软件系统分析与设计

谭火彬，潘海侠

北京航空航天大學
SCHOOL OF SOFTWARE BEIHANG UNIVERSITY 软件学院

设计后……

Chapter 11

内容安排

- 从设计到实现
- 构造软件系统
- 部署和上线运行
- 上线后运维
- 基于华为云的DevOps开发

内容安排

- 从设计到实现
- 构造软件系统
- 部署和上线运行
- 上线后运维
- 基于华为云的DevOps开发

设计模型与代码实现

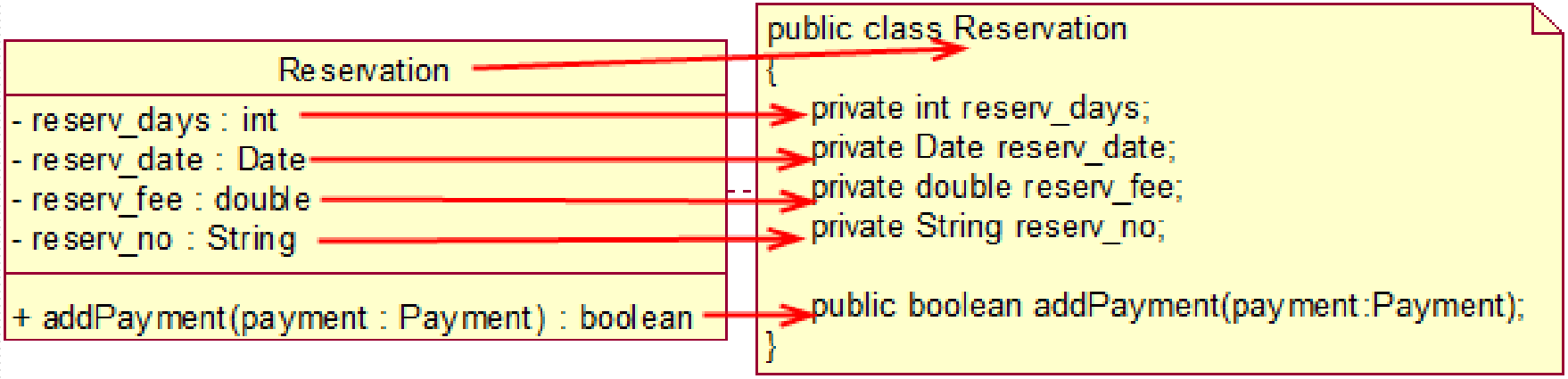
- 编写代码形成软件
  - 软件最终需要代码来实现
  - 模型只是为代码实现提供支持
  - 目前尚未产生成熟的可执行模型
- 正向工程（Forward engineering）
  - 由设计类图导出框架代码
  - 由交互图创建方法实现
  - 由状态机图生成状态转换控制代码
- 逆向工程（Reverse engineering）
  - 由源代码导出设计模型

正向工程-生成框架代码

- 什么是框架代码?
  - 代码在设计上的初步实现
- 类的框架代码包括那些?
  - 属性值定义：名称、类型、缺省值等
  - 操作的定义：名称、参数、返回类型等
  - 引用属性的表示
  - ……
- 根据设计类图产生框架代码
  - 用操作和简单属性定义一个类
  - 加入引用属性：角色名定义引用属性

