要操作**
4. 代码提交及代码评审
5. 华为云 DevCloud 代码托管服务及 CloudIDE
6. 静态代码检查

HUAWEI

Git 是什么

- Git 是目前世界上最先进的分布式版本控制系统
- Git 近乎所有操作都是本地执行



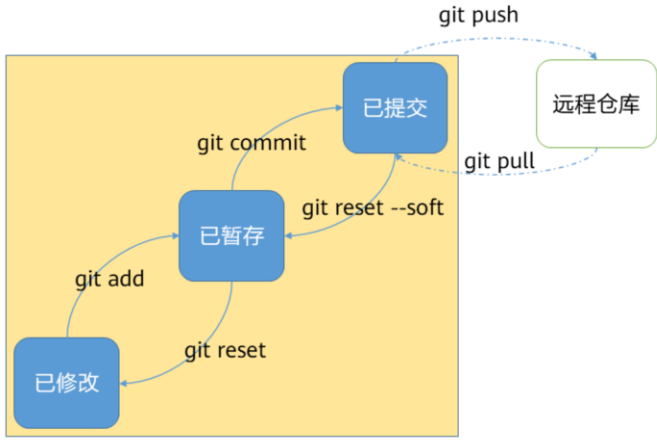


HUAWEI

- Git诞生于2002年，由Linux之父和他的团队开发并不断完善，它秉承了Linux的开源精神，为广大研发团队带来了非常好的版本控制体验。
- Git是分布式版本管理控制系统，相对于集中式它有很多优点。版本库本地化，本地仓库是服务器仓库的完整克隆，包括标签分支、版本记录等等。
- 支持离线提交，适合跨地域协同开发。分支切换快速高效、创建和销毁分支简单容易。由于Git在本地磁盘上就有项目的完整历史，所以在Git中绝大多数操作只需要访问本地文件和资源，不需要实时的与代码服务器连接。
- 举个例子，如果想查看当前版本与一个月前的版本之间引入的修改，Git会查找到一个月前的文件，做一次本地的差异计算,而不是从远程服务器拉回旧版本，在本地处理。这也意味着开发者离线或者没有VPN时，几乎可以进行任何操作。如果开发者在飞机或火车上想做些工作，可以先提交到本地仓库，直到有网络连接时再上传。
- 图标来源：https://git-scm.com/downloads/logos

Git 文件状态

- 文件的三种状态
  - 已修改
    - 修改后的 git 仓库中的文件
  - 已暂存
    - git add ‘已修改’文件
  - 已提交
    - git commit ‘已暂存’文件



HUAWEI

- 一次代码修改被成功的提交到远端仓库会经历4个阶段。本地工作区、暂存区、本地版本库、远端版本库，通过执行相应的git命令，文件在这4个区域跳转，并呈现不同的状态：
  - 已修改状态：包括三种文件，新增文件、被修改的文件或被删除的文件；
  - 已暂存状态：对修改的文件执行git add。或git remote操作。文件就变成了以暂存状态进入了暂存区；
  - 已提交状态：对于暂存的文件进行git commit操作，文件就变成了已提交状态，进入本地的版本仓库，如果想把本地的代码同步到远端仓库，使用git push，命令就可以同步了。

Git配置 (1)

- 查看 Git 版本
  - git --version
- 配置用户名和 Email
  - git config --global user.name youname
  - git config --global user.email name@mail.com
- 查看配置信息
  - git config -l

```
q00397277@DESKTOP-IAK9QLS MINGW64 ~/Desktop
$ git --version
git version 2.6.1.windows.1
```

```
q00397277@DESKTOP-IAK9QLS MINGW64 ~/Desktop
$ git config --global user.name "q00397277"

q00397277@DESKTOP-IAK9QLS MINGW64 ~/Desktop
$ git config --global user.email "quchen3@huawei.com"

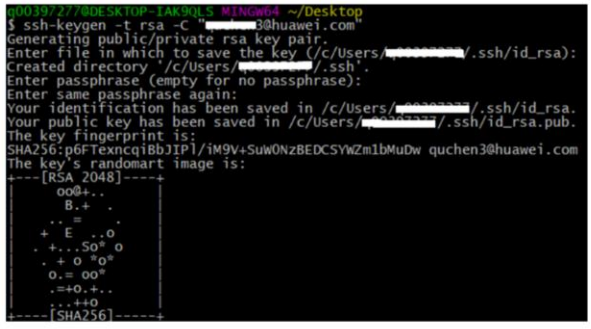
q00397277@DESKTOP-IAK9QLS MINGW64 ~/Desktop
$ git config -l
core.symlinks=false
core.autocrlf=false
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
http.sslcainfo=D:/Git/mingw64/ssl/certs/ca-bundle.crt
diff.astextplain.textconv=astextplain
rebase.autosquash=true
user.name=q00397277
user.email=quchen3@huawei.com
```


HUAWEI

- 使用git需要安装git客户端，本文安装的是git bash，开发者也可以根据自己的使用习惯，选择其他的客户端工具，比如TortoiseGit、source tree等等。
- 打开git bash终端界面，通过git version命令查看安装的工具版本，使用git config命令，配置相关的用户名、邮箱等等，配置成功后，可以通过git config list命令进行查看，也可以在用户的家目录下.git config的文件中查看相应的信息。

Git配置 (2)

- HTTPS 配置
  - git config --global credential.helper “cache”
- SSH 配置
  - 生成ssh-key `ssh-keygen`
  - 将public key添加到 remote 仓库
  - 添加 ssh 配置（~/.ssh/config）
    - 格式
    - 权限 600



HUAWEI

- 在本地与Git远端服务器传输中，包括两种通信方式，HTTPS和SSH：
  - HTTP采用用户名，密码授信，但是这种方式需要每次操作都重新输入，为了减少重复操作，通过git config的credential.helper将用户名，密码存储起来。
  - 第二种通信方式是SSH，根据ssh-keygen命令，会在本地产生一对秘钥，公钥是以.pub为后缀的文件，把公钥中的内容粘贴到服务器中，就可以使用SSH进行通信了。

配置忽略文件

- .gitignore 文件的位置
  - 可以配置在代码仓库的任意目录，但是下面规则是以 .gitignore 所在目录为根目录

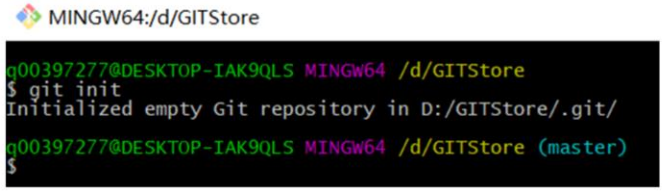
规则	作用
/mtk/	过滤整个文件夹
*.zip	过滤所有 .zip 文件
/mtk/do.c	过滤某个具体文件
!/mtk/one/txt	追踪（不过滤）某个具体文件

HUAWEI

- 在代码仓库中，可能有一些文件不需要进行版本管理，比如构建过程中产生的中间文件，或者其他的压缩文件，通过设置.gitignore中的规则，将这些文件排除在版本管理之外。

初始化仓库

- 创建第一个代码库
  - git init
- 从远端克隆仓库到本地
  - git clone ssh 或者 https





HUAWEI

- 如果本地有一个没有进行版本控制的代码仓库，只要进入到代码根目录下执行git init，就可以对整个文件夹进行管理了。
- 如果本地没有代码，想从代码服务器中克隆一份代码，只要知道仓库的ssh或者https的URL地址，执行git clone 加URL，就可以在本地拉取一份和服务器相同的代码了。

远程仓库管理

- 查看所有的远程仓库
  - git remote –v
- 添加远程仓库
  - git remote add remote-name remote_url
- 修改远程仓库命名
  - git remote rename old-remote-name new-remote-name
- 删除远程仓库
  - git remote remove remote-name

```
g00397277@DESKTOP-IAK9QLS MINGW64 /d/new-project (master)
$ git remote -v
origin  git@codehub-devcloud.rnd.huawei.com:zxcsxm00001/new-project.git (fetch)
origin  git@codehub-devcloud.rnd.huawei.com:zxcsxm00001/new-project.git (push)

g00397277@DESKTOP-IAK9QLS MINGW64 /d/new-project (master)
$ git remote add project-v1 project-v1.git

g00397277@DESKTOP-IAK9QLS MINGW64 /d/new-project (master)
$ git remote -v
origin  git@codehub-devcloud.rnd.huawei.com:zxcsxm00001/new-project.git (fetch)
origin  git@codehub-devcloud.rnd.huawei.com:zxcsxm00001/new-project.git (push)
project-v1      project-v1.git (fetch)
project-v1      project-v1.git (push)
```

```
g00397277@DESKTOP-IAK9QLS MINGW64 /d/new-project (master)
$ git remote rename project-v1 project-v2

g00397277@DESKTOP-IAK9QLS MINGW64 /d/new-project (master)
$ git remote -v
origin  git@codehub-devcloud.rnd.huawei.com:zxcsxm00001/new-project.git (fetch)
origin  git@codehub-devcloud.rnd.huawei.com:zxcsxm00001/new-project.git (push)
project-v2      project-v1.git (fetch)
project-v2      project-v1.git (push)
```

```
g00397277@DESKTOP-IAK9QLS MINGW64 /d/new-project (master)
$ git remote remove project-v2

g00397277@DESKTOP-IAK9QLS MINGW64 /d/new-project (master)
$ git remote -v
origin  git@codehub-devcloud.rnd.huawei.com:zxcsxm00001/new-project.git (fetch)
origin  git@codehub-devcloud.rnd.huawei.com:zxcsxm00001/new-project.git (push)
```


HUAWEI

- 通过git remote 建立远程仓库的链接，要关联一个远程库，使用命令git remote add origin git@server-name:path/repo-name.git；关联后，使用命令git push就可以将本地仓库的代码推送到这个关联的远程仓库。

分支管理

- 查看所有分支
  - git branch
  - -v 查看分支的最新的提交
  - -a 查看所有分支包括远程分支切换分支
  - git checkout branch_name
- 创建新的分支
  - git checkout –b new_branch　创建一个分支与当前分支相同的分支
  - git branch new_branch --track remote-name/remote-branch 创建一个分支
- 删除分支
  - git branch -d branch_name

```
q00397277@DESKTOP-IAK9QLS MINGW64 /d/test-project (master)
$ git branch
* master

q00397277@DESKTOP-IAK9QLS MINGW64 /d/test-project (master)
$ git branch -v
* master 3be19f9 merge "develop" into "master" 搞一个合并
```

```
q00397277@DESKTOP-IAK9QLS MINGW64 /d/test-project (master)
$ git checkout nenew_branch
Switched to branch 'nenew_branch'

q00397277@DESKTOP-IAK9QLS MINGW64 /d/test-project (nenew_branch)
$
```

```
q00397277@DESKTOP-IAK9QLS MINGW64 /d/test-project (master)
$ git checkout -b new_branch
Switched to a new branch 'nenew_branch'

q00397277@DESKTOP-IAK9QLS MINGW64 /d/test-project (nenew_branch)
$ git branch
  master
* nenew_branch
```

```
q00397277@DESKTOP-IAK9QLS MINGW64 /d/test-project (master)
$ git branch -d nenew_branch
Deleted branch nenew_branch (was 3be19f9).

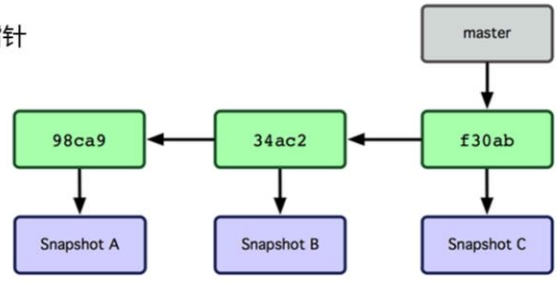
q00397277@DESKTOP-IAK9QLS MINGW64 /d/test-project (master)
$ git branch
* master
```


HUAWEI

- -a 查看所有分支包括远程分支（注意如果要查看所有远程分支，则需要 git fetch remote-name 获取 remote 仓库的信息）。

Git 分支 (指针1)

- 分支
  - 本质上是个指向commit对象的可变指针
  - 分支的默认名字为master
  - 每次提交的时候都会自动向前移动
  - 作用：从某个提交对象往回看的历史

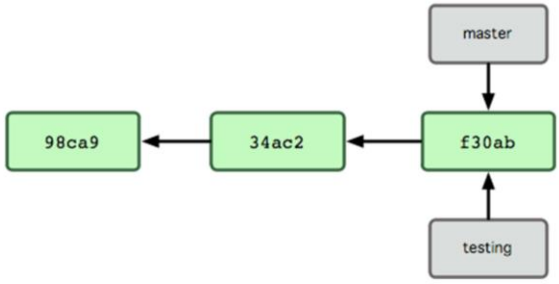


HUAWEI

- Git 中的分支，其实本质上仅仅是个指向 commit 对象的可移动指针。Git 会使用 master 作为分支的默认名字。
- 每次提交的时候都会自动向前移动，在若干次提交后，就有了一个指向最后一次提交对象的 master 分支。
- 从最后一次commit往回看，也可以看到Snapshot B ，SnapshotA的历史提交轨迹。

Git 分支 (指针2)

- 新建分支
  - 示例：新建一个testing分支
  - git branch testing

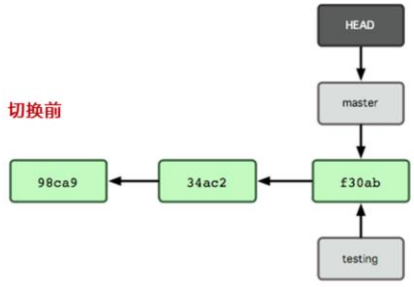


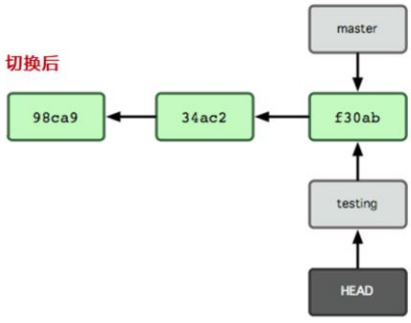
HUAWEI

- 分支的本质就是指针，那么新建一个分支，就等同于新建一个指针。图中，在master最新提交的基础上，新拉取一个testing分支，只需执行git branch testing命令就可以创建一个新分支。

Git 分支 (指针3)

- 如何识别当前分支
  - HEAD指针
  - 示例：切换到当前分支
  - git checkout testing



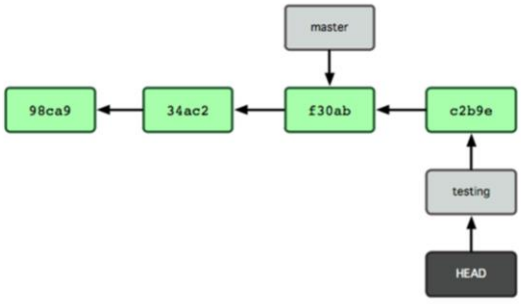


HUAWEI

- HEAD指针，它用来识别当前分支，也就是说，想在哪个分支上操作，就让HEAD指针指向谁。
- 图中，HEAD指向master，表明当前分支是master分支，所有的修改提交都是在master上进行的，当执行git checkout testing，HEAD指向testing分支的最后一次提交，表明当前分支是testing。

Git 分支 (指针4)

- 如何识别当前分支
  - 示例：做一次提交
  - vim test.rb
  - git commit -a -m 'made a change'
  - 每次提交后 HEAD 随着分支一起向前移动

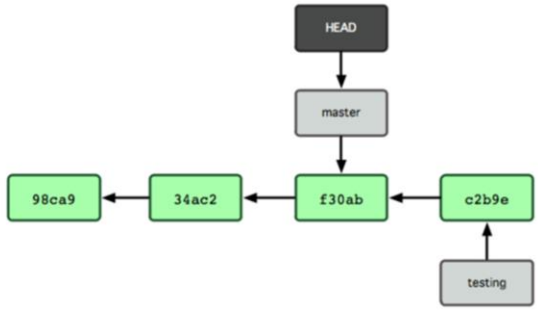


HUAWEI

- 在testing分支上来做一次提交操作，首先在本地工作空间，修改test.rb文件，git commit –a 提交到本地仓库，此时查看log记录，可以看到testing分支向前移动了，HEAD指针也随之向前移动。因为没有操作master分支，因此，master指针没有移动。

Git 分支 (指针5)

- 如何识别当前分支
  - 示例：切换回master分支
  - git checkout master

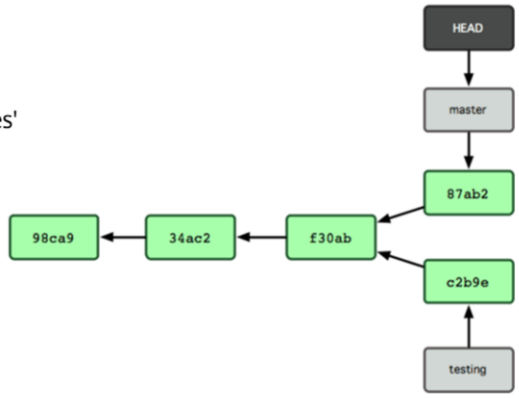


HUAWEI

- 切换回master分支，使用git checkout master命令，使HEAD指针指向master。

Git 分支 (指针6)

- 如何识别当前分支
  - 示例：在master分支上做一次提交
  - vim test.rb
  - git commit -a -m 'made other changes'

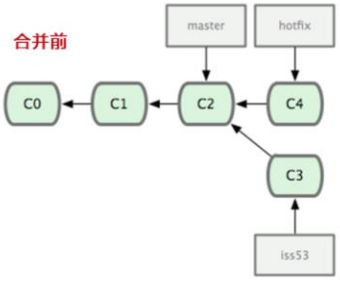


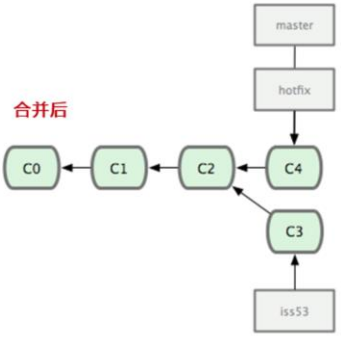
HUAWEI

- 在master分支上再来做一次提交操作。在本地工作空间，修改test.rb文件，git commit –a 提交到本地仓库，此时查看log记录，可以看到master分支也向前移动了，HEAD指针也随之向前移动。此时，testing分支和master分支都有了一次新提交，提交轨迹发生了分岔。

Git 分支 (合并1)

- 分支合并
  - 快进式合并 (fast forward)
    - 顺着一个分支走下去可以到达另一个分支的合并过程



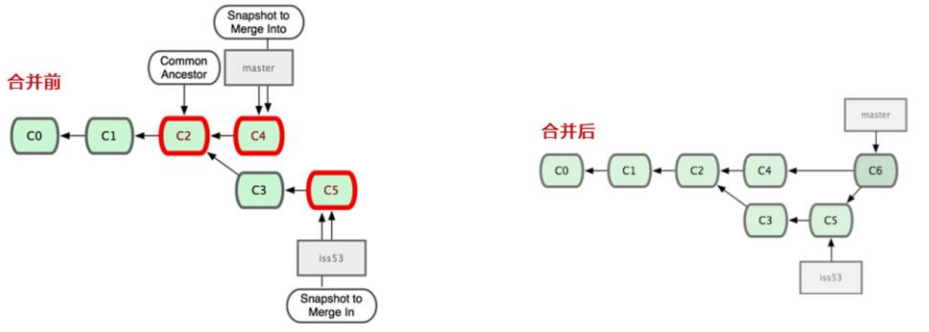


HUAWEI

- 无论哪种工作流都会涉及到分支合并（把一个分支中的修改整合到当前分支），主要有两种方法：合并（merge）和衍合（rebase）。合并有两种方式，快进式合并和非快进式合并，快进式合并比较简单，顺着一个分支走下去可以到达另一个分支的合并过程。
- 图中hotfix比master分支多了一个C4的提交，master分支向前走一步就可以到达hotfix分支，完成了从hotfix、向master分支的合并。

Git 分支 (合并2)

- 分支合并
  - 非快进式合并(non-fast forward)
    - 使用两个分支的末端以及它们的共同祖先进行一次简单的三方合并计算的合并过程

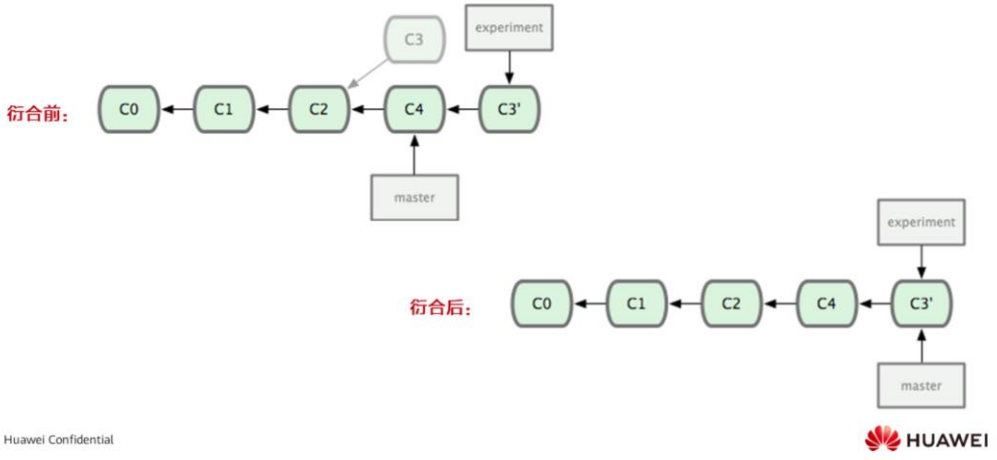


HUAWEI

- 非快进式合并也叫做三方合并。三方指的是master分支中的C4，Iss分支中的C5，以及他们两个之间的共同的父节点C2，和之前将分支指针向前推进所不同的是，git将此次三方合并的结果做了一个新的快照，并且自动创建一个新的提交指向它，被称作是一次合并提交。它的特别之处在于它不只有一个父的提交。需要指出的是，git会自行决定选取哪一个提交作为最优的共同祖先，并以此作为合并的基础。这和更古老的cvs系统，或者svn1.5版本之前不同，这些古老的版本管理系统中，用户需要自己选择最佳的合并基础，git的优势，使其在合并操作上比其他系统简单得多。

Git 分支 (合并3)

- 分支衍合

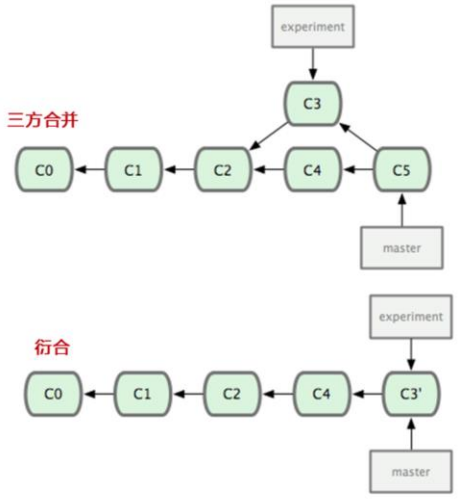


HUAWEI

- 衍合先将master分支新增节点C4以补丁形式保存在.git/rebase目录中，然后同步hotfix分支最新代码，再应用补丁。
- 分支衍合，衍合的原理是先将master分支上新增的节点以补丁的形式保存在.git rebase目录中，然后同步hotfix分支的最新代码。在git中，这种操作叫做变基。
- 可以使用rebase命令，将提交到某一分支上的所有修改都移至另一分支，这就好像重新播放一样。

Git 分支 (合并4)

- 分支衍合
- 对比
  - 和三方合并结果一致，但能产生更简洁的提交历史
- 适用
  - 在主干分支能干净应用的补丁。把解决分支补丁同最新主干代码之间冲突的责任，转化为由提交补丁的人来解决



HUAWEI

- merge和rebase的最终代码没有任何区别，但是变基使得提交历史更加整洁。
- 在查看一个经过变基合并的分支历史记录时会发现，尽管实际的开发工作是并行的，但他们看上去像是先后串行的一样，提交历史是一条直线，一般我们这么做的目的是为了确保在向远程分支推送时，能保持历史提交的一个整洁。
- 例如我们向某个人维护的项目贡献代码时，在这种情况下，首先在自己的分支里进行开发，当开发完成时，需要先将自己的代码变基的合并到自己的远程仓库master分支上，然后再向主项目提交合并请求。这样的话，该项目的维护者就不需要再进行整合工作，只需要快进合并即可。
- 注意到无论是通过变基还是通过三方合并，整合的最终结果所指向的快照始终是一样的，只不过提交历史不同罢了。变基是将一系列提交按照原有次序，依次应用到另一个分支上，而合并是把最终结果合在一起，分支合并就容易产生冲突。

Git 分支 (冲突解决1)

- 合并冲突
  - 类型1：修改了同一个文件的同一行
  - 解决方法：确认正确的修改
  - 示例：命令行解决

```
$ cat doc/README.txt
User1 hacked.
<<<<<<< HEAD
Hello, user2.
=======
Hello, user1.
>>>>>>> a123390b8936882bd53033a582ab540850b6b5fb
User2 hacked.
User2 hacked again.
```

```
User1 hacked.
Hello, user1 and user2.
User2 hacked.
User2 hacked again.
```

```
$ git add -u
$ git commit -m "Merge completed: say hello to all users."
```


HUAWEI

- 修改了同一个文件的同一行代码就会引起冲突。
- 在图中可以看到冲突文件的例子，在等号下面是源分支上代码的修改方式，在等号上面是目标分支的修改方式，因为这是一个逻辑问题，所以工具是无法解决的，这时候就需要开发人员去判断哪一个代码是正常的修改方式，保留正确的，删除错误的，再进行提交即可。

Git 分支 (冲突解决2)

- 合并冲突
  - 类型2：文件被重命名为不同的名字（树冲突）
    - 解决办法：确认哪个名字是正确的，删除错误的
    - 示例：命令行解决

```
$ git rm readme.txt
README: needs merge
doc/README.txt: needs merge
readme.txt: needs merge
rm 'readme.txt'
```

```
$ git rm doc/README.txt
README: needs merge
doc/README.txt: needs merge
rm 'doc/README.txt'
```

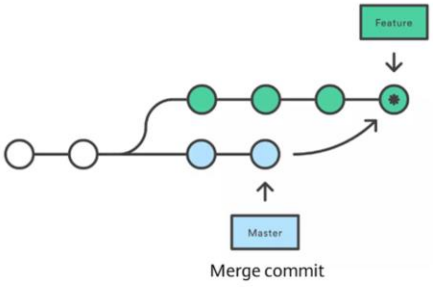
```
$ git add README
```


HUAWEI

- 冲突的第二种方式，文件被重命名为不同的名字，这种的解决方法也很简单，确认哪个名字是正确的，删除错误的再进行提交即可。

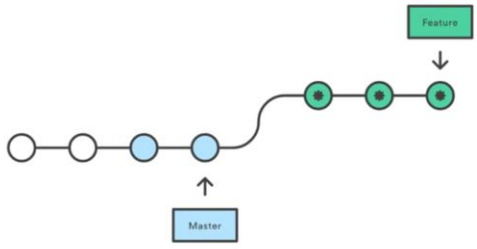
同步远程代码

- Merge 方式同步远程代码
  - git pull remote-name remote-branch



Merge commit

- Rebase 方式同步远程代码
  - git pull -r remote-name remote-branch



Rebase commit

HUAWEI

- 两种方式的区别：
  - Rebase 并没有进行合并操作，只是提取了当前分支的修改，将其复制在了目标分支的最新提交后面；
  - Rebase 的提交历史反映了项目过程中发生了什么，关注点在开发过程上面；
  - Merge 是一个合并操作，会将两个分支的修改合并在一起，默认操作的情况下会提交合并中修改的内容 ；
  - Merge 的提交历史忠实地记录了实际发生过什么，关注点在真实的提交历史上面 。
- 图中的例子，是分别采用两种方法，将远端仓库中的master分支的新增修改，合并到本地feature分支。可以看到，两种操作产生了不同的提交轨迹，使用rebase的操作使feature分支的提交顺序发生了改变，这也是我们为什么称之为变基操作。

跟踪文件

- 查看文件状态
  - git status
- 提交文件
  - git commit –m “提交一个新的文件”
  - git commit --amend
- 提交代码到远程仓库
  - git push remote-name remote-branch

```
g00397277@DESKTOP-IAK9QLS MINGW64 /d/test-project (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
nothing to commit, working directory clean
```

```
g00397277@DESKTOP-IAK9QLS MINGW64 /d/test-project (master)
$ git push git@codehub-devcloud.rnd.huawei.com:zxcsxm00001/test-project.git mast
er
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 327 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@codehub-devcloud.rnd.huawei.com:zxcsxm00001/test-project.git
   3be19f9..008bba3  master -> master
```


HUAWEI

- 通过git status命令查看文件状态，使用git commit 将文件从暂存区提交到本地仓库，使用git push命令将本地仓库代码推送到远端代码服务器。
- 使用git status，可以查看文件的状态。文件的状态包括已修改、已暂存、已提交。每一次操作之后，如何来确定现在的文件所处于的状态，就可以使用git status命令。
- 现在在本地空间修改了代码，想把代码提交到本地仓库中，就需要使用git commit的命令。加入参数m后面可以加具体的提交信息。建议写的信息是有效的信息，这样在查找历史记录时，这个就变得特别的有意义，可以让其他人清晰的看到提交代码的意图。
- 加上amend的参数，就可以对上一次提交追加修改，不单可以追加一些代码的修改，也可以对上一次提交的message信息进行修改。此时想把修改的本地代码推送到远程仓库，这时就可以使用git push命令，将本地仓库的代码同步到远程仓库。

查看提交记录

- 查看提交记录
  - git log
- 搜索历史记录
  - 根据 commiter 搜索
    - git log --author="q00397277*"
  - 根据提交信息查找
    - git log --grep="*A001*"
  - 根据 tag 信息查找
    - git tag -f

```
q00397277@DESKTOP-IAK9QLS MINGW64 /d/new-project (master)
$ git log
commit 02488dc76393f14b596b44f08156187bc696e783
Author: q00397277 <q00397277@huawei.com>
Date:   Wed Mar 4 11:16:04 2020 +0800

    Version 5.0

commit 2a0d6d5b0253f51f9b5a4e8b8ff241c7ed9e6a6c
Author: q00397277 <q00397277@huawei.com>
Date:   Wed Mar 4 11:15:47 2020 +0800

    Version 4.0

commit aa0ff73065d93bc13994f3fe6191d9630c96a6e0
Author: q00397277 <q00397277@huawei.com>
Date:   Wed Mar 4 11:15:29 2020 +0800

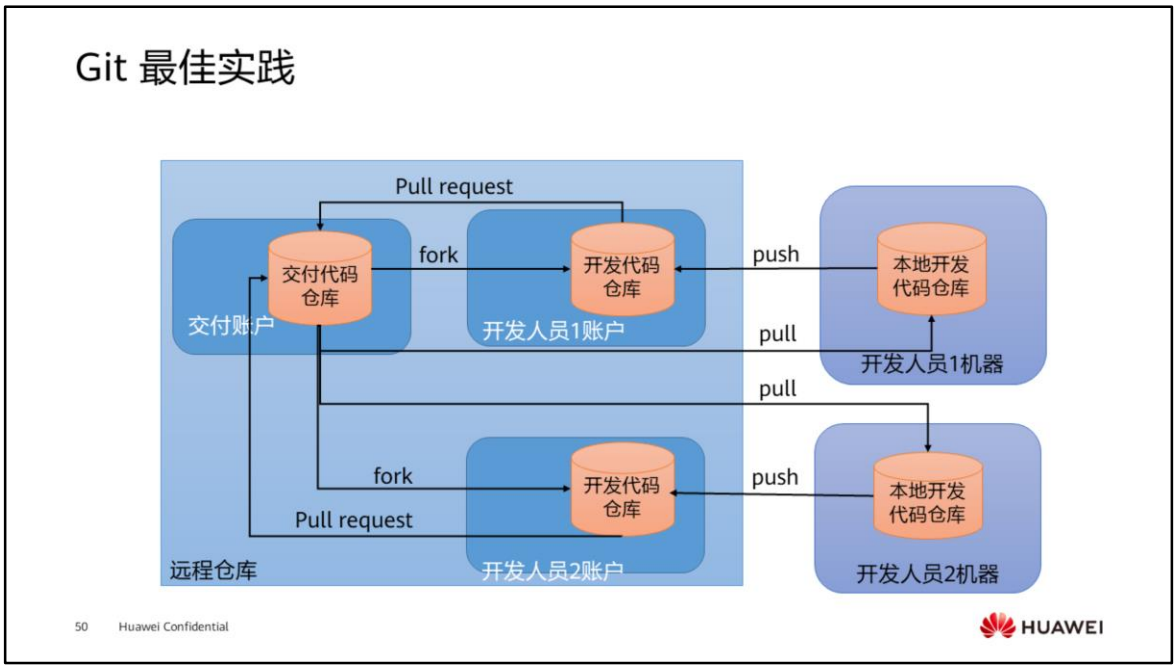
    Version 3.0

commit c15863e28d1f91fc27aa4ed6ac0e1fa2645f2e0c
Author: q00397277 <q00397277@huawei.com>
Date:   Wed Mar 4 11:15:04 2020 +0800

    Version 2.0
:...skipping...
```


HUAWEI

- 使用git log命令可以让我们清晰的看到代码提交的详细信息，如果特别多的时候，我们可以通过—author参数，通过提交人来过滤，也可以通过—grep参数，通过具体的提交信息来过滤。



- 左侧蓝框中是远端仓库，一个交付代码仓库，里面包含部署到生产环境的代码，还有两个开发人员从交付库中fork的远端仓库，包含了原来上游交付仓库中所有内容，如分支、Tag、提交。开发人员将代码拉取到本地进行开发，开发完成后push到个人远端库，最后向交付库中发起Pull Request请求，此时，项目架构师或者技术leader，负责审核把关代码质量，审核通过的代码才会进入交付代码仓库。通过这种方式，有效保证了交付代码仓库的质量。

目录

1. 持续集成理念、方法与实践
2. 代码托管与分支策略
3. Git 基本概念及主要操作
4. **代码提交及代码评审**
5. 华为云 DevCloud 代码托管服务及 CloudIDE
6. 静态代码检查

HUAWEI

代码提交过程 (1)

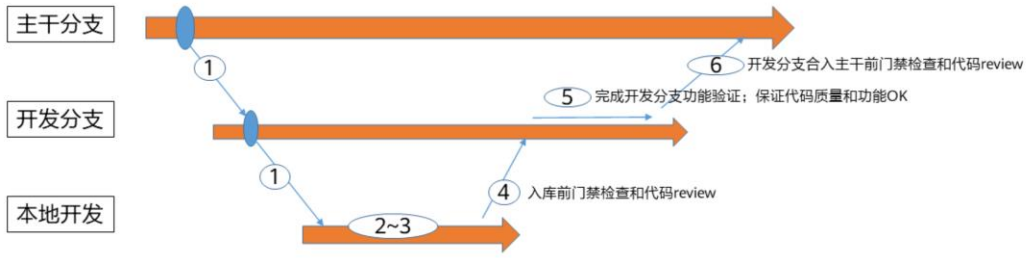
- 当团队成员较多，且在代码主干上进行开发时，需严格遵从纪律，保障团队高效协作；在进行分支-主干开发模式时，团队成员一般遵循如下步骤：

1. 创建开发分支，检出最近的成功代码
2. 本地修改代码
3. 在本地进行编译构建，确保本地修改没问题
4. 合入自己的开发分支，并触发分支门禁构建，通过之后进行代码审核再合入，确保代码质量达标
5. 开发进行功能验证
6. 合入到团队主干分支，触发主干门禁构建，门禁构建通过之后，再通过代码评审合入到主干，保障主干代码质量

HUAWEI

- 常见的开发模式有两种：
  - 直接在主干上进行开发；由于所有人都在主干开发，每一次提交都会直接影响主干代码，所以需要团队成员严格遵守纪律，从而保障团队高效协作。
  - 使用分支-主干模式进行开发，使用分支可以保证主干代码不受损害。
- 团队成员一般遵循如下步骤：
  - 第一步，创建开发分支, 检出最近的成功代码。 代码检出后，开发人员切换到开发分支，然后开始本地修改代码。某个功能修改代码完成，我们需要将修改提交到开发分支，为了保证本地修改没问题，在提交之前，需要在本地进行编译构建，即将产品打成二进制包。如果本地构建可以构建成功，则可以将代码提交到开发分支; 如果本地构建失败，需要查看问题所在，修改本地代码，修改完成再编译，直到构建成功提交。提交后，代码触发团队配置好的门禁包括静态检查，构建等过程，门禁通过后由项目其他成员进行手工代码审核，代码审核通过后，对开发分支的功能进行验证，验证通过后，向主干分支提出合并请求，项目成员对合并请求进行评审，评审通过后，代码合入主干分支，这样可以保证主干分支代码质量。

代码提交过程 (2)



HUAWEI

- 这是代码提交过程的图解。如图从主干分支创建开发分支，然后本地开发，为保证代码质量，代码提交入库前需做代码review和门禁检查；各项检测通过，进行功能验证，保证代码质量和功能没有问题，最后合入主干分支。

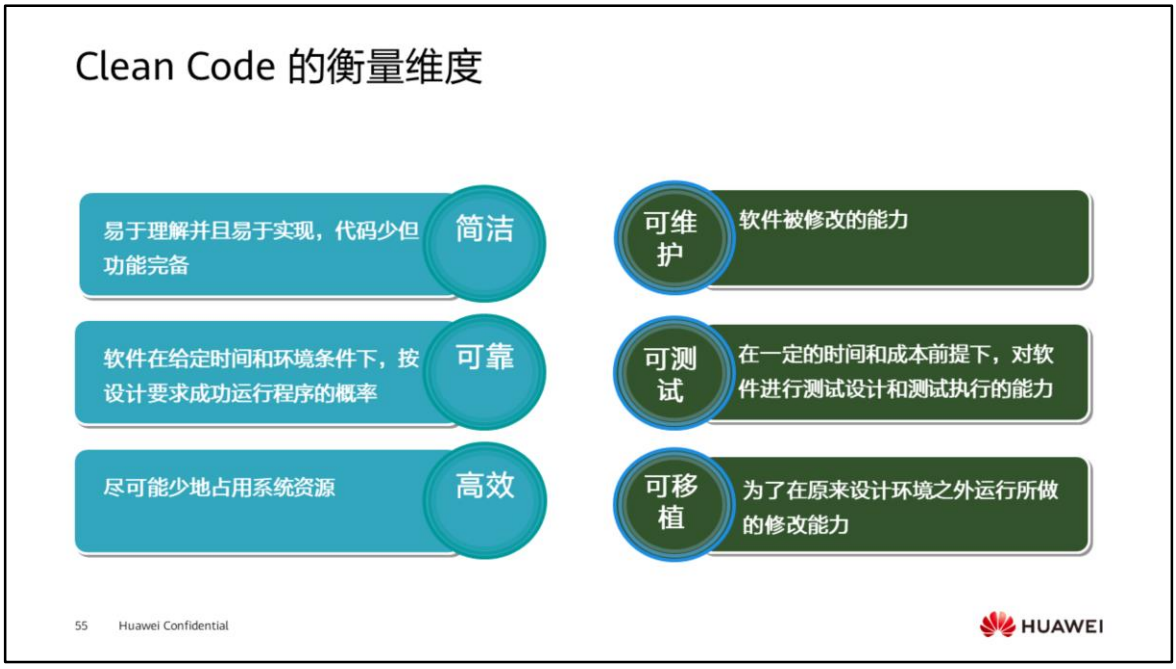
Clean Code

- 代码可信在华为就是用 Clean Code 来衡量

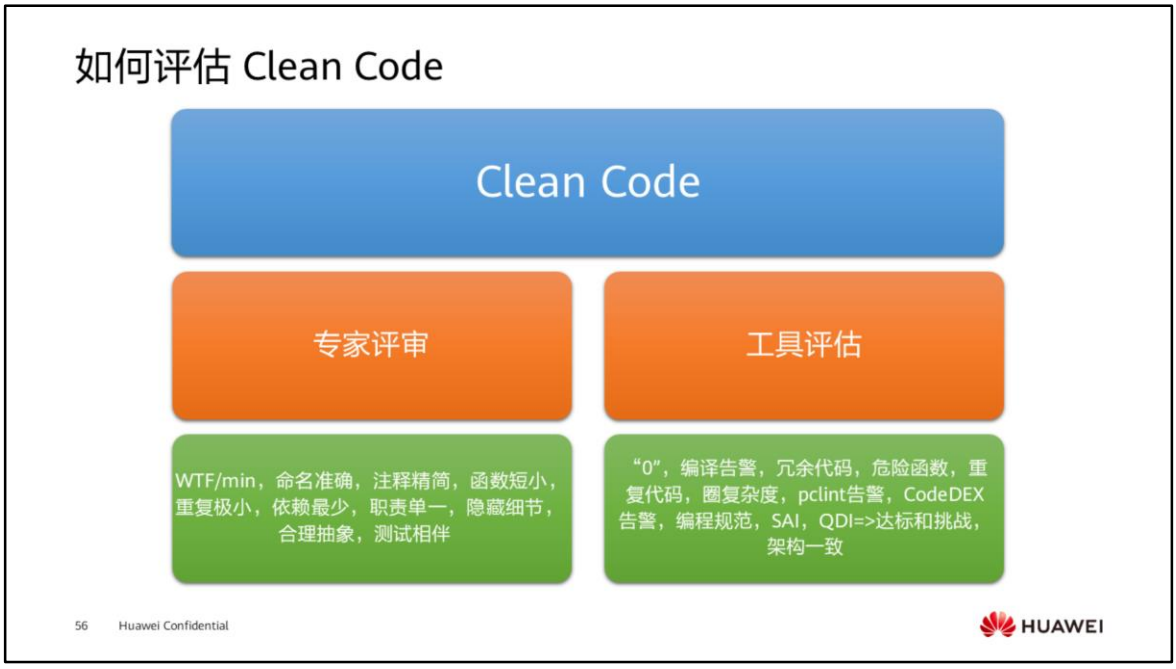


HUAWEI

- 代码提交的过程，也强调了代码质量的重要性，如何保证代码质量，靠的是代码评审。华为的代码可信标准是用clean code来衡量。
- 要满足clean code，代码就必须符合smart原则，即：简洁，可靠，高效，可维护，可测试，可移植。当然除了这些，代码还要保证安全性，不能被外部攻击。



- 简洁：易于理解并且易于实现，代码少但功能完备，代码的内部质量影响代码的外部质量，简洁是代码内部质量的核心。
- 可靠：软件在给定时间和环境条件下，按设计要求成功运行程序的概率，如果一段代码运行经常容易报错，这种代码达不到发布条件。
- 高效：代码被阅读和被修改的次数，远远多于它被编写的次数。所以要尽可能少地占用系统资源（效率）。
- 可维护性：是指软件被修改的能力，一段代码具备相应功能，但是重构起来很麻烦也是不可取的，所以代码应有很好的可维护性。
- 可测试性：是指在一定的时间和成本前提下，对软件进行测试设计和测试执行的能力。
- 可移植性：是指为了在原来设计环境之外运行所做的修改能力。



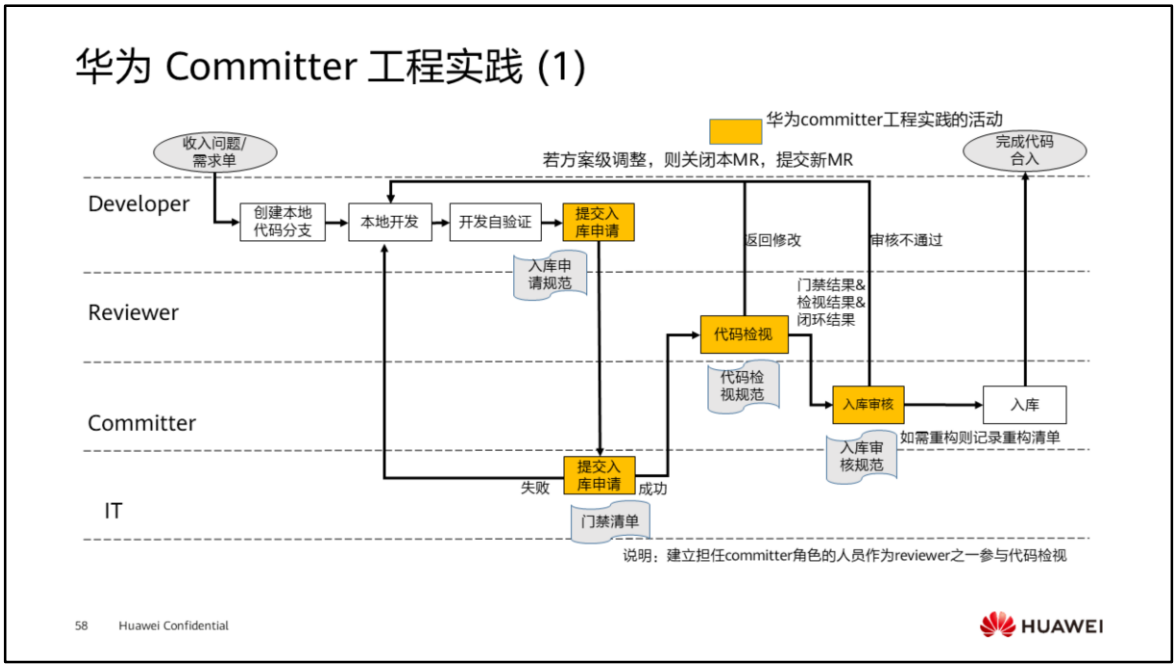
- 如何评估clean code？主要从两方面入手：专家评审，工具评估。
- 专家评审包括“每分钟爆粗口的次数”，命名准确，注释精简，函数短小，重复极小，依赖最少，职责单一，隐藏细节，合理抽象，测试相伴等维度。
- 每分钟爆粗口的次数：当看到写的很差的代码，往往会吐槽，甚至爆粗口，越差的代码爆粗口次数显然也就越多，反之则表示代码质量很好，用每分钟爆粗口次数简单直白的衡量代码质量，甚至有人将其称为衡量代码质量的唯一标准。
- 工具评估是指使用一些代码检查工具，对代码各方面维度包括编译告警，冗余代码，危险函数，重复代码，圈复杂度，pclint告警，CodeDEX告警，编程规范，SAI，QDI=>达标和挑战架构一致等进行检查。

Clean Code 可视化指标

指标名称	指标含义	编程指导
圈复杂度	衡量一个函数判定结构的复杂程度，表现为独立路径的条数，即合理的预防错误所需的测试的最少路径条数	减少函数的分支数（if/else, switch/case, for/while），分支逻辑尽量用函数封装
嵌套层数	函数中控制结构嵌套的深度	嵌套层数不要过深。即 {} 的嵌套不能太多
有效注释比例	函数中有明确含义的注释语句与函数物理行的比值	变量及函数名实现自注释，注释说明why而不是what，注释充分而简洁
有效代码行数	函数中有实际语义的语句个数，可理解为NBNC(非空非注释)去掉括号等分隔符之后的代码行数	函数体不要过大。逻辑语句不能太多，即";"的个数。要注意适当做函数封装.
函数参数个数	/	减少函数参数；引入结构体变量
局部变量个数	/	数据关系是由代码的功能引起的，尽量把单一功能的函数进行封装，就可以减少局部变量
非结构化语句的数量	影响函数条理性的语句（例如：异常退出语句、goto语句等）的数量	尽量不使用

HUAWEI

- 圈复杂度是指衡量一个函数判定结构的复杂程度，表现为独立路径的条数，即合理预防错误所需的测试的最少路径条数，该尽量减少函数的分支数，就是少用if/else，switch/case等。
- 嵌套层数：嵌套层数是指函数中控制结构嵌套的深度，指导意见是嵌套层数不要太多，即大括号的嵌套不能太多。
- 有效注释比例：函数中有明确含义的注释语句与函数物理行的比值，该怎么去做变量及函数名实现自注释，注释说明why而不是what。注释充分而简洁，有效代码行数。
- 有效代码行数：是指函数中有实际语义的语句个数，可以理解为NBNC去掉括号等分割符之后的代码行数。指导意见就是函数体不要过大，逻辑语句不能太多，即分号的个数，要适当注意做函数封装。
- 函数参数个数：减少函数参数；引入结构体变量。
- 局部变量个数：尽量减少函数参数，引入结构体变量，局部变量个数。数据关系是有代码的功能引起，尽量把单一功能的函数进行封装，就可以减少局部变量，非结构化语句的数量。
- 非结构化语句的数量：是指影响函数条理性的语句，例如异常退出语句，goto等。给出的意见就是尽量不使用这些非结构化语句。



- 华为committer工程实践：
  - Developer接受到问题或需求，然后创建本地分支进行功能开发，开发完成后进行自验证，验证成功后，向主干分支提交入库申请并且分别选择Reviewer和Committer。这时通过配置门禁工具，进行代码门禁检查；如果代码质量符合门禁要求，则由reviewer进行人工代码检视，Reviewer收到代码检视通知后，对代码进行检视并提出代码检视意见，代码检视结果符合代码检视规范，则将代码提交给committer，由committer按照入库规范审核，审核通过后进行代码入库。以上任意环节出现问题，都将代码打回，由developer重新再本地进行修改。

华为 Committer 工程实践 (2)

- 工程实践过程说明：
  - Developer在完成代码开发和自验证后，发起代码入库申请，分别选择Reviewer和Committer
  - IT工具触发门禁检查，门禁检查成功后，通知Reviewer检视
  - Reviewer收到代码检视通知后，对代码进行检视并提出代码检视意见
  - Developer收到检视意见后，完善代码并闭环检视意见，更新代码入库申请
  - 最后，Committer根据工具检查的结果和Reviewer的意见及闭环情况，并从代码和架构的长远视角进一步对代码进行审核，决策是否接纳本次代码入库申请
- 在保证代码质量的前提下，门禁检查和代码检视可并行执行。同时，Committer还可对代码检视的策略进行灵活调整

HUAWEI

华为 Committer 工程实践 (3)

序号	角色名称	关键任务
1	Developer	开发特性需求，修改问题，对合入代码质量承担第一责任
2	Reviewer	对业务逻辑的正确性、代码规范的遵从度、代码架构等方面进行检视，并提出修改意见。若有多名Reviewer，分工可各有侧重
3	Committer	审核（含代码检视）申请入库的代码质量，对审核的代码质量承担共同责任。担任Committer角色的人员应作为Reviewer之一

HUAWEI

代码入库审核的价值

- 基于当前可信诉求的提高，代码提交与合入权限应要分离，以避免攻击者冒用员工权限植入恶意代码；
- 同时通过检视和审核的意见对工程师进行辅导提升团队软件能力。
- 入库审核还能反向驱动前端代码检视提升，促进Developer具备编写Clean Code的能力。

HUAWEI

目录

1. 持续集成理念、方法与实践
2. 代码托管与分支策略
3. Git基本概念及主要操作
4. 代码提交及代码评审
5. **华为云 DevCloud 代码托管服务及 CloudIDE**
6. 静态代码检查

HUAWEI

CodeHub 的安全优势 - DevCloud 层级

- DevCloud层面的安全特性
  - 访问控制
  - 鉴权
  - 基于角色与权限的细粒度授权
  - 不可抵赖性
  - 数据保密性
  - 通讯安全
  - 数据完整性
  - 可用性



HUAWEI

- 代码托管服务也叫codehub，是云端的代码仓库。CodeHub安全性优势有两个层面，一个是devcloud层级，还有codehub层级。
- DevCloud层面的安全特性有：
  - 访问控制：公有云采用“租户+用户+用户组+角色”统一模型对权限进行控制，DevCloud在管理面按照本模型进行控制，在业务面基于角色权限模型进行控制，确保每个租户/用户只能访问已授权的资源。
  - 鉴权：DevCloud基于公有云统一的认证服务IAM来进行认证，用户在认证通过后，以返回的Token进行访问DevCloud服务。对于用户通过HTTPS/SSH访问代码仓库，将使用SSH Key或者仓库用户名及密码进行访问。
  - 基于角色与权限的细粒度授权：不同的角色，在不同的服务中，根据不同的资源，可以有不同的操作权限。这些权限，在具体的项目，具体的服务中，还可以做自定义的权限设置。
  - 不可抵赖性：DevCloud基于公有云IAM Token机制，所有操作都必须带有Token，此Token通过IAM系统有效性检查。租户身份是无法伪造的。DevCloud对所有租户的操作进行权限判断，只有符合条件才允许执行，并对所有关键操作进行审计记录。审计日志被持久化，可保留足够长时间，并可进行精确的回溯。

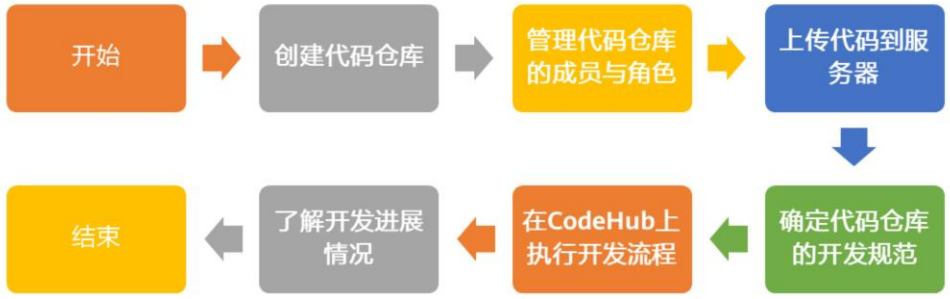
- 数据保密性：DevCloud的数据传输均采用HTTPS/SSH等安全协议，保证传输安全；对于敏感信息，DevCloud会进行加密等进行存储。
- 通讯安全：DevCloud对外提供的服务均使用HTTPS、SSH等安全协议，保证了通讯的安全性。
- 数据完整性：DevCloud的关键信息都保存在内部数据库中，通过事务等各种机制保障了数据的一致性。
- 可用性：DevCloud的各个服务都是集群方式，通过保证了服务的高可用性。
- 隐私：DevCloud不涉及到租户及用户的隐私。

CodeHub 的安全优势 - CodeHub 层级

- CodeHub 层面的安全特性：
  - 基于角色与权限的细粒度授权：在CodeHub层面，提供针对代码访问的，更加细粒度的授权模型。
- 不可抵赖性
  - 我们提供代码仓库的完整访问日志，供用户审计。
- 数据加密
  - 用户的代码在CodeHub中，是以加密方式存储的。

HUAWEI

CodeHub 的关键流程



HUAWEI

创建代码仓库

- 在代码托管的仓库列表页面，点击新建仓库
  - 分为代码托管服务的总列表，与某一个项目下的代码仓库列表
  - 未创建项目前，无法创建仓库——代码仓库一定要从属于某个项目







- 在新建代码仓库的时候，Devcloud提供了三种新建仓库的模式，分别是普通新建，模板新建和导入仓库。
  - 普通新建创建的是一个空白的仓库；
  - 模板新建可以使用Devcloud 提供的上百种经典仓库模板构建代码仓库，同时支持用户分享模板；
  - 导入仓库：支持导入外部的Git仓库和SVN仓库。



- 仓库创建完成，可以调整仓库成员的权限，这也体现了安全性。添加成员的方式是管理员可以从成员列表进入，管理代码仓库的成员，同时仓库角色对应到研发环节的各个成员。

上传代码到服务器

- 支持SSH/HTTPS协议
  - CodeHub支持通过SSH协议与HTTPS协议，读写代码仓库
  - 基本上所有主流的Git客户端，都支持这两种协议
- 支持WebHook
  - 在代码仓库发生变动时，可以主动的通知用户配置的Hooks服务器
- 支持IP白名单设置
  - 在IP白名单范围内的仓库访问才是允许的，除此之外其他IP发起的访问一律被拒绝



HUAWEI

- 上传代码到服务器的操作和本地仓库类似，需要先配置ssh或者https协议，再传代码至codehub。同时codehub支持webhook，在代码仓库发生变动时，可以主动的通知用户配置的Hooks服务器。Codehub还支持IP白名单设置，在IP白名单范围内的仓库访问才是允许的，除此之外其他IP发起的访问一律被拒绝。

确定开发规范

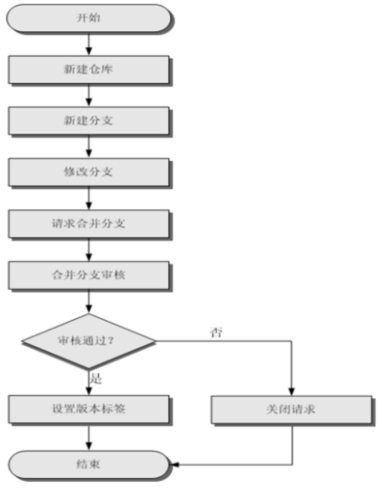
- 设定默认分支：master
- 确定保护分支：master 与 develop
- 确定分支的命名规范：
  - Features/abc【特性描述】
  - hotfix/115【问题单日期】
- 确定版本的命名规范，并用这个规范来打标签
- 确定团队内部统一的Commite规范
- 确定合并请求的参与者，职责

HUAWEI

- 可以在仓库配置中设置开发规范，确定默认分支，保护分支，分支命名，版本命名规范等。

CodeHub 开发流程

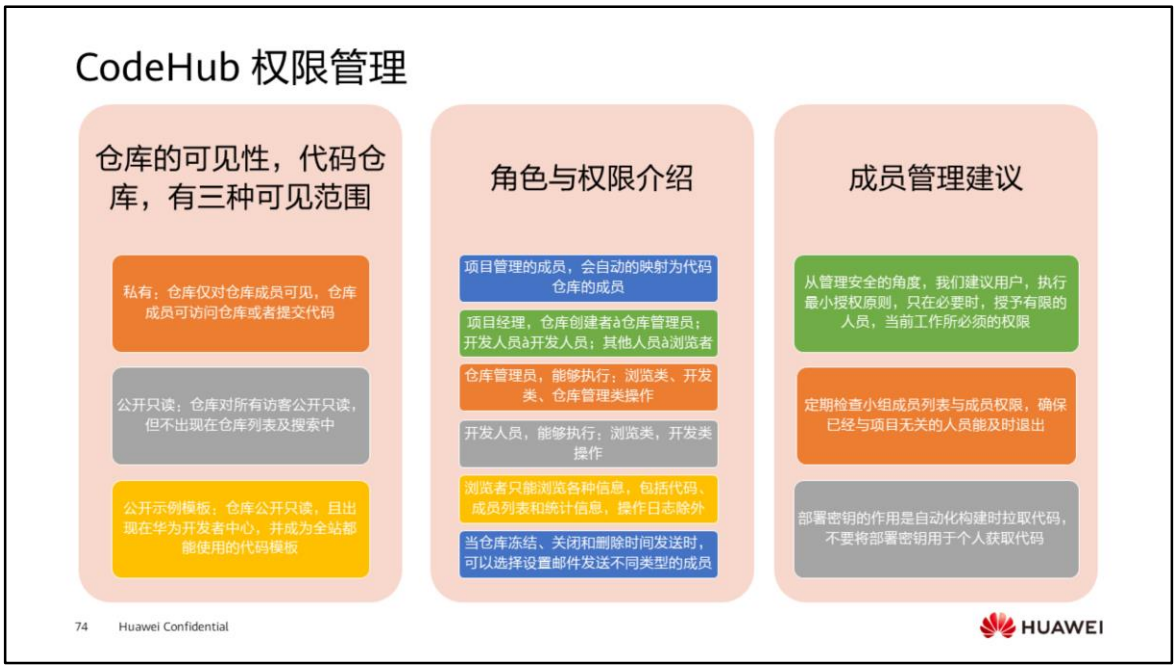
- 开发人员下载代码
- 创建工作分支
- 提交代码
- 创建合并请求
- 邀请团队成员，参与代码评审
- 仓库管理员，审视后合入代码



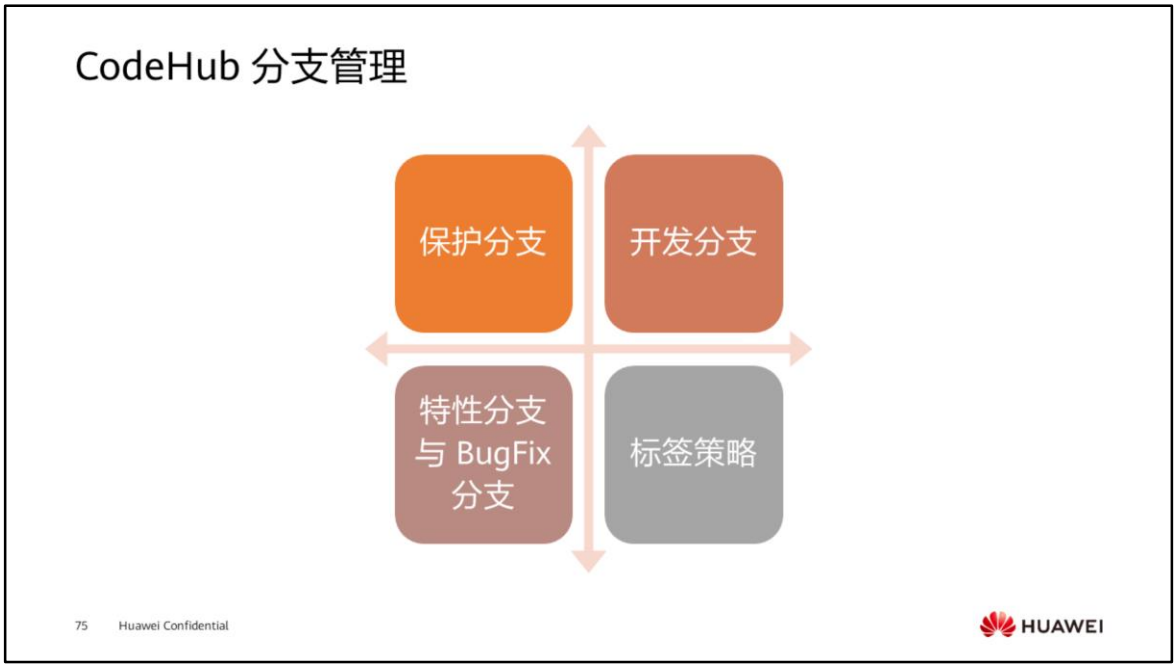
HUAWEI



- Codehub可以查看代码提交关联工作项，便于了解本次提交具体对应哪个功能，还可以查看提交历史，以及仓库统计，仓库统计是通过语言，提交频率等多个维度呈现的图形化统计功能。

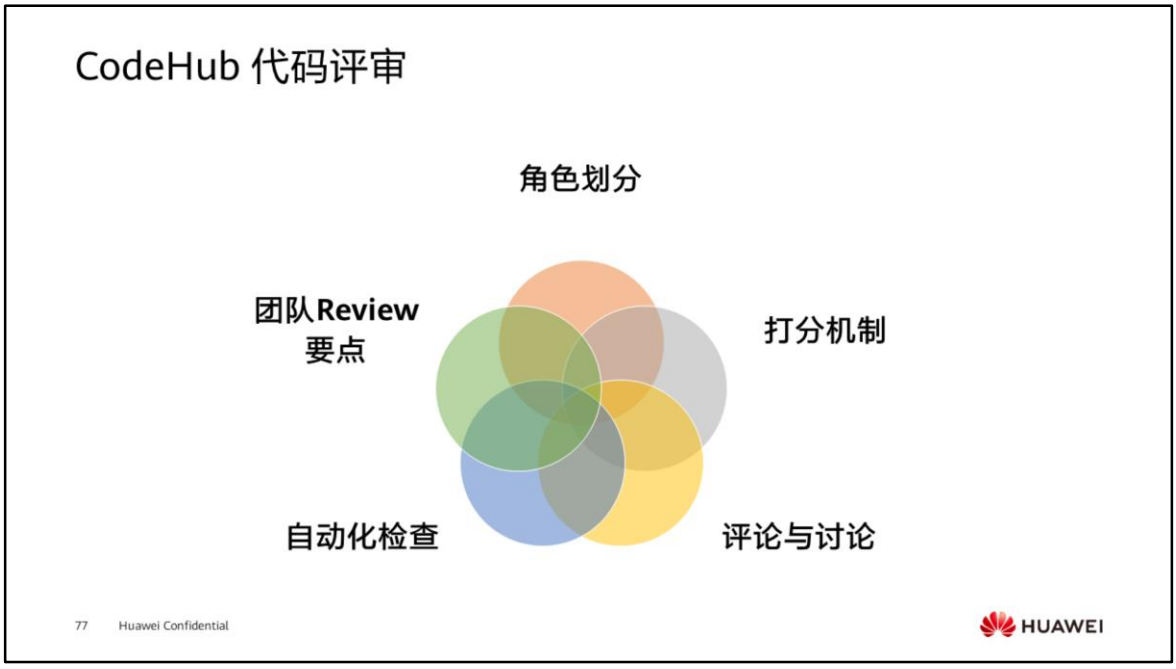


- Codehub中的代码仓库可见性有三种：私有，公开制度，公开示例模板。
  - 私有：仓库仅对仓库成员可见，仓库成员可访问仓库或者提交代码；
  - 公开只读：仓库对所有访客公开只读，但不出现在仓库列表及搜索中；
  - 公开示例模板：仓库公开只读，且出现在华为开发者中心，并成为全站都能使用的代码模板。
- 不同角色在代码仓库中，权限也是不同的：
  - 仓库管理员，能够执行：浏览类、开发类、仓库管理类操作；
  - 开发人员，能够执行：浏览类，开发类操作；
  - 浏览者只能浏览各种信息，包括代码、成员列表和统计信息，操作日志除外。
- 关于成员管理的几条建议：
  - 从管理安全的角度，我们建议用户，执行最小授权原则，只在必要时，授予有限的人员，当前工作所必须的权限；
  - 定期检查小组成员列表与成员权限，确保已经与项目无关的人员能及时退出；
  - 部署密钥的作用是自动化构建时拉取代码，不要将部署密钥用于个人获取代码。



- Code支持多种分支管理：
  - 通过设置保护分支，能够确保分支代码安全，并阻止强行推送代码，删除分支等操作；
  - 我们通常将 master 分支，设置为保护分支，小组成员的代码，都可以从开发分支，提交到 master 分支；
  - 如果严格执行 Git-Flow，则应增加一个 Develop 分支，也设置为保护分支。
- 开发分支：
  - 固定接受合入代码的分支，可以设置进阶的开发分支规则；
  - 自动合入代码，在小规模团队，简单项目中适用，能够大大的提升效率。复杂项目需慎用；
  - 自动关闭代码，在代码质量有严格要求的团队适用。
- 特性分支与 BugFix 分支：
  - 基于Git-Flow工作流，除了Master、Develop 和 Release 分支之外，会有多条短时间存在的特性分支和 BugFix 分支；
  - 在开发完成，代码被后合入后，这些分支应该被删除；
  - 分支的命名是团队内部协商的结果，需要有明确的约定规范。

- 分支合并的策略：
  - 明确分支的开发规范与 Commit 注释规范，在团队内部培养合理的提交注释格式；
  - 需要培养调整、重整提交历史的能力；
  - 基于 Code Review 的工作流程，明确仓库管理员，能够在何种情况下，合入代码。
- 标签策略：
  - 标签的作用，是标识阶段性的成果，关键在于版本的命名规范；
  - 拉取代码，不要将部署密钥用于个人获取代码。



- Codehub如何进行Code Review 主要通过图上这五个方面：
  - 角色划分
    - 一次代码评审（ Code Review ），包含提交者、合并者与评审者的固定角色，其他团队成员，也可以围观和评论；
    - 提交者发起代码的合并请求，并设置合并者与评审者（可多位人员）；
    - 评审者和合并者可以这个合并请求评分，或发出评论；
    - 合并者可以根据代码质量与自动化检查的结果，确定是否合入代码。
  - 打分机制
    - 代码仓库可以设置任一合并请求的最低合入得分，通过设置合理的分值，可以促进团队内，更多的成员，参与代码的评审；
    - 管理员可以+2 或-2 分，普通的成员，可以+1分或-1分；
    - 团队内的所有成员，都可以参与评分，提交者也可以邀请更多的人来参与打分。

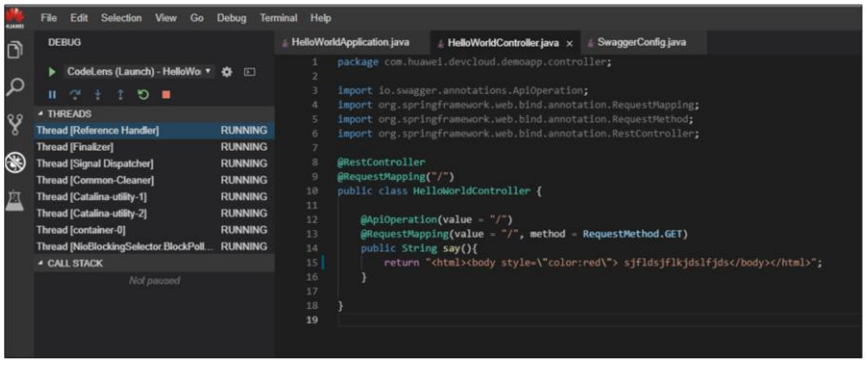
- 评论与讨论
  - 普通的评论，无需回复，也可以被忽略；
  - 发起一个讨论，代表着一个需要被解决的问题，如果该问题没有被解决，代码将无法被合入；
  - 也可以在浏览代码变更时，针对一行具体的代码，发起一个讨论；
  - 支持源代码的在线编辑能力；在讨论过程中，可以实时在线编辑源代码。
- 自动化检查作用
  - 通过配置，合并请求可以自动触发自动化的检查机制；
  - 代码的评审人可以参考自动化检查的结果，判断代码质量。
- 团队Review的要点
  - Code Review 是全员的工作，需要尽可能多的人参与
  - 如果一段代码太长，甚至读不懂，不要犹豫，这肯定是代码质量不高的标志；
  - 团队应该定期围绕代码进行交流，提示共同的代码质量意识，以及统一对于代码质量的标准。



- Git代码仓库
  - SSH/HTTPS协议支持
  - 分支与标签支持
  - 代码提交历史
  - 仓库网络
  - WebHook
- 安全特性
  - 成员权限管理
  - 保护分支与开发分支管理
  - 消息通知机制
  - 安全白名单
  - 操作日志
- 其他特色功能
  - Code Review
  - 在线编辑
  - 统计图表
  - 基于模板创建仓库
  - 导入Git/SVN仓库

CloudIDE 概述

- CloudIDE 是 DevCloud 的云端开发环境服务，向开发者提供按需配置、快速获取的工作空间（包含编辑器和运行环境），支持完成环境配置、代码阅读、编写代码、构建、运行、调试、预览等操作，并支持对接多种代码仓库。



HUAWEI

- CloudIDE 是 DevCloud 的云端开发环境服务，向开发者提供按需配置、快速获取的工作空间（包含编辑器和运行环境），支持完成环境配置、代码阅读、编写代码、构建、运行、调试、预览等操作，并支持对接多种代码仓库。
- 图中是入口以及系统界面。

CloudIDE 特性列表

特性	描述
云化和轻量化	依托华为云的计算和存储资源，实现云化开发环境供给，通过浏览器访问就可完成开发全过程，实现移动办公
快速按需容器化	用户工作空间基于全容器技术，极短时间即可按用户所需配置（计算和存储）启动并提供服务，用后随时释放
多语言和技术栈	支持40多种语言的语法高亮，支持Java等语言的语法补齐，支持7种预置技术栈，无需复杂配置环境即可就位
视图风格可切换	提供亮色和暗色两种视觉风格，编辑器提供多种视图布局，还支持满屏编辑和边栏收缩
后端环境可配置	提供页面终端（WebTerminal）直接访问后端容器环境，以命令行设置变量、处理文件和配置其他环境因素
构建运行和调试	提供命令管理器以支持构建（Build）和运行（Run），对Java等语言还支持断点调试
可对接三方服务	除了对Git仓库(DevCloud代码仓库或GitHub等）的支持，还提供对接第三方服务（即通过外网通道对接其他开放服务）的能力
企业化权限管控	提供基于黑白名单的访问控制能力，为企业租户提供掌控子用户行为的管理面

HUAWEI

CloudIDE 服务概述 (1)

- Cloud IDE: 定义一种新的在线编程体验

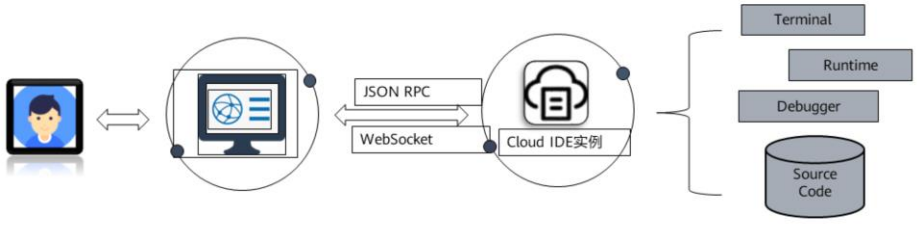


HUAWEI

- 传统的代码编辑器是轻量级，以编辑体验为中心，支持多语言高亮显示，并且可以快速访问文件。
- 本地化IED功能强大（比如eclipse，idea），通常支持代码编辑，项目系统，调试，自动化构建等。
- WebIDE则继承了双方的优点。

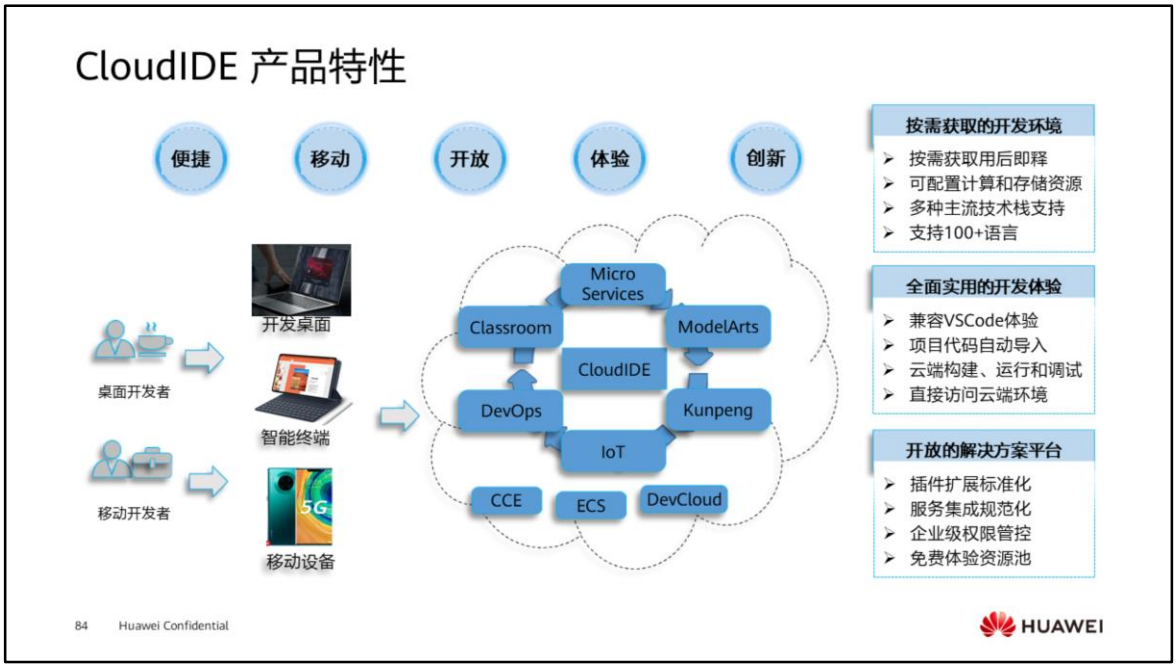
CloudIDE服务概述 (2)

- Cloud IDE: 运行在云端的轻量级集成开发环境
- Cloud IDE 服务特性



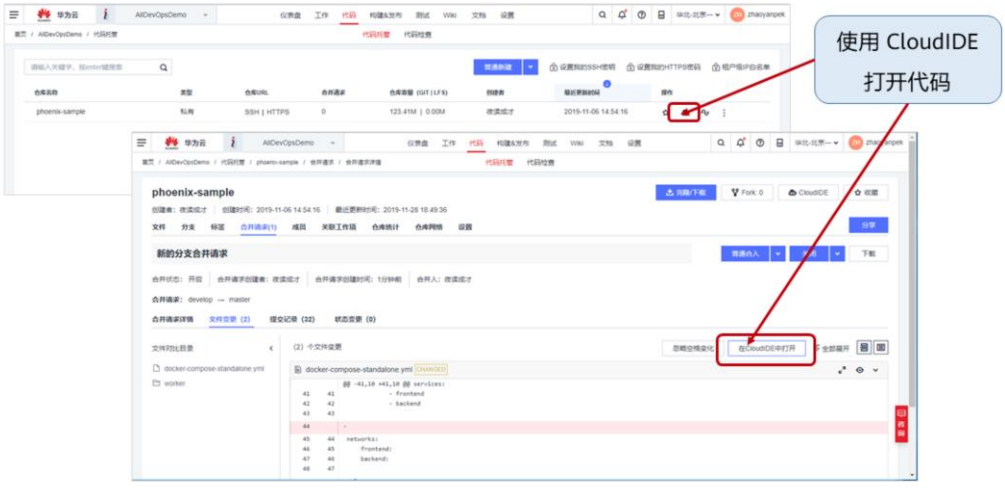
HUAWEI

- CloudIDE本质是运行在云端的轻量级集成开发环境，其服务特性有：
  - 按需获取，轻量，极速
  - 支持主流编程语言
  - 打通开发态和运行态
  - 丰富的插件生态
  - 支持华为鲲鹏原生



- 开发者可以通过开发桌面，智能终端，移动设备访问华为云cloudIDE，同时cloudIDE可以连通华为云其他应用：kunpeng，DevCloud, Modelarts等。

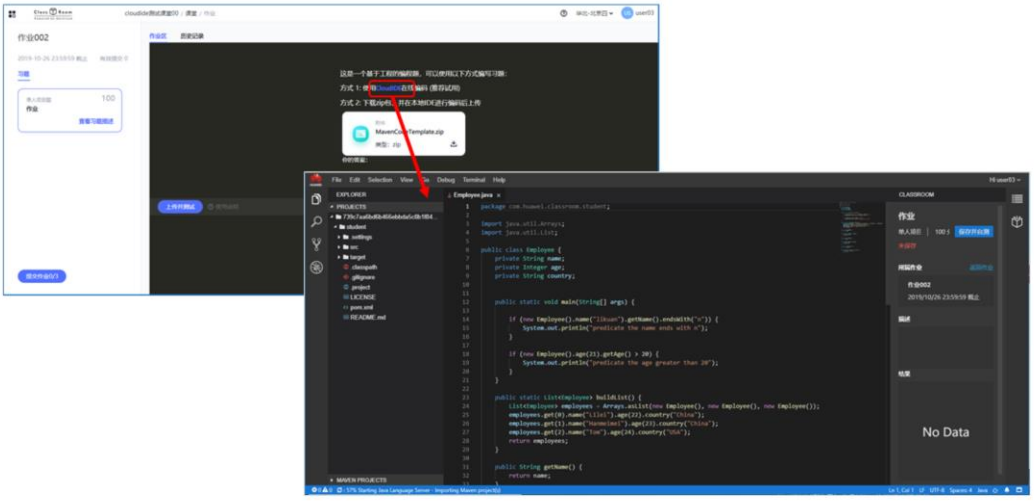
服务集成 (对接 CodeHub)



HUAWEI

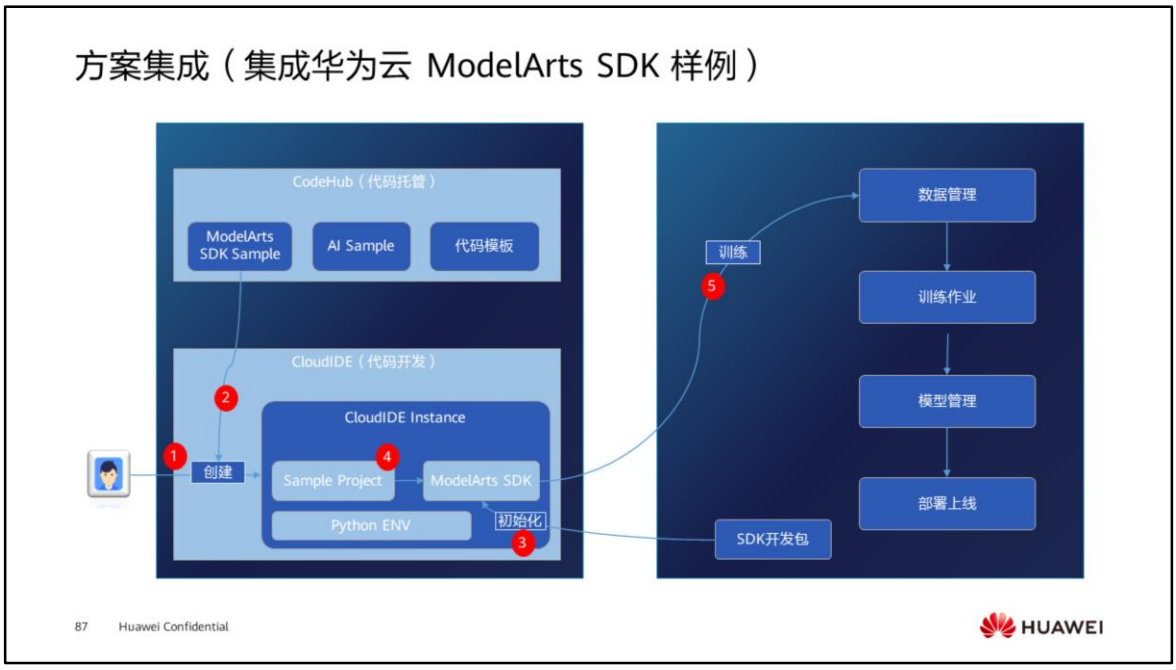
- 在DevCloud的很多应用中，如CodeHub 我们可以直接用CloudIDE打开代码文件。

服务集成 (对接 Classroom)



HUAWEI

- 在classroom应用中，学生可以直接在编程作业中进入CloudIDE，直接编写代码，省去了本地搭建研发环境的精力。



- 当多个服务连到一起可以组成一个方案，以集成华为云modelarts sdk为例，Modelarts是华为云AI学习平台。将代码上传至DevCloud的CodeHub，然后以代码为基础创建CloudIDE实例，并加载modelarts sdk，修改配置文件后，即可调用Modelarts应用进行模型训练了。

目录

1. 持续集成理念、方法与实践
2. 代码托管与分支策略
3. Git基本概念及主要操作
4. 代码提交及代码评审
5. 华为云 DevCloud 代码托管服务及 CloudIDE
6. **静态代码检查**

HUAWEI

什么是静态代码检查

- 静态代码检查，又称为静态程序分析（英语：Static program analysis)
- 是指在不运行计算机程序的条件下，进行程序分析的方法
- 大部分的静态程序分析的对象是针对特定版本的源代码，也有些静态程序分析的对象是目标代码

HUAWEI

为什么做静态代码检查

- 代码交付过程中的常见问题和风险

潜在问题	风险
重复代码过多	造成开发人力的浪费以及后期维护成本增加
编码风格糟糕	代码凌乱、不可读，难于维护与开发修改
圈复杂度过高	造成代码可维护性、可继承性降低，问题定位难度加大
编码安全风险	使用具有安全风险的函数，导致系统的安全性层级降低，加大系统的安全风险

HUAWEI

- 静态代码检查可以检查出代码当中潜在的问题，不运行程序就能检查出以上问题。

代码静态检查关注点

编号	检查点（举例）	概述
1	重复率	表示一段源代码在一个程序，或者一个团体所维护的不同程序中重复出现，是不希望出现的现象
2	代码风格	程序开发人员所编写源代码的书写风格，良好代码风格的特点是使代码易读
3	圈复杂度	衡量一个模块判定结构的复杂程度，数量上表现为独立线性路径条数，圈复杂度大说明程序代码可能质量低且难于测试和维护
4	代码安全	编码过程中，常见的安全问题包括（但不限于）：缓冲区溢出/跨站脚本攻击(XSS)/SQL注入/XML 注入/LDAP 注入

HUAWEI

- 静态代码检查的4个维度，对应之前提到的4种风险
  - 代码重复率
  - 编码风格
  - 圈复杂度
  - 代码安全

静态代码检查常用分析技术

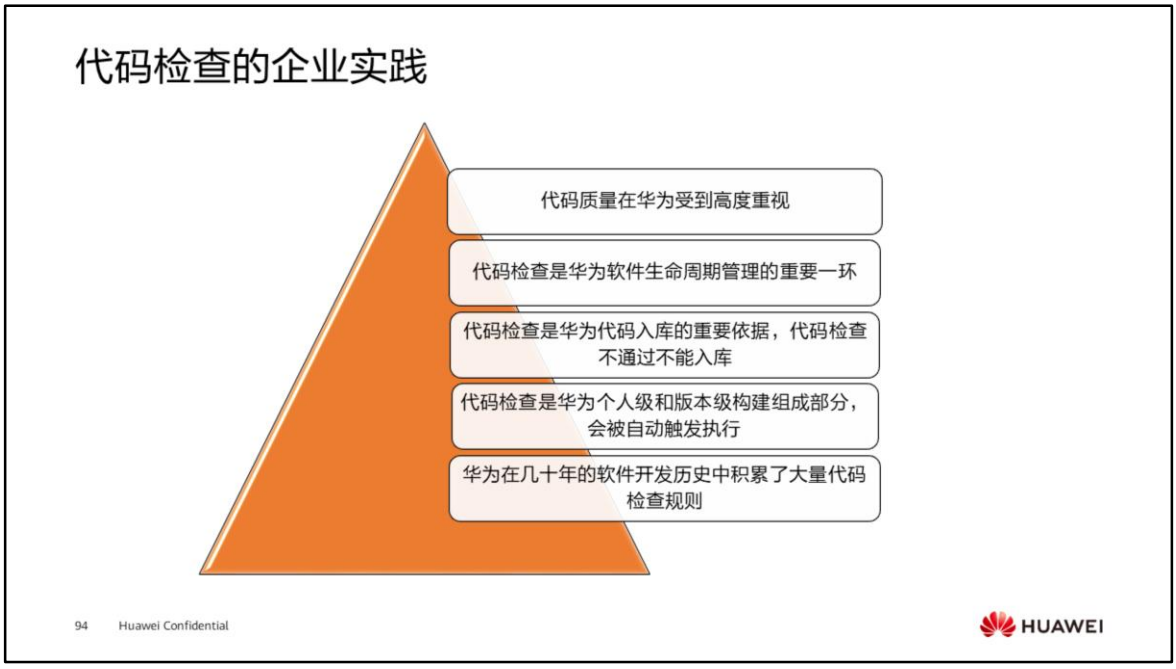
编号	分析技术	概述
1	词法分析	从左至右一个字符一个字符的读入源程序，对构成源程序的字符流进行扫描，通过使用正则表达式匹配方法将源代码转换为等价的符号（Token）流，生成相关符号列表
2	语法分析	判断源程序结构上是否正确，通过使用上下文无关语法将相关符号整理为语法树
3	数据流分析	对控制流图进行遍历，记录变量的初始化点和引用点，保存切片相关数据信息
4	无效代码分析	根据控制流图可分析孤立的节点部分为无效代码
5	抽象语法树分析	将程序组织成树形结构，树中相关节点代表了程序中的相关代码
6	语义分析	对结构上正确的源程序进行上下文有关性质的审查
7	控制流分析	生成有向控制流图，用节点表示基本代码块，节点间的有向边代表控制流路径，反向边表示可能存在的循环；还可生成函数调用关系图，表示函数间的嵌套关系
8	污点分析	基于数据流图判断源代码中哪些变量可能受到攻击，是验证程序输入、识别代码表达缺陷的关键

HUAWEI

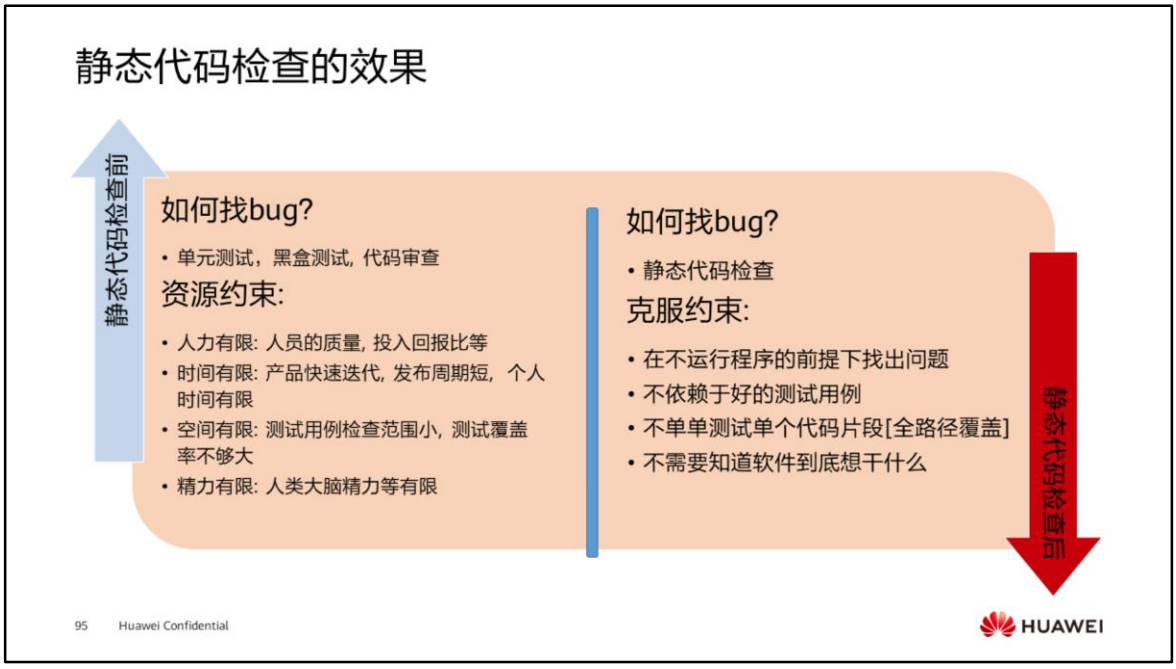
代码检查的常用工具

编号	检查工具	概述
1	Splint	Splint是开源的静态软件缺陷检测工具，用于检测用标准C实现的软件缺陷
2	Klocwork K8	支持 C/C++，JAVA，它能检测缓冲区溢出、内存泄露、安全漏洞等软件缺陷
3	Coverity	第一个能够快速、准确分析当今的大规模（几百万、甚至几千万行的代码）、高复杂度代码的工具。检测和解决C、C++、Java和C#源代码中最严重的缺陷的领先的自动化工具
4	Findbugs	是一个基于 java 语言的静态分析工具，它主要检查类或者 JAR 文件
5	PMD	采用 BSD 协议发布的 Java 程序代码检查工具，其核心是 javacc解析器
6	Cppcheck	一种开源的 C/C++代码缺陷静态检查工具，只检查编译器检查不出来的 bug，不检查语法错误

HUAWEI



- 代码检查在华为当中的一些实践：
  - 代码检查已经被集成到华为的软件生命周期当中，作为其中的重要一环存在；
  - 不通过代码检查就无法合入代码仓库；
  - 在个人级和版本级的流水线当中，都会被自动触发执行；
  - 通过多年的开发经验，现已积累了大量的代码检查规则，并将这些经验能力提供至华为云DevCloud平台的CodeCheck当中。



- 静态代码检查前我们通过单元测试，黑盒测试, 代码审查 等方法进行代码检查，这样做会有一些资源上的约束，比如：
  - 人力上的约束: 对测试人员的能力要求, 人员、精力的投入带来的回报是否可以满足期望；
  - 时间有限: 产品快速迭代, 发布周期短，个人时间有限；
  - 空间有限: 测试用例检查范围小, 测试覆盖率不够大；
  - 精力有限: 人类大脑精力等有限。
- 静态代码检查后:
  - 静态代码检查
  - 克服约束:
    - 在不运行程序的前提下找出问题；
    - 不依赖于好的测试用例；
    - 不单单测试单个代码片段[全路径覆盖；
    - 不需要知道软件到底想干什么。



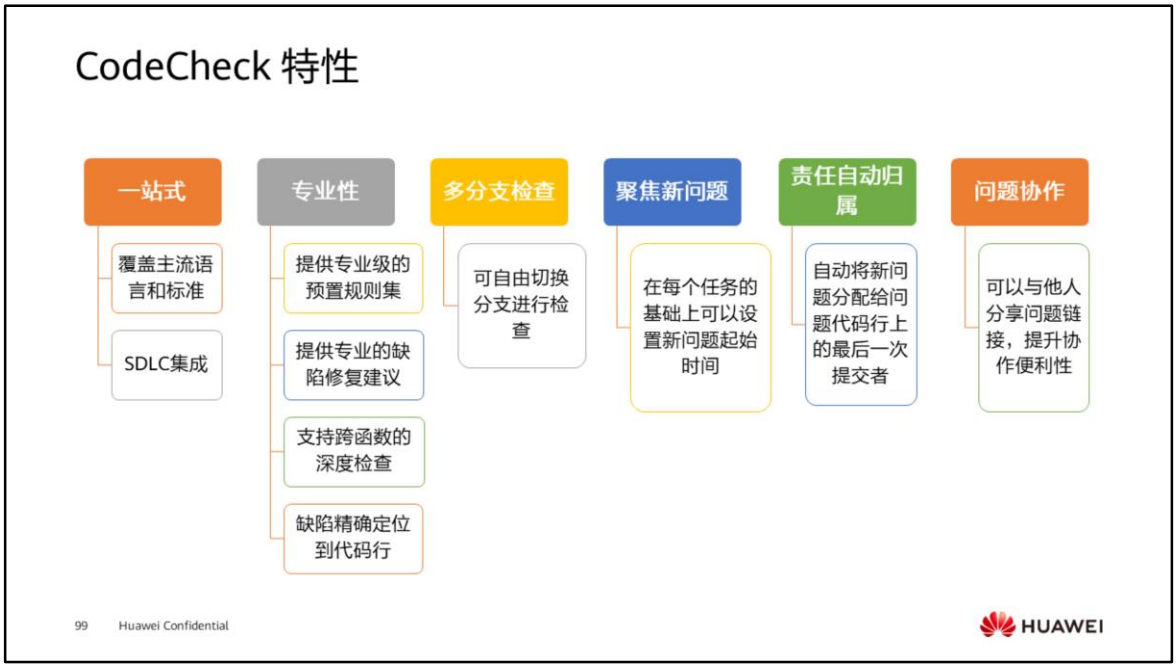
- CodeCheck当中已支持了下列的语言种类。支持混合语言检查。
- 图标来源：
  - http://findbugs.sourceforge.net/
  - https://www.logolynx.com/topic/pmd
  - https://www.freeiconspng.com/images/c-logo-icon
  - https://dev.to/psyked/eslint-visual-studio-code-editor-integration-for-the-win-jb8



- CodeCheck是根据规则集进行代码检查的，对于现有的代码检查工具而言，用户想要自定义规则集、对规则集进行集中管理、以及共享，都是较为困难的。
- CodeCheck规则集的则提供了这方面比较优秀的实践。自定义规则集、集中管理并共享规则集，在CodeCheck中都得以实现。



- Others：
  - 误报率高
  - 误报优化难
  - 误报问题重复报
- CodeCheck：
  - 支持跨函数的深度检查
  - 标记误报后不会再报
  - 预置规则集误报率低
  - 部分误报持续优化
- 上图中左半部分的代码片段，是一处（跨函数）的代码错误，在普通代码检查工具当中，非常容易漏报。在CodeCheck当中，则几乎不会出现这种现象。
- 右半部分是CodeCheck检查到的一处代码问题，如果在实际开发过程当中，认为这种问题可以忽略，则直接标记，之后则不会再对此处问题进行报错提示。



- 基本功能介绍
  - 一站式：
    - 覆盖 9 种主流语言，5种主流标准；
    - SDLC集成：提交代码时、合并代码时、触发了流水线当中的静态代码检查环节。
  - 专业性：
    - 华为30年经验累积，提供专业级的预置规则集；
    - 提供专业的缺陷修复建议，知道代码质量改进答复；
    - 支持跨函数的深度检查；
    - 缺陷精确定位到代码行。

- 高级特性
  - 多分支检查：
    - 在任务查看页面可以看到“已检查”和“未检查”的分支，可自由切换分支进行检查。
  - 聚焦新问题：
    - 在每个任务的基础上可以设置新问题起始时间。默认起始时间为当前任务上一次检查成功的时间，起始时间之后检查出的所有问题属于新问题。
  - 问题责任人自动归属：
    - 静态分析过程中，自动将新问题分配给问题代码行上的最后一次提交者。
  - 问题协作：
    - 可以与他人分享问题链接，提升协作便利性。

CodeCheck 产品典型操作流程



HUAWEI

创建代码检查任务

- 创建代码检查任务主要包含：
  1. 选择代码仓库
  2. 选择检查语言
  3. 完成任务创建



HUAWEI

执行代码检查任务

- 执行代码检查任务主要包含：
  1. 任务主页点击开始检查
  2. 等待任务检查完成（查看检查进度）
  3. 查看检查结果



HUAWEI

- 在CodeCheck当中，检查出的问题可以直接定位到代码片段，并对检查出的问题进行相应的描述。

执行代码检查任务

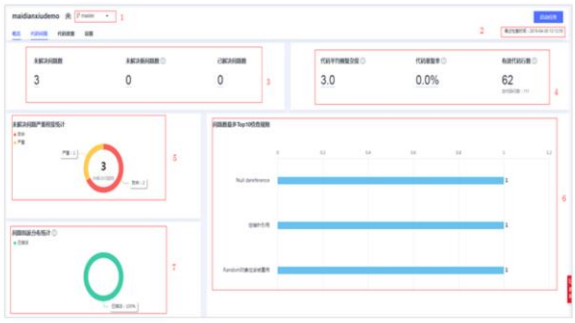
- 执行代码检查任务主要包含：
  1. 任务主页点击开始检查
  2. 等待任务检查完成（查看检查进度）
  3. 查看检查结果



HUAWEI

任务详情 - 概览视图

- 检查任务概览页面，主要包含以下内容：
  - 任务当前分支：默认为master分支
  - 任务最近检查时间
  - 未检查问题数、未解决新问题数、已解决问题数
  - 代码平均圈复杂度、代码重复率、有效代码行数
  - 未解决问题严重程度统计
  - 问题最多Top 10检查规则
  - 问题指派分布统计



HUAWEI

任务详情 - 问题视图

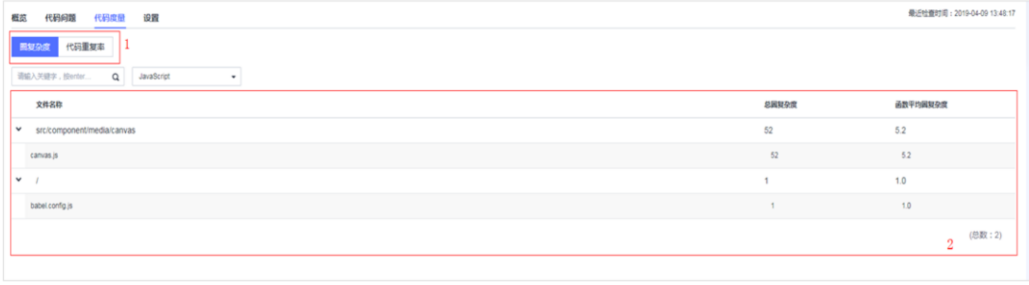
- 检查任务问题页面，主要包含以下内容：
  1. 过滤器功能：提供按问题级别、问题状态、问题检测时间、规则、文件目录、负责人过滤
  2. 单个问题卡片：包含问题所在文件、问题报错信息、问题级别、状态、负责人、修改建议等
  3. 问题可操作区域： 可手动更新问题状态 | 问题负责人、查看问题修改建议等
  4. 问题关键代码片段



HUAWEI

任务详情 - 代码度量视图

- 检查任务代码度量页面，主要包含以下内容：
  1. 代码度量指标（可切换过滤）
  2. 对应指标下的具体数据列表



HUAWEI

任务详情 - 设置视图

1. 任务名称、检查语言等的更新
2. 选择代码检查任务的规则集
3. 自定义执行计划
4. 自定义任务通知（比如检查完成时通知）
5. 自定义新问题起始时间等
6. 删除当前任务



HUAWEI

思考题

1. 以下对于Git Rebase 和 Merge 两种合并方式的区别，哪些选项描述是正确的?