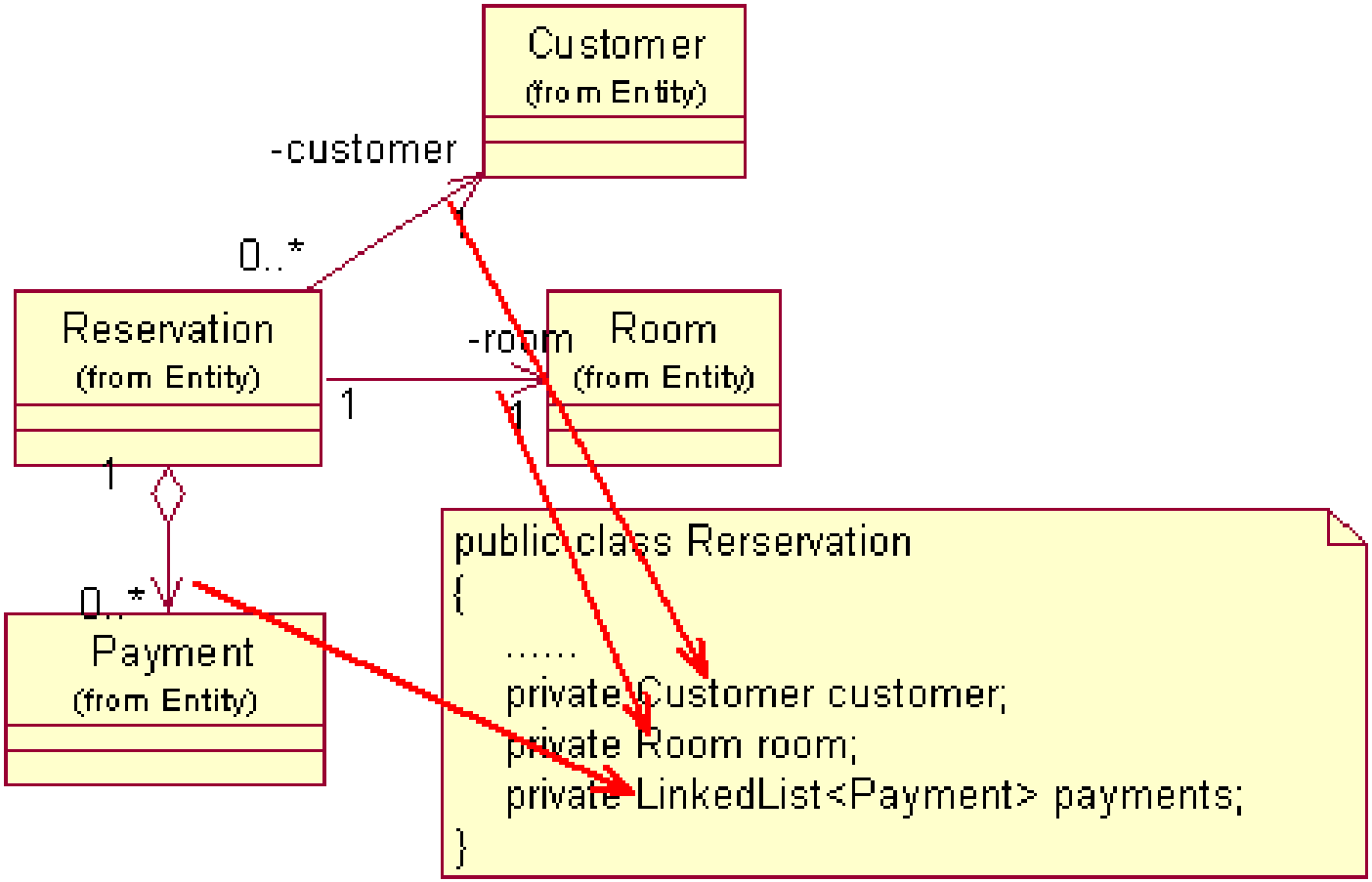
从设计类图产生框架代码-1

1. 用操作和简单属性定义一个类（属性、操作）



从设计类图产生框架代码-2

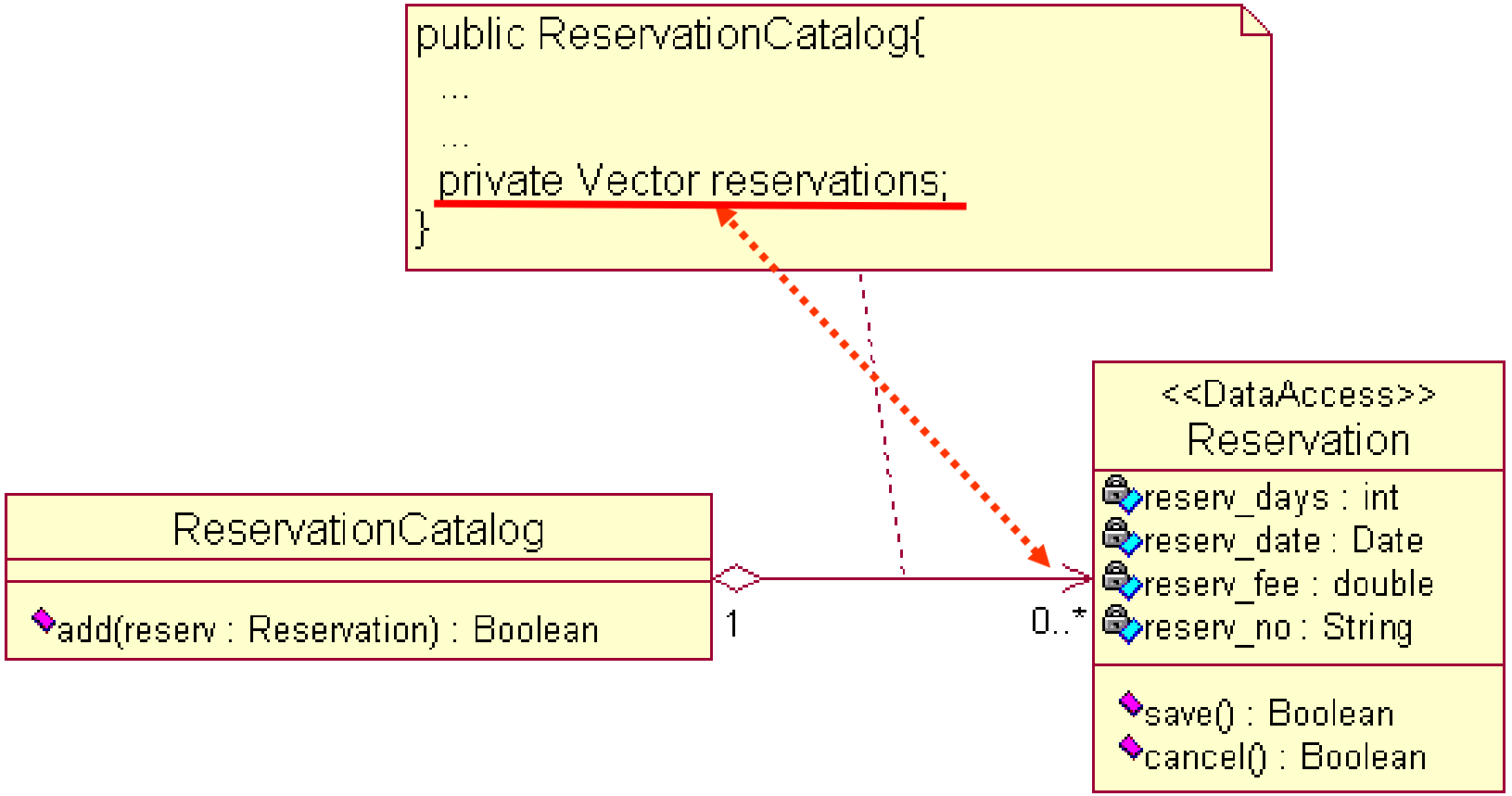
2. 加入引用属性（引用属性reference attribute：关联和导航；角色名role name）



EA支持框架代码的导出

- 多数UML CASE工具都支持框架代码导出
- 利用EA由设计类图生成框架代码
- 主要问题：
  - 设计模型中的细节存在错误，无法实现
  - 设计模型开发不够完善，无法导出代码

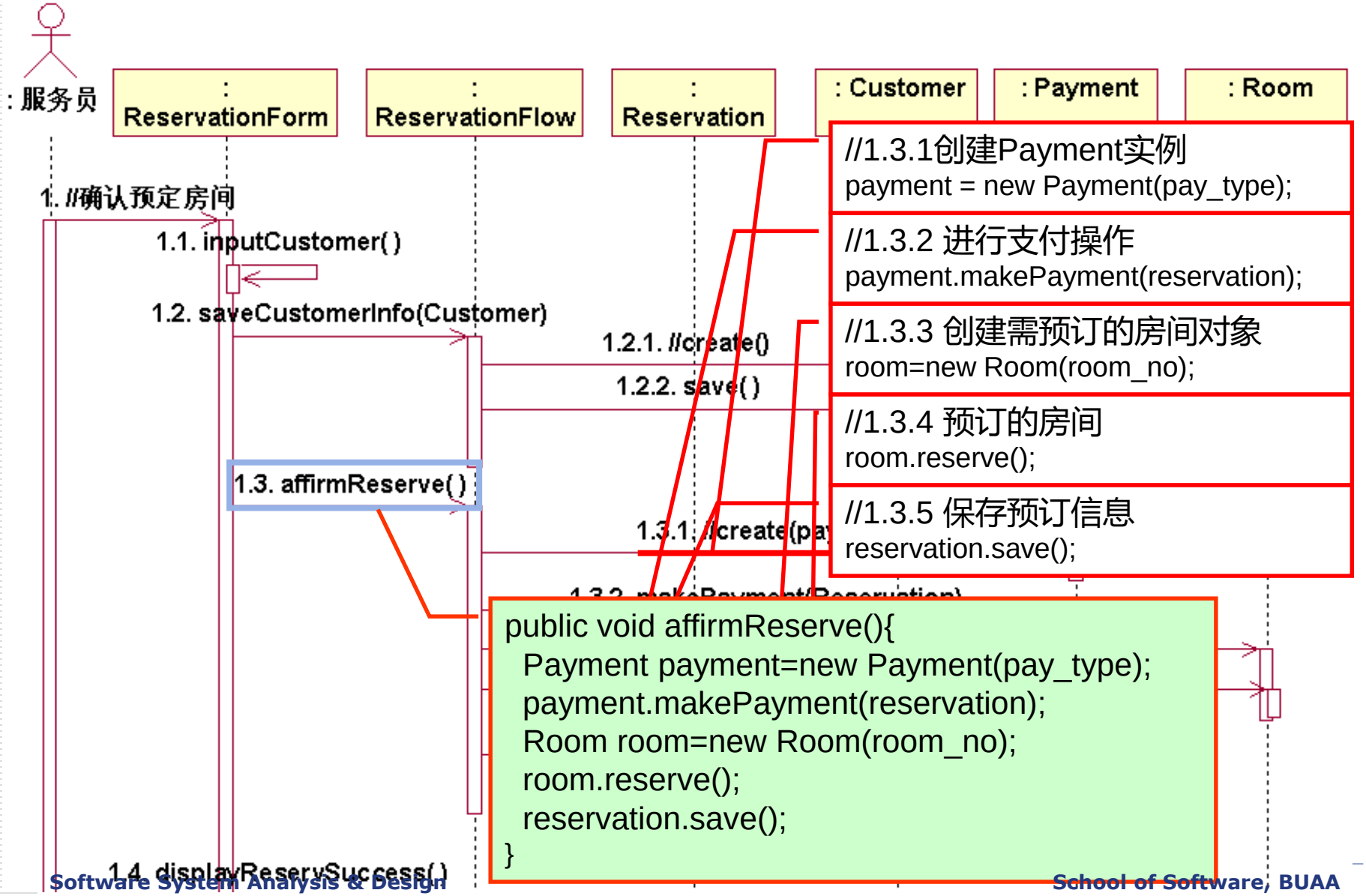
其它问题-一对多关系的实现



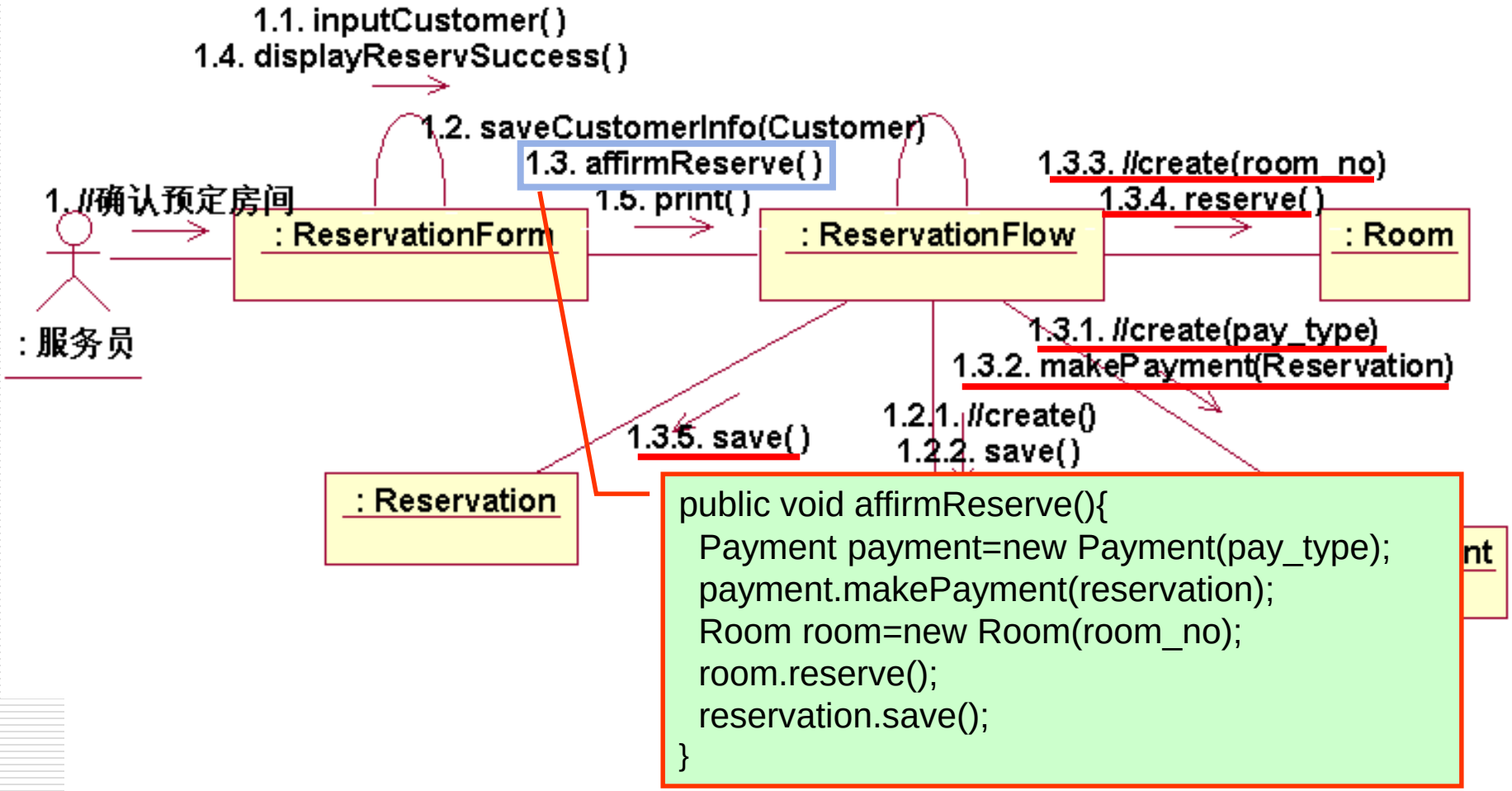
正向工程-创建方法实现

- 一个交互图显示出了响应操作调用而产生的消息传递；这些消息序列可以被翻译成该操作所对应的方法实现中的一系列语句
  - 由顺序图产生方法实现
  - 由通信图产生方法实现