A. Merge 是一个合并操作，会将两个分支的修改合并在一起，默认操作的情况下会提交合并中修改的内容

B. Rebase 的提交历史忠实地记录了实际发生过什么，关注点在真实的提交历史上面

C. Rebase 并没有进行合并操作，只是提取了当前分支的修改，将其复制在了目标分支的最新提交后面

D. Merge 的提交历史反映了项目过程中发生了什么，关注点在开发过程上面

HUAWEI

- 参考答案：AC

本章总结

- 介绍持续集成理念及基本原则，分析持续集成给软件项目带来的价值
- 详细讲解当前主流的分支策略，并对其适用的场景做逐一分析
- 详细介绍 Git 工具的基本操作及其原理
- 介绍代码提交、代码评审的过程，包括代码仓创建，代码编写，编译，调试以及入库和分支合并以及静态检查，分支合并等过程。
- 详细讲解DevCloud的代码托管服务CodeHub及其安全、权限、分支等场景；以及CloudIDE及其使用场景和跟业界的特性对比；
- 详细讲解静态检查的类型、功能以及对应的DevCloud静态检查服务CodeCheck的优势特性：多语言检查、规则集管理、问题可视化、统一报表管理等多功能

HUAWEI

更多信息

- 华为云官网：
  - https://www.huaweicloud.com/
- 华为云软件开发平台 DevCloud：
  - https://www.huaweicloud.com/devcloud/
- 华为云帮助中心：
  - https://support.huaweicloud.com/
- 华为云学院：
  - https://edu.huaweicloud.com/

HUAWEI

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and organization for a fully connected, intelligent world.

Copyright©2020 Huawei Technologies Co., Ltd.
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.



持续测试与反馈



HUAWEI

前言

- 本文对持续测试在软件项目开发中的应用进行了研究,提出了软件持续测试时遇到的关键问题并给出相应的解决思路。讨论了软件持续测试可以采用的辅助工具及相应的使用方法,给出了工具软件的配置方案，并说明了工具软件之间的相互关系和协作方法。

HUAWEI

目标

- 了解敏捷测试的重要性和基本理念
- 掌握敏捷测试概念、流程和方法
- 掌握华为云的测试能力，例如测试管理、接口测试、性能测试的基本操作
- 学会如何应用敏捷测试，保持项目整体测试的高质量高效率

HUAWEI

目录

1. **敏捷软件测试理念、方法与实践**
2. 测试管理
3. 常见的测试方法
4. 测试度量指标体系和质量评估

HUAWEI

测试概念及分类

- 什么是测试（Wikipedia）
- 在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。

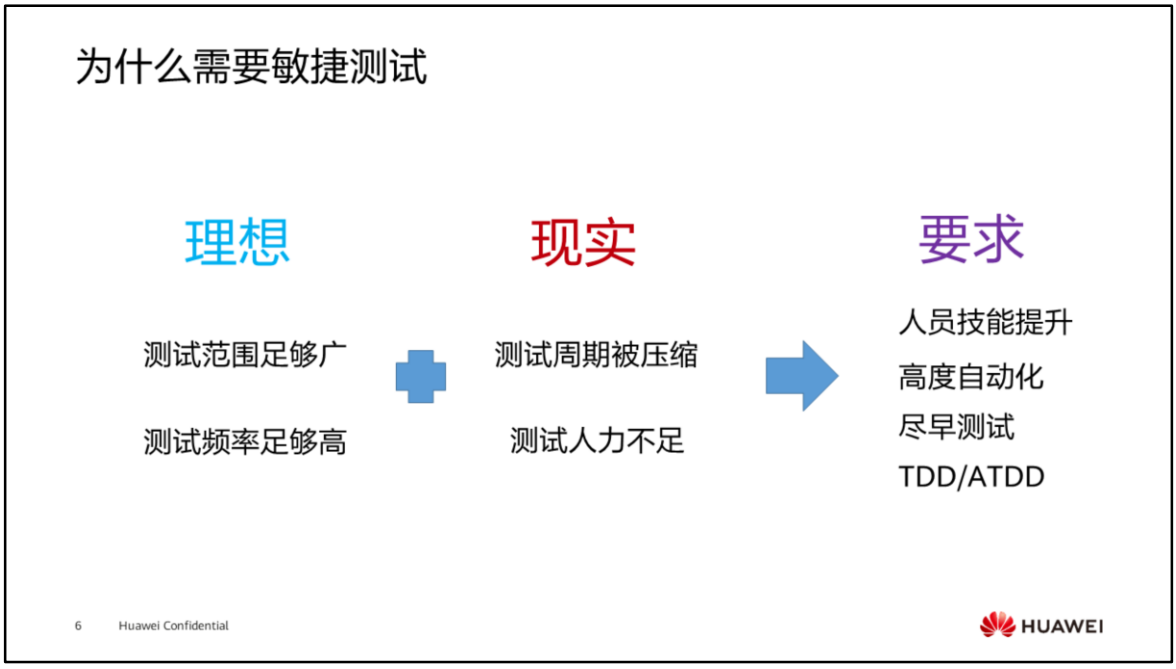
测试分类



HUAWEI

- 黑箱测试（black-box testing），是软件测试方法，测试应用程序的功能，而不是其内部结构或运作。测试者不需具备应用程序的代码、内部结构和编程语言的专门知识。测试者只需知道什么是系统应该做的事，即当键入一个特定的输入，可得到一定的输出。测试案例是依应用系统应该做的功能，照规范、规格或要求等设计。测试者选择有效输入和无效输入来验证是否正确的输出。此测试方法可适合大部分的软件测试，例如集成测试（integration testing）以及系统测试（system testing）。
- 白箱测试（white-box testing，又称透明盒测试glass box testing、结构测试structural testing等）是一个测试软件的方法，测试应用程序的内部结构或运作，而不是测试应用程序的功能（即黑盒测试）。在白盒测试时，以编程语言的角度来设计测试案例。测试者输入数据验证数据流在程序中的流动路径，并确定适当的输出，类似测试电路中的节点。可以应用于单元测试（unit testing）、集成测试（integration testing）和系统的软件测试流程，可测试在集成过程中每一单元之间的路径，或者主系统跟子系统中的测试。尽管这种测试的方法可以发现许多的错误或问题，它可能无法检测未使用部分的规范。
- 功能测试是按照软件的各个功能划分，进行有条理的测试。在功能测试时要保证测试项覆盖了所有的功能，以及各种条件的组合。

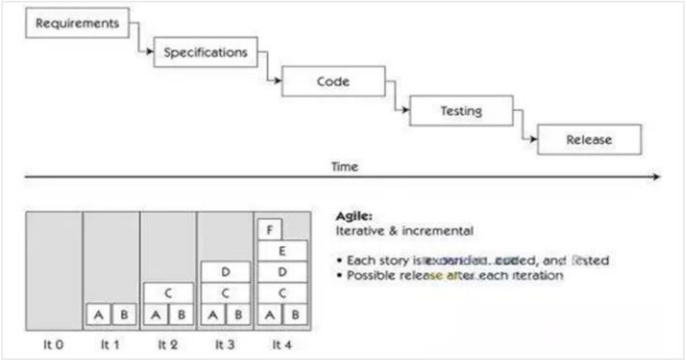
- 系统测试，一个完整的软件的整体角度来进行测试。系统测试和功能测试的主要区别是系统测试要利用所有真实的数据来进行测试，模拟真实的用户，同时关注点是整个软件系统，而不是单点的功能。
- 极限值测试是软件在各种特殊条件、特殊环境下能否能正常运行，进行测试。特殊条件一般指软件规定的最大值。最小值，以及在超过最大值超过最小值的情况下，进行一个测试。特殊环境一般是指软件运行的机器处于CPU、高负荷或者是网络高负荷的状态下进行测试。
- 性能测试是对软件性能的评价，简单的说就是衡量软件具有的响应能力。因此性能测试时采用测试手段，对软件的响应的及时性进行评价的一种方式。
- 单元测试是针对程序设计中的每一个单点函数进行测试的过程。
- 集成测试关注模块与模块之间的交互。
- 系统测试是端对端的，对整个系统进行测试，使用真实的用户场景和数据。
- 回归测试是验证新加的代码，是否会对现有的功能产生影响，确保现有功能正常可用。



- 理想情况下测试范围应该足够广，频率应该足够高，但这只是理想情况。项目执行中实际情况往往是因为赶进度，测试周期被压缩，同时测试人力也不足，这就要求测试人员技能必须提升，以当时同时通过高度的自动化、测试、左移等方法，尽早的发现问题，要做到这些就要开展敏捷测试。

敏捷软件测试理念

- 什么是敏捷测试（Wikipedia）？
- 敏捷测试是一种遵循敏捷软件开发原理的软件测试实践。敏捷测试涉及跨职能敏捷团队的所有成员，这些团队具有测试人员的特殊专业知识，以确保以可持续的速度定期提供客户期望的业务价值。
- 注重“以人为本”，快速迭代。



HUAWEI

- 敏捷测试和敏捷开发密切相关，讲究以人为本，以可持续的速度进行快速迭代。强调的是可持续敏捷不仅仅是速度快，更强调在快速的同时要保持稳定的质量。

敏捷测试宣言

测试是一个活动 胜于 测试是一个阶段
预防缺陷 胜于 发现缺陷
做测试者 胜于 做检查者
帮助构建最好的系统 胜于 破坏系统
团队为质量负责 胜于 测试为质量负责

HUAWEI

- 在敏捷测试宣言中可以发现，敏捷测试和传统测试有很多的不同之处。

敏捷测试原则

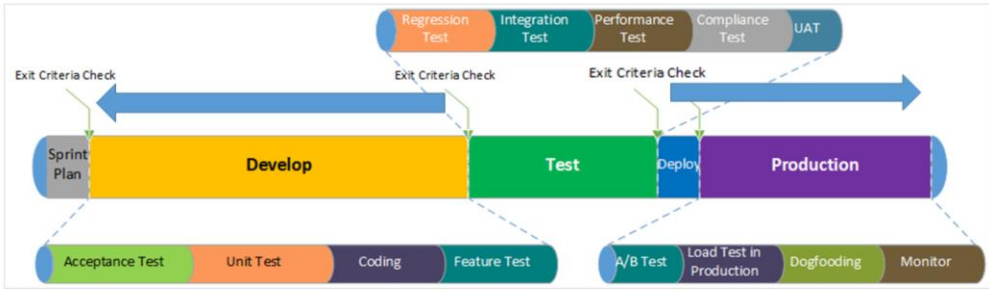
- 原则1：预防缺陷胜于发现缺陷
- 原则2：通过持续测试实现快速与高质量
- 原则3：成为全栈测试人员
- 原则4：从质量保证专项质量协助
- 原则5：渴望持续学习

Who	DevOps强调团队合作和紧密沟通，团队中每个成员都需要为质量负责。测试不再仅仅是测试人员的工作。开发人员和运维人员也要承担测试任务。
When	测试工作贯穿始终，从产品准备初期到上线之后。
Where	测试工作无处不在，一切能提高产品质量的工作就是测试工作。

HUAWEI

- 敏捷测试的5项原则则进一步解释了敏捷测试的特点，包括三个方面，第一，团队每个成员都要对质量负责，同时测试人员自身能力也要提升，第二，时间测试工作贯穿始终。第三，一切能提高产品质量的工作，就是测试工作。

测试左移和测试右移



- 测试左移，尽早开展测试
- 使用TDD、ATDD和BDD
- 通过契约测试，解耦前后端和服务间开发依赖
- 加强开发自测试

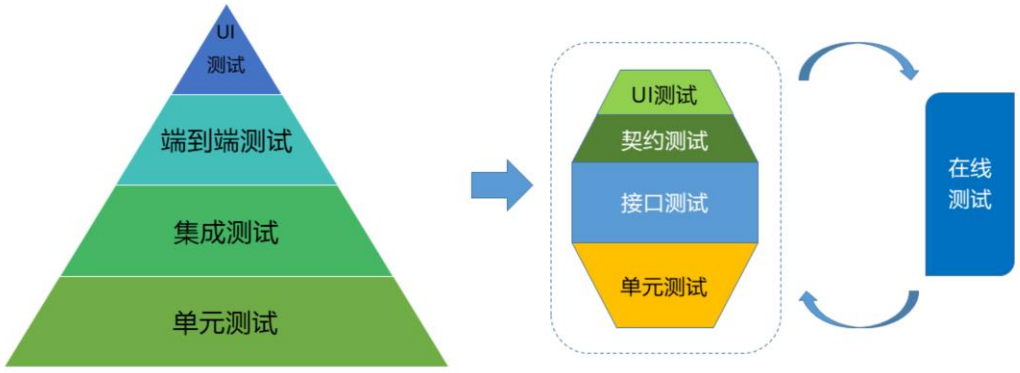
- 测试右移，覆盖发布和线上运行阶段
- 灰度测试
- 线上应用性能监控
- 线上拨测

HUAWEI

- TDD: Test Driven Development
- ATDD: Acceptance Test Driven Development
- BDD: Behavior Driven Development
- 具体要如何开展敏捷测试，通过测试的左移和右移，可以有效地减少测试时间，提升测试的效率。
- 左移测试所宜鼓励开发者自测试，在开发阶段就尽早地开展测试，使用测试驱动开发行为驱动开发，强制在编码之前就充分的考虑测试，通过契约测试和默克机制，解耦服务间的依赖，提升测试速度和稳定性。
- 测试右移，在发布阶段和线上阶段进行测试，测试类型主要包括灰度测试，在监控测试类型主要包括灰度测试、在线播测以及在线监控。
- 灰度测试是指灰度发布过程中进行的测试。
- 在线波测试一种主动监控，持续高频的对被测系统常见功能进行测试，在线监控，所以各种系统指标为核心进行被动监控。这一测试，在真实的环境中快速发现问题，快速反馈并闭环问题。

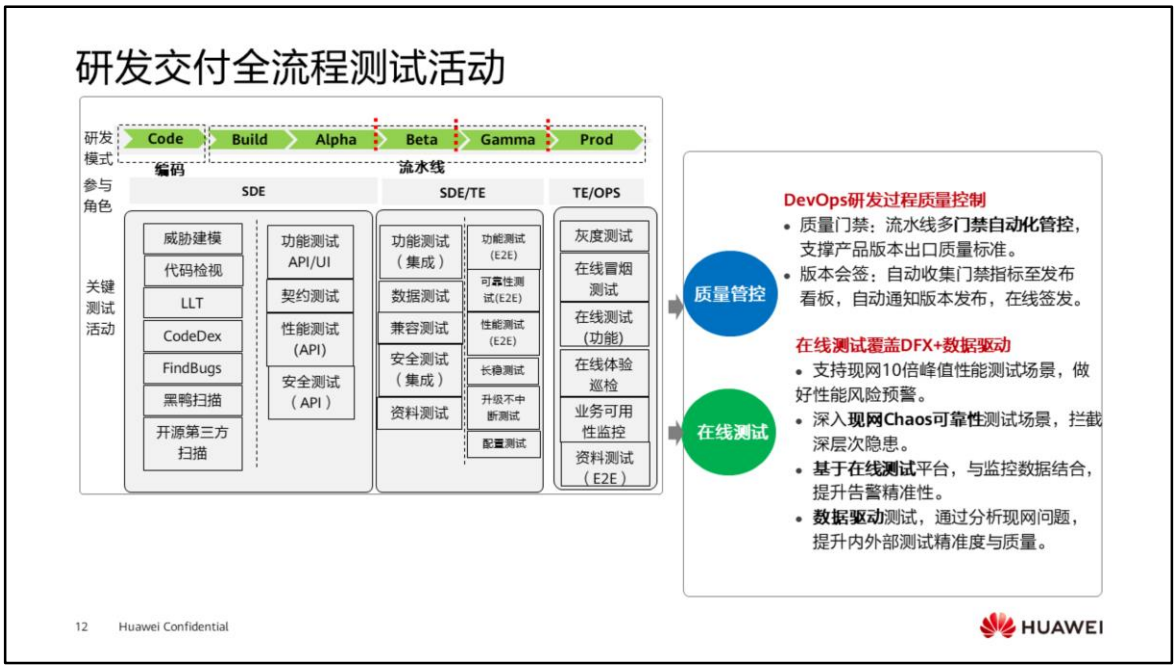
测试分层模型演进

- 由金字塔模型向纺锤模型的演进，减少单元测试，增强接口测试、契约测试、在线测试



HUAWEI

- 测试人员对测试的金字塔模型应该都不陌生，传统的金字塔中单元测试的比重比较大，在能力比较成熟，项目时间比较充足的团队中，这样做当然是没有问题的。
- 但实操中单元测试的写作难度和投入时间通常都比较大，这种情况下可以考虑由金字塔模型向纺锤型模型的演进，适当减少单元测试的数量，同时增加接口测试的数量，这样投入产出比可能会更高。



- SDE：Software Design Engineer
- TE： Test Engineer
- 测试活动将会贯穿研发的全流程，在流水线进行构建的时候，执行单元测试和一些静态大模扫描和一些静态蛋白扫描，这些测试通常由开发人员来开展，在开发自测试环境中，执行接口测试、性能测试和接口的安全测试，同样由开发人员开展，在集成测试、环境执行、系统级的功能测试、性能测试以及可靠性等测试，根据团队的组成和成熟程度开放测试，均可以开展在生产环境执行灰度测试，在线帽檐测试，在线播侧重各种在线测试。

目录

1. 敏捷软件测试理念、方法与实践
2. **测试管理**
3. 常见的测试方法
4. 测试度量指标体系和质量评估

HUAWEI

完整的测试过程

测试策略	测试设计	测试执行	测试报告
• 明确测试范围 • 制定测试计划 • 明确测试目标 • 组建测试团队 • 准备工具、环境	• 建立测试模型 • 设计用例 • 开发自动化脚本	• 回归测试 • 新特性验证 • 结果日志分析 • 提交缺陷	• 输出报告 • 评价版本质量 • 分析风险









HUAWEI

- 测试管理是对测试过程进行全生命周期的管理，那么一个完整的测试过程，包括测试策略、测试设计、测试执行和测试报告。

测试策略5W1H法

- 什么是测试策略：
- 测试策略描述测试工程的总体方法和目标。描述目前在进行哪一阶段的测试以及每个阶段内在进行的测试种类（功能测试、性能测试、覆盖测试等）以及测试人力安排等。

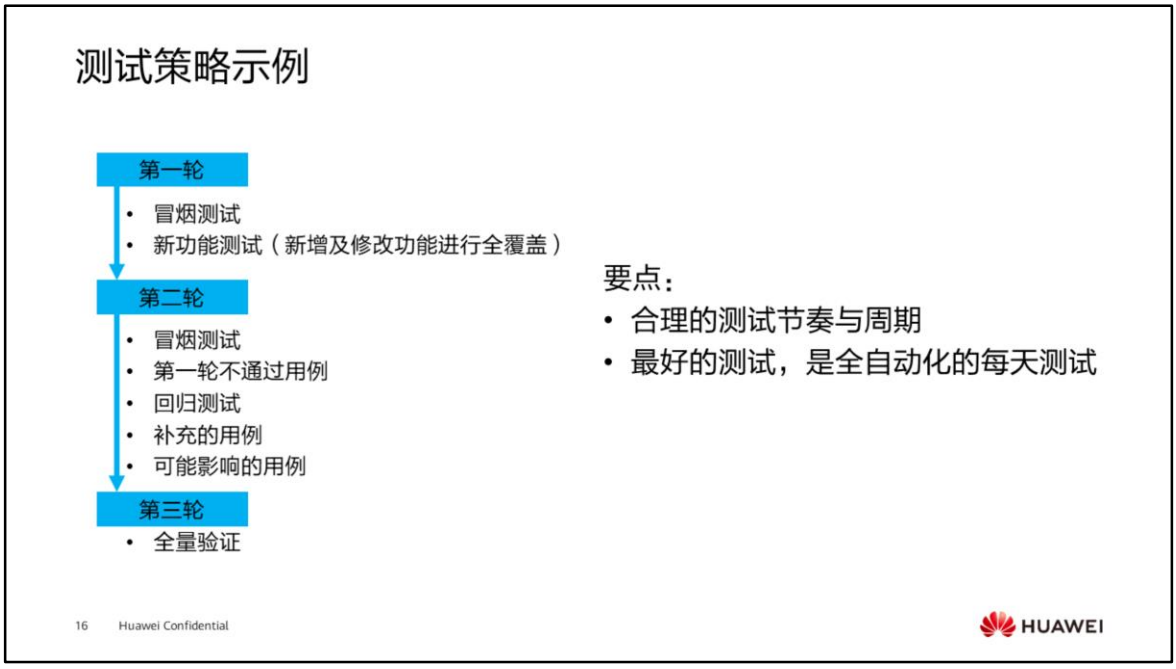


5W1H法

- **Why** 为什么要进行测试，测试目的
- **What** 测试的内容及范围，测哪些，确定测试重点（RBT等）
- **When** 测试的起止时间，考虑影响时间的因素
- **Where** 相关文档的存放位置，缺陷的存放，环境地址
- **Who** 测试人员安排
- **How** 选用何种工具及方法进行测试

HUAWEI

- RBT：Requirement Based Testing, 基于需求的测试



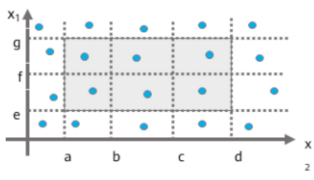
- 测试策略例子中，总共有三轮测试，第一轮进行冒烟测试以及新功能的测试。第二轮继续进行冒烟测试，以及一些回归测试，到了第三轮主要开展全量测试，进行测试的收尾，需要合理的安排测试的节奏和周期，最好是每天都通过自动化的方式进行测试。



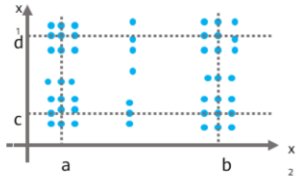
- 在云测中，支持新建测试计划，支持引入需求，支持对计划的进度进行展示。

测试设计常用方法

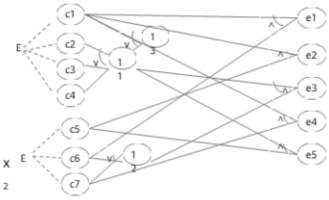
- 什么是测试设计?
  - 将软件需求转换成测试需求，并最终形成测试用例的过程
- 测试设计常用方法：



等价类划分法



边界值分析法



因果图法

其他方法：
正交表
场景分析
场景库，
场景因子
异常测试
步骤异常
数据异常
探索式测试
导流测试
.....

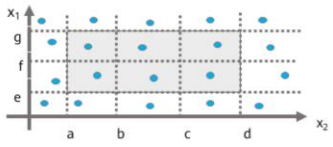
HUAWEI

- 常见的测试设计方法有等价类、边界值、因果图，还包括更多的正交表、场景分析等。

等价类划分法

- 等价类划分法是把所有可能输入的数据，即程序的输入域划分策划国内若干部分（子集），然后从每一个子集中选取少数具有代表性的数据作为测试用例。

- 案例：



等价类划分法

测试计算平方根的函数-输入：实数-输出：实数-需求说明：当输入一个0或比0大的数的时候，返回其正平方根；当输入一个小于0的数时，显示错误信息”平方根非法-输入值小于0”并返回0；库函数Print-Line可以用来输出错误信息。

等价类划分：
I.可以考虑作出如下划分：
a、输入 (i)<0 和 (ii)>=0
b、输出 (a)>=0 和 (b) Error
II.测试用例有两个： a、输入4，输出2。对应于 (ii) 和 (a) 。 b、输入-10，输出0和错误提示。对应于 (i) 和 (b) 。

HUAWEI

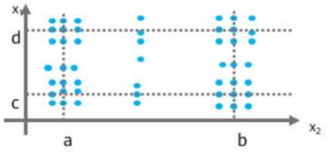
- 什么是等价类呢？等价类划分是把所有可能的输入数据划分成若干个子集，之所以叫等价，那么代表着每个子集是有代表性的。
- 在这一个例子中，是测试计算平方根的函数，输入是一个实数，输出也是个实数。当输入0或比0大的时候，返回其正数的平方根，当输入一个小于0的时候，展示错误信息，输入小于0，考虑从等价类角度进行以下的划分，输入小于0和大于0，输出则是大于0和错误，通过这个思想，可以设计两个测试用例，输入大于0的，对应第一种情况，输入小于0的对应第二种情况。

边界值分析法

- 边界值分析法就是对输入或输出的边界值进行测试的一种黑盒测试方法。通常边界值分析法是作为对等价类划分法的补充，这种情况下，其测试用例来自等价类的边界。

测试计算平方根的函数-输入：实数-输出：实数-需求说明：当输入一个0或比0大的数的时候，返回其正平方根；当输入一个小于0的数时，显示错误信息”平方根非法-输入值小于0”并返回0；库函数Print-Line可以用来输出错误信息。

- 案例：



边界值分析法

边界值分析：
划分(ii)的边界为0和最大正实数；
划分(i)的边界为最小负实数和0。

由此得到以下测试用例：a、输入 {最小负实数}b、输入 {绝对值很小的负数}c、输入 0d、输入 {绝对值很小的正数}e、输入 {最大正实数}

HUAWEI

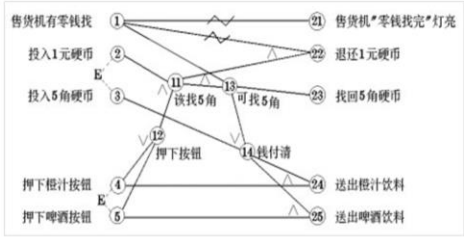
- 边界值分析是对等价类的一个补充。
- 通常其测试用例是来自等价类的边界，在上述的例子里，只划分了两个用例。那么通过边界值的分析，可以得知，边界是0和最大的实数，同样还有一个边界是最小的，负数和0。由此可以得到以下的用例。第一个，输入一个最小的负实数，第二个，输入一个绝对值很小的负数，由此可以得到五个用例，最小的负实数、绝对值很小的负数，0、绝对值很小的正数，以及最大的正数。

因果图法

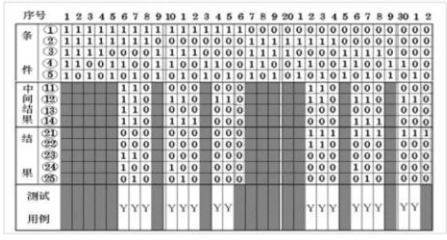
- 利用图解法分析输入的各种组合情况，从而设计测试用例的方法，它适合于检查程序输入条件的各种组合情况。
- 案例：

自动售货机售卖饮料测试。其规格说明如下：若投入5角钱或1元钱的硬币，押下〖橙汁〗或〖啤酒〗的按钮，则相应的饮料就送出来。若售货机没有零钱找，则一个显示〖零钱找完〗的红灯亮，这时在投入1元硬币并押下按钮后，饮料不送出来而且1元硬币也退出来；若有零钱找，则显示〖零钱找完〗的红灯灭，在送出饮料的同时退还5角硬币。

因果图



判定表

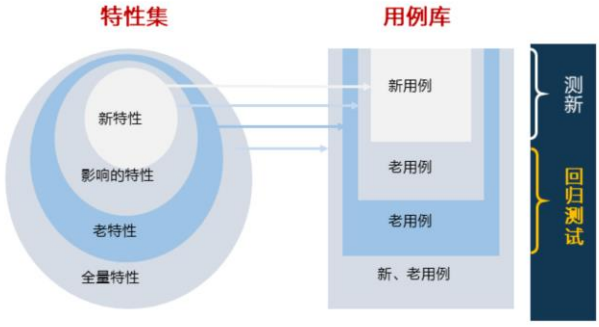


HUAWEI

- 因果图示例，关于自动售货机售卖饮料的测试，投入五角或一元硬币时，压下橙汁或啤酒的按钮，则相应的饮料送出来。如果没有零钱找，则显示零钱找完，这时候再投入一元硬币后饮料不出来，因为一元硬币也会出来，若有零钱找，则显示零钱找完，红灯灭，送出饮料，并退还五角硬币。那么从因果图可以得出以下结论，左边是因，右边是果，通过因和果的连线，得到一张判定表，黑色部分表示不可能发生的一些正交。

测试执行类型

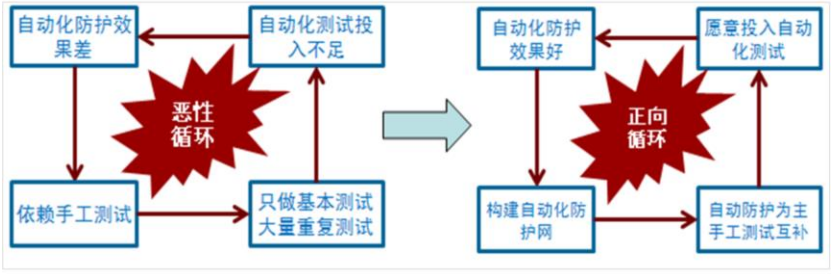
- 回归测试与新功能测试
- 回归测试：修改了旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误。
- 新功能测试：新加入测试的特性，可能对其他特性产生副作用，须进行某些程度的回归测试



HUAWEI

自动化测试目的

- 自动化测试不是为了发现问题，而是减少人力成本、降低重复工作
- 确保新特性引入不影响老特性
- 版本升级不丢特性
- 问题不会重复发生



HUAWEI

- 测试的执行往往是要通过自动化的方式来进行的。自动化测试不是为了发现问题，而是要减少人力成本，降低重复的工作。如果不进行自动化测试，那么会如何，自动化的防护效果将会很差？过度的依赖手工测试，那么将造成一个恶性循环，越来越依赖于手工测试，只能做一些最基本的大量重复的测试。而如果自动化的防护效果好，则可以进行正向循环，构建自动化的防护网，越来越快的提升测试效率。

问题单处理流程

- 什么是问题单：
  - 问题单是版本测试过程中发现的问题，也可以称作为bug、缺陷。



HUAWEI

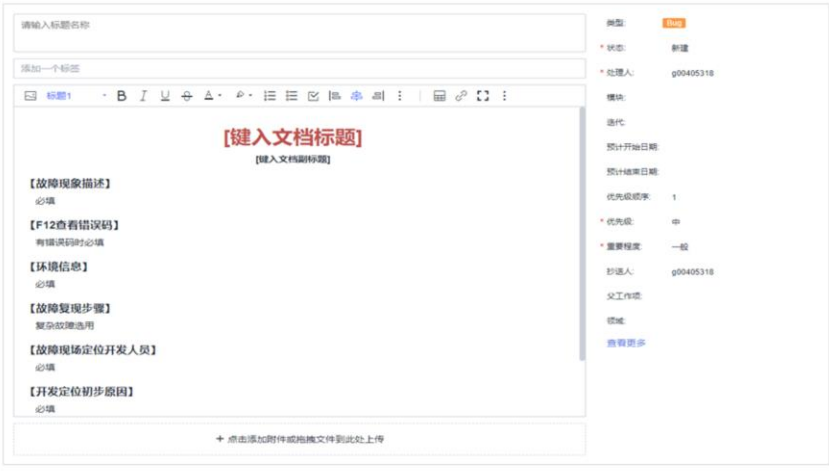
- 提单步骤 ：发现缺陷-》重现缺陷-》与开发确认-》提交问题单-》开发修复-》开发自验证-》提交版本-》测试验证-》关闭问题单
- 一个问题单应该包含的要素：
  - 问题单级别、类型
  - 问题描述
  - 根因分析
  - 处理意见
  - 测试建议
  - 关联的测试用例
  - 开发定位所需日志、截图等
  - 环境信息描述
- 执行过程中还会有问题单的处理流程，问题单是版本过程中发现的问题，也称为bug或缺陷，常见的提单步骤，涉及测试人员和开发人员、测试人员发现缺陷，重现缺陷，并联系开发人员确认缺陷，开发人员确认缺陷后，测试人员提交问题单，接下去由开发人员修复问题单，自验证并发布版本测试人员则进行验证并关闭问题单，这张图展示了云测的用例、问题单的关联以及问题单属性的展示。

云测 - 用例与问题单关联



HUAWEI

云测 - 问题单属性展示



HUAWEI

测试报告内容范围

- 测试报告用于记录测试过程，总结各轮次的测试情况，分析测试数据，归纳测试工作进行过程中暴露的问题与遗留的风险，给出相应的测试建议以供后续项目参考。

一份合格的测试报告包含的内容

- 产品版本说明
- 测试环境描述
- 测试结论
- 关键风险
- 不对外发布特性【可选】
- 受限发布特性【可选】
- 产品全量质量评价
- 产品技术指标评价
- 缺陷分析
- 覆盖率分析
- 测试策略回顾【可选】
- 测试过程评估【可选】
- 测试结果统计数据
- 遗留问题清单
- 其他

HUAWEI

- 执行之后，进入了测试报告的阶段，测试报告是记录了整个测试的过程，对测试过程进行总结归纳，一份合格的测试报告应该包含版本的说明环境描述、测试的结论、风险、对产品质量的评价、对技术指标的评价、对缺陷的分析、对覆盖率的分析测试结果进行各种统计，以及遗留问题的清单。

目录

1. 敏捷软件测试理念、方法与实践
2. 测试管理
3. **常见的测试方法**
   - **接口测试**
   - 性能测试
   - 契约测试（Mock测试）
   - WebUI测试
   - 可靠性与安全测试
   - 在线测试
4. 测试度量指标体系和质量评估

HUAWEI

API接口测试

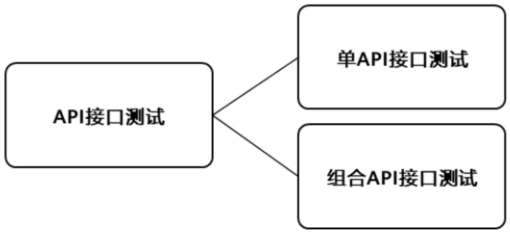
- API接口测试是测试系统组件间接口的一种测试，主要用于测试系统与外部其他系统之间的接口，以及系统内部各个子模块之间的接口。
- API接口测试重点：
  1. 检查接口参数传递的正确性。
  2. 接口功能实现的正确性。
  3. 输出结果的正确性。
  4. 对各种异常情况的容错处理的完整性和合理性。

HUAWEI

- API接口测试是测试系统组件间接口的一种测试，包含两种测试场景：一种是系统与外部其他系统之间的接口，另一种是系统内部各个子模块俗称子系统之间的接口。
- API接口测试有如下四个测试重点。
  - 检查接口参数传递的正确性
  - 接口功能实现的正确性
  - 输出结果的正确性
  - 对各种异常情况的容错处理的完整性和合理性

API接口测试分类

- 单API接口测试：主要关注参数取值和参数取值组合。
- 组合API接口测试：主要关注功能的完整性和正确性：
  - 完整性：完成一个特定的功能。
  - 正确性：执行结果正确，满足测试要求。



HUAWEI

- API接口测试分为单API接口测试、和组合API接口测试。
  - 单API接口测试是针对单个API接口的测试场景，主要关注单个功能的各种取值以及多个参数的取值的组合;
  - 组合API接口测试，是把多个API的逻辑串联起来测试。一个业务逻辑也经常是要需要通过调用多个API才能实现，所以组合API接口测试是比较贴切业务功能的，主要关注功能的完整性和正确性，完整性是指要完成一个特定的功能，正确性是指通过组合调用多个接口完成一个业务逻辑后还要关注执行结果的正确满足业务测试要求。

API接口测试面临的挑战

当前微服务迭代周期短、版本多，服务需要具备独立测试和快速验证能力，支撑单服务time to test测试耗时缩短以及测试活动前移。面临如下挑战：

- **服务不具备独立验证能力**

传统的技术架构使大量测试活动需要在集成测试等阶段进行验证，亟需落地服务独立测试能力，通过测试活动前移，提前发现问题，提升版本质量。

- **自动化用例开发效率很低**

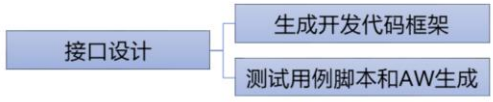
传统的技术框架自动化用例开发效率低，无法满足用例快速增长需求，自动化用例开发存在效率瓶颈。

HUAWEI

- API接口测试面临的挑战：随着微服务架构的流行，微服务开发迭代周期短，版本多，单个微服务需要具备独立测试和快速验证的能力，以支持测试活动前移。
- 目前主要面临如下两个：
  - 挑战一，服务不具备独立验证的能力、传统的技术架构使大量测试活动在集成测试等阶段进行验证而微服务需要独立测试能力，通过测试活动前移提前发现问题从而保证版本质量。
  - 挑战二，自动化用例开发效率很低。在传统的技术框架下自动化用例开发效率低无法满足用例快速增长的需求。

API接口测试的三原则

- 同源： 同源原则就是设计、开发、测试三个活动基于同一源头开展



- 独立测试： DevOps模式下，50%+服务涉及服务依赖，服务自测试前移到开发自测试阶段开展，创建Mock服务使环境解耦，否则微服务独立测试难以开展
- 100%自动化测试：在开发自测试阶段，开发人员时间有限，为了提高转测效率应该尽量高效使用自动化，也就是说能自动化解决的工作尽量自动化解决，就是我们倡议的100%自动化

HUAWEI

- API接口测试需要满足的三个原则：
  - 第一，同源是指设计开发、测试三个活动，基于同一个源头开展，最好是基于完备的API定义文档，API定义文档就是我们所说的同源。
  - 第二，独立测试在当前流行的DevOps模式下，微服务需要具备独立测试的能力，而50%以上的服务都涉及到服务之间的依赖，必须通过Mock等技术才能使环境解耦，否则微服务独立测试很难开展。
  - 第三，100%的自动化测试在开发自测阶段，为了提高测试效率，快速得到测试结果，应尽量高效使用自动化测试, 倡议是100%的自动化。

API接口测试设计过程



API接口测试流程图

1、**接口梳理**：参考接口设计文档、业务流图等；

2、**API接口参数分析**：熟悉接口参数和作用；

3、**业务场景梳理**：业务场景分析、涉及的API组合、业务逻辑关联等；

4、**单接口测试设计**：单接口单参数取值、单接口多参数组合、测试输入数据设计、返回检查点设计等;

5、**多接口组合场景设计**：多接口组合调用，还原真实业务场景;

HUAWEI

- 基于接口设计文档，从功能流程、接口参数、数据遍历、组合交互四方面开展接口的功能测试分析：
  - 功能流程：分析业务功能流，从流程分支角度进行分析设计
  - 接口参数：从接口的参数组合进行分析和覆盖
  - 数据遍历：通过合理取样，完成对接口数据的覆盖遍历
  - 组合交互：多接口间交互影响，组合测试分析
- API接口测试分为如下五个步骤：
  - 一，接口梳理接口梳理指的是参考接口设计文档、业务流图等梳理出我们要测试的所有接口。
  - 二，API接口参数分析熟悉接口参数类型和每一个参数的作用。
  - 三，业务场景梳理开展业务场景分析，分析出涉及的API组合、业务逻辑之间的关联关系等。
  - 四，单接口测试设计，单接口单参数取值，单接口多参数组合，测试输入数据设计、返回检查点、设计等。
  - 五，多接口组合场景设计、多接口组合、调用、还原真实的业务场景。

API接口测试自动化流程

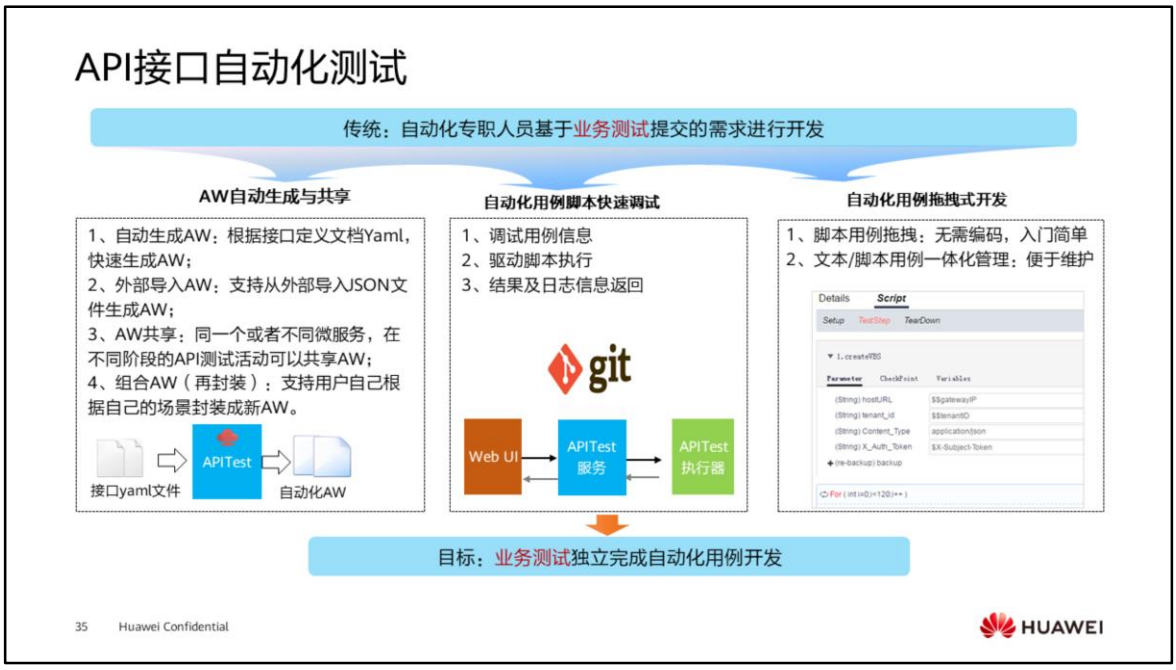
基于接口进行测试用例设计、建模，通过APItest实现接口封装调用，脚本开发，集成到流水线，实现了单服务测试。





HUAWEI

- 如下步骤：
  - 参考接口设计文件，进行接口分析；
  - 进行接口用例设计；
  - 对接口调用进行封装；
  - 开发接口用例脚本；
  - 执行接口自动化用例；

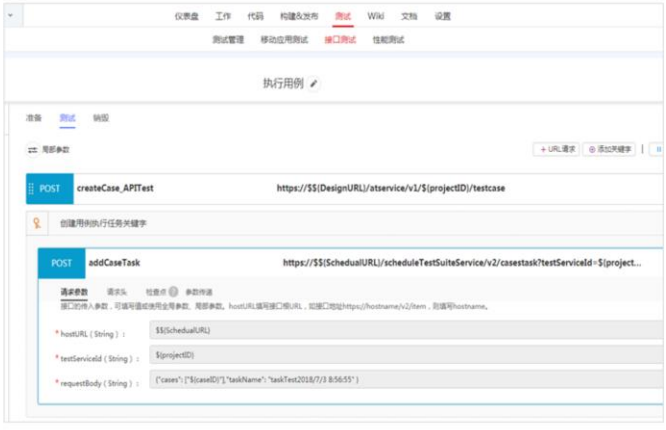


- AW: Action Word，也可理解为关键字，可复用的执行逻辑单元，方便自动化脚本扩展和复用；
- 例如：
  - 将一个API接口封装成一个AW，之后其他用例如果要调用这个API，则可以直接复用这个AW;
  - 当然我们也可以将几个API封装成一个组合AW，形成一个逻辑单元，其他用例在需要复用这块逻辑的时候，直接调用这个AW。
- 华为云在API接口自动化测试的一些优秀实践在传统模式下，API接口自动化测试是自动化专职人员基于业务测试提交的需求进行开发，而在华为云云测平台有如下优秀的技术实践：
  - 第一，AW自动生成与共享AW是指Action Word也可以理解为关键字，关键字指的是可复用的执行逻辑单元。之所以要封装AW主要是为了方便自动化脚本扩展和复用，例如，将一个API接口封装成一个AW或者叫关键字之后，其它用例如果要调用API接口，则可以直接复用AW，当然也可以将几个AW封装成一个组合AW形成一个逻辑单元。其它用例在需要复用这块逻辑的时候，直接调用AW就行了。
  - AW自动生成与共享AW自动化生成是指根据接口定义文档，或者是叫一般是YAML文档且能在不同微服务间实现AW的共享。

    - 第二，自动化用例脚本快速调试，完成自动化用例脚本开发后，可在界面上快速发起用例调试，驱动脚本执行且能可视化的展示用例的结果，和每一步的日志信息。
    - 第三，优秀的技术实践是指自动化用例拖拽式开发，指的是无需编码，在界面上拖拽AW填充参数即可实现自动化用例的开发。
- 通过以上三个关键的技术突破的目标是降低自动化用例的开发门槛，提升自动化用例开发效率，从而让业务测试能够完成自动化用例的开发和执行。

云测 - APITest接口自动化测试

使用DevCloud测试工具APITest进行接口和接口级场景功能自动化测试，结合华为优先研发实践和研发理念，测试用例免代码编写、一键导入swagger接口定义自动生成脚本模板，基于脚本模板组装编排、管理接口自动化测试用例。



简单

- 0编码开发成本
- 0环境搭建成本
- 2步接口测试
- 1步性能测试

高效

- 可视化编排测试用例
- 测试关键字批量复用
- 云端弹性资源池
- 实时反馈测试结果

专业

- HTTP等4种API协议
- 高级测试场景编排
- 7类30+预置检查点
- 多环境测试参数管理
- 流水线持续自动化测试

HUAWEI

- 华为云DevCloud云测平台里的APITest的工具的介绍，和接口及场景功能自动化测试，结合华为优秀的研发实践和研发理念，测试用例免编写代码、一键导入swagger接口定义文档可以自动生成脚本模板，基于脚本模板再组装、编排，管理接口自动化测试用例、从而实现简单、高效专业地完成API接口的自动化测试。

目录

1. 敏捷软件测试理念、方法与实践
2. 测试管理
3. **常见的测试方法**
   - 接口测试
   - **性能测试**
   - 契约测试（Mock测试）
   - WebUI测试
   - 可靠性与安全测试
   - 在线测试
4. 测试度量指标体系和质量评估

HUAWEI

性能测试很重要



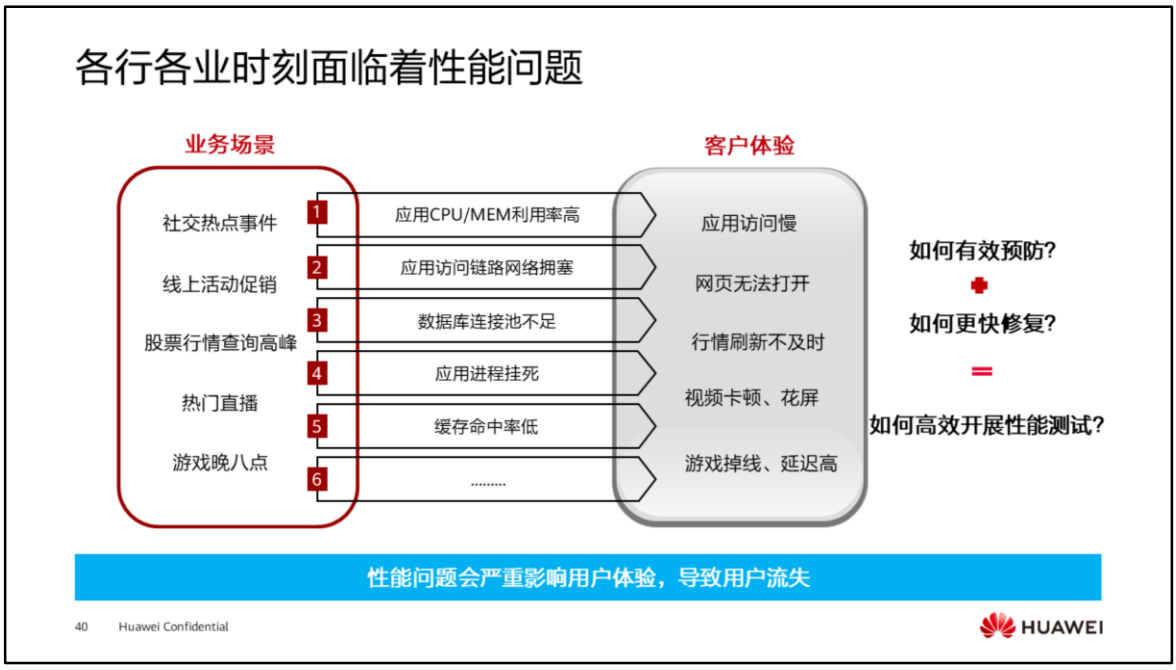
应用出现性能问题，会产生不可挽回的经济损失

- Amazon的统计显示客户访问的响应时间每延长1s，Amazon一年就会减少16亿美元销售额，首页打开时间每增加100ms，网站的销售量会减少1%
- 据估计，每年电子商务网站都会因载入速度过慢，而损失11亿~13亿美元的收入
- 响应速度在Google的PR（PageRank）评分中占有一定比例，也就是说，一个好的网站，是不可能让他的页面加载速度很慢的。

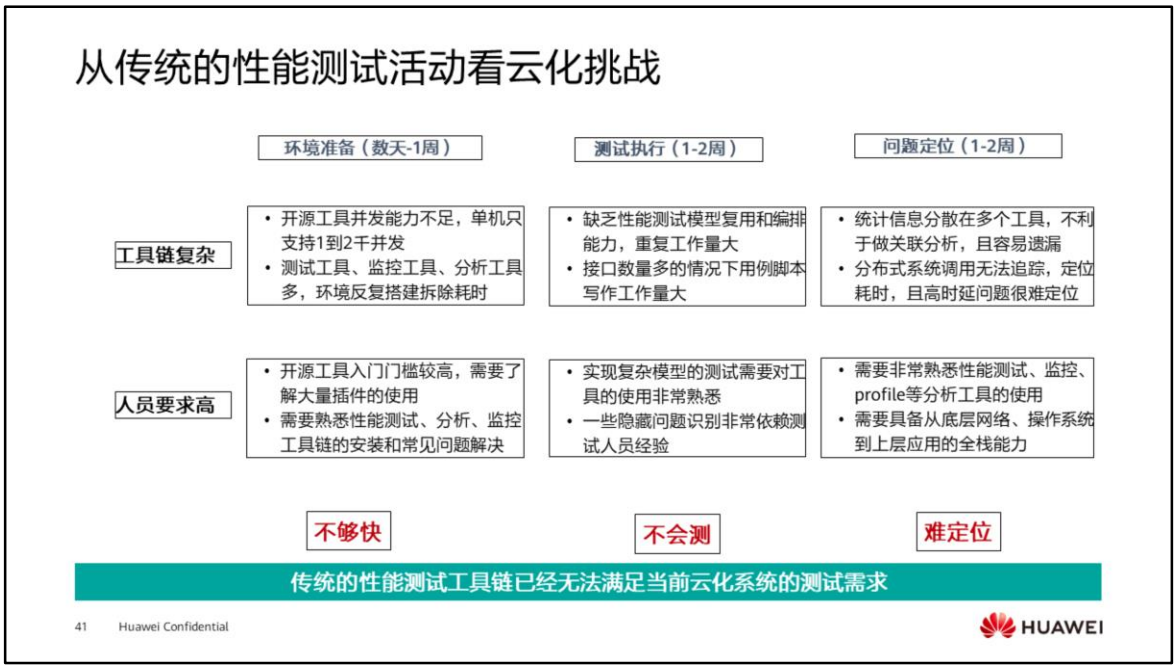
——摘自《大型网站性能监测、分析与优化》

HUAWEI

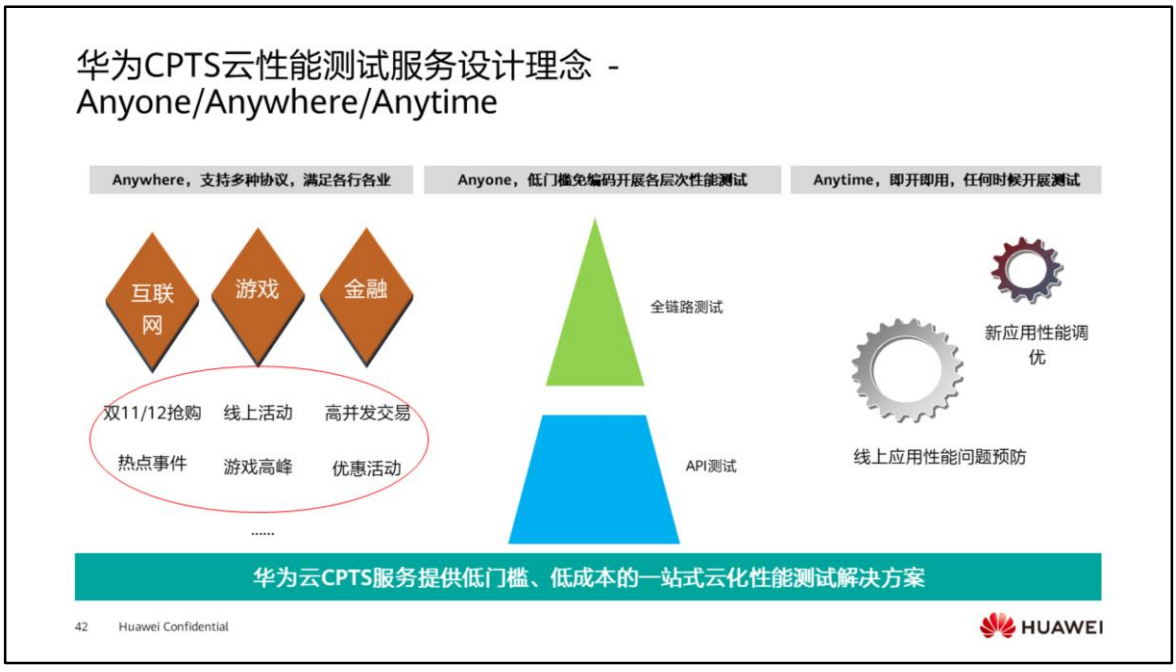
- 性能测试的重要性：应用如果出现性能问题不仅仅是用户体验的损失，而是会产生不可挽回的经济损失。我们看一下这几个统计数据，亚马逊的统计显示客户访问的响应时间每延长一秒，亚马逊一年就会减少16亿美元销售额。首页打开时间每增加100毫秒，网站的销售量会减少1%，都会因载入速度过慢而损失11亿至13亿美元的收入。响应速度在Google的PageRank评分中占有一定的比例，一个好的网站是不可能让他的页面加载速度很慢的，所以网站性能及性能测试是非常重要的。



- 各行各业面临的典型的性能问题。通过这张图来对照看一下典型的业务场景，由于性能问题导致应用访问慢甚至无法访问线上活动促销，比如大促，由于性能问题，导致用户网页无法打开，现在直播比较火，热门直播由于性能问题导致视频卡顿花屏等，还有比如说股票、行情查询、游戏。

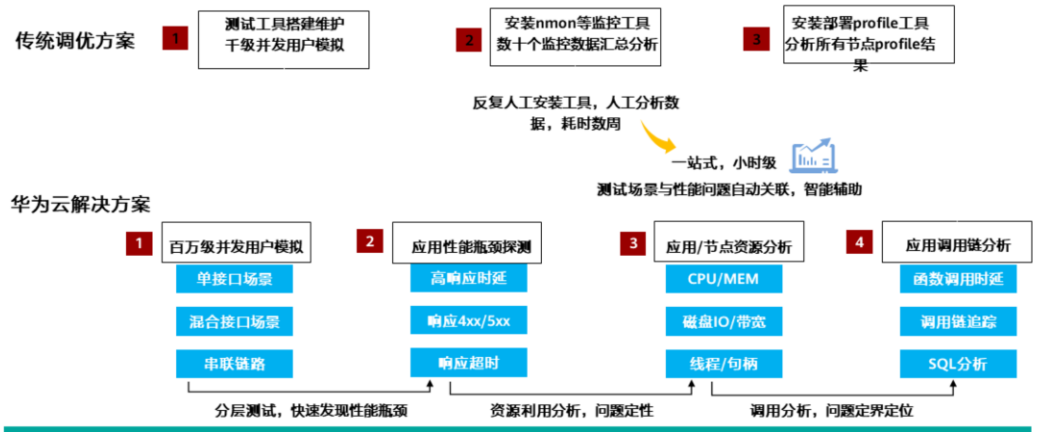


- 从传统的性能测试活动看云化性能测试的挑战。
- 传统的性能测试有如下两个痛点。
  - 第一，工具链复杂从环境准备到测试执行到问题定位都非常耗时，往往需要数周时间才能完成一次性能测试；
  - 第二，痛点对人员的要求非常高，开源工具门槛很高，测试模型设计和测试用例编写需要对工具非常的熟悉，性能问题的定位更是需要测试人员有丰富的性能测试经验。
- 总结起来就是不够快，不会测，难定位，传统性能测试工具链无法满足当前云化系统的测试需求。



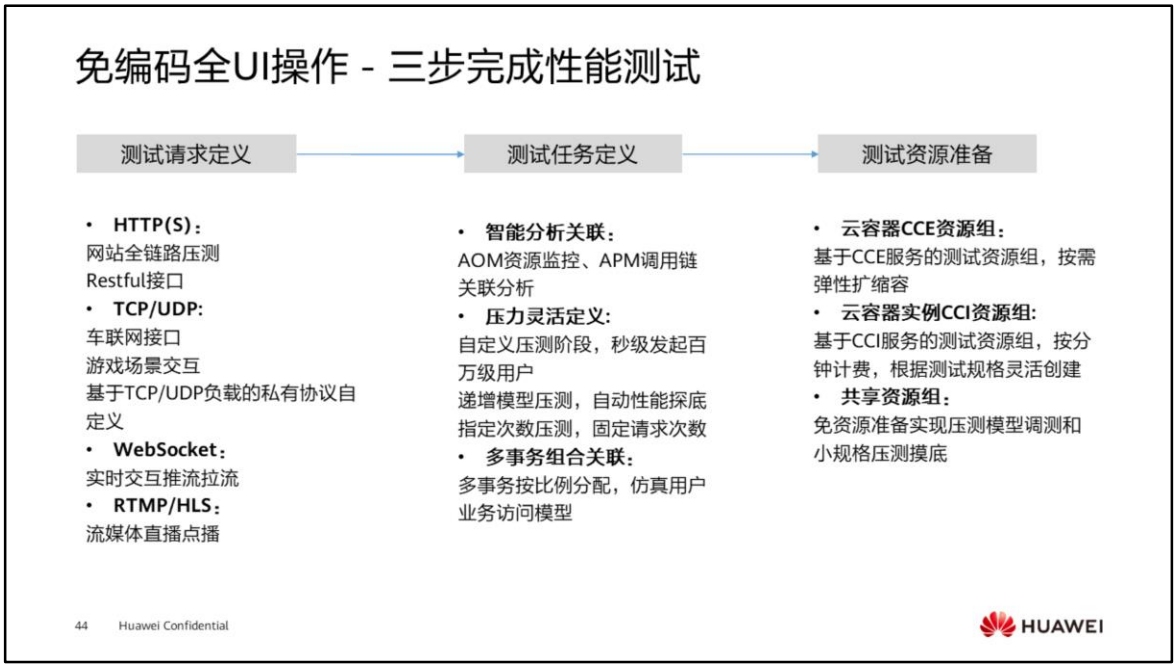
- CPTS: 云性能测试服务，是华为公有云上的一个独立的云服务，专注于性能测试的云化测试工具。

CPTS服务一站式云化性能测试解决方案



百万级高并发引擎，全链路瓶颈分析能力支撑测试周期由周级降低至小时级

HUAWEI



- 云容器引擎（CCE）
  - 云容器引擎（Cloud Container Engine，简称CCE）提供高度可扩展的、高性能的企业级Kubernetes集群，支持运行Docker容器，提供了Kubernetes集群管理、容器应用全生命周期管理、应用服务网格、Helm应用模板、插件管理、应用调度、监控与运维等容器全栈能力，为您提供一站式容器平台服务。借助云容器引擎，您可以在华为云上轻松部署、管理和扩展容器化应用程序。
- 云容器实例（CCI）
  - 云容器实例（Cloud Container Instance， CCI）服务提供 Serverless Container（无服务器容器）引擎，让您无需创建和管理服务器集群即可直接运行容器。通过CCI您只需要管理运行在Kubernetes上的容器化业务，无需管理集群和服务器即可在CCI上快速创建和运行容器负载，使容器应用零运维，使企业聚焦业务核心，为企业提供了Serverless化全新一代的体验和选择。
  - 而Serverless是一种架构理念，是指不用创建和管理服务器、不用担心服务器的运行状态（服务器是否在工作等），只需动态申请应用需要的资源，把服务器留给专门的维护人员管理和维护，进而专注于应用开发，提升应用开发效率、节约企业IT成本。传统上使用 Kubernetes 运行容器，首先需要创建运行容器的 Kubernetes 服务器集群，然后再创建容器负载。

目录

1. 敏捷软件测试理念、方法与实践
2. 测试管理
3. **常见的测试方法**
   - 接口测试
   - 性能测试
   - **契约测试（Mock测试）**
   - WebUI测试
   - 可靠性与安全测试
   - 在线测试
4. 测试度量指标体系和质量评估

HUAWEI

契约测试的背景

- Microservice是一种架构风格，它将单个的应用设计成一组服务的集合。

- **优点**
  - 高度模块化
  - 可独立部署
  - 技术多样性

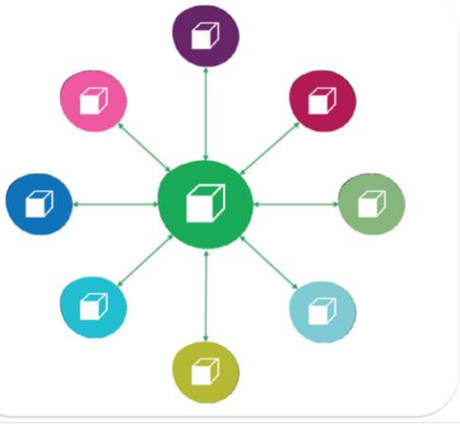
- **痛点**
  - 分布式
  - 最终一致性
  - 管理复杂性



怎么测试微服务（ Microservice ）？

HUAWEI

微服务如何测试



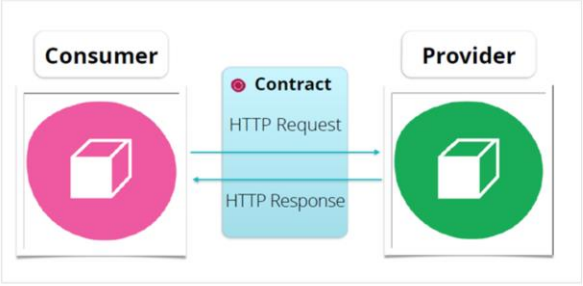
当一个Service已经同时被多个使用者调用的时候，怎么保证Service的修改不会对其他所有使用者造成影响?

Contract Testing

HUAWEI

- 中间的微服务被周边众多服务依赖和互相调用。

什么是契约测试 (Contract Testing)



Consumer: Service的使用者，向provider发起HTTP请求来获取数据；

Provider： Service的提供者，接收consumer的HTTP请求并返回数据；

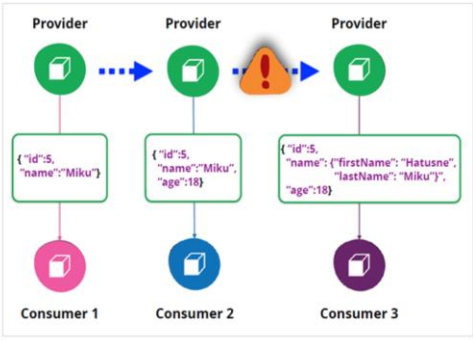
Contract： 契约，一种定义在consumer与provider之间的交互方式；

契约测试是一种针对外部服务的接口进行的测试，它能够验证服务是否满足消费方期待的契约。这个契约包含了对输入和输出的数据结构的期望，性能以及并发性。

HUAWEI

- 两个微服务，一个作为Consumer，一个作为Provider，
- Consumer向Provider发起HTTP请求，获得HTTP响应，请求和响应这个过程通过契约（Concract）固化下来。

为什么要开展契约测试



Consumer Driven
可以保证Provider对Contract的修改永远都基于Consumer的需求，并且是全体Consumer需求的总和

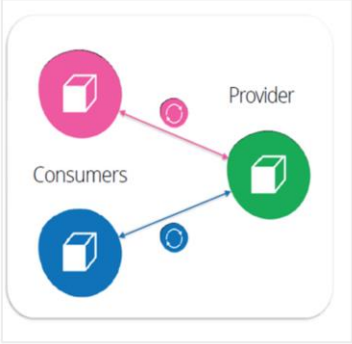
Contract Testing
为Consumer Driven Contract提供了质量保证；

Contract Testing 验证Provider是
否按照期望的方式与Consumer进行交互。

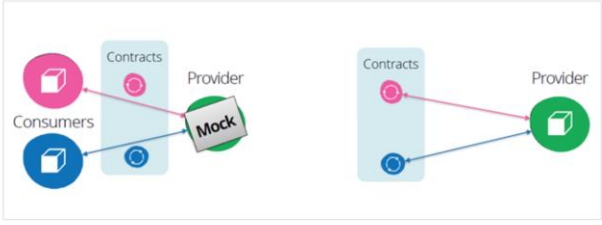
HUAWEI

- 3个Consumer对Provider有依赖，而且依赖的需求不一样，契约是对3个provider的请求的汇总之和。

Contract Testing 的实践方式



产品环境



Consumer
测试环境

Provider
测试环境

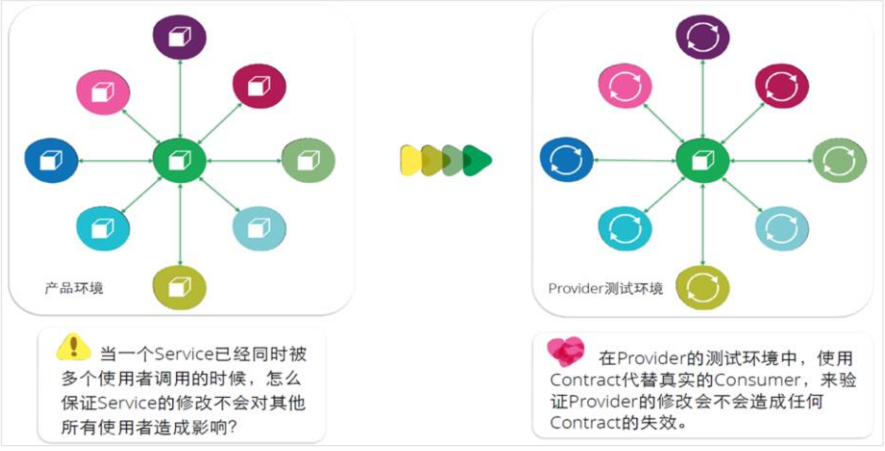
两种关键技术：

1. 通过Mock技术解除Consumer对Provider的依赖；
2. 通过契约测试保证Provider的修改对Consumer保持一致；

HUAWEI

- 消费者与生产者的关系:
  - 左起第一张图是真实的产品环境： 消费者与生产者的调用关系；
  - 第二张图是表达通过mock技术解除consumer对provider的依赖；
  - 第三张图是表达通过契约测试保证Provider的修改对Consumer保持一致；

使用Contract来验证Provider



HUAWEI

- 产品环境（线上真实环境）多个consumer（周边节点的service）对Provider（中心节点的service）有依赖
- 测试环境: 通过契约模拟掉多个consumer（周边节点的service）对Provider（中心节点的service）有依赖

Mock技术介绍

微服务测试面临问题：
1、如何实现环境无依赖，服务间无依赖?
2、如何实现快速测试，支撑服务快速上线?

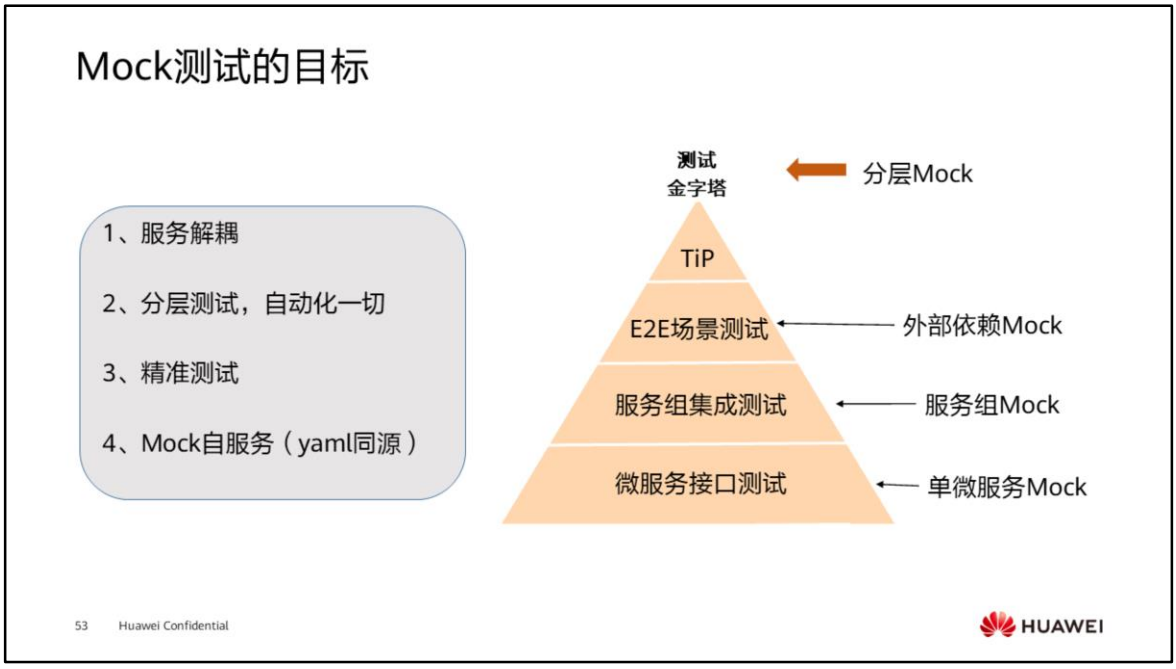


Mock帮你搞定

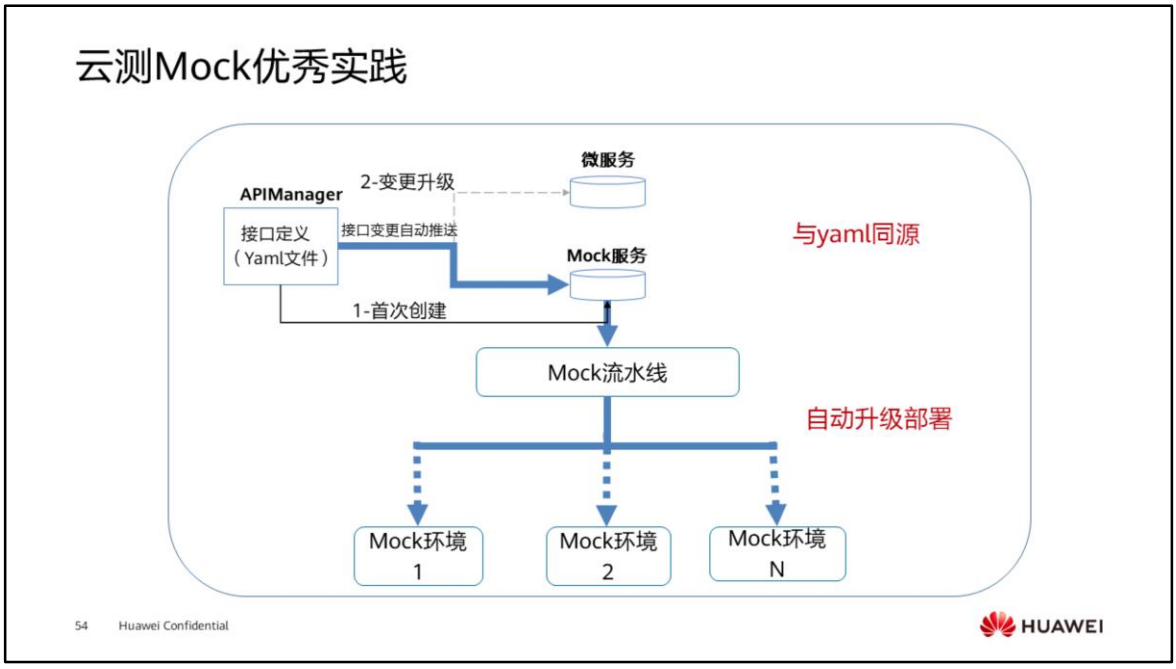
Mock测试的概念：

mock测试就是在测试过程中，对于某些不容易构造或者不容易获取的对象，用一个虚拟的对象来创建以便测试的测试方法。

HUAWEI



- Mock测试的目标
  - 服务解耦： 通过mock解除对其他服务的依赖；
  - 分层测试：微服务接口测试，服务组集成测试，场景测试等；
  - 精准测试： 通过mock技术精准模拟其他服务的返回；
  - Mock自服务： 将服务自己对外提供的接口mock化，从而达到与yaml（API接口定义文档）同源；
  - TIP: 线上测试，也叫生产环境测试；



- APImanager通过yaml文件来对API接口进行定义；
- 当yaml文件变更（API接口有变更），Mock服务感知到后，通过Mock流水线，自动升级部署mock实例。

目录

1. 敏捷软件测试理念、方法与实践
2. 测试管理
3. **常见的测试方法**
   - 接口测试
   - 性能测试
   - 契约测试（Mock测试）
   - **WebUI测试**
   - 可靠性与安全测试
   - 在线测试
4. 测试度量指标体系和质量评估

HUAWEI

WebUI测试的场景

- 冒烟测试（主业务流程）
- 新功能测试
- 回归测试
- 兼容性测试 （一套测试脚本，多个浏览器执行）
- 完成手动测试无法完成的工作下班后无人值守测试



新功能测试由于需求不稳定，变化较多，并不适合进行自动化测试；

回归测试和兼容性测试，前者需要执行大量的回归测试用例，后者需要将这些用例在不同的浏览器重复执行，我们完全可以利用自动化测试工具，将这些重复、低效、易错的工作做得更为高效和专业。

HUAWEI

- WebUI测试的场景：包括了兼容测试、回归测试、新需求测试冒烟测试。
- 新功能测试是因为需求不稳定变化比较多不太适合进行自动化测试。而回归测试和兼容性测试，前者执行了大量的回归测试用例，后者需要这些用例在不同浏览器里面重复执行。那么完全可以利用自动化的方式将这些重复、低效易错的工作变得更加高效，从而节约人力成本。

Web自动化测试

- 更快的测试速度。带来更高的测试效率。一般而言，运行一遍功能测试都要以小时为单位，有的甚至以天为单位。而自动化测试则一般都在分钟级别，如果运行在分布式环境下，甚至可以降到秒级。
- 更高的测试覆盖率。测试场景越复杂，所需的测试用例越多。当测试场景的复杂度超过一定程度后，纯手工的功能测试显然就无力覆盖所有的测试用例了，并且随着复杂度的升高，测试覆盖率会越来越低。
- 更好的稳定性和可扩展性。功能测试靠人，自动化测试靠机器，因此，无论是运行测试的稳定性，还是测试能力的可扩展性（比如从测试1个应用变为测试10个应用）



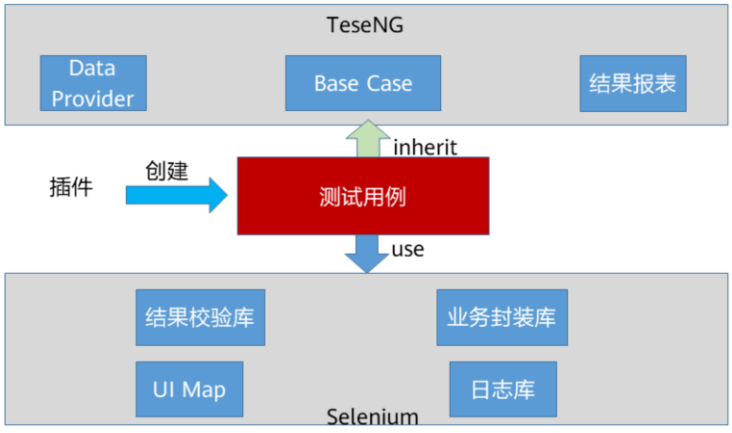
手工测试　　自动化测试

HUAWEI

- Web自动化测试有三大优点：
  - 更快的测试速度，
  - 更高的测试覆盖率，
  - 更好的稳定性，可扩展性。
- 图中所示：做完WebUI的测试后发现要对不同浏览器做重复的测试，如果是通过人工测试的话，右边做自动化测试，部署了一些脚本，在某一个真机上面，从而驱动脚本，在不同浏览器下做自动化测试，完全可以做到下班之前把脚本挂上，然后第二天早上过来看这个脚本执行结果，从而解放测试的人工时间。

Selenium+TestNG的WebUI自动化框架

- 当前自动化通用框架如下图：



实现自动化的方式：

1. 测试和编写业务逻辑；
2. 编写配置文件testng.xml；
3. 运行TestNG类型的工程

HUAWEI

- DataProvider ： DataProvider 使得对同一个方法的测试覆盖变的非常轻松，非常适合进行边界测试，只要给出多种测试数据就可以针对一个测试方法进行覆盖
- Base Case： 提供基础的方法和框架
- 结果报表：运行TestNG，TestNG默认情况下，会生产两种类型的测试报告HTML的和XML的。 测试报告位于 "test-output" 目录下
- 结果校验库：执行过程中添加检查点进行校验
- UI Map： 生成脚本必须的Map文件
- 业务封装库： 封装成共有的方法
- 日志库： 执行日志

- Selenium+TestNG的WebUI自动化框架，如开源的，常见的一些自动化的框架，就是Selenium+TestNG的实验方式：
    - 第一，测试和编写业务逻辑，
    - 第二，编写配置文件TestNG.xml，
    - 第三，运行TestNG类型的工程，里面会有Data Provider、Base Case、结果报表。Data Provider是使用对同一个方法的测试覆盖变得非常容易，非常适合边界测试，Base Case，提供基础的方法和框架，结果报表。顾名思义，就是运行TestNG，TestNG默认情况下会产生两种类型的测试报告，html的和xml的，测试报告位于“test out-put”目录下面，下面的Selenium会有四大组件，结果检验库，执行过程中添加检查点进行校验。业务封装库封装成共有的方法可以供大家来共同共用。UI Map，生成脚本必须的Map文件，日志库，顾名思义，执行日志存在里面。

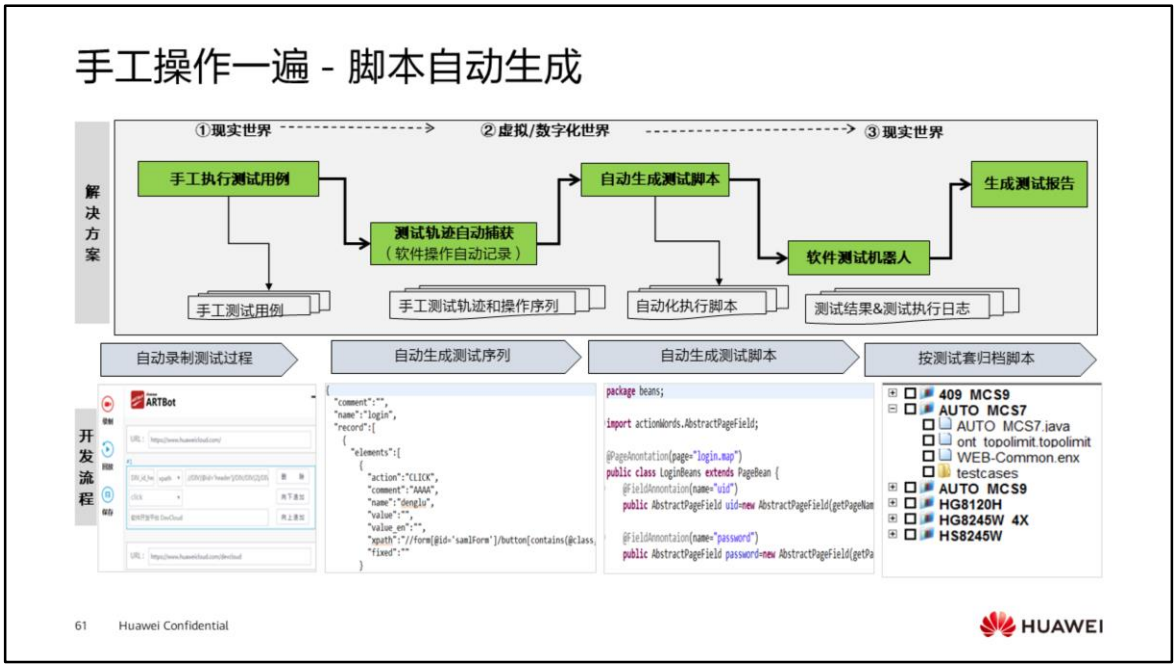
云测 - 一站式Web全自动化测试解决方案

- Selenium+TestNG需要基本的编程经验，而且需要查阅大量的API说明，不利于手工测试人员快速上手。
- DevCloud的Web自动化满足了手工测试人员期望“不用编码就能实现Web自动化测试”。具体的方案如下：



HUAWEI

- 云测里面一站式的Web全自动化测试的解决方案：
  - Selenium+TestNG需要基本的编程经验，需要查阅大量API说明，不太适合手工测试人员快速上手。手工人员期望不用编码就能实现。
  - Web自动化具体的方案如下：
    - 手工操作一遍，工具自动记录，测试脚本。
    - 通过图像化的脚本编排，提取测试步骤共有的key word。
    - 基于上面的执行框架进行回放执行。
    - 接入工厂或者是到执行机上面去，让它自动或者定时的执行，结果上报，并且生成对应的测试报告。



- 测试轨迹自动捕获：自动记录操作序列和检查点序列
- 自动生成测试脚本：操作序列流转化成自动化脚本
- 软件测试机器人： 自动执行测试脚本

图形化编排 - 可视化脚本编写方案

- 使用java作为脚本开发语言，对脚本编写人员有一定的门槛要求，使用可视化通过拖拽的方式开发脚本，不需要java语言基础，降低了门槛要求
- 手工编写脚本，开发效率低，可视化通过拖拽的方式编写脚本，提升开发效率
- 所见即所得，可视化通过业务KW组装成用例，业务流程清晰、直观

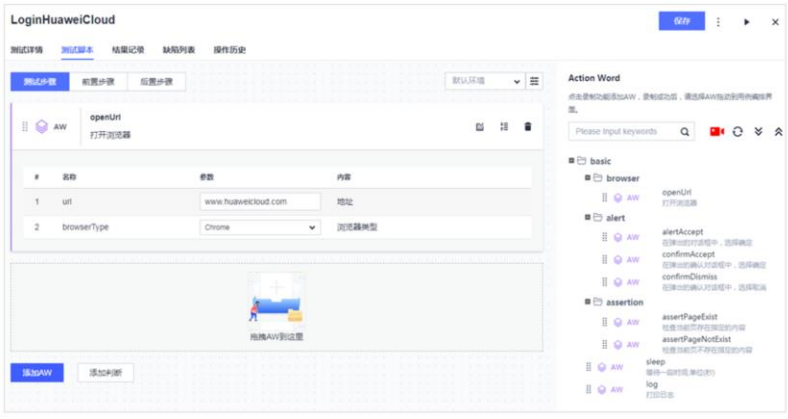


HUAWEI

- 云测里面的WebUI自动化里面主要的一些能力：
  - 第一，手工测试一遍，脚本自动生成，手工执行测试脚本，然后系统会自动记录操作轨迹，然后系统会自动生成测试脚本，然后通过软件测试机器人，其实就是所谓的执行机，然后去执行脚本，最后生成测试报告。

云测 - WebUI自动化测试优秀实践

- 使用DevCloud测试工具ARTBot进行WebUI场景功能自动化测试，结合华为优先研发实践和研发理念，测试用例免代码编写、录制回放机制生成脚本，基于脚本模板组装编排、管理WebUI自动化测试用例。



简单
- 0编码开发成本
- 0环境搭建成本
- 3步生成WebUI脚本
- 1步WebUI自动化执行

高效
- 可视化编排测试用例
- 测试关键字批量复用
- 云端弹性资源池
- 实时反馈测试结果

专业
- 高级测试场景编排
- 定制化用户数据配置
- 固化常用步骤
- 流水线持续自动化测试

HUAWEI

- 第二个能力，图形化的编排，可视化脚本编写方案，可以看到Key Word（KW）的生成，然后保存到业务的Key Word(KW)的库，然后通过组装生成可视化的业务流程，并且生成最后的Java脚本、云测WebUI自动化优秀实践，就是简单、高效、专业。

软件可靠性定义和方法

- 软件可靠性是指软件在所规定的环境条件下和规定的时间内正确地完成任务的能力。
- 可靠性测试方法主要有两种类型:

  1、基于操作剖面开展测试

  2、基于故障注入开展测试

HUAWEI

- 软件可靠性的定义和方法可靠性测试方法主要有两种：
  - 基于操作剖面开展测试
    - 按照软件操作剖面对软件进行随机测试的测试方法。根据软件在客户实际使用过程的各种操作的使用概率选择操作剖面，对软件系统进行可靠性测试。
  - 基于故障注入开展测试
    - 故障注入是按照选定的故障模型，采用人工的方法将故障注入到特定的目标系统中，同时采集系统对所注入故障的反应信息，通过这些信息对系统进行可靠性分析。

软件故障注入技术

- 与其他可靠性评价技术相比，故障注入方法能方便灵活、便捷有效地处理各种可靠性问题，得到了越来越多的关注。故障注入技术有模拟实现的故障注入、硬件实现的故障注入和软件实现的故障注入三种。
  - 软件故障注入
  - 静态故障注入
  - 动态注入

HUAWEI

- 软件故障注入：是通过修改硬件或软件的状态变量或相关数据来模拟故障的产生，加速系统的失效，分为静态注入和动态注入两种类型。
- 静态故障注入：主要通过程序变异的方法，通过改变原程序，使被测系统文件静态的存在错误，从而使其运行时出现故障。静态注入占用很少的系统资源，能够较好地保持系统原来的时序，这种注入法有很好的优化性。
- 动态注入：是在被测系统正常运行过程中，实现故障注入。该种方式是根据被测系统的运行状态或条件注入故障的，具有灵活性。

目录

1. 敏捷软件测试理念、方法与实践
2. 测试管理
3. **常见的测试方法**
   - 接口测试
   - 性能测试
   - 契约测试（Mock测试）
   - WebUI测试
   - **可靠性与安全测试**
   - 在线测试
4. 测试度量指标体系和质量评估

HUAWEI

可靠性设计 - 测试指标

类别	关注灾难场景	指标	描述	测试/运维活动
数据可靠性	运维误操作或基础平台问题造成数据大量丢失	数据恢复用时（RTO）	从数据丢失开始，到数据恢复的时间	数据恢复演练
		数据恢复点（RPO）	数据丢失后，能够保证数据恢复到之前的某个时间点	定时备份、每日备份巡检
系统可用性	单点故障对业务的影响： 虚拟机或容器下电； 虚拟机重启； 网络闪断、中断； 进程异常退出； 资源类：CPU、内存占用率高；	故障检测时间	从故障发生，到系统检测到故障的时间， 比如HA对后端健康检查的频率	单节点故障下的高可用测试
		故障隔离时间	从故障发生到系统管理节点隔离故障的时间， 比如HA隔离后端掉线节点的时间	单节点故障下的高可用测试
		故障定位时间	从故障发生到检测告警平台发现并定位到 最小故障单元（如进程）的时间	监控告警测试
		故障恢复时间	单节点故障能够自动或手动被恢复的时间	单节点故障下的高可用测试
升级部署	升级时业务中断	上线、升级时间	从提交代码到完成部署的时间	升级流水线测试
		升级中断时间	升级过程中业务中断的时间	升级中断测试
业务可用度	现网业务异常	业务可用度	全年内业务可用的时间百分比，业界一般为3个9到4个9	现网拨测

HUAWEI

- 可靠性设计里面会有哪些度量的测试指标，主要分四种：数据可靠性、系统可用性、升级部署，业务可用度，主要关心的是数据恢复用时（RTO）、数据恢复点（RPO）故障检测时间、故障隔离时间，故障定位时间，故障恢复时间、上线升级时间、升级中断时间和业务可用度。

可靠性测试 - Chaos混沌测试

- 随着微服务和分布式云架构的崛起，Web 变得日趋复杂，“随机性”的故障因此变得越来越难以预测，而我们对这些系统的依赖却与日俱增，企业需要想办法解决这些问题，因为等到下一次事故发生就为时已晚。为此，混沌工程应运而生。



2008年Netflix开始将服务从数据中心迁移到云上，之后就开始尝试在生产环境中开展一些系统弹性的测试。过了一段时间，这个实践过程才被称为混沌工程。最早被大家熟知的是“混乱猴子”（Chaos Monkey），因为其在生产环境中随机关闭服务节点而“恶名远扬”。进化成为“混乱金刚”（Chaos Kong）之后，这些之前获得的小规模益处被扩大到非常大。规模的扩大得益于一个叫作“故障注入测试”（Fault Injection Test，FIT）的工具

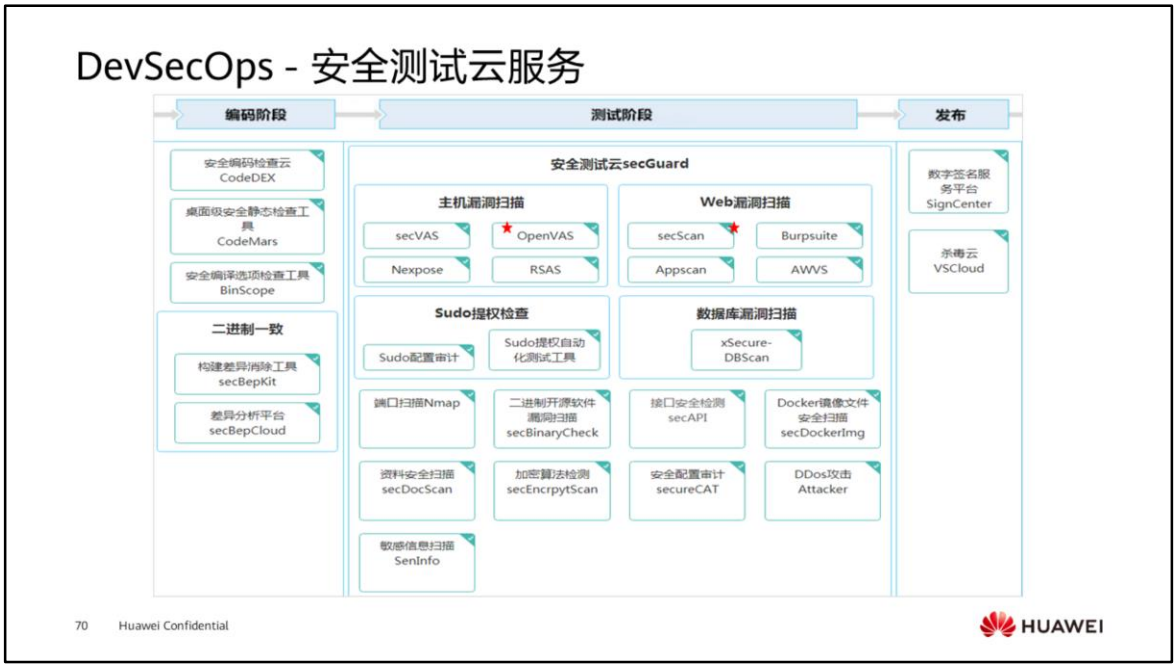
HUAWEI

- 微服务的故障给公司造成巨大损失，也给用户带来很大的麻烦，影响他们进行在线购物、交易或打断他们的工作。即使是一些简单的故障也会触及公司的底线，因此，宕机时间就成为很多工程团队的 KPI。
- 2017 年，有 98% 的企业表示，一小时的宕机时间将给他们带来超过 10 万美元的损失。一次服务中断有可能让一个公司损失数百万美元。最近，英国航空的 CEO 透露，2017 年 5 月发生的一次技术故障造成数千名乘客滞留机场，给公司造成 8000 千万英镑的损失。2008年，Netflix也开始从数据中心迁移到云上，然后开展一些系统的弹性的测试，这个实践过程被称为混沌测试。这就是最开始“Chaos Monkey”的一个混沌测试的原型。

什么是 DevSecOps

- 安全在很多团队中都是常见的隔离点。安全的核心关注点是保护团队，而有时这也意味着创建延缓新服务或是新产品发布的障碍或策略，用于保障任何事都能被很好的理解和安全的执行，并且没有给团队带来不必要的风险。
- 因为安全隔离点方面的明显特征和它可能带来的摩擦，开发和运维有时会避开安全要求以满足客观情况。在一些公司，这种隔离形成了一种产品安全完全是安全团队责任的期望，并取决于安全团队去寻找产品的安全缺陷或是可能带来的问题。
- DevSecOps 是通过给开发或是运维角色加强或是建立安全意识，或是在产品团队中引入一个安全工程师角色，在产品设计中找到安全问题，从而把安全要求汇聚在 Devops 中。

HUAWEI



- 云测未来会上线主机漏洞扫描 secVAS 和 web漏洞扫描 secScan
- secVAS：自动化 Web 漏洞扫描工具
- secScan：自动化主机漏洞扫描工具
- OpenVAS：开放式漏洞评估系统，其核心部件是一个服务器，包括一套望楼漏洞测试程序，可以检测远程系统和应用程序中的安全问题
- Burpsuite：Web渗透-扫描工具，用于攻击web 应用程序的集成平台。它包含了许多工具，并为这些工具设计了许多接口，以促进加快攻击应用程序的过程
- NexPose：Nexpose 是一款极佳的漏洞扫描工具，可以更新其漏洞数据库，以保证最新的漏洞被扫描到
- RSAS(Remote Security Assessment System)：绿盟科技远程安全评估系统，主动诊断安全漏洞
- AppScan：AppScan是用于web项目的安全测试工具，扫描网站所有url，自动测试是否存在各种类型的漏洞。
- AWVS: AWVS是知名的WEB漏洞扫描工具,通过网络爬虫测试网站安全,检测流行的Web应用攻击

目录

1. 敏捷软件测试理念、方法与实践
2. 测试管理
3. **常见的测试方法**
   - 接口测试
   - 性能测试
   - 契约测试（Mock测试）
   - WebUI测试
   - 可靠性与安全测试
   - **在线测试**
4. 测试度量指标体系和质量评估

HUAWEI

为什么要做在线测试

- 为什么要做在线测试?