确定预订房间操作实现(顺序图)



确定预订房间操作实现(通信图)



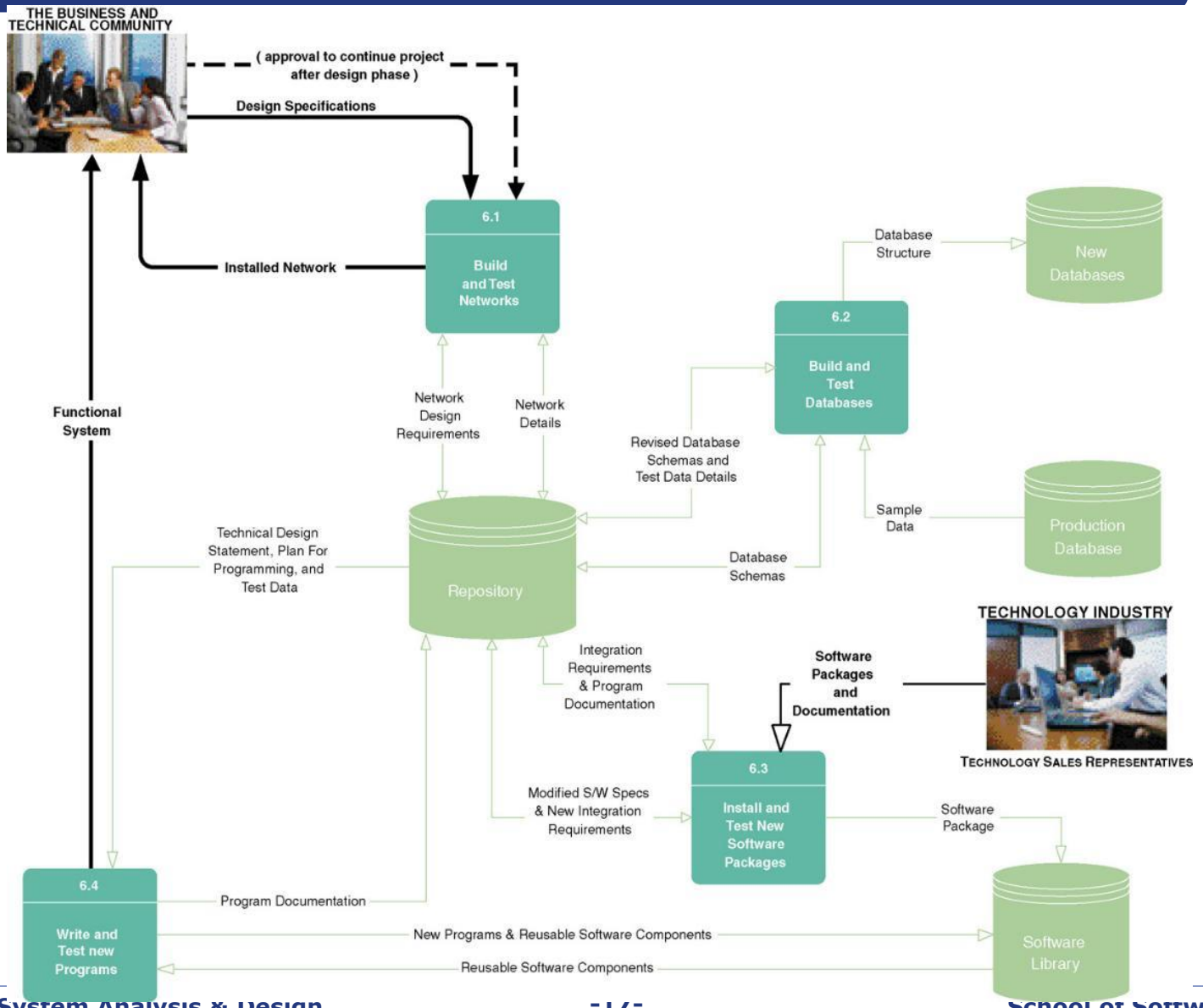
逆向工程-由代码导出模型

- 逆向工程
  - 根据源代码导出设计模型
  - 设计类图、设计交互图
- 主要作用
  - 开始编码后，很多地方势必会和设计模型不一致，此时可以通过逆向工程更新设计模型，从而需要保持设计模型的有效性
  - 已有的系统缺少相关文档，通过逆向工程获得系统的设计模型，以便理解和完善文档
- 示例：利用EA导入JDK源代码

内容安排

- 从设计到实现
- 构造软件系统
- 部署和上线运行
- 上线后运维
- 基于华为云的DevOps开发

Completing The Construction Phase



Construction Phase – 1. Build and Test Networks

- Build and Test Networks
  - Often system build around existing networks.
  - If system calls for new network functionality, must by built and tested prior to programs that use that it.
- Roles
  - Network designer
    - Designs LAN and WAN connectivity
  - Network administrator builds and tests
    - Network architecture standards
    - Security
  - Systems analyst
    - Facilitates
    - Ensures that business requirements are not compromised

Construction Phase – 2. Build and Test Databases

- Implement database schema
- Test with sample data
- Deliver unpopulated database structure
- Roles
  - System users
    - Provide and/or approve test data
  - Database designer/programmer
    - Build tables, views, stored procedures (if relational database)
  - Database administrator
    - “Tune” database for optimum performance
    - Security
    - Backup and recovery
  - Systems Analyst
    - Build non-corporate, applications-oriented database
    - Ensures business requirements compliance

Construction Phase – 3. Install and Test New Software

- If system requires purchased or leased software, must be installed and tested.
- Roles
  - Systems analyst
    - Clarifies business requirements
  - System designer
    - Clarifies integration requirements
  - Network administrator
    - Install software package
  - Software vendor/consultant
    - Assist in installation and testing
  - Applications programmer
    - Test according to integration requirements

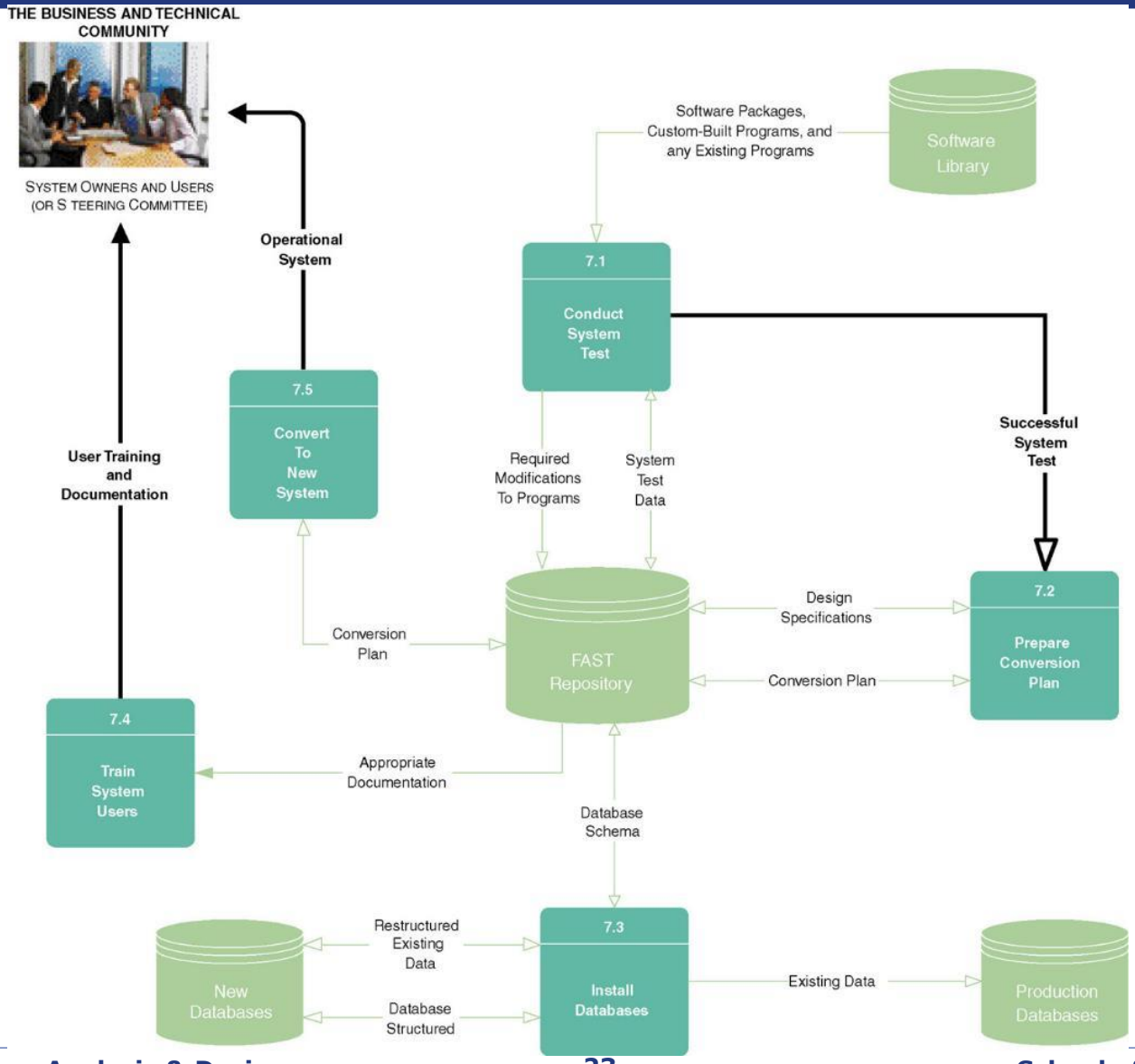
Construction Phase – 4. Write and Test New Programs

- Develop in-house programs
  - Reuse available software components in library
  - Write new components
  - Test
  - Document
- Roles
  - Systems analyst
    - Clarifies business requirements
  - System designer
    - Clarifies program design and integration requirements
  - Application programmer (or team)
    - Writes and tests in-house software

内容安排

- 从设计到实现
- 构造软件系统
- 部署和上线运行
- 上线后运维
- 基于华为云的DevOps开发

Completing The Deployment Phase



Deployment Phase - 1. Conduct System Test

- Test network, databases, purchased software, new in-house software, and existing software to make sure it all works together.
- Roles
  - Systems analyst
    - Develops system test data
    - Communicates problems and issues
  - System builders (database, network, programmers)
    - Resolve problems revealed during testing
  - System owners and users
    - Verify whether or not system operates correctly
- May result in return to construction phase
  - Iterate until successful system test

Systems Acceptance Test

- Systems acceptance test (系统验收测试) – a test performed on the final system wherein users conduct a verification, validation, and audit test
  - Uses real data over an extended time period
  - Extensive test that addresses: verification testing, validation testing, and audit testing.

Deployment Phase – 2. Prepare Conversion Plan

- Plan for how to convert from old system to new system.
  - How to install and populate databases
  - How to train users
  - Finalize documentation
  - Conversion issues
- Roles
  - System analyst/Project manager
    - Develop a detailed conversion plan
  - Steering committee
    - Approves plan and timetable

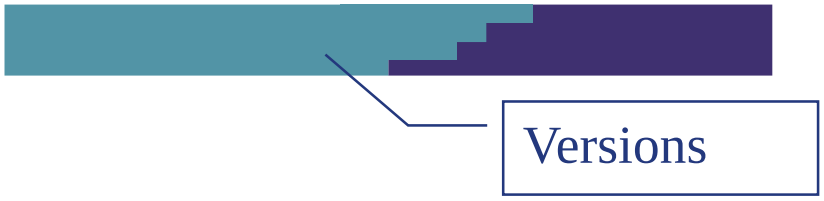
Installation Strategies

- Abrupt cutover

- Parallel conversion

- Location conversion

- Staged conversion



Deployment Phase – 3. Install Databases

- Populate new system databases with existing data from old system
  - Generally have to restructure data as it is populated
  - Must confirm that data is translated correctly
- Roles
  - Application programmers
    - Write (or use) special programs to extract data from existing databases and populate new databases
  - Systems analyst/designer
    - Calculate database sizes and estimate time for installation