  生产环境是测试工程师最有用的资源之一， 在线测试给我们提供的一些挑战如下：

  1.我们如何知道在开发人员自测或测试环境中没有问题的服务将在生产环境中工作?

  2.我们可以从生产中收集哪些信息，帮助我们发布更高质量的产品?

  3.我们如何检测和应对软件升级后发现的问题?

- 在线测试基于以上3个挑战出发，利用生产环境真实数据回流至研区环境提升产品质量，成为当前服务/微服务研发必需的测试活动

HUAWEI

- 在线测试其实就是生产环境下面的一个测试，在线测试是面临着三个挑战，然后基于上面的三个挑战，要利用生产数据真实数据回流到研发环境中，提高产品质量，成为当前服务微服务研发必须的测试活动。

什么是在线测试

- 在线测试是指在生产环境中开展的，或者利用生产环境数据开展的各种类型的测试活动，我们在做在线测试活动过程中要能回答以下几个疑问

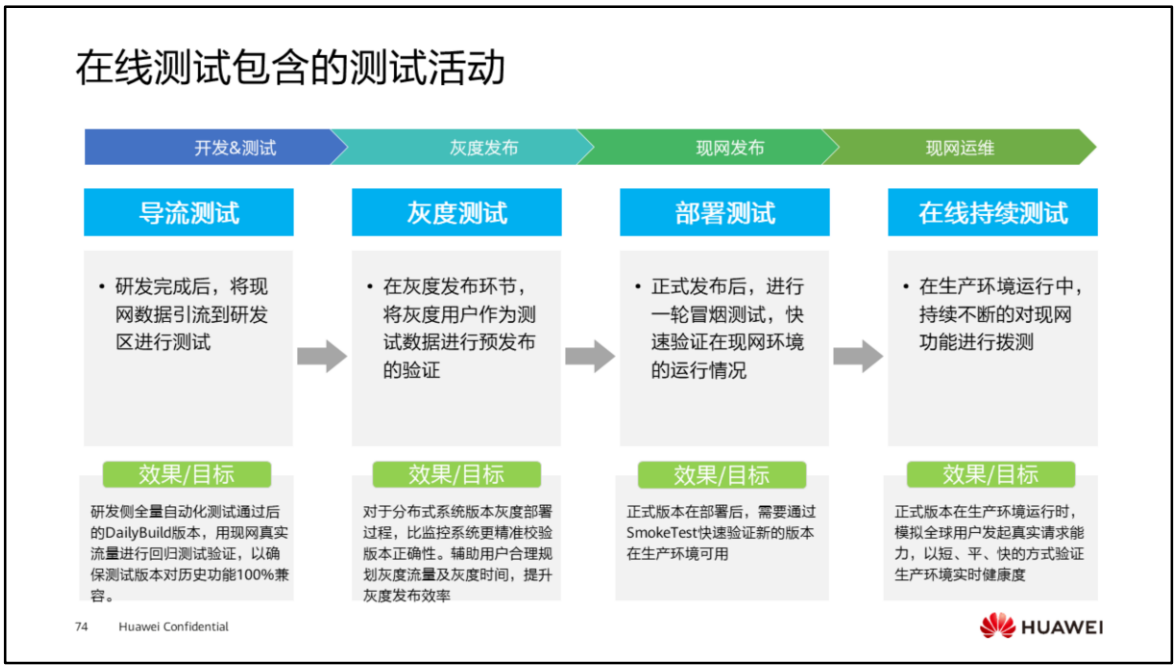
如何构造生产环境数据进行测试?

灰度发布应该观察多久时间? 需要观察哪些数据?

如何模拟现网用户真实使用场景测试？

现网架构越来越复杂，如何保证系统时时刻刻都是可用的?

HUAWEI



- 在线测试包含的测试活动：
  - 导流测试，就是将现网数据引流到研发区进行测试。
  - 灰度测试，在灰度发布环节，将灰度用户作为测试数据进行预发布的验证。
  - 部署测试，正式发布前进行一轮冒烟测试，快速验证现网环境下是否正常。
  - 在线持续测试，在生产环境中持续不断的对现网功能进行拨测。

在线测试 - 导流测试

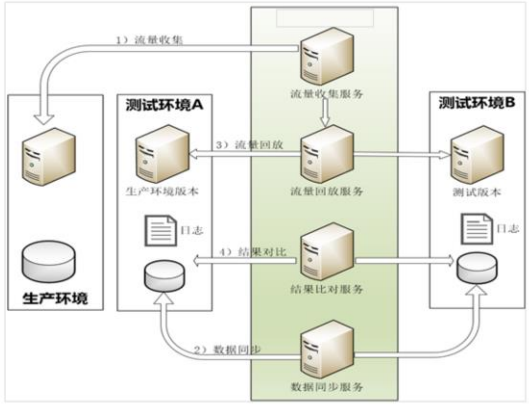
- 导流测试原理：
  - 将生产环境用户真实数据引流至研发区验证环境，通过对比研发区的验证结果与生产环境真实结果是否一致判断新版本是否存在潜在问题
- 导流测试要点：

  1.请求收集：收集生产环境用户真实请求

  2.数据同步：生产环境冷备DB数据，同步至A/B两套环境

  3.流量回放：将真实流量回放至A/B两套环境

  4.结果对比：比对A/B两套环境流量执行结果



HUAWEI

- 导流测试的基本原理：将真实的数据引流到研发区验证环境，将对比研发区鉴定结果与生产环境真实结果是否一致，判断新版本是否有潜在的问题，从图中得知，把生产环境的数据然后返回到研发区来，然后进行数据和流量的回放，通过日志里面进行结果对比。

在线测试 - 部署测试

- 部署测试原理：
  - 部署在生产环境后，快速启动基于现网真实数据对该发布版本进行快速验证，保证新版本重要功能运行正常。
- 部署测试要点：

  1.测试时机：在生产环境部署后快速启动部署测试

  2.测试数据：基于生产环境的真实数据请求

  3.测试范围：执行核心功能和常用功能所对应的测试用例

  4.测试结果：核心功能或者常用功能用例执行失败立即定位解决或回退

HUAWEI

- 部署服务的主要要点掌握这四点:
  - 测试时机，在生产环境部署和快速启动部署测试
  - 测试数据，基于生产环境的真实数据请求
  - 测试范围，执行核心功能和常用功能对应的测试用例
  - 测试结果，核心功能和常用功能用例执行，失败后立即定位，解决或者回退版本

在线测试 - 在线持续测试

- 在线持续测试原理：
  - 产品/服务发布后，在生产环境对产品/服务重要功能进行7*24小时持续拨测以便尽早或先于用户发现问题并协助研发人员快速修复上线。
- 在线持续测试要点：

  1.筛选用例：筛选版本重要功能对应的测试用例

  2.定时拨测：定时（每隔5分钟、1小时）对重要功能进行拨测

  3.现网告警：拨测用例执行失败并发送告警短信，提示告警等级

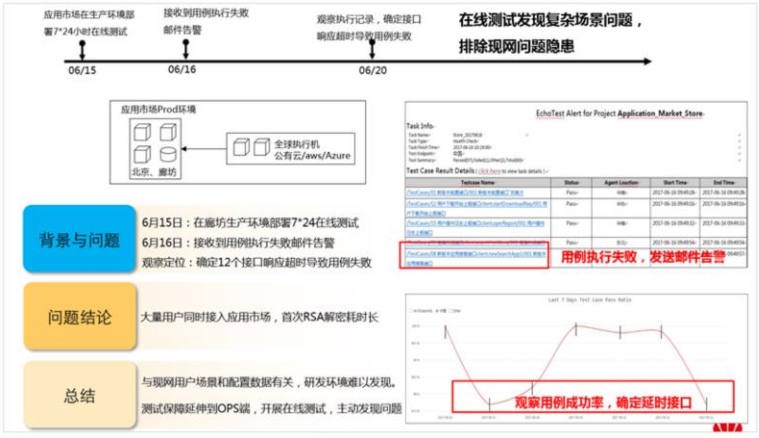
  4.处理上线：研发人员根据告警快速处理并快速修复上线



HUAWEI

- 在线持续测试是在线测试很关键的一个能力。产品发布之后，在生产环境下对产品服务重要工程进行7X24小时持续拨测，以尽早的或者先于用户发现问题，并协助研发人员快速修复上线。
- 一旦上线的服务发现问题，是通过自动化的脚本，实时的拨测，及时发现问题，而不是说从用户那儿得到反馈，说当前什么服务宕机了，或者是派测试人员去手工去测试，因为有时候现网问题，可能跟环境，包括部署的方式，包括某一些服务器异常了，都可能会影响最后的结果。
- 所以在线持续测试的要点：
  - 第一，筛选我们重要的用例
  - 第二，定时的拨测
  - 第三，现网告警，就是说一旦服务出现问题了，立刻就有告警信息以邮件或者短信的方式给提供过来
  - 第四，及时的根据告警信息快速处理并且快速修复上线在线

云测在线持续测试优秀实践



关键技术：

1、用例共享：研发、在线测试用例资源共享，提升研发效率

2、执行方式灵活：用例按节点区域、并行串行、定时执行等多种执行策略，满足不同测试需求

3、模拟全球用户持续发起在线测试：全球部署测试客户端,模拟全球用户，跨区域持续测试被测服务，实现分布式部署，实时发起线上真实请求并收集数据

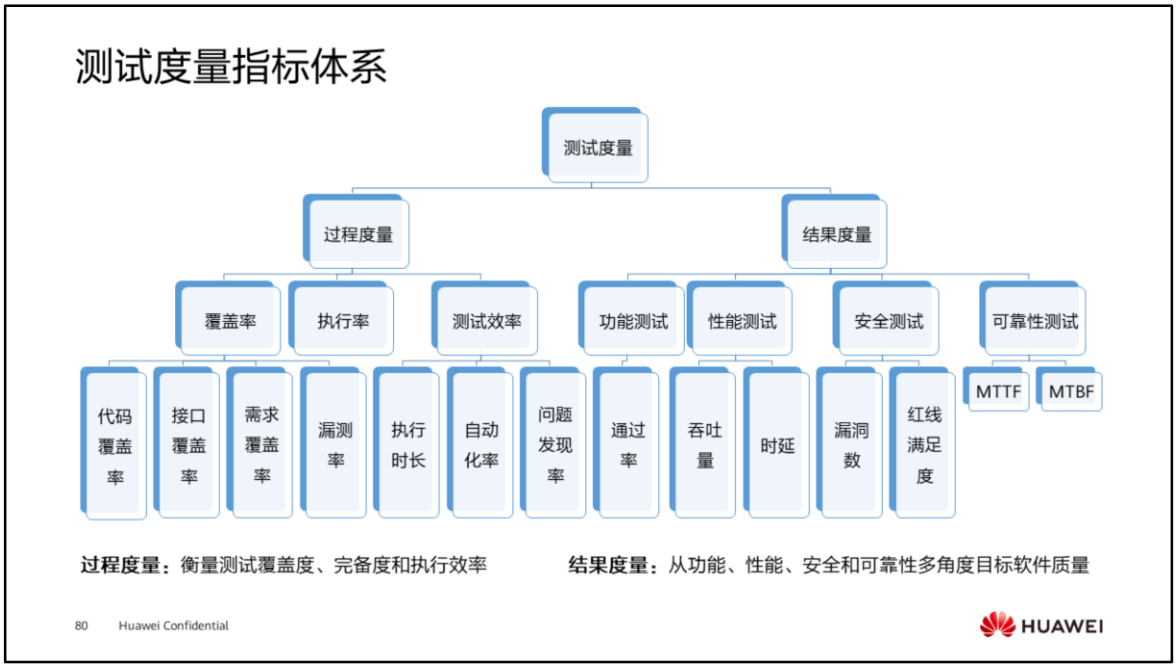
HUAWEI

- 在线测试实践案例：应用市场-开展7*24小时在线测试，发现复杂场景性能问题

目录

1. 敏捷软件测试理念、方法与实践
2. 测试管理
3. 常见的测试方法
4. **测试度量指标体系和质量评估**

HUAWEI



- 测试度量指标体系用来衡量会一个好的测试需要执行的相关操作。

质量退出与质量评价

测试退出条件

当达到了必要的信心级别，风险可以接受时

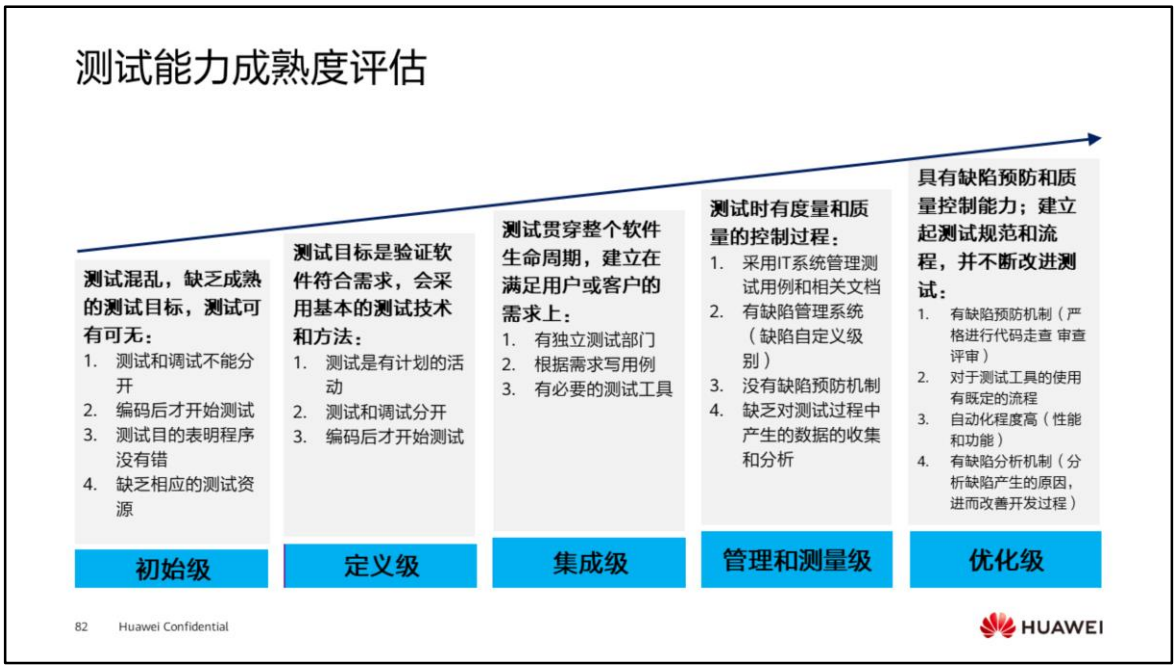
当发现缺陷的代价 > 缺陷发生引起的代价时

当达到测试完成标准（退出/成功标准）时



HUAWEI

- 测试满足退出条件说明当前质量是可控的。



- 从以下几个方面评估测试能力成熟度：
  - 第一，初始级，测试混乱，缺乏成熟的测试目标，测试可有可无。
  - 第二，定义级，测试的目标是验证软件符合需求，会采用基本测试技术和方法。
  - 第三，集成级，测试贯穿整个软件的生命周期，建立在满足用户和的需求上的。
  - 第四，管理和测试级，测量集，测试是有度量和质量的控制，这里面会就涉及到一些it的系统用例或者是相关文档。
  - 第五，优化级，具有缺陷预防和质量控制，建立起测试规范和流程，并不断改进测试。这是最开始提到的，整个的持续测试的一个范畴。

敏捷软件测试 - 方法与实践

- DevCloud云测和移动应用测试实践：云端一站式测试解决方案



- 高效：一站式云端测试平台，高效协同，开展用例设计、测试执行、缺陷提交，提高测试效率
- 双向追溯：通过需求-用例-缺陷双向追溯，确保所有需求都被正确测试，杜绝漏测、误测
- 自动化测试：提供Android和iOS兼容性测试、接口测试、性能测试，集成流水线支持持续自动化测试
- 可视化：多维度产品质量看板，全方位评估产品质量，保障产品高效验收

HUAWEI

- 敏捷软件测试的一个整体的解决方案。云测里提供一站式的测试解决方案。

思考题

1. 持续测试与传统测试的区别？为什么我们要在微服务测试过程中采用持续测试的方式？
2. 从测试方法来说，有哪些测试方法，分别针对的场景是什么？
3. 测试质量退出的标准是什么？

HUAWEI

- 理想情况下测试范围应该足够广，频率应该足够高，但这只是理想情况。项目执行中实际情况往往是因为赶进度，测试周期被压缩，同时测试人力也不足，这就要求测试人员技能必须提升，以当时同时通过高度的自动化、测试、左移等方法，尽早的发现问题，要做到这些就要开展敏捷测试。
- 测试方法：
  - 黑箱测试（black-box testing），是软件测试方法，测试应用程序的功能，而不是其内部结构或运作。此测试方法可适合大部分的软件测试，例如集成测试（integration testing）以及系统测试（system testing）。
  - 白箱测试（white-box testing，又称透明盒测试glass box testing、结构测试structural testing等）是一个测试软件的方法，测试应用程序的内部结构或运作，而不是测试应用程序的功能（即黑盒测试）。可以应用于单元测试（unit testing）、集成测试（integration testing）和系统的软件测试流程，可测试在集成过程中每一单元之间的路径，或者主系统跟子系统中的测试。
  - 功能测试是按照软件的各个功能划分，进行有条理的测试。在功能测试时要保证测试项覆盖了所有的功能，以及各种条件的组合。

    - 系统测试，一个完整的软件的整体角度来进行测试。系统测试和功能测试的主要区别是系统测试要利用所有真实的数据来进行测试，模拟真实的用户，同时关注点是整个软件系统，而不是单点的功能。
    - 极限值测试是软件在各种特殊条件、特殊环境下能否能正常运行，进行测试。特殊条件一般指软件规定的最大值。最小值，以及在超过最大值超过最小值的情况下，进行一个测试。特殊环境一般是指软件运行的机器处于CPU、高负荷或者是网络高负荷的状态下进行测试。
    - 性能测试是对软件性能的评价，简单的说就是衡量软件具有的响应能力。因此性能测试时采用测试手段，对软件的响应的及时性进行评价的一种方式。
    - 单元测试是针对程序设计中的每一个单点函数进行测试的过程。
    - 集成测试关注模块与模块之间的交互。
    - 系统测试是端对端的，对整个系统进行测试，使用真实的用户场景和数据。
    - 回归测试是验证新加的代码，是否会对现有的功能产生影响，确保现有功能正常可用。
- 测试退出条件：
    - 当达到了必要的信心级别，风险可以接受时；
    - 当发现缺陷的代价 > 缺陷发生引起的代价时；
    - 当达到测试完成标准（退出/成功标准）时。

本章总结

- 敏捷测试的测试理念、方法与实践。
- 测试的四个阶段：测试策略、测试设计、测试执行和测试报告
- 测试方法和覆盖场景：接口测试、性能测试、契约测试、webUI测试、可靠性与安全测试、在线测试以及移动应用自动化测试
- 测试质量的度量指标、质量的评估以及测试能力成熟度的分类：初始级、定义级、集成级、管理和测试级和优化级。

HUAWEI

学习推荐

- 华为云官网：
  - https://www.huaweicloud.com/
- 华为云软件开发平台DevCloud：
  - https://www.huaweicloud.com/devcloud/
- 华为云帮助中心：
  - https://support.huaweicloud.com/
- 华为云学院：
  - https://edu.huaweicloud.com/

HUAWEI

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and organization for a fully connected, intelligent world.

Copyright©2020 Huawei Technologies Co., Ltd.
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.



持续安全与审计



HUAWEI

前言

- DevSecOps是在DevOps实践基础上，在持续构建的各阶段引入针对产品的安全保障活动，强调开发、运维/运营、安全团队的融合运作，是DevOps的增强和完善。

HUAWEI

- 本章节对DevSecOps理念和方法进行研究，将质量、软件工程能力、安全等要求，通过规范、软件/API/用例、工具固化到DevSecOps生产线，自动化的、透明的融入服务管理全生命周期，在保障安全可信的同时又不会方案灵活性、达成产品的“稳定可靠、安全可信”。

目标

- 了解DevSecOps理论、方法、实践
- 理解DevSecOps在软件生命周期管理的重要意义
- 掌握研发过程中安全策略与工具的基本分类
- 理解安全编码检查、漏洞扫描对软件开发过程管理的重要意义
- 学会如何应用安全编码检查、漏洞扫描工作保持产品项目整体安全性

HUAWEI

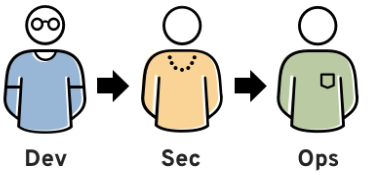
目录

1. **DevSecOps的概念与案例**
2. 安全策略与工具

HUAWEI

DevSecOps概念

- DevSecOps是在DevOps实践基础上，在持续构建的各阶段引入针对产品的安全保障活动，强调开发、运维/运营、安全团队的融合运作，是DevOps的增强和完善。



HUAWEI

- DevOps可以提供持续构建环境和部署方法，使产品高速、高质量交付。

DevOps给安全带来的挑战和思考

- DevOps给安全带来的挑战：
  - DevOps提高了速度并扩张了规模，让安全团队面临着新的挑战。
  - 安全从业人员必须将DevOps原则引入他们的工具和工作流，以便跟上现代应用交付的步伐。
  - 安全解决方案应该使用DevOps所热衷的模块化、自动化和可扩展的技术进行构建和部署。
  - DevOps的概念，比如标准化配置、不变的基础设施以及镜像系统等都有助于减轻安全问题的复杂性。
  - DevOps风格的透明性和问责制是确保在安全团队内进行有效合作的关键。

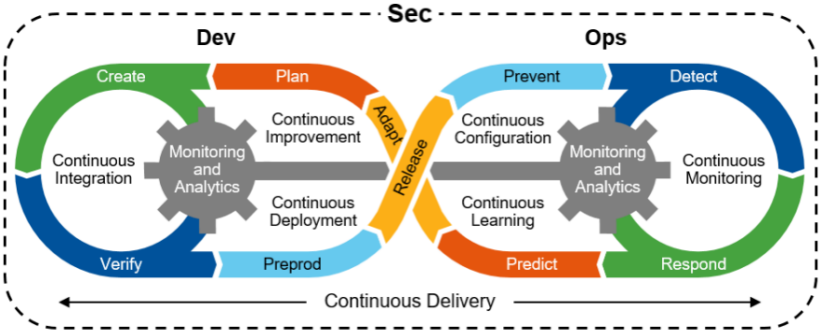


HUAWEI

图片来源：https://tldrsec.com/blog/appsec-cali-2019/

DevSecOps的价值和实践

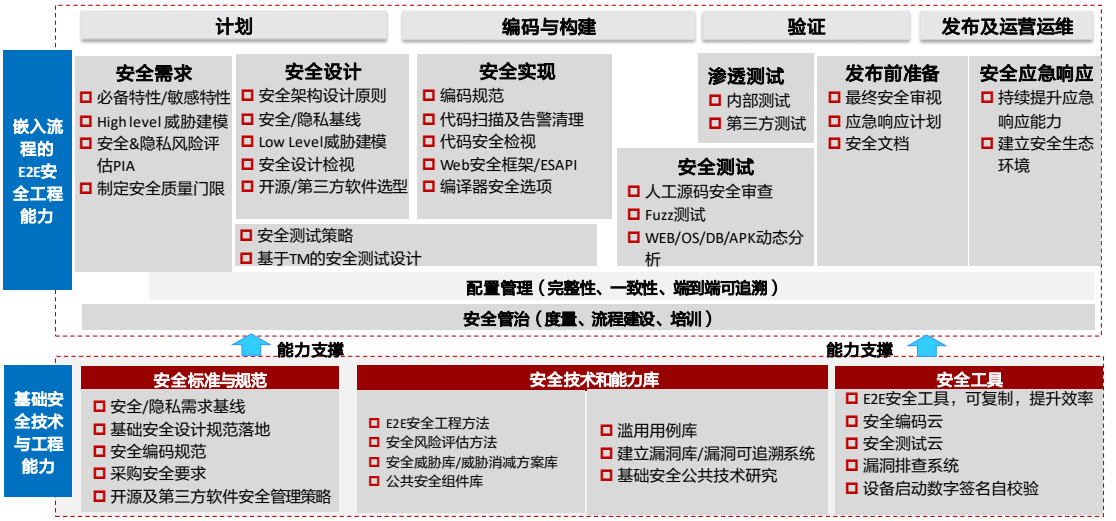
- 权衡DevOps速度与现有安全要求的需求催生了一个名为DevSecOps的模型。DevSecOps基于“安全问题，人人有责”的原则。它强调应用程序开发人员可以怎样把安全检查与他们的集成和部署流水线构建到一起。



HUAWEI

- 安全贯穿始终，Gartner给出的最佳实践有：
  - 安全控制尽可能的自动化和模块化
  - 对所有的应用实施一个轻量化的威胁建模
  - 通过IAM和基于角色的权限控制来实现职责分离
  - 在开发过程中扫描开源软件
  - 在开发过程中扫描漏洞及配置文件
  - 扫描自定义的代码、应用及API
  - 将脚本、模板作为敏感代码进行对待
  - 衡量系统完整性并确保加载正确
- 图片来源：https://www.bankinfosecurity.com/devsecops-keys-to-success-a-11299

华为实践 - 贯穿研发过程的安全保障体系



HUAWEI

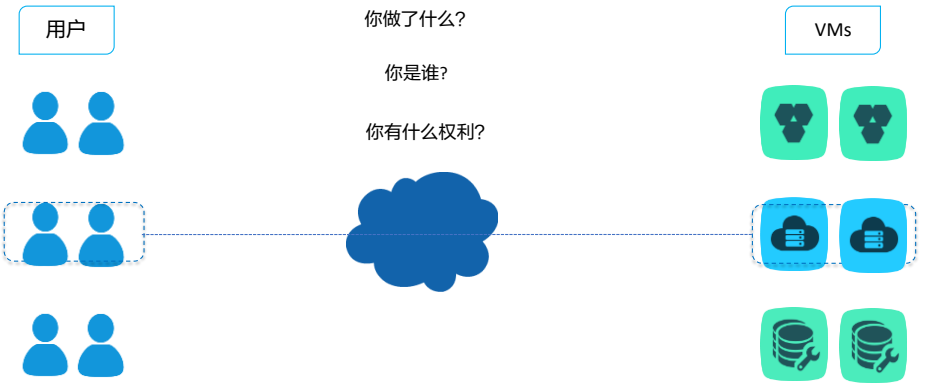
- 华为在践行DevOps时，有一套贯穿研发过程的安全保障体系，从计划阶段、编码与构建阶段、验证阶段、发布与运营阶段嵌入了多个安全工程能力，包括安全需求、安全设计、安全实现、安全测试及安全事件响应，通过一系列的安全管制，也逐步完善了自己的安全标准，安全技术和能力库，并且衍生了更多的安全工具来实现更高的安全指令。

目录

1. DevSecOps简介
2. **安全策略与工具**
   - 操作和资源可管控
   - 编码安全与工具
   - 漏洞扫描与工具
   - 安全运维与工具

HUAWEI

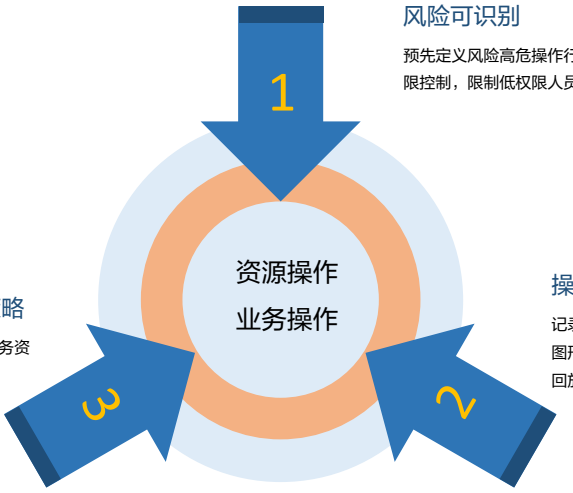
为什么需要操作全程可管控



HUAWEI

- 为什么需要全程可监管?
- 如果操作没有管控，操作人员的误操作、高危操作、恶意操作导致的业务中断行为将无法控制。哪些操作运维人员可以执行如果不加以限制，删库跑路将不再只是一个“网络段子”。

什么是操作全程可管控



风险可识别

预先定义风险高危操作行为，对于高危行为进行权限控制，限制低权限人员执行高危操作。

操作可审计

记录操作人员的字符（Telnet、SSH）、图形化操作（RDP、VNC），操作过程可回放，实现事后追溯和举证。

安全策略

通过严格的策略限制可以访问业务资源环境的人员，杜绝越权访问。

HUAWEI

- 什么是操作全程可管控
  - 资源操作+业务操作
- 如何实现
  - 配置安全策略
  - 风险识别和处置
  - 操作可审计、追溯

资源操作可管控

NO.1 风险可识别

SA-云服务基线管理，提供了“身份认证”、“访问控制”、“日志审计”、“数据安全”、“基础防护”这五方面的风险。

NO.2 操作可审计

云审计服务（Cloud Trace Service，以下简称CTS），通过配置追踪器云审计服务可以实现安全审计>问题定位>资源追踪。

NO.3 安全策略

统一身份认证（Identity and Access Management，简称IAM）：策略根据授权的精细程度，分为细粒度策略和Role-Based Access Control（RBAC）策略。

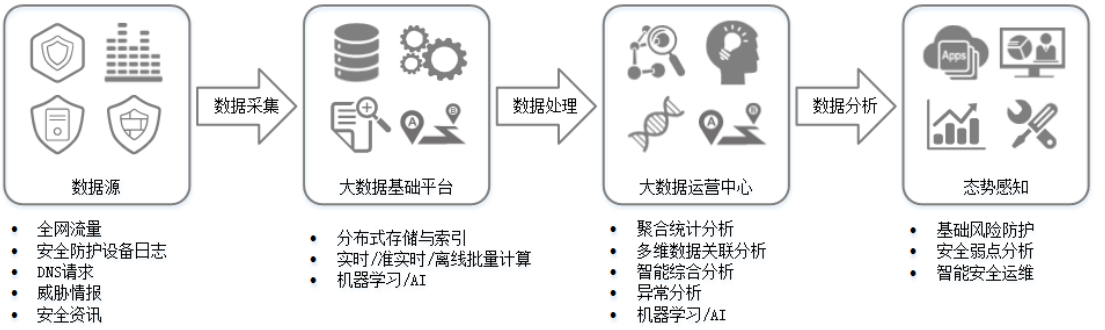


HUAWEI

- 用户可通过云服务基线查看各项风险的详细信息和指导建议。
  - 身份认证
    - 检查是否启用IAM用户
    - 检查租户的用户列表，是否至少有两个已启用的用户，且其中一个用户所属的用户组不是admin用户组。启用IAM用户，是指通过IAM服务启用除企业管理员以外的其他用户，且该用户不属于admin组。
  - 访问控制
    - 网络ACL规则检查
    - 检查所有网络ACL的规则是否存在不安全规则，即是否存在开放过大的访问控制策略。不安全规则，即方向为入方向，动作为允许，协议为任一类别协议，源地址为0.0.0.0/0（所有地址），源端口范围为1-65535或0，目的地址为0.0.0.0/0（所有地址），目的端口范围为1-65535、0或者特定的业务端口，如22。
  - 安全组规则检查
    - 检查所有安全组的规则是否存在不安全规则，即是否存在开放过大的访问控制策略。不安全规则，即方向为入方向，协议为任一类别协议，源地址为0.0.0.0/0（所有地址），端口为1-65535或者特定的业务端口，如22。

态势感知 SA

- 态势感知功能：总览、态势感知大屏、威胁检测、威胁分析、安全编排、主机漏洞扫描、网站漏洞扫描、应急漏洞、主机基线检查、云服务基线检查、告警设置、日志管理



- 全网流量
- 安全防护设备日志
- DNS请求
- 威胁情报
- 安全资讯

- 分布式存储与索引
- 实时/准实时/离线批量计算
- 机器学习/AI

- 聚合统计分析
- 多维数据关联分析
- 智能综合分析
- 异常分析
- 机器学习/AI

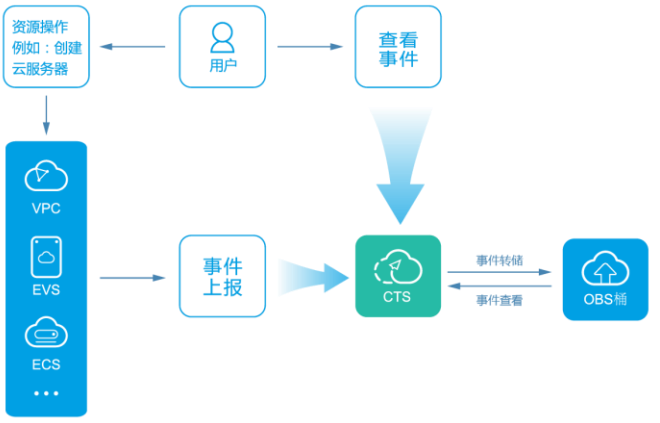
- 基础风险防护
- 安全弱点分析
- 智能安全运维

HUAWEI

- 态势感知（Situation Awareness，SA）是可视化威胁检测和分析的平台。态势感知能够检测出超过20大类的云上安全风险，包括DDoS攻击、暴力破解、Web攻击、后门木马、僵尸主机、异常行为、漏洞攻击、命令与控制等。利用大数据分析技术，态势感知可以对攻击事件、威胁告警和攻击源头进行分类统计和聚合分析，为用户呈现出全局安全攻击态势。
- 通过采集全网流量数据和安全防护设备日志信息，并利用大数据安全分析平台进行处理和分析，态势感知检测出威胁告警，同时将主机安全、Web防火墙和DDoS流量清洗等安全产品上报的告警进行汇聚，最终为用户呈现完整的全网攻击态势，进而为安全事件的处置决策提供依据。
- 图片来源：华为云

云审计服务 CTS

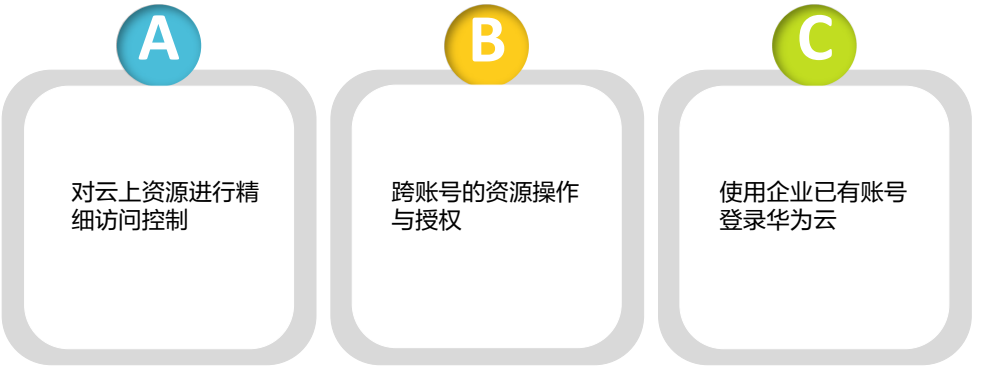
- 云审计服务的功能主要包括：
  - 记录审计日志
  - 审计日志查询
  - 审计日志转储
  - 事件文件加密
  - 关键操作通知



HUAWEI

- 日志审计模块是信息安全审计功能的核心必备组件，是企事业单位信息系统安全风险管控的重要组成部分。在信息系统逐步云化的背景下，包括我国SAC/TC在内的全球各级信息、数据安全管理部门已对此发布多份标准，如：ISO IEC27000、GB/T 20945-2013、COSO、COBIT、ITIL、NISTSP800等。
- 云审计服务（Cloud Trace Service，以下简称CTS），是华为云安全解决方案中专业的日志审计服务，提供对各种云资源操作记录的收集、存储和查询功能，可用于支撑安全分析、合规审计、资源跟踪和问题定位等常见应用场景。
- 云审计服务的功能主要包括：
  - 记录审计日志：支持记录用户通过管理控制台或API接口发起的操作，以及各服务内部自触发的操作。
  - 审计日志查询：支持在管理控制台对7天内操作记录按照事件来源、事件名称、操作类型、资源名称/ID、事件状态和时间范围等多个维度进行组合查询。
  - 审计日志转储：支持将审计日志周期性的转储至对象存储服务（Object Storage Service，简称OBS）下的OBS桶，转储时会按照服务维度压缩审计日志为事件文件。
  - 事件文件加密：支持在转储过程中使用数据加密服务（Data Encryption Workshop，简称DEW）中的密钥对事件文件进行加密。
  - 关键操作通知：支持在检测到部分关键操作时，使用消息通知服务（Simple Message Notification，简称SMN）向用户发送邮件、短信通知。
- 图片来源：华为云

统一身份认证 IAM



HUAWEI

- 统一身份认证（Identity and Access Management，简称IAM）是华为云提供权限管理的基础服务，可以帮助您安全地控制华为云服务和资源的访问权限。
- IAM无需付费即可使用，您只需要为您账号中的资源进行付费。

DevCloud成员权限管理

- DevCloud权限管理

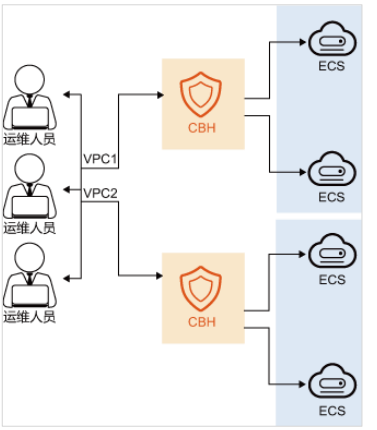


HUAWEI

- 为了保证代码安全，DevCloud在代码托管服务设置了成员管理功能：成员项目成员必须成为代码仓库成员，才能查看仓库中的代码。代码仓库管理员可以邀请该仓库所在项目的成员成为仓库成员。

云堡垒机 CBH

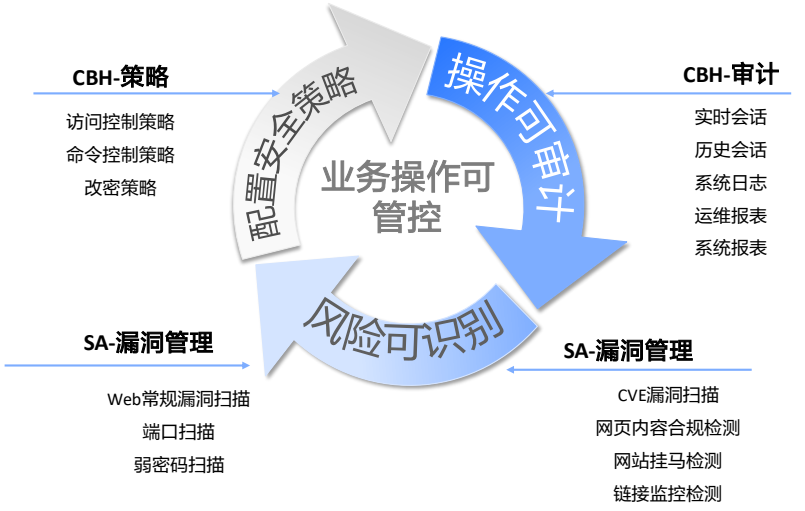
- 云堡垒机目前支持云堡垒机实例和运维专家服务两大服务模块。
- 云堡垒机实例对应一个独立运行的云堡垒机系统，云堡垒机系统具备核心系统运维和安全审计管控功能，符合安全合规审查要求，为用户提供安全统一的运维管理平台。
  - 运维人员一次登录，便可对多台资源进行一站式管理。
  - 实时记录所有运维操作和日志，便于审计。
  - 符合“网络安全法”等法律法规，满足合规性规范审查要求。



HUAWEI

- 运维云堡垒机提供了用户管理、用户组管理、角色管理，认证方式支持账号和密码、USBKey和动态令牌多种令牌认证模式。
- 云堡垒机提供了访问控制策略、命令控制策略和改密策略等三种方式。

业务操作可管控



HUAWEI

- 专家服务是华为基于多年运维实战经验，为企业用户提供的实时人工专家咨询服务。

目录

1. DevSecOps简介
2. **安全策略与工具**
   - 操作和资源可管控
   - **编码安全与工具**
   - 漏洞扫描与工具
   - 安全运维与工具

HUAWEI

安全编码理念

- **安全编码(Secure Coding)**是一种开发计算机软件的实践，可以防止意外引入安全漏洞。缺陷，错误和逻辑缺陷始终是导致可被普遍利用的软件漏洞的主要原因。



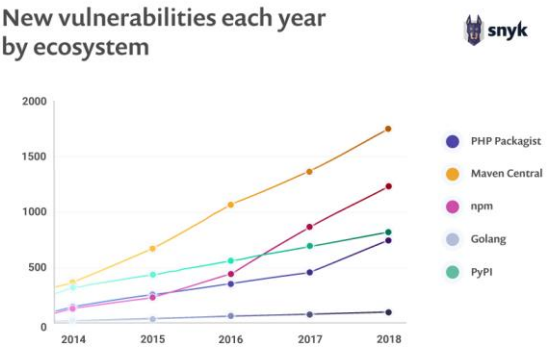
让产品生来安全

- 客户迫切需要更安全的软件。
- 导致敏感、机密或个人身份数据泄露的安全bug属于隐私类问题，并可能导致法律后果。
- 引发可靠性问题的安全bug则可能减少系统正常运行时间，并致使系统违反服务水平协议（SLA）。
- 安全与开发质量息息相关，如果开发过程控制良好，后期修补缺陷的费用会急剧减少。

HUAWEI

软件组成分析

- 许多组织都使用WordPress，Magento，Drupal甚至jQuery之类的开源库/框架/解决方案，它们每天都在发现新的漏洞。



HUAWEI

- 2017年9月，一个受欢迎的信用评级机构Equifax遭到破坏，导致1.43亿张信用卡号，个人身份信息（PII）等泄露。在构建主要Web应用程序的“ Struts2 Web Framework”（缺少安全补丁）中发现了被利用的漏洞。就像Equifax一样。

软件研发安全的目标



确保数据不被非法访问与窃取

确保数据是完整的、未被篡改的

要求保护资源在需要时可访问

HUAWEI

典型案例

C/C++案例：

- 字符串操作不当带来的风险
- 源字符串长度大于目标缓冲区
- 危险的堆操作及带来的风险
- 命令注入
- 不安全函数使用

JAVA案例：

- mybatis使用不当导致SQL注入风险
- OS命令注入
- 字符串操作不当带来的风险
- 目录遍历攻击
- XML文件使用相关安全风险

HUAWEI

- 危险的堆操作
  - 内存拷贝时未判断目标内存长度的有效性（目标内存过小）
  - 内存申请完毕后未判断空指针（空指针引用）
  - 使用已经释放的内存
  - 调用不匹配的内存管理操作（new, delete与malloc , free混用）
  - 重复释放内存（double free）
- 堆管理不当带来的风险
  - 拒绝服务攻击：由于堆被破坏，可导致程序崩溃；如果被攻击者通过恶意输入控制这种情况，攻击者可以使程序崩溃从而让合法用户无法继续使用。
  - 执行任意代码：攻击者输入的恶意数据被当做指令执行。
- 不安全函数是如memcpy、strcpy，Linux系统都整理输出了它的一个替换的安全函数集，如memcpy替换成memcpy_s，通常都是以_s结尾。

典型错误 - 重复释放内存

- 重复释放内存

【错误示例】

```
char * ptr = (char*)malloc(SIZE);
...
if (abrt)
{
   free(ptr);
}
...
free(ptr);
/*【错误】可能会发生内存的重复释放*/
```



【正确示例】

```
char* ptr = (char*)malloc(SIZE);
...
if (abrt)
{
   free(ptr);
   return;
   /*【修改】添加return，确保重复释放不会发生*/
}
...
free(ptr);
ptr = NULL;
```



HUAWEI

- 释放内存时需要注意是否是已释放过的内存，注意不要重复释放。
- 重复释放内存（double-free）会导致内存管理器出现问题。在一定情况下重复释放内存，有可能导致“堆溢出”漏洞，可以被用来执行恶意代码，具有很大的安全隐患。

典型错误 - 直接执行命令，未对命令字符串做校验

- 使用Runtime.exec()调用命令解释器执行命令时，未对命令字符串做校验

【错误示例】

```
//示例代码
class DirList
{
   public static void main(String[] args) throws Exception
   {
      String dir = System.getProperty("dir");
      Runtime rt = Runtime.getRuntime();
      Process proc = rt.exec("cmd.exe /C dir " + dir);
      // ...
   }
}
```

攻击者可在“dir”属性中包含恶意的命令，然后被Runtime.exec()执行

【正确示例】

```
//示例代码
class DirList {
   public static void main(String[] args) throws Exception {
      File dir = new File(System.getProperty("dir"));
      if (!dir.isDirectory()) {
         System.out.println("Not a directory");
      }
      else {
         for (String file : dir.list()) {
            System.out.println(file);
         }
      }
   }
}
```

对于获取目录下文件列表的操作，jdk中已经提供了api实现，可避免使用Runtime.exec()

HUAWEI

- 系统中应该尽量避免使用Runtime.exec()，通常建议使用系统的JDK提供的API来实现对应的功能，若涉及到一些JDK未提供API的平台强相关的特性（如windows注册表访问），可优先考虑使用本地方法调用实现。

开源静态代码分析工具

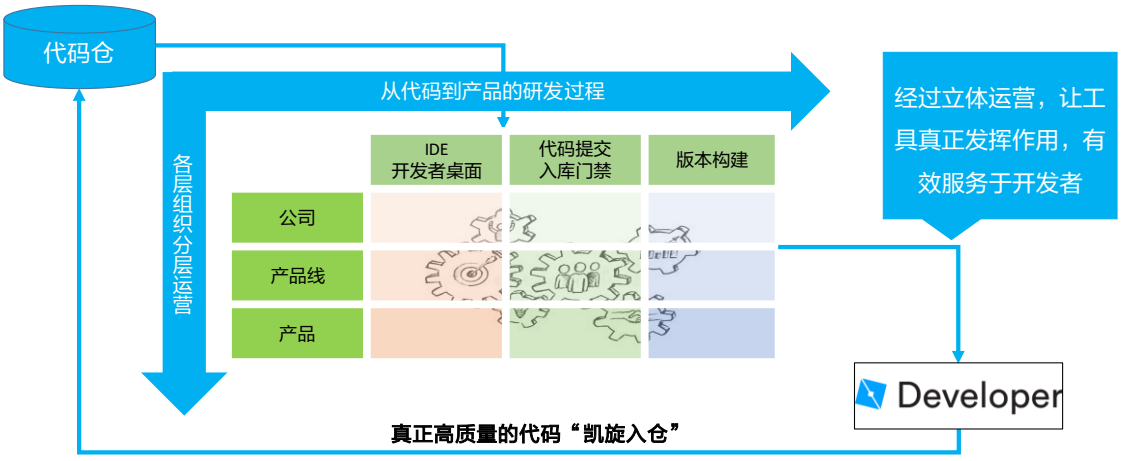
- 使用自动化工具执行安全代码审查可以清除许多低挂的果实，例如SQL注入，跨站点脚本，反序列化漏洞等等。

Scanners	Java	.NET	C, C++	Python	Golang	Ruby/Rails	PHP
Bandit	X	X	X	√	X	X	X
Brakeman	X	X	X	X	X	√	X
Codesake Dawn	X	X	X	X	X	√	X
FindBugs	√	X	X	X	X	X	X
FindSecBugs	√	X	X	X	X	X	X
Flawfinder Flawfinder	X	X	√	X	X	X	X
Graudit	√	√	X	√	X	√	√
LGTM	√	X	√	√	X	X	X
Progpilot	X	X	X	X	X	X	√
PreFast (Microsoft)	X	X	√	X	X	X	X
Puma Scan	X	√	X	X	X	X	X
.NET Security Guard	X	√	X	X	X	X	X
RIPS	X	X	X	X	X	X	√
phpcs-security-audit	X	X	X	X	X	X	√
SonarQube	√	√	√	√	√	√	√
VisualCodeGrepper (VCG) -	X	X	√	X	X	X	√

HUAWEI

- 对于基于Java的应用程序，我们可以使用一个名为“ FindSecBugs”的工具，该工具对代码进行深入分析（而不会给出过多的误报），并针对已在代码中识别出的所有漏洞提供全面的报告。可以根据漏洞的数量和严重性来做出决定，以继续运行管道或在管道可以继续之前无法修复漏洞。

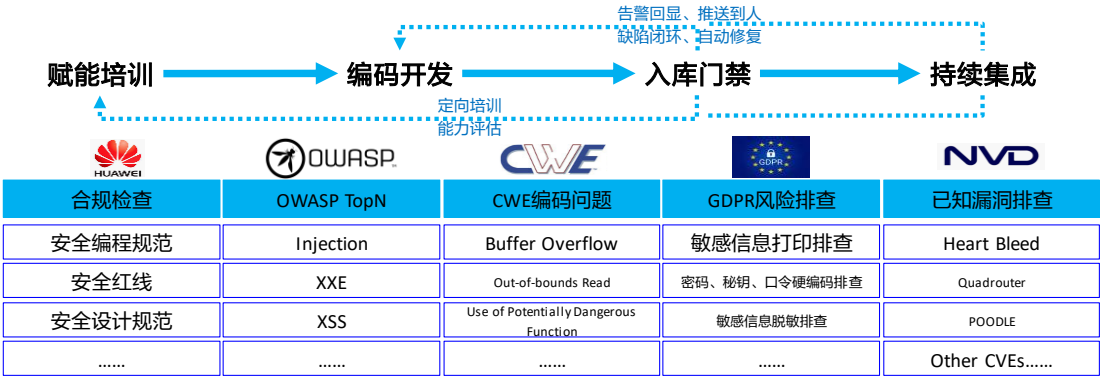
立体的华为云安全编码检查服务 - 华为运营实践



HUAWEI

- 华为建立了从开发者桌面到版本构建各级检查、从公司到产品的各组织层面立体运营，在“九宫格”的每一段都有各自不同的策略，同时又有延续性，保证工具的检查效果，同时又不过份打扰开发者，让工具真正有用。

安全编码检查工具需要具备的能力



合规检查	OWASP TopN	CWE编码问题	GDPR风险排查	已知漏洞排查
安全编程规范	Injection	Buffer Overflow	敏感信息打印排查	Heart Bleed
安全红线	XXE	Out-of-bounds Read	密码、秘钥、口令硬编码排查	Quadrouter
安全设计规范	XSS	Use of Potentially Dangerous Function	敏感信息脱敏排查	POODLE
……	……	……	……	Other CVEs……



HUAWEI

- 流程支撑
  - 从开发人员桌面至版本构建，提供嵌入流程的全自动安全编码检查。
  - 根据检查结果提供部分自动修复能力，并定向推送培训服务。
- 检查服务
  - 支持各类华为安全规范、checklist的自动化检查。
  - 通过词法、语法、语义、数据流、控制流分析，具备业界知名安全编码问题全种类检查 拦截能力。
  - 通过工程定制、AI等技术手段控制检查误报率远好于业界平均水平。
- 能力扩展
  - 各类安全知识、能力例行完善，有效归纳，支撑上层检查能力，并可向最终用户提供有针对性的知识参考。
  - 发现的问题、告警可审计、可管理。
  - 能力、规则可定制、可扩展，灵活应对各类检查场景；精确制导，快速响应拦截需求，分层定制。

静态代码检查



HUAWEI

- 代码检查（CodeCheck）是基于云端实现代码质量管理的服务，软件开发者可在编码完成后执行多语言的代码静态检查和安全检查，获取全面的质量报告，并提供缺陷的改进建议和趋势分析，有效管控代码质量，帮助产品成功。

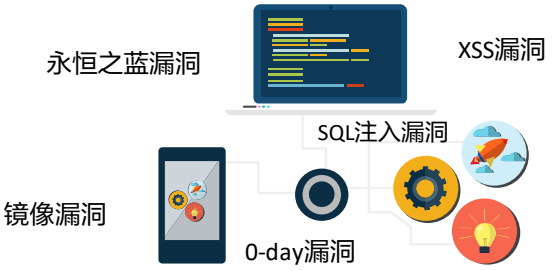
目录

1. DevSecOps简介
2. **安全策略与工具**
   - 操作和资源可管控
   - 编码安全与工具
   - **漏洞扫描与工具**
   - 安全运维与工具

HUAWEI

什么是漏洞

- 漏洞或脆弱性（Vulnerability），是指计算机系统安全方面的缺陷，使得系统或其应用数据的保密性、完整性、可用性、访问控制等面临威胁。
- 在《GB/T 25069-2010 信息安全技术术语》，将脆弱性定义为“资产中能被威胁所利用的弱点”。





HUAWEI

- 漏洞会影响到很大范围的软硬件设备，包括操作系统本身及其支撑软件，网络客户和服务器软件，路由器和安全防火墙等。换而言之，在这些不同的软硬件设备中都可能存在不同的安全漏洞问题。
- 常见漏洞按照分类可以分为三大类：主机漏洞、Web漏洞和数据库漏洞。
- 主机漏洞：内存破坏类漏洞、CGI类漏洞、输入验证类漏洞、配置错误类漏洞、系统本地补丁、常见协议弱口令、木马病毒。
- Web漏洞：SQL注入攻击、跨站脚本、文件包含、远程代码执行、主流CMS漏洞。
- 数据库漏洞：常见类型数据库漏洞，如Oracle、MySQL、SQL Server、DB2、Informix、Sybase。
- 我们经常听到的比如SQL注入漏洞、XSS漏洞，是2种web应用上的漏洞，永恒之蓝漏洞，发生在Windows系统上的越权漏洞， 0-day漏洞，是已经被发现(有可能未被公开)，而官方还没有相关补丁的漏洞。镜像漏洞：是指Docker这类容器化技术中镜像所包含的漏洞。这是指这些都是漏洞不同分类产生的名称。

漏洞分类



HUAWEI

- 基于威胁类型分类：
  - 获取控制：劫持执行程序流程，使程序执行指定任意指令或命令，控制应用系统或操作系统。
  - 获取信息：劫持程序访问预期外的资源并被攻击者获取，影响系统的机密性。
  - 拒绝服务：导致目标应用或系统暂时或永久性失去相应服务的能力。
- 基于技术类型分类：
  - 内存破坏类：栈缓冲区溢出，堆缓冲区溢出，静态数据区溢出。
  - 错误逻辑类：安全检查的实现逻辑上上有问题。
  - 设计错误类：系统设计上对安全机制的考虑不足导致在设计阶段引入安全漏洞。
  - 错误配置类：系统运维过程中默认不安全的配置状态。
  - 输入验证类：对用户输入没有做充分的检查过滤就用于后续操作导致的漏洞。

近年来关于web漏洞排名

- 下图展示了OWASP在2013年版与2017年版中统计的Top 10 web安全风险变化。

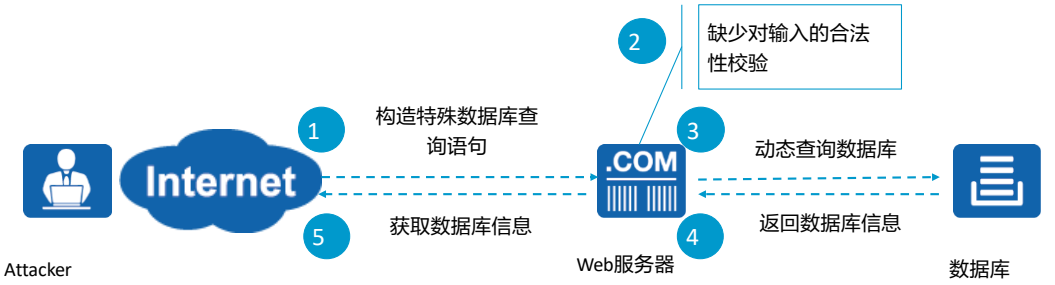
2013年版《OWASP Top 10》		2017年版《OWASP Top 10》
A1 – 注入	→	A1:2017 – 注入
A2 – 失效的身份认证和会话管理	→	A2:2017 –失效的身份认证
A3 – 跨站脚本（XSS）	↘	A3:2017 –敏感信息泄漏
A4 – 不安全的直接对象引用 [与A7合并]	∪	A4:2017 – XML外部实体（XXE） [新]
A5 – 安全配置错误	↘	A5:2017 – 失效的访问控制 [合并]
A6 – 敏感信息泄漏	↗	A6:2017 – 安全配置错误
A7 – 功能级访问控制缺失 [与A4合并]	∪	A7:2017 – 跨站脚本（XSS）
A8 – 跨站请求伪造 （CSRF）	☒	A8:2017 – 不安全的反序列化 [新，来自于社区]
A9 – 使用含有已知漏洞的组件	→	A9:2017 –使用含有已知漏洞的组件
A10 – 未验证的重定向和转发	☒	A10:2017 – 不足的日志记录和监控 [新，来自于社区]

HUAWEI

- OWASP(Open Web Application Security Project)是一个非盈利的全球性安全组织，致力于应用软件的安全研究。其发布的OWASP Top10会统计当年10大web安全威胁。
- OWASP Top10漏洞体系长期以来作为Web安全的一个参考标准，其中注入、XSS长期出现其中，是web安全中最常见攻击行为。

SQL注入

- SQL注入是指将不受信任的数据作为命令或查询的一部分发送到解析器时，会产生注入缺陷。攻击者的恶意数据可以诱使解析器在没有适当授权的情况下执行非预期命令或访问数据。



HUAWEI

- SQL注入能导致数据丢失、破坏或泄露给无授权方，缺乏可审计性或是拒绝服务。注入有时甚至能导致主机被完全接管。
- SQL注入的危害很大，常见的防御措施有：
  - 净化用户输入
  - SQL语句使用参数化查询
  - 对SQL语句采用预编译等

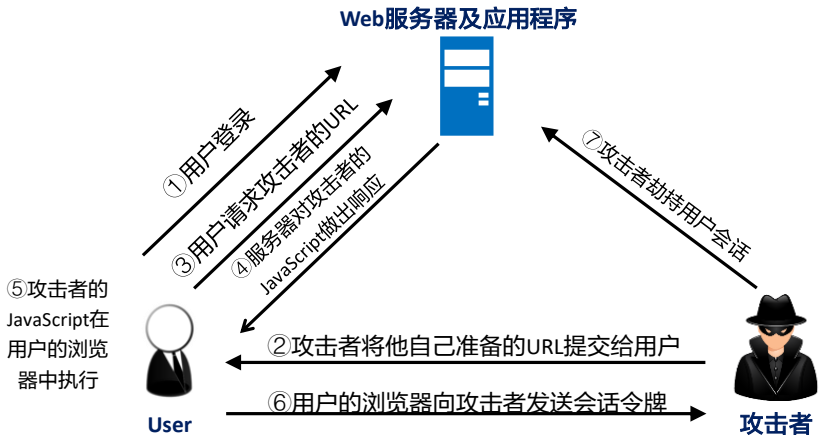
跨站脚本

- 跨站脚本（XSS）是近年来最为流行的网络攻击方式之一，已经占到了网络攻击相当大的比例，众多网站，如著名的Facebook等都遭遇过此类攻击。国内知名的新浪微博遭遇的网络病毒实际上也是跨站脚本漏洞所导致的。
- 从本质上看，跨站脚本实际上是一种恶意代码执行的方式。主要原因在于网站对于用户提交的数据过滤不严格，导致问题产生。

HUAWEI

- XSS起初的危害主要是盗取用户的Cookie信息。Cookie信息中包含了浏览者和网站服务器之间的常用记录，包括登录记录、浏览记录等。如果攻击者得到了用户的Cookie信息，可以模仿用户和网站进行交互，得到更多想要的数据。
- 例如，攻击者在某论坛上发布一个URL链接，当用户点击该链接后，用户的浏览器就会将Cookie信息发送到攻击者的网站。攻击者收集到Cookie信息后可以以浏览者身份访问网站、进入邮箱等，继续窃取私人信息。它常见的危害包括发生钓鱼攻击、会话劫持或者被植入Web蠕虫。
- 跨站脚本简称为Cross Site Scripting，因为容易和另一个名词“层叠样式表”（Cascading Style Sheets，CSS）混淆，为了区别，网络安全人士习惯将其简称为XSS攻击。
- 反射型
  - 跨站代码一般存在于链接中，请求这样的链接时，跨站代码经过服务端反射回来，这类跨站的代码一般不存储到服务端。
- 存储型
  - 这是利用起来最方便的跨站类型，跨站代码存储于服务端（比如数据库中）。
- DOM型：
  - OM xss是基于dom文档对象模型，前端脚本通过dom动态修改页面，由于不与服务端进行交互，而且代码是可见的，从前端获取dom中的数据在本地执行。
    常见的可以操纵dom的对象：URL，localtion,referrer等
- 还有一个常见的攻击叫跨站伪造请求CSRF，从攻击方式上分，它并不属于XSS攻击。

反射型XSS攻击的实施步骤

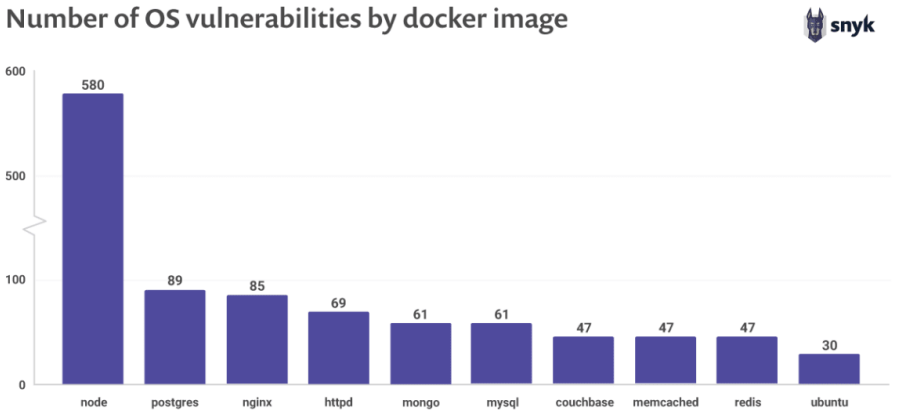


HUAWEI

- 用户正常登陆Web服务器，会得到一个包含session的cookie，现在很多网站会把识别用户身份的session放在cookie内，后续客户的访问携带该session，即可不需要再次登陆。
- 攻击者通过某种方式让客户打开一个包含攻击信息的页面，比如发送广告邮件。
- 用户打开该URL之后会发现是Web服务器上的一个正常页面。
- 但是该正常页面的html标签内包含了JS脚本，通过JS脚本可以读取用户的cookie的值，并发送到攻击者。
- JS脚本可能是包含在图片标签内，该JS脚本执行的行为:去第三方站点加载一个图片链接，该链接地址指向攻击者，并在链接后加上类似document.cookie的方法，这样用户执行JS脚本就会向攻击者请求图片并携带Web服务器的cookie。
- 攻击者收到针对图片的请求，并获取到cookie，响应图片请求之后，用户侧图片正常显示，但是此时cookie已经被传递给了攻击者。
- 攻击者获取cookie之后，利用其中的session信息，就可以进行会话劫持，仿冒用户登陆站点，并进行后续攻击行为。

Docker镜像漏洞

- Docker Hub 上最流行的 10 个镜像都包含已知的安全漏洞

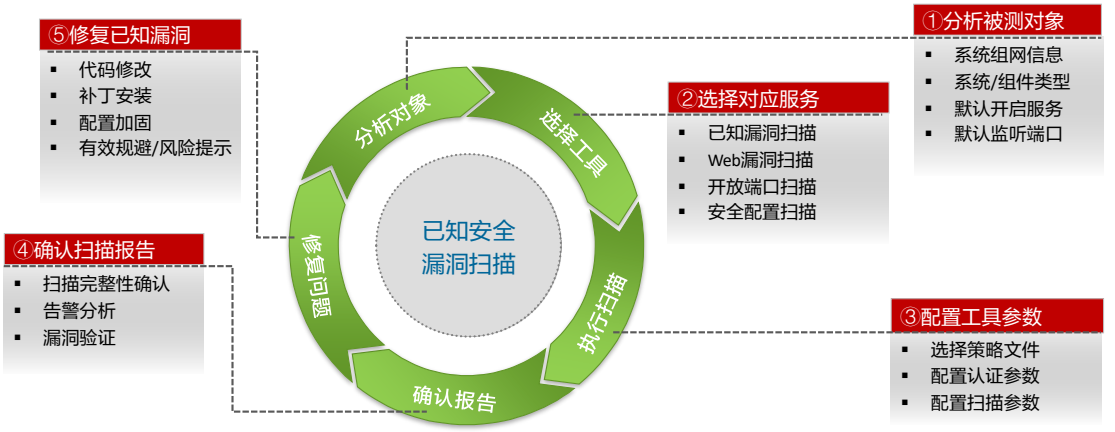


HUAWEI

- 上图（摘自《开源安全报告-2019》）可知，Docker Hub 上最流行的 10 个镜像都包含已知的安全漏洞。选用最小化基础镜像，即只包含项目确实需要的系统工具和库的镜像，就能最小化系统的攻击面，确保所用操作系统是安全的。

漏洞扫描基本理念

- 安全测试阶段的安全漏洞扫描主要包括主机漏洞扫描、Web漏洞扫描、端口扫描、安全配置合规检查等内容。



HUAWEI

- 安全测试阶段的安全漏洞扫描主要包括主机漏洞扫描、Web漏洞扫描、端口扫描、安全配置合规检查等内容。
- 主机漏洞扫描主要是针对目标主机系统，对系统上运行的操作系统、数据库、web等组件进行漏洞扫描，通过识别组件版本，与已有漏洞库进行匹配，检测目标系统是否包含已知漏洞，或者通过对漏洞进行验证，检测目标系统是否存在漏洞。
- Web漏洞扫描主要是针对运行的web应用进行扫描，他往往会先获取Web应用的所有URL，并对URL进行漏洞检查，检测是否存在SQL注入、XSS攻击等问题。
- 开放端口扫描：往往会和主机漏洞扫描集成在一起，主要是检测整个系统对外开发的端口，所有开发端口都可能存在潜在的安全风险，开发及运维人员需要明确哪些那段是必须的，哪些是可以关闭的。
- 安全配置检查：安全配置漏洞不是指软硬件或者协议本身设计或实现上的问题，而是由于软硬件不正确的配置和部署所造成的。安全配置检查基于业界已有的安全配置最佳实践进行扫描，提前发现安全配置漏洞并进行安全加固。
- 漏洞扫描通常的执行流程包括：分析被测对象，包括组网、组件信息、开发端口服务等，选择对应的扫描服务，配置正确的参数，扫描后，对扫描结果进行分析确认，并确定消减措施后实施加固，再执行扫描，已确认对应的风险已经消减。我们需要进行很多轮的扫描，直到已知的风险都已消减或者遗留的风险可控。

为什么要进行漏洞扫描

- 产品上线后遭到黑客攻击，影响产品可用性，大量用户投诉
- 产品数据库被入侵，导致大规模数据泄露，面临法律风险
- 产品漏洞被利用，导致业务瘫痪并波及内网，引发灾难性事故
- 产品无法通过政府信息安全等级标准，项目竞标失败
- 产品不满足行业、地区软件准入要求，无法进入目标市场
- 软件交付项目漏洞多，客户不认可，项目款回收难
- 产品在维保期内安全问题投诉不断，售后服务成本超支
- 产品安全问题引发舆论广泛关注，企业声誉受损
- ……

HUAWEI

漏洞扫描典型案例

01 使用有漏洞的三方件

02 协议使用不安全算法

03 Linux口令复杂度不合规

04 跨站点脚本（XSS）

05 SQL注入

HUAWEI

主机漏洞扫描案例 - 使用有漏洞的三方件

- ✓ 业界安全研究人员发现开源漏洞，生成漏洞CVE编号
- ✓ 系统厂商分析各自产品是否受影响，发布漏洞补丁及安全公告
- ✓ 安全工具基于业界系统厂商的安全公告制定检查规则，比对被测目标的组件版本是否受漏洞影响
- ✓ 测试人员基于报告信息，验证存在漏洞，打补丁修复

shellshock漏洞允许用户执行任意代码

CVE-ID
CVE-2014-6271， CVE-2014-7169
Description
GNU Bash through 4.3 processes trailing strings after function definitions in the values of environment variables, which allows remote attackers to execute arbitrary code via a crafted environment ******, aka "**ShellShock**".

SUSE 安全公告		
SUSE Linux Enterprise Server 11 SP3	•bash >= 3.2-147.22.1 •bash-doc >= 3.2-147.22.1 •bash-x86 >= 3.2-147.22.1 •libreadline5 >= 5.2-147.22.1 •libreadline5-32bit >= 5.2-147.22.1 •libreadline5-x86 >= 5.2-147.22.1 •readline-doc >= 5.2-147.22.1	Patchnames slessp3-bash

主机漏洞扫描工具基于版本比对报告被测系统存在该漏洞

	ComponentName	InstallVersion	FixedVersion
output	bash	3.2-147.17.30	3.2-147.22.1
	bash-doc	3.2-147.17.30	3.2-147.22.1
	libreadline5	5.2-147.17.30	5.2-147.22.1
	readline-doc	5.2-147.17.30	5.2-147.22.1

修复方法：升级bash组件 yum update bash

登入系统，可验证存在该漏洞

```
linux-0u8f:/ # rpm -qa | grep bash
bash-3.2-147.17.30
bash-doc-3.2-147.17.30
linux-0u8f:/ # rpm -qa | grep readline
libreadline5-5.2-147.17.30
readline-doc-5.2-147.17.30
linux-0u8f:/ #
```

经验总结 使用第三方组件，需要跟踪第三方组件的官方安全公告信息，及时联系厂商获取最新安全补丁，修复漏洞。

HUAWEI

- 案例一：主机漏洞扫描中针对操作系统补丁的检查。业界安全研究人员发现开源漏洞并生成漏洞CVE编号，系统厂商分析各自产品是否受影响，如果受影响，则会发布漏洞补丁及安全公告，安全工具基于业界系统厂商的安全公告制定检查规则，比对被测目标的组件版本是否受漏洞影响，测试人员基于报告信息，验证存在漏洞，并按要求打补丁升级组件修复该漏洞。
- 这里的例子，说明操作系统上的BASH命令，有shellshock漏洞，允许用户执行任意代码，从而在未授权的情况下任意访问系统。

主机漏洞扫描案例 - 协议使用不安全算法

- ✓ 业界安全研究人员发现并论证曾经广泛使用的某些密码算法不再安全。
- ✓ 安全工具基于业界共识对此类不安全的密码算法制定检查规则，探测被测目标相关服务所使用的是否为不安全算法。
- ✓ 测试人员基于报告信息，验证存在漏洞，修改服务配置或业务代码进行修复。

华为密码算法应用规范	
2.4.2 规则：禁止在密码协议中使用截短的消息认证码	说明：密码协议中（如TLS、SSH、IKE等），握手阶段往往使用消息认证码（MAC）验证消息的完整性，协议标准有时支持选取截短的消息认证码。此时，消息认证码安全性也因截短而降低，如针对多种密码协议（如TLS、SSH等）的SLOTH攻击就可以利用截短哈希值构造碰撞。 禁止使用hmac-md5以及hmac-md5-96/hmac-sha1-96等截短哈希

主机漏洞扫描工具基于报文交互得到SSH支持的MAC算法判断漏洞

Vul Detect Result

The following weak client-to-server MAC algorithms are supported by the remote service:

hmac-md5-96
hmac-sha1-96

登入验证：登入系统，查看SSH服务配置验证存在该漏洞；

```
linux-0u8f:/ # grep "MACs" /etc/ssh/sshd_config
MACs hmac-sha2-256,hmac-sha1-96,hmac-md5-96
linux-0u8f:/ #
```

修复方法：改为 MACs hmac-sha2-512,hmac-sha2-256

远程验证：SSH连接被测设备，抓取报文验证存在该漏洞

Source	Destination	Protoc	Leng	Info
10.175.43.209	10.184.38.1…	SSH…	870	Server: Key Exchange Init

```
mac_algorithms_client_to_server length: 38
mac_algorithms_client_to_server string: hmac-sha2-256,hmac-sha1-96,hmac-md5-96
mac_algorithms_server_to_client length: 38
mac_algorithms_server_to_client string: hmac-sha2-256,hmac-sha1-96,hmac-md5-96
```

经验总结 对于端口矩阵上的各TCP/UDP服务，均需覆盖扫描，验证工具探测得到的服务类别、服务版本、服务使用的算法与系统设计一致，不应使用已知的不安全密码算法。

HUAWEI

- 案例二：扫描开发协议中使用的不安全算法。业界安全研究人员发现并论证曾经广泛使用的某些密码算法不再安全，比如TLS1.0、SSL2.0/SSL3.0等，同时，针对这类协议中使用的算法套，随着技术的进步，很多算法也已不安全，比如MD5、SHA1等。安全工具基于业界共识对此类不安全的密码算法制定检查规则，探测被测目标相关服务所使用的是否为不安全算法。
- 针对协议的扫描，除了不安全算法，还有协议本身已经不安全，比如安装tftp提供ftp服务、开启telnet服务。

安全配置扫描案例 - Linux口令复杂度不合规

- ✓ 简单的口令容易被暴力破解工具猜测出来，在系统设置的时候，应强制用户使用健壮口令。
- ✓ 口令必须包含大写和小写字母、数字、特殊字符，业界规范有要求密码长度至少14个字符。
- ✓ 根据客户需求和使用场景，设计符合业界标准要求的口令复杂度规则。

华为Linux安全配置操作规范	2.3.7 规则：设置用户账号口令的复杂度
说明： 口令不能太容易被猜测，太短的口令容易被暴力破解工具猜测出来。纯数字或纯字母的口令容易被暴力破解工具猜测出来，根据应用需要可提高复杂度。口令必须包含大写和小写字母、数字、特殊字符，至少8个字符长度，这是最低的口令复杂度要求，推荐采用更长的14个字符的口令，以满足业界知名规范要求。产品应根据客户需求和使用场景，设计符合业界标准要求的口令长度。	实施指导： 1.通过修改/etc/login.defs文件，限制口令长度，添加以下内容： PASS_MIN_LEN 14 //14或8，推荐采用更长的14个字符的口令 2. 通过修改/etc/pam.d/passwd文件，限制口令长度及复杂度，添加以下内容： password required pam_cracklib.so try_first_pass retry=3 minlen=14 dcredit=-1 ucredit=-1 ocredit=-1 lcredit=-1 \\长度不小于14或8，推荐采用更长的14个字符的口令，至少包含一个数字，一个大写字母，一个特殊字符，一个小写字母

问题确认：登录系统shell，执行cat /etc/login.defs 和 cat /etc/pam.d/passwd 命令，查看实际配置内容不满足配置要求

```
linux:~ # cat /etc/login.defs  | grep PASS_MIN_LEN
linux:~ #
```

```
linux:~ # cat /etc/pam.d/passwd | grep password
password include       common-password
linux:~ # cat /etc/pam.d/common-password | grep pam_cracklib.so
password requisite pam_cracklib.so try_first_pass retry=3 minlen=8 minclass=3 dcredit=0 ucredit=0 ocredit=0 lcredit=0
linux:~ #
```

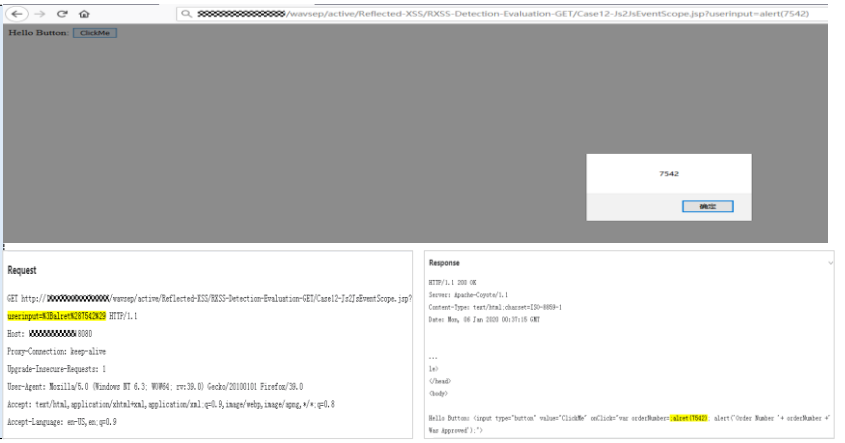
加固方案：对使用PAM模块的Linux系统，可以修改PAM规则，使用支持口令复杂度检查的插件（如pam_cracklib.so）实现口令复杂度检查（password requisite pam_cracklib.so try_first_pass retry=3 minlen=14 dcredit=-1 ucredit=-1 ocredit=-1 lcredit=-1 ）

HUAWEI

- 案例三：安全配置扫描：针对配置安全，最典型的应该是口令复杂度。简单的口令容易被暴力破解工具猜测出来，在系统设置的时候，应强制用户使用健壮口令。
- 针对口令的加固，除了配置密码复杂度，要实现防暴力破解，还可以实施下列措施：登陆延迟、锁定账号、验证码、IP白名单+告警或日志。
- 对口令的保护，除了配置复杂度，口令本身也要做保护，如口令要进行加密存储。

Web漏洞扫描案例 - 跨站点脚本攻击XSS

- ✓ XSS是指恶意攻击者往Web页面里插入恶意Script代码，当用户浏览该页之时，嵌入其中Web里面的Script代码会被执行，从而达到恶意攻击用户的目的。带来的危害有盗取Cookie、投放蠕虫等
- ✓ 检测原理：识别被测试系统中的请求参数，将参数值替换成js脚本，若响应中对注入脚本未进行转码，则判断此处有XSS风险



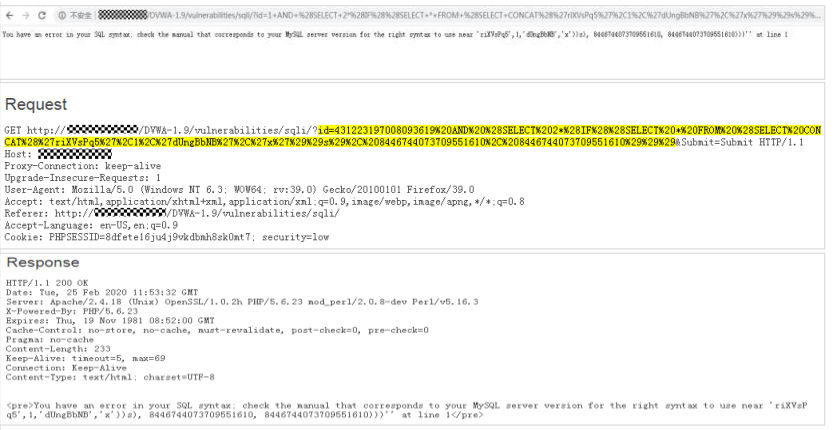
修订建议：XSS本质上是一种html注入，也就是将html代码注入到网页中。其防御的根本就是在将用户提交的代码显示到页面上时做好一系列的过滤与转义。

HUAWEI

- 案例四：对存在XSS攻击漏洞的靶场进行访问，在访问的URL后添加额外的内容，一般是一段JS脚本，当用户点击ClickMe按钮提交GET请求后，如果响应的页面出现的弹窗提示，这就表明添加的JS脚本已经执行。
- 基于这样的工作原理，Web漏洞工具扫描时，在获取URL后，会自动构建并添加playload后发起请求，然后通过检测响应是否包含特定内容的方式来检测xss漏洞。

Web漏洞扫描案例 - SQL注入

- ✓ 当应用程序将用户输入的内容，拼接到SQL语句中，一起提交给数据库执行时，就会产生SQL注入威胁。攻击者可以利用这部分可控制的内容，注入自己定义的语句，改变SQL语句执行逻辑，让数据库执行任意自己需要的指令。
- ✓ SQL注入的危害有：信息泄露、网站数据被篡改、网站被挂马等



修订建议：对SQL语句采用预编译是防御SQL注入的最佳方法 。

HUAWEI

- 案例五：SQL注入，同样我们对具备SQL注入漏洞的靶场进行访问，可以看到这里的请求是反馈id=1的账号，我们猜测这里的1会直接传递给数据库进行查询，因此我们再URL后构造额外的sql语句，比如select * from某个表，再请求发出后，我们查看响应，发现这里回复的是 SQL语句语法错误，说明，目标系统存在SQL注入，只要我们不停的尝试后面追加的内容，是可以实现数据库信息的直接读取的。
- 基于这样的工作原理，Web漏洞工具扫描时，在获取URL后，会自动构建并添加playload后发起请求，当然这里的playload一般是一段语法错误的SQL语句，然后通过检测响应是否包含特殊错误信息来检查是否存在SQL注入漏洞。

DevCloud Web漏洞扫描服务

- DevCloud Web 漏洞扫描服务结合华为内部优秀的安全理念和安全实践，将Web漏洞检测融入测试流程中。 Web 漏洞扫描服务采用爬虫自动收集URL，用户无需付出环境搭建成本，一键即可完成扫描。扫描完成后，为用户提供清晰简洁的扫描报告和修复建议，帮助用户从多角度分析资产风险。





简单
- 零环境搭建成本
- 爬虫自动收集URL
- 融入测试流程
- 一键完成扫描

精准
- 采用 Web2.0 智能爬虫技术，全面准确
- 扫描插件不断优化，提高检测准确率
- 自动发现资产指纹信息

专业
- 支持云内外网站扫描
- 关注业界紧急CVE爆发漏洞情况
- 清晰简洁的扫描报告，多角度分析资产风险

HUAWEI

- DevCloud Web漏洞扫描服务结合华为内部优秀的安全理念和安全实践，将Web漏洞检测融入测试流程中。 Web漏洞扫描服务采用爬虫自动收集URL，用户无需付出环境搭建成本，一键即可完成扫描。扫描完成后，为用户提供清晰简洁的扫描报告和修复建议，帮助用户从多角度分析资产风险。

DevCloud 主机漏洞扫描服务

- DevCloud 主机漏洞扫描服务结合华为内部优秀的安全理念和安全实践，将主机漏洞检测融入测试流程中。丰富的漏洞信息库，支持多种通用的Linux操作系统及华为自研EulerOS系统，扫描精度高。扫描完成后，为用户提供清晰简洁的扫描报告和修复建议，帮助用户从多角度分析资产风险。



简单
- 端口自动探测
- 系统自动识别
- 一键完成扫描
- 融入测试流程

精准
- 基于版本的精确匹配
- 明确漏洞修复建议
- 漏洞更新时效性高
- 支持系统类型广

专业
- 清晰简洁的扫描报告
- 丰富的漏洞库信息
- 支持多种通用Linux系统
- 支持华为自研EulerOS系统
- 支持高危组件漏洞检测

HUAWEI

- DevCloud 主机漏洞扫描服务结合华为内部优秀的安全理念和安全实践，将主机漏洞检测融入测试流程中。丰富的漏洞信息库，支持多种通用的Linux操作系统及华为自研EulerOS系统，扫描精度高。扫描完成后，为用户提供清晰简洁的扫描报告和修复建议，帮助用户从多角度分析资产风险。

DevCloud 安全配置扫描服务

- DevCloud 安全配置扫描服务结合华为内部优秀的安全配置实践，并融入测试流程中的主机漏洞扫描服务。对通用操作系统的配置项进行完善的检测，包含系统类、Web容器类、数据库类等组件。扫描完成后，为用户提供清晰简洁的扫描报告和修复建议，帮助用户快速规避资产风险。





简单
- 零环境搭建成本
- 检测对象自动探测
- 一键完成策略扫描
- 扫描参数自动识别

精准
- 检测规则全面准确
- 插件不断优化，提高检测准确率
- 修复建议明确

专业
- 支持云内外网站扫描
- 清晰简洁的扫描报告，多角度分析资产风险
- 检测对象多样：包含系统类、Web容器类、数据库类等

HUAWEI

- DevCloud 安全配置扫描服务结合华为内部优秀的安全配置实践，并融入测试流程中的主机漏洞扫描服务。对通用操作系统的配置项进行完善的检测，包含系统类、Web容器类、数据库类等组件。扫描完成后，为用户提供清晰简洁的扫描报告和修复建议，帮助用户快速规避资产风险。

目录

1. DevSecOps简介
2. **安全策略与工具**
   - 操作和资源可管控
   - 编码安全与工具
   - 漏洞扫描与工具
   - 安全运维与工具

HUAWEI

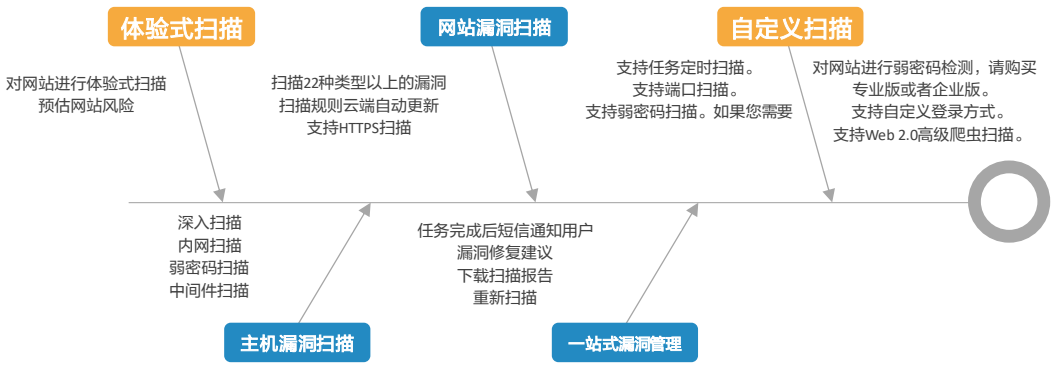
华为云租户业务防护方案



HUAWEI

- 漏洞扫描服务具有web网站扫描和主机扫描两种扫描能力。
  - web网站扫描采用网页爬虫的方式全面深入的爬取网站url，基于多种不同能力的漏洞扫描插件，模拟用户真实浏览场景，逐个深度的分析网站细节，帮助用户发现潜在的安全隐患。同时内置了丰富的无害化扫描规则，以及扫描速率动态调整能力，可有效避免用户网站业务受到影响。
  - 主机扫描需要经过用户授权（支持账密授权、脚本授权方式）访问用户主机，能够自动发现并检测主机操作系统、中间件等版本漏洞信息和基线配置，基于实时同步官网更新的漏洞库匹配漏洞特征，帮助用户及时发现主机安全隐患。

漏洞扫描服务 VSS



HUAWEI

- 漏洞扫描服务VSS是基于远程的扫描方式，用户无需额外部署agent。
- 漏洞扫描服务帮您快速检测出您的网站和主机存在的漏洞，提供详细的漏洞分析报告，并针对不同类型的漏洞提供专业可靠的修复建议。
- 体验式扫描
  - 第一次使用漏洞扫描服务的新用户，在未进行域名认证时，可以对网站进行体验式扫描，预估网站风险。
- 主机漏洞扫描
  - 支持深入扫描
    - 通过配置验证信息，可连接到服务器进行OS检测，进行多维度的漏洞、配置检测。
  - 支持内网扫描
    - 可以通过密钥的方式访问业务所在的服务器，适配不同企业网络管理场景。
  - 支持弱密码扫描
    - 多场景可用
      - 全方位的OS连接，涵盖90%的中间件，支持标准Web业务弱密码检测、操作系统、数据库等弱口令检测。
    - 丰富的弱密码库
      - 丰富的弱密码匹配库，模拟黑客对各场景进行弱口令探测，同时支持自定义字典进行密码检测。

企业主机安全 HSS

资产管理
资产管理功能可深度扫描出主机中的账号、端口、进程、Web目录和软件信息。

漏洞管理
漏洞管理功能将检测Linux软件漏洞、Windows系统漏洞和Web-CMS漏洞。

入侵检测
入侵检测功能可识别并阻止入侵主机的行为，实时检测主机内部的风险异变，检测并查杀主机中的恶意程序，识别主机中的网站后门。

基线检查
基线检查功能可扫描出主机系统和关键软件含有风险的配置信息。

网页防篡改
网页防篡改功能可发现并阻止篡改指定目录下文件的行为，快速恢复被篡改的文件。



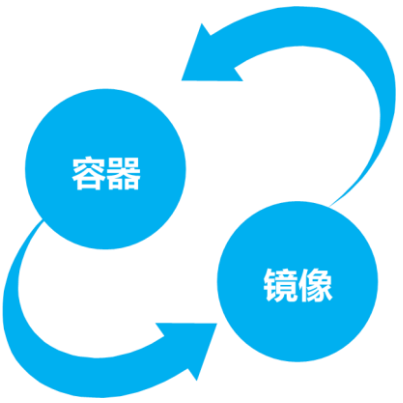
HUAWEI

- 企业主机安全服务（Host Security Service，HSS）是一个用于保障主机整体安全的安全服务。企业主机安全服务通过资产管理、漏洞管理、入侵检测及基线检查功能，可全面识别并管理主机中的信息资产，实时监测主机中的风险并阻止非法入侵行为，帮助您管理主机的安全状态，使您的主机顺利通过网络安全等级保护测评。
- 在主机中安装客户端后，您的主机将受到HSS云端防护中心全方位的安全保障，在安全控制台可视化界面上，您可以统一查看并管理同一区域内所有主机的安装状态和安全风险。

容器安全服务 CGS

镜像漏洞管理

容器安全服务可扫描云容器镜像服务中的私有镜像、官方镜像、以及节点中所有正在运行的镜像，发现镜像中的漏洞并给出修复建议，帮助用户得到一个安全的镜像。

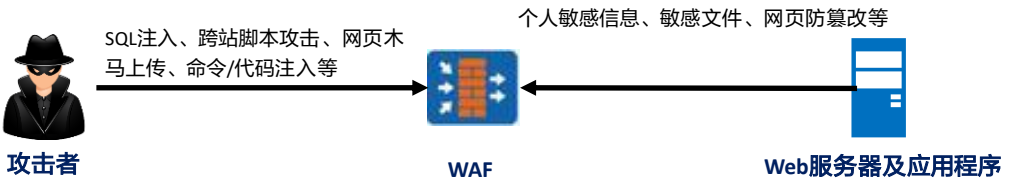


HUAWEI

- 容器安全服务通过配置安全策略，帮助企业制定容器进程白名单和文件保护列表，从而提高容器运行时系统和应用的安全性。
- 容器安全服务扫描所有正在运行的容器，发现容器中的异常（包括逃逸漏洞攻击、逃逸文件访问等）并给出解决方案。
- 漏洞管理
  - 扫描节点中所有正在运行的容器镜像，发现镜像中的漏洞并给出修复建议，帮助用户得到一个安全的镜像。
- 安全策略
  - 通过配置安全策略，帮助企业制定容器进程白名单和文件保护列表，确保容器以最小权限运行，从而提高系统和应用的安全性。
- 运行时监控
  - 监控节点中容器运行状态，发现违反容器安全策略的进程运行和文件修改，以及容器逃逸行为。

Web应用防火墙 WAF

- Web应用防火墙（Web Application Firewall，WAF），通过对HTTP(S)请求进行检测，识别并阻断SQL注入、跨站脚本攻击、网页木马上传、命令/代码注入、文件包含、敏感文件访问、第三方应用漏洞攻击、CC攻击、恶意爬虫扫描、跨站请求伪造等攻击，保护Web服务安全稳定。



思考题

（单选）下列关于字符串操作的说法，正确的是（ ）：

A. strncpy、strncat等带n版本的字符串操作函数使用可以有效的防止‘\0’结束符丢失

B. 来自fgets(), getenv()的字符串在复制之前，应该获取并检查其数据长度

C. 环境变量来自计算机内部，其安全性是可预测的，因此可以不加检查的作为字符串处理函数的输入

D. 当源内存和目标内存存在重叠时，strcpy会出现错误，而memcpy_s能正确地实施拷贝，但这也增加了一点点开销

HUAWEI

- 参考答案：B
- A：带n版本函数，容易产生结束符丢失问题。
- B：fgets()/getenv()获取的内容都是不可信的，因此需要添加校验。
- C：环境变量是不可预测的，所以应该添加校验。
- D：memcpy_s不能操作重叠内存，memmove_s可以。

思考题

（多选）SQL注入是Web应用中常见的安全问题，下列哪些操作可以有效的缓解该类问题？（）：

A. SQL语句使用参数化查询；

B. 利用存储过程动态拼接查询语句；

C. 利用正则表达式对传入参数做白名单校验；

D. 对SQL语句采用预编译；

HUAWEI

- 参考答案：ACD
- A：参数化查询是一种简单有效的防止SQL注入的查询方式。
- B：存储过程使用参数化查询时，也应该使用参数化查询，如果通过拼接的SQL也存在被注入的风险。
- C：对输入进行合法性校验，只接受在白名单中的字符，可有效降低风险。
- D：对SQL语句采用预编译是防御SQL注入的最佳方法。

本章总结

- DevSecOps概念及总结
- 讲解了DevSecOps的实践案例
- DevSecOps安全策略与工具介绍
- DevSecOps华为云实践
- 安全编码对软件开发的重要性、理念
- 常见的C/JAVA编码漏洞原理、案例和常见编程错误
- 华为安全编码检查体系及能力简介

HUAWEI

更多信息

- 华为云官网：
  - https://www.huaweicloud.com/
- 华为云软件开发平台DevCloud：
  - https://www.huaweicloud.com/devcloud/
- 华为云帮助中心：
  - https://support.huaweicloud.com/
- 华为云学院：
  - https://edu.huaweicloud.com/

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and organization for a fully connected, intelligent world.

Copyright©2020 Huawei Technologies Co., Ltd.
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.



持续部署与发布



HUAWEI

前言

在持续交付中， “部署流水线”确保代码和基础设施始终处于可部署状态，所有提交到主干的代码都可以安全地部署到生产环境。持续交付可实现整个软件发布流程的自动化。本章通过持续交付进行引入，最终DevOps工程师将在此阶段对已通过标准化测试流程的部署就绪构件进行最终的部署和发布。

本章将通过介绍DevOps中编译构建、部署、发布的流程和实践进行持续部署与发布的学习，同时对容器化应用的构建部署和自动化理念进行学习。

HUAWEI

目标

- 学完本课程后，您将能够：
  - 了解持续交付的理念和特点；
  - 了解容器技术与容器化应用在持续交付中的作用；
  - 了解自动化技术和理念；
  - 掌握编译构建的原理和方法；
  - 掌握部署发布的策略和方法。

HUAWEI

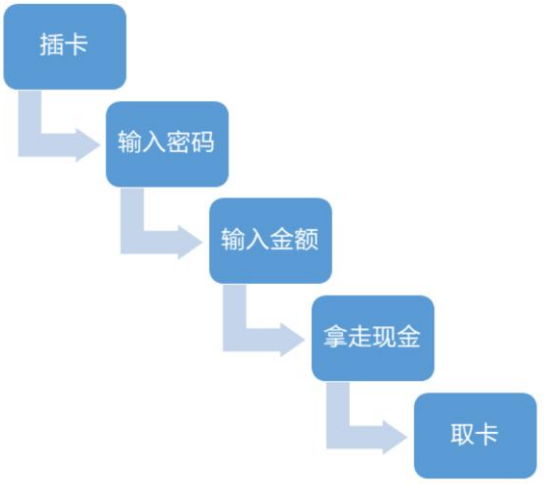
目录

1. **持续交付与持续部署**
2. 微服务架构与微服务化应用
3. 容器技术与容器化应用
4. 自动化的实现“一切即代码”
5. 自动化编译构建

HUAWEI

- 本章一共9个小节：
  - 持续交付与持续部署
  - 微服务架构与微服务化应用
  - 容器技术与容器化应用
  - 自动化的实现“一切即代码”
  - 自动化编译构建
  - 制品和包管理
  - 自动化部署
  - 发布管理
  - 自动化交付流水线

一个ATM取款系统





HUAWEI

一个ATM取款系统

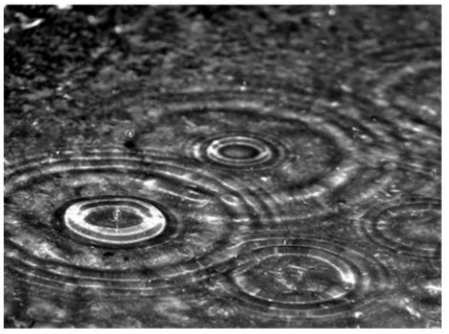
- 技术实现：
  - ATM机：机械系统控制，智能卡识别，接收用户输入，连接银行系统，监控等等
  - 传输层网络：数据加密，数据完整性，监控
  - 核心银行系统：账户认证，账务信息，审计，风控
- 挑战：
  - 紧耦合，系统复杂、错综交互，动一发而牵全身
  - 团队规模大，集团军作战



HUAWEI

软件交付的挑战

- 一个系统的修改带来涟漪效应，受影响系统随着系统复杂度增加而呈级数增长
- 被影响的系统可能还会造成二次涟漪效应
- 给需求规划，架构设计，开发过程，测试过程等整个软件开发过程造成极大的困扰



HUAWEI

- 软件交付的挑战：
- 第一协作成本高，业务响应慢。在传统架构中，一般公司内部都会按照功能模块划分工作，每一次新版本上线总会出现各种问题，例如分支、合并、冲突、代码不一致等带来很大的协作成本、沟通成本。
- 第二，系统复杂度增加，难以维护。随着业务量不断的增加和扩展，以及随着组织人员的变化，业务代码也会变得越来越难以维护，一次小小的改动可能会带来灾难性的风险。
- 第三、错误不能隔离。当所有的业务功能模块都聚集在一个程序集当中的时候，如果其中一个小的功能模块出现问题，那么都有可能会造成整个系统的崩溃，这就是软件开发中常说的涟漪效应。被影响的系统如果较为复杂还可能会造成二次涟漪效应。
- 第四，应用扩展能力差，传统架构不能够按需扩展。

系统耦合的三个级别





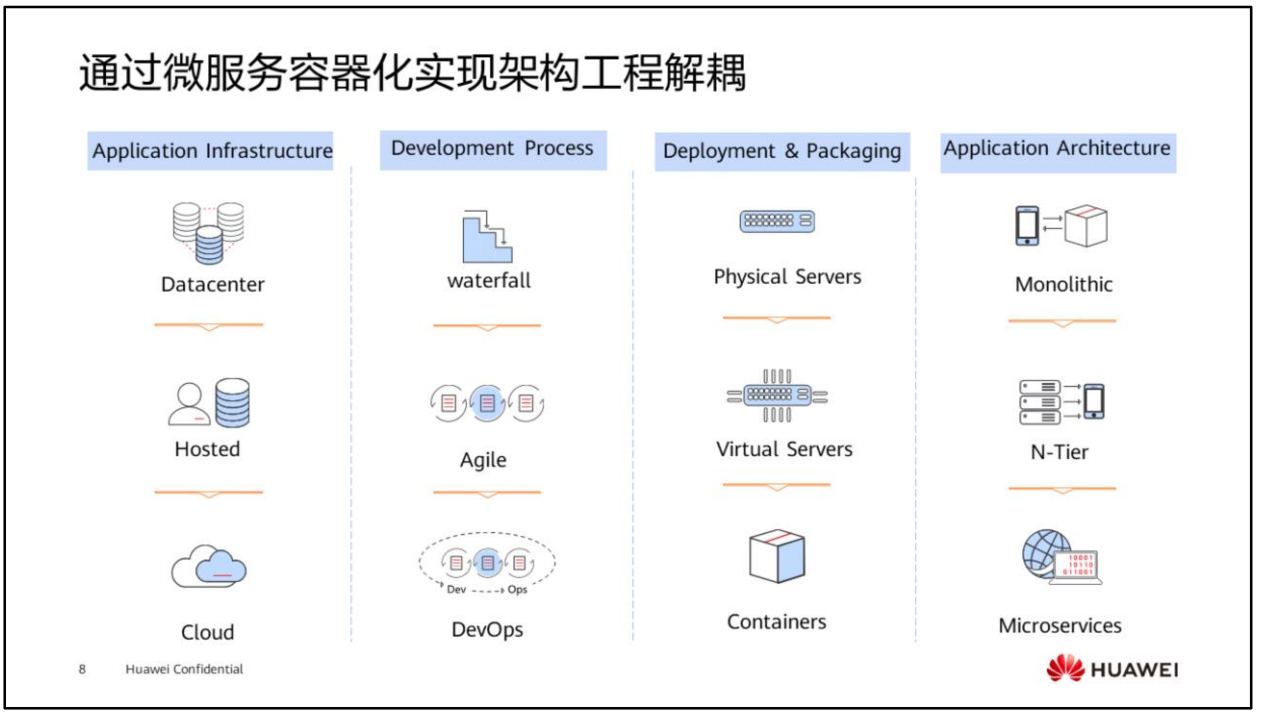


代码级耦合	组件级耦合	服务级耦合
一个开发人员的修改即可影响整个系统 项目团队规模 < 20	延迟影响至运行时 多团队协作成为可能 接口定义不通用 无法跨技术栈使用	延迟影响至生产环境 多团队，多技术栈协作成为可能 接口显视定义 服务自治

团队自由度，业务敏捷能力，交付速度，质量控制，系统复杂度，运维复杂度

HUAWEI

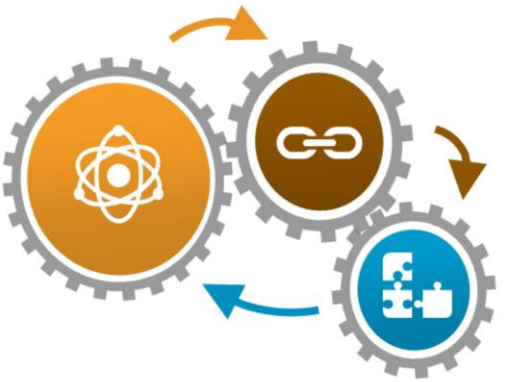
- 高内聚低耦合，它已经成为软件设计质量的标准之一。
- 耦合就是对某元素与其他元素之间的连接、感知和依赖的量度，这里所说的元素既可以是功能对象也可以指系统、子系统、模块。
  - 代码级耦合，常见于我们的单体架构项目中，这种项目往往模块非常多，且模块的边界模糊，依赖关系不清晰，代码质量参差不齐，整个项目非常复杂。
  - 组件级耦合，将一个大的软件系统按照分离关注点的形式，拆分成多个独立的组件。
  - 服务级耦合，也就是我们常说的微服务，它通过将功能分散到各个离散的服务中，以实现对解决方案的解耦，通过服务级耦合多团队多技术栈协作成为可能。
- 随着耦合级别的提高，团队自由度、业务敏捷能力、交付速度及质量都得到了很大的提高。



- 微服务的特点是小而独立，轻量级通讯正是因为小，所以每个微服务的逻辑相对简单，代码量也小很多，进而修复问题及修改变更会快很多，因为独立每个微服务都可以单独进行修改测试部署，使多个微服务可以并行演进，从而可以缩短测试部署和发布的时间，同样是因为独立每个微服务都可以根据实际情况单独的横向拓展，贯彻高内聚低耦合的架构思想。

通过自动化实现软件快速部署和发布

- 开源的自动化的工具
- “一切即代码”
- 统一的平台、流水线部署与发布
- 实现一定功能的子系统，解耦，拼接

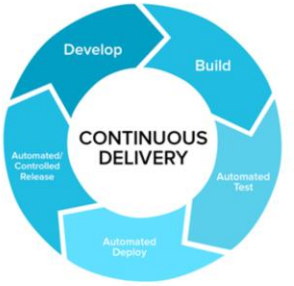


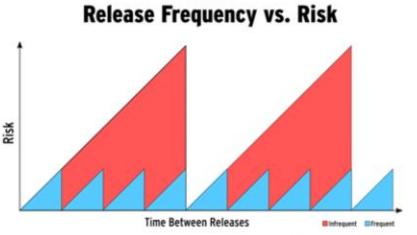
HUAWEI

- 尽可能地自动化也是DevOps的关键原则之一。
- 通过自动化的方式实现持续集成与持续交付。如果开发的功能服务相对独立，那么就能越快实现持续集成与交付，使业务得到尽快的部署与发布。那么微服务能够帮助我们更好的进行持续交付。

持续交付概念

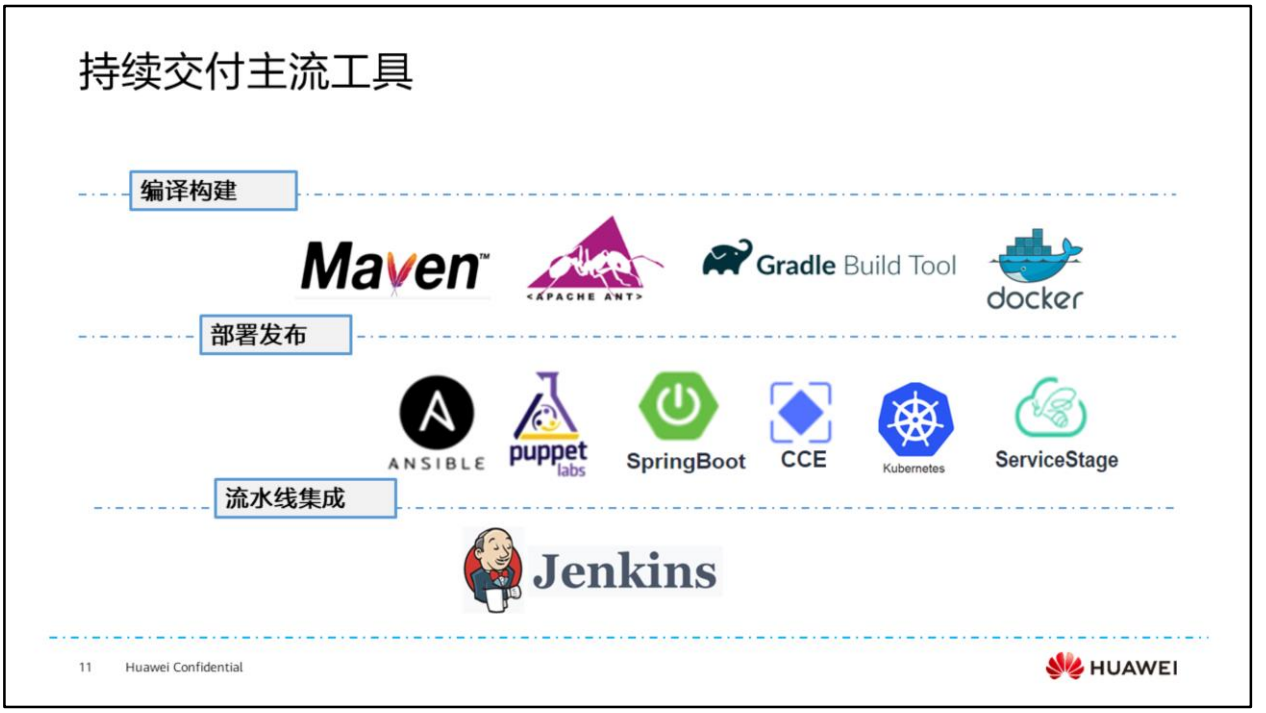
- 持续交付（英语：Continuous Delivery，缩写为CD），是一种软件工程手法，让软件产品的产出过程在一个短周期内完成，以保证软件可以稳定、持续的保持在随时可以发布的状况。
- 它的目标在于让软件的建置、测试与释出变得更快以及更频繁。这种方式可以减少软件开发的成本与时间，减少风险。
- 持续交付以持续集成为基础。





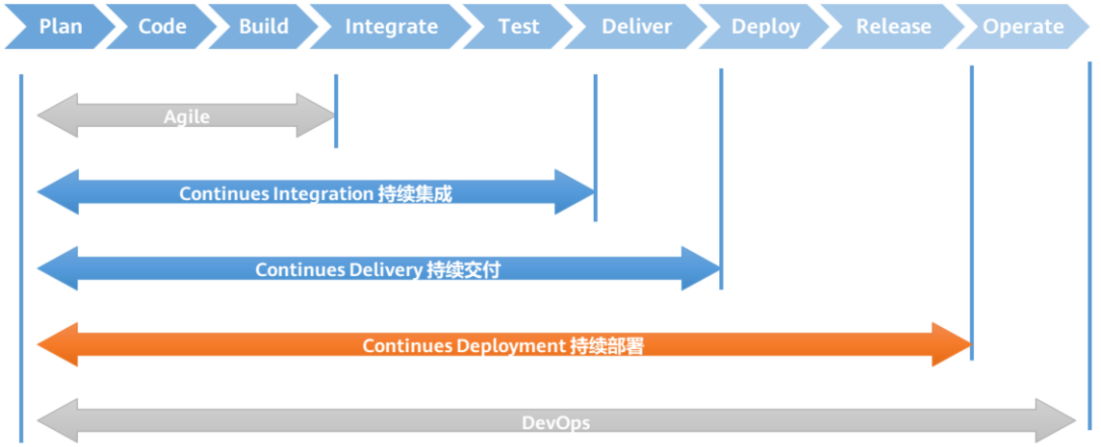
HUAWEI

- 持续交付以持续集成为基础。将集成后的代码部署到更贴近真实运行环境的类生产环境中。给质量团队或者用户，以供评审。如果评审通过，代码就进入生产阶段。持续交付优先于整个产品生命周期的软件部署，建立在高水平自动化持续集成之上。
- 试想下，如果说等到所有东西都完成了，才向下个环节交付，导致所有的问题只能在最后才爆发出来，这样会导致解决成本巨大，甚至无法解决。比如我们完成单元测试后，可以把代码部署到连接数据库的staging环境中，进行更多的自动化测试如果代码没有问题，可以继续手动部署到生产环境中，当然，持续交付不是指软件每一个改动都要尽快部署到生产环境中，它指的是任何的代码修改都可以在任何时候实施部署。



- 开源产品详情见下方链接;
  - Maven：http://maven.apache.org/
  - Ant：http://ant.apache.org/
  - Gradle：https://gradle.org/
  - Docker：https://www.docker.com/
  - Ansible：http://www.ansible.com.cn/
  - Puppet：https://puppet.com/
  - SpringBoot：https://spring.io/projects/spring-boot
  - Kubernetes：https://kubernetes.io/
  - Jenkins：https://jenkins.io/
- CCE与ServiceStage为华为云产品

持续部署的生命周期

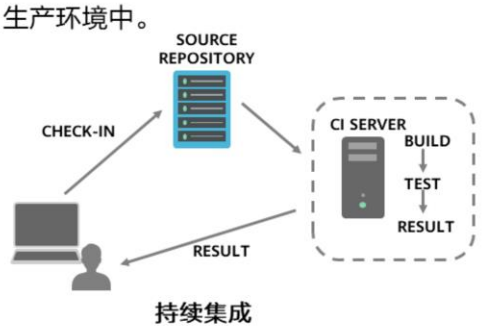


HUAWEI

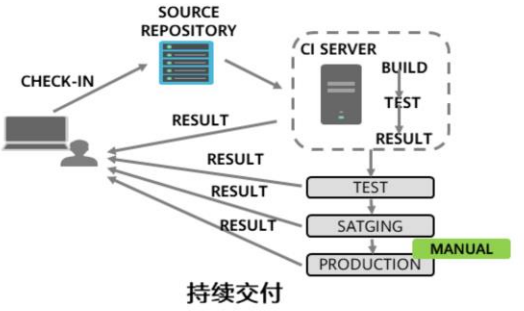
- 敏捷集中在开发阶段，强调需求的快速反馈。
- DevOps的落地需要CI、CD&CD的工具支撑。
- 持续交付、持续部署与 DevOps 的含义很相似，所以经常被混淆。但是它们是不同的两个概念。DevOps 的范围更广，它以文化变迁为中心，特别是软件交付过程所涉及的多个团队之间的合作（开发、运维、QA、管理部门等），并且将软件交付、部署、运维的过程自动化。另一方面，持续交付、持续部署是一种自动化的手段，关注点在于将不同的过程集中起来，并且更快、更频繁地执行这些过程。
- 因此，DevOps 可以说是三个持续的一个产物，持续交付和持续部署直接汇入 DevOps。

持续集成与持续交付

- **持续集成**(Continues Integration) 强调开发人员提交了新代码之后，立刻进行构建、单元测试。根据测试结果，我们可以确定新代码和原有代码能否正确地集成在一起。
- **持续交付**(Continuous delivery) 是持续集成的延伸，将集成后的代码部署到类生产环境，确保可以以可持续的方式快速向客户发布新的更改。如果代码没有问题，可以继续手动部署到生产环境中。



持续集成



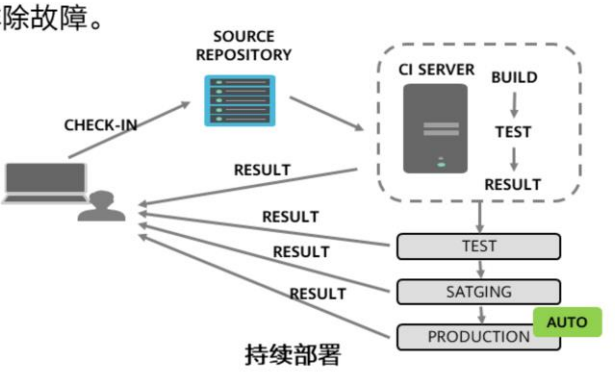
持续交付

HUAWEI

- 持续集成指的是，频繁地(一天多次)将代码集成到主干。持续集成的目的，就是让产品可以快速迭代，同时还能保持高质量。它的核心措施是，代码集成到主干之前，必须通过自动化测试。只要有一个测试用例失败，就不能集成。持续集成并不能消除 Bug，而是让它们非常容易发现和改正。
- 持续交付(Continuous delivery)指的是，频繁地将软件的新版本，交付给质量团队或者用户，以供评审。如果评审通过，代码就进入生产阶段。持续交付可以看作持续集成的下一步。它强调的是，不管怎么更新，软件是随时随地可以交付的。

持续部署是理想的工作流

- **持续部署**(continuous deployment) 则是在持续交付的基础上，把部署到生产环境的过程自动化。这意味着除了自动化测试之外，还可以自动完成发布过程，并且可以通过单击按钮随时部署应用程序。应该尽早部署到生产环境，以确保可以小批次发布，在发生问题时可以轻松排除故障。

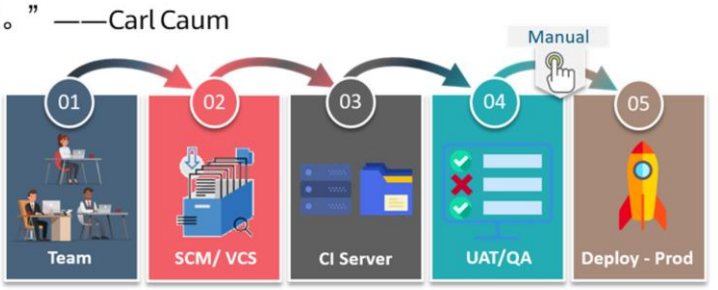


HUAWEI

- 持续部署(continuous deployment)是持续交付的下一步，指的是代码通过评审以后，自动部署到生产环境。持续部署的目标是：代码在任何时刻都是可部署的，可以进入生产阶段。

持续部署与持续交付的关系

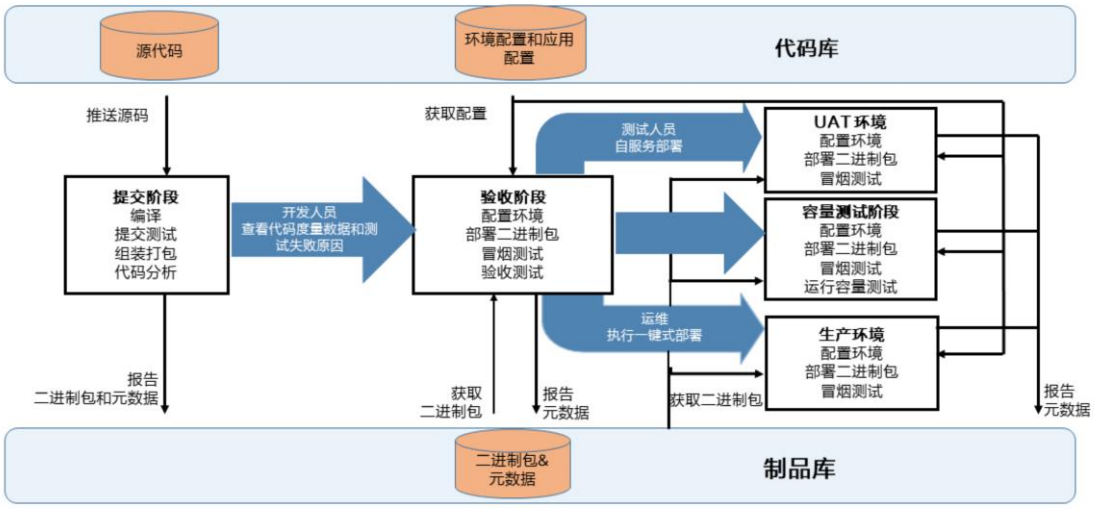
- 持续部署与持续交付
  - 持续部署意味着所有的变更都会被自动部署到生产环境中。持续交付意味着所有的变更都可以被部署到生产环境中，但是出于业务考虑，可以选择不部署。如果要实施持续部署，必须先实施持续交付。
- “持续交付并不意味着每一次变化都要尽快部署到生产环境中，而是意味着每一次变化都是随时可以部署的。”——Carl Caum



HUAWEI

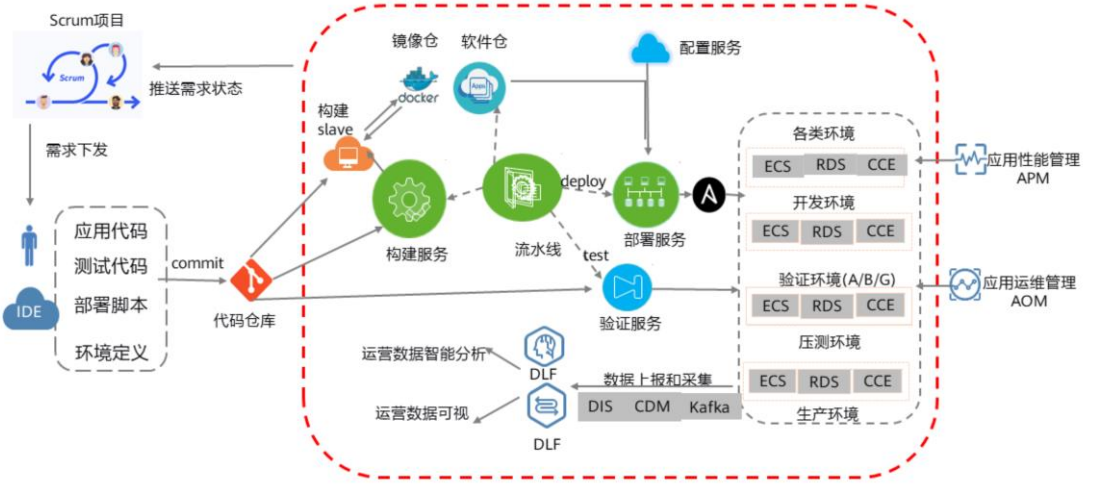
- 持续部署不适合交付软件的公司，适合交付服务的公司。

持续部署流水线衔接开发与生产



HUAWEI

持续部署工具链全景图



HUAWEI

思考题

1. 关于持续集成、持续部署和持续交付的说法，以下哪些是正确的（）？

   A. 持续交付以及持续部署等概念的核心：将技术行为与业务决策解耦。

   B. 持续交付是指，所有开发人员都在主干上进行开发工作，在需要发布时直接发布。

   C. 持续部署是指，在持续交付的基础上，由开发人员或运维人员自助式的定期向生产环境部署优质的构建版本。

   D. 持续集成，要求每当开发人员提交了新代码之后，就对整个应用进行构建，并对其执行全面的自动化测试集合。

HUAWEI

- 参考答案：ACD

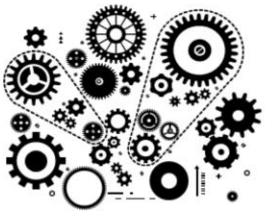
目录

1. 持续交付与持续部署
2. **微服务架构与微服务化应用**
3. 容器技术与容器化应用
4. 自动化的实现“一切即代码”
5. 自动化编译构建

HUAWEI

应用架构向微服务架构演进

第一代：单体架构



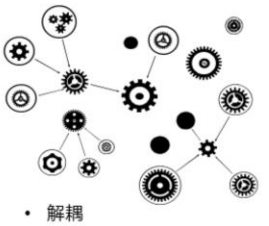
- 紧耦合
- 系统复杂、错综交互，动一发而牵全身
- 重复制造各种轮子：OS、DB、Middleware
- 完全封闭的架构

第二代：SOA架构



- 松耦合
- 在大型、超大型企业中仍然流行
- 通常通过ESB进行系统集成
- 大团队：100~200人
- TTM: 1年、半年、月
- 集中式、计划内停机扩容

第三代：微服务架构

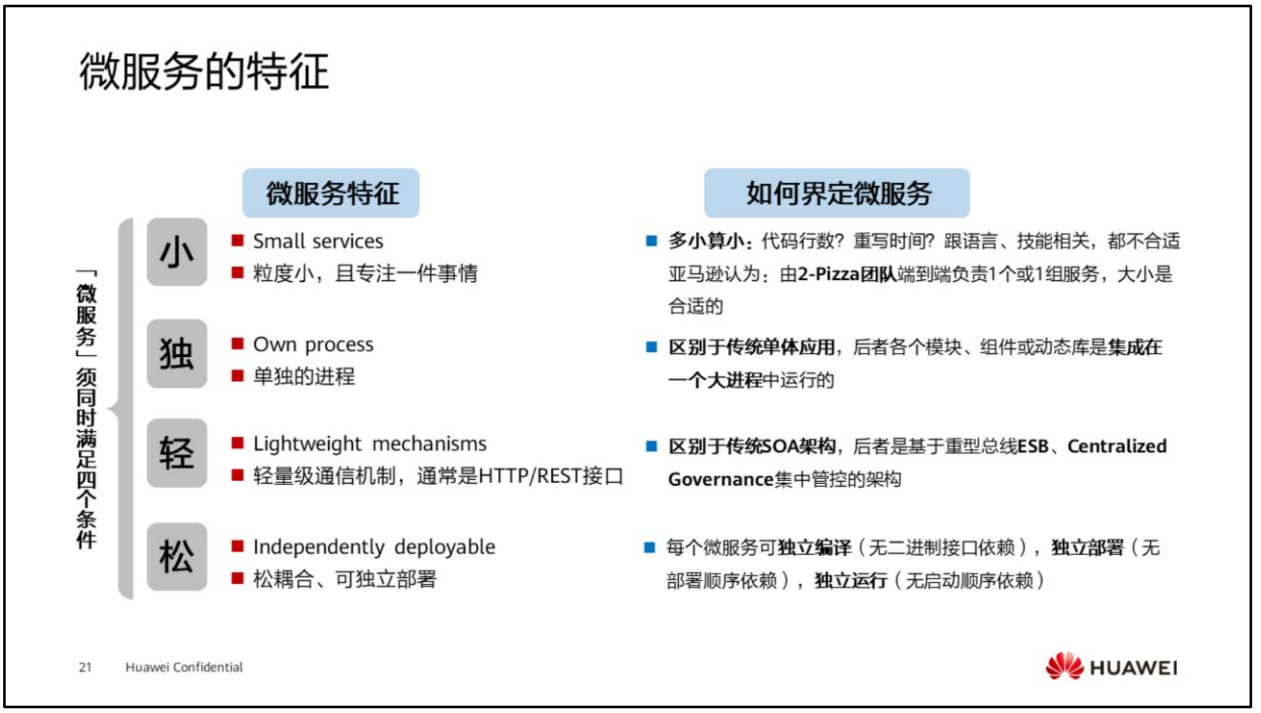


- 解耦
- 互联网公司、中小企业、初创公司
- 小团队：2 Pizza Team
- TTM: 按天、周进行升级发布
- DevOps: CI, CD, 全自动化
- 可扩展性：自动弹性伸缩
- 高可用：升级、扩容不中断业务

微服务是当前和未来的主流架构，带来的核心价值是能缩短业务上线周期和保障业务运行高可靠

HUAWEI

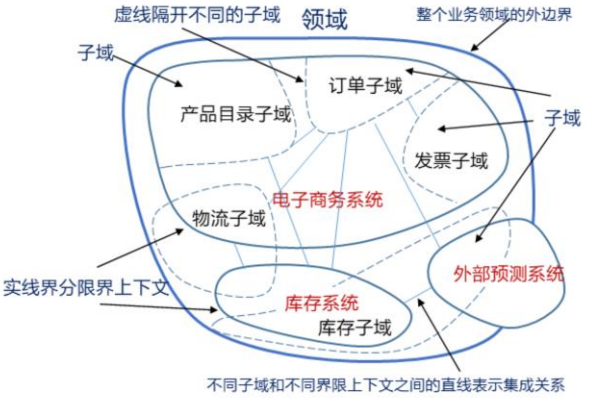
- 微服务架构有别于更为传统的单体式方案，可将应用拆分成多个核心功能。每个功能都被称为一项服务，可以单独构建和部署，这意味着各项服务在工作（和出现故障）时不会相互影响。
- 微服务带来了更高效和更经济的弹性能力。



- 微服务最主要的就是：小、独、轻、松。就是说微服务要小，模块边界要更清晰，支持独立部署独立演进，每个微服务都应该可以独立部署，独立演进，独立升级的。另外允许技术多样性，就是在微服务构成的一个整体的应用系统里面，每一块的业务用最适合的技术去实现，而不是都统一用一种语言去实现，这也是微服务非常重要的一个特点。

微服务设计 – DDD领域驱动模型

- 领域模型就是能够精确反映领域中某一知识元素的载体，这种知识的获取需要通过与领域专家(Domain Expert)进行频繁的沟通才能将专业知识转化为领域模型。领域模型无关技术，具有高度的业务抽象性，它能够精确的描述领域中的知识体系；同时它也是独立的，我们还需要学会如何让它具有表达性，让模型彼此之间建立关系，形成完整的领域架构。
  - 会员中心：负责用户账号登录、用户信息的管理
  - 商品中心：负责商品的展示、导航、维护
  - 订单中心：负责订单的生成和生命周期管理
    - 历史完成订单
    - 待支付订单
    - 取消订单
  - 交易中心：负责交易相关的业务
  - 库存中心：负责维护商品的库存
  - 促销中心：负责各种促销活动的支持



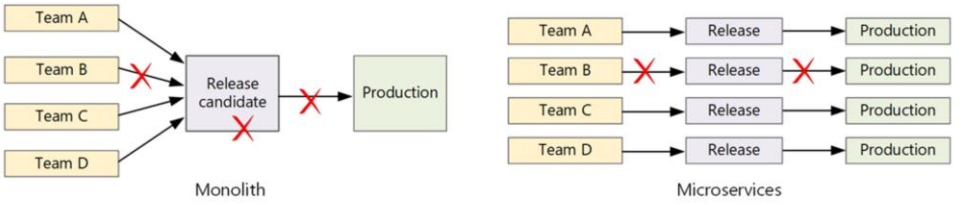
- **所以从本质上作为一种架构设计方法的DDD和作为一种架构风格的Microservices都是为着追求高响应力目标而从业务视角去分离复杂度的手段。**

HUAWEI

- 有时一个领域往往太复杂，涉及到的领域概念、业务规则、交付流程太多，导致我们没办法直接针对这个大的领域进行领域建模。所以，我们需要将领域进行拆分，本质上就是把大问题拆分为小问题：
  - 子域的边界；
  - 核心子域与非核心子域；
  - 公共支撑子域；
  - 子域之间的联系；

微服务架构支撑持续部署

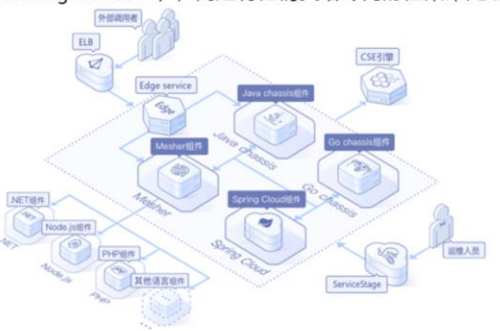
- 在传统的整体应用程序是一个生成管道的输出就是可执行应用程序。 所有开发工作都会馈送到此管道中。 如果发现了高优先级 bug，则必须集成、测试然后发布修复程序，这可能会延缓新功能的发布。通过拥有良好分解模块和使用功能分支代码更改的影响降至最低，可以缓解这些问题。 但是，随着应用程序变得越来越复杂和更多功能的添加，整体应用程序的发布过程往往会变得更加脆弱且可能中断。
- 根据微服务的理念，永远不应出现一个要让每个团队遵守的冗长发布周期。 服务“A”的团队随时可以发布更新，而无需等到已合并、测试并部署服务“B”中的更改。



HUAWEI

微服务底座(CHASSIS)

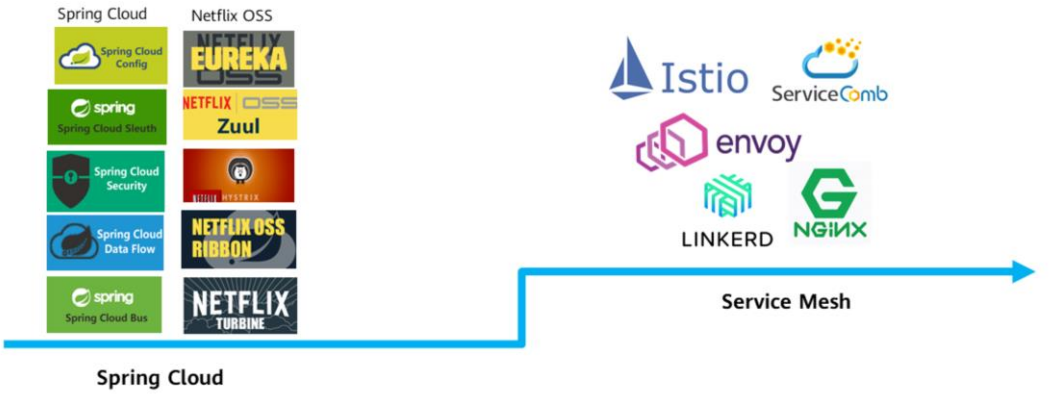
- Chassis，直译成中文是底盘的意思，它是一种微服务模式，由微服务大师 Chris Richardson提出。
- 当开始开发一个微服务应用时，经常会花大量的时间去关注一些横切面问题（cross-cutting concerns），比如：日志框架（log4j/logback），健康检查，metrics，分布式追踪；除了这些常见的问题，还有一些针对于特定技术的横切面问题。
- 在这种模式中，用户并不需要自己去处理横切关注点（crosscutting concern），而是将他们交给专门的框架来处理。用户可以更聚焦业务逻辑本身，简单、快速的开发微服务。
- SpringCloud、Dubbo、ServiceComb



HUAWEI

- 在单体应用里面都是进程内的调用，不用考虑什么服务发现、服务调用、服务负载均衡。当把一个大的业务拆分成了10个或者是更多的微服务之后，新增的成本和门槛也随之增加，故障点也会是增加，类似于日志框架、服务发现、健康检查这些在所有服务中都要解决的问题叫做横切面问题。

微服务框架和生态呈现多样化

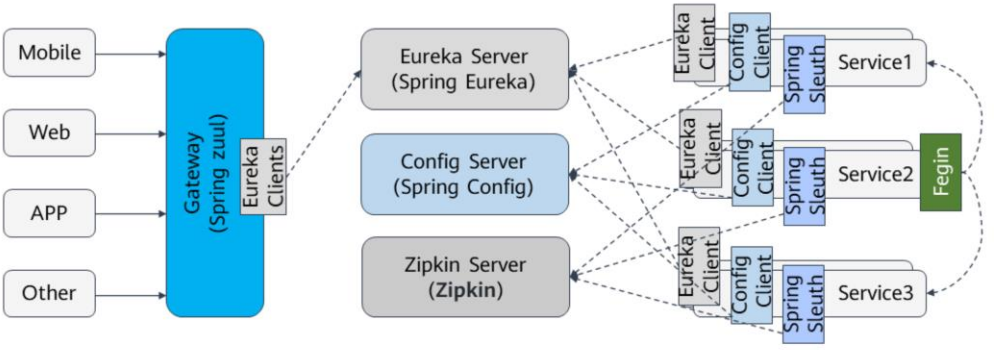


HUAWEI

- SpringCloud相关组件，详情参考SpringCloud中文网：https://www.springcloud.cc/
- Istio：https://istio.io/
- envoy：https://www.envoyproxy.io/
- ServiceComb：http://servicecomb.apache.org/
- Nginx：http://nginx.org/
- Linkerd：https://linkerd.io/

Spring Cloud微服务框架

- **Spring Cloud是一系列框架的有序集合。**它利用**Spring Boot**的开发便利性巧妙地简化了分布式系统基础设施的开发，如**服务发现注册、配置中心、消息总线、负载均衡、断路器、数据监控**等，都可以用Spring Boot的开发风格做到一键启动和部署。Spring Cloud通过Spring Boot风格进行再封装屏蔽掉了复杂的配置和实现原理，最终给开发者留出了一套简单易懂、易部署和易维护的分布式系统开发工具包。

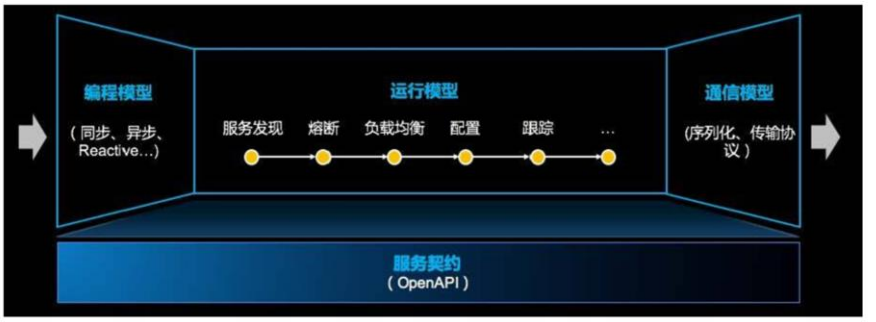


HUAWEI

- SpringCloud才是目前完整的微服务解决框架，功能非常强大：
  - Zuul是Netflix的一个基于JVM路由和服务端的负载均衡器。它的作用就是服务转发，接收并转发所有内外部的客户端调用。使用Zuul可以作为资源的统一访问入口，同时也可以在网关做一些权限校验等类似的功能。
  - Eureka是Netflix的服务注册和发现组件，将所有的可以提供的服务都注册到它这里来管理，其它各调用者需要的时候去注册中心获取，然后再进行调用。
  - Config为服务端和客户端提供了分布式系统的外部化配置支持。它包含了Client和Server两个部分，Server提供配置文件的存储、以接口的形式将配置文件的内容提供出去，Client通过接口获取数据、并依据此数据初始化自己的应用。
  - Zipkin是分布式实时数据追踪系统，其主要功能是聚集来自各个异构系统的实时监控数据，Sleuth为服务之间调用提供链路追踪。
  - Feign是一个声明式的Web Service客户端，它的目的就是让Web Service调用更加简单。它整合了Ribbon和Hystrix，从而让我们不再需要显式地使用这两个组件。

ServiceComb微服务框架

- ServiceComb 作为一个功能完善的微服务框架，包括应用框架代码生成，服务注册发现、服务配置管理、服务监控、服务调用追踪、多通信协议支持等功能，为开发者提供端到端的应用开发体验。
- ServiceComb 具备服务化契约增强、事件驱动等优势特性，并提供了分布式事务追踪能力，能够比较好的解决上述企业应用微服务开发所遇到的问题。ServiceComb 采用开放架构设计理念，兼容多种业界流行微服务框架，开发者亦可根据自身业务需求定制业务插件。

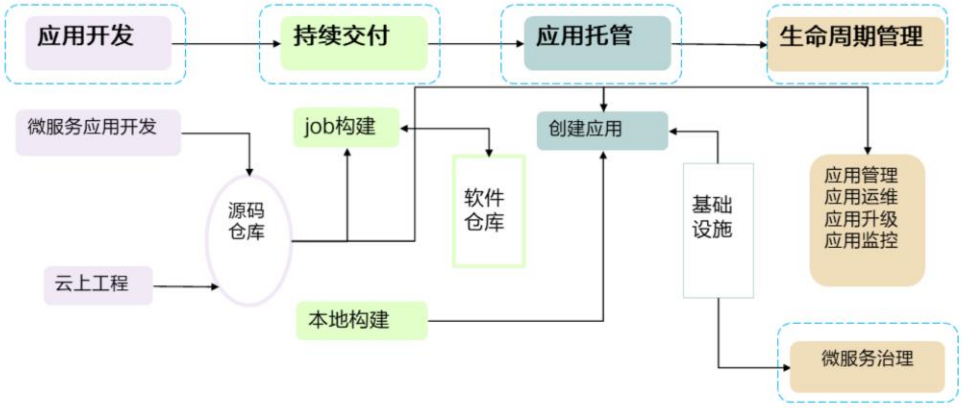


HUAWEI

- 服务契约的作用贯穿ServiceComb的三个模型，而不是简单地作为接口文档。
  服务契约将三个模型解耦，这使得运行模型中的同一套微服务治理逻辑既可以用于不同的开发风格代码，也可以用于不同的通信方式，让框架的功能扩展能力更好。同时接口契约规范了provider和consumer双方的交互行为，让开发与测试之间、不同微服务的开发之间的沟通协作效率更高。
- CSEJavaSDK作为一个带服务契约的REST开发框架，在使用上不会像传统的Servlet开发方式那么随心所欲，但随着系统规模扩大、复杂度提升，服务契约带来的好处将会明显大于其在开发方式上的限制。
- 图片来源：http://servicecomb.apache.org/cn/

华为微服务解决方案ServiceStage平台

- ServiceStage是一个应用托管和微服务管理平台，可以帮助企业简化部署、监控、运维和治理等应用生命周期管理工作。ServiceStage面向企业提供微服务、移动和Web类应用开发的全栈解决方案，帮助企业数字化快速转型。

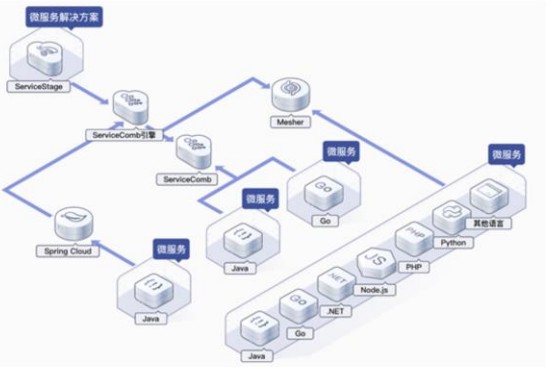


HUAWEI

- 包含应用开发、持续交付、应用托管、生命周期管理和微服务治理五大功能。

场景一: 服务应用解决方案

- 提供ServiceComb、Spring Cloud和Service Mesh多种解决方案，帮助用户建设企业级微服务平台。
- **优势**
  - API First：支持基于Swagger的API管理
  - 多语言微服务：支持JAVA、GO、Node.js、PHP、Python等等
  - 技术开放：支持ServiceComb、Spring Cloud、Service Mesh

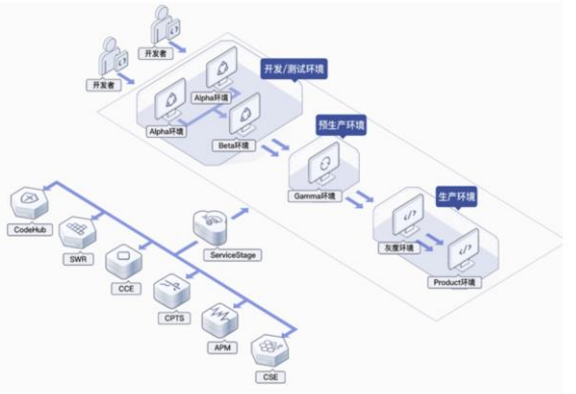


HUAWEI

- 图片来源：华为云

场景二: 微服务的持续交付解决方案

- 基于ServiceStage流水线，实现了集成环境统一、交付流程标准化，您可以实现全流程“自助式”开发、自验、集成验证与上线。
- **优势**
  - 一键生成持续交付环境：自动生成应用框架代码、构建、部署及测试环境
  - 多语言应用：Java、Go、Node.js、PHP、Python、Ruby
  - 多种源码仓库：CodeHub、GitHub、Gitee、GitLab、Bitbucke



HUAWEI

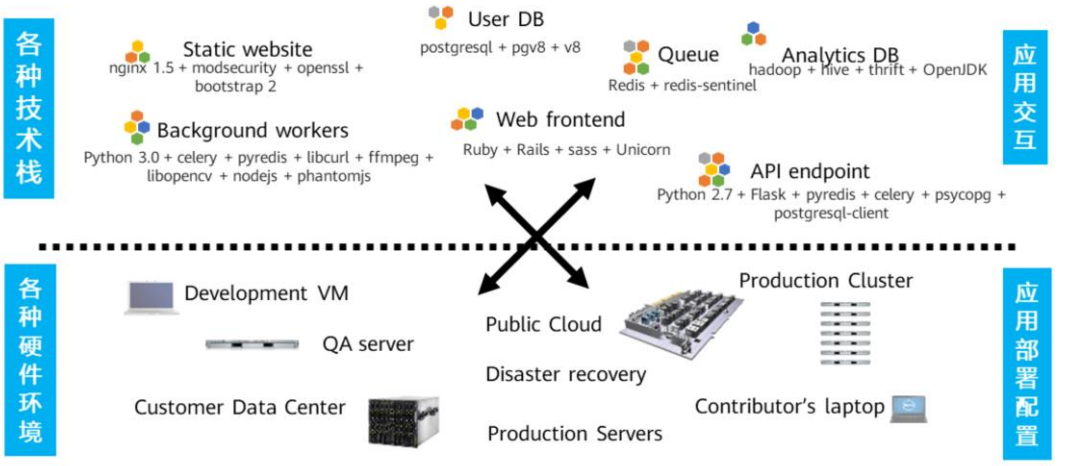
- 图片来源：华为云

目录

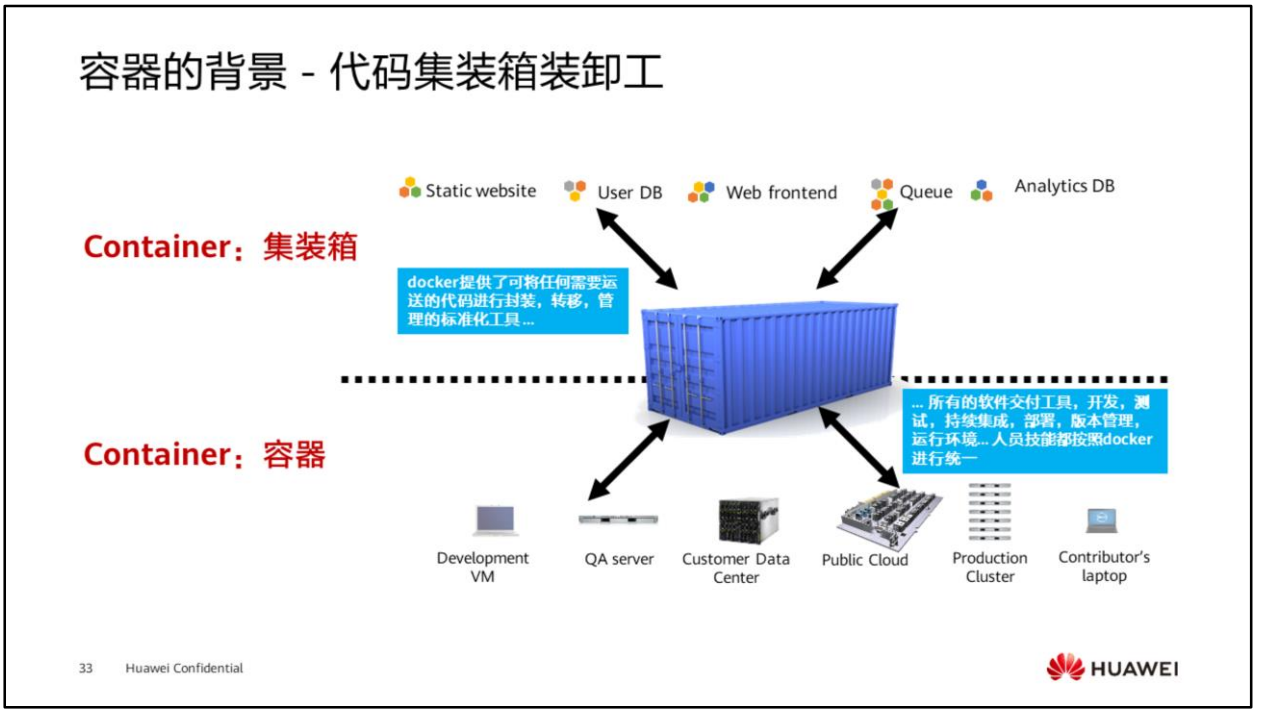
1. 持续交付与持续部署
2. 微服务架构与微服务化应用
3. **容器技术与容器化应用**
   - **容器技术概述**
   - 容器编排与容器化部署
4. 自动化的实现“一切即代码”
5. 自动化编译构建

HUAWEI

容器的背景 - 高度分散和异构化的部署环境



HUAWEI



- 容器，相当于软件行业的集装箱，也就是英文里的container，它就像我们运输业的集装箱一样，把货物都打包在了一起，那容器把软件应用进行了封装，把软件运行所依赖的环境都放在了一起，形成了软件界的统一运输规范。docker是容器技术的一种，它是将任何需要运送的代码进行封装，转移，管理的标准化工具，所有的软件交付工具、开发、测试、持续集成、部署、版本管理、运行环境都按照docker进行统一。

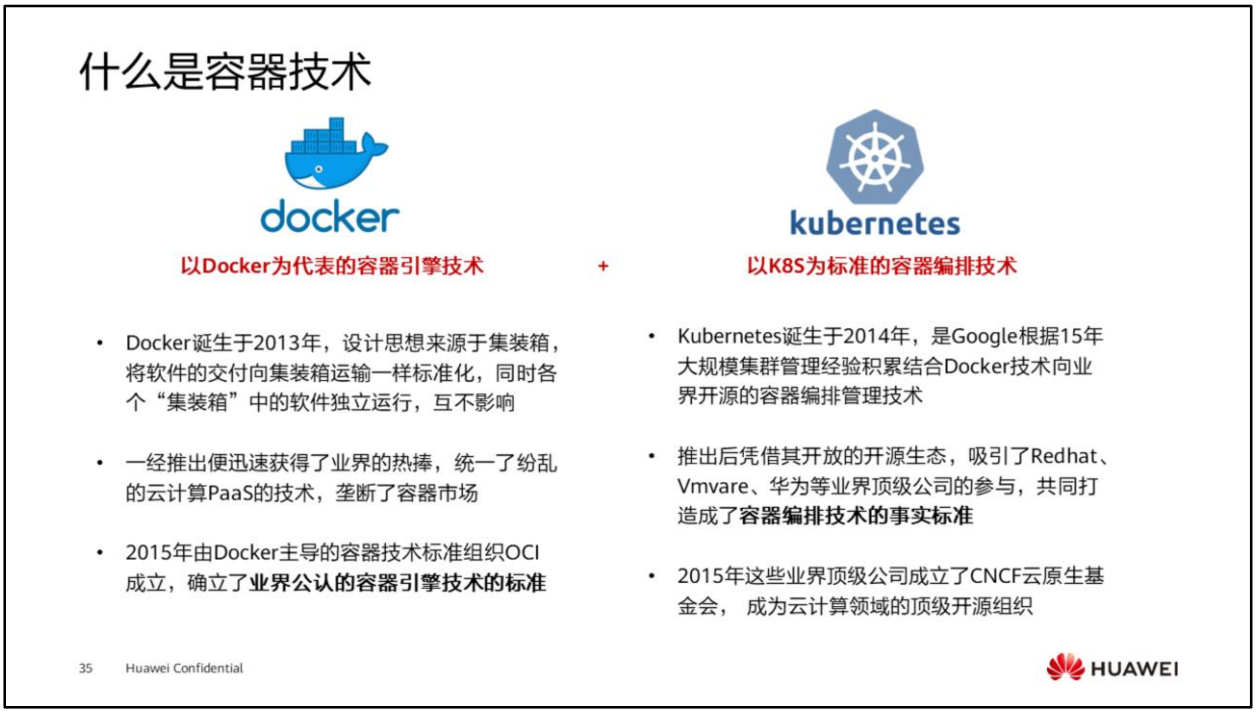
容器实现了快速交付和部署

- **容器**是操作系统内核自带能力，容器是在Linux内核实现的**轻量级高性能资源隔离机制**



HUAWEI

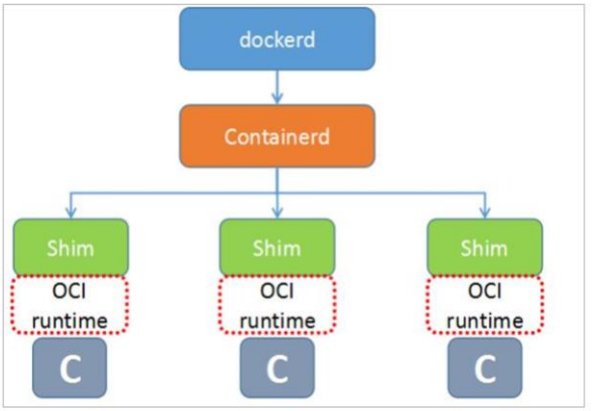
- 关键价值：
  - 快速交付和部署：一站式部署/运维容器应用，一键式滚动升级
  - 提升资源利用率：更细粒度地划分资源，提高资源利用率
  - 保障业务高可用：秒级弹性扩容，快速响应并发高峰
  - 复杂系统管理简单：单一重型应用解耦拆分为多个轻量模块，每个模块升级、伸缩更加灵活，轻松



- 容器技术在云计算领域可以分为两大类，一个是是以Docker为代表的容器引擎技术，另一个是以K8S为标准的容器编排技术。
- Docker代表容器引擎技术是因为docker主导了容器引擎的标准OCI，OCI实际上定义了容器的编译方式、技术上的约束，很多其他容器技术都遵循或者兼容OCI标准，如Kata安全容器。
- Docker的两个版本：
  - docker CE community edition是免费的开源社区版本。
  - docker EE enterprise edition是商用版，提供收费的技术支持。
- Kubernetes是业界公认的容器编排领域的事实标准，几乎所有的公有云厂商的容器技术都是基于kubernetes实现的。
- Kubernetes是完全的开源软件，社区开源版本直接运用于生产的风险比较大，所以各厂商呢都应用开源版本做了商业性增强、测试。

什么是Docker

- 最主流的容器运行时技术，提供了容器的运行时能力，经过多次容器技术标准的演进和OCI标准的确立，目前Docker的架构如下如所示：



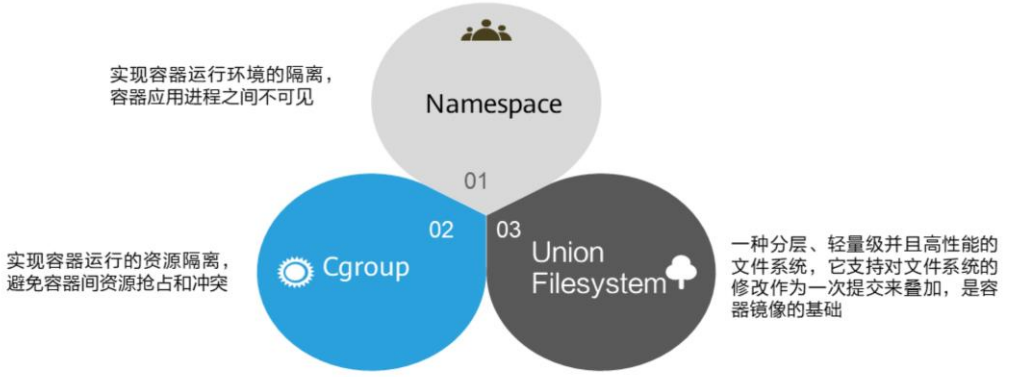
- Docker的技术特点：
  - 容器镜像定义了统一的软件交付标准
  - 文件挂载提供了更小资源占用的运行环境
  - 共享OS内核使容器本质上只是一个进程，秒级启停
- Docker的技术优势：
  - 统一的交付标准可以屏蔽环境差异，使能DevOps
  - 更小的资源消耗，提高资源利用率，匹配微服务架构
  - 极速的弹性伸缩、故障恢复，解放运维生产力

HUAWEI

- Docker这个英文本意是码头工人，docker容器像是软件的集装箱，提供了统一的软件打包能力，docker就是指搬运集装箱的码头工人。
- Docker是最主流的容器运行时，即容器引擎。containerd是容器技术标准化之后的产物，为了能够兼容OCI 标准，独立负责容器运行时和生命周期（如创建、启动、停止、中止、信号处理、删除等），从daemon中剥离，向docker engine提供运行容器的API，通过grpc通信；每创建一个容器，containerd都会创建一个shim进程，每个shim会调用runc（前身是libcontainer）进行实际的容器运行时管理，比如创建namespace的隔离。

Docker的关键技术

- Docker所应用三个关键技术，实现轻量化的隔离技术：

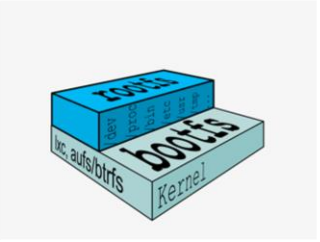


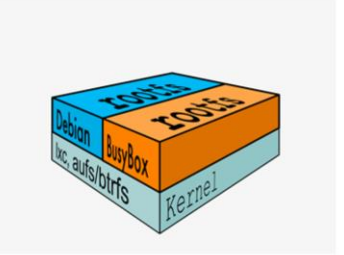
HUAWEI

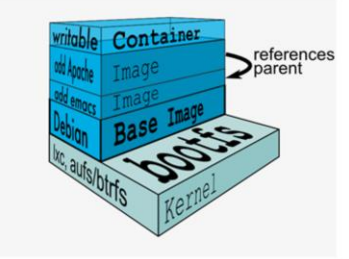
- Namespace负责运行环境的隔离，即每个容器都是一个独立进程，通过namespace技术进行隔离，每个容器互相不可见，包括进程隔离、网络隔离、文件隔离。
- Cgroup是负责运行资源的隔离或者说独占，可以为每个容器指定资源数量，互相不侵占。
- Union filesystem是解决应用运行的小型化统一标准，容器镜像提供了容器运行的基础，但容器镜像并不等于容器。

什么是Docker镜像

- Docker镜像文件image，提供了一个快速部署的模板，包含了已经打包好的应用程序和运行环境，基于image可以快速部署多个相同的容器。







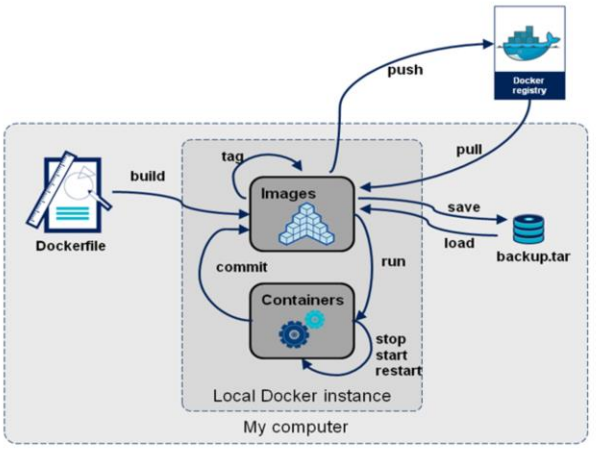
- 优点：
  - 多个容器可以共享image存储，节省存储空间
  - 部署多个容器时，base image可以避免多次拷贝，实现快速部署

HUAWEI

- Docker镜像是一个特殊的文件系统，除了提供容器运行时所需的程序、库、资源、配置文件外，还包含了一些为运行时准备的一些配置参数，镜像不包含任何动态数据，它的内容在构建之后也不会改变。类似于一个快速部署的模版，打包好了应用程序和运行环境。
- 镜像的特点是分层存储，由于镜像包含OS完整的Root文件系统，其体积往往是庞大的，因此在Docker设计时，充分利用了Union FS的技术，将其涉及为分层存储的架构，所以严格意义来说，镜像只是一个虚拟的概念，其由多层文件系统联合组成。
- 实际上容器镜像是一系列分层的只读文件，而当容器镜像运行为容器时，就会在镜像的最上层添加一个可写的层，也就是容器层，所有对于运行时容器的修改其实都是对这个容器读写层的修改，所有对容器的变化，比如写新的文件，修改已有文件和删除文件，都只会作用在这个容器层之中。

Dockerfile构建容器镜像

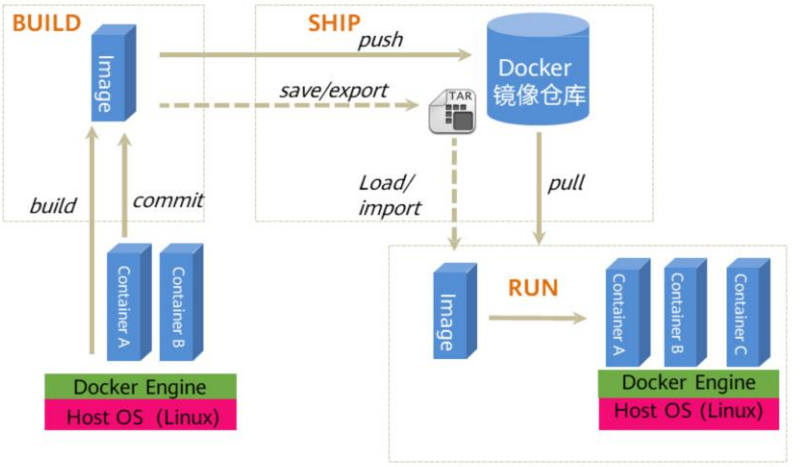
- 镜像是应用发布的标准格式，通过Dockerfile描述，可构建为一个tar包。



HUAWEI

- Docker镜像的生命周期管理：
  - docker pull 从镜像仓库拉取一个镜像
  - docker push 推送镜像到镜像仓库
  - docker build 将Dockerfile构建为镜像
  - docker tag 为镜像打上一个标签
- Docker容器的生命周期管理：
  - docker run 运行一个容器
  - docker stop 停止一个容器
  - docker start 启动一个容器
  - docker restart 重启一个容器
  - docker kill 杀掉一个运行的容器
  - docker pause 暂停一个容器

Docker发布系统工作机制

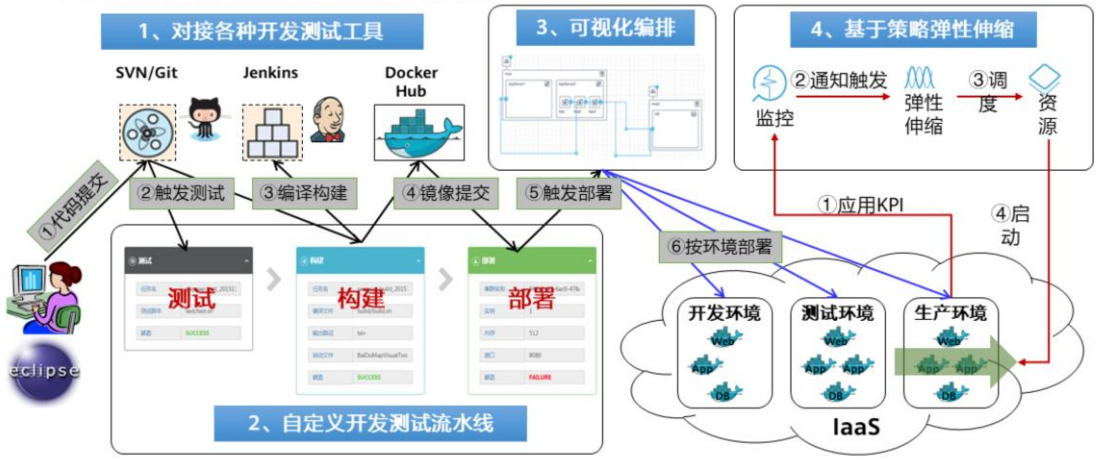


HUAWEI

- 镜像提供了一种标准化运输或者发布方式，发布系统工作机制就是build、ship、run。
- Docker的重要组件：
  - Docker引擎
  - Docker客户端
  - 镜像仓库

Docker容器的持续部署流水线

- 以容器镜像作为DevOps流程的标准交付件，将应用程序代码和运行环境（所依赖的库文件和系统服务、环境配置等）进行整体打包发布，统一打包格式，实现跨阶段的标准化发布流程。



HUAWEI

- 首先开发人员使用git工具提交代码，触发测试，这里的测试通常是对于代码的测试，白盒测试。Jenkins编译构建，Jenkins根据代码工程中的dockerfile即可构建为docker容器镜像，然后使用镜像，触发部署，通过k8s集群进行可视化编排与容器集群管理，如弹性伸缩，滚动升级等，统一格式打包的镜像可以一致性地部署到开发、测试、生产环境，容器镜像作为DevOps流程的标准交付件，实现了跨阶段的标准化发布流程。

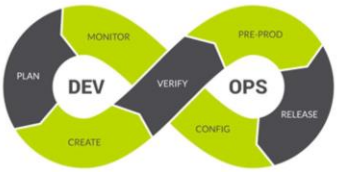
容器让开发和运维协同工作

开发人员

关心容器内：

- code
- Libraries
- Package Manager
- Apps
- Data

针对任何环境都采用同样方式打包
除了开发时，也要关心运行时



运维人员

只需要担心容器外的事情：

- 远程访问
- 监控
- 网络配置

所有的容器采用同样的方式启动，停止，复制，接入和升级

HUAWEI

- Docker的技术特点：
  - 容器镜像定义了统一的软件交付标准
  - 文件挂载提供了更小资源占用的运行环境
  - 共享OS内核使容器本质上只是一个进程，秒级启停
- Docker的技术优势：
  - 统一的交付标准可以屏蔽环境差异，使能DevOps
  - 更小的资源消耗，提高资源利用率，匹配微服务架构
  - 极速的弹性伸缩、故障恢复，解放运维生产力

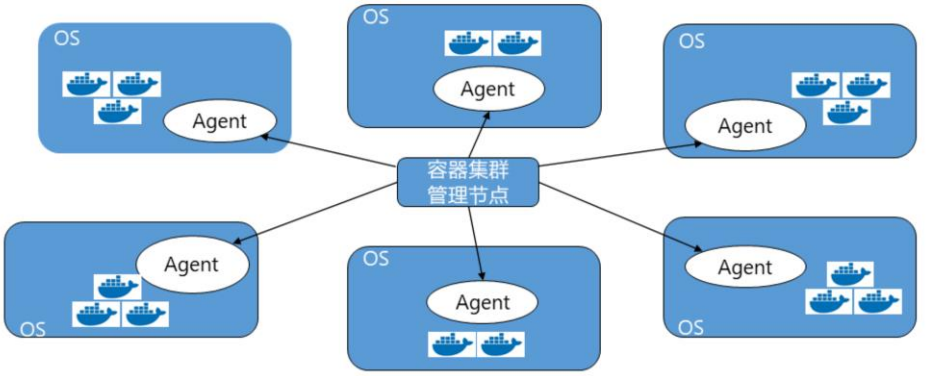
目录

1. 持续交付与持续部署
2. 微服务架构与微服务化应用
3. **容器技术与容器化应用**
   - 容器技术概述
   - **容器编排与容器化部署**
4. 自动化的实现“一切即代码”
5. 自动化编译构建

HUAWEI

什么是容器编排

- 一组互相连接的操作系统（实体机/虚拟机/容器）形成一个整体的资源池
- 可以扩展到上千个节点，具备自我修复，扩展和收缩能力
- 作为环境抽象层存在，使得应用不必关心所运行的节点

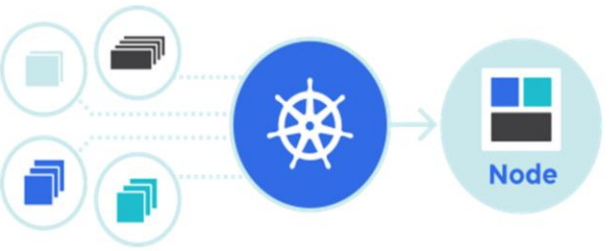


HUAWEI

- 为了管理分布在不同节点上成百上千个容器，这些节点会形成一个容器集群，容器集群实际上是一组互相连接的操作系统，可能是实体机、虚拟机或者容器，容器集群的管理节点会在集群的每个业务节点中部署一个agent，形成一个整体的资源池，并且作为环境抽象层存在。
- 容器应用可以运行在这个统一的资源池中，并不需要关心具体运行在哪个节点上，会有管理节点进行统一协调与分配，轻松实现容器应用的快速部署，并且可以扩展到上前个节点上，具备自我修复与弹性伸缩能力。

什么是Kubernetes

- Kubernetes 是用于自动部署，扩展和管理容器化应用程序的开源系统
  - 生产级别的容器编排系统
  - 数据中心OS
  - 新生代的云平台

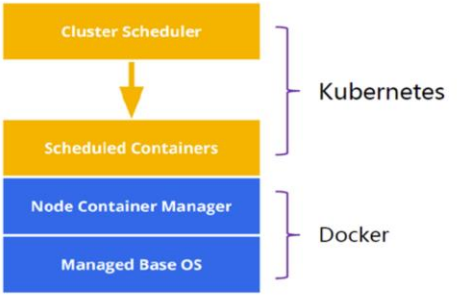


HUAWEI

- kubernetes是容器编排的标准，是谷歌根据自己内部的容器系统Borg，去除掉了业务属性，贡献给开源社区的产品，它是完全的开源软件。
- 容器编排系统以集群作为一个整体，一个集群看做一个完整的kubernetes的产品，大多数云厂商会在上面封装一个管理层，提供集群层面的管理。

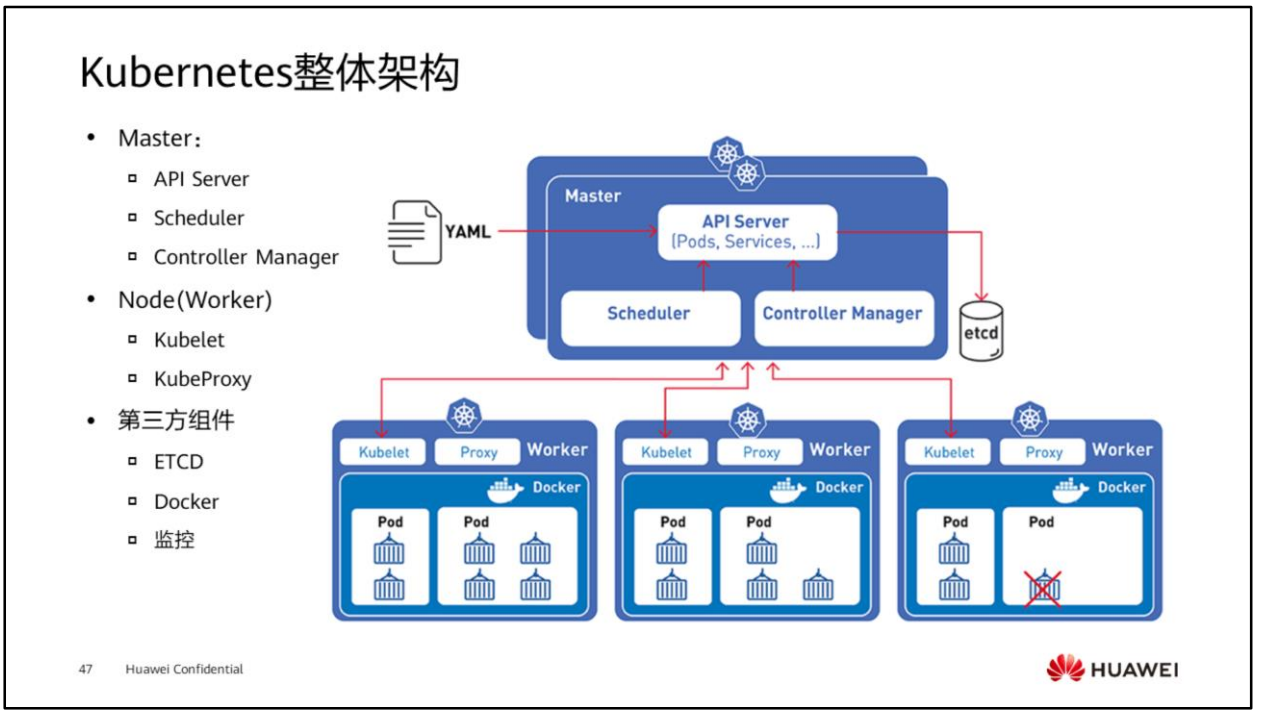
Kubernetes与Docker的关系

- 容器打包了软件对环境的依赖，保证同一软件交付件在开发、测试、生产不同环境的一致性。
- 容器高效的存取、分发、部署机制，提升了软件生产与交付效率。
- Kubernetes提供资源的管理和容器的调度技术，提供容器应用生命周期管理、弹性伸缩、监控运维的基本机制，决定容器之间如何进行交互。
- 由Docker、Kubernetes所带动的容器标准化大幅简化了各种环境管理工作，提升了部署流程自动化程度，因此大大加速了DevOps理念落地进程。



HUAWEI

- Docker容器是基于操作系统层，打包了软件对环境的依赖，保证同一软件交付件在开发、测试、生产不同环境的一致性，对节点上的容器资源实现一定的管理能力。
- k8s针对的是容器集群的管理，可能对于几个容器应用来说，你可以在docker的容器节点上手动地进行操作与管理，但是针对很多的容器应用的时候，比如成百上千个容器，Kubernetes提供资源的管理和容器的调度技术，提供容器应用生命周期管理、弹性伸缩、监控运维的基本机制，决定容器之间如何进行交互。
- 由Docker、Kubernetes所带动的容器标准化大幅简化了各种环境管理工作，提升了部署流程自动化程度，因此大大加速了DevOps理念落地进程。

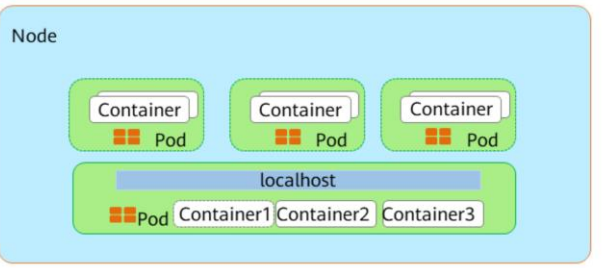


- Master管理节点：
  - API Server：提供了接口供客户端（kubectl）或者K8s其他组件，管理cluster的各种资源，主要提供Kubernetes API，提供对Pods，Services，ReplicationController等对象的CRUD，处理REST操作，并验证它们。
  - etcd：更新相应的对象API，负责保存cluster的配置信息和各类资源的状态信息。
  - Scheduler：负责决定将Pod放在哪个Node上运行，在调度时会充分考虑高可用、性能、数据亲和性的需求。
  - Controller Manager：负责管理Cluster各种资源，由多种Controller组成，包括replication controller、namespace controller等。
  - Replication controller：管理deployment\replicaset\daemonset等的生命周期。
- Node运行工作负载的节点，也叫作worker，它运行容器应用，Node由master统一管理，Node负责监控并汇报容器的状态，同时根据master的要求管理容器的生命周期：
  - Kube-proxy：提供统一的访问能力。转发Service请求到Pod，如果有多个副本，kube-proxy会实现负载均衡。
  - Kubelet：Node上的Agent，当Scheduler确定在某个Node上运行pod后，会将Pod的具体配置信息（image、volume等）发送给该节点的kubelet，kubelet会根据这些创建和运行容器，并向master报告运行状态。Kubelet管理pods和它们的容器、镜像、卷等。
  - Kubectl：k8s的命令行工具，kubectl连接APIServer，通过kubectl可以部署和管理应用，查看各种资源、创建、删除更新各种组件。

Kubernetes关键概念 - POD

- 在Kubernetes中，Pod是能够创建、调度、和管理的最小部署单元，是一组容器实例的集合，而不是单独的应用容器。
- 同一个Pod里的容器共享同一个网络命名空间，IP地址及端口空间和卷。
- 从生命周期来说，Pod是短暂的而不是长久的应用。Pod被调度到节点，保持在这个节点上直到被销毁。



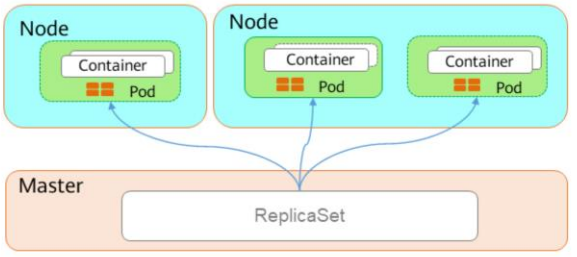


HUAWEI

- Pod是k8s最小工作单元，包含一个或多个容器，pod中的容器会作为一个整体被master调度到一个node中运行。Pod中的所有容器使用同一个网络namespace，即相同的IP地址和Port空间，这些容器可以共享存储。

Kubernetes关键概念 - ReplicaSet

- ReplicaSet，副本控制器
- 确保Pod的一定数量的份数(replica)在运行。如果超过这个数量，控制器会杀死一些，如果少了，控制器会启动一些
- ReplicaSet 用于解决 pod 的扩容和缩容问题
- 通常用于无状态应用



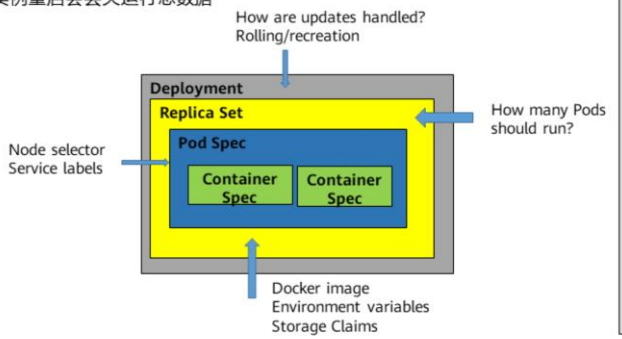

```
apiVersion: extensions/v1beta1
kind: ReplicaSet
metadata:
  name: frontend
spec:
replicas: 3
selector:
    matchLabels:
      tier: frontend
    matchExpressions:
      - {key: tier, operator: In, values: [frontend]}
  template:
    metadata:
      labels:
        app: guestbook
        tier: frontend
    spec:
      containers:
      - name: php-redis
        image: gcr.io/google_samples/gb-
frontend:v3
        resources:
          requests:
            cpu: 100m
            memory: 100Mi
        env:
        - name: GET_HOSTS_FROM
          value: dns
        ports:
        - containerPort: 80
```


HUAWEI

- Replicaset是副本控制器，主要作用是控制deployment应用副本的数量。
- ReplicaSet 解决了 pod 的扩容和缩容问题，通常用于无状态应用，因为无状态应用经常会涉及到扩缩容。

Kubernetes关键概念 - Deployment

- Deployment即在K8S中部署的无状态的应用：
  - 典型应用代表：Nginx，Tomcat
  - 不必为应用保持状态/持久化数据
  - 每个实例对于同一个请求响应的结果完全一致
  - 实例重启会丢失运行态数据



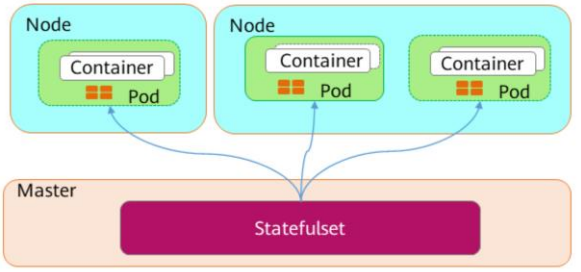

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```


HUAWEI

- 无状态应用的本质就是一个应用的多个实例之间完全没有区别，每个请求在不同的实例返回的结果都是一样的，k8s对他们的处理也是随机的，比如缩容，会随机杀掉几个容器。
- 对于用户来说，用户会建一个deployment，deployment自己会建replicset，replicset会建POD。

Kubernetes关键概念 - Statefulset

- Statefulset即在K8S中部署的有状态的应用：
  - 典型应用代表：mysql，redis
  - 每个实例具有唯一的网络标识符
  - 生命周期的操作将有序的在不同实例进行
  - 一般会挂载持久化存储保证数据持久性



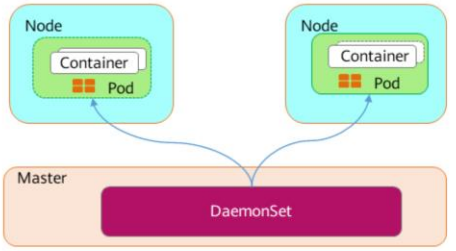

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
spec:
    serviceName: "nginx"
    replicas: 2
    selector:
      matchLabels:
      app: nginx
    template:
      metadata:
        labels:
          app: nginx
      spec:
        containers:
        - name: nginx
          image: k8s.gcr.io/nginx-slim:0.8
          ports:
          - containerPort: 80
            name: web
          volumeMounts:
          - name: www
            mountPath: /usr/share/nginx/html
```


HUAWEI

- 有状态应用本质是在运行过程中会保存数据或状态的工作负载称为“有状态工作负载“，每个实例都是唯一的。
- 有状态应用一般都会需要挂载持久化存储来保证数据的持久化。
- 有状态应用除了数据之外，每个实例都是一个独立的，比如mysql，会区分主从实例，那么在重启之类的操作的时候，每个实例的重启是有顺序的。

Kubernetes关键概念 - DaemonSet

- DaemonSet能够让所有（或者一些特定）的Node节点运行同一个pod。当节点加入到kubernetes集群中，pod会被（DaemonSet）调度到该节点上运行，当节点从kubernetes集群中被移除，被（DaemonSet）调度的pod会被移除，如果删除DaemonSet，所有跟这个DaemonSet相关的pods都会被删除。
- 在使用kubernetes来运行应用时，很多时候我们需要在一个区域（zone）或者所有Node上运行同一个守护进程（pod），例如如下场景：
  - 每个Node上运行一个分布式存储的守护进程，例如glusterd，ceph
  - 运行日志采集器在每个Node上，例如fluentd，logstash
  - 运行监控的采集端在每个Node，例如prometheus node exporter，collectd等



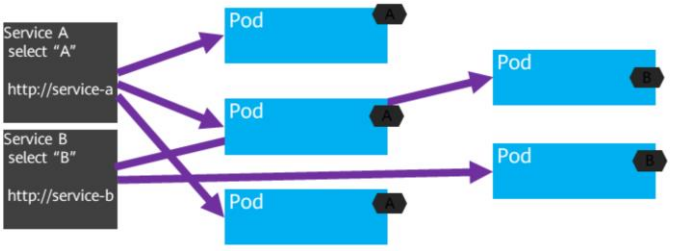

```
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: tomcat-ds
spec:
  template:
    metadata:
      labels:
        app: tomcat
    spec:
      containers:
      - name: tomcat
        image: tomcat:8.0.30-jre8
        ports:
        - containerPort: 8080
```


HUAWEI

- Daemonset，是一个特殊的应用对象，可以让所有的node节点上运行某个程序。

Kubernetes关键概念 - Service

- Service定义了pods的逻辑集合和访问这个集合的策略。Pods集合是通过定义Service时提供的Label选择器完成的。
- Service的引入旨在保证pod的动态变化对访问端透明，访问端只需要知道service的地址，由service来提供代理。
- Service的抽象使得前端客户和后端Pods进行了解耦。
- Kubernetes Services支持TCP和UDP协议，默认是TCP。
- Kubernetes支持两种基本的服务发现方法：环境变量和DNS。

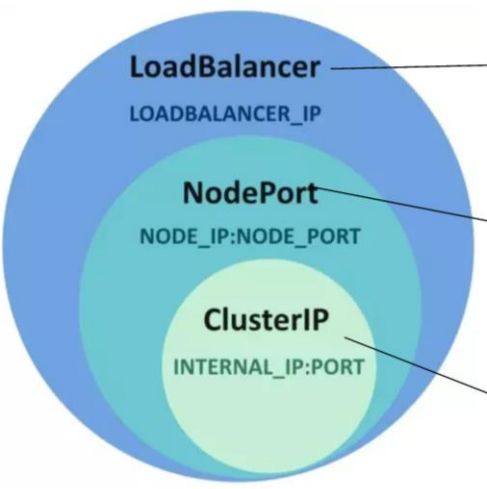