Deployment Phase – 4. Train Users

- System users trained and provided with documentation
- Roles
  - System analyst
    - Plan trainings
    - Conduct trainings
    - Write documentation
    - Help users through the learning period
  - System owners
    - Approve release time for training
  - System users
    - Attend training
    - Accept system

Deployment Phase – 5. Convert to New System

- Ownership transfers from analysts and builders to end users.
- Roles
  - Systems analyst/Project manager
    - Carries out conversion plan
    - Correct shortcomings
    - Measure system acceptance
  - System owners
    - Provide feedback
  - System users
    - Provide feedback

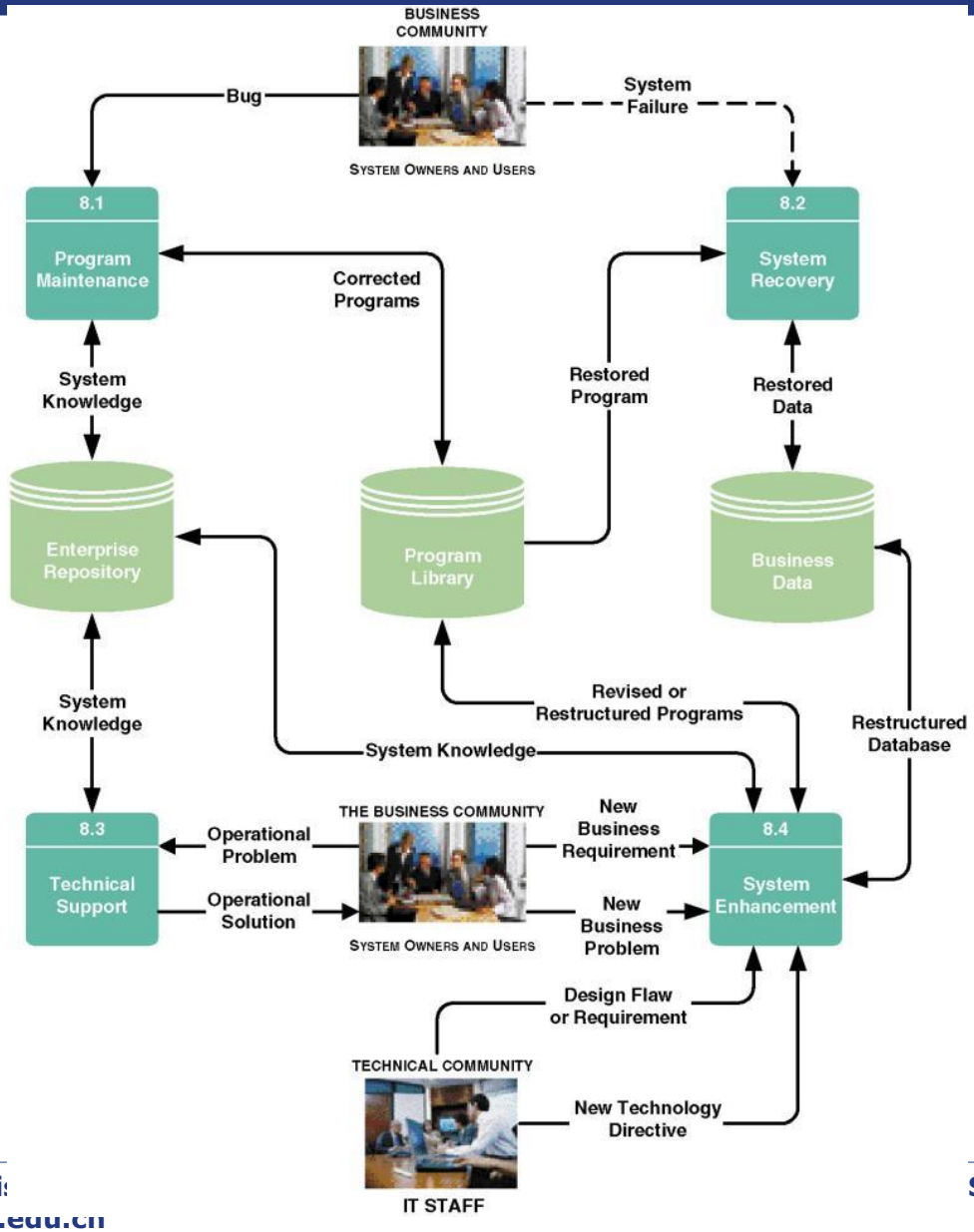
内容安排

- 从设计到实现
- 构造软件系统
- 部署和上线运行
- 上线后运维
- 基于华为云的DevOps开发

Support versus Operation

- *Systems support* (系统支持) is the on-going technical support for users, as well as the maintenance required to fix any errors, omissions, or new requirements that may arise.
- *Systems operation* (系统运行) is the day-to-day, week-to-week, month-to-month, and year-to-year execution of an information system's business processes and application programs.
- An operational system is a system that has been placed into operation. Frequently called a **production system** (生产系统).