```
kind: Service
apiVersion: v1
metadata:
  name: my-service
spec:
  selector:
    app: A
  ports:
  - protocol: TCP
    port: 80
    targetPort: 9376
    nodePort: 30061
  type: ClusterIP
```


HUAWEI

- Service负责提供容器应用的访问，是通过4层协议访问，通过端口进行区分。
- Service定义了pods的逻辑集合和访问这个集合的策略。Pods集合是通过定义Service时提供的Label选择器完成，Kubernetes Services支持TCP和UDP协议，默认是TCP。

Service的类型和解释



使用一个ClusterIP & NodePort，但是会向cloud provider申请映射到service本身的负载均衡
LoadBalancer跟nodePort其实是同一种方式。区别在于LoadBalancer比nodePort多了一步，就是可以调用cloud provider去创建LB来向节点导流

除了使用cluster ip外，也将service的port映射到每个node的一个指定内部port上，映射的每个node的内部port都一样
nodePort的原理在于在node上开了一个端口，将向该端口的流量导入到kube-proxy，然后由kube-proxy进一步导给对应的pod

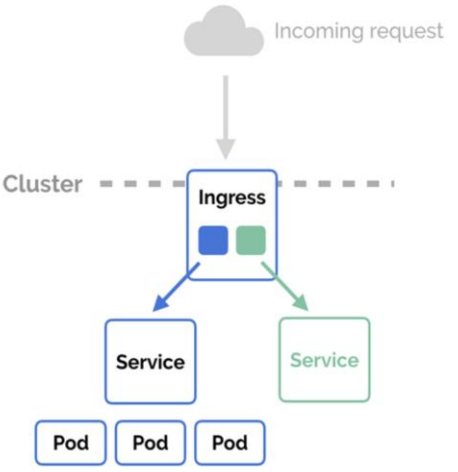
使用集群内的私有ip
clusterIP主要在每个node节点使用iptables，将发向clusterIP对应端口的数据，转发到kube-proxy中。然后kube-proxy自己内部实现有负载均衡的方法，并可以查询到这个service下对应pod的地址和端口，进而把数据转发给对应的pod的地址和端口

HUAWEI

- Service有3种类型：ClusterIP，NodePort，LoadBalancer，Service 定义的是一种四层转发，也可以理解为四层负载均衡。是通过 iptables（或IPVS）的方式实现的，只支持随机转发。

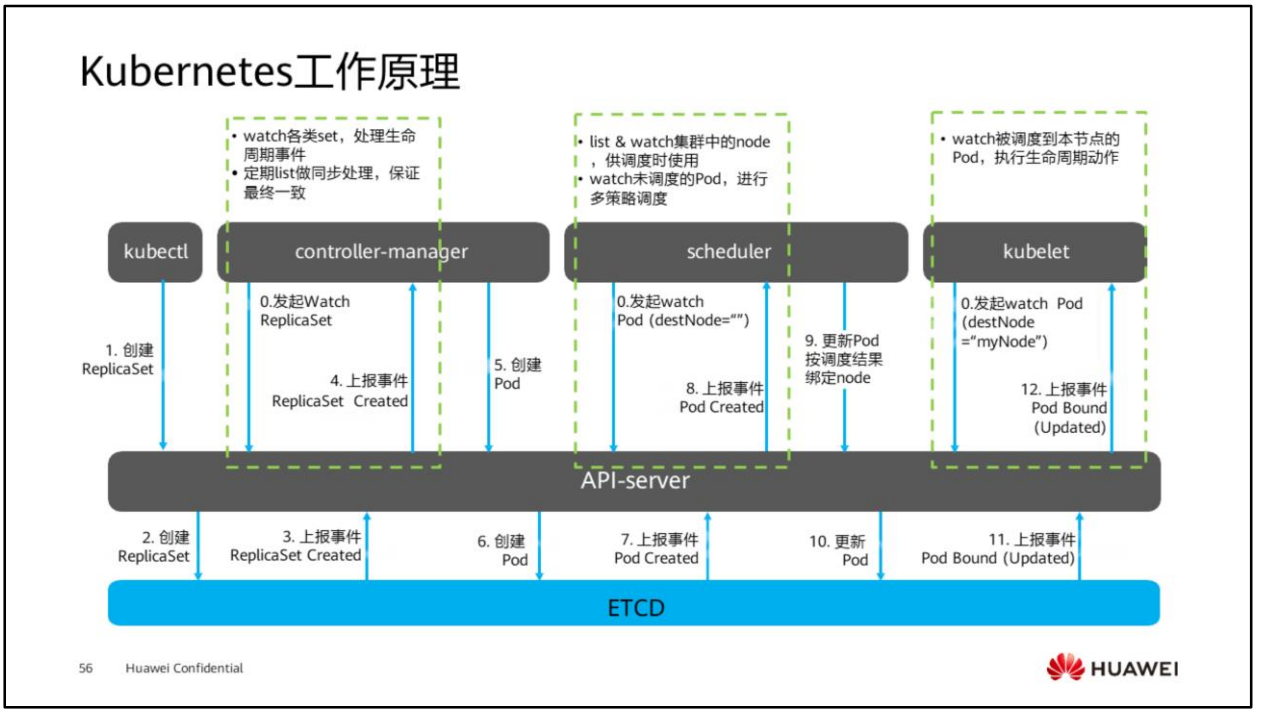
Kubernetes关键概念 - Ingress

- service和pod仅可在集群内部网络中通过IP地址访问。所有到达边界路由器的流量或被丢弃或被转发到其他地方。
- Ingress是授权入站连接到达集群服务的规则集合。
- 可以给Ingress配置提供外部可访问的URL、负载均衡、SSL、基于名称的虚拟主机等。用户通过POST Ingress资源到API server的方式来请求ingress。Ingress controller负责实现Ingress，通常使用负载均衡器，它还可以配置边界路由和其他前端，这有助于以HA方式处理流量。



HUAWEI

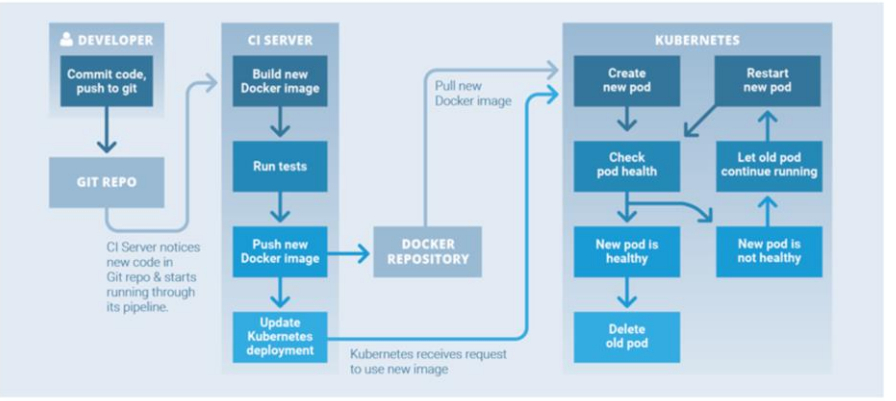
- Service提供了4层的访问，ingress提供了7层的访问，在集群北向提供7层负载均衡能力，对接创建的service实现7层转发能力，一般通过nginx实现网络转发。
- 七层负载均衡是采用了增强型弹性负载均衡，在四层负载均衡访问方式的基础上支持了URL配置，通过对应的URL将访问流量分发到对应的服务。同时，服务根据不同URL实现不同的功能。
- 通过配置公网类型和私网类型的负载均衡实例可以实现公网的七层路由转发和内网（同一VPC内）的七层路由转发。



- 在整个生命周期流程中，controller-manager会持续watch各类set，定期同步，保证最终一致性，而scheduler会持续watch集群的node供调度使用，kubelet会持续watch被调度到本节点的 Pod，执行生命周期动作。

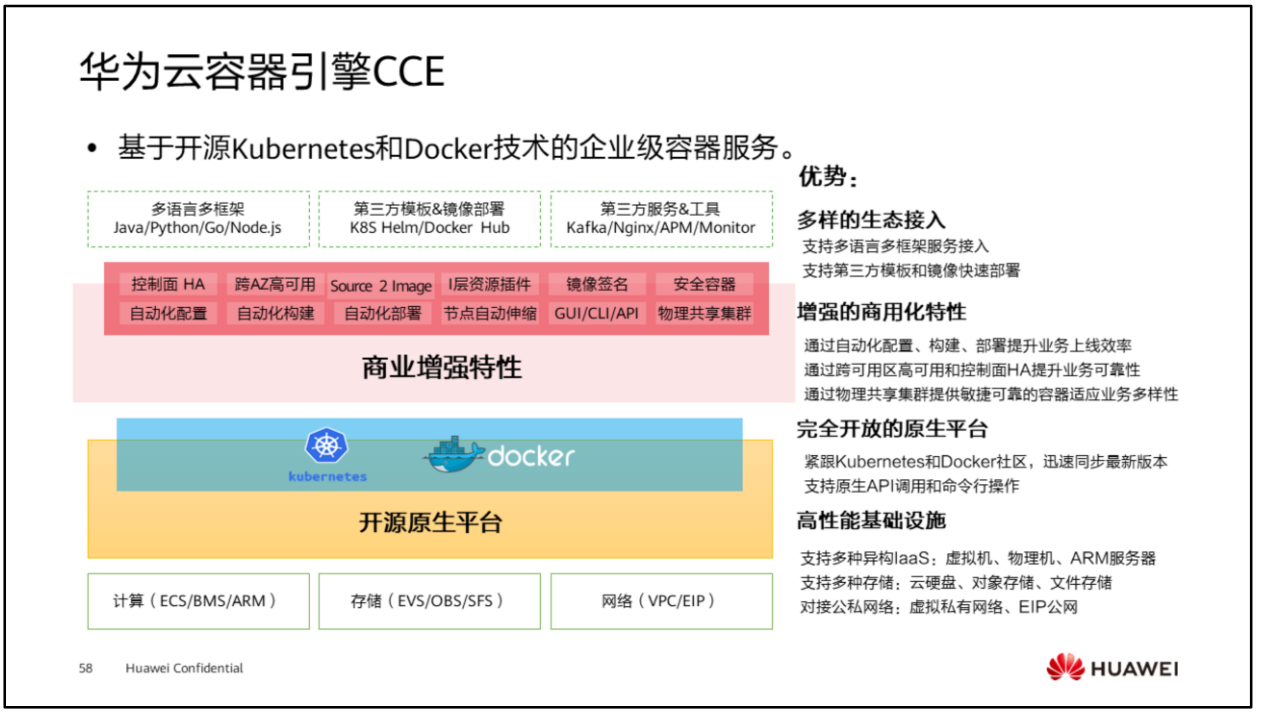
应用容器化之CI/CD

- 容器化技术 Docker 和 Kubernetes 可以方便的与原有的 CI/CD 工具链集成，降低了应用从构建到部署流程的复杂性，加速业务的快速迭代。



HUAWEI

- Docker和kubernetes实现了应用容器化，并且促进了CI/CD的流程，降低了应用从构建到部署流程的复杂性，加速业务的快速迭代。



- 华为云容器引擎CCE -基于开源Kubernetes和Docker，提供了高性能、高可靠的企业级容器服务。
- CCE团队国内最早投入Kubernetes社区的团队，是容器开源社区主要贡献者和容器生态领导者。CCE服务是全国最早的kubernetes商业产品，是全球首批通过CNCF一致性认证的产品。自2014年商用以来，已累计为5000多名用户提供容器化服务。
- CCE的主要价值在于开放和开源的生态、增强的商业化特性、灵活兼容易购基础设施。

CCE产品关键特性



一站式部署和运维

一键创建Kubernetes集群，自动化部署、运维容器应用，整个生命周期都在容器服务内一站式完成。

灵活的弹性伸缩策略

支持按性能、按时间、按周期的弹性伸缩策略，用户可自由组合策略以应对多变的突发浪涌。

可定制化的DevOps能力

支持Java/Python/Go/Node.JS/Vert.x等语言代码开发一键式生成镜像能力。

IaaS资源深度整合

直接使用华为云ECS/BMS、网络(VPC/EIP)、存储（EVS/OBS/SFS）能力。

丰富的应用调度策略

支持多种亲和反亲和调度策略，用户可方便地根据应用特点在高性能和高可靠中找到平衡点。

兼容原生Kubernetes和Docker

紧跟社区，3个月同步最新版本，兼容原生API和命令行。

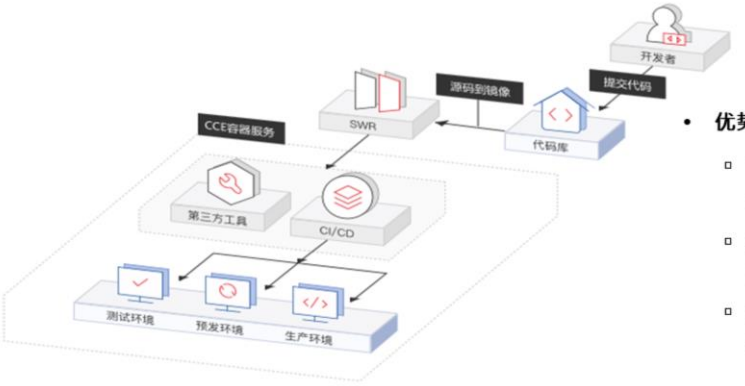
高可用业务不中断

集群控制面HA和跨AZ高可用，容器应用优雅伸缩，安全下线，保障业务不掉线。

HUAWEI

应用场景 - DevOps持续交付

- 可定制化持续交付流程
  - 配合第三方开发流水线开发测试工具，自动完成从代码提交到应用部署的DevOps完整流程，替代部署复杂、迭代缓慢的传统方式，提高企业代码交付和部署的效率。



- **优势**
  - **可定制自动化：**实现从代码变更、代码构建、镜像构建和应用部署的全流程自动化
  - **开放能力：**兼容第三方代码扫描、检测和测试工具，满足开发阶段多样性需求
  - **持续改进：**基于容器的一致性环境，可大幅提升业务迭代速度

HUAWEI

思考题

1. 关于容器技术，以下哪些说法是正确的（）？

   A. Docker镜像文件image，包含了已经打包好的应用程序和运行环境，多个容器无法共享image存储。

   B. 使用“华为云容器引擎CCE”可以一键创建Kubernetes集群。

   C. Docker为应用提供了一个统一的环境，避免应用在不同主机上因为环境不同而产生问题。

   D. K8S中deployment的每个实例对于同一个请求响应的结果未必完全一致。

HUAWEI

- 参考答案：BC

目录

1. 持续交付与持续部署
2. 微服务架构与微服务化应用
3. 容器技术与容器化应用
4. **自动化的实现“一切即代码”**
5. 自动化编译构建

HUAWEI

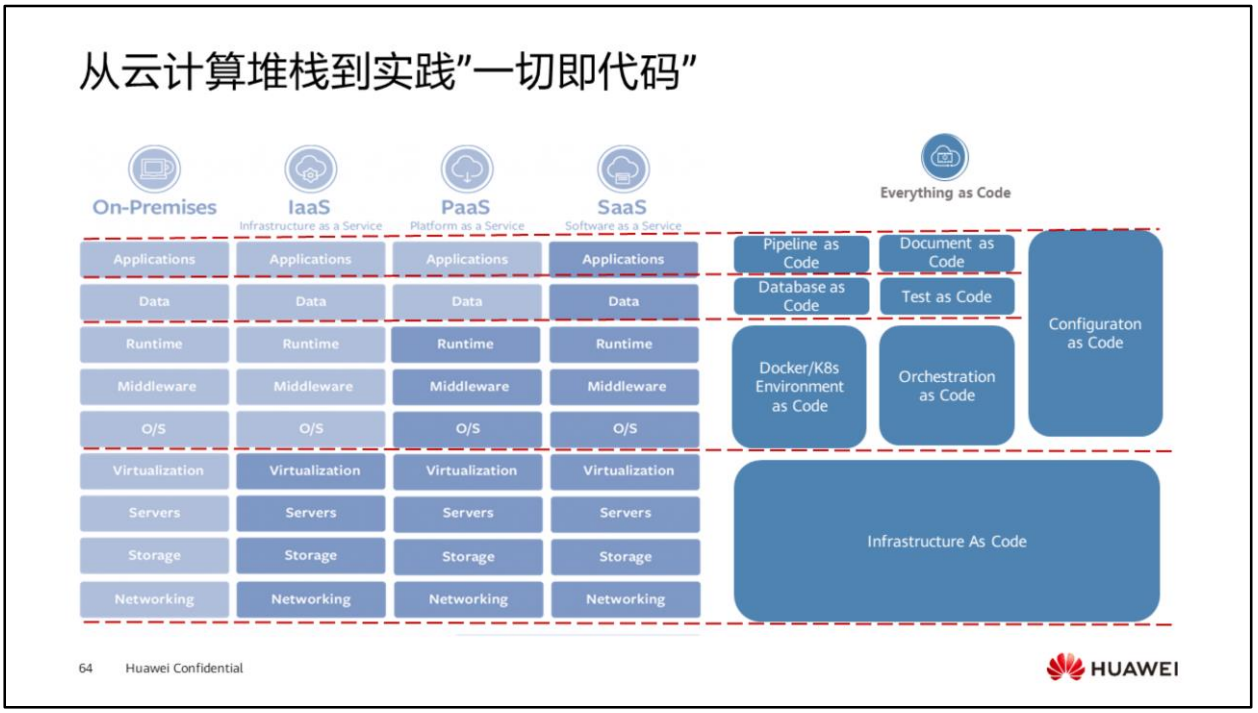
没有自动化之前的软件部署

- 全程人员参与，浪费大量时间。
- 如果节点多，时间就成倍增长。
- 人为干预，可能导致人为失误多。
- 无自动化回滚，或者难以回退。



HUAWEI

- 随着微服务的增多，运维部署的成本也直线上升，如果我们还是采用之前手动的部署方式，无疑会导致工作量的剧增。



- 一切即代码，这个说法最早来源于基础设施即代码。
- 一切即代码实际讨论的就是自动化，利用自动化来处理繁琐的工作，或者在某个任务过于庞大和复杂，无法手动处理时考虑自动化。
- 从基础设施、环境、编排、文档、流水线几个方面探讨自动化实现。

基础设施即代码是什么



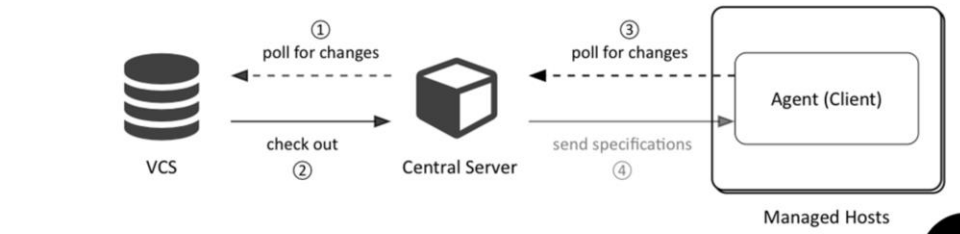
HUAWEI

- 对基础设施的要求：
  - 更快的相应速度
  - 灵活
  - 安全可靠
- 基础设施即代码的目标就是通过可重用的代码来实现IT环境变更的自动化。

Infrastructure | Environment | Orchestration | Document | Pipeline

基础设施即代码解决方案

- Infrastructure as Code基础设施即代码：用代码或配置文件来声明应用系统使用的基础设施资源。

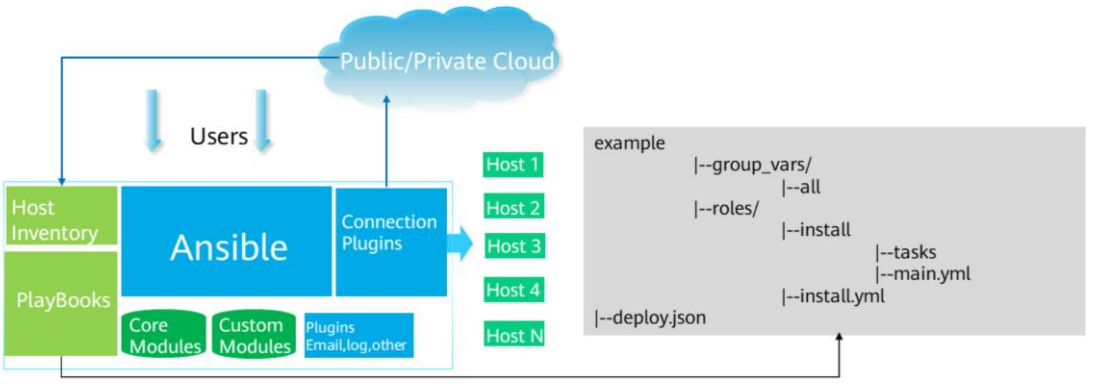


CHEF SALTSTACK puppet labs ANSIBLE

- 一旦采用了环境定义脚本，实现对环境的控制后，需要将环境定义脚本纳入版本管理中，并且之后所有的环境变更都应该先修改环境定义脚本，由环境定义脚本触发。
- 通过自动化部署工具即可以实现“基础设施即代码”，也就是不需要为每一台虚拟机安装基础软件或调整系统参数，一一进行手动升级。只需要把环境的创建过程使用代码的形式描述出来，并且提交到代码库中。任何的环境变更都必须通过修改代码、提交，然后总是使用代码库中的最新版本重新构建环境。这样所有的机器的状态就是可预测的，并且是一致的。常见的开源自动化配置管理工具有Puppet、Chef、SaltStack和Ansible。
- Puppet是这四款工具中最早的自动化配置管理工具，它有完善的报告工具及一些其他功能。详情见：https://puppet.com/
- Chef类似一个大厨，它配置的原理就跟做一道菜一样，一份ruby文件是一份菜谱recipe、而组织菜谱的配置脚本是烹饪手册cookbook、还有到远程连接配置服务器时使用的knife命令工具。详情见：https://www.chef.io/products/chef-infra/
- SaltStack这个配置管理系统，它能够与大规模的系统进行高速通信。详情见：http://docs.saltstack.cn/
- Ansible是轻量级的配置管理工具，功能结构上与SaltStack类似。它可以说是当下较为流行的一个工具，能做到精简快速。详情见：http://www.ansible.com.cn/

Ansible - 轻量级的自动化部署工具

- Ansible部署的基本原理是使用Ansible执行部署任务。Ansible是一款开源的自动化运维部署工具，它使用部署playbook来配置具体的部署任务，这个部署playbook我们可以称之为剧本。



HUAWEI

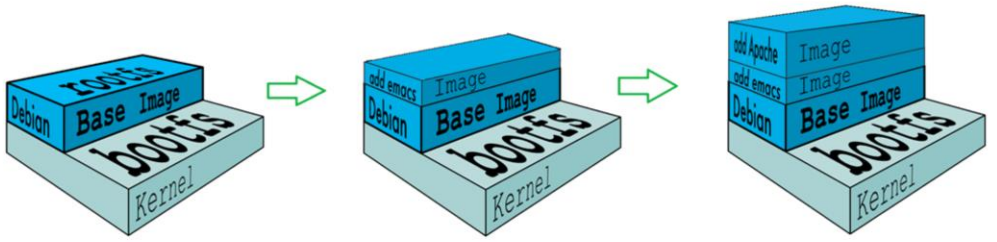
- Ansible是一个开源配置管理工具，可以使用它来自动化任务部署应用程序，实现IT基础架构Ansible可以用来自动化日常任务，比如服务器的初始化、配置、安全基线配置、更新和打补丁系统安装软件包等。
- Ansible与Chef、Puppet以及SaltStack相比，Ansible更轻量级，是无客户端的，仅需通过SSH连接客户机执行任务即可。
- Ansible有控制节点和受控节点，控制节点即安装了Ansible的主机，也叫Ansible服务器端管理，主要用于发布运行任务执行控制命令，Ansible的程序都安装在控制节点上。受控节点，也叫客户机，就是用Ansible执行任务的客户服务器模块。Ansible的架构如下：
  - Ansible：核心引擎
  - Connection plugins：Ansible基于连接插件连接到各个主机上
  - Host inventory：定义可管控的主机列表
  - Playbooks：按照所设定编排的顺序执行完成安排任务
  - Core Modules：自带模块
  - Custom Modules：自定义模块
  - Plugins：完成模块功能的补充
- Ansible现在默认自带450多个模块
- Playbook剧本是利用Yaml标记语言编写的可重复执行的任务的列表，Playbook的优势：
  - 可读性好
  - 和脚本语言的交互性
  - 脚本模块化，配置灵活
  - 每个步骤都配有名称，方便查看结果
  - 易于实现
- "Yet Another Markup Language"（仍是一种标记语言）

Infrastructure | Environment | Orchestration | Document | Pipeline

Dockerfile - 容器镜像构建文件

- 实际上，Docker Hub 中 99% 的镜像都是通过在base镜像中安装和配置需要的软件构建出来的。比如我们现在构建一个新的镜像，Dockerfile 如图：

```
FROM debian ①
RUN apt-get install emacs ②
RUN atp-get install apache2 ③
CMD ["/bin/bash"] ④
```



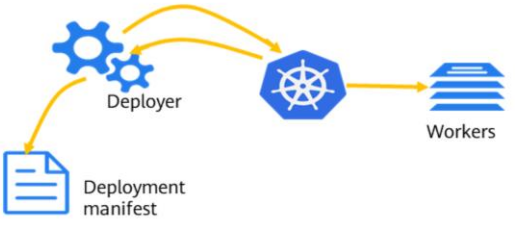
①FROM debian ②RUN apt-get install emacs ③RUN apt-get install apache2

HUAWEI

- DockerFile：文件指令集，描述如何自动创建Docker镜像。
  - Dockerfile是包含若干指令的文本文件，可以通过这些指令创建出dokcer image。
  - Dockerfile文件中的指令执行后，会创建一个个新的镜像层。
- Dockerfile一般由4部分组成：
  - 基础镜像信息
  - 维护者信息
  - 镜像操作指令
  - 容器启动指令
- Dockerfile常见参数：
  - FROM <image>或者FROM <image>:<tag>：基于哪个镜像，<image>首选本地是否存在，如果不存在则会从公共仓库下载
  - WORKDIR：切换目录用，可以多次切换（相当于cd命令），对RUN，CMD，ENTRYPOINT生效
  - RUN：执行指令
  - EXPOSE：EXPOSE指令告诉容器在运行时要监听的端口，但是这个端口是用于多个容器之间通信用的（links），外面的host是访问不到的。要把端口暴露给外面的主机，在启动容器时使用-p选项
  - VOLUME：指定挂载点/data
  - CMD：用于容器启动时指定的服务

Kubernetes manifest

- 通过Kubernetes manifest文件可以用YAML形式来描述在Kubernetes中以Pod的形式运行的应用容器，以及要运行多少个应用的副本。




```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx  image:
        nginx:1.7.9  ports:
        - containerPort: 80
status: {}
```

typeMeta

objectMeta

期望状态

实际状态

HUAWEI

Terraform多云资源管理

- “Multi-Cloud”即企业将多个应用部署在多个云平台上，比如华为云、AWS、Azure、OpenStack或其他云平台。如何对不同云平台的资源进行统一管理?
- Terraform可以用来管理多层次的资源。从上层的软件配置到底层的网络、系统配置。通过一致的模板形态定义基础设施的创建/更新/销毁的全生命周期。
- Terraform使用云提供商 API 来配置基础架构，使用云提供商提供的身份验证机制，构建及访问您的服务器。



HUAWEI

- 可参看Terraform官网：https://www.terraform.io
- Terraform提供了对资源和提供者的灵活抽象，该模型允许表示从物理硬件、虚拟机和容器到电子邮件和DNS提供者的所有内容。
- Terraform与其他系统并不互相排斥，它可以用于管理小到单个应用程序或大到整个数据中心的不同对象。

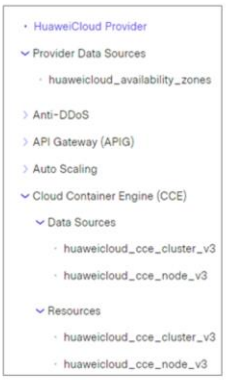
Terraform管理华为云资源

- 下面列举了华为云的模板，可以看到模板形态一致，定义resource，填写不同的参数，比如数量、镜像、实例类型，对于Multi-Cloud的多平台基础设施管理将大幅度降低学习成本。

```
provider "huaweicloud" {
  user_name   = "user"
  tenant_name = "tenant"
  domain_name = "domain"
  password    = "pwd"
  # the auth url format follows: https://iam.{region_id}.myhwclouds.com:443/v3
  auth_url    = "https://iam.cn-north-1.myhwclouds.com:443/v3"
  region      = "cn-north-1"
}

# Create a web server
resource "huaweicloud_compute_instance_v2"  "test-server" {
  name                        = "test-server"
  image_name  = "Standard_CentOS_7_latest"
  flavor_name = "s1.medium"
}
```

使用模板创建一个S1规格的ECS服务器举例



资源类型除了虚拟机，容器集群外，
还有数据路，中间件，安全认证，密钥管理等

HUAWEI

- 详细内容可参看：https://www.terraform.io/docs/providers/huaweicloud/index.html

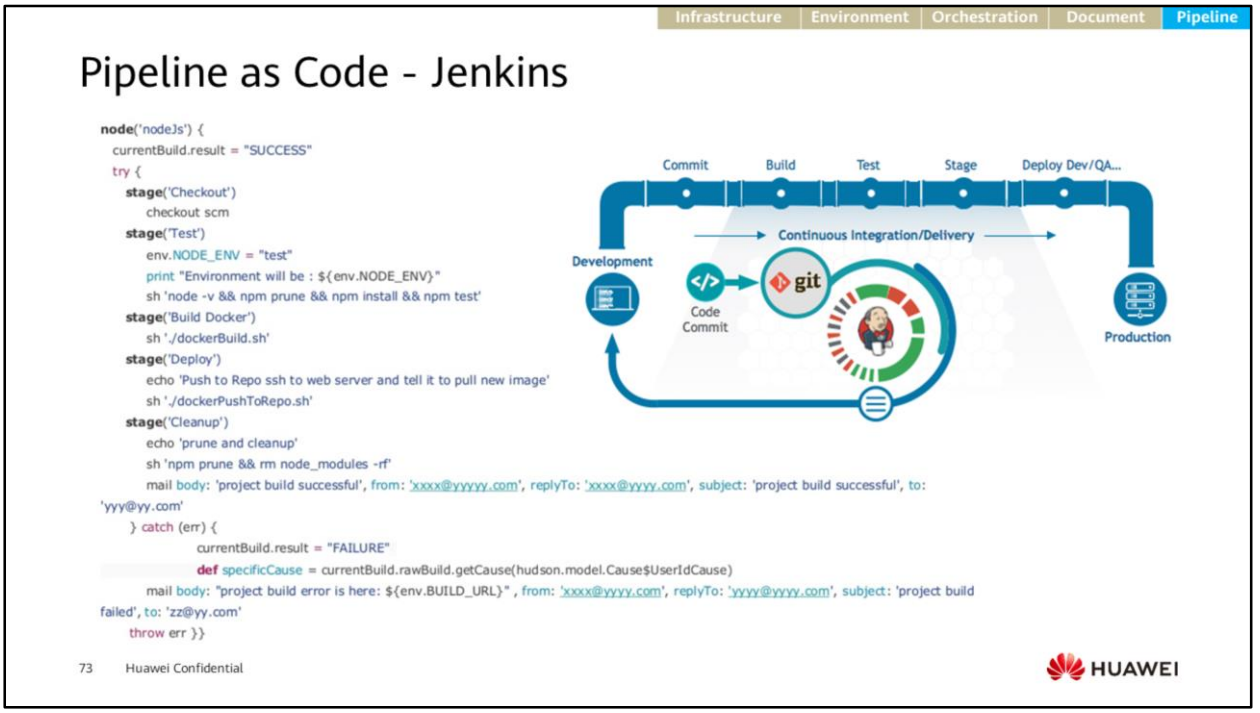
Document as Code - Markdown

- **Markdown**是编写软件文档的最广泛的形式。
- 它是一种轻量级的标记语言，可以轻松转换为HTML或其他格式。
- 通过**Markdown桌面文本编辑器、Markdown浏览器在线编辑器或文档自动生成工具**（例如LaTex）来发布以Markdown格式编写的文档。



HUAWEI

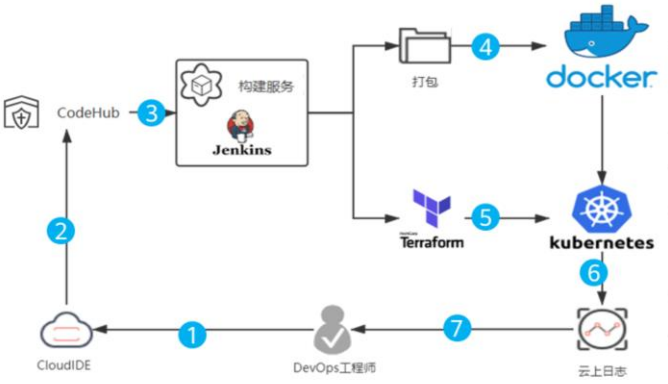
- Markdown成为首选的原因之一是我们几乎可以使用任何纯文本编辑器来创建Markdown文件。
- 我们可以通过Markdown桌面文本编辑器、markdown浏览器、在线编辑器或自动生成软件文档工具来发布以Markdown格式编写的文档。
- 通过git版本控制系统向管理代码一样管理Markdown文件，实现文档及代码。



- 图片来源：https://www.slideshare.net/MichalZiarnik/pipeline-as-code-new-feature-in-jenkins-2-68397885
- Jenkins允许我们在项目的特定目录下放置一个Jenkins file文件，Jenkins file是Jenkins核心特性pipeline的脚本，由groovy语言实现。Jenkinsfile一般放在项目根目录，随项目一起受源代码管理软件控制。
- 好处：
  - 实现流水线上的代码评审、迭代
  - 对流水线进行审计跟踪
  - 作为流水线的单一可信数据员，能够被项目的多个成员查看和编辑
- node为流水线分配了一个执行者和工作区，没有node流水线就无法工作
- Stage代表阶段，一个pipeline可以划分成若干个stage，每个stage代表一组操作

实现”一切即代码”自动化

- 通过Jenkins触发自动集成流水线，自动触发容器镜像生成，并通过Jenkins触发Terraform，打破了云厂商之间的产品界限，实现容器应用部署至Kubernetes中。自动化流程促进了持续交付，促进计算技术的革新与发展。



① 应用程序源代码变更。
② 代码提交到 GitHub/CodeHub。
③ Jenkins/构建服务 持续集成触发器。
④ Jenkins 触发容器镜像生成。
⑤ Jenkins 触发 Terraform，使用镜像构建的容器集群。
⑥ 云上收集并分析日志。
⑦ 监控应用程序并进行改进。

HUAWEI

- Terraform + 容器技术的结合极大降低了云产品的使用难度。而且一套符合标准的基础设施即代码文档是可以被分发及复用的，这有助于打破云厂商之间的产品界限，促进云计算技术的革新与发展。

思考题

1. 关于“一切即代码”，以下哪项说法是正确的（ ）？

   A. 没有自动化前的软件部署，也可做到“回滚”，且人工效率较高。

   B. IaC中，着重强调程序开发中的代码编写，基础设施与应用本身无关，不应被视作代码。

   C. Docker和Kubernets中都使用yaml文件进行描述。

   D. “一切即代码”是一种思想，通过可重用的代码来实现 IT 环境变更的自动化，并对代码进行版本控制。

HUAWEI

- 参考答案：D

目录

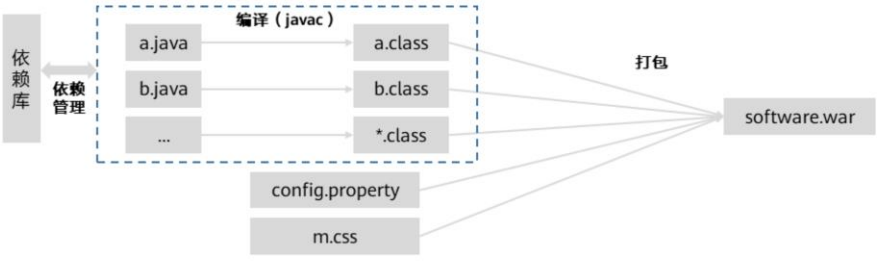
1. 持续交付与持续部署
2. 微服务架构与微服务化应用
3. 容器技术与容器化应用
4. 自动化的实现“一切即代码”
5. **自动化编译构建**

HUAWEI

什么是编译构建

- **编译构建是指把软件的源代码编译成目标文件，并把配置文件和资源文件等打包的过程。**
- **输入**是源代码文件、库文件、配置文件、资源文件等。
- **输出**是软件包：一个可以直接部署的软件，或者是一个可以被其它软件使用的lib。
- **编译器**完成“编译”过程，编译管理工具完成“构建”过程。

下面以一个典型的Java web程序为例，展示其编译构建过程：依赖管理 + 编译 + 打包 = 编译构建



HUAWEI

- Java字节码类文件（.class）是Java编译器编译Java源文件（.java）产生的“目标文件”。它是一种8位字节的二进制流文件

编译构建的挑战



HUAWEI

- 环境搭建耗时费力，且易因环境差异引入问题
  - 研发人员耗费大量精力到环境的搭建配置及调测，不能聚焦业务开发。不仅造成开发人员精力的浪费，也由于时间消耗带来等待成本。同时，很难保证本地构建环境的完全同步，引发问题。
- 本地硬件配置不高，编译构建速度慢
  - 编译构建硬件资源消耗大，但中小型企业和创业者受资金投入限制，硬件配置普遍不高，造成编译构建速度慢，影响开发效率。
- 突发项目资源消耗大，结束后闲置
  - 企业经常有突发性项目，对编译构建资源的需求突发性增加，让企业陷入两难境地，既不想影响突发项目进度，又不愿投入大量资金购买后期闲置设备。
- 多语言不能并行构建
  - 随着互联网的迅猛发展，多语言混合编程成为常态，本地构建环境受硬件限制，难于在同一环境上支持多种构建语言和应用类型并发构建。

编译构建是DevOps的基础

- CI（持续集成）/CD（持续交付）流水线是DevOps的基础实践。
- 编译构建（build）是CI（持续集成）最基础环节。
- CI（持续集成）的其它环节，如代码检查、单元测试等，都高度依赖编译构建。



HUAWEI

流行的编译构建工具

TIOBE Top 10编程语言

Feb 2020	Feb 2019	Change	Programming Language
1	1		Java
2	2		C
3	3		Python
4	4		C++
5	7	^	C#
6	5	v	Visual Basic .NET
7	6	v	JavaScript
8	8		PHP
9	9		SQL
10	20	^^	Swift

对应的部分构建工具

编程语言	构建工具
JavaScript	NPM/Gulp/Grunt
Java	Gradle/Maven/Ant
C/C++	Make/CMake
Python	Setuptool/PyInstaller
C#	MSBuild

HUAWEI

编译构建工具演进史



HUAWEI

- Ant：手动地去进行配置，比如依赖的库，输入输出的位置，甚至在本地安装这些库
- Maven：约定大于配置，简化了用户使用，自动去下载依赖的这些仓库
- Gradle：配置即代码，更强大更灵活

Java构建工具

- 在JAVA开发工具圈中，目前最主流的有以下三个开发工具，依赖管理已经成为了项目构建自动化工具中的一个主要部分。





HUAWEI

- Ant对应Lvy依赖
- Ivy与ant集成是通过在ant的构建文件里定义ivy任务实现调用ivy的功能，从而实现依赖管理的。
- Maven：http://maven.apache.org/
- Ivy：https://ant.apache.org/ivy/
- Gradle：https://gradle.org/

Maven介绍

- Maven 是 Apache 下的一个纯 Java 开发的开源项目，基于项目对象模型（缩写：POM），可以对 Java 项目进行构建、依赖管理。
  - Maven官网链接：http://maven.apache.org/
  - Maven下载链接：http://maven.apache.org/download.cgi
  - Maven安装指导：http://maven.apache.org/install.html



HUAWEI

- Maven是一款跨平台的Apache开源项目管理工具，主要服务于基于Java平台的项目构建、依赖管理和项目信息管理。它奉行约定优于配置（Convention Over Configuration）原则。
- Maven：http://maven.apache.org/

Maven依赖管理

- 在Java世界中，可以用groupId、artifactId、version组成的Coordination（坐标）唯一标识一个依赖。
- pom.xml文件中一个典型的依赖引用如下图，Maven编译时会自动拼接路径和文件名，去本地或远程仓查找。



存储这些组件的仓库有远程仓库和本地仓库之分

HUAWEI

- Maven的世界中拥有数量庞大的软件构件，在引入坐标概念前没有任何一种统一的方式来唯一标识一个构件，人们要想使用这些构件只能在分散的网站上进行搜索，无法自动化，费时费力。
- Maven中定义了一组规则：世界上任何一个构件都可以使用maven坐标来唯一标识。
- Maven坐标的元素包括：groupId、artifactId、version必选，packaging可选，classifier不能直接定义
  - groupId：定义当前Maven项目隶属的实际组织项目（一个组织项目可能包含多个Maven项目）。该元素表示方式与java包名类似，与反向域名一一对应，如org.sonatype.nexus，但通常不建议定于Maven项目对应的组织或公司，因为一个组织或公司下可能会有多个项目。
  - artifactId：定义实际组织项目中的一个Maven项目（模块），通常推荐使用“组织项目名-maven项目名”方式表示以方便后续构件寻找，例如nexus-indexer。默认情况下，Maven中生成的构件名称会以artifactId作为开头，如nexus-indexer-1.2.jar。
  - version：定义当前Maven项目的所处的版本，如1.0.0；2.0.0。
  - packaging：定义当前Maven项目的打包方式，通常打包方式与所生成构件的文件扩展名相对应，如jar、war。
  - Classifier：用来帮助定义构建输出一些附属构件，如xxx-javadoc.jar,xxx-source.jar。

Maven仓库分类



本地仓库

存储在本地磁盘
默认在${user.home}/.m2下



远程仓库

华为开源镜像站作为Maven代理仓库，可以加快jar包下载速度



中央仓库

Maven团队维护的jar包仓库
http://repo1.maven.org/maven2/

HUAWEI

Maven仓库搜索顺序



本地仓库搜索	远程仓库搜索	中央仓库搜索
• 本地仓库找到，直接返回 • 本地仓库没有找到，去远程仓库搜索	• 没有配置远程仓库，去中央仓库搜索 • 远程仓库找到，下载到本地仓库 • 远程仓库没有找到，搜索下一个远程仓，依次类推，如果所有远程仓都未找到，搜索中央仓	• 中央仓库找到，下载到本地仓库 • 中央仓库没有找到，前台打印错误信息

HUAWEI

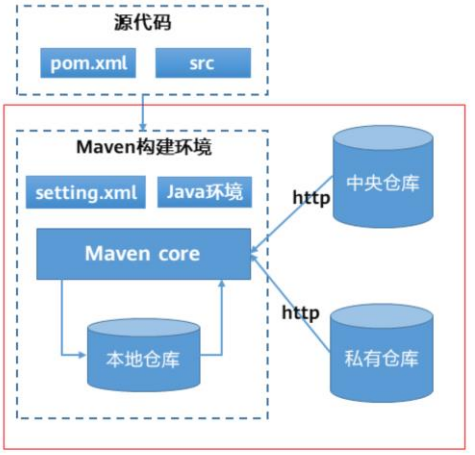
- 如果配置了mirror ， mirrorOf 匹配的repository会被替换。
- 例如：

```
<mirror>
    <id>spring</id>
    <url>https://repo.spring.io/plugins-release/</url>
    <mirrorOf>1central</mirrorOf>
</mirror>

<repositories>
      <repository>
        <id>1central</id>
        <url>https://mirrors.huaweicloud.com/repository/maven/kunpeng1/</url>
        <releases>
          <enabled>true</enabled>
        </releases>
       </repository>
```

编译构建实践 - Maven构建

- 最主流的java构建工具Maven：需要大量配置和维护
- DevCloud：选择一个Maven模板，一键式生成Maven构建环境并完成配置

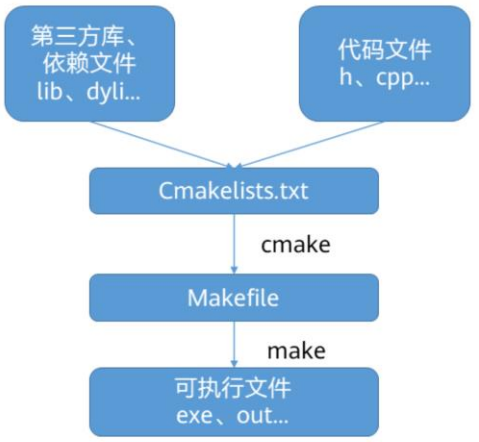




HUAWEI

编译构建实践 - CMake构建

- 最主流的C/C++构建工具CMake：需要大量配置和维护



- DevCloud：选择一个CMake模板，一键式生成CMake构建环境并完成配置



HUAWEI

- 过CMake生成Makefile
- 在通过Make生成所谓的exe

编译构建实践 - 容器镜像构建

- Dockerfile是一个文本格式的配置文件，包含创建镜像所需要的全部指令。基于在Dockerfile中的指令，用户可以快速创建自定义的镜像。
  - FROM 指定一个基础镜像
  - RUN运行命令
  - COPY添加本地数据到容器
  - CMD容器默认运行的命令

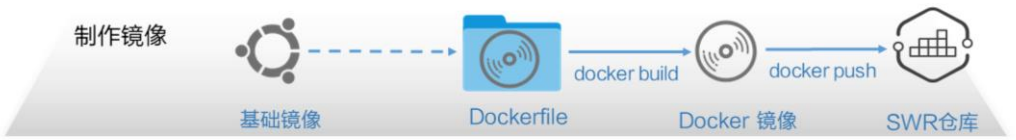
```
#选择ubuntu最小镜像
FROM ubuntu

#创建工程路径
RUN mkdir -p /opt/gift-server

#软件包到容器
COPY gift-server/target/gift-server-1.0.0.jar /opt/gift-server

#拷贝启动脚本
COPY gift-server/run.sh /opt/gift-server

#设置容器启动时执行
CMD ["/opt/gift-server/run.sh"]
```



HUAWEI

云上构建优势

传统的、企业本地的编译构建：花费大量人力物力，维护成本高

学习并搭建环境

- 专职/兼职的持续集成工程师
- 学习持续集成理论/技能，熟悉构建工具
- 购买虚拟机
- 安装工具（Jenkins, java, Maven ....）
- 配置/搭建系统
- 配置job(Jenkins job)

持续维护

- 专职/兼职的持续集成工程师
- 工具升级（Jenkins, java, Maven...）
- 不断填坑，解决问题
- 资源不够，扩容
- 不断增加新环境（支持新语言，新场景）

DevCloud提供开箱即用的、专业的编译构建服务：学习成本、使用成本低，无需维护

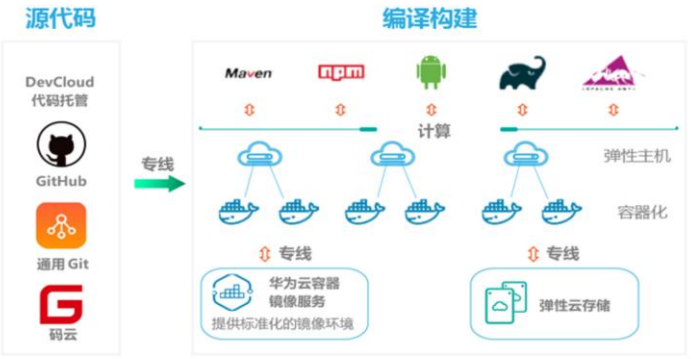
- 服务即工具，开箱即用
- 预置了你需要的大部分环境，并由DevCloud维护
- 资源弹性，按需使用
- 新语言，新场景支持由DevCloud专业提供
- 跟其它DevOps服务无缝集成，一站式使用



HUAWEI

DevCloud编译构建服务

- 华为云DevCloud编译构建服务（CloudBuild）为开发者提供配置简单的混合语言构建平台，实现编译构建云端化，支撑企业实现持续交付，缩短交付周期，提升交付效率。

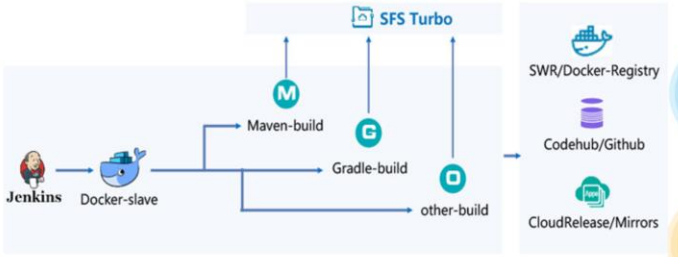


HUAWEI

- 适用于不同来源的代码，可以对不同的语言进行编译构建，并且可以通过编译构建服务预置好所依赖的第三方镜像仓等，生成可运行的包文件。
- 图片来源：华为云，详情见https://support.huaweicloud.com/productdesc-codeci/devcloud_pdtd_40001.html

DevCloud编译构建特性

- 将Jenkins配置的构建任务放入容器中运行，借助弹性文件服务下的SFS Turbo，对依赖包进行缓存。构建时优先从缓存中取包，没有找不到包才会去仓库下载，以达到加速构建目的。



全场景

支持10+种语言，15+种框架，覆盖主流软件开发场景

C/C++/Java/Python/NodeJS/C#/Android Mave/Gradle/Ant/Npm/MSbuild/CMake

快速

并行、缓存，网络多种加速技术

- 云上弹性资源，任务并行执行
- 全局和租户两级缓存
- 华为云专线网络，传输更快

易扩展

灵活对接不同的代码托管服务和构建环境

- 可调度用户自定义构建环境
- 支持不同的源码托管服务
- 可调度用户的持续集成集群

HUAWEI

- 租户缓存：Maven/Gradle使用的缓存
- 全局：Android SDK
- 图片来源：华为云，详情见https://support.huaweicloud.com/productdesc-codeci/devcloud_pdtd_40001.html

思考题

1. 关于编译构建，以下哪些说法是正确的（ ）？

   A. 云上构建花费大量人力物力，维护成本高。

   B. 华为云DevCloud编译构建服务支持多种语言的构建，如Maven，Cmake。

   C. 用户可以在华为云DevCloud编译构建服务中，基于在Dockerfile中的指令，快速创建自定义的镜像。

   D. 编译构建是指把软件的源代码编译成目标文件的过程，无需关注配置文件和资源文件。

HUAWEI

- 参考答案：BC

目录

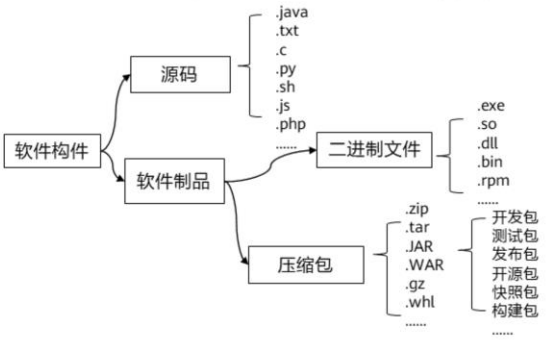
6. **制品和包管理**
7. 自动化部署
8. 发布管理
9. 自动化交付流水线

HUAWEI

- 本章一共9个小节：
  - 持续交付与持续部署
  - 微服务架构与微服务化应用
  - 容器技术与容器化应用
  - 自动化的实现“一切即代码”
  - 自动化编译构建
  - 制品和包管理
  - 自动化部署
  - 发布管理
  - 自动化交付流水线

软件制品的基本概念

- 软件研发过程中的**“源码”**和**“软件制品包”**（通常被通俗称为**“二进制包”**）都是很关键的资产。软件制品包通常是源码文件的集合或者编译后的产物，因此主要有二进制包和压缩包两种形式。软件制品包的管理和复用在发布管理有着关键的作用。
  - 源码文件：文件名、版本、大小、创建时间
  - 软件制品文件：文件名、版本、大小、创建时间、依赖、许可、风险



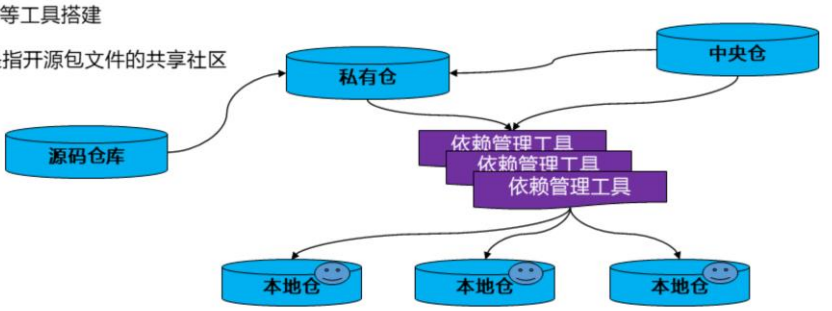
源码文件	包文件
经常频繁修改	修改较少
一般较小	通常较大
增量修改	覆盖修改
修改增量存储	修改全量存储
频繁对比，分支，标签	基本没有
属性值较少	属性值多
异地分发简单	异地分发较困难

HUAWEI

- 软件制品相对于源码文件，通常较大（M~G级别都有），一般修改也较少且通常采用覆盖方式。
- 软件制品一般的生命周期环节也较多，因此需要管理更多的元属性来标识不同阶段的状态，通常软件制品不放在源码库中一同管理。

软件制品的一般管理方法

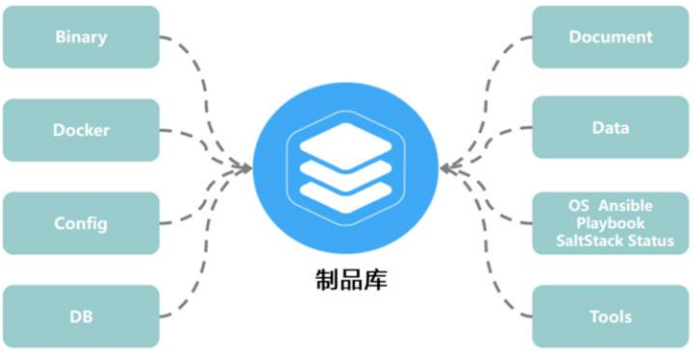
- 包文件通常不放在源码库中管理，而是使用专门的包文件仓库（repository）进行存储并配合包文件依赖管理工具（Maven、NPM、Ivy等）进行使用。包文件仓库可以大致分为本地仓库、私服仓库、中央仓库三种。
  - 本地仓库是指开发者个人PC中包文件的存储
  - 私服仓库通常是企业为了提升包文件使用性能搭建的局域网内共用的包文件仓库，通常使用开源的Nexus、Artifactory等工具搭建
  - 中央仓库是指开源包文件的共享社区



HUAWEI

软件制品库的基本概念

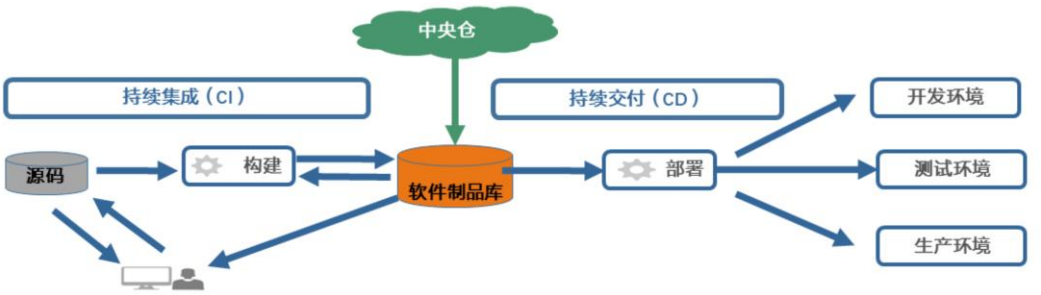
- 什么是软件制品库？指能够统一管理各种类型的二进制制品，同时无缝对接现有的标准化构建和发布工具的软件平台。



HUAWEI

软件制品库在DevOps工具链中的作用

- **“软件包”**及其属性的管理是发布过程管理的基础，也是软件开发过程中的重要资产。
- **“软件制品库”**，用于管理软件开发过程产生的**“软件包”**，它是连接持续集成和持续交付的重要环节，软件包的发布评审、追溯和安全控制等操作通常在其中进行。



HUAWEI

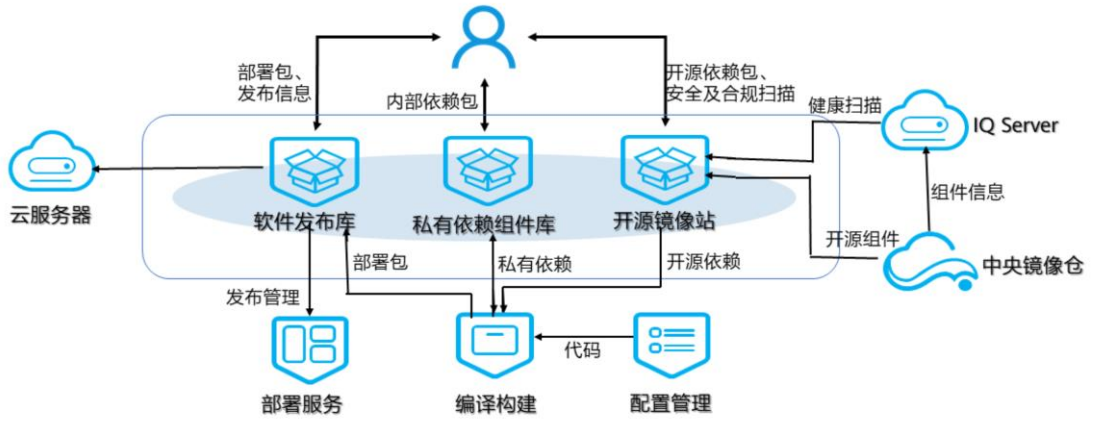
- 软件服务变更通常在一个分布式、复杂的环境中进行。发布管理对于保障软件服务正确快速变更有种不可或缺的意义，发布管理能力除了是交付过程必备环节，随着对软件交付速度不断提升的诉求及激烈的市场竞争，它也逐渐成为一种核心竞争力。
- 软件开发交付过程中，随着松耦合架构的推广，开源构件及框架的广泛使用，持续集成（CI）与持续交付（CD）等自动化手段的逐渐普及，软件构件的复用及依赖管理对发布准确性和及时性的影响越来越大，逐渐受到重视并日益规范。特别是在中大型项目中，良好的构件管理能带来较大的发布效率和软件质量提升。
- 软件制品库的作用如下：
  - IDE对接
  - 快速获取依赖包
  - 识别开源包风险
  - 中央仓库对接
  - 开源包镜像，提升下载效率
  - 开源包策略管理，过滤有害包文件
  - 构建工具对接
  - 复用私有包，提升构建效率
  - 开源依赖包获取，提升构建效率
  - 部署工具对接
  - 利用有序调度不同环境部署，实现持续交付

典型的软件制品

- 软件制品有多种格式，如Maven、Pypi、NPM等：
  - Maven是一个项目管理工具，它包含了一个项目对象模型 (Project Object Model)，一组标准集合，一个项目生命周期(Project Lifecycle)，一个依赖管理系统(Dependency Management System)，和用来运行定义在生命周期阶段(phase)中插件(plugin)目标(goal)的逻辑。当你使用Maven的时候，你用一个明确定义的项目对象模型来描述你的项目，然后Maven可以应用横切的逻辑，这些逻辑来自一组共享的（或者自定义的）插件。
  - NPM的全称是Node Package Manager，是一个NodeJS包管理和分发工具，已经成为了非官方的发布Node模块（包）的标准。
  - pip 是一个现代的，通用的 Python 包管理工具。提供了对Python 包的查找、下载、安装、卸载的功能。

HUAWEI

DevCloud中的软件制品库

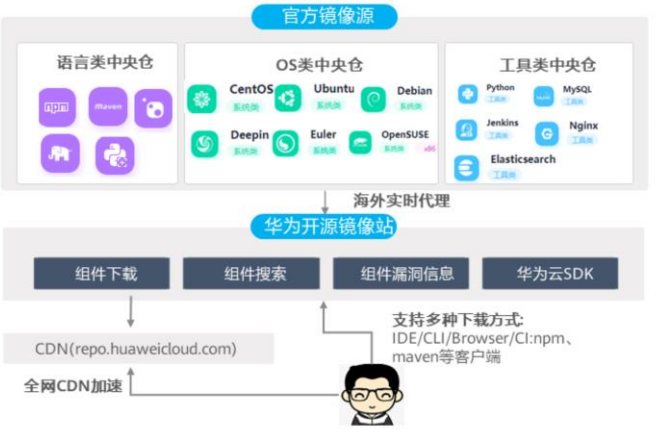


HUAWEI

- 私有依赖库： 私有依赖库是由华为云提供的二方组件仓库，提供组织内部私有二进制组件的管理与共享，目前只支持Maven格式的组件管理，下一步将提供NPM、Python、GO等组件的管理
- 软件发布库：是面向软件开发者提供软件发布的云服务，提供软件仓库、软件发布、发布包下载、发布包元数据管理等功能，通过安全可靠的软件仓库，实现软件包版本管理，提升发布质量和效率，实现产品的持续发布
- 开源镜像站：华为开源镜像站（Mirrors）是由华为云提供的三方组件仓库，主要提供开源组件、开源操作系统及开源DevOps工具等下载，致力为用户提供全面、高速、可信的开源组件/OS/工具下载服务
- 容器镜像服务：华为云容器镜像服务（SoftWare Repository for Container）是一种支持容器镜像全生命周期管理的服务，提供简单易用、安全可靠的镜像管理功能，帮助用户快速部署容器化服务

华为开源镜像站

- 华为开源镜像站（Mirrors）是由华为云提供的三方组件仓库，主要提供开源组件、开源操作系统及开源DevOps工具等下载，致力为用户提供全面、高速、可信的开源组件/OS/工具下载服务。



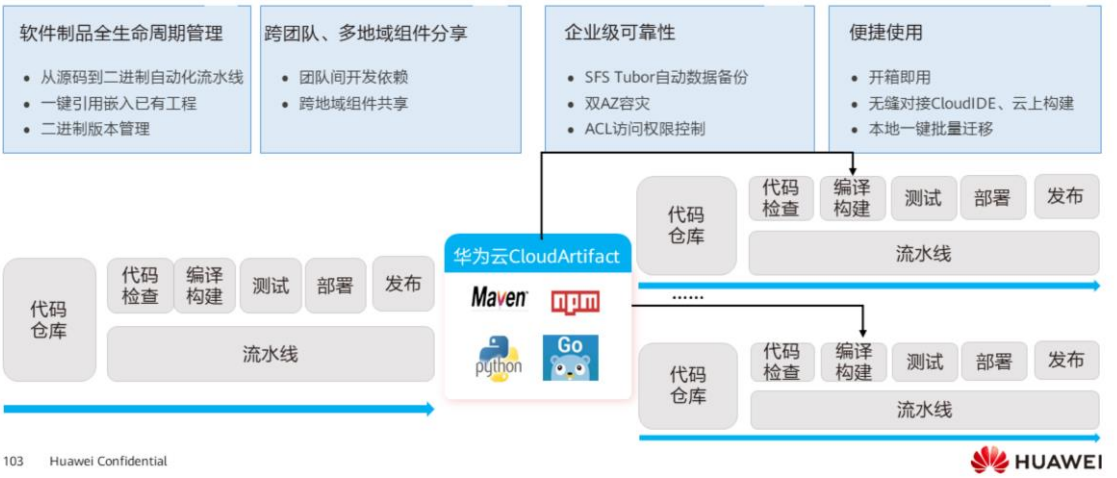


HUAWEI

- 详情请访问：https://mirrors.huaweicloud.com/
- 华为开源镜像站会及时的同步官方的镜像元对组件进行漏洞扫描，保证安全，并提供多种下载方式。
- 目前华为云开源镜像站已经提供了100多种安全可信的镜像，并成为了Maven Central唯一官方镜像库

私有依赖库CloudArtifact

- 私有依赖库是由华为云提供的二方组件仓库，提供组织内部私有二进制组件的管理与共享，目前只支持Maven格式的组件管理，下一步将提供NPM、Python、GO等组件的管理。



HUAWEI

- 私有依赖库是提供企业内部管理，私有公共组件的私服，适配各种语言及依赖包管理工具。

发布库CloudRelease

- 发布库：为软件开发企业提供管理软件发布过程的能力，保障软件发布过程的规范化、可视化及可追溯。相对于开发过程中的“源代码”，发布服务关注和管理的是开发产生的待部署的“软件包”（通常由源码编译构建或打包而成）及其生命周期元数据（如名称、大小等基本属性、由哪些源码生成、构建时间等）。
- 发布库服务目标：通过安全可靠的软件仓库，实现软件包版本管理，提升发布质量和效率，实现产品的持续发布。

HUAWEI

发布库特性

- 丰富的文件操作：支持文件重命名、批量删除、批量恢复、上传、下载、文件名搜索等操作。
- 构建包自动归档：编译构建属性自动关联软件包，编译构建产物自动归档到发布库。
- 服务无缝集成：支持构建服务快速上传、部署服务从发布库快速取包部署。
- 多视图追溯：提供目录、构建和发布三种视图支持用户进行软件包生命周期追溯。



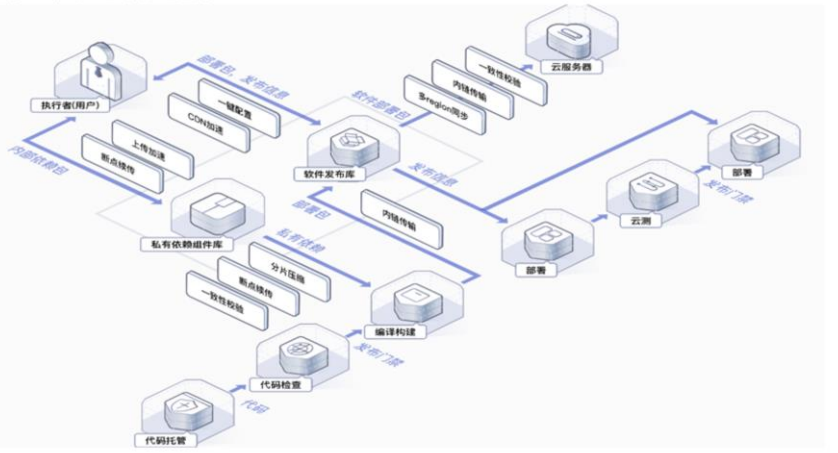


HUAWEI

- 利用软开云的流水线能力，调度起在其上的构建任务和部署任务
- 利用构建任务执行过程中的“上传软件包到软件发布库”动作完成软件包的上传操作
- 利用部署任务的软件包来源从指定软件包或者构建任务来从发布库进行下载

DevCloud发布库场景流程图

- 开发者可以基于发布服务提供的多类型软件仓库能力，配合编译构建、部署、流水线服务快速搭建起持续交付自动化能力，实现快速发布。



HUAWEI

- 图片来源：华为云

什么是容器镜像管理

- 容器化程度越来越高，企业中使用容器部署的应用越来越多，容器镜像也越来越多，为保证容器镜像的安全性和可管理性，需要制定标准的镜像管理组织方式。
- 企业应用镜像需要标准化管理，推荐以下几个层次的镜像管理：
  - OS 层：OS 的基础镜像，包含各种 Linux 操作系统及不同版本
  - 依赖层：各种与中间件的运行环境，如 java，python，go 等
  - 应用中间件层：各种软件层，包括中间件，数据库等
  - 用户业务应用层：业务应用层，比如基于 tomcat 的各种 OA，ERP 等业务应用



HUAWEI

如何进行容器镜像管理

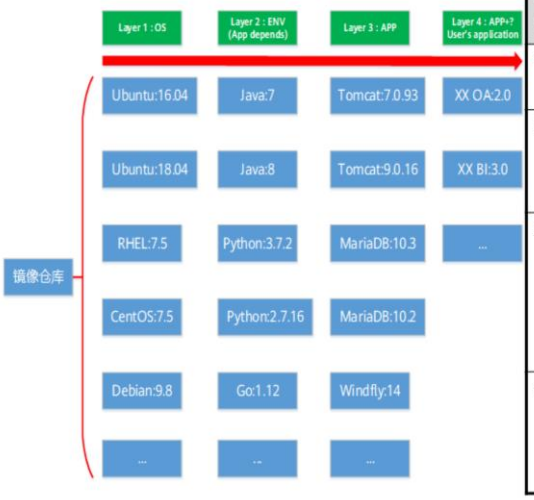
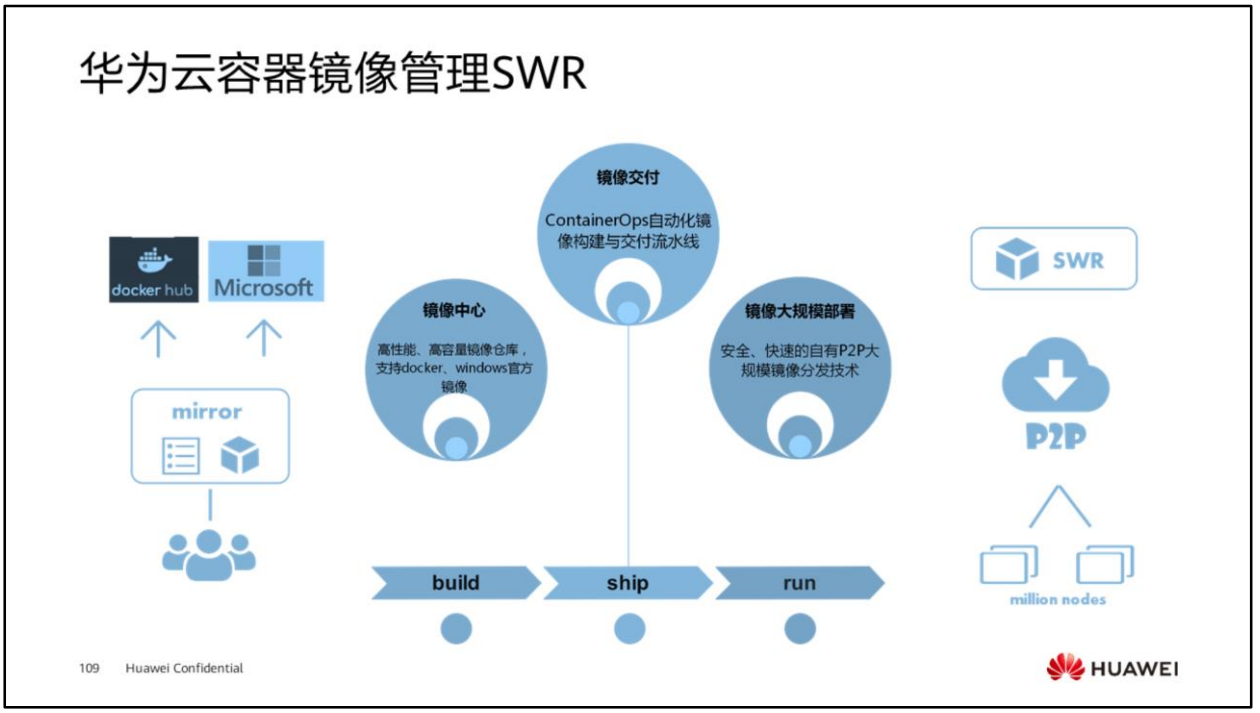


Image UUID	Type	Name	Build or Buy	Base Image	Maintainer	Note
xxxxx	OS基础镜像	rhel7	Buy	/	Operations	可自己构建或者向厂商订阅获取
xxxx	依赖层镜像	Java7	Build	rhel7	Operations	供应商支撑比较差，一般需要自己构建，或者从Docker Hub 获取
xxx	中间件软件镜像	Tomcat7	Build	Java7	Middleware Team	相对稳定，需要新功能或安全补丁时才考虑版本升级。一般需要自己构建，或者从Docker Hub 获取。
xx	用户业务应用镜像	OA2.0	Build	Tomcat7	Development Team	供应商支撑比较好，一般能提供镜像。如果是自己开发的应用，需要自己构建。

HUAWEI

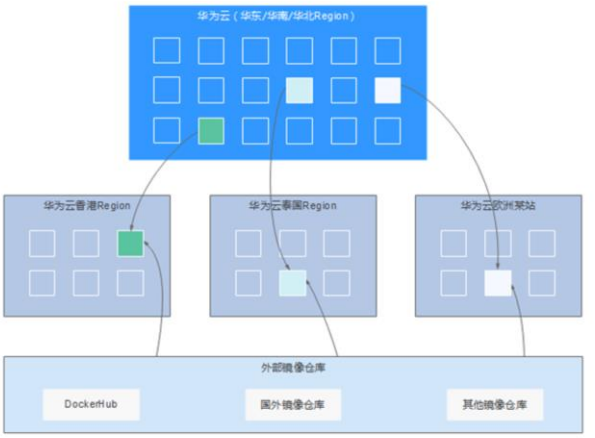
- 说明：
  - UUID为镜像的UUID，可以通过docker images 查看
  - 类型为四种镜像中的一种
  - 镜像名称可以通过docker images 查看
  - 自建或购买表示镜像的来源
  - 父镜像表示本容器镜像的依赖镜像，可通过容器镜像的Dockerfile的From字段获取



- 容器镜像服务（Software Repository for Container，简称SWR）是一种支持镜像全生命周期管理的服务， 提供简单易用、安全可靠的镜像管理功能，帮助您快速部署容器化服务。您可以通过界面、Docker CLI和原生API上传、下载和管理Docker镜像。
- 容器镜像服务可配合云容器引擎CCE、云容器实例CCI使用，也可单独作为容器镜像仓库使用。
- 图片来源：华为云

容器镜像管理之镜像加速

- 华为自主专利的镜像下载加速技术，使用CCE集群下载时可确保高并发下能获得更快的下载体验。
- 智能调度全球构建节点，根据所使用的镜像地址自动分配至最近的主机节点进行构建，可以拉取国外镜像；根据负载自动分配到空闲节点，可以加速镜像的获取效率。

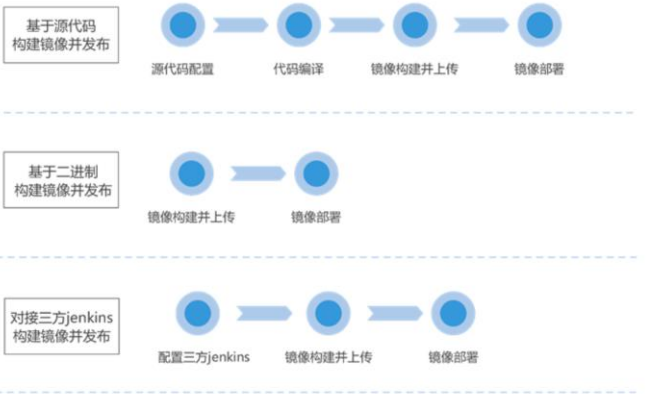


HUAWEI

- 图片来源：华为云

容器流水线ContainerOps

- **容器化底座，实现资源的快速弹性伸缩，即需即取**
  - 流水线容器化架构，环境依托Kubernetes构建，底层设施可根据使用量的大小自动伸缩，提升了流水线效率，节约维护人力
- **内置多种典型CI/CD流水线，开通服务即可使用**
  - 参考华为内部十多个产品的已有实践，结合市场用户调研，ContainerOps系统中默认提供了近十种流水线，用户可以根据自身需要直接使用



HUAWEI

- 图片来源：华为云

思考题

1. 以下哪些是华为云存放软件制品的功能或服务?
   A. 华为云容器镜像管理SWR
   B. 发布库CloudRelease
   C. 私有依赖库CloudArtifact
   D. 代码仓库CodeHub

HUAWEI

- 参考答案：ABC

目录

6. 制品和包管理
7. **自动化部署**
8. 发布管理
9. 自动化交付流水线

HUAWEI

部署的发展历程



手工管理时代

- IP地址存放在Excel表里。
- 用SSH连接服务器。
- 手工执行，进行部署配置等工作。



自动化时代

- 数据库方式管理IP地址—CMDB。
- 基于ssh或者agent的分布式脚本执行平台。
- 运用自动化工具，用脚本自动化执行。



DevOps阶段

- 统一支持开发和运营的交付工作及环境管理工作。

HUAWEI

- 部署一共经过三个阶段的发展
  - 单体结构部署的过程，较为简单，通过手工操作就可以完成。
  - 随着分布式应用的普及，出现了很多自动化部署工具，采用脚本的形式，按照提前准备好的逻辑，按序执行部署
  - 以应用为中心的微服务快速发布的诉求，通过流水线出发部署任务，实现标准化，版本化的部署

传统部署下的矛盾

- 在一些企业中，存在开发人员手动部署的方式，每次需要发布、更新都要远程连接到服务器上，手动地部署新版本。
  - 容易出错、效率低
  - 依赖人员、耗费大量人力成本
  - 流程繁琐、操作复杂
  - 对于机器集群部署服务实现困难



- 所以，在随着公司的业务的不断扩大、项目迭代速度变快，实现自动化部署显得尤为重要。

HUAWEI

自动化部署

- “自动化部署”将可交付产品，快速且安全地交付用户使用的一套系统和工具。 系统会自动构建、测试并准备代码变更，以便将其发布到指定环境的过程，包括开发环境、预发布环境、生产环境等。

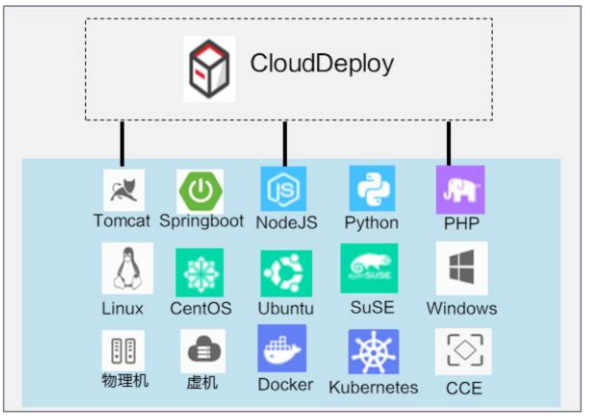


HUAWEI

- 自动化部署，包含应用管理、环境管理、部署、编排、执行等功能。

DevCloud部署服务

- 华为云CloudDeploy服务提供自动化部署能力，可实现如下功能：



部署应用到物理机、虚机、容器

- 支持将应用部署到物理机，虚拟机，容器。
- 支持主机有Agent或无Agent模式提供应用监控数据。
- 支持以代理机的方式进行部署。
- 支持部署到应用管理平台，纳管容器云集群或客户的私有K8S的集群。

支持多种技术栈应用的部署

- 支持Tomcat,Springboot,Nodejs等多种技术栈。
- 支持SpringCloud，ServiceComb，Dubbo等微服务应用部署和治理能力。
- 提供通用模板，且支持自定义模板，提供25+原子步骤组装成部署任务。

支持与流水线，运维管理无缝集成

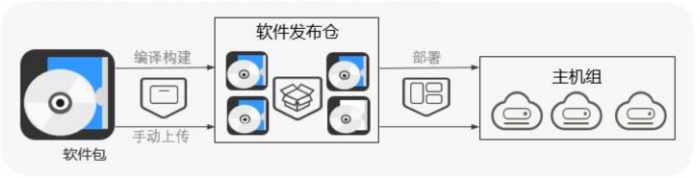
- 在流水线能关联部署任务。
- 通过流水线关联构建，代码检查，测试等服务，实现端到端Devops流程。
- 应用部署后的可运维管理。

HUAWEI

- 图片来源：华为云

部署服务的工作原理

- **云主机部署**用户通过手动上传、或者编译构建任务，将软件包保存在软件发布库中；部署任务将软件包上传并安装到云主机中。



- **容器部署**用户通过手动上传、或者编译构建任务，将镜像保存到容器镜像仓中；部署任务将镜像上传到容器镜像集群中。

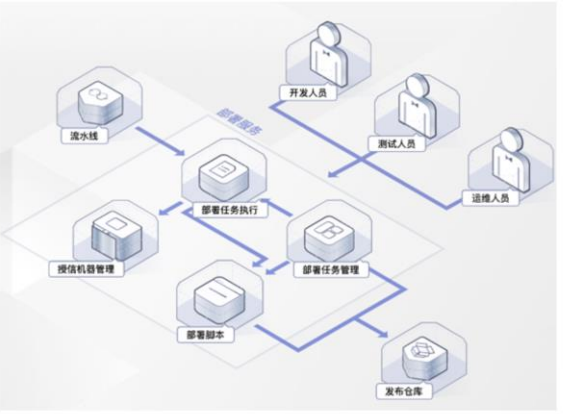


HUAWEI

- 图片来源：华为云

部署服务功能模块

- 华为云软件开发服务提供一键式自动化部署，部署机器无需安装代理，过程可视化，主持主流编程语言和技术框架，支持并行部署和流水线无缝集成，实现部署环境标准化和部署过程自动化。



HUAWEI

- 自动化部署作为流水线中间的一环，它位于构建任务与发布之间。
- Devcloud提供Ansible部署框架，基于Ansible的部署，目标主机无需安装代理。
- 开发、测试、运维人员通过统一平台进行部署，部署环境标准化。
- CloudDeploy服务中包含部署任务执行，部署任务管理，主机管理的模块，部署过程可视化。

部署服务关键特性 - 系统模版

- 使用系统模板快速创建部署任务：提供SpringBoot、Tomcat、CCE容器应用部署、Docker应用部署等系统模板快速创建部署任务。
- 支持保存自定义模板：可以将组合的部署步骤保存为自定义模板，其他部署任务就可以直接使用该模板。



HUAWEI

- 图片截图来源于华为云

部署服务关键特性 - 批量部署

- 实现批量部署：多个目标主机并行执行部署任务。
- 主机组：主机组是多台主机的组合。将多个主机添加到某一主机组中，通过该主机组在多台主机上并行执行部署任务。
- 环境：环境用于管理一组网络互通的资源，可预定义生产环境，测试环境等。环境中可添加主机组，数据库，容器集群等资源，作为部署任务的对象。



HUAWEI

- 环境是计算、存储、网络等基础设施的集合。环境内部网络互通环境中，通常可以包含主机、集群、数据库、ELB、微服务、引擎等资源。用户通过创建不同的环境实现不同资源的分组管理，如开发环境、测试环境等等。

部署服务关键特性 - 自定义组合部署步骤

- 提供25+部署步骤自由组合任务





HUAWEI

- 图片截图来源于华为云

部署服务关键特性 - 支持参数化

- 支持部署任务内容参数化：部署任务支持参数设置并动态执行，通过配置任务参数用户可以根据自定义的参数执行任务。



HUAWEI

- 通过${变量}引用动态的参数。

Tomcat部署步骤示例



HUAWEI

- 图片截图来源于华为云

部署服务应用场景

- **简单Web应用**
  - 应用：用户需要开发简单Web应用并使用软件开发云编译构建和部署。
  - 适用场景：适用于Web应用程序和微服务的部署，支持Tomcat、Java、Python、NodeJS、Go、PHP、Ruby语言。
- **微服务容器部署**
  - 应用：大型应用软件网站有很多的组件或微服务组合协调，需要软件开发云实现容器化部署。
  - 适用场景：支持Linux下多服务、多组件、多层次的大型应用，用户需要自定义部署模板，定制适合自己应用的部署脚本和部署参数，并应用流水线编排各个组件的统一部署，实现全系统的自动化部署。

HUAWEI

思考题

1. 以下关于自动化部署的说法，哪些是正确的（ ）？

   A. DevCloud部署服务中，可以实现多个目标主机的批量部署。

   B. DevCloud部署服务中，支持部署模板，选择模板后不可以更改部署步骤。

   C. 使用开发人员手动部署的方式进行部署时，一旦出现问题，可以马上发现并着手修复。

   D. DevCloud部署服务中，支持部署内容和主机组参数化。

HUAWEI

- 参考答案：AD

目录

6. 制品和包管理
7. 自动化部署
8. 发布管理
9. 自动化交付流水线

HUAWEI

发布的定义和目标

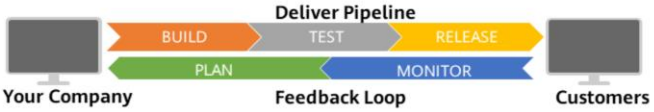
- **软件发布**：将开发好的应用程序正式公布出来的过程
- ITIL中定义的发布管理目标包含：
  - 规划、协调软件和硬件的实施（或安排实施）
  - 针对分发和安装对系统实施的变更而设计和实施有效的程序
  - 确保与变更相关的硬件和软件是可追溯和安全的（正确的、经过批准和测试的版本才能被安装）
  - 在新发布的规划和试运行期间与用户进行沟通并考虑他们的期望
  - 保障发布软件的原始拷贝被安全地存放在最终软件库，配置信息存储在配置管理数据库

HUAWEI

- 发布的目标是提供软件、仓库、软件发布、发布包下载、发布包元数据管理等功能，通过安全可靠的软件仓库实现软件包版本管理，提升发布质量和效率，实现产品的持续发布

部署与发布的关系

- 将部署和发布解耦，实现低风险发布
  - 部署：在特定的环境上安装指定版本的软件。部署可能与某个特性的发布有关，也可能无关。
  - 发布：把一个/组特性提供给（所有或部分）客户。



- **基于环境的发布模式**
  - 关注两个或更多环境，但只有一个环境正在接受实时客户流量
  - 新代码部署到非生产环境中
  - 该版本将流量切换至此环境
  - 只需要很少或者不需要更改应用程序
- **基于应用的发布模式**
  - 通过Feature Toggle来启用特定应用程序功能的发布和展示

HUAWEI

- 代码和环境架构要满足：特性发布不需要变更应用的代码
- 如果混淆部署和发布，就很难界定谁对结果负责
- 解耦部署和发布，可以提升开发人员和运维人员快速且频繁部署的能力，同时产品负责人承担发布成功的责任
- 按需部署，让特性发布成为业务和市场决策，而不是技术决策
- 基于环境的发布模式：
  - 关注两个或更多环境，但只有一个环境正在接受实时客户流量
  - 新代码部署到非生产环境中
  - 该版本将流量切换至此环境
  - 只需要很少或者不需要更改应用程序
  - 包括：Blue-green 蓝绿部署、Canary 金丝雀发布、Cluster immune systems 集群免疫系统
- 基于应用的发布模式：
  - 通过Feature Toggle来启用特定应用程序功能的发布和展示
    - 启用范围
    - 开发团队
    - 内部员工
    - 1%的客户
    - 整个客户群
  - 这种模式的例子也包括
    - Dark Launching

发布策略总览

- 基于应用的发布模式更安全
  - 轻松回滚
  - 缓解性能压力，优雅降级
  - 采用面向服务架构提高恢复能力，服务解耦
  - 将代码部署真正与特性发布解耦
  - 实现假设驱动开发和A/B测试
  - 实现Dark Launch黑启动（案例：Facebook聊天功能，Baidu关键字联想功能）

策略	零停机	生产流量测试	针对特定目标用户	机器资源成本	回退间隔	对用户的负面影响	搭建复杂度
蛮力	×	×	×	低	慢	高	低
金丝雀发布（单服务器组）	×	√	×	低	慢	中	低
滚动发布（单服务器组）	√	√	×	低	慢	小	中
蓝绿发布（双服务器组）	√	×	×	高	快	中	中
金丝雀发布（双服务器组）	√	√	×	高	快	小	中
滚动发布（双服务器组）	√	√	×	高	快	小	中
功能开关（Feature Flag）	√	×	×	低	快	中	中
A/B测试	√	√	√	中	快	小	中
影子测试	√	√	×	高	无	无	高

HUAWEI

- 基于应用的发布模式，主要是通过特性开关来启用特定应用程序功能的发布和展示，具体的发布策略，如表中所示
- 基于应用发布模式的缺点：
  - 有代码侵入
  - 特性开关也是技术债务，需要定期清理
  - 同时增加测试的复杂性，需要打开所有特性开关（也许需要排列组合），还需要测试特性开关功能本身是否正常

单服务器组发布

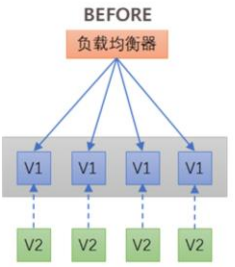
- 传统数据中心中，机器资源比较紧张，应用机器基本是预先静态分配好的（一般由运维负责分配），原来应用 A运行在这 n 台机器上，那么下次升级发布的应用 A 也运行在这 n 台机器上，所以称为单服务器组发布方式。

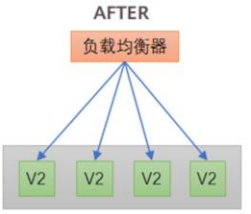
HUAWEI

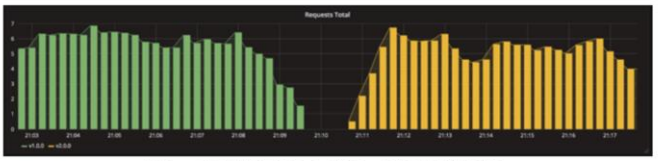
- 包含蛮力发布、金丝雀发布、滚动发布。

蛮力发布 (单服务器组)

- 蛮力发布：即完全靠手工完成软件的升级与发布，在发布过程中会进行人为的中断，进行应用升级。
- **优势：**
  - 简单成本低
- **劣势：**
  - 服务中断用户受影响，出了问题回退也慢
- **适用场合：**
  - 开发测试环境
  - 非关键应用（用户影响面小）
  - 一般适用于初创公司，在业务闲时进行







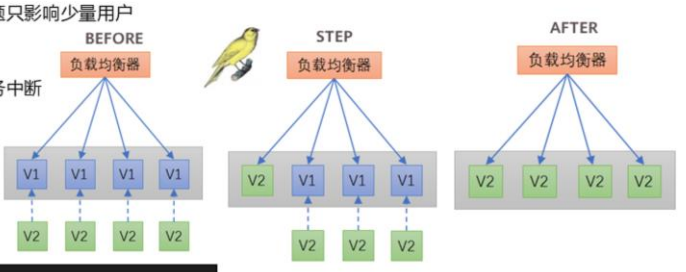
绿色：老版本的请求流量　黄色：新版本的请求流量

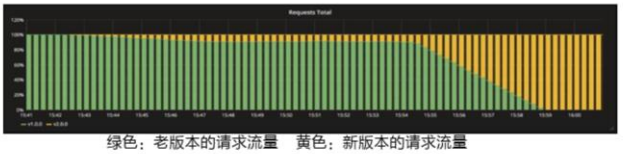
HUAWEI

- 全靠手工完成软件的升级与发布，流程简单，成本低廉。
- 具体流程：先将老版本 V1 全部下掉，再将新版本发到机器上去。这种方式会引入服务中断（停机），在开发测试环境是可行的，但对于生产环境发布，会直接影响用户的使用体验，这种方式一般是不建议的。

金丝雀发布 (单服务器组)

- 在金丝雀开始后，先启动一个新版本应用，但是并不直接将流量切过来，而是测试人员对新版本进行线上测试，启动的这个新版本应用，就是我们的金丝雀。
- **优势：**
  - 用户体验影响小，金丝雀发布过程出现问题只影响少量用户
- **劣势：**
  - 发布自动化程度不够，发布期间可引发服务中断
- **适用场合：**
  - 对新版本功能或性能缺乏足够信心
  - 用户体验要求较高的网站业务场景
  - 缺乏足够的自动化发布工具研发能力





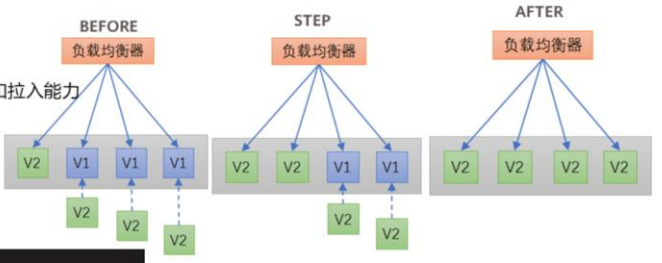
绿色：老版本的请求流量　黄色：新版本的请求流量

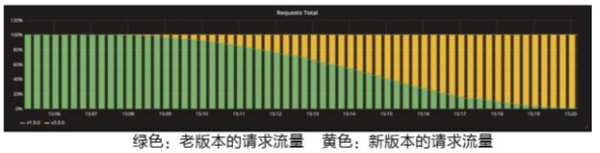
HUAWEI

- 具体流程：金丝雀发布一般先发 1 台，或者一个小比例，例如 2% 的服务器，主要做流量验证用，也称为金丝雀 (Canary) 测试。如果金丝测试通过，则把剩余的 V1 版本全部升级为 V2 版本。如果金丝雀测试失败，则直接回退金丝雀，发布失败。

滚动发布 (单服务器组)

- 滚动发布：一般是取出一个或者多个服务器停止服务，执行更新，并重新将其投入使用。周而复始，直到集群中所有的实例都更新成新版本。
- **优势：**
  - 用户体验影响小，体验较平滑
- **劣势：**
  - 发布和回退时间比较缓慢
  - 发布工具比较复杂，LB 需要平滑的流量摘除和拉入能力
- **适用场合：**
  - 用户体验不能中断的网站业务场景
  - 有一定的复杂发布工具研发能力





绿色：老版本的请求流量　黄色：新版本的请求流量

HUAWEI

- 滚动发布在金丝雀发布基础上的进一步优化改进，是一种自动化程度较高的发布方式，用户体验比较平滑，是目前成熟型技术组织所采用的主流发布方式。
- 滚动式发布一般先发 1 台，或者一个小比例，如 2% 服务器，主要做流量验证。
- 滚动式发布需要比较复杂的发布工具和智能 LB，支持平滑的版本替换和流量拉入拉出：
  - 先将老版本 V1 流量从 LB 上摘除
  - 清除老版本，发新版本 V2
  - 再将 LB 流量接入新版本。这样可以尽量保证用户体验不受影响。
- 用一次滚动式发布一般由若干个发布批次组成，每批的数量一般是可以配置的（可以通过发布模板定义）。例如第一批 1 台（金丝雀），第二批 10%，第三批 50%，第四批 100%。每个批次之间留观察间隔，通过手工验证或监控反馈确保没有问题再发下一批次，所以总体上滚动式发布过程是比较缓慢的 (其中金丝雀的时间一般会比后续批次更长，比如金丝雀 10 分钟，后续间隔 2 分钟)。

双服务器组发布

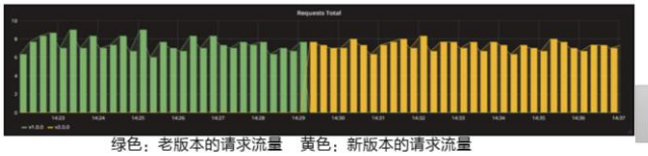
- 随着云计算和虚拟化技术的成熟，特别是容器等轻量级虚拟化技术的引入，计算资源受限和申请缓慢问题已经逐步解决，可以做到弹性按需分配。为一次发布分配两组服务器，一组运行现有的 V1 老版本，一组运行待上线的 V2 新版本，再通过 LB 切换流量方式完成发布，这就是所谓的双服务器组发布方式。

HUAWEI

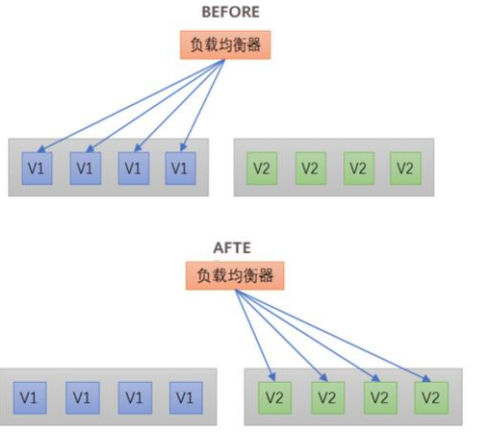
- 具体的发布策略有蓝绿发布、金丝雀发布、滚动发布功能开关、AB测试、影子测试。

蓝绿发布 (双服务器组)

- 蓝绿发布，是一种可以保证系统在**不间断提供服务**的情况下上线的部署方式。蓝绿部署的模型中包含两个集群，部署过程中，并不停止掉老版本，而是直接部署一套新版本，等新版本运行起来后，再将流量切换到新版本上。
- **优势：**
  - 升级切换和回退速度非常快
- **劣势：**
  - 切换是全量的，如果 V2 版本有问题，则对用户体验有直接影响
  - 需要两倍机器资源
- **适用场合：**
  - 对用户体验有一定容忍度的场景
  - 机器资源有富余或者可以按需分配（AWS 云，或自建容器云）
  - 暂不具备复杂滚动发布工具研发能力



绿色：老版本的请求流量　黄色：新版本的请求流量

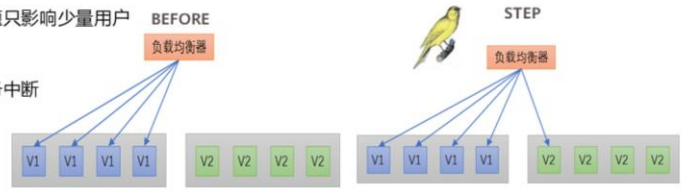


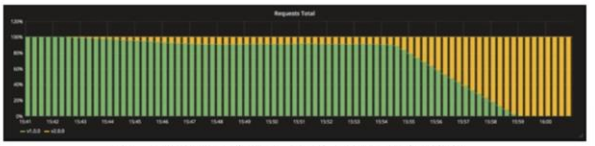
HUAWEI

- 蓝绿发布仅适用于双服务器组发布，可以认为是对蛮力发布的一种简单优化发布方式。V1 版本称为蓝组，V2 版本称为绿组，发布时通过 LB 一次性将流量从蓝组直接切换到绿组，不经过金丝雀和滚动发布，蓝绿发布由此得名。出现问题回退也很直接，通过 LB 直接将流量切回蓝组。发布初步成功后，蓝组机器一般不直接回收，而是留一个待观察期，视具体情况观察期的时间可长可短，观察期过后确认发布无问题，则可以回收蓝组机器。

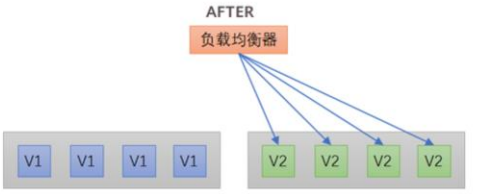
金丝雀发布 (双服务器组)

- 对蓝绿部署的一种简单优化，发布时先从绿组拉入 1 台金丝雀，待金丝雀验证通过再发全量。
- **优势：**
  - 用户体验影响小，金丝雀发布过程出现问题只影响少量用户
- **劣势：**
  - 发布自动化程度不够，发布期间可引发服务中断
- **适用场合：**
  - 对新版本功能或性能缺乏足够信心
  - 用户体验要求较高的网站业务场景
  - 缺乏足够的自动化发布工具研发能力





绿色：老版本的请求流量　黄色：新版本的请求流量

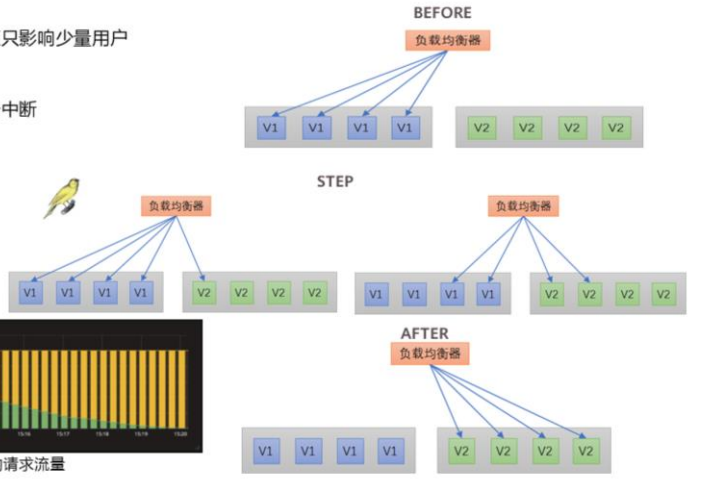


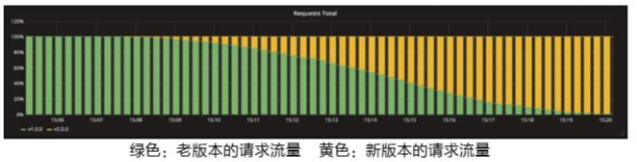
HUAWEI

- 对比蓝绿发布，金丝雀发布方式的优势是有一个生产流量的金丝雀验证过程，可以减轻 V2 可能有问题的风险和影响面。

滚动发布 (双服务器组)

- 滚动式发布是对上面的蓝绿和金丝雀发布的进一步优化，按批次增量滚动发布，提供更平滑的用户体验。
- **优势：**
  - 用户体验影响小，金丝雀发布过程出现问题只影响少量用户
- **劣势：**
  - 发布自动化程度不够，发布期间可引发服务中断
- **适用场合：**
  - 对新版本功能或性能缺乏足够信心
  - 用户体验要求较高的网站业务场景
  - 缺乏足够的自动化发布工具研发能力





绿色：老版本的请求流量　黄色：新版本的请求流量

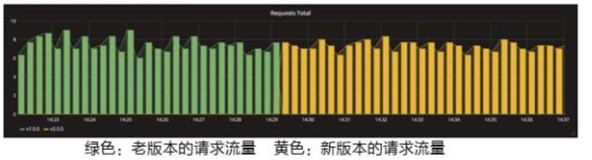
HUAWEI

- 滚动式发布是对上面的蓝绿和金丝雀发布的进一步优化，按批次增量滚动发布，提供更平滑的用户体验。发布前先申请一批新服务器，数量一般和 V1 版本相同，将 V2 版本应用发布到新服务器上。
- 例如如果在 AWS 云上，则可以直接调用 API 申请一批新 VM，如果用容器云 Kubernetes，则可以直接启动一批新容器（使用 V2 版本容器镜像）。
- 一般会先通过 LB 拉入 1 台 V2 版本的机器，这台机器也相当于金丝雀，用于流量验证。
- 逐步按批次完成发布，每批只需要通过 LB 拉入 V2 版本，再拉出对应数量的 V1 版本。
- 批次之间留有观察间隔，通过手工或监控反馈确保没有问题再继续发布。
- 发布有问题回退很快，直接通过 LB 将流量切回 V1 即可。
- 完成发布后，一般 V1 版本要保留观察以备万一，比如留 1 天，1 天后没有问题则回收 V1 机器资源。

功能开关 - Toggle/FeatureFlag/Switch

- 利用代码中的功能开关（Feature Flag/Toggle/Switch）来控制发布逻辑，一般不需要复杂的发布工具和智能 LB 配合，是一种相对比较低成本和简单的发布方式。这种方式也是支持现代 DevOps 理念，研发人员可以灵活定制和自助完成的发布方式。
- **优势：**
  - 升级切换和回退速度非常快
  - 相对于复杂的发布工具，实施比较简单，成本相对低廉
  - 研发能够灵活定制发布逻辑，支持 DevOps 自助发布
- **劣势：**
  - 切换是全量的，如果 V2 版本有问题，则对用户体验有直接影响
  - 对代码有侵入，代码逻辑会变复杂，需要定期清理老版本逻辑，维护成本变高
- **适用场合：**
  - 对用户体验有一定容忍度的场景
  - 已有配置中心或开关中心服务
  - 暂不具备研发复杂发布工具能力





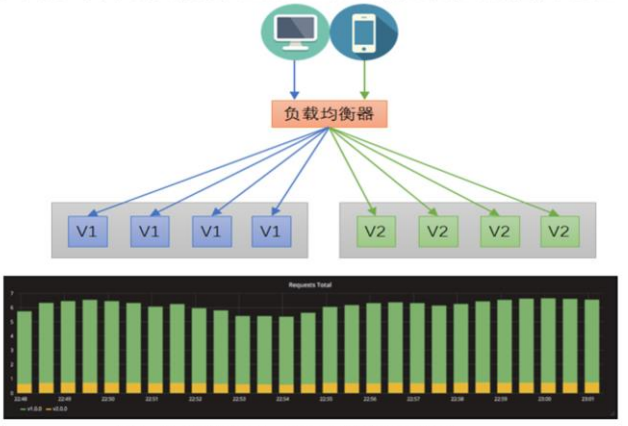
绿色：老版本的请求流量　黄色：新版本的请求流量

HUAWEI

- 需要一个配置中心或者开关中心这样的一个服务支持，通过配置中心运维或研发人员可以在运行期间动态配置功能开关的值。
- 功能开关发布只是配置中心的一种使用场景，配置中心还能支持其他动态配置场景，功能开关服务一般提供客户端的SDK方便开发人员集成，在运行期客户的SDK会同步最新的开关值。
- 技术实现有推方式也有拉方式或者推拉结合方式，新功能和老功能住在同一代码中，新功能隐藏在开关后面，如果开关没有打开，则走老代码逻辑，如果开关打开则走新代码逻辑，技术实践上可以理解为一个简单的if else逻辑。
- 应用上线后开关先不打开，然后运维或研发人员通过开关中心打开新功能，经过流量验证新功能没有问题后，则发布完成，如果有问题，这随时可以通过开关中心切回老功能逻辑。

A/B测试

- A/B测试是用来测试应用功能表现的方法，例如可用性、受欢迎程度、可见性等等。A/B测试目的在于通过科学的实验设计、采样样本代表性、流量分割与小流量测试等方式来获得具有代表性的实验结论，并确信该结论在推广到全部流量可信。
- **优势：**
  - 用户体验影响小
  - 可以使用生产流量测试
  - 可以做到针对某类特定目标用户进行测试
- **劣势：**
  - 搭建复杂度相对高，有一定技术门槛
- **适用场合：**
  - 核心关键业务，比如涉及资金的
  - 具备一定的 A/B 测试平台研发能力



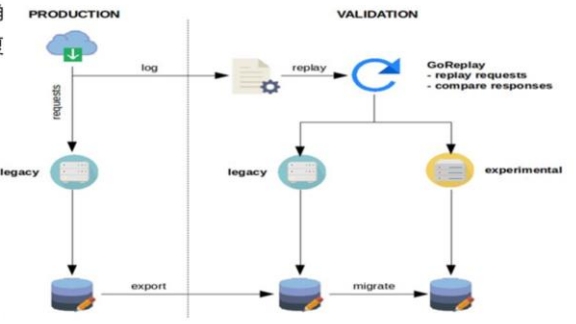
绿色：控制组请求流量　黄色：目标组的请求流量

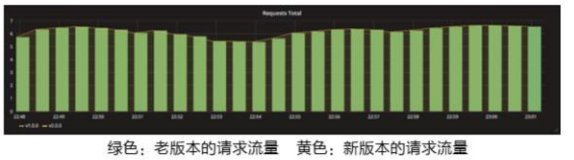
HUAWEI

- 原来的 PC 端和手机端都访问老版本 V1 服务（也称 A 组或控制组），当 V2 新版本（也称 B 组或实验组）发布以后，为了验证 V2 的功能正确性，同时也为了避免 V2 有问题时影响所有用户，先通过 LB 将手机端的流量切换到 V2 版本，经过一段时间的 A/B 比对测试和观察（主要通过用户和监控反馈），确保 V2 正常，则通过 LB 将全部流量切换到 V2。
- 基于 LB 方式实现 A/B 测试，LB 需要能够通过某种条件做流量路由，例如通过 Client IP，设备类型，浏览器类型，甚至是定制的 HTTP Header 或查询字符串。
- 功能开关和AB测试有点相似，但功能开关一般是无状态和全量的，无法做到针对某类特定用户进行测试。而AB测试一般是有状态的，能够根据事物和用户级别的状态，可以实现针对某类特定用户进行测试。

影子测试

- 对于一些涉及核心业务的遗留系统的升级改造，为了确保万无一失，使用影子测试，它采用比较复杂的流量复制、回放和比对技术实现。
- **优势：**
  - 对生产用户体验完全无影响
  - 可以使用生产真实流量进行测试（复制比对）
- **劣势：**
  - 搭建复杂度很高，技术门槛高，数据库的导出复制是难点
  - 外部依赖不能太多，否则测试部署成本很高，且比对测试更加复杂和不稳定
- **适用场合：**
  - 核心关键业务，比如涉及资金的
  - 具备一定影子测试平台研发能力，包括流量复制、数据库导出复制和分发比对系统

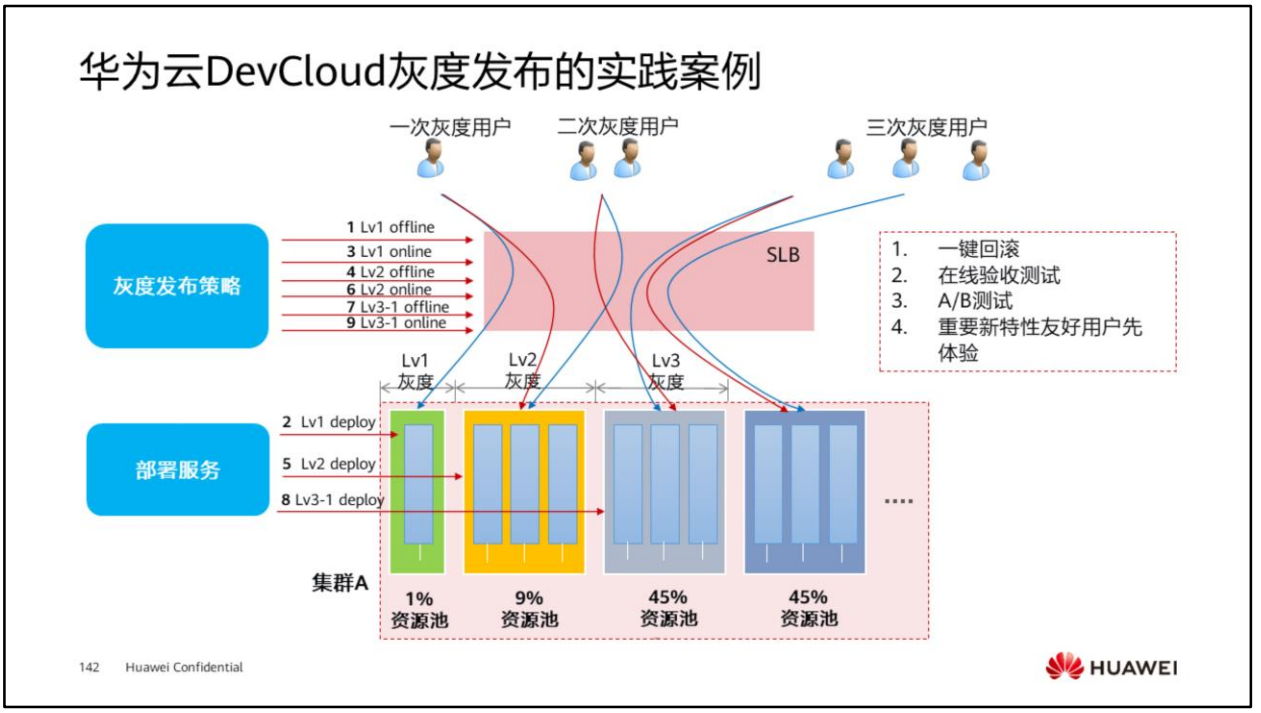




绿色：老版本的请求流量　黄色：新版本的请求流量

HUAWEI

- 目标实现老的 legacy 服务迁移升级到新的 experimental 服务。测试开始前，需要在测试环境部署一份 legacy 服务和 experimental 服务，同时将生产数据库复制两份到测试环境。同时需要将生产请求日志收集起来，将请求分发到 legacy 服务和 experimental 服务，收到响应后进行比对，如果所有响应比对成功，则可以认为 legacy 服务和 experimental 服务在功能逻辑上是等价的；如果有响应比对失败，则认为两者在功能逻辑上不等价，需要修复 experimental 并重新进行影子测试，直到全部比对成功。根据系统复杂度和关键性不同，比对测试时间短的可能需要几周，长的可达半年之久。
- 影子测试因为旁路在独立测试环境中进行，可以对生产流量完全无影响。



- 整个DevCloud的产品中采取的灰度发布服务主要特色有三个:
  - 根据用户画像，精准分层用户，灰度逐步递进，全部保证能够检测到所有的用户分群。
  - 提供了多种灰度策略，如特性开关、AB测试、在线验收测试、友好用户测试等模式。
  - 精准把控灰度批次比例，借助SLB服务，严格按照1%、9%、45%、45%，谨慎比例逐步放大灰度群体。

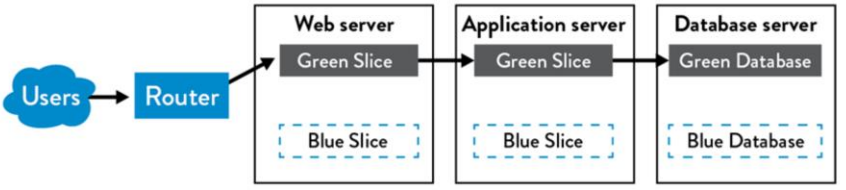
iLearning实践背景

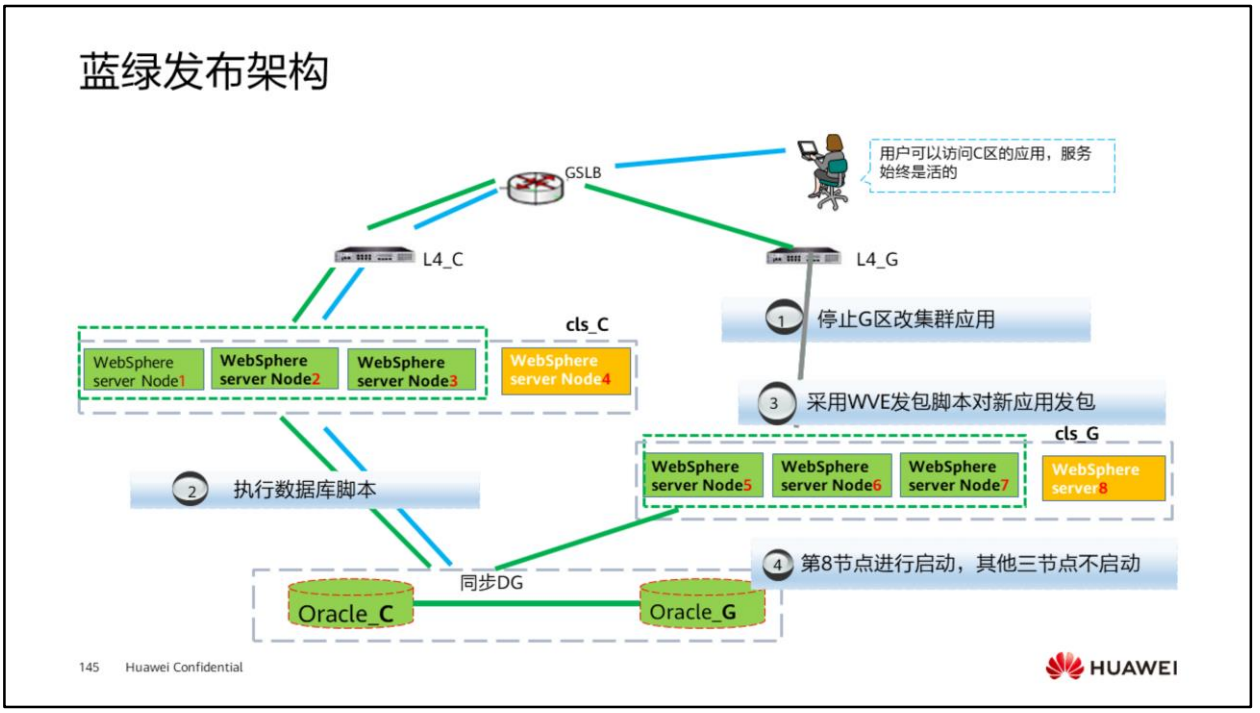
- **问题与挑战：**
  - iLearning在前期的敏捷转型中已经建立了基于Scrum流程的敏捷交付，两周一个迭代进行交付。然而如果每两周都要将产品部署到生产环境交付给用户，团队非常痛苦，因为每两周就需要周末加班熬夜进行部署，并赶在用户上班使用之前验证部署结果。
  - 因此即便团队可以两周交付一些特性，也很难每两周都将其发布给用户，难以进一步缩短交付周期，对业务的响应能力不够快。

HUAWEI

iLearning实践方案

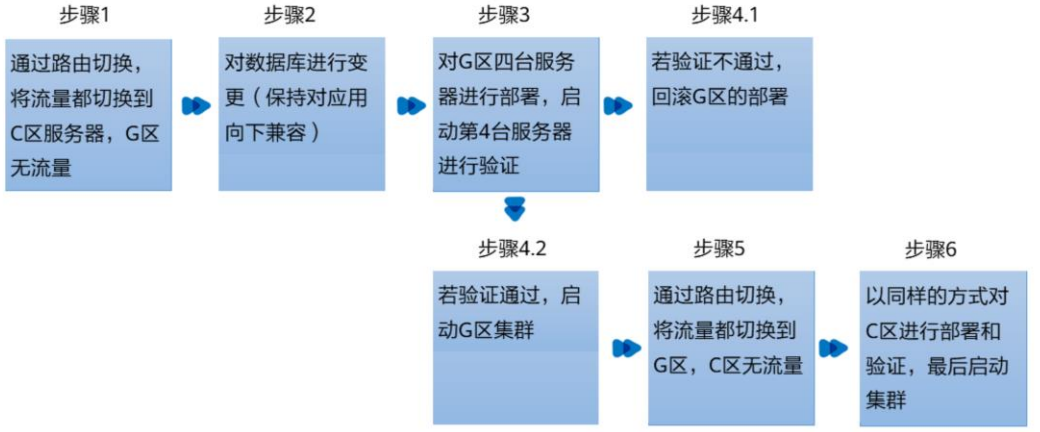
- 要进一步缩短交付周期，就必须做到生产环境部署发布的不停机，既要对用户无感知不影响用户体验，同时也要保证部署是成功和安全的。解决这个问题我们引入了“蓝绿部署”的实践，在华为也称为“双活”。
- 蓝绿部署使用两套一致的环境，一套在线提供服务，一套闲置准备用于下个版本的部署。有时也将两套环境都在线提供服务，可互为容灾，但注意这种模式下在部署过程中系统的容量会有波动，应避免在系统负载峰值时进行。iLearning采用的是后一种方式。





- 通过路由切换，将流量都切换到C区服务器、G区无流量。
- 对数据库进行变更，保持对应用的向下兼容。
- 对G区的4台服务器进行部署，启动第4台服务器进行验证，若验证不通过回滚G区的部署，如果验证通过启动G区集群。
- 通过路由切换，将流量都切换到G区C区无流量，以同样方式，最后对C区进行部署和验证，最后启动集群。

蓝绿发布实践方案



HUAWEI

思考题

1. 以下关于发布管理的说法，正确的是（）？

   A. 任何情况下，都不建议采用蛮力发布。

   B. 利用代码中的功能开关来控制发布逻辑，即使新版本出现问题，也不会对用户体验造成太大影响。

   C. 发布管理的目的是为了在不影响用户体验的情况下进行版本更新，同时保证版本质量。

   D. 影子测试对生产用户体验完全无影响，所有场合都可适用。

HUAWEI

- 参考答案：C

目录

6. 制品和包管理
7. 自动化部署
8. 发布管理
9. **自动化交付流水线**

HUAWEI

流水线历史



1910年，福特在高地公园建立的新工厂，第一次向世界展示了流水线大规模制造的效率：将制造一台T型车的时间从728分钟缩短至93分钟。



大规模流水线的组装方式将复杂的工作分解为简单重复劳动，将待组装的车辆通过流水线的方式流转，大大提升了生产效率，并且将一天车的售价从700美元降至350美元。



上世纪70年代，得益于计算机技术的应用，汽车工厂开始出现了自动控制的机械手臂，将工人从重复而繁重的体力劳动中解放出来。

HUAWEI

- 流水线一词来自于工业化制造业的一个发明通过它重塑了工业化制造的标准和效率。

流水线驱动DevOps

- 流水线实现DevOps模式下持续开发、持续测试、持续集成、持续部署和持续监控这些活动的编排并自动化执行、结果反馈，实现商业敏捷。

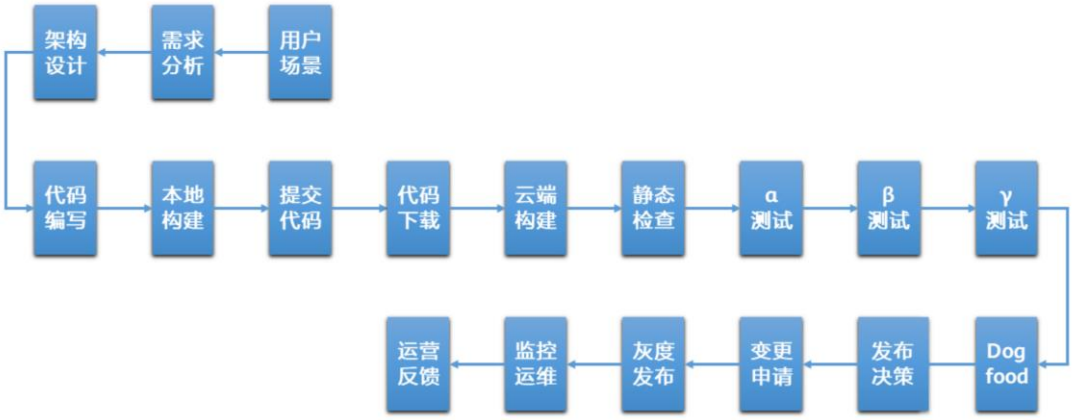
- 为什么需要流水线：
  - CI、CD无缝结合快速迭代，提升交付效率，适应市场需求频繁变化
  - 每个环节都会有反馈，实现完整快速的反馈循环
  - 每个环节都可设置质量门禁，实现质量的层层保障
  - 开发、测试、运维人员协作参与，改善公司组织文化、提高员工的参与感



HUAWEI

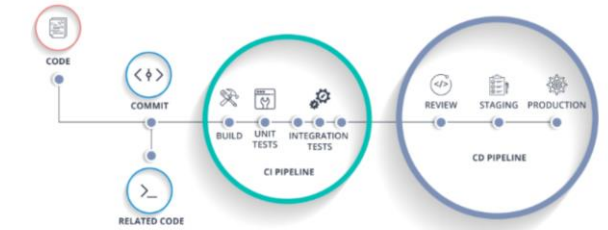
- 一条流水线它就是一个或多个自动化任务或人工任务组合起来的进行交付的一个工具。
- 把一个或多个任务或特定的任务特定的顺序来运行起来达到交付。
- 通过流水线可以让部署成为一个日常的低风险的工作来解决开发的岗位和运维岗位之间的这种根本的长期的冲突。

流水线流程



HUAWEI

流水线助力持续交付高质量应用





分层分级 分层分级流水线，可通过子流水线-stage-job实现多级灵活串并行调度

快速调度 纳管研发过程中编译构建、代码检查、部署等多种自动化任务

质量管控 设置自动化质量门禁，人工卡点，等待时间等控制任务

HUAWEI

- 通过流水线运作，管理者能够看到流线中的一些反馈或者数据来帮助他来进行决策。

流水线的功能实现

工作流编排

- 可按需定制自动化工作流程，配置阶段内任务为串行或并行执行，满足用户一键DevOps持续交付场景

自动触发流水线

- 通过配置流水线触发器，可支持代码提交自动执行流水线

流水线定时执行

- 用户可按周或按日指定流水线定时自动执行，并将结果通知给指定的用户，实现人休息而版本级交付不休息

支持多种任务

- 纳管研发过程中编译构建、代码检查、部署、测试等多种自动化任务，此外还提供延迟执行、人工审核等流水线内置任务

质量管控

- 支持多重质量门禁，任务健康度统计等，呈现价值流中耗时长、错误率高的阻塞点，做质量守护的一道大闸

其它

- 自定义权限配置
- 准确报错，异常提示联动在线帮助平台
- 开放对接jenkins任务

⋮

HUAWEI

关键特性 - 工作流编排

- 流水线由各阶段(Stage)组成、阶段内每一个块是任务原子(Task)
- 阶段支持串/并行、自动/手动等配置
- 阶段分为自定义阶段、发布仓库两种类型
- 支持10+种任务类型，根据业务需求自由组装流水线
- 支持自由拖拽、复制、编辑各阶段任务
- 流水线上任务支持一键跳转编辑

任务类型

- 构建
- 部署
- 代码检查
- 接口测试
- 门禁
- 子流水线
- 延迟执行
- 人工审核
- 扩展类型
- Jenkins task

首阶段展示关联代码仓信息



自定义阶段内自由组合任务



发布仓库阶段内展示软件包信息



HUAWEI

关键特性 - 流水线参数

- 流水线参数传递至各任务，实现产物的价值流动
- 流水线参数支持文本，主机组、枚举、自增长等四种参数类型
- 流水线参数分为静态参数和动态参数。静态参数值总等于预设值，动态参数可被传参修改
- 统一使用${xxx}进行参数化配置

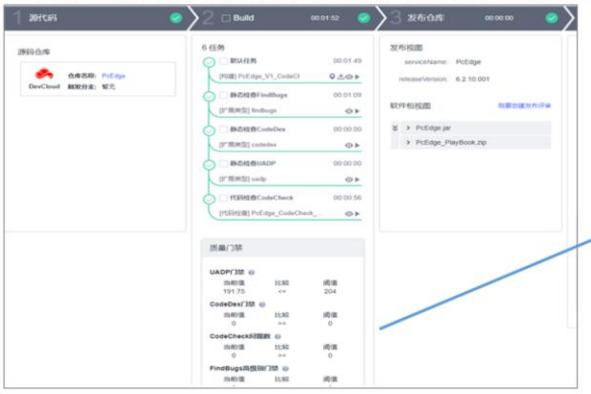
典型使用场景: 通过流水线参数将本次构建产物传递给部署任务



HUAWEI

关键特性 - 自动化质量门禁

- 门禁支持关联代码检查服务，提供专业级的代码检查能力，使缺陷发现前移
- 门禁支持关联接口测试服务，针对用例成功率进行看护
- 任务执行成功统计以健康度形式可视化，便于识别价值流动阻塞点





HUAWEI

关键特性 - 自动触发流水线

- 用户可以选择软开云代码代码托管服务中指定仓库的分支作为触发器，当分支上有push请求时，会自动触发流水线执行
- 两步轻松实现触发器配置
  - 编辑流水线，添加构建任务，流水线自动关联代码仓库
  - 进入触发器界面，选择指定分支，保存流水线配置生效



HUAWEI

关键特性 - 定时执行流水线

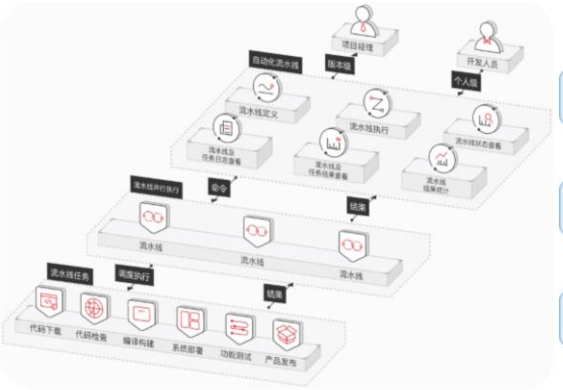
- 计划功能提供流水线定时执行能力，支持每日定时执行、每周定时执行等多种模式
  - 每日定时模式允许选择每日多个时间点执行流水线
  - 每周定时执行模式可以选择具体日期及时间
- 两步轻松实现流水线定时配置
  - 点击流水线计划按钮
  - 进入定时执行配置界面，按需选择时间，保存功能即可生效



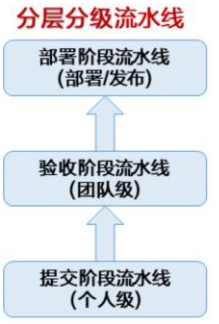


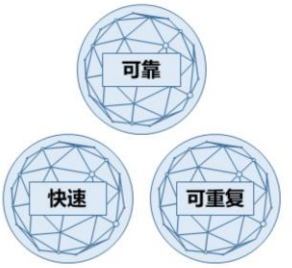
构建灵活的CI/CD流水线实践

- **持续集成实践**
  - 流水线配置代码检查服务，每天定时执行，保障代码质量
  - 用户提交代码后，自动触发流水线执行编译构建、代码检查等任务



- **持续交付实践**
  - 配置构建、代码检查、部署、测试等多种自动化任务，完成一键式出包发布上线生产全流程
  - 流水线执行失败时，发送邮件及时知会到项目经理

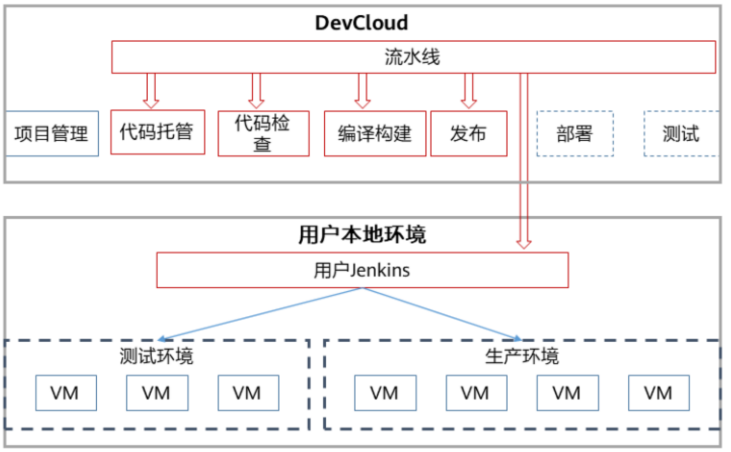




HUAWEI

实践案例 - 通过Jenkins构建线上线下混合流水线

- 用户使用场景：
  - 用户选择在云上开发，不一定会用DevCloud的所有服务
  - Jenkins做为业界最知名的开源持续交付工具，如果用户已自行建设好，业务上云后仍然想保留此系统
- 示例：
  - 用户使用DevCloud开发，本地有测试和生产环境，通过Jenkins已经定制了一套部署发布流程。测试生产环境不愿意上云。



HUAWEI

思考题

1. 以下哪些是DevCloud流水线的特点（）？

   A. 流水线当中不能加入子流水线。

   B. 可设置自动触发流水线、定时执行流水线。

   C. 流水线执行过程中可以交由人工审核，审核通过后继续执行该流水线。

   D. 流水线执行过程中，如果未通过质量门禁，则会终止执行该流水线。

HUAWEI

- 参考答案：BCD

本章总结

- 持续交付与持续部署的概念与意义
- 容器化应用在持续交付中的作用
- 自动化的实现“Everything As Code”
- 自动化编译构建与构建服务介绍
- 制品和包管理介绍
- 自动化部署与部署服务介绍
- 发布管理与发布策略
- 自动化交付流水线介绍

HUAWEI

更多信息

- 华为云官网：
  - https://www.huaweicloud.com/
- 华为云软件开发平台DevCloud：
  - https://www.huaweicloud.com/devcloud/
- 华为云帮助中心：
  - https://support.huaweicloud.com/
- 华为云学院：
  - https://edu.huaweicloud.com/

HUAWEI

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and organization for a fully connected, intelligent world.

Copyright©2020 Huawei Technologies Co., Ltd.
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.



持续运维与监控



HUAWEI

前言

- 随着容器技术的普及，越来越多的企业通过微服务框架开发应用，微服务架构的流行，给应用开发带来了灵活性、扩展性、伸缩性以及高可用性等优势，业务实现更多使用云上服务，运维也转向云上的运维服务，对于云上应用的运维也提出了新的挑战。

HUAWEI

目标

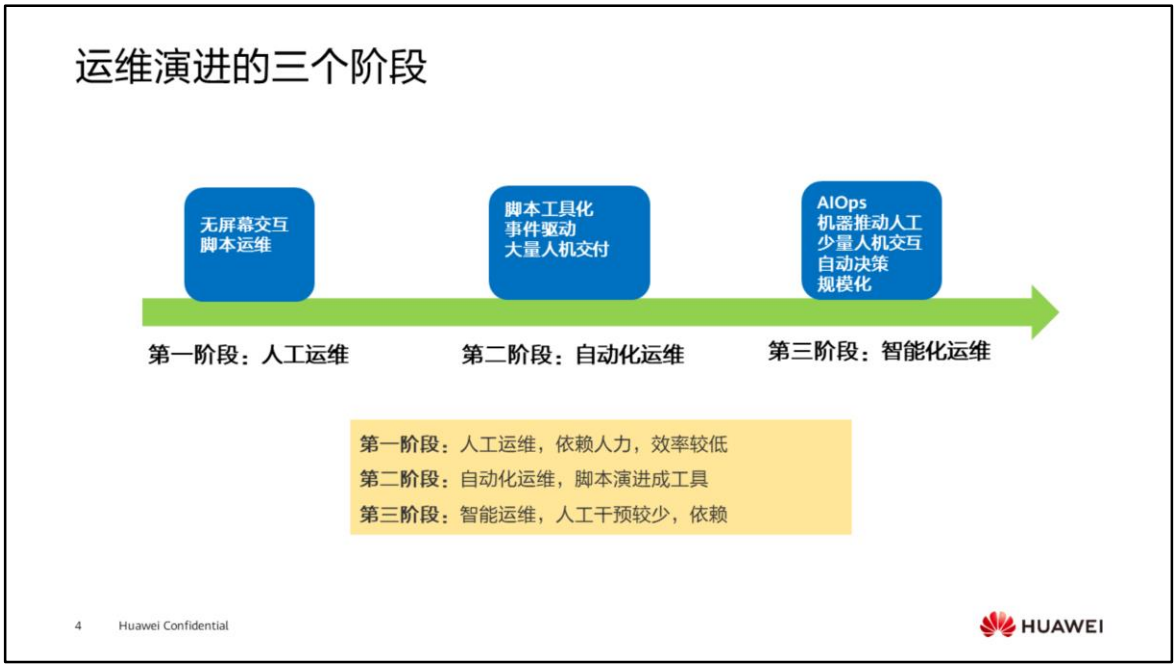
- 学完本课程后，您将能够了解：
  - 传统运维和云上运维的区别，云上运维的目标与挑战
  - 基础监控平台概述及华为云云上运维解决方案
  - 云上服务日志管理与审计
  - 云上应用性能监控与优化
  - 云上微服务治理概述及实现
  - 应用服务网格的概念及应用场景

HUAWEI

目录

1. **运维的演进与DevOps**
2. 运维监控平台概述与云上监控
3. 云上服务日志管理与审计
4. 应用性能管理
5. 微服务治理概述
6. 应用服务网络

HUAWEI



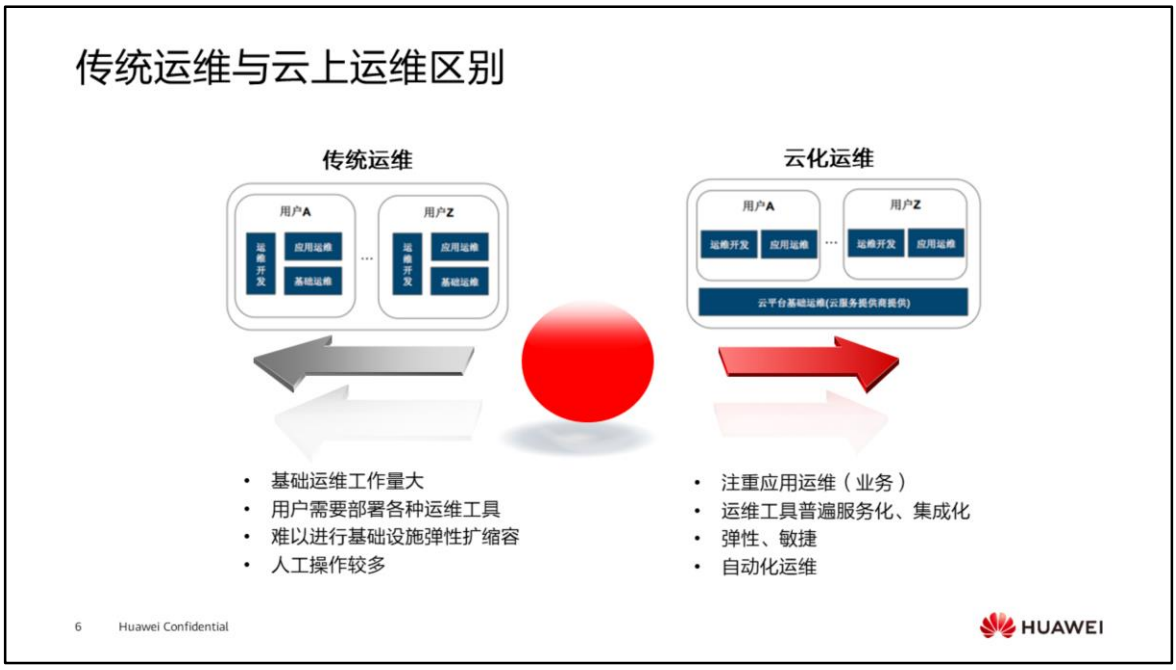
- 第一阶段：黑屏，人工运维。人工的运维依赖，人力、效率比较低下。
- 第二阶段：白屏，自动化运维。自动化运维最大的特点就是把脚本变成工具，然后push到机器上进行自动化的执行。这样可以减少人工干预，来实现自助的运维。
- 第三阶段：智能化运维。AIOps是一个趋势，它可以通过少量的人机交互，通过自驱动的方式来实现智能化的运维。最重要的是人机交互变少了，无屏虽不可能但是个趋势，少量的人机交互，它有自决策、自驱动。

讨论 (5min)



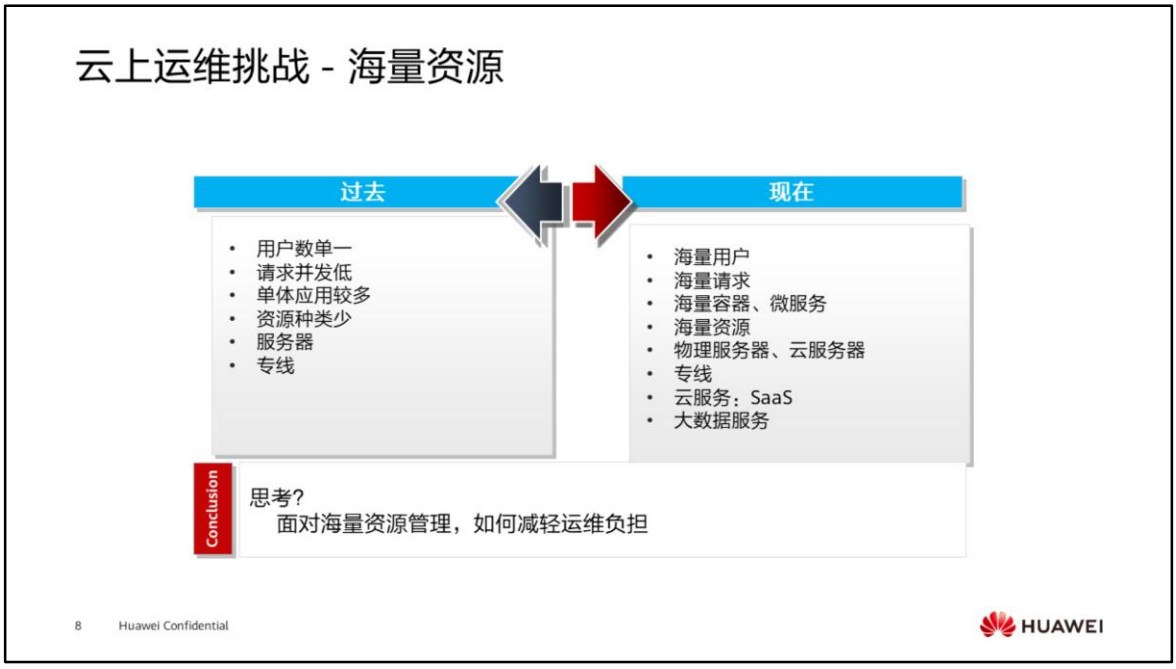
云上运维和传统运维有什么区别?

HUAWEI

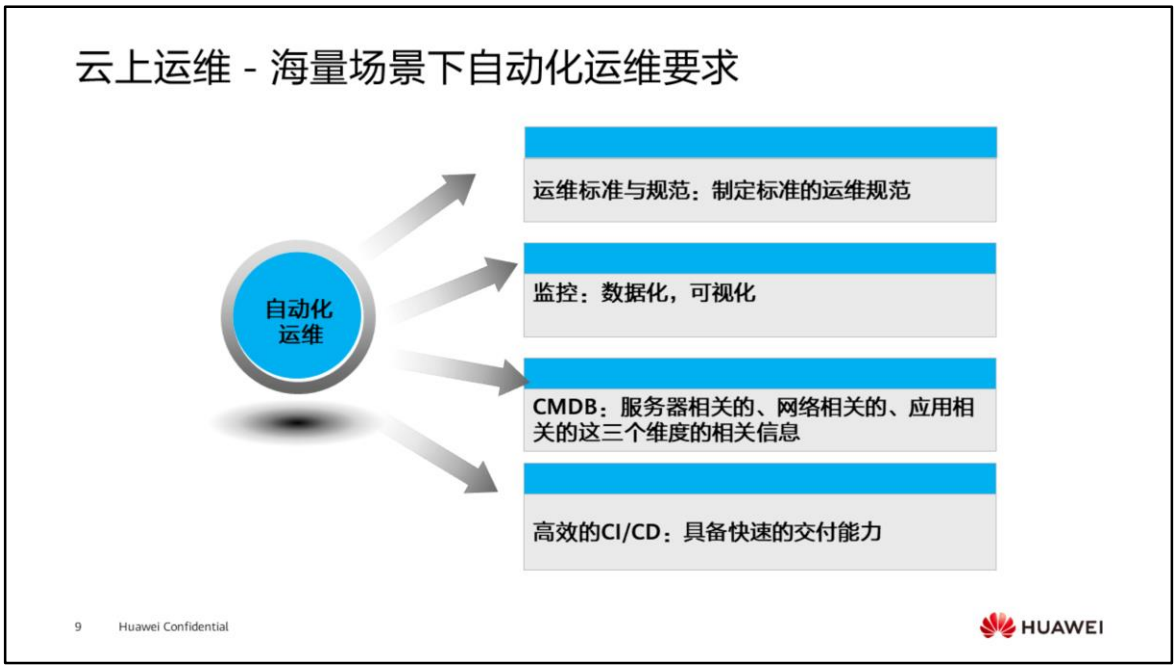


- 传统运维与云上运维区别：
- 基础运维工作量大
  - 一般来说，很多企业的运维部门主要工作包括基础运维（针对企业IT基础设施的运维）、应用运维（针对企业具体业务的运维）。较大的运维部门可能还有单独的运维开发，负责为公司运维部门开发运维工具和平台。当用户决定上云（尤其是IaaS公有云），就表示用户已经把基础运维以及相关的工具平台开发工作交付给云供应商，而把应用运维作为整个运维部门的核心。
- 公共组件普遍服务化
  - 在企业内部，不同业务IT系统都会用到很多公共组件（如数据库，消息队列等）。由于团队的技术背景不同，业务要求不同，经常会出现这些公共组件选型的不一致性，导致上线后的运维负担非常重。所以，企业内部运维部门的一个重要工作就是把这些公共组件标准化并最终服务化，能够对具体业务部门完全透明。这样即可以降低运维部门自身的运维成本，又可以提高业务部门的开发效率。

- 弹性、敏捷
    - 在云计算平台中，无论是存储服务，计算服务还是网络服务都会提供弹性伸缩，按需付费的功能。运维团队需要把云计算供应商提供的弹性能力充分暴露给业务研发团队，并鼓励业务研发团队为弹性基础设施做设计（如支持状态无关和水平扩展），甚至参与到业务架构中的设计，充分使用如华为云弹性伸缩/AOM容器弹性伸缩等一系列弹性服务。这样一方面可以大幅度降低运维成为（如业务扩容、缩容都能够自动完成），另外也可以满足基础设施成本的弹性需求，降低整个业务的运营成本，提升在市场中的竞争力。
- 自动化运维
    - 无论是传统的ITIL体系还是ITOM领域，运维管理体系的基础都是CMDB（配置管理数据库）。在云平台上，所有的基础设施服务都会提供API接口，从而变成了可编程的基础设施。这就为CMDB中的基础设施管理完全自动化提供了前提条件。例如，一台云主机启动起来后，运维管理体系能够感知到它的加入，能够按照预先指定的自动化部署脚本对它进行IT系统部署。在部署完成后主动加入系统的负载均衡器，开始服务用户。因为基础设施的可编程，上面整个部署流程完全可以和基础设施实施一体化自动管理。



- 在传统的运维环境下，用户的数量比较单一，请求量比较单一，而且单体应用相对来说比较多，资源的种类也比较少，主要是一些服务器或者一些专项的运维。而现在有海量的用户、海量的请求，海量的容器、海量的虚拟机、海量的微服务模块，同时又有海量的IaaS资源、PaaS资源等。其他的，还有大数据平台，AI平台等。



- 运维标准与规范：制定运维标准，标准会在工具里固化。
- 监控：泛监控，运行时，静态，数据化，可视化泛监控，不是说传统的监控，是把线上想知道的一切都数据化，最终数据不是给人看的，是给机器去消费的，数据是我们的生产资料，不是可视化，那不是我们的目标。
- CMDB
  - CMDB 应该放什么，一般放服务器相关的、网络相关的、应用相关的这三个维度的相关信息。为自动化运维提供数据基础。
- 高效的CI/CD：一定要具备快速的交付能力，主要体现这两个方面：第一，新开发的能力能不能快速上线，第二，想扩容一台机器能不能快速扩出来。这两个能力抽象出来是三块。

自动化运维 - 批量迁移与弹性伸缩

- 目标：应用管理是日常要做的操作，规模化，要快速对一个单元建站、扩容、缩容。

场景1 批量迁移



应用迁移
配置迁移
主机迁移
磁盘迁移

传统：迁移工具
现在：迁移服务，例如，从线下迁移服务至云上，可以使用云迁移服务完成业务的迁移

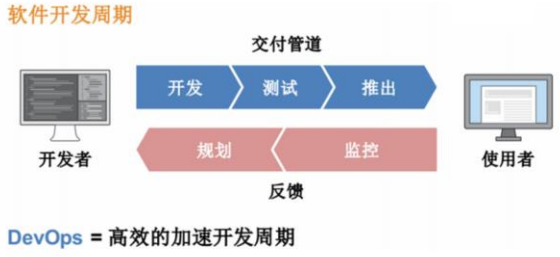
场景2 弹性伸缩

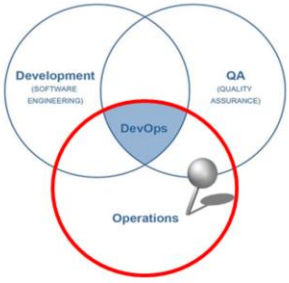
主机弹性伸缩
　　配置触发条件，满足条件后，自动的进行主机维度的扩缩容。

容器弹性伸缩
　　一般以容器的资源使用状态为扩缩容依据，当满足条件后，迅速弹出/收缩容器实例，满足业务需要。

HUAWEI

DevOps的开发与运维



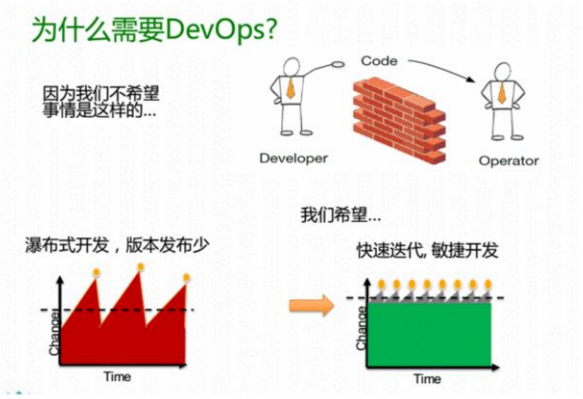


Dev和Ops的双向融合，Ops参与研发流程，提供运维方面的建议和要求；而Dev介入运维活动，更好地重用IT能力，减少不成熟技术的引入。

HUAWEI

- 传统的IT系统流程一般包含Dev和Ops两部分。
- Dev和Ops的双向融合：Ops参与研发流程，提供运维方面的建议和要求；而Dev介入运维活动，更好地重用IT能力，减少不成熟技术的引入。

为什么需要DevOps



打破开发、测试和运维之间的壁垒，让软件开发的最后一公里——运维环节，变得更加敏捷和适应变化，开发和运维职能的紧密协作是DevOps运维的核心思想。

HUAWEI

DevOps下的运维系统特点



HUAWEI

- 高可用：运维系统高可用是基础
- 高成功率：运维监控需要精准的指标，精确的操作，来保证运维动作的搞成功率
- 高效率：运维系统一定要具备很高效率，快速扩容、快速部署。
- 可回滚：运维操作需要具备可回滚的能力
- 自动化运维：减少人机交互，将更多的运维工作交给系统
- 应用管理：以应用为中心，实现应用的持续部署、集成、运维等

DevOps运维目标

- 保障系统的稳定运行，达成服务的质量目标



HUAWEI

- 运维对象的可用性改进包括
  - 延迟优化：不同组件之间网络调用的延时优化
  - 性能优化：运维过程中针对于业务出现的性能问题，需要针对性的提出性能优化方案，优化相关部件
  - 效率优化：提高各部件资源利用率
  - 变更管理：管理业务在不同环境的部署、上线
  - 监控：实时监控系统的运行状态，掌握系统异常
  - 紧急事务处理：定位、处理、管理故障
  - 容量规划与管理：为 应用、中间件、基础设施等提供选型。

云上运维 - AIOPS

- AIOps，即 Artificial Intelligence for IT Operations，智能运维，将人工智能应用于运维领域，基于已有的运维数据（日志、监控信息、应用信息等），通过机器学习的方式来进一步解决自动化运维没办法解决的问题。



HUAWEI

- AIOps围绕质量保障、成本管理和效率提升的基本运维场景，逐步构建智能化运维场景。
  - 在质量保障方面，细分为异常检测、故障诊断、故障预测、故障自愈等基本场景；
  - 在成本管理方面，细分为指标监控，异常检测，资源优化，容量规划，性能优化等基本场景；
  - 在效率方面，分为智能变更、聊天机器人等基本场景。
- 目标：利用大数据、机器学习和其他分析技术，通过预防预测、个性化和动态分析，直接和间接增强IT业务的相关技术能力，实现所维护产品或服务的更高质量、合理成本及高效支撑。

讨论 (5min)

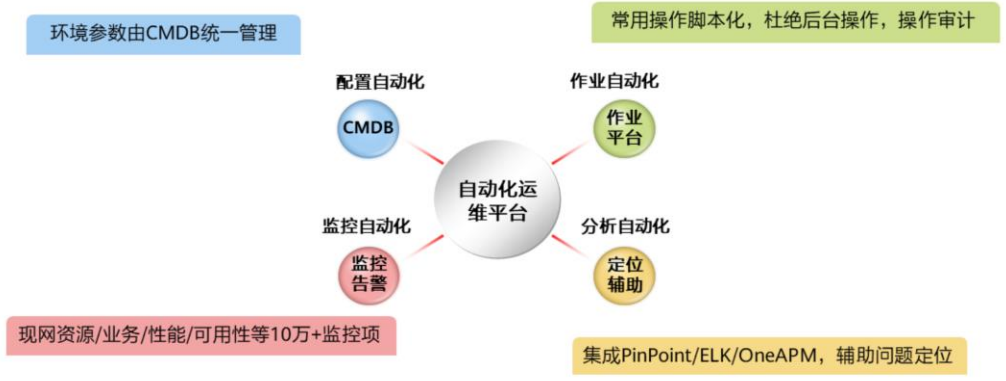


你理想中的AIOps应该具备哪些能力?

HUAWEI

云上运维实践

- 监控自动化、配置自动化、作业自动化、日志分析自动化。



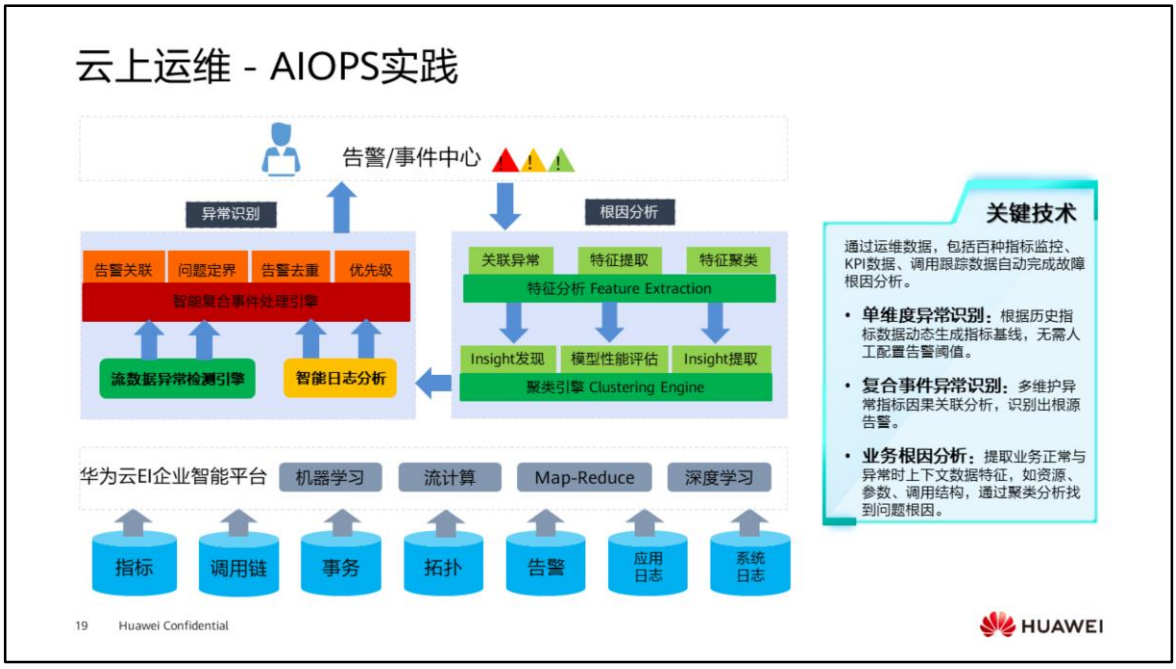
HUAWEI

- 60%以上的生产事故源于人为操作不当，所有的运维团队都极力避免对生产环境过多人为操作

华为云立体化运维解决方案



HUAWEI



- 通过运维数据，包括百种指标监控、KPI数据、调用跟踪数据自动完成故障根因分析。
- 单维度异常识别：根据历史指标数据动态生成指标基线，无需人工配置告警阈值。
- 复合事件异常识别：多维护异常指标因果关联分析，识别出根源告警。
- 业务根因分析：提取业务正常与异常时上下文数据特征，如资源、参数、调用结构，通过聚类分析找到问题根因。

目录

1. 运维的演进与DevOps
2. **运维监控平台概述与云上监控**
3. 云上服务日志管理与审计
4. 应用性能管理
5. 微服务治理概述
6. 应用服务网络

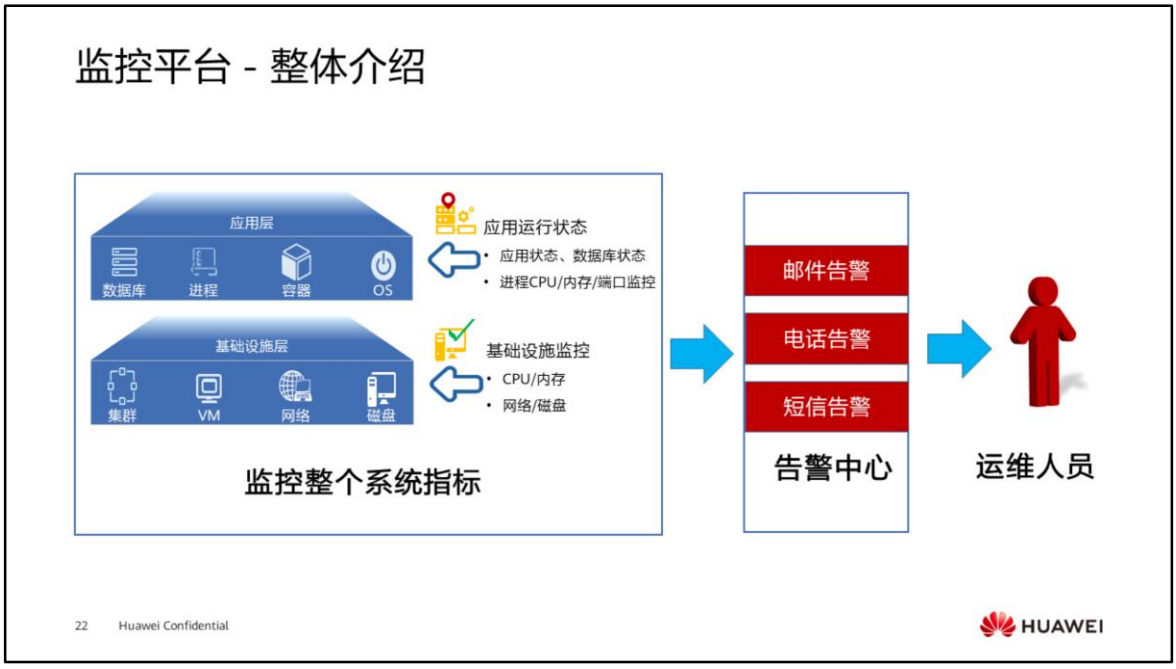
HUAWEI

监控平台 - 监控的目的

- 当产品上了规模后，在线上的运行状况十分良好？是否存在性能瓶颈和潜在的Bug？
- 产品实际运行在产品服务器时占用的内存平均和峰值有多少？是否合理？是否有调优的余地？
- 服务的实际用户流量有多少？相比前段时间是增加了还是减少了？

HUAWEI

- 根据历史监控数据，对为了做出预测
- 发生异常时，即使报警，或做出相应措施
- 根据监控报警及时定位问题根源
- 通过可视化图表展示，便于直观获取信息



- 一个完整的监控平台需要至少包含两部分，监控和告警。
- 使用脚本、工具等方式获取整个系统的不同指标值，涵盖基础设施、应用。
- 获取监控项比如进程的CPU/内存，节点的磁盘使用状态等。
- 同时针对每种指标设置告警，一旦指标发生异常，可以通过多种方式告知运维人员及时修复。

监控平台 - 监控指标

监控指标范围要求：覆盖主机、应用多个维度

主机监控指标

网络：入方向网络流速、数据包个数、错误包个数等
磁盘：磁盘读/写速率、磁盘读写状态、可用磁盘空间、可用磁盘空间
显卡：显存容量、显存使用率、GPU使用率
主机系统：CPU内核总量、CPU使用率、可用虚拟内存、NTP服务器状态等

应用监控指标

进程：CPU内核总量、 CPU内核占用、 CPU使用率、句柄数、物理内存总量、物理内存使用率、进程状态、虚拟内存总量

监控指标周期要求：一般以1分钟为监控周期，重要的指标需要做到秒级监控，例如1s/5s一个采样点

HUAWEI

讨论 (5min)



使用过哪些监控工具，具备什么样的特点?

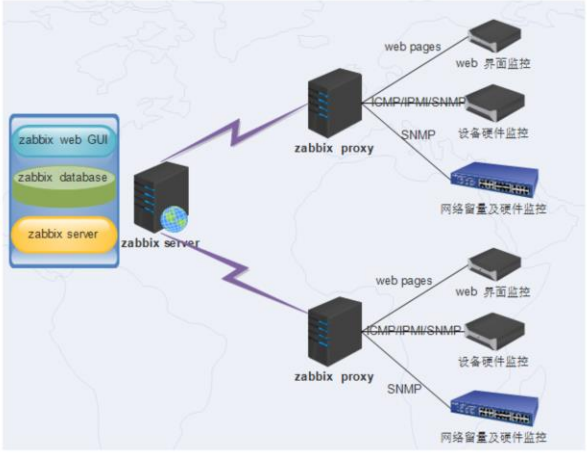
HUAWEI

传统运维工具



- 运维人员技能要求高，配置烦杂，同时需要维护多套系统。
- 无法关联分析，虽然指标很多，但需要根据运维经验逐一排查。
- 对于pinpoint这类分布式追踪系统，学习和使用成本高，并且稳定性较差。
- Kibana：Kibana 是一个免费且开放的用户界面，能够让您对 Elasticsearch 数据进行可视化，并让您在 Elastic Stack 中进行导航。详情参考：https://www.elastic.co/cn/kibana
- logstash：Logstash 是免费且开放的服务器端数据处理管道，能够从多个来源采集数据，转换数据，然后将数据发送到您最喜欢的“存储库”中。详情参考：https://www.elastic.co/cn/logstash
- elastic：开源的搜索解决方案。详情参考：https://www.elastic.co/cn/
- Zabbix：适用于任何IT基础架构、服务、应用程序和资源的监控解决方案。详情参考：https://www.zabbix.com/
- Grafana：是一个跨平台的开源的度量分析和可视化工具,可以通过将采集的数据查询然后可视化的展示,并及时通知。详情参考：https://grafana.com/
- Prometheus：是一个开源监控告警解决方案。详情参考：https://prometheus.io/
- Zipkin：是一个开放源代码分布式的跟踪系统。详情参考：https://zipkin.io/
- pinpoint：是用 Java 编写的 APM（应用性能管理）工具，用于大规模分布式系统。详情参考：https://pinpoint.com/

主机监控 - Zabbix



- **Zabbix是一款常用的监控工具，可以提供监控数据采集**
- **Zabbix指标采集流程：**
  - 采集器agentd需要安装到被监控的主机上，负责定期收集各项数据，并发送到zabbix server端，zabbix server将数据存储到数据库中，zabbix web根据数据在前端进行展现和绘图

HUAWEI

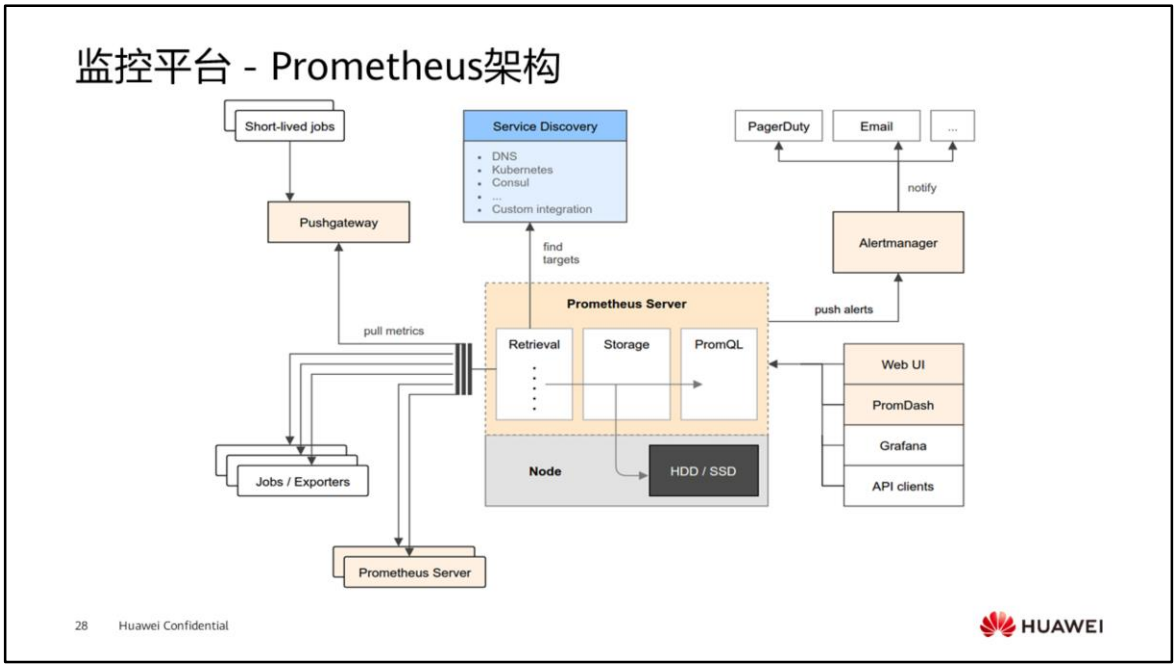
监控平台 - Prometheus

- Prometheus是一套开源的监控&报警&时间序列数据库的组合,起始是由SoundCloud公司开发的。成立于2012年，之后许多公司和组织接受和采用prometheus，他们便将它独立成开源项目，并且有公司来运作，该项目有非常活跃的社区和开发人员，目前是独立的开源项目，任何公司都可以使用它。



HUAWEI

- 2016年，Prometheus加入了云计算基金会，成为kubernetes之后的第二个托管项目.google SRE的书内也曾提到跟他们BorgMon监控系统相似的实现是Prometheus。
- 现在最常见的Kubernetes容器管理系统中，通常会搭配Prometheus进行监控。
- Prometheus是一套开源的监控、报警、时间序列数据库的组合，最初由SoundCloud公司开发。Prometheus 在2016加入 CNCF（Cloud Native Computing Foundation），成为继Kubernetes之后的第二个由基金会主持的项目。目前社区非常活跃，有高达1.4万个Star。Prometheus非常适合监控Kubernetes。
- 与一般的监控系统相比，Prometheus具有如下特点：
  - 时序数据库存储监控数据能够存储更大量的数据，不依赖于其他存储系统，安装非常简单。
  - 灵活的查询语言（PromQl）。
  - 通过基于HTTP的pull方式采集时序数据，可以通过Pushgateway进行时序列数据推送。
  - 多种可视化和仪表盘支持。



- Prometheus Server主要由三部分组成。Retrieval：负责定时去指定的API上抓取采样指标数据。Storage：负责将采样数据高效安全地持久化存储到磁盘中。PromQL：负责提供查询。
- Prometheus Server端会存储所有数据，并对这些数据进行分析，基于规则进行报警。
- Prometheus 利用自身的时序数据库，有着非常高效的存储。
- Alertmanager是独立于Prometheus之外的一个组件，提供十分灵活的报警方式。另外，可以通过Web UI、Grafana、API clients等可视化界面查询数据。
- Prometheus支持通过配置文件、文本文件、ZooKeeper、Consul、DNS SRV lookup等方式指定抓取目标。Prometheus 是通过主动获取的方式采集数据，但是客户端有两种方式输送数据，一种方式是业务服务提供HTTP接口，Prometheus定时从业务服务中获取数据；另外一种方式是业务服务推送数据到Pushgateway,Prometheus定时从Pushgateway中获取数据。

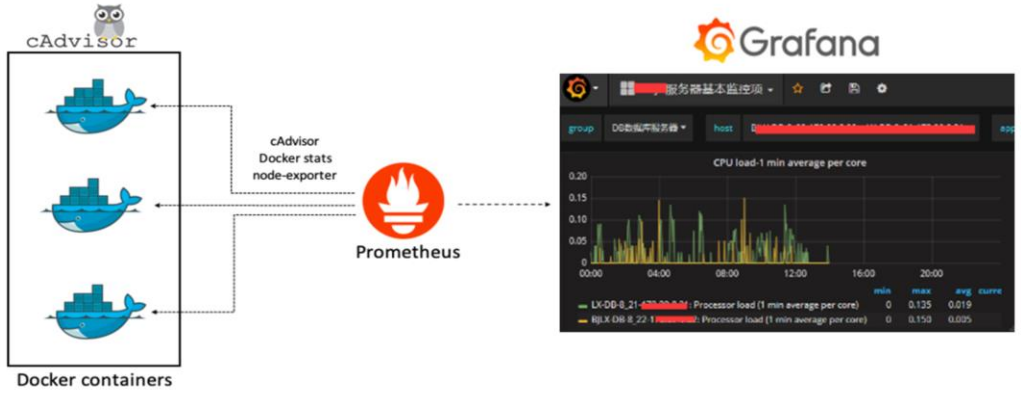
监控平台 - Prometheus+Grafana



HUAWEI

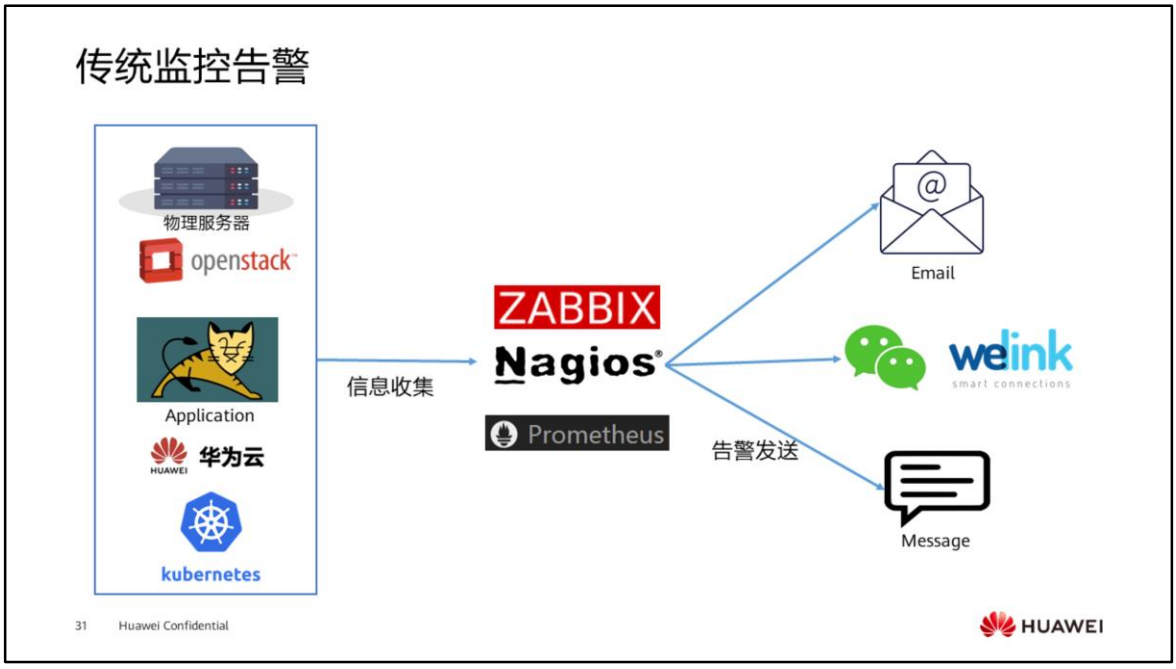
- prometheus是由谷歌研发的一款开源的监控软件，目前已经被云计算本地基金会托管，是继k8s托管的第二个项目。prometheus可以很好的监控容器（docker）相关指标。
- 主流用法：
  - prometheus作为监控指标处理中心，对接Grafana进行指标展示。
  - Prometheus 和 Grafana 是一对完美的组合，把它们集成起来非常简单。

容器监控



HUAWEI

- 图片出处：3ms

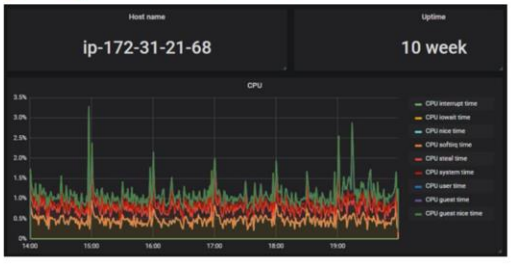

传统监控告警
物理服务器
openstack
Application
华为云
kubernetes
信息收集
ZABBIX
Nagios
Prometheus
告警发送
Email
welink
smart connections
Message
31
Huawei Confidential
HUAWEI

- 图片出处：3ms

监控视图







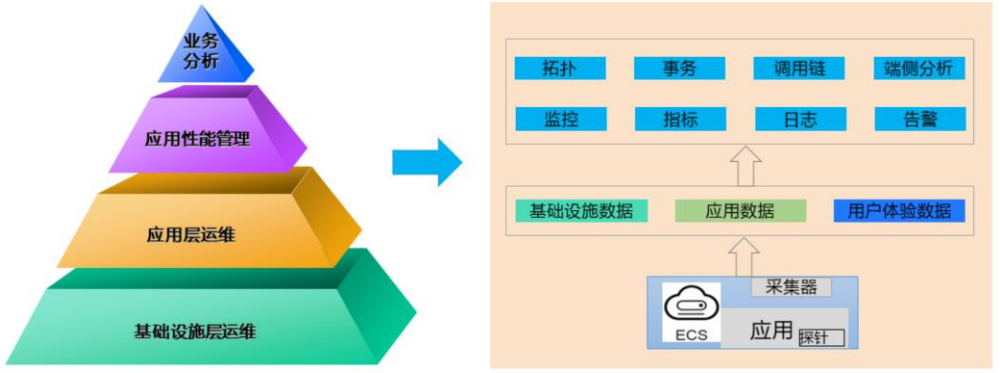


HUAWEI

- 多种监控面板，难以整合及管理

云上运维实践 - 华为云云上运维解决方案

- 实现资源、应用、应用用户体验全面运维，不管是IT经理、运维人员还是开发人员，都能一站式获取所需，完成云上运维。



HUAWEI

- 华为云云上运维解决方案可以完成基础设施层、应用层、应用性能层这三层的一站式运维。其工作原理就是在云服务器上安装采集器和探针。采集器主要用来采集指标、日志、事件等运行、运维数据，探针主要是安装在应用中，用来采集应用的调用链、用户体验等数据。采集器和探针会将基础设施数据、应用数据、用户体验数据采集并分析处理，之后通过监控指标、日志、拓扑、事务、调用链等展现出来，实现资源、云上业务、用户体验全面运维，不管是IT经理、运维工程师还是开发人员，都能够一站式获取所需，完成云上运维。
- 从图上可以看到，基础设施层、应用层、性能管理是不同的三层，采集器和探针也是分开的。这样也是为了满足不同规模的云上业务的运维诉求。将基础设施层、应用层运维看做是基础运维能力的话，性能管理层运维则是高级运维。安装采集器即可获取到基础运维能力，通过指标、监控、日志、告警等完成基础运维。在应用中安装探针即可获取高级运维能力，通过拓扑、事物、调用链等完成应用的高级监控、性能分析、用户体验等运维。

华为云AOM运维管理平台



HUAWEI

- 应用运维管理（简称AOM）是云上应用的一站式立体化运维管理平台，实时监控应用及云资源，采集各项指标、日志及事件等数据分析应用健康状态，提供告警及数据可视化功能，帮助用户及时发现故障，全面掌握应用、资源及业务的实时运行状况。

AOM监控对象

主机包括弹性云服务器（ECS）、裸金属服务器（BMS）和GPU加速云服务器（GACS）。



应用是您根据业务需要，对相同或者相近业务的一组服务进行逻辑划分。
应用的类型
包括系统应用和自定义应用。

容器监控的对象仅为通过CCE部署的工作负载、通过ServiceStage创建应用。

当您购买了弹性负载均衡（ELB）、虚拟私有云（VPC）、关系型数据库（RDS）、分布式缓存服务（DCS）后，无需额外安装其他插件，即可在AOM界面监控这些云服务的运行状态及各种指标。

HUAWEI

AOM - ECS监控



HUAWEI



- 按照集群维度进行主机监控，获取集群、主机、服务关联关系。

AOM告警管理

- 告警是指AOM自身，或AOS、ServiceStage、CCE、APM等外部服务在异常情况或在可能导致异常情况下上报的信息，并且您需采取相应措施清除故障，否则会由于AOM自身或外部服务的功能异常而引起业务的异常。



HUAWEI

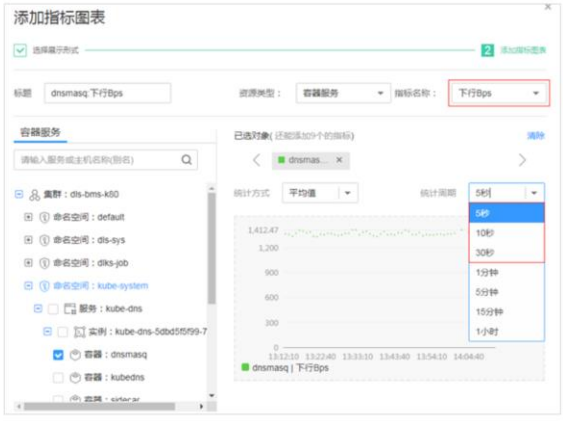
- 通过静态阈值规则可对资源的指标设置阈值条件，当指标数据满足阈值条件时，会产生阈值告警（阈值告警即由阈值规则触发而产生的告警）；当没有指标数据上报时，会产生数据不足事件（数据不足事件即由阈值规则触发而产生的事件）
- AOM已对接消息通知服务（Simple Message Notification，简称SMN），通过在SMN界面设置通知策略，当静态阈值规则的状态（正常、超限阈值、数据不足）发生变化时，会以邮件或短信等方式通知，以便您在第一时间发现异常并进行处理。



- 告警中心：支持各类指标告警。
- 要使用告警，需要先设置告警条件，即告警阈值规则。AOM支持一条一条或批量设置告警规则。告警规则可以是各类指标，主机、网络、磁盘、文件系统、服务、进程等的指标，如状态、CPU、内存、磁盘可用空间、磁盘空间容量、磁盘读写速率等；应用的吞吐量、成功率、时延、错误调用数等指标。
- 设置好条件之后，会产生告警，呈现在告警列表中。告警列表支持按照时间、名称、告警源、清除状态、级别过滤。
- 告警通知：告警中心对接了消息通知服务，支持通过邮件、短信、http等方式将告警信息通知到用户。
- 告警订阅：如果需要将告警转发到第三方系统中，可以使用告警订阅功能将数据转发到分布式消息服务的kafka队列中去，然后第三方系统从kakfa中消费数据。
- 查看、清除告警。

AOM - 自定义仪表盘

- 支持多种图标形式（曲线图、服务状态、资源TOP）
- 仪表盘支持全屏监控和全屏轮播功能，帮助您全面、深入地掌握监控数据。



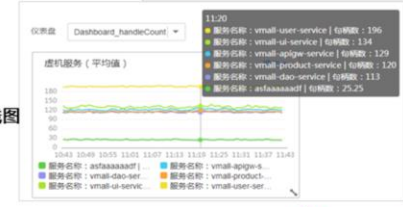
资源TopN图表



服务状态图表



曲线图



HUAWEI

目录

1. 运维的演进与DevOps
2. 运维监控平台概述与云上监控
3. **云上服务日志管理与审计**
4. 应用性能管理
5. 微服务治理概述
6. 应用服务网络

HUAWEI

日志的概念



- 日志：网络设备、系统及服务程序等，在运作时都会产生一个叫log的事件记录；每一行日志都记载着日期、时间、使用者及动作等相关操作的描述。

HUAWEI

- 图片出处：3ms

日志的使用场景



运维

需要日志管理系统统一记录设备、系统的运行情况，用于监控服务器状态、定位排查故障、分析性能瓶颈等。
要求：日志采集/导出便捷，索引功能快速、全面、易用

审计

需要记录服务器、防火墙等设备的日志，供安全审计系统使用，用于监控系统安全、进行可信取证等。
要求：数据完整、安全传输、加密保存等。





运营

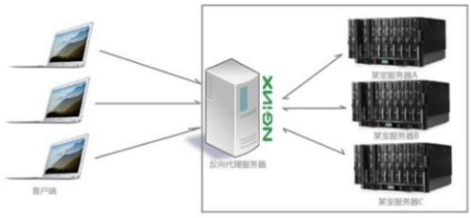
需要提供业务信息，例如页面的PV、UV、停留时间等，用于了解业务运营状况、分析用户行为特征等。
要求：日志服务提供多样化的可视化报表展示功能。

HUAWEI

- 图片出处：3ms

网络日志分析 (Nginx)

- Nginx 是一款自由的、开源的、高性能的 HTTP 服务器和反向代理服务器，为WEB服务器提供负载均衡，应对海量客户端并发访问的压力；



HUAWEI

- Nginx日志主要分为两种：access_log(访问日志)和error_log(错误日志)。
- 错误日志：记录了访问出错的信息，可以帮助我们定位错误的原因。
- 访问日志：我们可以得到用户的IP地址、浏览器的信息，请求的处理时间等信息。

Nginx访问日志格式介绍

参数	说明	示例
$remote_addr	客户端地址	219.227.111.255
$remote_user	客户端用户名称	—
$time_local	访问时间和时区	18/Jul/2014:17:00:01 +0800
$request	请求的URI和HTTP协议	“GET /article-10000.html HTTP/1.1”
$http_host	请求地址，即浏览器中你输入的地址（IP或域名）	www.ha97.com 198.98.120.87
$status	HTTP请求状态	200
$upstream_status	upstream状态	200
$body_bytes_sent	发送给客户端文件内容大小	1547
$http_referer	url跳转来源	https://www.google.com/
$http_user_agent	用户终端浏览器等信息	“Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; SV1; GTB7.0; .NET4.0C;
$ssl_protocol	SSL协议版本	TLSv1
$ssl_cipher	交换数据中的算法	RC4-SHA
$upstream_addr	后台upstream的地址，即真正提供服务的主机地址	10.36.10.80:80
$request_time	整个请求的总时间	0.165
$upstream_response_time	请求过程中，upstream响应时间	0.002

HUAWEI

Linux系统日志介绍

- 常用系统日志清单

日志类型	日志路径	内容
整体系统信息	/var/log/message	公共日志文件，记录Linux内核消息及各种应用程序的公共日志信息，包括启动、IO错误、网络错误、程序故障等
启动日志	/var/log/boot.log	一般包含系统启动时的日志，包括自启动的服务
登录失败日志	/var/log/btmp	记录所有失败登录信息。非文本文件，可以使用last -f /var/log/btmp进行查看。
Cron计划日志	/var/log/cron	cron计划任务的日志，每当cron任务被执行的时候都会在这个文件里面记录
安全日志	/var/log/secure	包含验证和授权方面信息。例如，sshd会将所有信息记录（其中包括失败登录）在这里。

HUAWEI

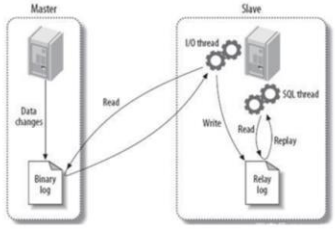
- 通常定位应用问题时，如果发现应用运行日志没有发现异常，就会排查操作系统日志，查看问题发生时是否环境有问题。

MySQL日志（binlog）

- 概念

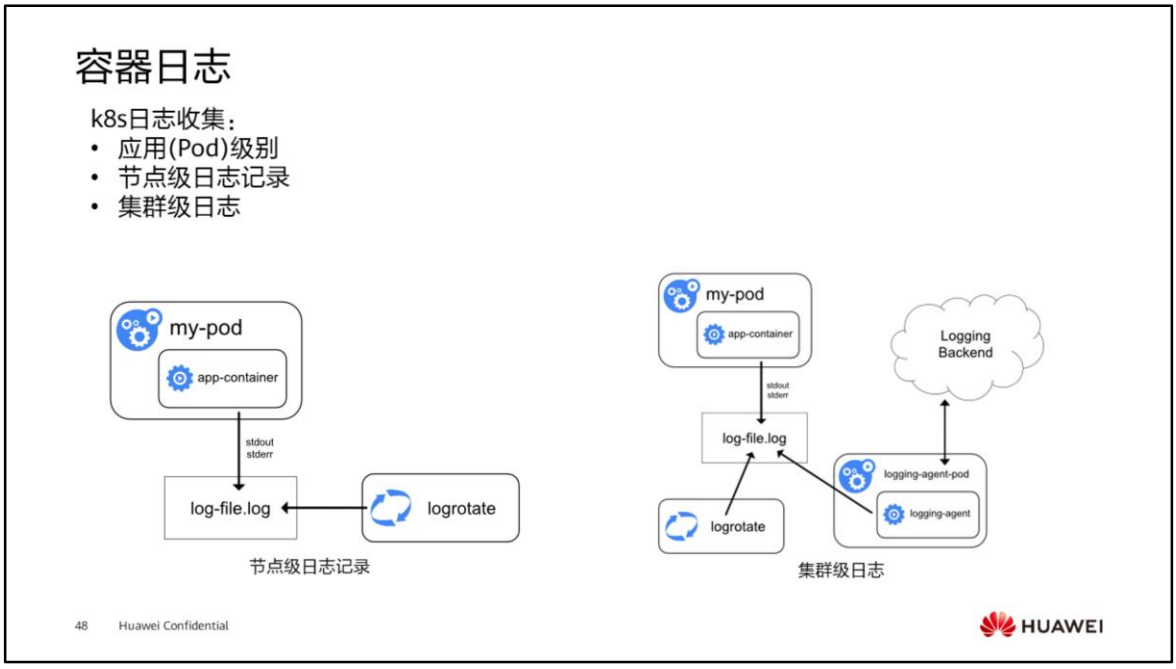
binlog是一个二进制格式的文件，用于记录用户对数据库更新的SQL语句信息，例如更改数据库表和更改内容的SQL语句都会记录到binlog里，但是对库表等内容的查询不会记录。默认情况下，binlog日志是二进制格式的，不能使用查看文本工具的命令（比如，cat，vi等）查看，而使用mysqlbinlog解析查看。

- 作用
  - 恢复：用户误删除的数据，可能根据历史数据对数据库表进行恢复；
  - 复制：主从数据同步，实现高可靠；
  - 审计：追踪数据库历史操作记录，用于安全分析等；



主从复制通过binlog进行同步

HUAWEI



- Docker日志：
  - 容器引擎日志一般都交给了系统日志，不同的操作系统会放在不同的位置。
  - 容器日志可以理解是运行在容器内部的应用输出的日志，默认情况下，docker logs 显示当前运行的容器的日志信息，内容包含 STOUT(标准输出) 和 STDERR(标准错误输出)。
- Kubernetes日志：
  - 应用(Pod)级别，Pod级别的日志 , 默认是输出到标准输出和标志输入，实际上跟docker 容器的一致。使用 kubectl logs pod-name -n namespace 查看
  - 节点级别，容器化应用写入 stdout 和 stderr 的任何数据，都会被容器引擎捕获并被重定向到某个位置。 例如，Docker 容器引擎将这两个输出流重定向到某个日志驱动 ， 该日志驱动在 Kubernetes 中配置为以 json 格式写入文件。
  - 集群级别，集群级别的日志收集 , 有三种：
    - 使用在每个节点上运行的节点级日志记录代理。
    - 在应用程序的 pod 中，包含专门记录日志的 sidecar 容器。
    - 将日志直接从应用程序中推送到日志记录后端。
  - 参考：https://kubernetes.io/docs/concepts/cluster-administration/logging/

业务运行日志

- 运行日志内容包括：运行时间+模块名+线程号+级别+类名/方法名+执行结果（返回值）
- 运行日志主要作用：问题分析和业务分析（了解业务的执行情况）

```
11:41:57.666233.T25 MRB INFO: iMapTask::iMapSynRecvBySn synrecv success!
11:41:57.666925.T25 MRB INFO: RESPONSENBC,VER=10,SN=1,LEN=26,FLG=0x10,ERR=0
CMD=95202,SIP=00000000,RIP=7f000001,SPID=51,RPID=51,SPH=11321,RPH=11321,SM=101,RM=1,SCH=141,RCH=876,SDID=0,RDID=0

11:41:57.667067.T2c MRB INFO: iMapTask::iMapSynRecvBySn synrecv success!
11:41:57.667184.T24 MRB INFO: RESPONSENBC,VER=10,SN=1,LEN=58,FLG=0x10,ERR=0
CMD=95202,SIP=00000000,RIP=7f000001,SPID=51,RPID=51,SPH=11321,RPH=11321,SM=101,RM=1,SCH=140,RCH=875,SDID=0,RDID=0