Systems Support Activities



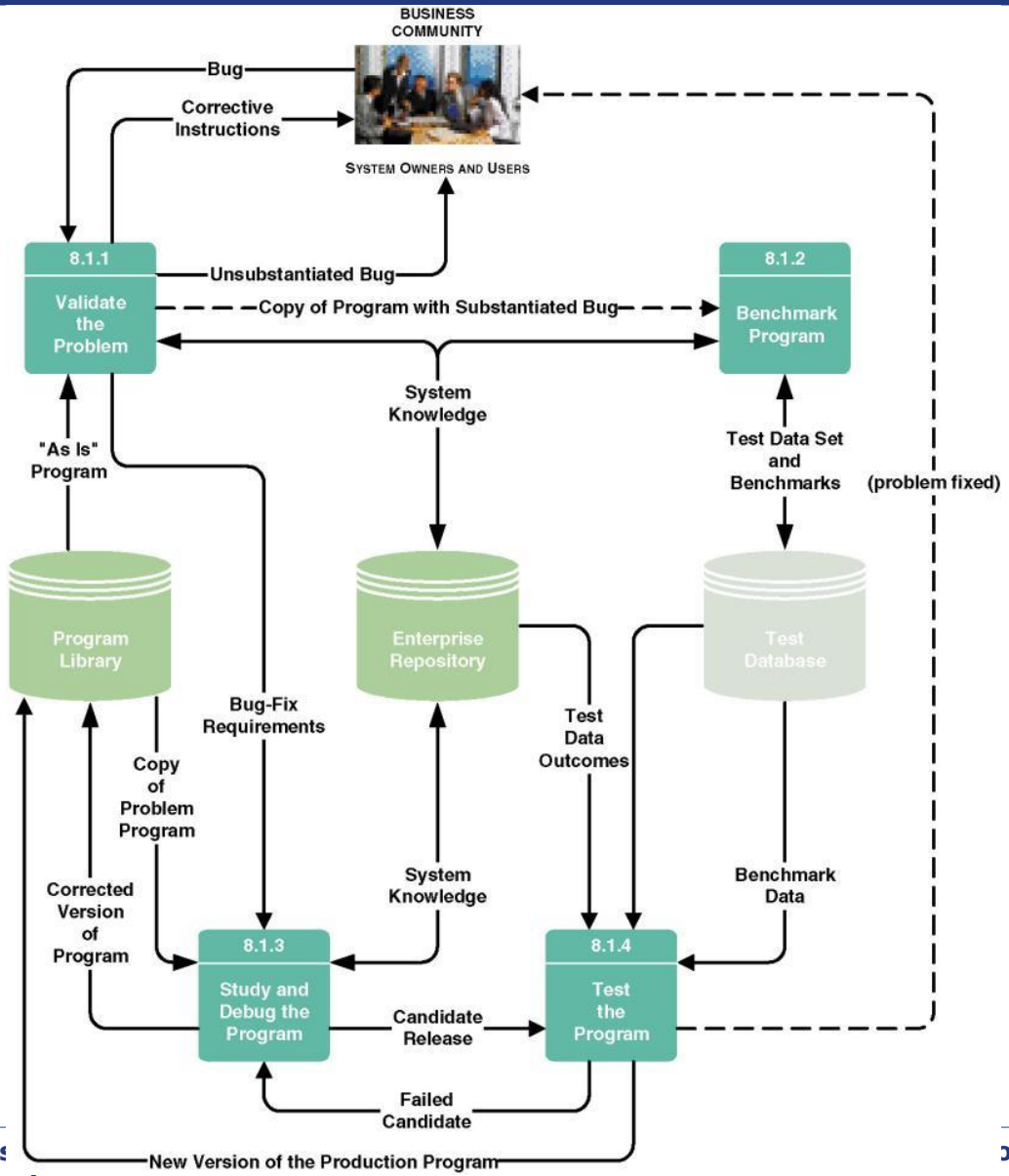
System Support Activities

- *System maintenance* (系统维护) corrects “bugs” or errors that slipped through the system development process.
- *System recovery* (系统恢复)is the restoration of the system and data after a system failure.
- *Technical support* (技术支持) is any assistance provided to users in response to inexperience or unanticipated situations.
- *System enhancement* (系统改进) is the improvement of the system to handle new business problems, new technical problems, or new technology requirements.

System Maintenance Objectives

- To make predictable changes to existing programs to correct errors.
- To preserve those aspects of the programs that were correct, and to avoid “ripple effects” of changes that may adversely affect the correctly functioning aspects.
- To avoid, as much as possible, the degradation of system performance.
- To complete the task as quickly as possible without sacrificing quality and reliability of the system.

System Maintenance Tasks



System Maintenance Tasks

1. Validate the problem.
2. Benchmark the program.
   - test script - a repository of test cases to be executed against all program revisions.
3. Study and debug the program to fix:
   - Poor program structure.
   - Unstructured (or poorly structured) logic.
   - Prior maintenance (so-called “ripple” effects.)
   - Dead code.
   - Poor or inadequate documentation.
4. Test the program.

System Recovery Activities

1. Recover from user's terminal.
   - Menu option, Reboot
2. Systems operations personnel correct server problem (network admin, database admin, webmaster).
3. Data administrator recovers lost data or corrupted files.
   - Lost transactions must be reprocessed (roll forward)
   - Partially processed transactions must be undone (roll back)
4. Network administrator fixes LAN or WAN problem.
5. Technicians or vendor reps fix hardware problem.
6. Software bug must be trapped and fixed.