11:41:57.667280.T2f MRB INFO: iMapTask::iMapSynRecvBySn synrecv success!
11:41 时间 线程ID 模块名 级别 nvoke cmd=95202 r 类名：方法名        执行结果
11:41                                                              Failed!
11:41:57.667405.T2f TOPOADP ERROR iMapTopoAdpImpl::OnBatchDeleteLink Failed! rtn=10003
11:41:57.667477.T2f TOPOADP ERROR: DeleteLink(LinkID:23:4865:3158299:3158300) Failed!
11:41:57.667581.T2f TOPOADP INFO: iMapTopoAdpImpl::OnBatchDeleteLink leave.
11:41:57.667607.T2f TOPOADP INFO: HANDLE TOPO_CMD_REMOVE_LINK_BY_EMS leave.
11:41:57.667948.Ta MRB INFO: SendFrame,RPID=1089,CMD=15
11:41:57.669577.T2c TOPOADP INFO: iMapTopoAdpImpl::OnBatchDeleteLink leave.
11:41:57.669637.T2c TOPOADP INFO: HANDLE TOPO_CMD_REMOVE_LINK_BY_EMS leave.
11:41:57.669932.Ta MRB INFO: SendFrame,RPID=1089,CMD=15
11:41:57.681745.T27 TOPOADP INFO: iMapTopoAdpEmHandler::pushEvent enter.
11:41:57.681786.T27 TOPOADP INFO: iMapTopoAdpEmHandler::pushEvent SourceClass:EAMServer, EventType:EAMEvent!
11:41:57.681819.T27 TOPOADP INFO: iMapTopoAdpEmHandler::EAMEventProc enter.
```


HUAWEI

分享（5min）



您认为日志输出要遵循哪些原则?

HUAWEI

业务日志输出规范

- 为了方便开发人员在海量日志中快速准确的找到对应的业务日志，除了对日志输出的内容要求清晰之外，还要控制日志的数量，一方面节省本地磁盘的开销，另一方面避免日志太多导致历史日志被老化，因此要求开发人员遵循统一的《日志输出规范》。

1、日志内容使用英文描述，语义要完整准确，涉及到度量信息需要标识相应的度量单位，不允许出现不明含义的魔鬼数字
2、在所有出错或业务失败的地方，必须记录日志，日志中包括错误描述；
3、密码等敏感信息不能以明文形式记录在日志文件中；
4、在INFO级别日志中，不能出现error、fail、fault这些敏感词汇；
5、执行成功的调用和过程禁止打印WARN及以上级别的日志；
6、禁止循环中打印日志；
7、禁止短时间内频繁重复输出相同信息的日志点；
8、单服务实例在非Debug级别下产生的运行日志，最大应小于30KB/min；

日志级别重要性：一方面可以快速过滤ERROR或WARNING的日志，另一方面为了保存更久的日志可以通过动态调整日志输出的级别，如停止输出INFO级别的日志。

HUAWEI

开源日志系统ELK三剑客

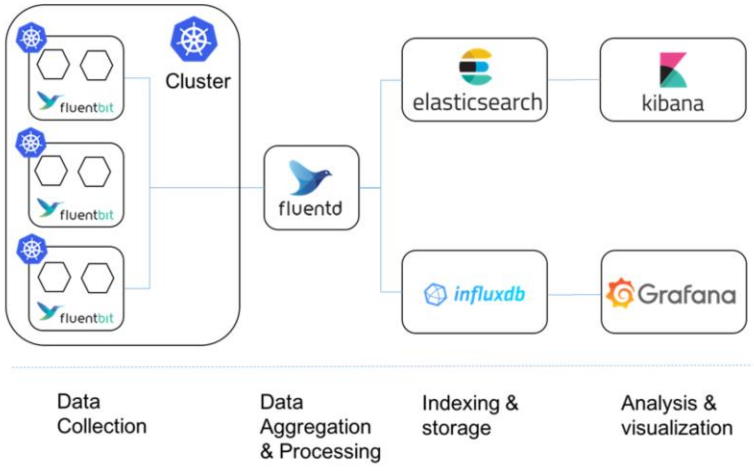


Elasticsearch 是一个基于Lucene搜索引擎的NoSQL数据库。
Logstash 是一个日志管道工具，接受数据输入，执行数据转换，然后输出数据。
Kibana 是一个界面层，在Elasticsearch之上工作。

HUAWEI

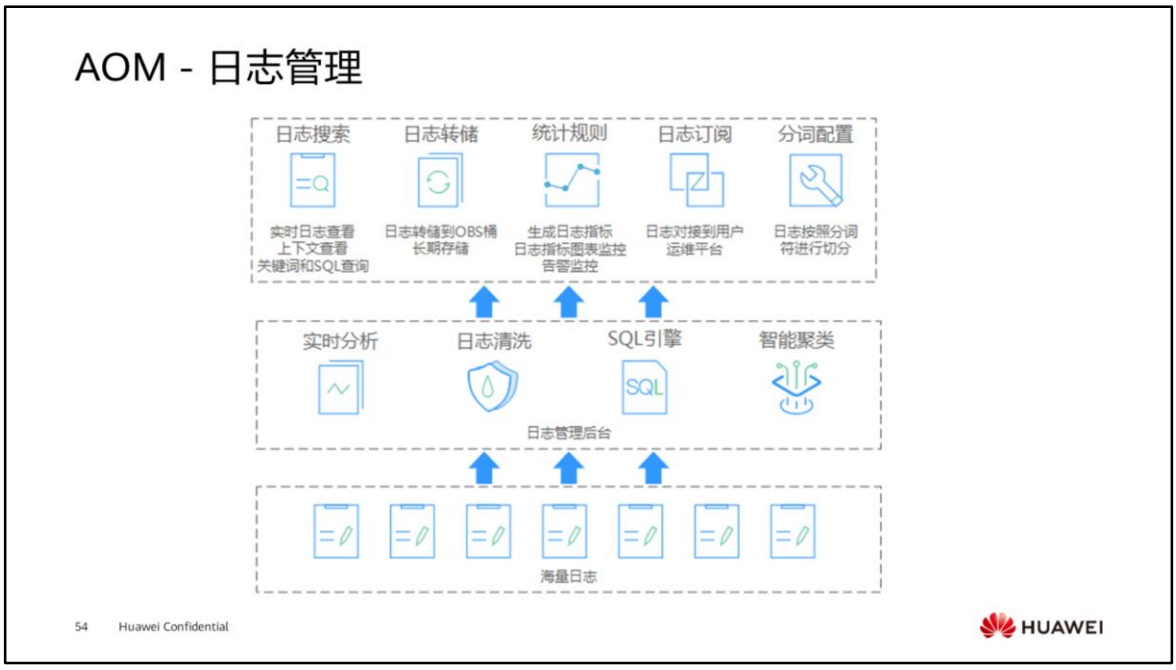
- elastic：开源的搜索解决方案。详情参考：https://www.elastic.co/cn/

容器日志采集



HUAWEI

- Kubernetes：https://kubernetes.io/zh/
- fluentbit：https://fluentbit.io/
- elasticsearch：https://www.elastic.co/cn/
- kibana：https://www.elastic.co/cn/kibana
- influxdb：https://www.influxdata.com/
- Grafana：https://grafana.com/



- 日志管理
  - 日志搜索：日志文件支持开启实时日志查看功能以持续查看最新日志。使用日志检索功能可帮您快速在海量日志中查询到所需的日志，还可结合日志的来源信息和上下文原始数据一起辅助定位问题。同时您可对原始日志进行查询与分析，也可对结构化后的日志进行SQL查询与分析。
  - 日志转储：根据您购买套餐的不同，AOM提供7天或30天的日志存储时长。AOM支持将日志转储到对象存储服务（Object Storage Service，简称OBS）的OBS桶中，以便进行长期存储。如果您有更长时间的日志存储需求，可添加日志转储。
  - 统计规则：日志包含了系统性能及业务等信息，例如，关键词ERROR的多少反应了系统的健康度，关键词BUY的多少反应了业务的成交量等，当您需要了解这些信息时，可创建统计规则。统计规则创建成功后，AOM能够针对您配置的关键词周期性地进行统计，并生成指标数据，以便您实时了解系统性能及业务等信息。同时，还可以针对日志指标添加阈值规则，当满足阈值条件时产生阈值告警，以便您能在第一时间发现异常并进行处理。
  - 日志订阅：当您需要将AOM的日志实时对接到您的运维平台时，可开启日志订阅功能。开启后，AOM通过调用分布式消息服务（DMS）的发送接口发送日志到指定的Kafka队列中，您可从DMS的Kafka队列中消费日志。
  - 分词配置：通过配置分词可将日志内容按照分词符切分为多个单词，在日志搜索时可使用切分后的单词进行搜索。
- 图片来源：华为云

云日志服务LTS

- 日志组

日志组（LogGroup）是云日志服务进行日志管理的基本单位，可以创建日志流以及设置日志存储时间。

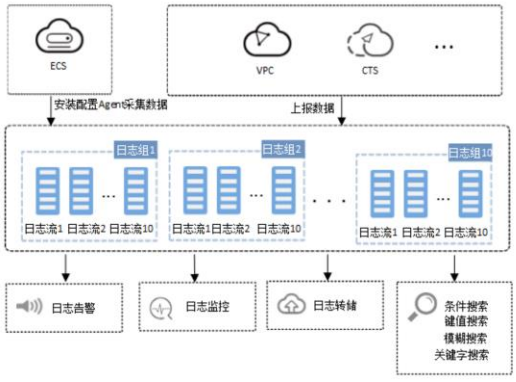
- 日志流

日志流（LogStream）是日志读写的基本单位，日志组中可以创建日志流，方便对日志进一步分类管理。

- 日志采集

采集器（Agent）：支持3种数据类型：

  - 1）在ECS或私有云节点上安装采集器，通过配置日志路径即可完成采集。
  - 2）SDK/API：用户通过接口的方式上报日志数据；
  - 3）云服务：ELB/VPC/WAF/CCE等云服务自动完成数据对接；

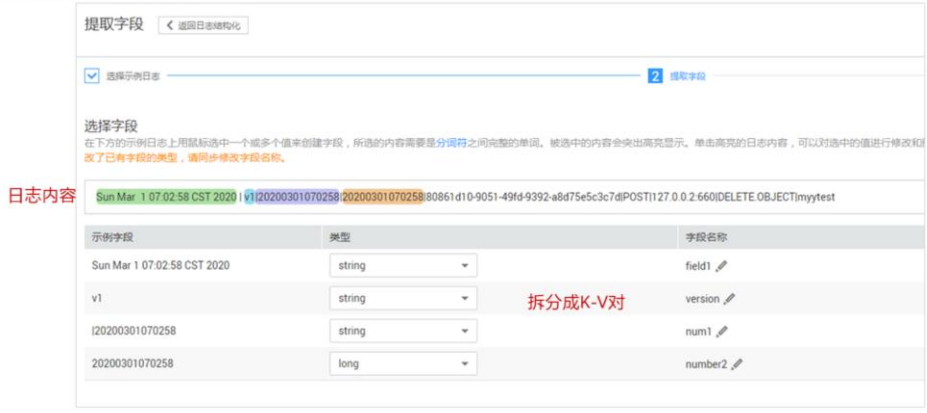


HUAWEI

- 图片来源：华为云

LTS功能之数据结构化

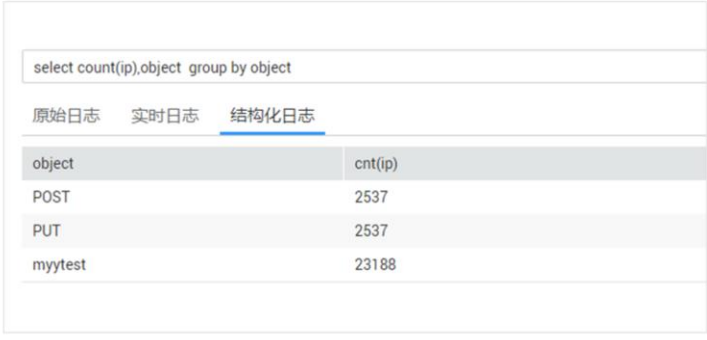
- 结构化：将文本数据通过正则拆分的方式，转换成键值对。结构化后的数据可以像访问数据库一样，通过SQL查询、排序、聚合。



HUAWEI

LTS功能之数据结构化

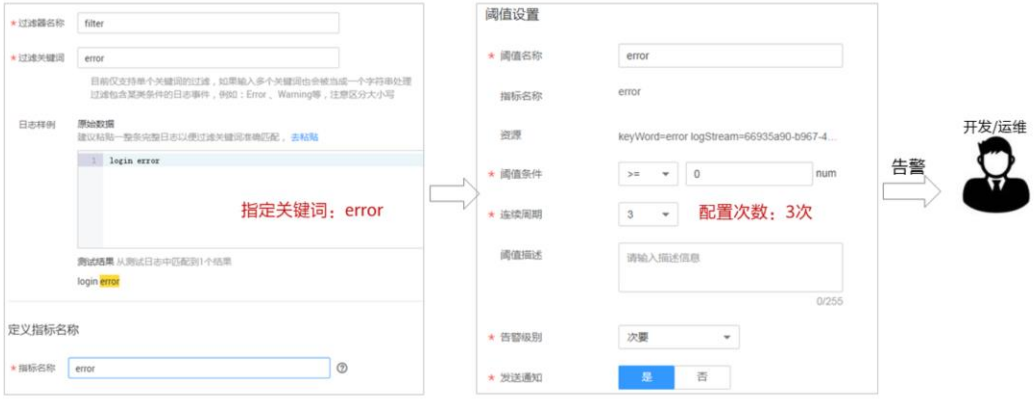
- SQL查询：LTS内置SQL语法引擎，支持30种常用语法，用户通过SQL快速查询和聚合日志内容。



HUAWEI

LTS功能之关键词告警

- 关键词告警：实时监控上报日志中是否包含关键词，如ERR、Exception等，当出现预定次数后，可以产生告警，通知开发人员及时修复业务。



LTS功能之转储

日志转储：将日志从LTS转储到OBS，可永久保存





日志冷/热分离的好处：
1）成本优势：短期的日志存放在LTS，长期的存储到OBS；
2）OBS支持永久保存，LTS支持1~30天，后续规划至180天；

HUAWEI

分享（5min）



1. 您是否有使用过开源或商用日志管理工具?
2. 这些工具帮助你解决哪些问题?

HUAWEI

操作日志

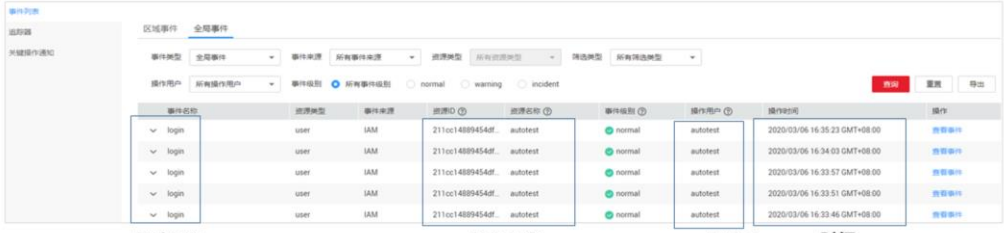
- 概念：
  - 操作日志：按照操作的触发者可以分为用户操作日志、系统操作日志。
  - 用户日志：跟踪用户（最终用户，系统管理员、操作维护员和系统监控员等）的业务相关操作的记录。
  - 系统日志：对系统定时任务执行或事件触发的操作的记录。

HUAWEI

- 操作日志与运行日志的区别:
  - 操作日志记录用户通过客户端或API访问业务系统时，记录的业务操作行为，如登陆/下单等增删改事件。一般业务系统要求完整的记录用户的操作，用于后续的审计或问题分析工作。
  - 运行日志主要是用于记录系统内部函数的运行结果和调用关系等，通常用于问题分析。
  - 操作日志有严格的输出规范和要求，如时间、用户名、操作类型、操作结果等；运行日志通常没有严格的输出规范，输出的内容也很灵活。

华为云操作日志：云审计服务CTS

- 云审计：为您提供云账户下资源的操作记录，通过操作记录您可以实现安全分析、资源变更、合规审计、问题定位等场景。
- 按照服务上线要求，云审计已完成174个服务数据接入。



具体操作　　操作对象　　操作人　　时间

HUAWEI

云审计服务 - 追踪器

- 追踪器：用于记录OBS（对象存储服务）每一次数据的读/写操作，当OBS数据被访问/窃取时，通过CTS可以找到线索。



HUAWEI

云审计服务 - 关键操作通知

Step1

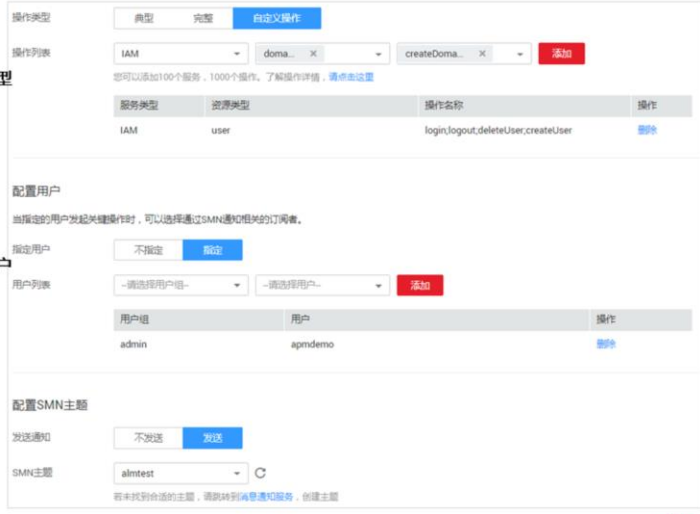
指定云服务和操作类型

Step2

选择所有或指定的用户

Step3

发送邮件或短信



HUAWEI

分享（5min）



1. 您用什么方式记录操作日志?
2. 这些操作日志主要用途是什么?

HUAWEI

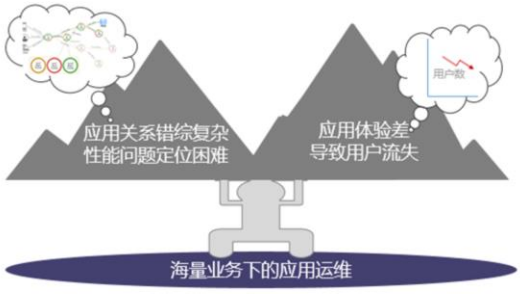
目录

1. 运维的演进与DevOps
2. 运维监控平台概述与云上监控
3. 云上服务日志管理与审计
4. **应用性能管理**
5. 微服务治理概述
6. 应用服务网络

HUAWEI

为什么需要应用性能监控 - 业务爆发式增长

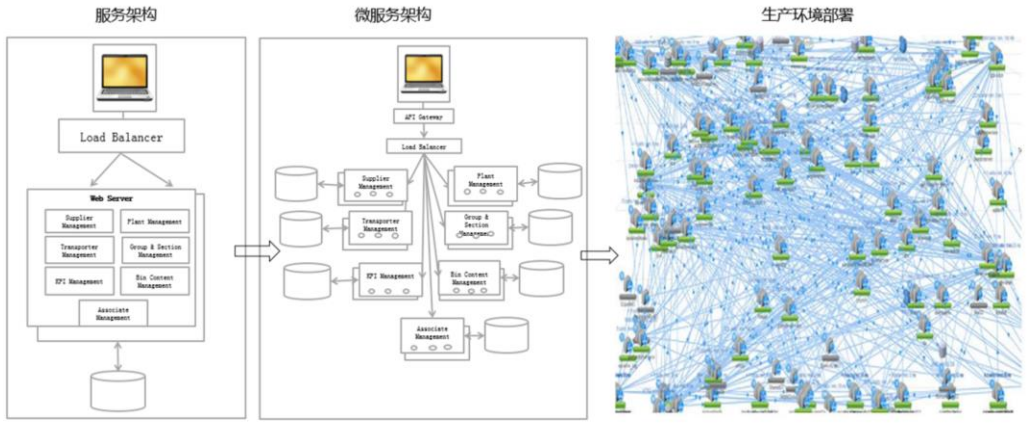
- 在云时代，分布式微服务架构下应用日益丰富，用户数量爆发式增长，纷杂的应用异常问题接踵而来。传统运维模式下，多套运维系统上的各项指标无法关联分析， 运维人员需要根据运维经验逐一排查应用异常，分析定位问题效率低，维护成本高且稳定性差。



HUAWEI

- 海量业务下应用运维面临以下两个方面的挑战：
  - 大型分布式应用关系错综复杂，分析定位应用问题困难，应用运维面临如何保障应用正常、快速完成问题定位、迅速找到性能瓶颈的挑战。
  - 应用体验差导致用户流失。运维人员不能实时感知并追踪体验差的业务，未能及时诊断应用异常，严重影响用户体验。

为什么需要应用性能监控 - 微服务化趋势明显



HUAWEI

- 微服务对业务进行拆分，微服务数量增多，研发运维管理对象过成倍数增长、对业务应用的性能和容量带来新挑战：
  - 管理对象增加：
  - 版本多、变更更为频繁：
  - 服务调用复杂性增加、故障概率增加、问题定界定位困难

什么是APM

- APM 全称是 Application Performance Management, 是指对应用程序的性能和可用性的监控管理。

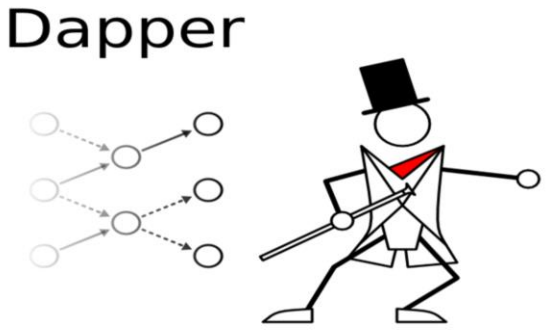


HUAWEI

- APM主要是针对企业 关键业务的IT应用性能和用户体验的监测、优化，提高企业IT应用的可靠性和质量。旨在确保最终用户获得高质量的体验，降低IT总拥有成本TCO（Total Cost of Ownership ），包括产品采购到后期使用、维护的成本。这是一种公司经常采用的技术评价标准。
- 常见的应用性能监控管理包括应用的接口性能和错误监控，分布式调用链路跟踪，以及其他各类用于诊断(内存，线程等)的监控信息，随着云管端的不断延展，已经延伸到了前端，手机App端，IOT等领域监控，并且与容器监控、服务器监控，以及其他平台组件如中间件、数据库等层面的监控一起协同起来，将整个应用给看护起来。

分布式调用链路跟踪

- 目前市面的系统基本都是参考 Google 的 Dapper来做的。跟踪业务请求的处理过程，完成对应用系统在前后端处理、服务端调用的性能消耗跟踪，通过汇聚业务系统各处理环节的实时数据，实现对应用的全链路性能监测。

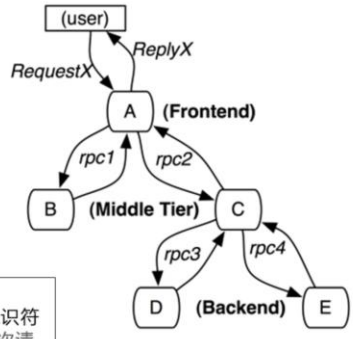


HUAWEI

- Dapper英文版本：http://static.googleusercontent.com/media/research.google.com/zh-CN//pubs/archive/36356.pdf
- Dapper中文版本：https://bigbully.github.io/Dapper-translation/

分布式调用跟踪原理

- 分布式跟踪系统出现的原因是因为在分布式系统中一次请求中会包含很多的RPC，迫切需要一些可以帮助理解系统行为和分析性能问题的工具，需要诊断具体哪个服务性能容量存在问题，从而对整个系统造成瓶颈。



GOOGLE基于标注的方案(annotation-based)：
分布式跟踪系统要做的就是记录每次发送和接受动作的跟踪标识符(message identifiers)和时间戳(timestamped events)，将一次请求涉及到的所有服务串联起来，只有这样才能弄清楚一次请求的完整调用链。

HUAWEI

- 分布式服务的跟踪系统需要记录在一次特定的请求后系统中完成的所有工作的信息。举个例子，图1展现的是一个和5台服务器相关的一个服务，包括：前端（A），两个中间层（B和C），以及两个后端（D和E）。当一个用户（这个用例的发起人）发起一个请求时，首先到达前端，然后发送两个RPC到服务器B和C。B会马上做出反应，但是C需要和后端的D和E交互之后再返还给A，由A来响应最初的请求。对于这样一个请求，简单实用的分布式跟踪的实现，就是为服务器上每一次你发送和接收动作来收集跟踪标识符(message identifiers)和时间戳(timestamped events)。

讨论（5min）



使用过哪些APM工具，具备什么样的特点?

HUAWEI

主流APM产品的对比

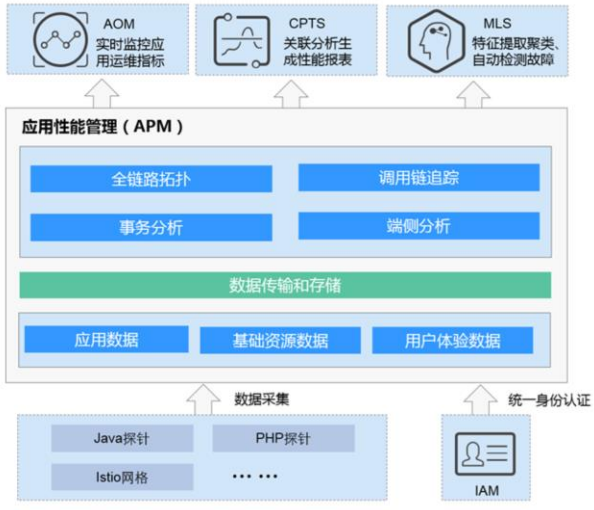
类别	Zipkin	Pinpoint	SkyWalking	CAT
实现方式	拦截请求，发送（HTTP，mq）数据至zipkin服务	java探针，字节码增强	java探针，字节码增强	代码埋点（拦截器，注解，过滤器等）
接入方式	基于linkerd或者sleuth方式，引入配置即可	javaagent字节码	javaagent字节码	代码侵入
颗粒度	接口级	方法级	方法级	代码级
全局调用统计	×	√	√	√
traceid查询	√	×	√	×
报警	×	√	√	√
JVM监控	×	×	√	√

HUAWEI

- CAT是一个更综合性的平台,提供的监控功能最全面，但研发进度及版本更新相对较慢。
- Zipkin由Twitter开源，调用链分析工具，基于spring-cloud-sleuth得到广泛使用，非常轻量，使用部署简单。
- Skywalking专注于链路和性能监控，国产开源，埋点无侵入，UI功能较强。能够加入Apache孵化器，设计思想及代码得到一定认可，后期应该也会有更多的发展空间及研发人员投入。目前使用厂商最多。版本更新较快。
- Pinpoint专注于链路和性能监控，韩国研发团队开源，埋点无侵入，UI功能较强，但毕竟是小团队，不知道会不会一直维护着，目前版本仍在更新中。

华为云APM介绍

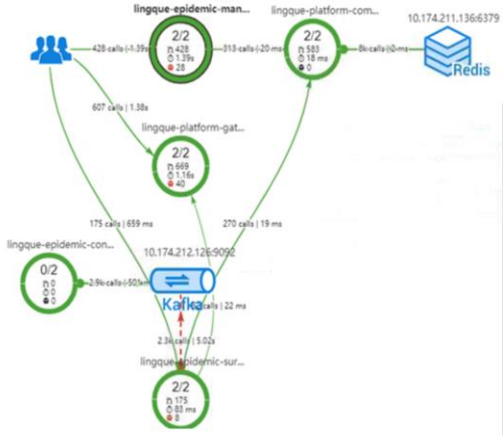
- 华为云应用性能管理服务（Application Performance Management，简称APM）是实时监控并管理云应用性能和故障的云服务，提供专业的分布式应用性能分析能力，可以帮助运维人员快速解决应用在分布式架构下的问题定位和性能瓶颈等难题，为用户体验保驾护航。



HUAWEI

- APM作为云应用诊断服务，拥有强大的分析工具，通过拓扑图、调用链、事务分析可视化地展现应用状态、调用过程、用户对应用的各种操作，快速定位问题和改善性能瓶颈。

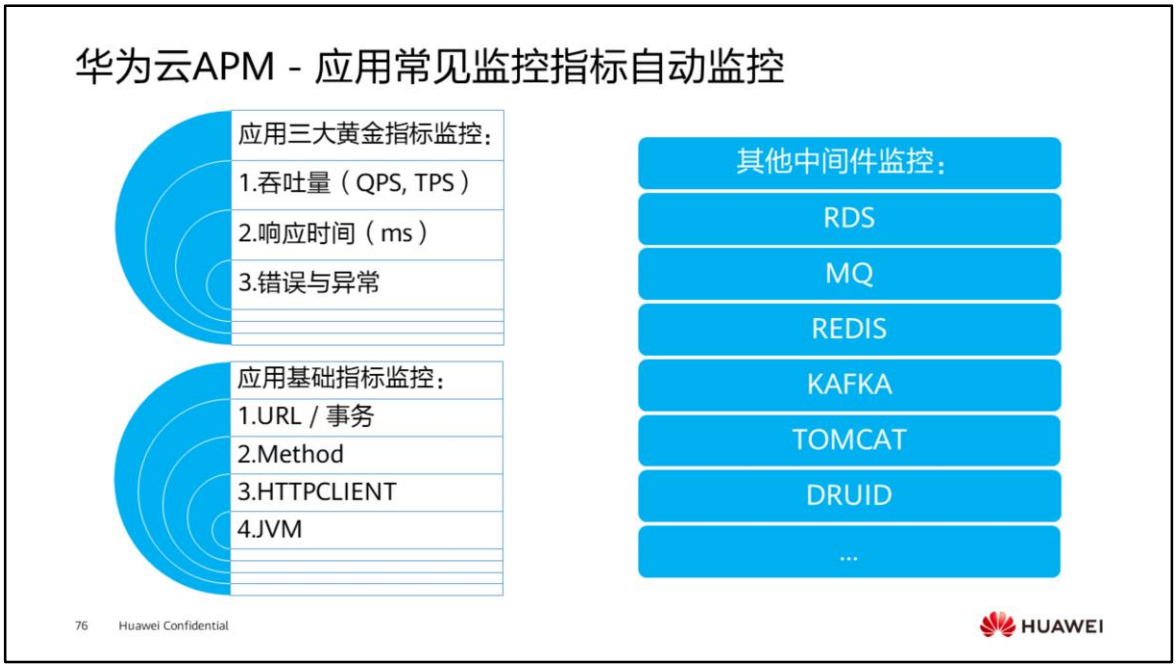
华为云APM - 自动发现调用链路和调用拓扑





HUAWEI

- 极简监控配置：应用拓扑图和调用链自动形成和分析



- 极简监控配置：无侵入式探针，指标自动集成，中间件监控自动添加，批量设置告警



- 业务会话监控：通过对业务监控，业务会话监控：监控每笔交易的KPI数据，提升用户体验。VMALL作为一个电商应用，主交易线为登录网站-选择商品-下单购买-支付，这个就是业务会话。那这个业务中，各个流程中的时延、错误都影响着vmall的用户体验。用户体验在业界有一个标准，就是Apdex。它从用户角度出发，将应用响应时间的表现，转为用户对于应用的满意度评价，量化范围为0-1。值越小，表示用户体验越差。根据vmall性能测试估值出一个apdex阈值。当响应时间在1倍阈值内，表示用户满意，1-4倍阈值内，表示用户可容忍；超过4倍时，表示用户已经烦躁了。业务会话监控会将整个业务的调用次数、错误、时延、apdex都用数字量化出来，这样错误和慢响应就都比较明显了。针对错误或慢响应，还可以查看事务拓扑，即查看该事务和其他的服务联系及情况，评估影响范围；可以设置阈值告警，以保障异常出现时快速响应；可以查看调用关系，找到真正慢或错误在哪里的代码。



- 应用拓扑：应用关系与异常一目了然，精准下钻定位故障。拓扑图是对应用间调用关系和依赖关系的可视化展示。拓扑图主要是由圆圈、箭头连线、资源组成。每个圆圈代表一个应用，圆圈上每个分区代表一个实例。每个圆圈中的分数表示正被调用的实例数/可调用实例数。分数下的内容分别表示在当前所选的时间中应用被调用次数、响应时延、错误数。每个箭头连线代表一个调用关系。调用次数越多，连线越粗。连线上的数据表示调用次数和整体时延。拓扑使用Apdex对用户满意度进行量化，并使用不同颜色对不同区间Apdex的值进行标识。红色表示体验较差，黄色表示体验一般，绿色表示用户较满意。针对错误调用和体验差的调用，可以查看调用链进一步定位原因。拓扑图也支持按照时间、事务、topsql等维度进行筛选。



- 调用链追踪：性能瓶颈与异常原因分钟识别。前面都提到了调用链，那接下来看下调用链、调用关系。调用链可跟踪、记录业务的调用过程，可视化地还原业务请求在分布式系统中的执行轨迹和状态，主要用于性能及故障快速定界。例如针对前面的错误，我们查看调用关系，看到这里有失败。查看失败详情，发现这里报error，具体是SQL语法问题。根据这里的提示即可找到最终的原因。调用链中针对崩溃问题也可以进行堆栈追踪，本次实验我们不做详细介绍，如果您感兴趣，可以通过官方资料或视频学习。

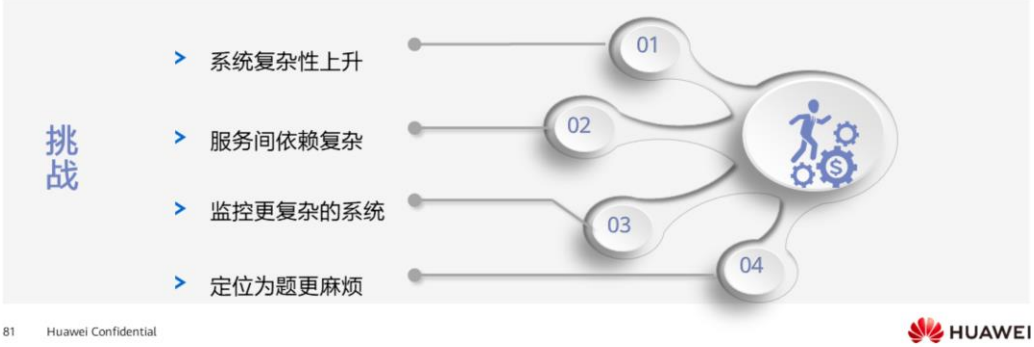
目录

1. 运维的演进与DevOps
2. 运维监控平台概述与云上监控
3. 云上服务日志管理与审计
4. 应用性能管理
5. **微服务治理概述**
6. 应用服务网络

HUAWEI

微服务治理

- 微服务 - 是一种用于构建应用的架构方案。可将应用拆分成多个核心功能。每个功能都被称为一项服务，可以单独构建和部署，这意味着各项服务在工作（和出现故障）时不会相互影响。
- 运维监控挑战 - 这意味着一个应用下面可能会有成百上千个微服务需要治理，如果我们对他们分别运维监控的话，无疑这是个非常庞大且复杂的工作。



HUAWEI

- 衔接话术：我们在使用微服务框架时遇到的运维与监控问题，需要一个技术体系，跟快跟好的支持自己的微服务架构，那么经过实践，我们总结出一下几种常见且有效的微服务治理能力。

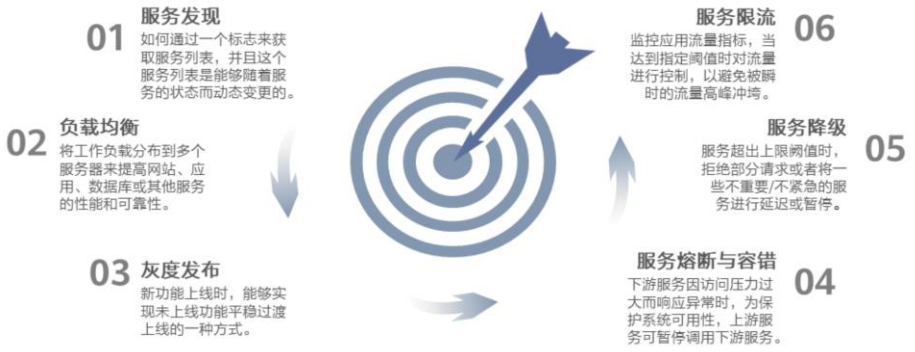
讨论（5min）



微服务治理包含哪些能力，为什么我们需要用到这些能力?

HUAWEI

微服务治理 - 能力列表

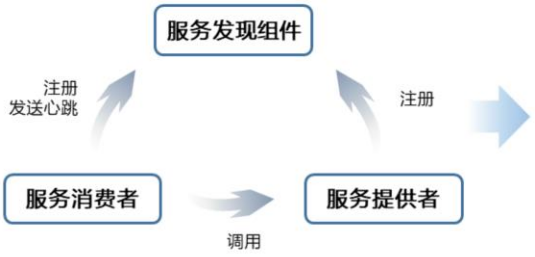


HUAWEI

- 不限于以上几种，另外还有服务容错和安全认证等等...

服务发现 (1)

- 服务治理领域最重要的问题就是服务发现与注册。
- 在一个微服务应用中，一组运行的服务实例是动态变化的，实例有动态分配的网络地址，因此，为了使得客户端能够向服务发起请求，必须要有服务发现机制，在此基础之上进行微服务的其他治理功能。



客户端服务发现

客户端通过查询服务注册中心，获取可用的服务的实际网络地址（IP 和端口）。然后通过负载均衡算法来选择一个可用的服务实例，并将请求发送至该服务。

优点：架构简单，扩展灵活，方便实现负载均衡功能。
缺点：强耦合，有一定开发成本。

服务端服务发现

客户端向load balancer 发送请求。load balancer 查询服务注册中心找到可用的服务，然后转发请求到该服务上。和客户端发现一样，服务都要到注册中心进行服务注册和注销。

优点：服务的发现逻辑对客户端是透明的。
缺点：需要额外部署和维护高可用的负载均衡器。

HUAWEI

服务发现 (2)

服务注册中心

· 服务发现的核心
· 保存可用服务实例的网址(IP Address & Port)
· 可用性和实时更新功能。

Self-Registration

· 服务实例必须自己主动的到注册中心注册和注销。比如可以使用 heartbeat 机制了实现。

Third-Party-Registration

· 通过其他组件来实现服务注册功能。



服务发现：在微服务架构中，服务发现作为一种基本机制，必须得到实现，而服务目录则是一个可选项，如果有的话，用户体验会更好一些。

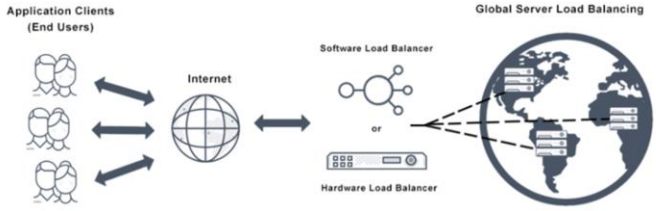
HUAWEI

负载均衡 (1)

- 概述：在分布式系统中，负载均衡（Load Balancing）是一种将任务分派到多个服务端进程的方法。负载均衡保证了分布式服务器中，不会有某一个服务器负载过高。通过将任务均匀的分派到各个服务器，负载均衡可以提高应用的响应速度和可用性。

使用场景：

- 应用于高访问量的业务
- 横向扩张系统
- 消除单点故障
- 同城容灾（多可用区容灾）
- 跨地域容灾

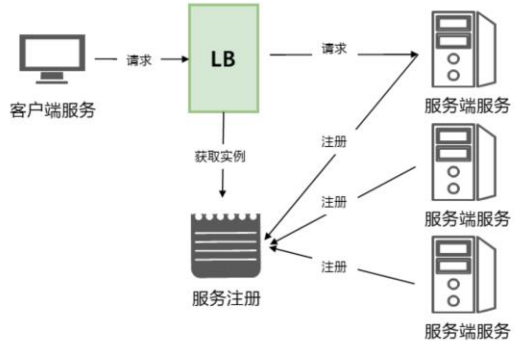


HUAWEI

- https://networkencyclopedia.com/global-load-balancer/ - 图片来源
- https://juejin.im/post/5caeeccee51d456e781f1fcf - 参考
- 使用场景
- 负载均衡主要应用于以下场景中：
- 场景一：应用于高访问量的业务
  - 如果应用访问量很高，您可以通过配置监听规则将流量分发到不同的ECS实例上。此外，您可以使用会话保持功能将同一客户端的请求转发到同一台后端ECS，提高访问效率。
- 场景二：横向扩张系统
  - 可以根据业务发展的需要，通过随时添加和移除ECS实例来扩展应用系统的服务能力，适用于各种Web服务器和App服务器。
- 场景三：消除单点故障
  - 可以在负载均衡实例下添加多台ECS实例。当其中一部分ECS实例发生故障后，负载均衡会自动屏蔽故障的ECS实例，将请求分发给正常运行的ECS实例，保证应用系统仍能正常工作。

- 场景四：同城容灾（多可用区容灾）
    - 为了提供更加稳定可靠的负载均衡服务，阿里云负载均衡已在各地域部署了多可用区以实现同地域容灾。当主可用区出现机房故障或不可用时，负载均衡仍然有能力在非常短的时间内（大约30s中断）切换到另外一个备可用区恢复服务能力；当主可用区恢复时，负载均衡同样会自动切换到主可用区提供服务。
    - 使用负载均衡时，您可以将负载均衡实例部署在支持多可用区的地域以实现同城容灾。此外，建议您结合自身的应用需要，综合考虑后端服务器的部署。如果您的每个可用区均至少添加了一台ECS实例，那么此种部署模式下的负载均衡服务的效率是最高的。
    - 如下图所示，在负载均衡实例下绑定不同可用区的ECS实例。正常情况下，用户访问流量将转发至主可用区内的ECS实例；当可用区A发生故障时，用户访问流量将转发至备可用区内的ECS实例。此种部署既可以避免因为单个可用区的故障而导致对外服务的不可用，也可以通过不同产品间可用区的选择来降低延迟。
    - 如果您采取如下图所示的部署方案，即在负载均衡实例的主可用区下绑定多台ECS实例，而在备可用区没有任何ECS实例。当主可用区发生故障时会造成业务中断，因为备可用区没有ECS实例来接收请求。这样的部署方式很明显是以牺牲高可用性为代价来获取低延时。
- 场景五：跨地域容灾
    - 您可以在不同地域下部署负载均衡实例，并分别挂载相应地域内不同可用区的ECS。上层利用云解析做智能DNS，将域名解析到不同地域的负载均衡实例服务地址下，可实现全局负载均衡。当某个地域出现不可用时，暂停对应解析即可实现所有用户访问不受影响。

负载均衡 (2)

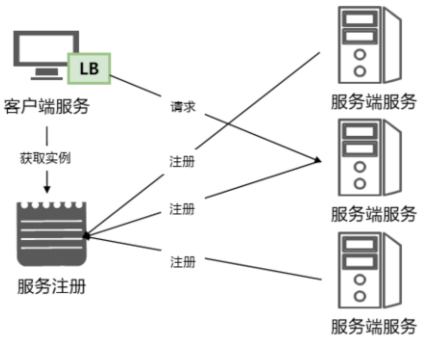


服务端负载均衡

- 微服务实例启动时会将自己的实例信息（包括IP、端口号等）注册到注册中心，并且通过心跳机制维持本实例的在线状态。
- Consumer通过通过负载均衡机制从负载均衡缓存的provider实例列表中选取一个作为本次请求发送的地址。

HUAWEI

负载均衡 (3)



客户端负载均衡

- 微服务实例启动时会将自己的实例信息（包括IP、端口号等）注册到注册中心，并且通过心跳机制维持本实例的在线状态。
- consumer定时去服务中心查询provider的实例。
- consumer调用provider时会通过负载均衡机制从缓存的provider实例列表中选取一个作为本次请求发送的地址。
- 内置的负载均衡策略有RoundRobin、Random、WeightedResponse、SessionStickiness，其中默认使用的是RoundRobin。

HUAWEI

负载均衡 (4)

CSE 负载均衡实战举例：

- 启动一个consumer实例，三个provider实例，调用consumer的greeting方法，可以看到consumer默认是使用RoundRobin策略调用三个provider的。连续调用consumer6次，三个provider的日志记录分别是：

```
2019-02-25 11:38:30,347 [INFO] 192.168.0.45 - - Mon, 25 Feb 2019 11:38:30 CST /provider/v0/greeting 200 65 8 5c73633634a31c51
2019-02-25 11:38:33,615 [INFO] 192.168.0.45 - - Mon, 25 Feb 2019 11:38:33 CST /provider/v0/greeting 200 65 3 5c73633902d84f8a
```

实例1

```
2019-02-25 11:38:31,495 [INFO] 192.168.0.45 - - Mon, 25 Feb 2019 11:38:31 CST /provider/v0/greeting 200 65 3 5c7363376ed08941
2019-02-25 11:38:34,674 [INFO] 192.168.0.45 - - Mon, 25 Feb 2019 11:38:34 CST /provider/v0/greeting 200 65 3 5c73633a652f94c0
```

实例2

```
2019-02-25 11:38:32,625 [INFO] 192.168.0.45 - - Mon, 25 Feb 2019 11:38:32 CST /provider/v0/greeting 200 65 2 5c736338a0353b2e
2019-02-25 11:38:35,697 [INFO] 192.168.0.45 - - Mon, 25 Feb 2019 11:38:35 CST /provider/v0/greeting 200 65 3 5c73633b85f29c50
```

实例3

从时间顺序上可以看出，consumer服务的6次请求是由三个provider实例轮流处理的。

HUAWEI

灰度发布 (1)

- 什么是灰度发布：为保障新特性能平稳上线，可以通过灰度发布功能选择少部分用户试用，降低发布风险。

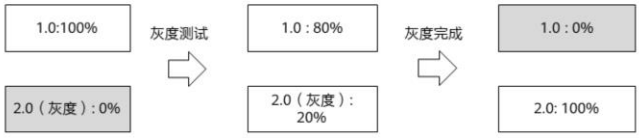
更安全的发布方式，让业务上线不仅仅是快。



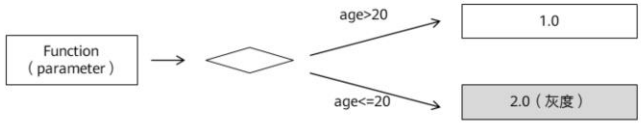
灰度发布 (2)

ServiceComb微服务支持两种灰度策略：

- 流量权重：



- 自定义参数：根据接口参数进行灰度导流



HUAWEI

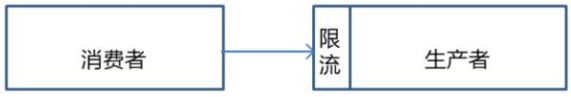
灰度发布 (3)

- CSEJavaSDK支持灰度发布功能，可以实现版本平滑过渡升级。支持按百分比引流和按请求参数特征引流两种方式，百分比引流过程举例：
- 假设一个系统要新上线一个功能 some-feature，需要用户Alice做内部测试后才能完全发布：
  - 线上正在运行服务 some-feature-A，新部署一个服务 some-feature -B，使用新版本的Image和ConfigMap
  - 去负载均衡器页面修改转发规则，新增 header 转发规则将 name=Alice的流量分配给 some-feature-B，剩余流量给 some-feature-A
  - 登录 Alice 用户进行测试，并更新 some-feature-B 的Image和ConfigMap，直至验证通过
  - 修改 some-feature-A 的Image和ConfigMap与 some-feature-B 一致
  - 删除负载均衡器上的转发规则，所有流量指向服务 some-feature-A
  - 删除服务 some-feature-B

HUAWEI

服务限流 (1)

- **限流**：在流量高峰时，可根据消费者优先级 适当调整流量限制，保护生产者不被 流量击垮。
- **目的**：是通过对并发访问/请求进行限速或者一个时间窗口内的的请求进行限速来保护系统，一旦达到限制速率则可以拒绝服务（定向到错误页或告知资源没有了）、排队或等待（比如秒杀、评论、下单）、降级（返回兜底数据或默认数据，如商品详情页库存默认有货）。



HUAWEI

服务限流 (2)

- 一般开发高并发系统常见的限流策略有：
  - 限制总并发数（比如数据库连接池、线程池）
  - 限制瞬时并发数（如nginx的limit_conn模块，用来限制瞬时并发连接数）
  - 限制时间窗口内的平均速率（如Guava的RateLimiter、nginx的limit_req模块，限制每秒的平均速率）
  - 其他还有如限制远程接口调用速率、限制MQ的消费速率。
  - 另外还可以根据网络连接数、网络流量、CPU或内存负载等来限流。

HUAWEI

服务限流 (3)

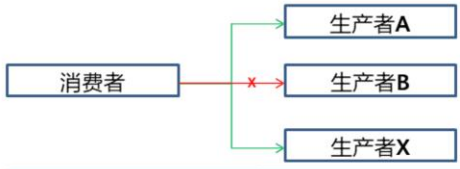
- 华为云CSE支持default、微服务、schema（契约）、operation（接口）限流四个粒度的限流
  - provider服务上配置限流策略为对edge服务限流至1QPS，通过edge服务连续调用consumer和provider服务。
  - 观察provider服务中打印的日志，可以看到每秒钟provider只处理一次来自edge服务的请求，其他请求都以429状态码返回的。而对于来自consumer的请求，provider正常处理并返回200状态码。

```
09:44:37 CST /provider/v0/hello/Alice 200 13 2 5c75eb8566ff9bf7
09:44:38 CST /provider/v0/hello/Bob 200 11 1 5c75eb86c817a894 or
09:44:38 CST /provider/v0/hello/Alice 200 13 1 5c75eb866c170711
09:44:38 CST /provider/v0/hello/Bob 429 41 1 5c75eb86281fa768 or
09:44:38 CST /provider/v0/hello/Alice 200 13 1 5c75eb868440acd4
09:44:38 CST /provider/v0/hello/Bob 429 41 1 5c75eb860ea48971 or
09:44:38 CST /provider/v0/hello/Alice 200 13 2 5c75eb867246b63d
09:44:38 CST /provider/v0/hello/Bob 429 41 0 5c75eb8625404de or
09:44:38 CST /provider/v0/hello/Alice 200 13 2 5c75eb86f17d2d98
09:44:38 CST /provider/v0/hello/Bob 429 41 0 5c75eb86fbecc4a4 or
09:44:38 CST /provider/v0/hello/Alice 200 13 1 5c75eb86a680379a
09:44:38 CST /provider/v0/hello/Bob 429 41 1 5c75eb8686971aac or
09:44:39 CST /provider/v0/hello/Alice 200 13 1 5c75eb8762f30c07
09:44:39 CST /provider/v0/hello/Bob 200 11 1 5c75eb87567287cd or
09:44:39 CST /provider/v0/hello/Alice 200 13 1 5c75eb877cd27728
09:44:39 CST /provider/v0/hello/Bob 429 41 1 5c75eb8715293b46 or
09:44:39 CST /provider/v0/hello/Alice 200 13 1 5c75eb87d02ef8c3
09:44:39 CST /provider/v0/hello/Bob 429 41 1 5c75eb87a133c166 or
```


HUAWEI

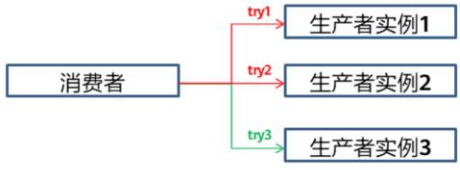
服务熔断与容错 (1)

目的：保护系统整体的可用性。



- 熔断：

在某个生产者在指定时间段持续出现 故障时，消费者主动断开其连接，如 果生产者故障排除，则连接自动恢复。

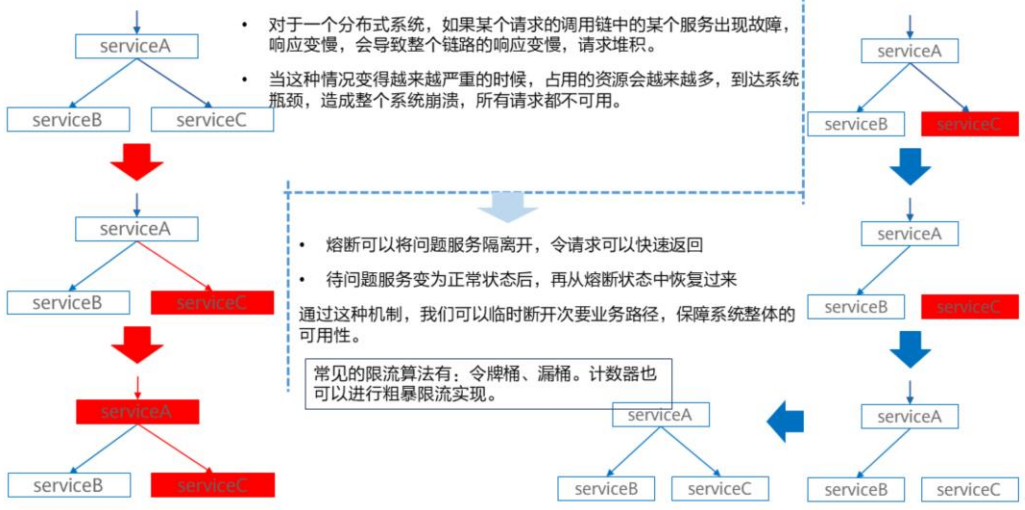


- 容错：

在消费者访问生产者的某一个实例失 败时，则根据容错策略自动进行错误 处理，如：选择另一个实例重试、按 时间间隔持续重试同一实例，或者快 速返回失败不重试。

HUAWEI

服务熔断与容错 (2)



- 对于一个分布式系统，如果某个请求的调用链中的某个服务出现故障，响应变慢，会导致整个链路的响应变慢，请求堆积。
- 当这种情况变得越来越严重的时候，占用的资源会越来越多，到达系统瓶颈，造成整个系统崩溃，所有请求都不可用。

- 熔断可以将问题服务隔离开，令请求可以快速返回
- 待问题服务变为正常状态后，再从熔断状态中恢复过来

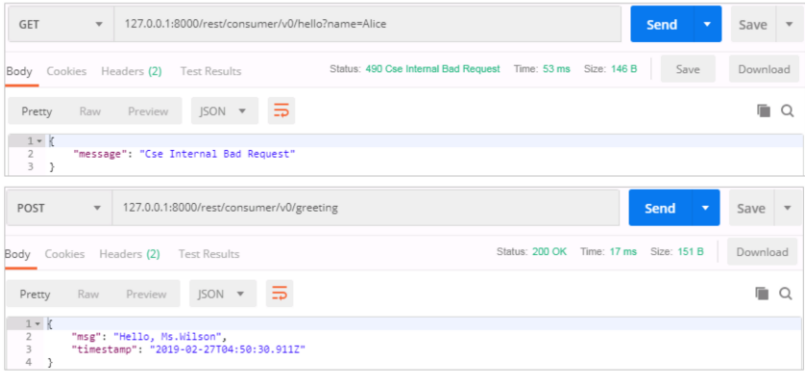
通过这种机制，我们可以临时断开次要业务路径，保障系统整体的可用性。

常见的限流算法有：令牌桶、漏桶。计数器也可以进行粗暴限流实现。

HUAWEI

服务熔断 - 手动熔断

- 熔断可以手动开启，也可以自动开启。其触发方式不同，但效果相同。
  - 这里我们选择edge服务，手工熔断其调用consumer服务的sayHello方法的路径，然后手工熔断edge服务调用consumer服务sayHello方法的路径。
  - 通过edge服务调用consumer服务的sayHello方法和greeting方法，可以看到sayHello方法已经调不通了，但greeting方法仍然能够调通



HUAWEI

服务熔断 - 自动熔断

配置自动熔断规则：

```
private DynamicLongProperty helloDelay = DynamicPropertyFactory.getInstance().getLongProperty( propName: "delay.sayHello",  defaultValue: 0);

@Path("/hello")
@GET
public String sayHello(@QueryParam("name") String name) {
  // RPC 调用方式体验与本地调用相同
  if (helloDelay.get() > 0) {
    try {
      Thread.sleep(helloDelay.get());
    } catch (InterruptedException e) {
      LOGGER.error("sayHello sleeping is interrupted!", e);
    }
  }
  return helloService.sayHello(name);
}
```

consumer服务的sayHello方法里增加线程sleep的逻辑

启动provider、consumer、edge服务后，在配置中心设置delay.sayHello=40000，多次发请求通过edge调用consumer的sayHello方法，触发edge服务到consumer服务sayHello方法请求路径的熔断。
可以看到熔断发生后，edge调用consumer的sayHello方法不会等待30秒超时后再返回错误，而是立即就返回错误了



HUAWEI

服务降级 (1)

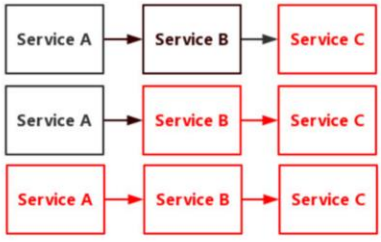
- 降级：在生产者出现故障时，消费者可主动 断开与生产者的连接，以保护消费者 避免故障传染，能正常对外提供服务。
- 目的：保证重要或基本服务正常运行，非重要服务延迟使用或暂停使用。如若不熔断，很有可能会产生雪崩。



HUAWEI

服务降级 (2)

- 降级策略有三个相关的技术概念：“隔离”、“熔断”、“容错”：
  - “隔离”是一种异常检测机制，常用的检测方法是请求超时、流量过大等，一般的设置参数包括超时时间、同时并发请求个数等。
  - “熔断”是一种异常反应机制，“熔断”依赖于“隔离”。熔断通常基于错误率来实现，一般的设置参数包括统计请求的个数、错误率等。
  - “容错”是一种异常处理机制，“容错”依赖于“熔断”。熔断以后，会调用“容错”的方法，一般的设置参数包括调用容错方法的次数等。



把这些概念联系起来：当"隔离"措施检测到N次请求中共有M次错误的时候，"熔断"不再发送后续请求，调用"容错"处理函数。

当前ServiceComb提供两种容错方式，分别为返回null值和抛出异常。

HUAWEI

- 图片为雪崩

服务降级 (3)

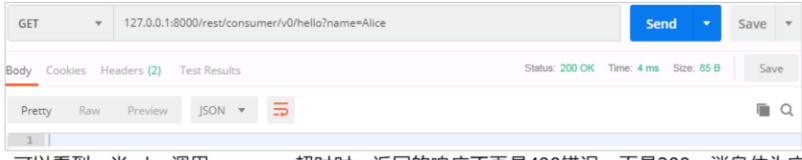
以CSE距离，我们来看紧急的效果。设置降级策略为返回null：

设置前：



请求超时

设置后：



触发熔断

可以看到，当edge调用consumer超时时，返回的响应不再是490错误，而是200，消息体为空

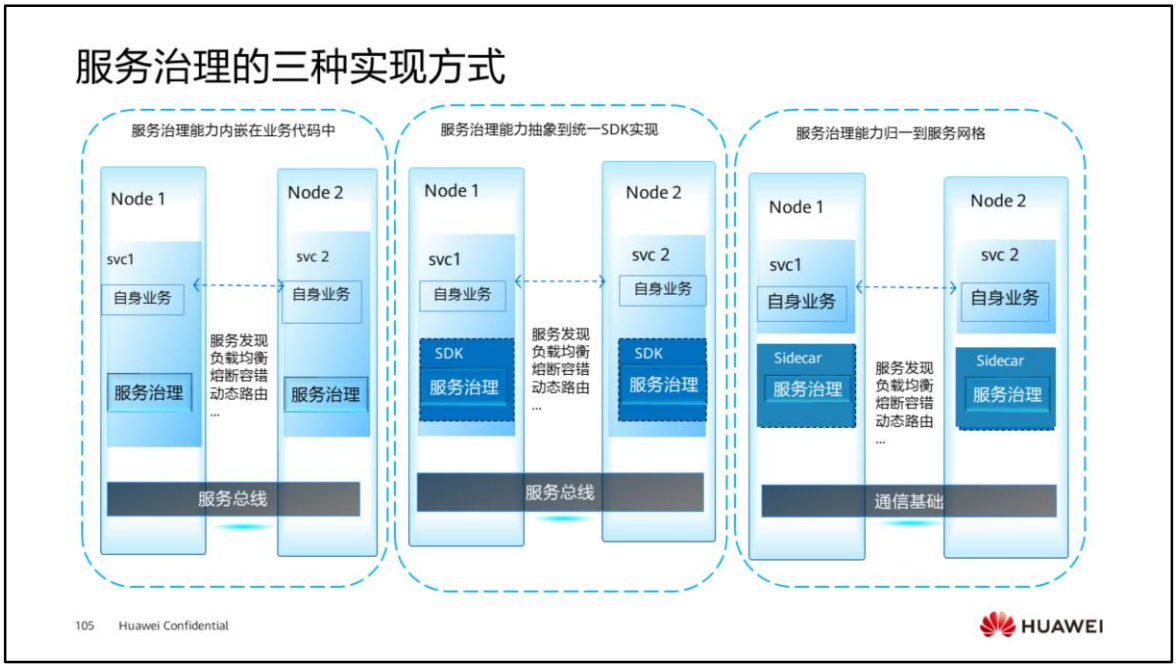
熔断发生后，edge服务返回的也是空消息体

HUAWEI

目录

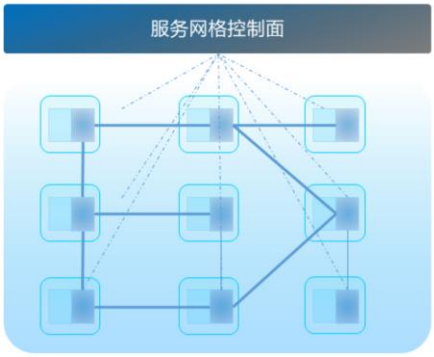
1. 运维的演进与DevOps
2. 运维监控平台概述与云上监控
3. 云上服务日志管理与审计
4. 应用性能管理
5. 微服务治理概述
6. **应用服务网络**

HUAWEI



- 第一代优点：
  - 简单使用依赖少
- 第一代限制：
  - 代码耦合、代码重复高、运维复杂、开发要求高
- 第二代优点：
  - 代码重复少；治理逻辑代码和业务代码分开
- 第二代限制：
  - SDK语言绑定；代码侵入；基于SDK开发学习门槛高；在用系统改造代价大；治理能力升级影响用户业务；
- 第三代优点：
  - 独立进程，用户业务非侵入；语言无关；治理逻辑升级业务无感知；已有系统无需改造即可治理；可以渐进的微服务化；
- 第三代限制：
  - 代理的性能和资源开销

服务网格：一种云原生的应用层网络技术



- 云原生：面向弹性、（微）服务化、去中心化业务场景
- 应用层：以应用为中心，关注应用的发布、监控、恢复等
- 网络：关注应用组件之间的接口、流量、数据、访问安全等

HUAWEI

Istio是Kubernetes上原生的服务网格技术

- Istio是一个提供连接、保护、控制以及观测功能的开放平台，通过提供完整的非侵入式的微服务治理解决方案，能够很好的解决云原生服务的管理、网络连接以及安全管理等服务网络治理问题。

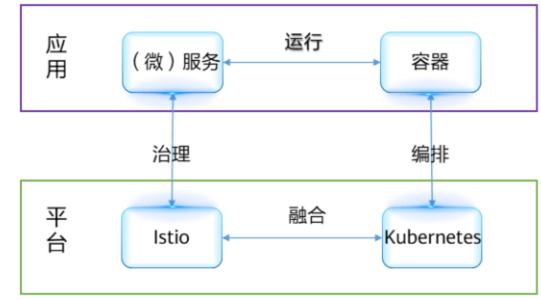


HUAWEI

- 随着微服务的大量应用，其构成的分布式应用架构在运维、调试、和安全管理等维度变得更加复杂，开发者需要面临更大的挑战，如：服务发现、负载均衡、故障恢复、指标收集和监控，以及A/B测试、灰度发布、蓝绿发布、限流、访问控制、端到端认证等。
- 在较高的层次上，Istio有助于降低这些部署的复杂性，并减轻开发团队的压力。它是一个完全开源的服务网格，可以透明地分层到现有的分布式应用程序上。它也是一个平台，包括允许集成到任何日志记录平台、遥测或策略系统的API。Istio的多样化功能使您能够成功高效地运行分布式微服务架构，并提供保护、连接和监控微服务的统一方法。
- 服务网格（Service Mesh）通常用于描述构成应用程序的微服务网络以及应用之间的交互。它的需求包括服务发现、负载均衡、故障恢复、指标收集和监控以及通常更加复杂的运维需求，例如 A/B 测试、金丝雀发布、限流、访问控制和端到端认证等。

Istio + Kubernetes：云原生应用治理 + 云原生应用设施

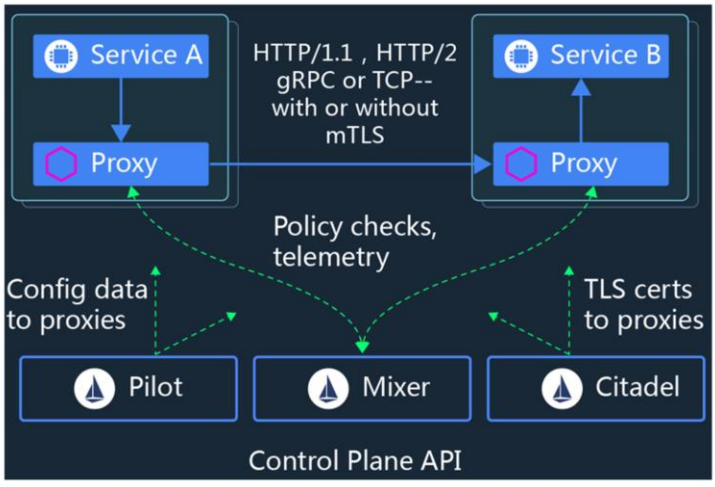
- 容器和微服务共同的轻量、敏捷的特点，微服务运行在容器中日益流行；
- Kubernetes在容器编排领域成为事实标准
- Istio 和 Kubernetes 紧密结合。基于Kubernetes构建，补齐了Kubernetes的治理能力，提供了端到端的微服务运行治理治理平台。
- Istio 提供ServiceMesh方式无侵入微服务治理能力，成为微服务治理的趋势



对于云原生应用，采用Kubernetes构建微服务部署和集群管理能力，采用Istio构建服务治理能力，将逐渐成为应用微服务转型的标准配置。

HUAWEI

Istio服务架构

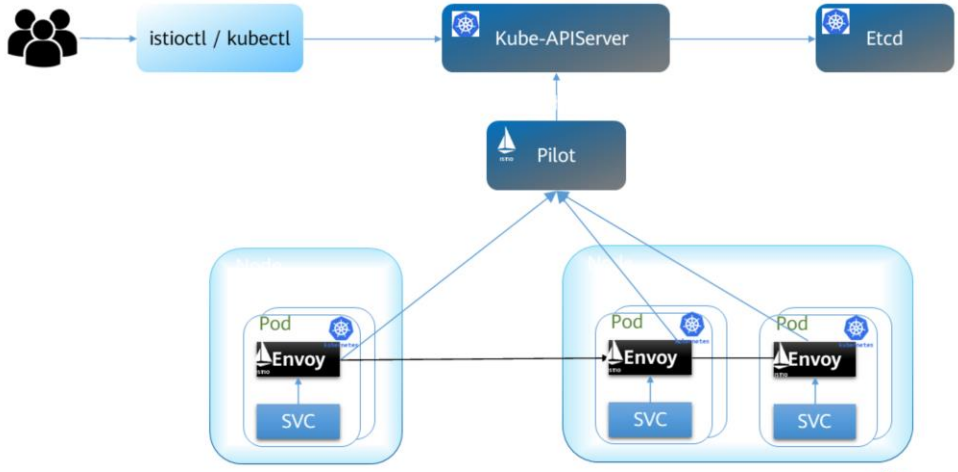


HUAWEI

- 应用服务网格在逻辑上分为控制层面和数据层面：
  - 控制层面：管理代理（默认为Envoy），用于管理流量路由、运行时策略执行等。
  - 数据层面：由一组以 sidecar 方式部署的智能代理（Envoy）组成。这些代理可以调节和控制微服务及 Mixer之间所有的网络通信。

- 应用服务网格主要由以下组件构成：
    - Istio管理器（Pilot）：负责收集和验证配置，并将其传播到各种Istio组件。它从策略执行模块（Mixer）和智能代理（Envoy）中抽取环境特定的实现细节，为他们提供用户服务的抽象表示，独立于底层平台。此外，流量管理规则（即通用4层规则和7层HTTP/gRPC路由规则）可以在运行时通过Pilot进行编程。
    - 策略执行模块（Mixer）：负责在服务网格上执行访问控制和使用策略，并从智能代理（Envoy）和其他服务收集遥测数据。依据智能代理（Envoy）提供的属性执行策略。
    - Citadel：Citadel通过内置身份和凭证管理赋能强大的服务间和最终用户身份验证。可用于升级服务网格中未加密的流量，并为运维人员提供基于服务标识而不是网络控制的强制执行策略的能力。
    - Galley：Galley 代表其他的 Istio 控制平面组件，用来验证用户编写的 Istio API 配置。
    - 智能代理（Envoy）：作为一个独立的组件与相关微服务部署在同一个Kubernetes的pod上，并提供一系列的属性给策略执行模块（Mixer）。策略执行模块（Mixer）以此作为执行策略的依据，并发送到监控系统。

Istio遵从Kubernetes架构：CRD、容器化、Pod



HUAWEI

Istio关键功能

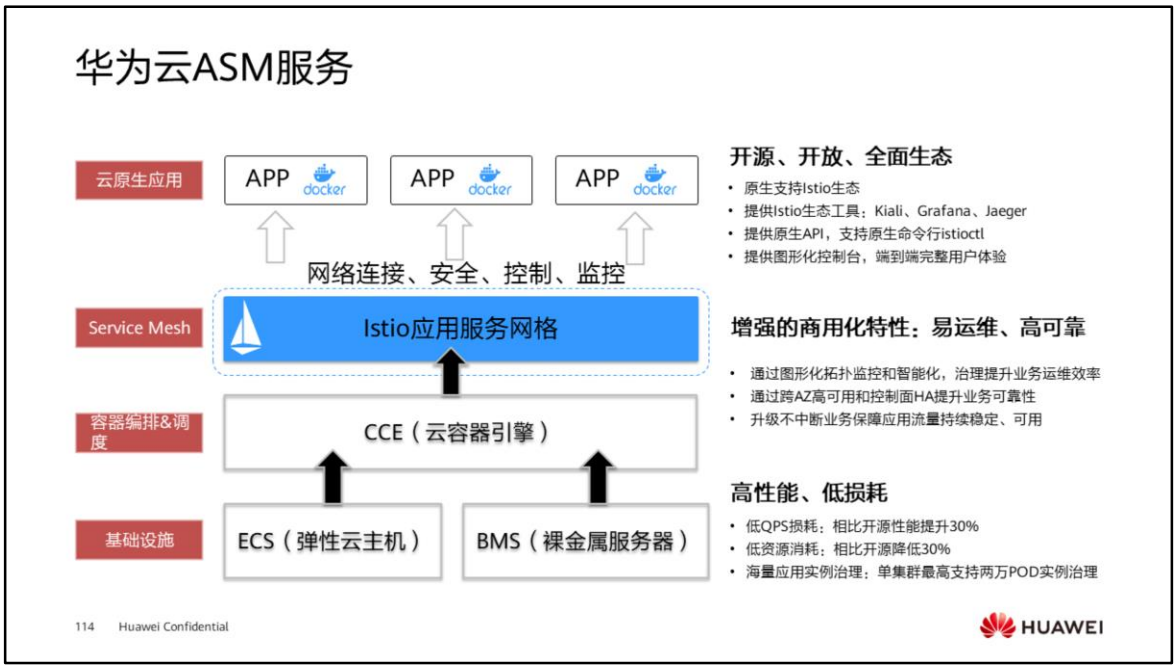


HUAWEI

Istio与Spring Cloud的对比

比较观点	Istio	Spring Cloud
代码形态	代码非侵入	治理能力和业务代码在一个进程
开发语言	语言无关	只支持Java
已有系统治理需求	无需任何改变，治理能力直接可用。	用SDK重新开发，改造成本高，不可行。
单体应用微服务化	可渐进微服务化。Istio不限于微服务的治理，微服务化的服务和未微服务化的服务间访问被治理。	一次全部改造，风险大。
治理能力扩展升级	业务发布件无影响。治理框架的管理面和数据面Proxy升级，用户业务完全无感知。	业务发布件需要升级。治理能力导致SDK需要升级(或SDK Bug修复）引起用户的发布件变更，引入新的工作量和潜在风险
K8S Native	基于k8s能力构建： 1.使用k8s 服务名，使用和K8S一致的服务发现； 2.完全K8S使用体验。Sidecar自动pod注入，业务无感知，和部署普通K8s负载无差别； 3. 无需额外的APIServer和规则策略定义，基于K8S CRD扩展。	和K8S无结合 1.两套服务发现，K8s中Pod的迁移可能导致注册中心数据更新不及时，短期访问不通。 2.和K8S没有结合，K8S只是提供了运行环境，K8S未来的扩展功能不能使用； 3.需自安装和维护控制面来管理治理规则
社区活跃度	社区活跃，版本迭代稳健，每小时都有commit。	Netflix提供的Spring Cloud核心组件（Eureka、Hystrix等）停止开发

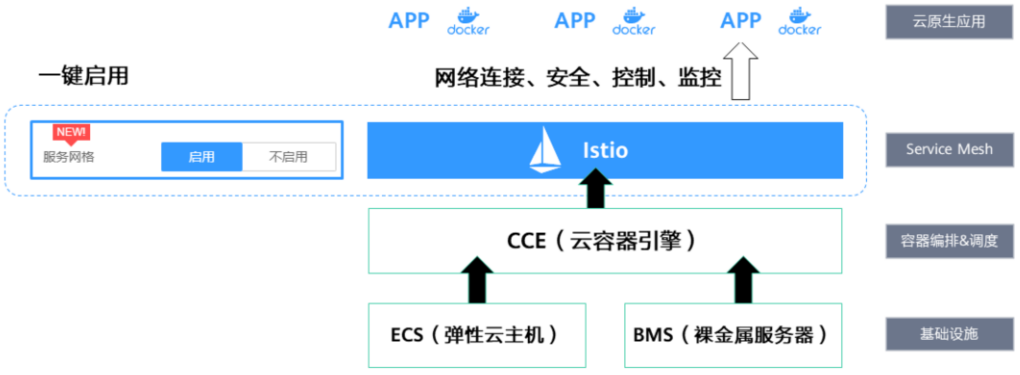
HUAWEI



- 华为云ASM服务：基于Istio的高性能、高可靠、非侵入的企业级Service Mesh服务
  - 应用服务网格（Application Service Mesh，简称ASM）是华为云基于开源Istio推出的服务网格平台，它深度、无缝对接了华为云的企业级Kubernetes集群服务云容器引擎（Cloud Container Engine，简称CCE），在易用性、可靠性、可视化等方面进行了一系列增强，可为客户提供开箱即用的上手体验。
  - 应用服务网格提供非侵入式的微服务治理解决方案，支持完整的生命周期管理和流量治理，兼容Kubernetes和Istio生态，其功能包括负载均衡、熔断、限流等多种治理能力。应用服务网格内置金丝雀、蓝绿、A/B Test等多种灰度发布流程，提供一站式自动化的发布管理。应用服务网格基于无侵入的监控数据采集，提供实时流量拓扑、调用链等服务性能监控和运行诊断，构建全景的服务运行视图。
  - ASM的最大优势是相较于传统的微服务框架，如SpringCloud、Dubbo等，在于语言无关同时不需要修改代码，是业界公认的云原生技术所推动的第三代微服务技术

华为云ASM服务 - 一键启用

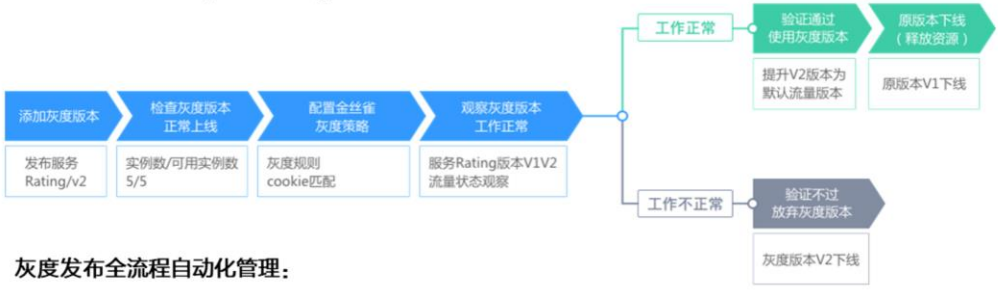
- 无需安装，一键启用，无侵入实现应用治理



HUAWEI

华为云ASM服务 - 灰度发布

- 内置金丝雀、A/B Testing典型灰度发布流程



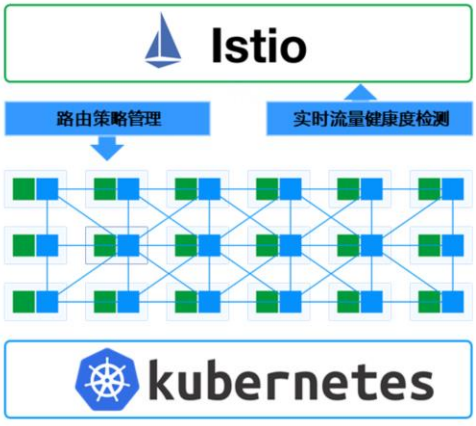
灰度发布全流程自动化管理：

- 灰度版本一键部署，流量切换一键生效
- 配置式灰度策略，支持流量比例、请求内容（Cookie、OS、浏览器等）、源IP
- 一站式健康、性能、流量监控，实现灰度发布过程量化、智能化、可视化

HUAWEI

华为云ASM服务 - 流量管理

- 策略化的智能路由与弹性流量管理



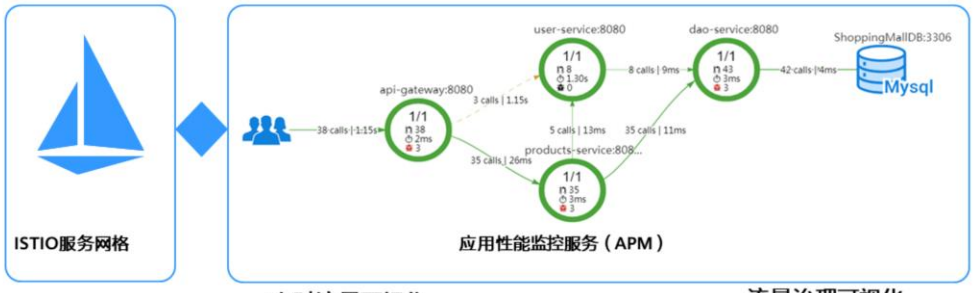
无侵入智能流量管理：

- 权重、内容等路由规则，实现应用灵活灰度发布
- HTTP会话保持，满足业务处理持续性诉求
- 限流、熔断，实现服务间链路稳定、可靠
- 服务安全认证：认证、鉴权、审计等，提供服务安全保障基石

HUAWEI

华为云ASM服务 - 可视化

- 图形化应用全景拓扑，流量治理可视化



实时流量可视化

- 链路健康状态
- 响应时延
- 链路请求数
- 链路异常响应

流量治理可视化

- 路由管理
- 限流、熔断
- 故障注入

HUAWEI

- 图片出处：华为云

思考题

1. （判断）限流是为了改变网络流量所经过的途径而修改路由信息的技术。

( ) TRUE

( ) FALSE

HUAWEI

- 参考答案：TRUE

本章总结

- 本章介绍了运维的演进及云上监控方案的介绍，讲解了云上日志管理和审计流程，并结合当下微服务架构的流行趋势，总结了微服务的应用性能管理和治理方式。

HUAWEI

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

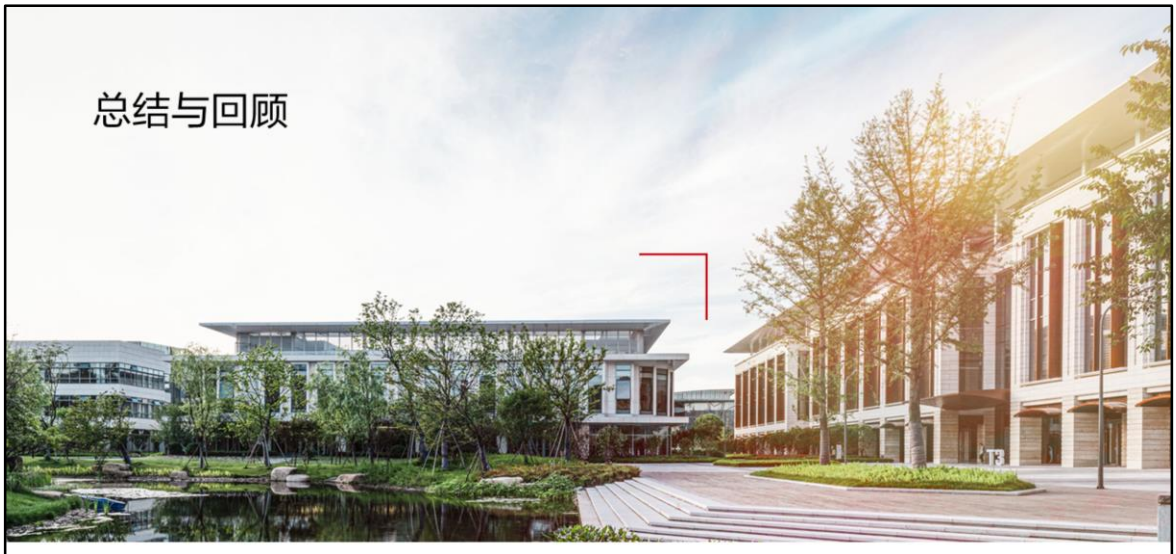
Bring digital to every person, home, and organization for a fully connected, intelligent world.

Copyright©2020 Huawei Technologies Co., Ltd.
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that
could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

HUAWEI

总结与回顾



HUAWEI

前言

- 经过前七章的学习，相信你已经对华为端到端的DevOps流程有了更深刻的理解。接下来我们介绍华为云端到端DevOps实现案例，从最佳实践中深入理解DevOps开发和运维是如何运作，并最终实现工作效率的提高。本章最后详细介绍DevOps的转型路径，将所学的实践经验运用到开发工作中，对全部的学习做一个系统性的回顾，并掌握如何落地DevOps转型。

HUAWEI

目标

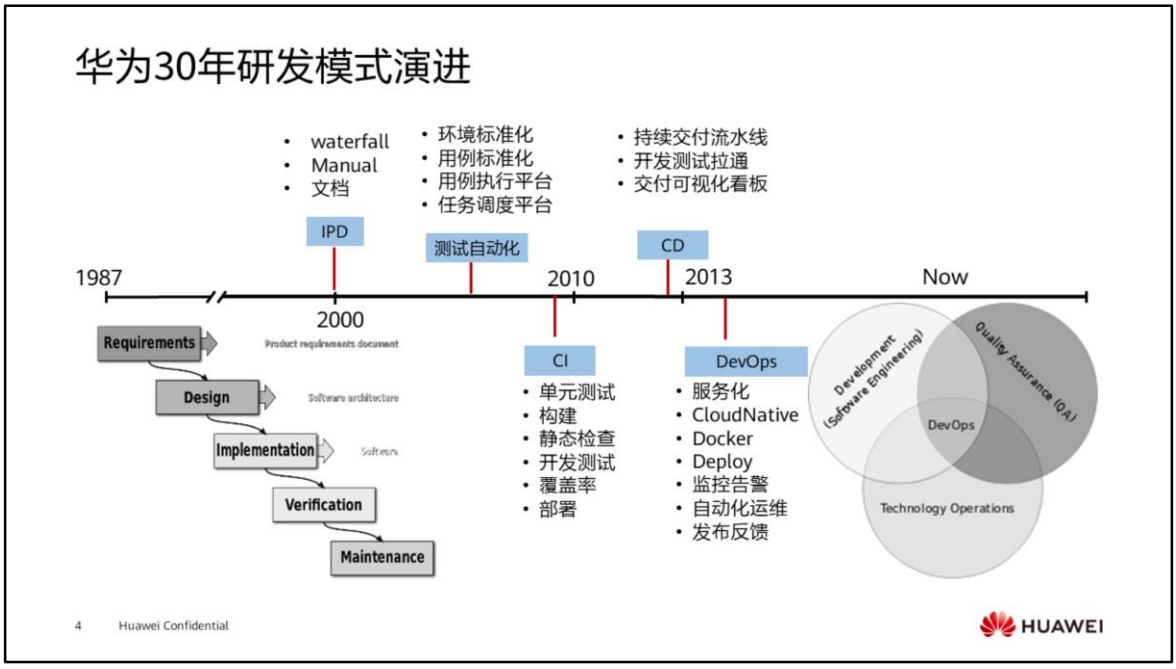
- 学完本章课程后，您将能够：
  - 通过华为DevCloud案例熟悉DevOps端到端流程
  - 掌握DevOps转型的具体阶段

HUAWEI

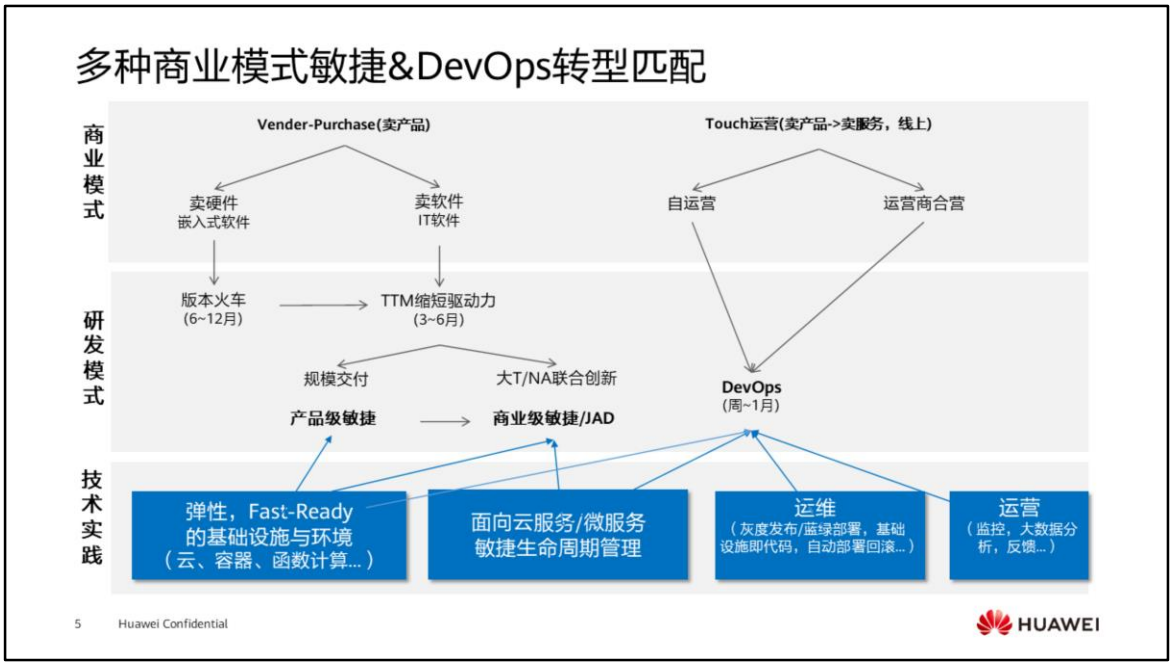
目录

1. **华为云端到端DevOps实践**
2. DevOps转型路径

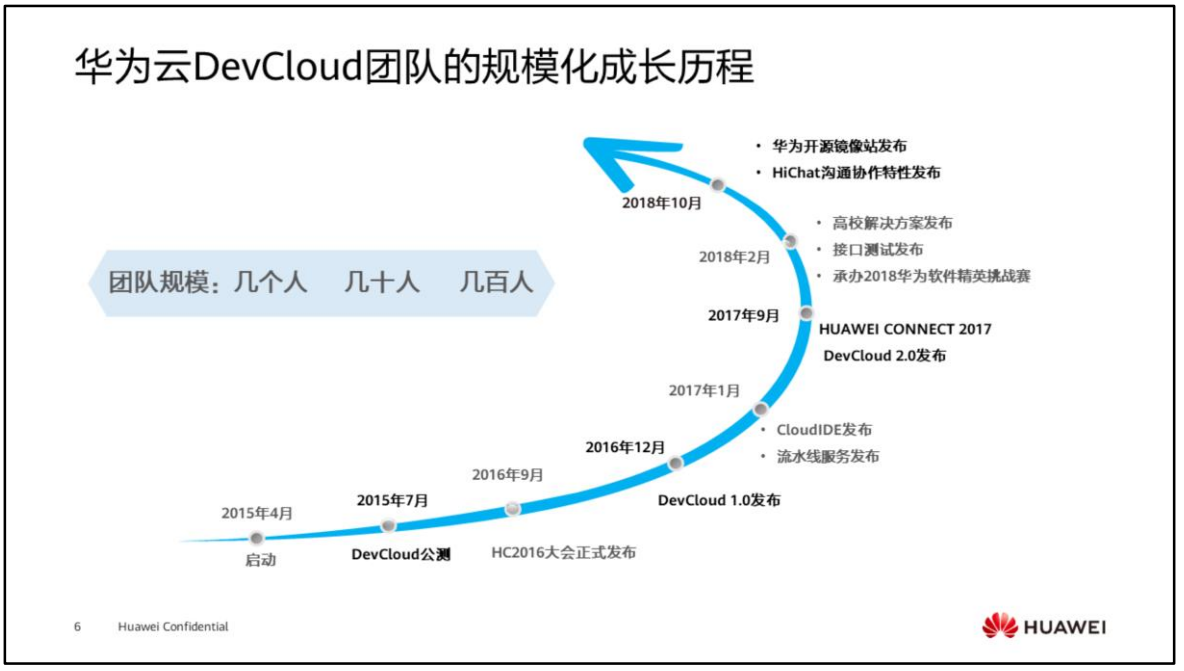
HUAWEI



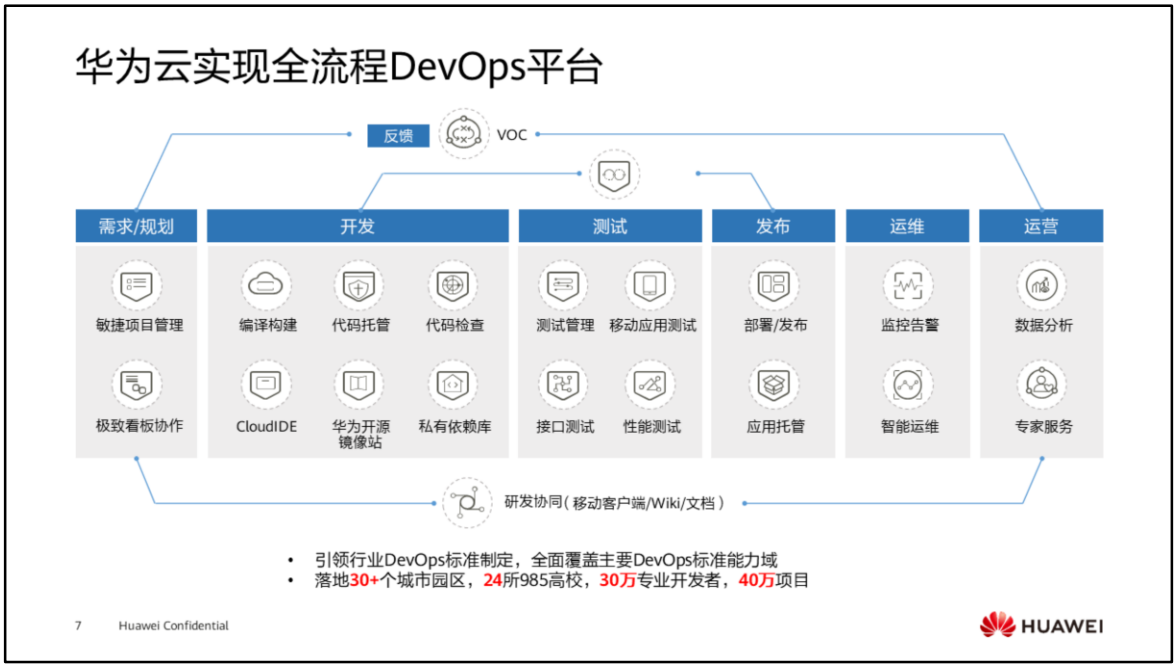
- 华为是87年创建，那时做软件开发用的都是瀑布流。
- 到了2000年左右引进了IPD开发模式，这个模式用了很久，降低了研发费用的同时提高效率。
- 10年前后，CI/CD的理念传播过来，开始采用其中的一些原则，改变了开发方式。后来DevOps的理念也传播过来了，华为不同的部门根据自己的业务情况，逐渐转型微服务化云原生使用容器等多项最佳实践。



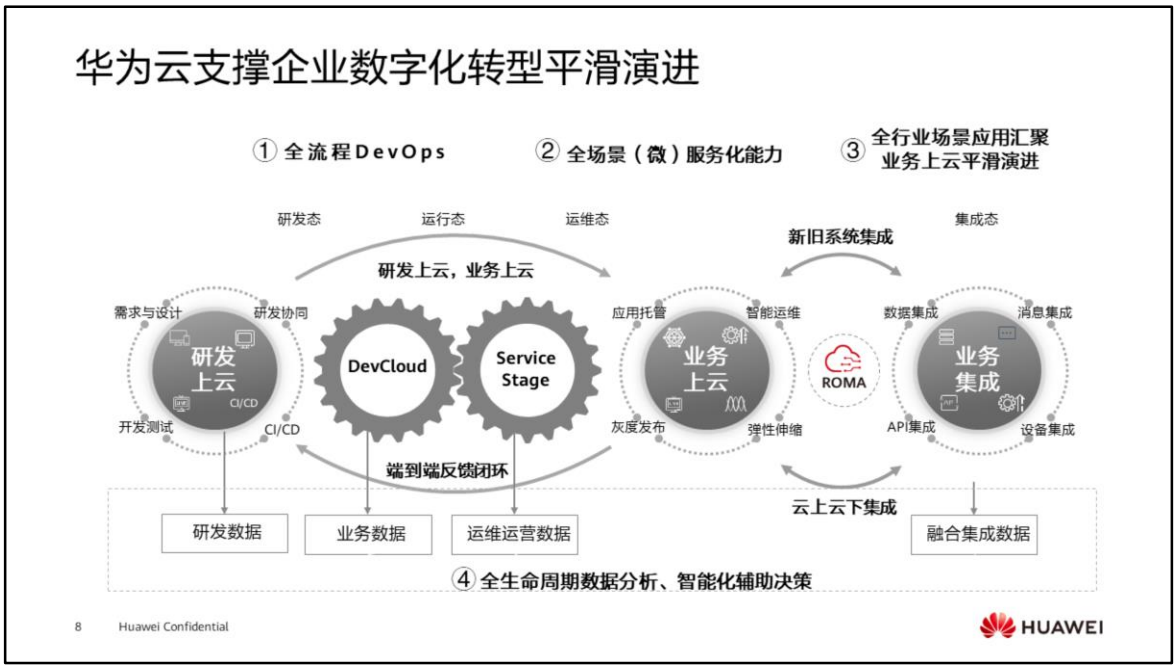
- 华为云DevCloud支持多种商业模式敏捷&DevOps转型匹配。



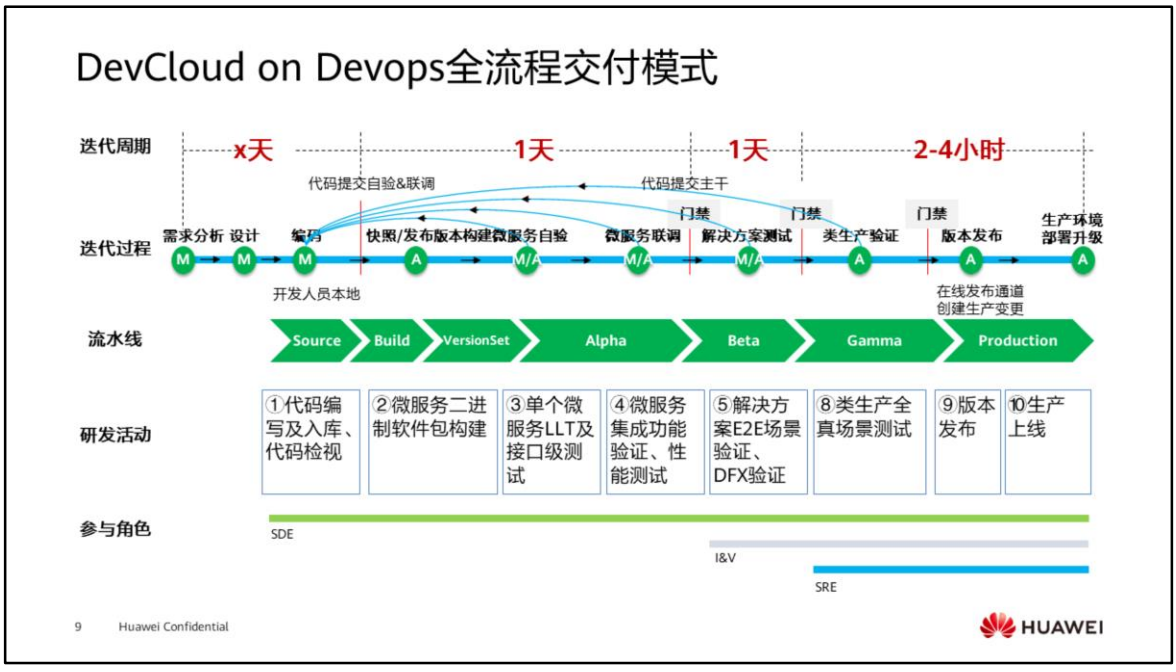
- 华为云DevCloud是华为对外开放出来的开发者工具。这个工具原本是内部使用的，后来慢慢转为商用，并且根据市场需要增加功能，团队也在逐渐扩大。本土图为华为云DevCloud团队的规模化成长历程。
- DevCloud发展历程：
  - 2015.07公测：只提供项目管理、代码托管、编译构建服务。
  - 2016.09：正式发布
  - 2016.12 1.0版本：增加代码检查、部署服务、发布服务，代码托管等功能。
  - 2017.09 2.0版本：移动应用测试发布、Wiki特性发布、文档管理特性发布。
- 通过华为云DevCloud 实现全流程DevOps，包括从需求规划到开发、测试、发布、运维、运营、反馈 整个这一套流程，至今包括超过15个服务，实现端到端的DevOps。



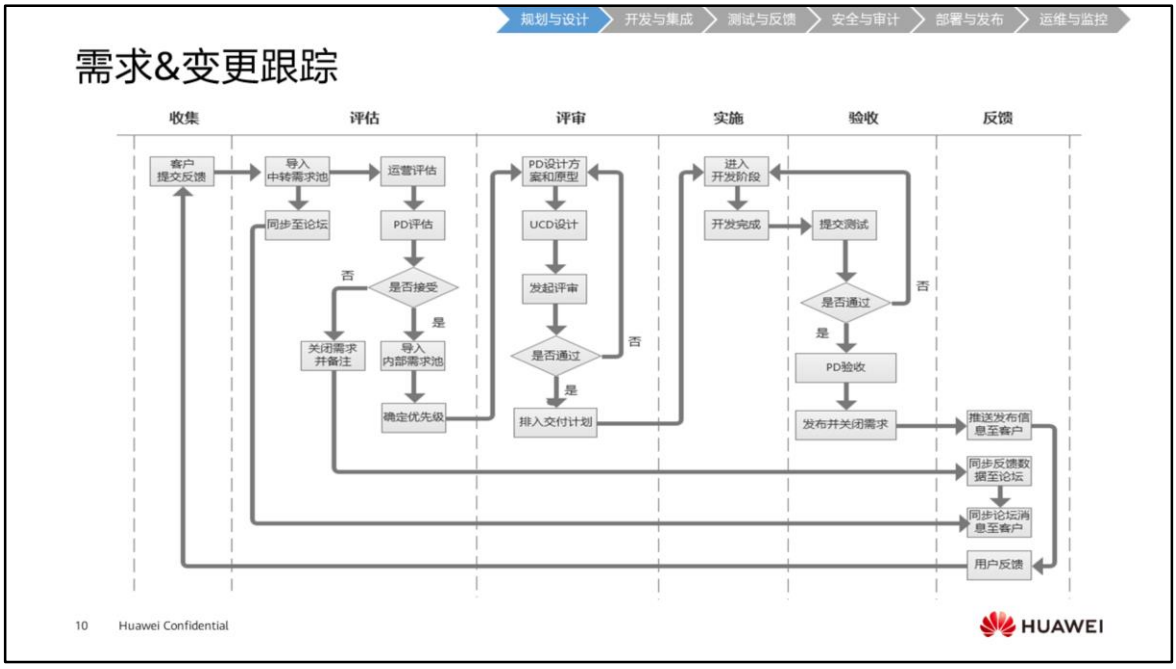
- 我们从2015年开始，首次上线3个服务，经过2016~2017年的发展，上线了8大服务，到目前为止已经超过15个服务。
- 华为云DevCloud真正实现了全流程覆盖，从研发上云到业务上云All In One，全流程DevOps平台。



- 企业数字化转型是一个复杂的系统工程，涉及到研发流程、研发模式、商业模式、组织和文化的变化，不是一蹴而就的事情。要通过小步快跑、持续迭代、平滑演进、持续改进，这种方式来推进。这是我们认为最成熟和稳健的途径。
- DevCloud：研发上云的故事。
- ServiceStage：业务上云的故事。

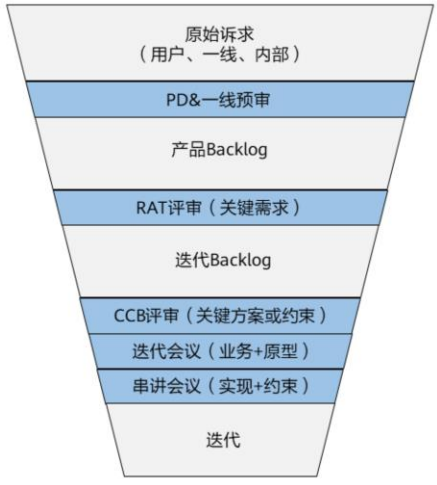


- 建立从代码提交至上线生产的全流程自动化，各级门禁控制质量，高效高质量交付。



- DevCloud所有需求纳入RAT需求管理，统一分析跟踪闭环，各角色能力协作，确保需求交付无遗漏。体现端到端跟踪，RAT包含所有需求管理（如何保证devops下有序跟踪）。

需求流动过程管理



需求优先级排序：业务优先级

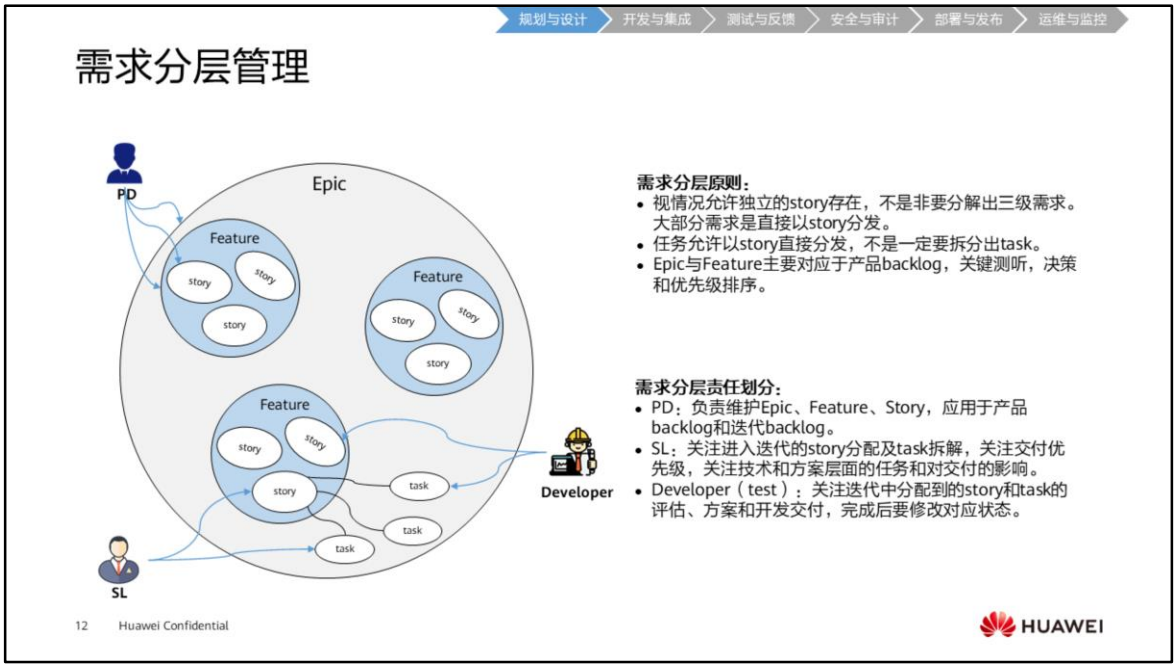
- 问题>需求
- 解决方案需求>产品需求
- 跨服务依赖需求>独立需求
- 清晰需求>不清晰需求
- UCD完成需求>UCD等待需求
- 80%功能需求+20%体验需求
- 价值用户需求>普通用户需求
- 通用需求>定制需求

趟过的坑：

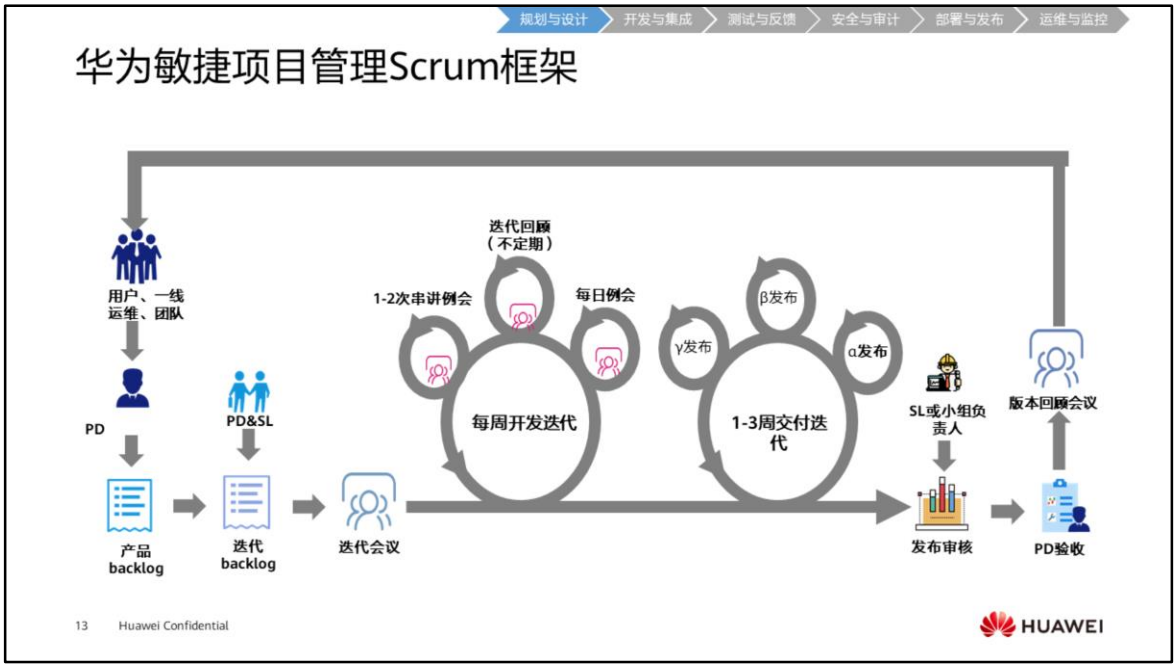
- 严格按照三层模型拆分，工作量大；
- 全部只用story管理，任务无法承载，交付吞吐率无法度量。
- 前后台任务拆分/不拆分
- 需求依赖未标明和识别
- 详细完整的story方案
- UCD任务未纳入管理导致等待

HUAWEI

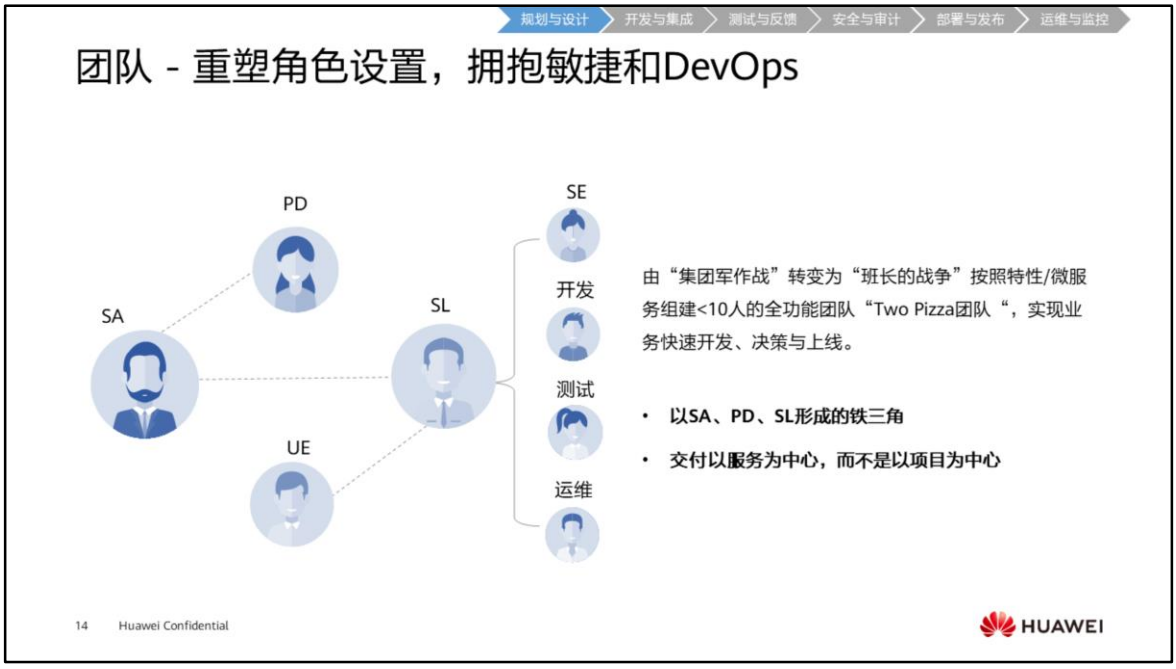
- 漏斗形需求通道消除浪费和风险。



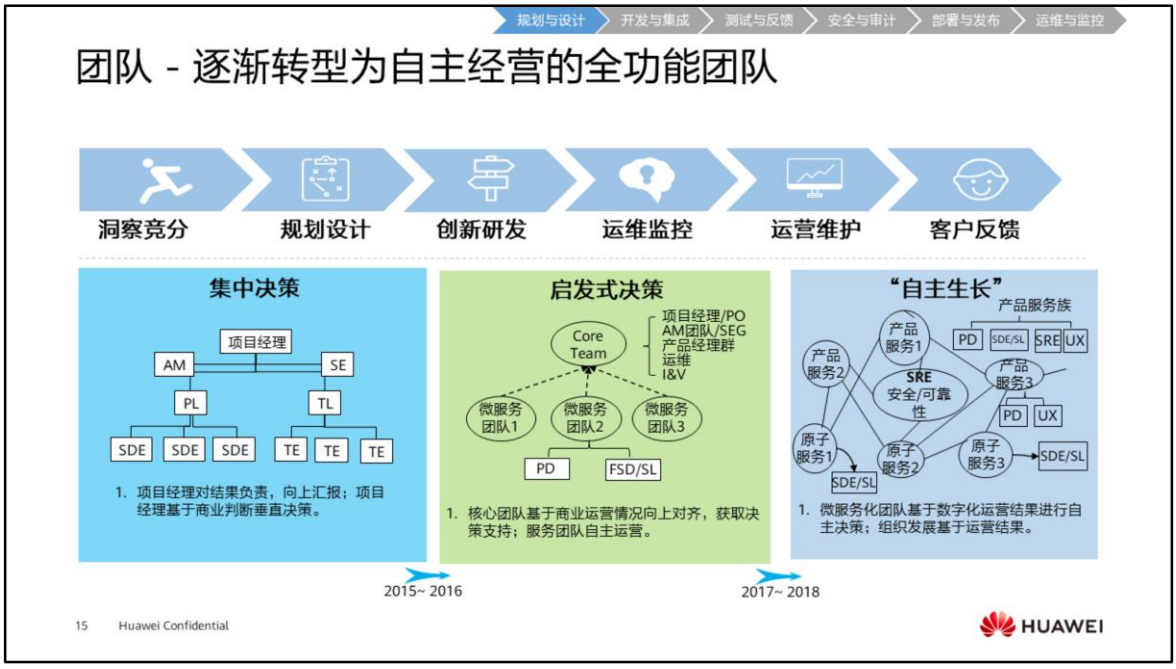
- Story拆分INVEST原则：
  - Independent（解耦）：尽可能拆分独立的用户故事，尽量避免依赖。
  - Negotiable（易交流）：使用用户语言描述，让用户、产品、开发、测试各角色方便理解交流。
  - Valuable（传递价值）：用户故事要关注带来的用户价值或解决的问题，而不仅仅是功能。
  - Estimable（易估算）：足够小和清晰让开发、测试都容易评估工作量、优先级。
  - Small（足够小）：用户故事要在1个迭代内可以被完 成，有助于小批量开发和交付和响应变化。
  - Testable（可测）：用户故事应该包含一个可被测试和验收的小场景，否则无法确认其完成时间。



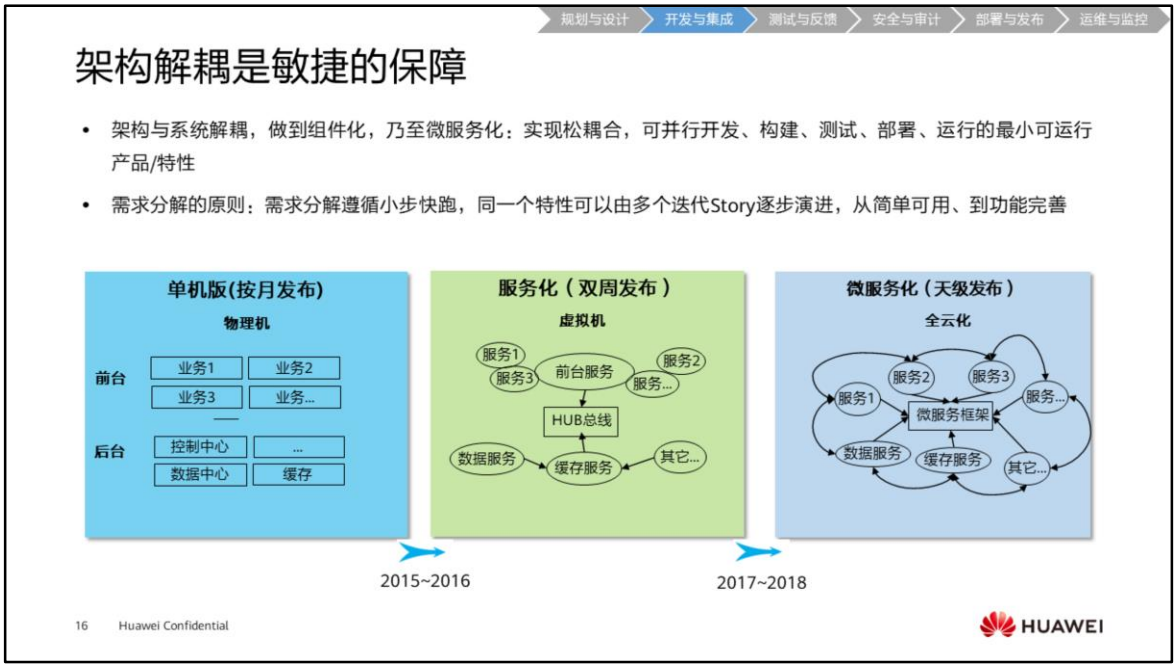
- 华为敏捷项目管理流程，基于Scrum框架进行交付侧扩展。
- 融合了DevOps的新型敏捷模式强调持续交付CD（包含持续构建、测试、以及持续部署、发布、反馈），加速了产品推向真实用户，并及时获得海量反馈的决策数据源，比仅仅依靠PO或者少数粉丝用户反馈的决策准确率大为提升。



- 团队 - 重塑角色设置，实现快速自我决策
  - Sponsor：干系人（项目/业务/服务）
  - SA：负责客户场景及解决方案
  - PD：Product Owner，负责产品规划、设计、分析
  - UE：UCD(User Centered Design)工程师，负责用户研究、交互设计、美工、视觉
  - SL：微服务/特性经理Service Leader，兼任敏捷Scrum Master，带领团队进行开发
  - SE：技术leader，系统工程师，负责架构、系统设计
  - 开发：负责代码实现
  - 测试：负责测试验证
  - 运维：负责部署、发布、运维、监控

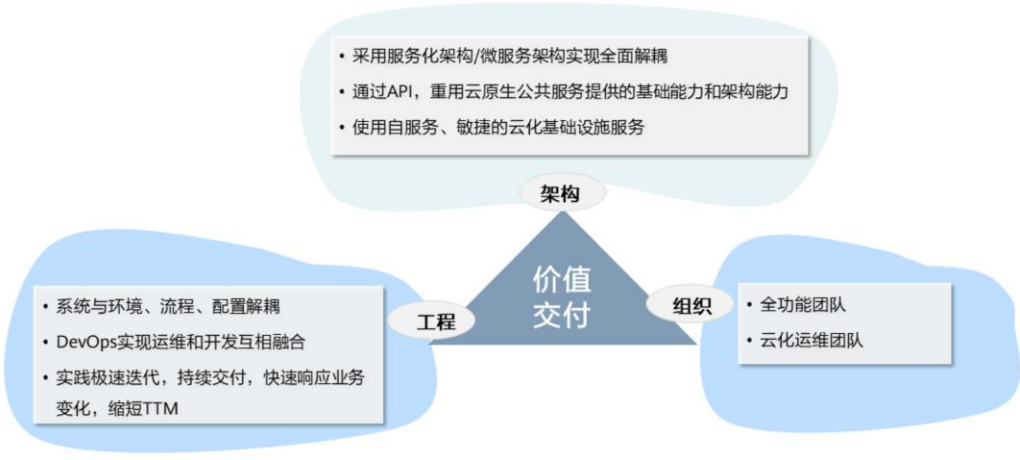


- 服务的全生命周期团队：逐渐转型为自主经营的全功能团队。对特性/部件/服务，完整的实施规划/需求/设计/开发/测试并独立部署、交付、运维（DevOps场景）的项目型团队。
- 面向云服务/微服务的架构，向敏捷/DevOps全功能团队转型。
- 在敏捷转型基础上，进一步横向整合从业务-研发-运营的团队和全流程。
- 组织：服务/微服务全功能团队，产品经理负责制。
- 流程：全流程整合，高度自动化，一键发布。



- 架构解耦，最小可行产品是敏捷的保障
- 由横向分层的大系统 ->纵向解耦的小系统演进
- 各个微服务/特性，可由独立团队并行开发交付
- 阶段一：传统架构及瀑布试研发模式，发布上线耗时1个月。
- 阶段二：服务化改造及使用云化产品交付模式，交付周期缩短为1-2周。
- 阶段三：微服务版本独立发布模式，配合端到端的独立交付流水线，实现天级N次上线能力。

DevCloud能力构建

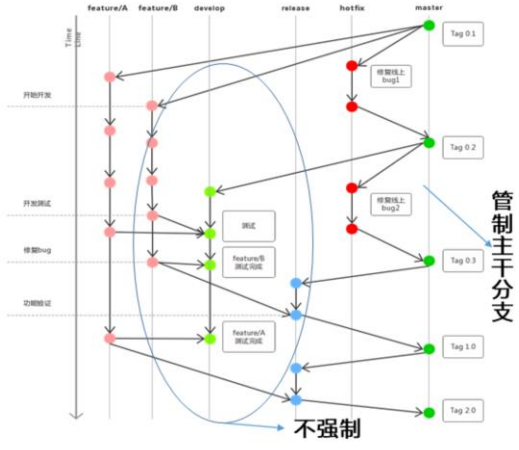


HUAWEI



- 微服务开发：应用CodeHub进行代码托管，合并请求，在线检视，入库前强制检视审核，代码质量明显提高，有效拦截部分编码类问题。

分支管理



特点：

1）兼顾了 TrunkBased 的“易于持续集成”和 GitFlow 的“易于管理需求”特点

2）测试/发布分支的特性组成是动态的，调整起来比较容易

3）最终合入主干的代码经过了功能验证和评审的，能有效减少生产出错的概率

4）分支明确，不会混乱

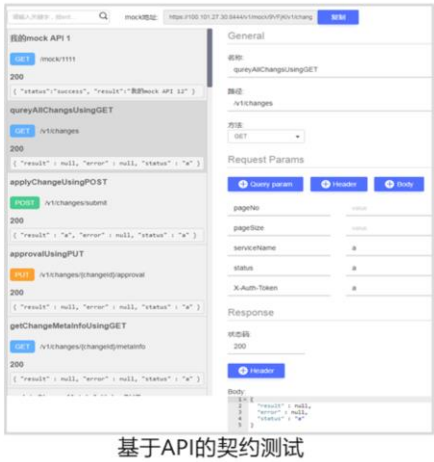
HUAWEI

- 分支管理：一个项目里划分不同的分支，来实现功能开发、bug修复、版本发布，以及开发过程中的冲突处理。

微服务API设计

API设计实现的功能：

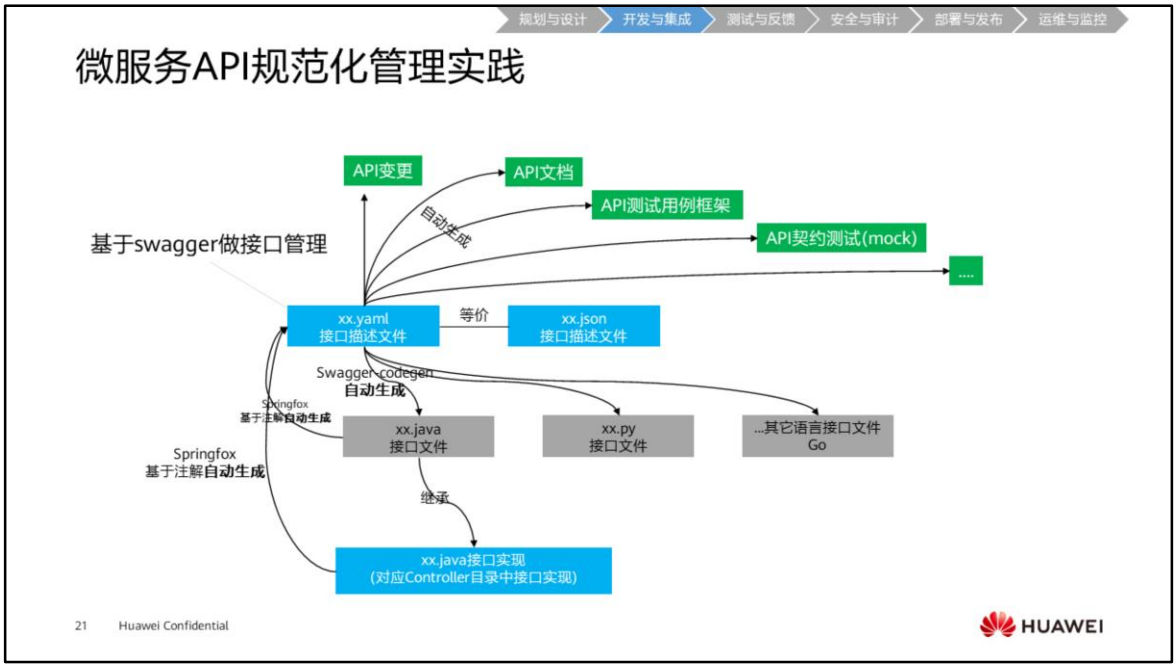
- API在线编辑与评审
- API可视化列表
- API文档自动化生成，与代码关联
- API变更管理，减少相互依赖问题
- 基于API的契约测试



基于API的契约测试

HUAWEI

- 微服务API设计：可通过APIManager进行接口在在线设计与评审，生成在线接口文档，自动识别接口变更，借助基于API的契约测试加强团队协作效率，减少依赖问题。



- API规范化：遵循Swagger2.0规范，基于Swagger的API管理，后续基于接口描述生成文档、用例框架、调测桩等。

DevCloud敏捷测试实践

- 构建CloudNative测试能力，实现测试活动解耦，在线业务测试与监控

分层流水线

通过个人级，服务级，产品级分层自动化流水线来分层确保质量
提升功能/安全/性能/可靠性自动化测试覆盖情况
利用自动化技术，提升用例执行效率与稳定性，监控反馈流水线各测试环节执行效率

质量门禁

不合格的制品无法流入下一环节
让流水线来拦截问题，让自动化门禁代替人工管理
分层门禁，规则自定

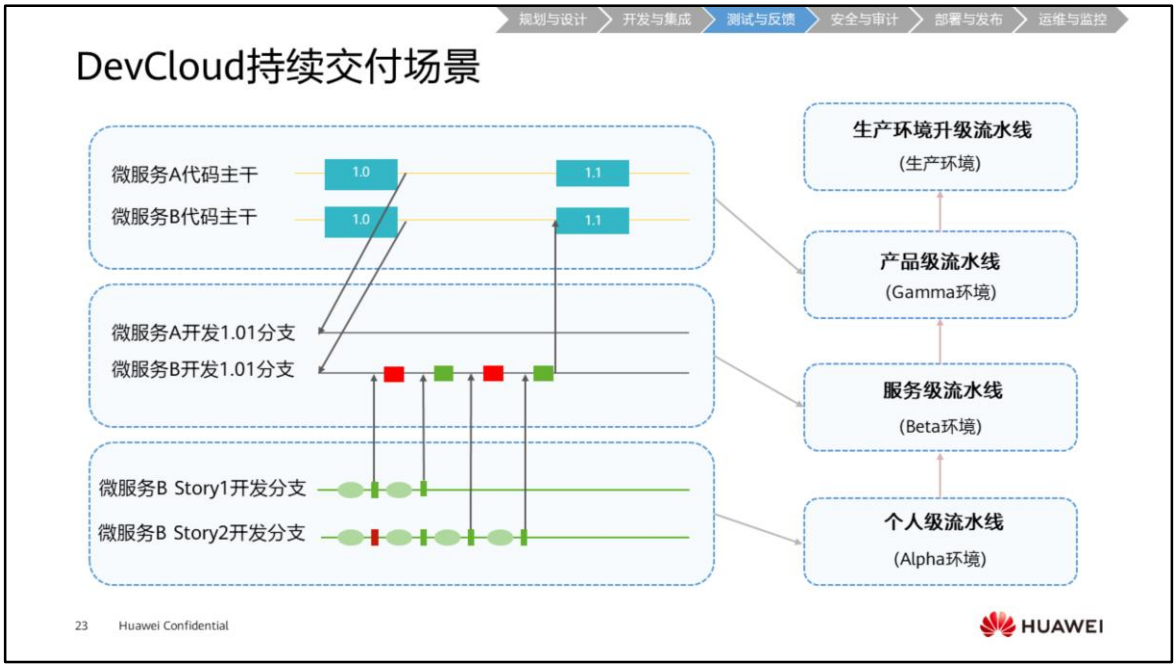
在线测试

在线业务测试，对生产环境业务主线与重点功能持续在线拨测，先于用户发现业务异常并告警
在线体验巡检，每日生产环境巡检E2E主线场景体验
在线7x24监控，错误事务和运行异常对比分析告警推送

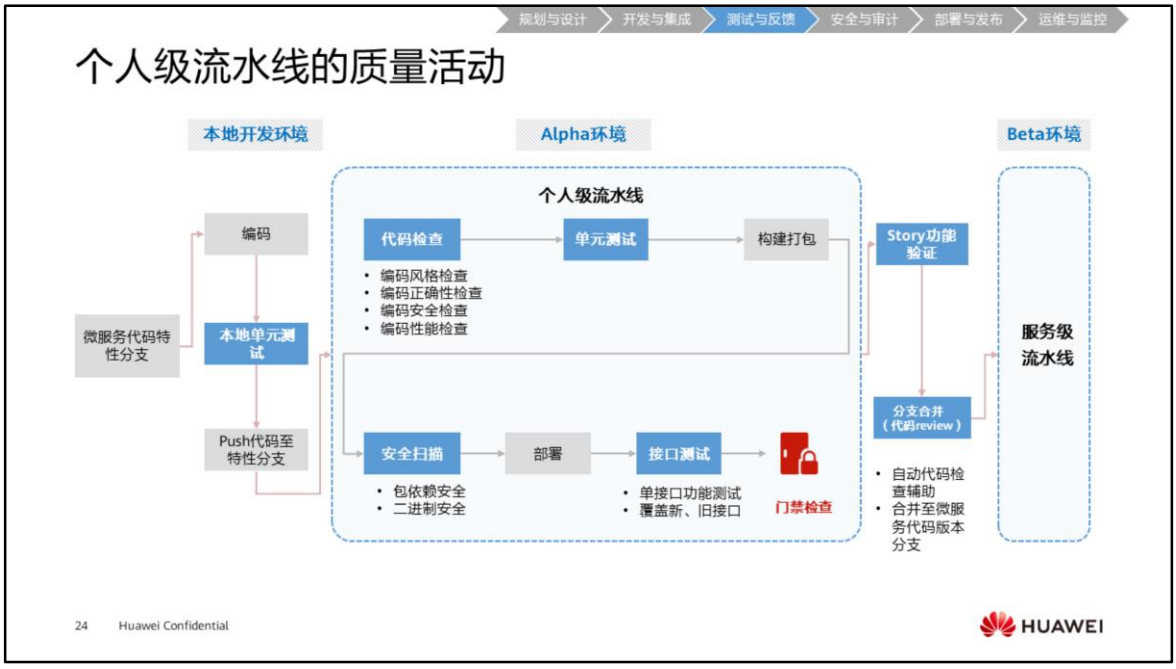
可视化

测试过程可视化，让质量暴露在阳光下
团队所有成员知道项目的质量，促进团队成功

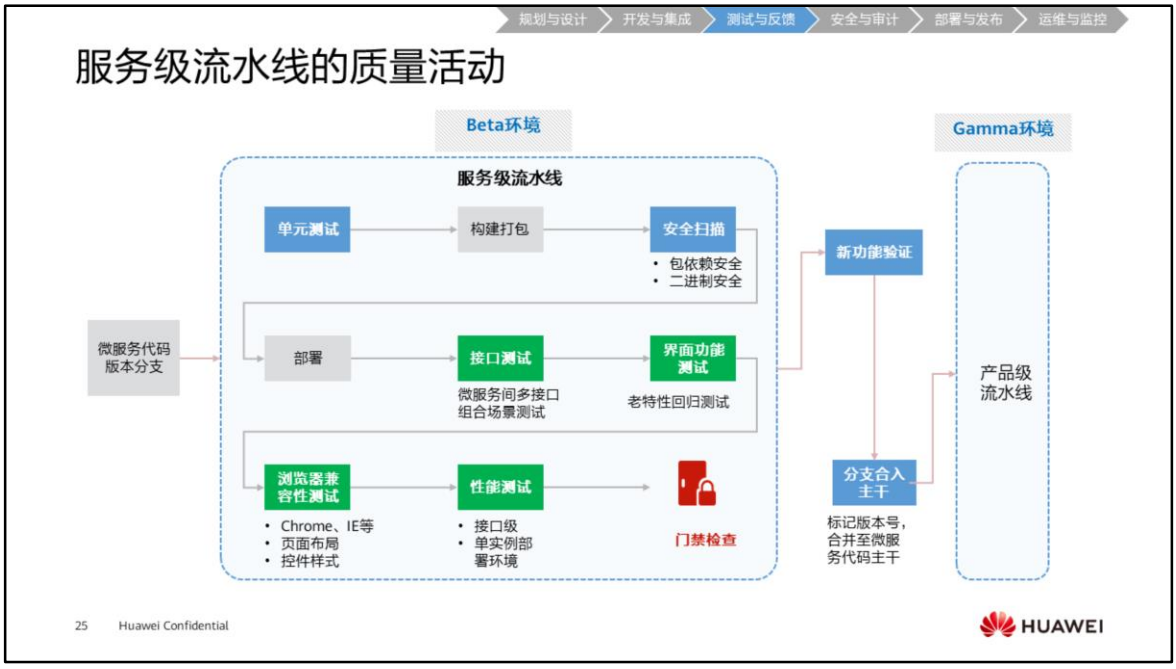
HUAWEI



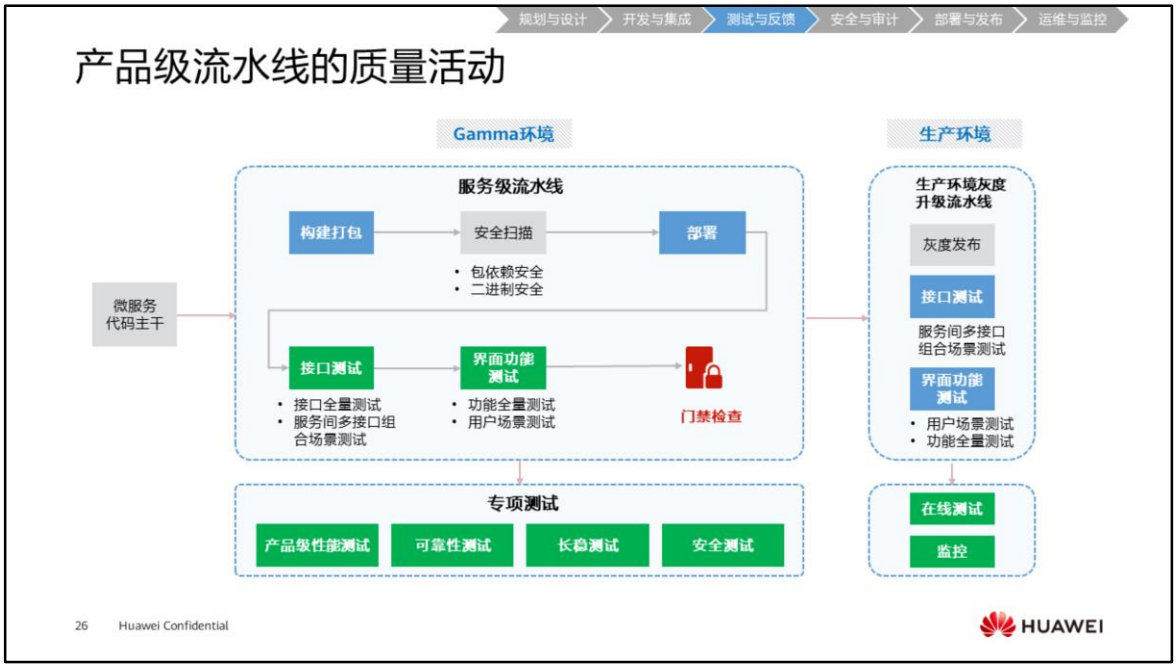
- DevCloud 持续交付代码分支、流水线、环境对应关系



- 个人级流水线从开发、alpha环境至beta环境，对应微服务个人或特性分支。
- 代码检查门禁：
  - 指定检查规则的问题数为0
- 单元测试门禁：
  - 方法覆盖率达标
  - 行覆盖率达标
- 安全扫描门禁：
  - 问题数为0
- 接口测试门禁：
  - 测试通过率达标
  - 测试覆盖率达标



- 服务级流水线从alpha环境至beta环境，对应微服务版本分支。
- 单元测试门禁：
  - 方法覆盖率达标
  - 行覆盖率达标
- 安全扫描门禁：
  - 问题数为0
- 接口测试门禁：
  - 测试通过率达标
  - 测试覆盖率达标
- 浏览器兼容性测试门禁：
  - 问题数为0
- 性能测试门禁：
  - TPS、RT达标



- 产品级流水线从gamma环境至生产环境，对应微服务版本主干。

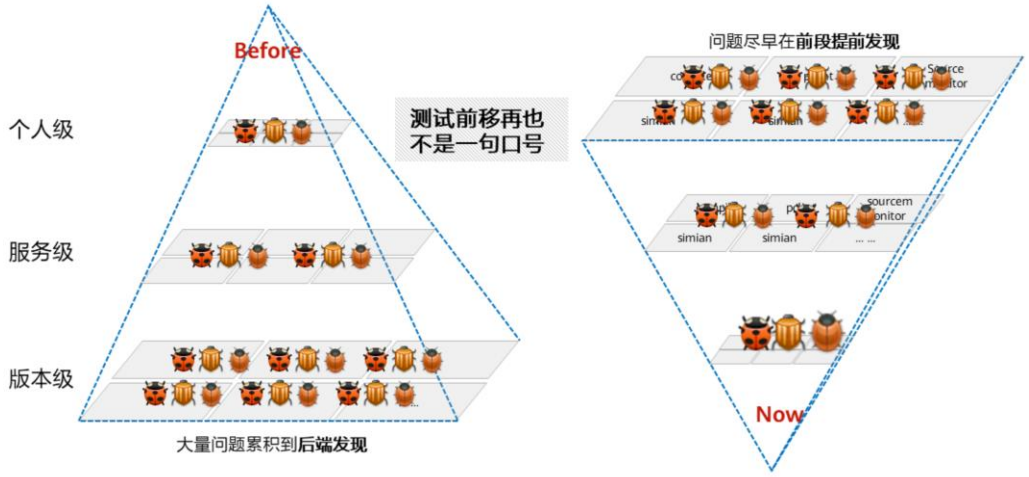


- 自动化代码检查，提前构筑质量，避免后期修复带来高成本。



- 质量门禁自动检查流水线产出物的DoD(Definition of Done)达标情况。

测试环节效果对比



HUAWEI

规划与设计 开发与集成 测试与反馈 安全与审计 部署与发布 运维与监控

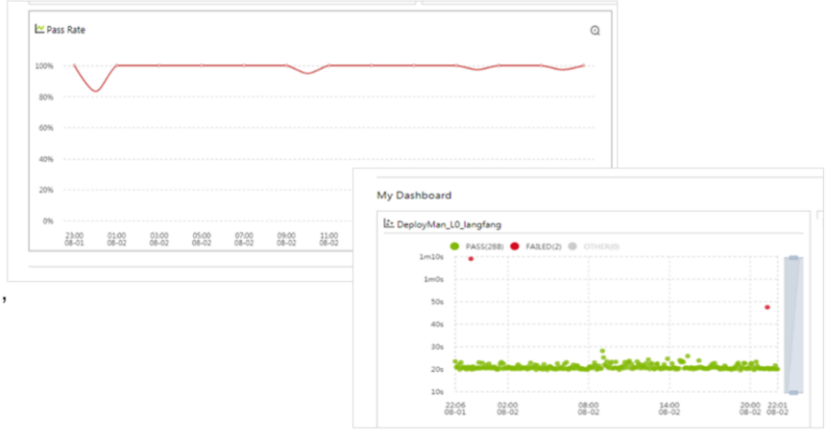
DevCloud在线功能拨测持续监控

- 先于用户发现问题，赢得解决问题的时间，在用户感知之前修复问题。

在线拨测工具：

生产环境7*24小时

模拟用户操作



注：使用华为内部工具，目前未对外商用

HUAWEI

- 先于用户发现问题，赢得解决问题的时间，在用户感知之前修复问题。



- 直观监控服务质量
- 直观监控服务质量：If You Can’t Measure It, You Can’t Manage It - 爱德华.戴明

某产品应用效果总结

- 交付周期（从接纳客户需求到上线）时间从30天提高到2周
- 版本遗留缺陷从平均1.02 到 0.42，下降60%

个人级交付循环



闭环周期：60 15分钟

1、**代码质量门槛**
2、基于变更，增量构建
3、一键式，构建提交、检视
4、20万个LLT用例保障体系
5、时长：**33->8分钟**

服务级交付循环



闭环周期：2小时 30分钟

1、 **小迭代开发**
2、自动化工厂：
15000个机架用例
400万个次执行频率
3、时长：**46->14分钟**

产品版本级交付循环



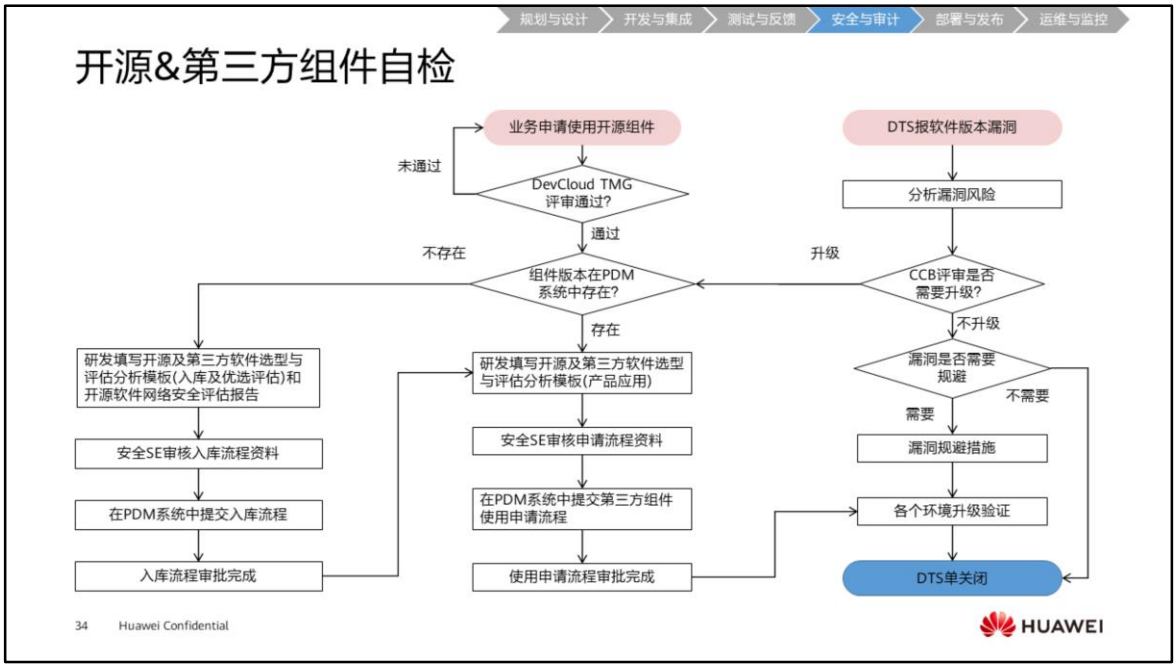
闭环周期：4小时 2小时

1、极速回归，**每日版本转测试**
2、版本70%缺陷前端闭环
3、时长：**118->58分钟**

HUAWEI



- 网络安全： 基于DevOps流水线，全流程保障网络安全，持续提升安全设计、编码、测试、运维能力。
- 联合团队安全组、测装架构部、网络安全能力中心组建DevCloud安全保障小组，公测/商用/送检前重点保障产品安全。



- 开源&第三方组件： 严格遵照先申请后使用要求，每PI开源扫描自检。

开源&第三方组件自检实践

- 开源第三方组件优选率可视化统计及其更新发布周期。

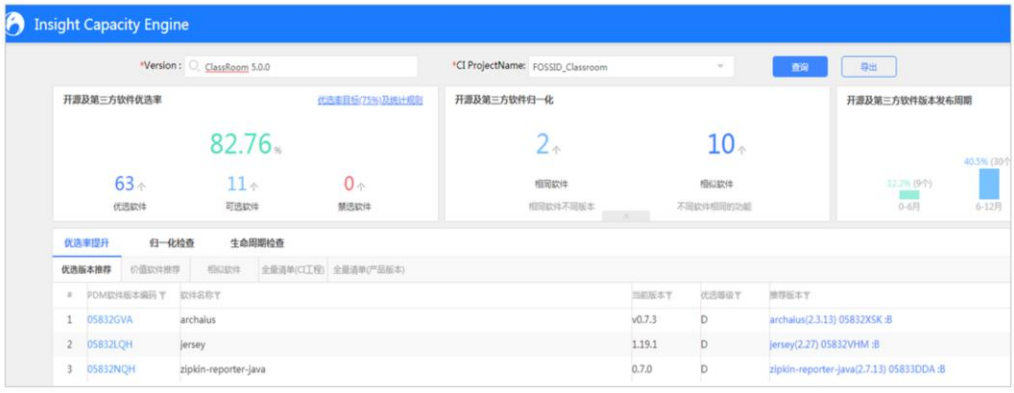


HUAWEI

- 开源&第三方组件： 对齐公有云统一标准，开源第三方组件优选率>=80%
- Codecheck第三发组件扫码总览
- 优选推荐
- 风险统计

开源&第三方组件自检实践

- 开源第三方组件优选率提升点及优选版本推荐

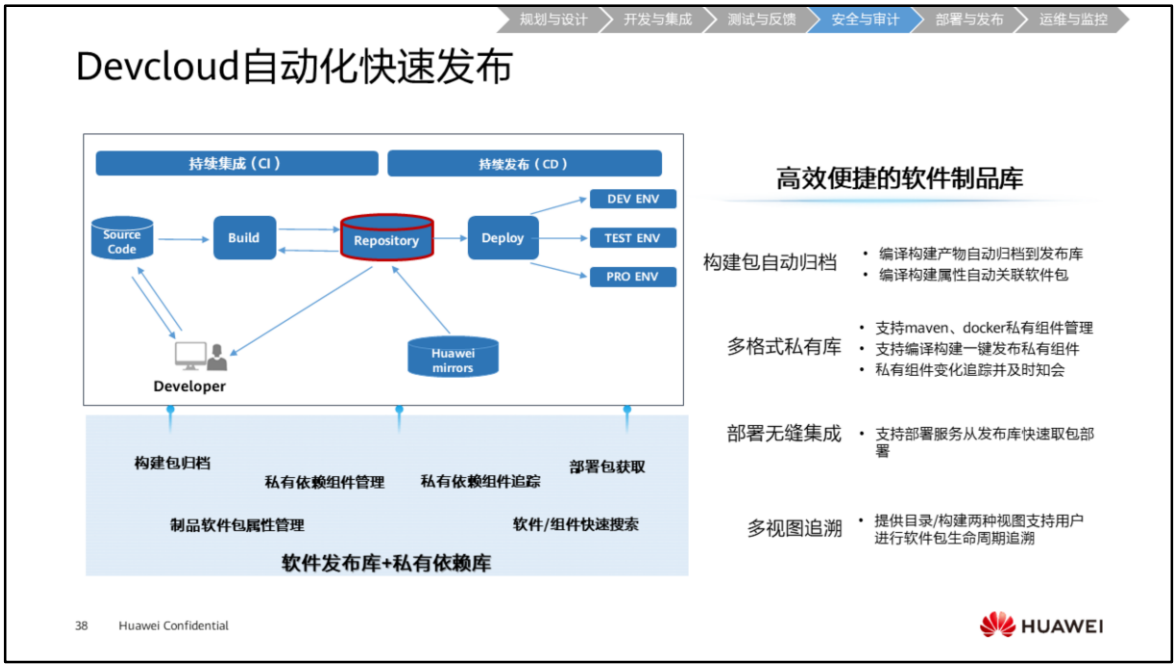


HUAWEI

- 开源&第三方组件： 对齐公有云统一标准，开源第三方组件优选率>=80%
- Codecheck第三发组件扫码总览
- 优选推荐
- 风险统计



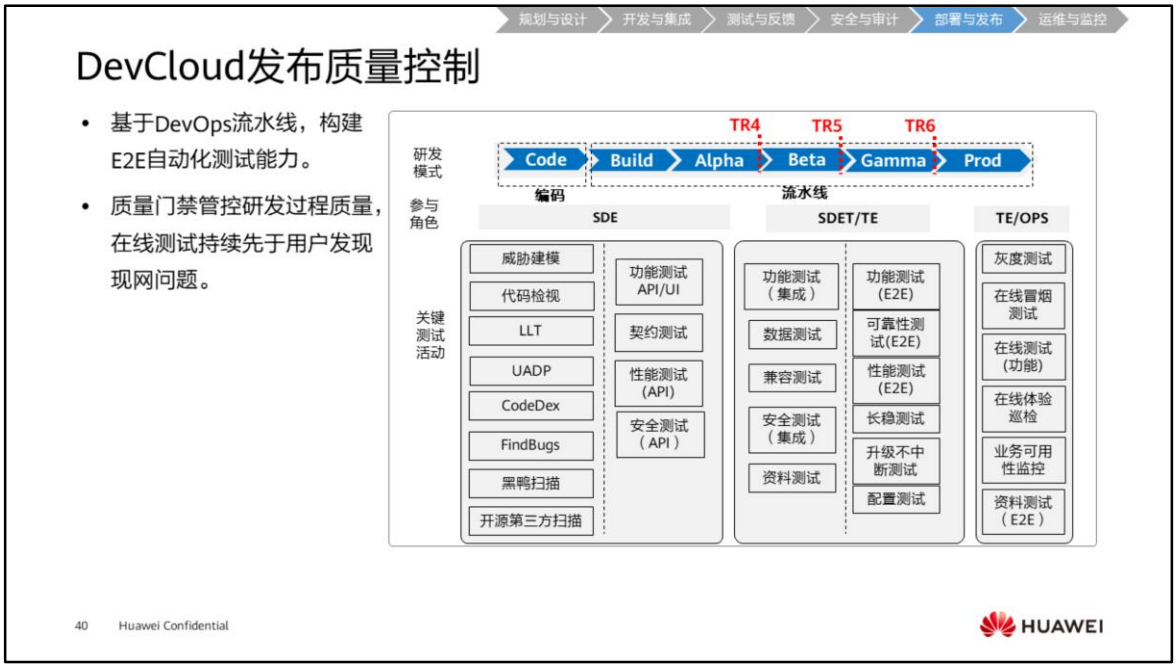
- 部署应用到物理机、虚机、容器
  - 支持将应用部署到物理机，虚拟机，容器，即使是用户私有的物理机，虚拟机，容器集群，只要能连通就能进行部署；并且主机部署还支持以代理机的方式进行部署；容器可支持部署到华为云CCE以及客户的私有K8S的集群
- 支持多种技术栈应用的部署
  - 支持tomcat, springboot, nodejs等多种技术栈应用的部署，提供通用模板比如springboot, tomcat等方便用户创建部署任务，并支持用户自定义模板，提供25+原子步骤组装成部署任务
- 支持与流水线集成打通部署流水线
  - 在流水线能关联部署任务，并可以通过流水线关联构建，代码检查，测试等服务，实现端到端的DevOps流程
- 图片出处：3ms



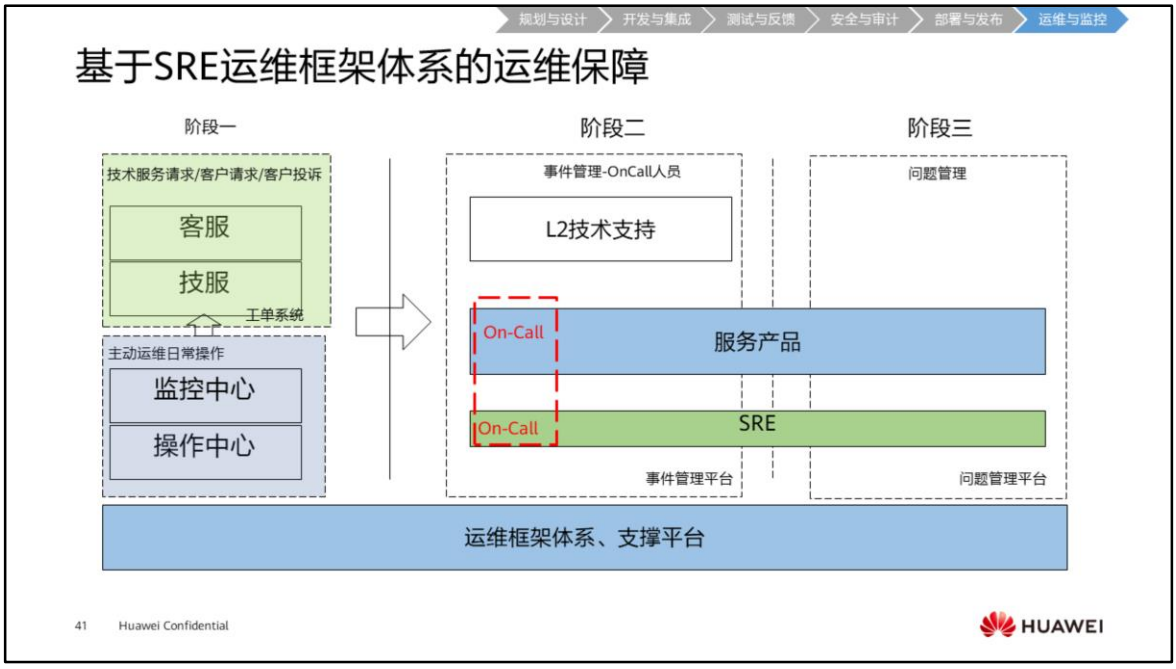
- Devcloud自动化快速发布：软件包资产的可视化管理和追溯
- 流水线：打通从代码提交至服务上线自动化流水线，代码提升至生产上线最快X小时
- 支持产品版本内建出口质量标准，流水线门禁集中管理，提升版本出口质量
- 关键环节的自动化质量门禁策略



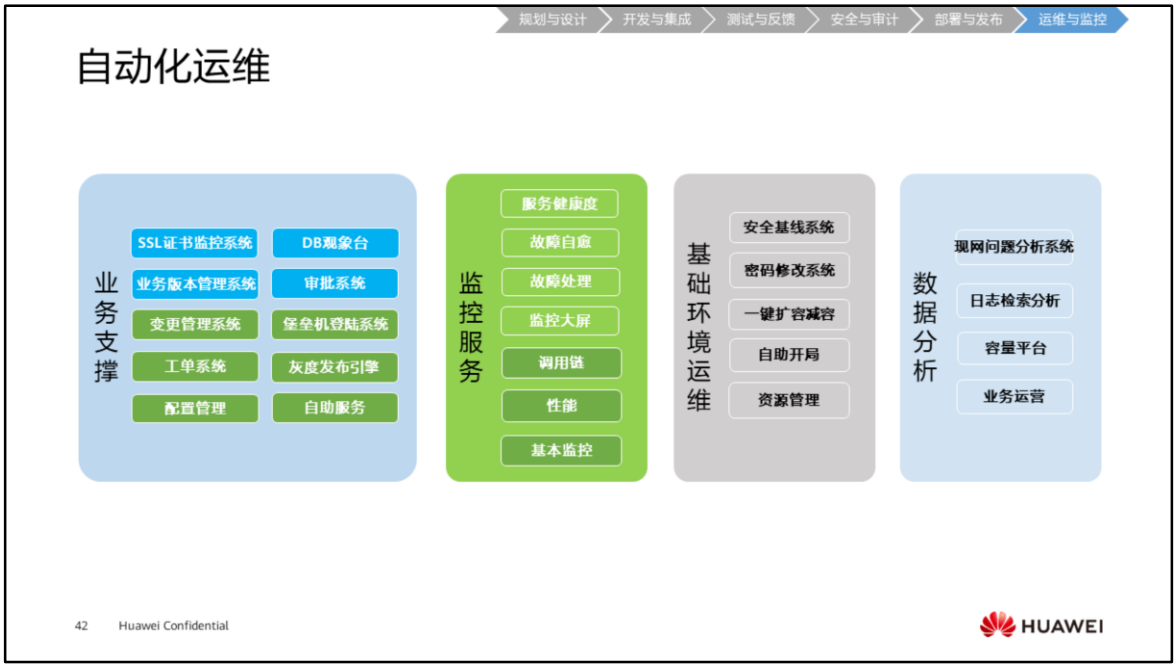
- 迭代计划、发布、以及版本号命名：每周一个迭代，可以多次发布。
- 版本追溯：现网服务节点的版本可见，并可追溯该版本的发布、软件包、构建记录、验证记录、已经代码仓库中的每次提交。



- 测试与质量：基于DevOps流水线，构建E2E自动化测试能力，质量门禁管控研发过程质量，在线测试持续先于用户发现现网问题。



- 运维保障：基于SRE运维框架体系要求，具体化软件开发云一二三线运维支撑体系。
- 原则一：统一 一线
  - 客服、技服、操作中心、监控中心统一；各服务产品不自建，且需要进行培训赋能。
- 原则二：共建二线
  - 服务On-Call接口组人员需要稳定，如需更换，需要获取SRE团队批准。
- 原则三：体系遵从
  - 服务On-Call人员、接口人员、开发人员，如需在云现网进行相关工作，需完全遵守SRE运维框架体系相关规定。



- 自动化运维：引进自动化运维平台，运维作业50%~80%自动化。

自动化运维

- 运维管理平台能力

能力项	详细说明
变更发布	依托devcloud流水线服务，Devops自动化持续发布，灰度变更0中断。
日志管理	公有云ELK日志平台。
监控告警	基础环境监控、性能监控、调用链监控，覆盖面100%
巡检	每日巡检三次，涵盖基础环境、安全、性能，覆盖面100%
拨测	5分钟一次，覆盖所有服务L0级用例，在线实时短信、邮件告警。
工单管理	公有云Cloud Incident MGMT工单平台
资源管理	普罗米修斯、zeus运维平台

HUAWEI

- 自动化运维：引进自动化运维平台，运维作业50%~80%自动化。

目录

1. 华为云端到端DevOps实践
2. **DevOps转型路径**

HUAWEI

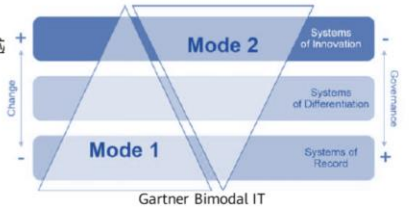
总体概览：DevOps实施阶段



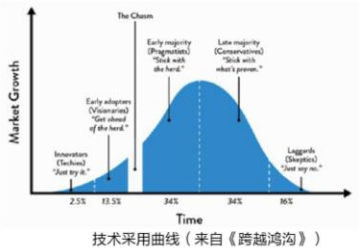
HUAWEI

准备阶段1：选择合适的试点项目

- **绿地项目与棕地项目**
  - 改进棕地项目时，不但要努力降低复杂性，提高可靠性和稳定性，而且还应该更快，更安全，更容易变更；团队可能更愿意尝试。
  - 两者皆有可能
- **兼顾在SoR记录型系统和SoE交互型系统**
  - Gartner双模IT：SOR侧重于做得正确，SOI侧重于做得快速。
  - 高绩效组织能够兼顾高水平的生产能力和可靠性，每一位客户都应该同时享有速度和质量。
  - 两者皆有可能
- **从最有同理心和最乐于创新的团队开始，扩大DevOps的范围**
  - 发现创新者和早期采用者
  - 赢得沉默的大多数，识别“钉子户”
  - 尽早展示成果并积极宣传，将大目标分解成渐进式的小步骤



Gartner Bimodal IT



技术采用曲线（来自《跨越鸿沟》）

HUAWEI

- 在实施团队DevOps的过程中要先选择合适的试点项目，选择的标准可以参考以上三个方面。
- 绿地项目和棕地项目最初用于描述城市规划和建设项目。2014年DevOps企业峰会上分享的转型案例中，棕地项目所占的比例超过60%。
  - Green field: 大部分是试点项目，全新的软件项目，有机会构建全新的应用和基础设施，对已有的代码库、流程和团队没有太多顾虑
  - Brown field: 已服务客户的产品或服务，大部分有很多技术债务，没有自动化测试，紧耦合架构

准备阶段2：组建全功能团队

参考康威定律设计组织结构，使团队安全和独立地工作。

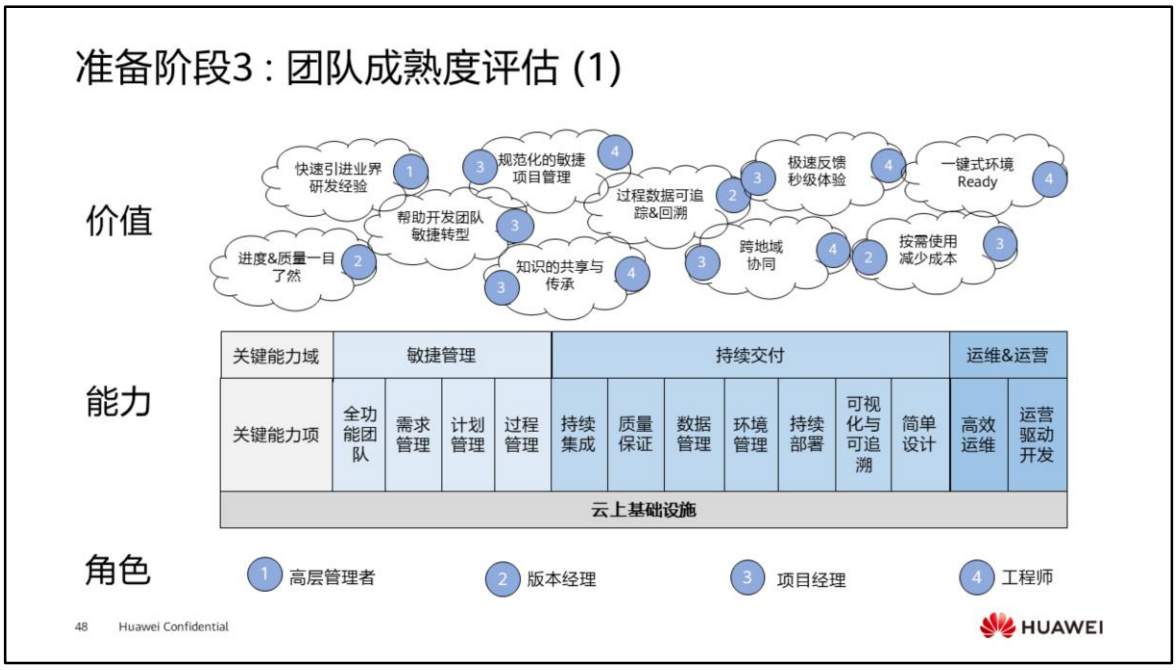
- 软件开发团队的结构对软件产品的架构和成果有巨大的影响。
- 利用康威定律组织团队，减少工作交接次数，提升交付速度和成功率。
- 小团队独立运作，彼此充分解耦，避免过多的沟通与协调。
- 松耦合架构
  - 面向服务架构SOA，由限界上下文的松耦合服务组成
  - 微服务架构
- 两个比萨原则，保持小规模



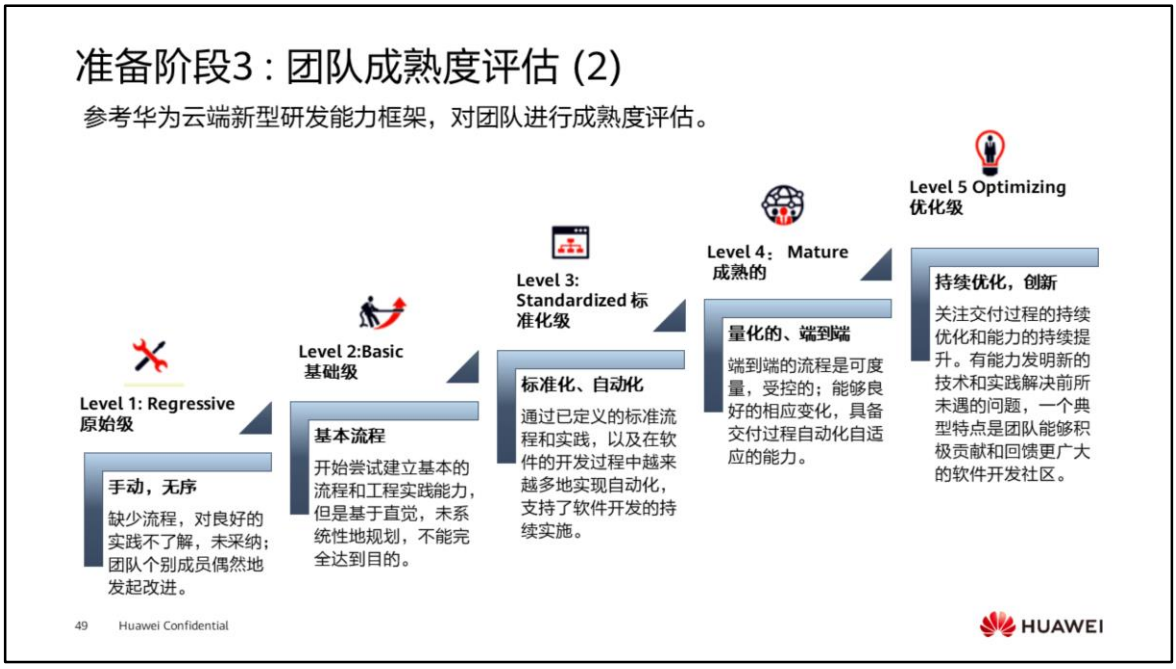
“系统设计受限于组织自身的沟通结构，组织规模越大，灵活性就越差，这种现象也就越明显” – Melvin Conway, 1968

HUAWEI

- 右图：Robert Fernandez博士定义了组织结构类型的3种类型：
- 想要什么样的系统，就搭建什么样的团队，服务的理念：团队之间应该是inter-operate，not integrate。
- 康威指出：系统设计受限于组织自身的沟通结构
  - 第一定律：组织沟通方式会通过系统设计表达出来
  - 第二定律：时间再多一件事情也不可能做得完美，但总有时间做完一件事情
  - 第三定律：线型系统和线型组织架构间有潜在的异质同态特性
  - 第四定律：大的系统组织总是比小系统更倾向于分解



- 参考华为云端新型研发能力框架，对团队进行成熟度评估，实施DevOps前和实施后都要对团队进行成熟度评估。



- 评估之后的结果分为这5个level。

准备阶段3：团队成熟度评估 (3)

Devcloud为企业提供专业评估诊断，对被评估对象在DevOps领域方面的能力或现状进行分析、评判并给出建议。

DevOps成熟度评估[主观评估]
主观评估就是基于主观信息或依据做出判断的评估方式。
问卷调查就是一种最典型的主观评估方式。

VS

用户故事成熟度评估[客观评估]
客观评估是基于客观事实或数据做出判断的评估方式。
基于系统数据进行计算得出结论，就是一种典型的客观评估方式。

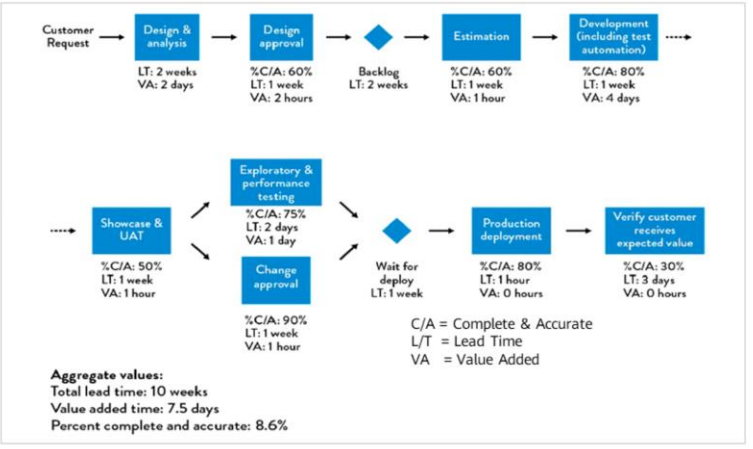




HUAWEI

准备阶段4：绘制价值流图，识别阻碍点

- 召集所有关键成员，绘制价值流图。
- 记录主要的步骤和细节，让相关干系人拥有同样的视图。
- 重点分析和优化，识别阻碍：需要等待数周甚至数月的工作；引发或处理重大返工的工作。
- 确定想要改进的度量指标，通过探索各种假设，然后分析结果，不断迭代和重复，将获得的经验用于下一次实验。



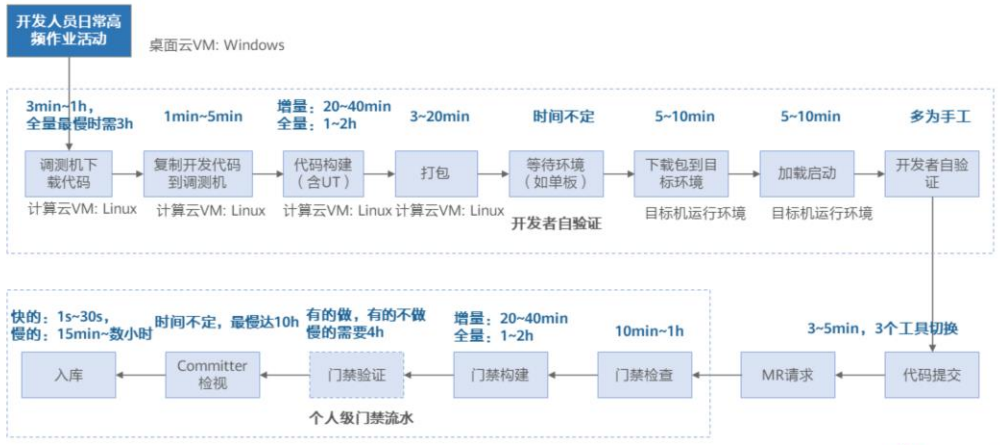
价值流图示例（来源：《精益企业》）

HUAWEI

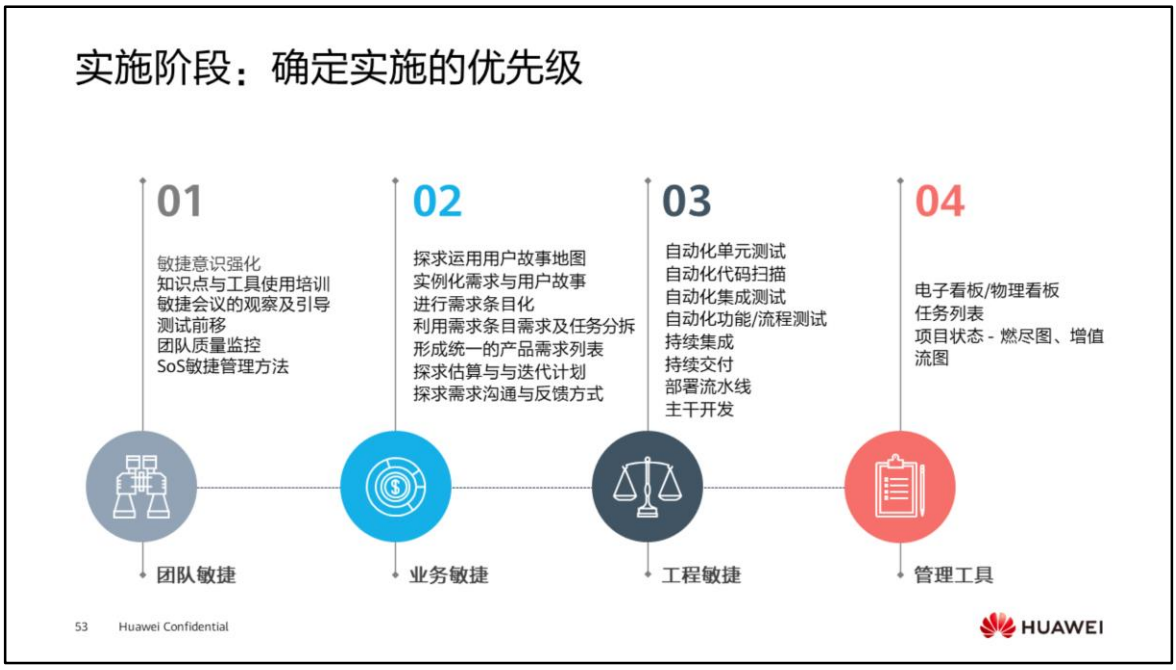
- LT：Lead Time，是一个过程从发起到执行完毕之间间隔的时间。在精益生产中为缩短和预测价值流中的前置时间。可以采用这些方式：减小批量尺寸、减少在制品、避免返工、确保不把次品传递到下游、持续不断的全局优化。
- C/A：Complete & Accurate，返工效能比率，关注返工指标。
- 可视化价值流图要注意一下4点：
  - 端到端的流程映射
  - 识别浪费，低效，瓶颈
  - 确定改进度量指标
  - 创建改进故事板

准备阶段4：价值流分析实例

日常作业活动耗时：最短耗时约73分钟，最长耗时15小时。

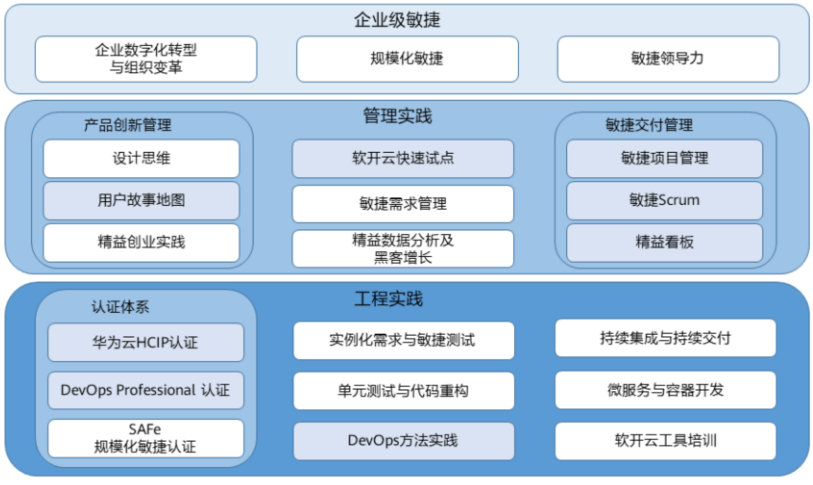


HUAWEI



- 团队敏捷包括：
  - 敏捷意识强化(观察、辅导、约谈等）
  - 知识点与工具使用培训(Scrum、站会、看板等)
  - 敏捷会议的观察及引导（计划、评审、回顾会、需求梳理会、站会等）
  - 测试前移
  - 团队质量监控 (轮值QA等)
  - SoS敏捷管理方法

实施阶段1：基础知识培训、理论储备



HUAWEI

软件研发过程：管理属性和工程属性



管理属性 - 项目规划



工程属性 - 软件交付

DevOps端到端全流程涉及2个维度，分别为管理属性的项目规划，与工程属性的研发交付。

项目规划：
使用DevCloud提供端到端的需求版本管理能力，每个工作项上都可以设置迭代字段代表需求所属的项目规划；而与这一需求相关的任务/测试用例/缺陷/问题等可以相互关联。

软件交付：
某一规划版本中的任务等工件驱动开发人员完成编码后，开发人员可以将代码变更与工作项进行关联；同时DevCloud构建服务会自动生成交付版本号，并将其所包含的代码变更与之关联，因而形成了从“规划版本”到“交付版本”的跟踪能力。同时，使用构建服务还可以保证交付版本完全受控，确保开发/测试/交付版本的一致性。

HUAWEI

- 左侧是Scrum流程简图，右侧是流水线简图。

实施阶段2：推动落地团队级敏捷举措

项目	详细说明
目　　标：	引入Scrum框架流程，打造高效敏捷团队，实现迭代快速交付
参与人员：	顾问、试点团队
主要活动：	整个敏捷改善实施过程，将会运行两个单个周期为2周的迭代，将会穿插： • 敏捷Scrum项目管理培训 • 构建基于ProjectMan的敏捷项目管理流程 • 依据Scrum流程框架，实施具体指导，包括但不限于：敏捷团队组建，基于User Story Mapping的需求整理，任务划分与估算，Product Owner指导，Product Backlog的组织梳理，敏捷发布规划，Sprint计划会议，每日例会，进度跟踪与可视化管理，Sprint演示与回顾； • 在实施指导结束后，进行敏捷实施经验总结与持续改善规划
工 作 量：	20人天
交 付 物：	• 打印版培训材料 • 敏捷流程改善实施附属物 • 定制化的敏捷项目管理流程 • 敏捷项目管理平台
客户职责：	• 保证敏捷改进团队成员积极参与 • 根据实施计划提供相应的资源

HUAWEI

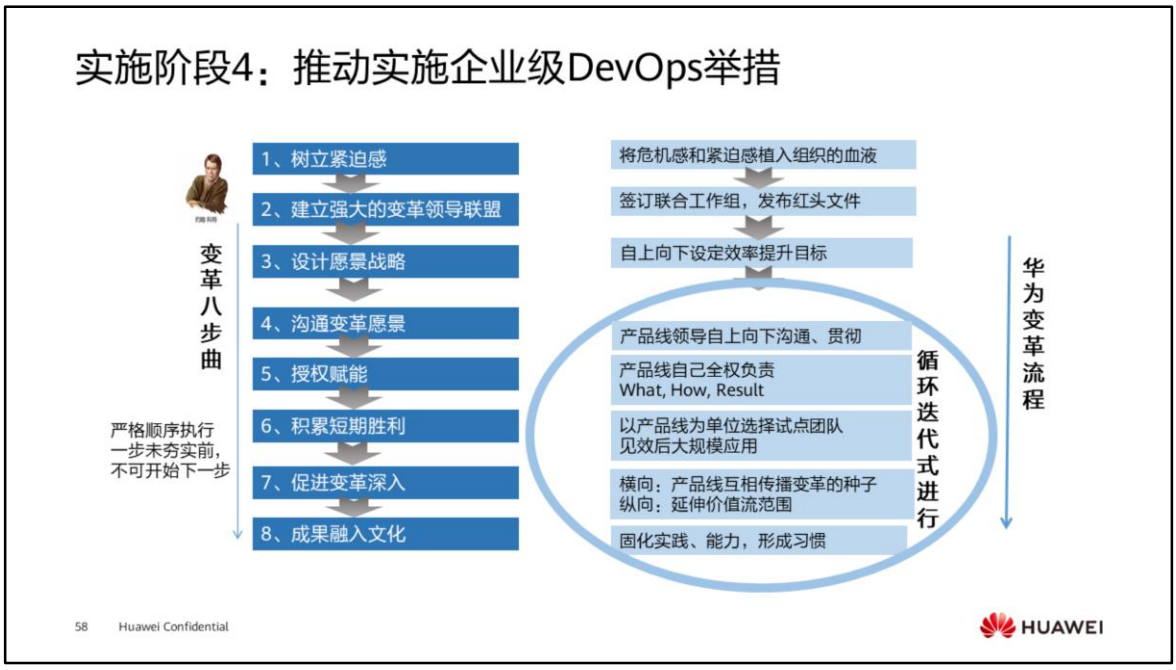
- 团队级Scrum的试点迭代标准包，是指针对一个10人左右的团队，由华为顾问专家通过培训、辅导等方式，帮助该团队理解Scrum，以及在一个2星期的Sprint迭代周期内，应用华为云DevCloud平台进行日常工作的服务包。
- 华为云DevCloud提供团队级Scrum导入实施专家服务

实施阶段3：推动DevOps工程实践举措

项目	详细说明
目　　标：	引入DevOps工程实践，提高代码质量，提高工程效率
参与人员：	顾问、试点团队
主要活动：	在本阶段的改善实施过程中，通过现场指导，将会穿插： • 持续集成与持续交付培训 • 构建基于Codehub的主干分支模式 • 搭建持续集成环境； • 搭建测试环境：开发人员自测环境、验收测试环境； • 实现代码的自动化构建、每日构建、自动化测试、自动化部署、代码的自动扫描； • 在实施指导结束后，将会进行实施经验总结与后续改进规划
工 作 量：	20人天
交 付 物：	• 打印版培训材料 • 各种可工作的工程实践环境及实施附属物 • 持续集成与持续交付平台
客户职责：	• 保证敏捷改进团队成员积极参与 • 专人配合，搭建各种IT设施与环境 • 根据实施计划提供相应的资源与工具

HUAWEI

- 基于客户软件产品生产交付的现状，设计基于华为云DevCloud的DevOps建设方案，便于现有团队迁移到DevOps平台上。
- 华为云Devcloud提供团队级DevOps实施专家服务



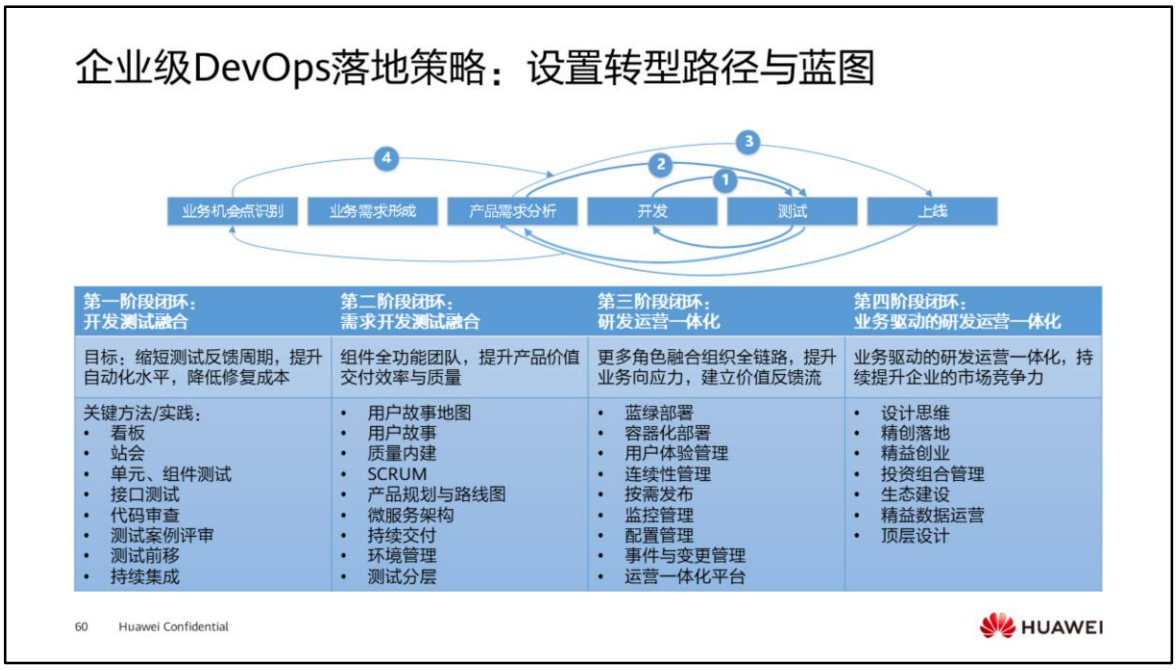
- 在DevOps转型成功的路上，实现企业级敏捷是转型成功的最理想状态。
- 企业规模化DevOps转型怎么实现：我们采用变革八步曲结合华为变革流程，通过多次循环迭代，最终完成企业DevOps转型。

企业级DevOps落地策略：组建专门的转型团队

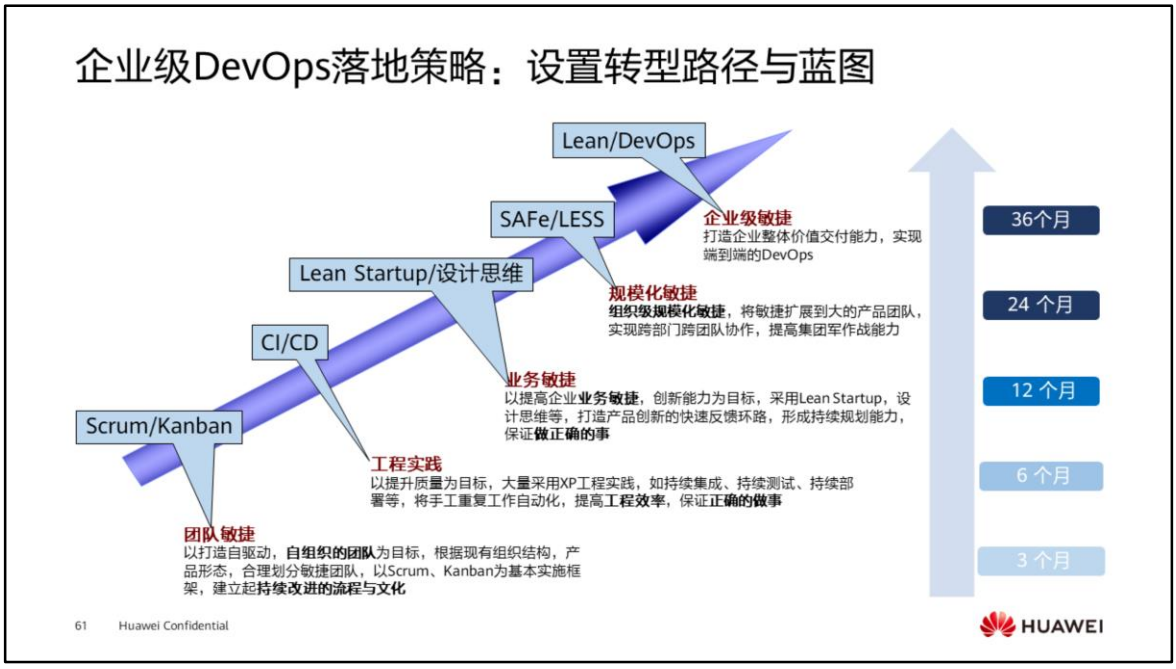
分组（简称LAMP）	主要职责	原则与数量
领导组（Lead）	• 明确转型目标与方向 • 定期了解进展、问题、风险并给予反馈 • 提供组织级资源支持 • 全行宣贯、打造敏捷文化	全力推进组织级敏捷与DevOps转型，保持稳定，1位总体负责人，3-5人
管理组（Management）	• 根据整体转型策略，跟踪进展、移除障碍、控制风险，推动目标实现 • 推动与带领团队参与转型，身先士卒 • 向领导组定期汇报工作	高效沟通，密切协作，根据转型里程碑、试点团队来源与状态调整每个阶段核心成员，4-6人
执行组（Amplifier）	• 与管理组紧密配合，对齐转型目标与关键举措 • 日常工作中积极推进敏捷与DevOps实践落地 • 传播敏捷思想和DevOps实践、发展他人 • 同步和反馈信息	在转型过程中不断学习与成长，相互分享，不定期沟通，不断发展壮大，人数待定
平台组（Platform）	• 根据转型规划，完成DevOps工具链、持续交付流水线等基础设施的搭建 • 架构设计与把控、平台部署与维护、提供相应资源 • 及时响应一线团队使用平台的需求与问题	敏捷运作，启动阶段建议全职，后续阶段可以根据情况兼职，15-20人

HUAWEI

- 来源：ThoughtWorks
- DevOps转型面临的一个固有挑战：与公司当前的业务发生冲突，公司仍采取各种措施延续和保护当前的运作流程。为了适应市场变化，需要改变工作方式，需要组建转型团队以确保转型过程中从管理层到基层团队目标一致、沟通高效、响应快速，从而有效实现组织转型目标。其关键点是确保组织管理层深度参与。

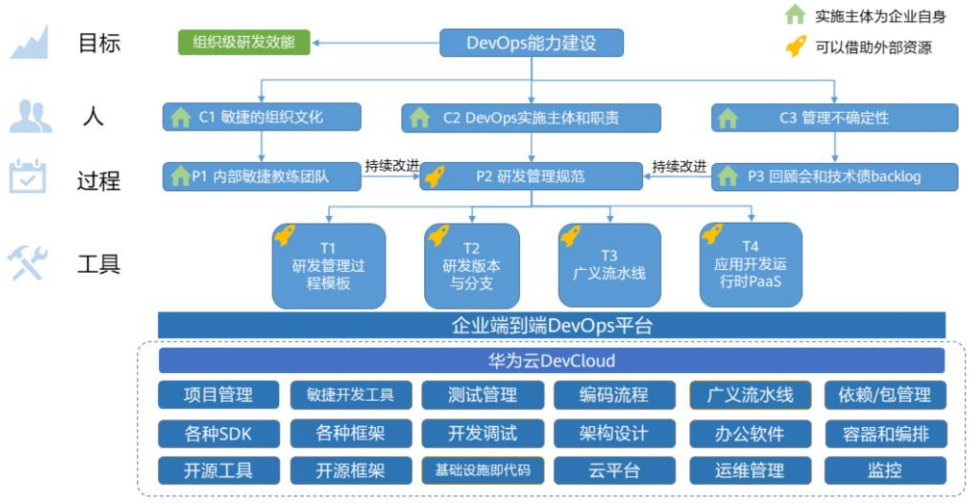


- 第一阶段闭环：开发测试融合，组建研发部门内部的跨职能团队，提升自动化水平，降低修复成本；
- 第二阶段闭环：需求开发测试融合，将产品、研发、测试等角色融合，组建跨职能团队，提升产品交付价值与质量；
- 第三阶段闭环：研发运营一体化，实施产品自运营、自运维，打破了市场、研发、运维部门之间的壁垒，更多角色融入交付链路，提升业务响应力，建立价值反馈流；
- 第四阶段闭环：目标是逐步实现所有业务线都以跨职能团队为最小组织单元，实现业务敏捷性，持续提升企业的市场竞争力。



- 转型从团队级开始，向项目级，版本级和企业级过渡。
- 转型应分阶段，分优先级。
- 转型过程中，需要在组织内部建立一个行之有效的管理组织和运作方式，相关等部门配合做好敏捷文化宣传建设工作。
- 大型企业转型必须站在组织级高度，考虑的不仅是当年的目标，而是一个3到5年，甚至更长久的发展规划。

回顾总结：DevOps转型整体方案



HUAWEI

思考题

1.（单选题）以下对于Two-Pizza 团队规模说法正确的是?

A. 团队一般大于15人小于20人

B. 团队人数为两张披萨切成的块数

C. 两个披萨能让队员吃饱的人数

D. 做两个披萨需要的人数

HUAWEI

- 答案：C

本章总结

学习完本课程，您已经系统性地掌握了DevOps的管理体系、最佳实践、三步工作法的具体落地方法，尤其贯穿课程中的华为云Devcloud的真实操作案例，相信您已经切身感受到这些优秀方法与实践应用的全过程、所解决的问题以及产生的价值，您已经快速成长为实践DevOps的优秀开发者。

HUAWEI

更多信息

- 华为云官网：
  - https://www.huaweicloud.com/
- 华为云软件开发平台DevCloud：
  - https://www.huaweicloud.com/devcloud/
- 华为云帮助中心：
  - https://support.huaweicloud.com/
- 华为云学院：
  - https://edu.huaweicloud.com/

HUAWEI

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and organization for a fully connected, intelligent world.

Copyright©2020 Huawei Technologies Co., Ltd.
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