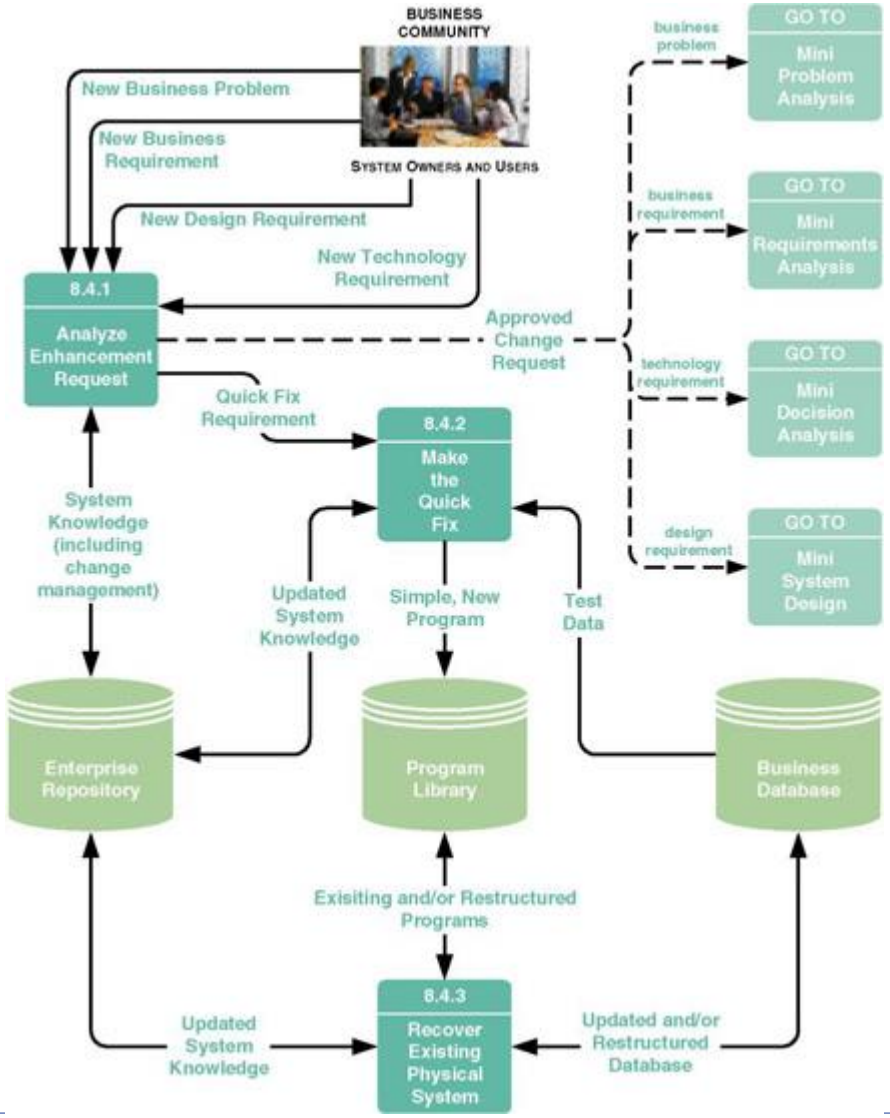
Technical Support

- Being on call to assist users
- Typical tasks:
  - Routinely observing use of system
  - Conducting user-satisfaction surveys and meetings
  - Changing business procedures for clarification
  - Providing additional training
  - Logging enhancement ideas and requests in repository

System Enhancement Triggers

- New business problems
- New business requirements
- New technology requirements (inclusive of hardware and software upgrades)
- New design requirements

System Enhancement Tasks



System Enhancement Tasks

1. Analyze enhancement request.
2. If appropriate, make the quick fix.
   - Changes that can be made without
     - Restructuring stored data
     - Updating stored data
     - Inputting new data
   - In other words, reports and outputs
   - Requirements that exceed this should be subjected to systems analysis and design to consider implications.
   - Can be written in 4GLs

System Enhancement Tasks (continued)

3. Recover the existing physical system:
   - Updating repository and documentation for changes
   - Database recovery and restructuring
   - Program analysis, recovery, and restructuring
     - Software metrics - mathematically proven measurements of software quality and developer productivity.
     - Code reorganization of modularity and/or logic
     - Code conversion from one language to another
     - Code slicing to create reusable software components or objects out of existing code

System Obsolescence

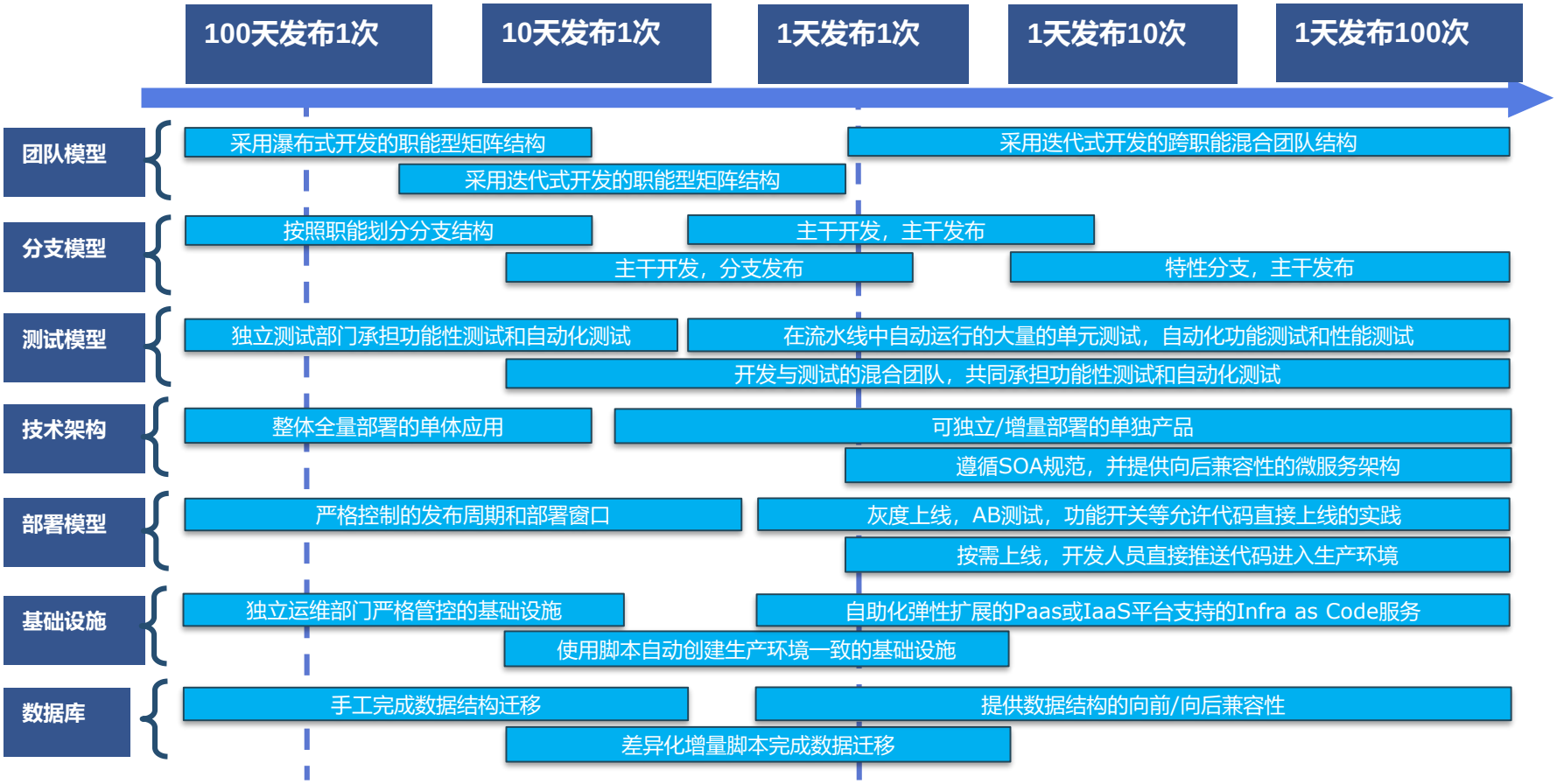
- All systems degrade over time
- At some point, not cost-effective to support and maintain
- Leads to a new systems development project
- Full circle to chapters 2-9

内容安排

- 从设计到实现
- 构造软件系统
- 部署和上线运行
- 上线后运维
- 基于华为云的DevOps开发

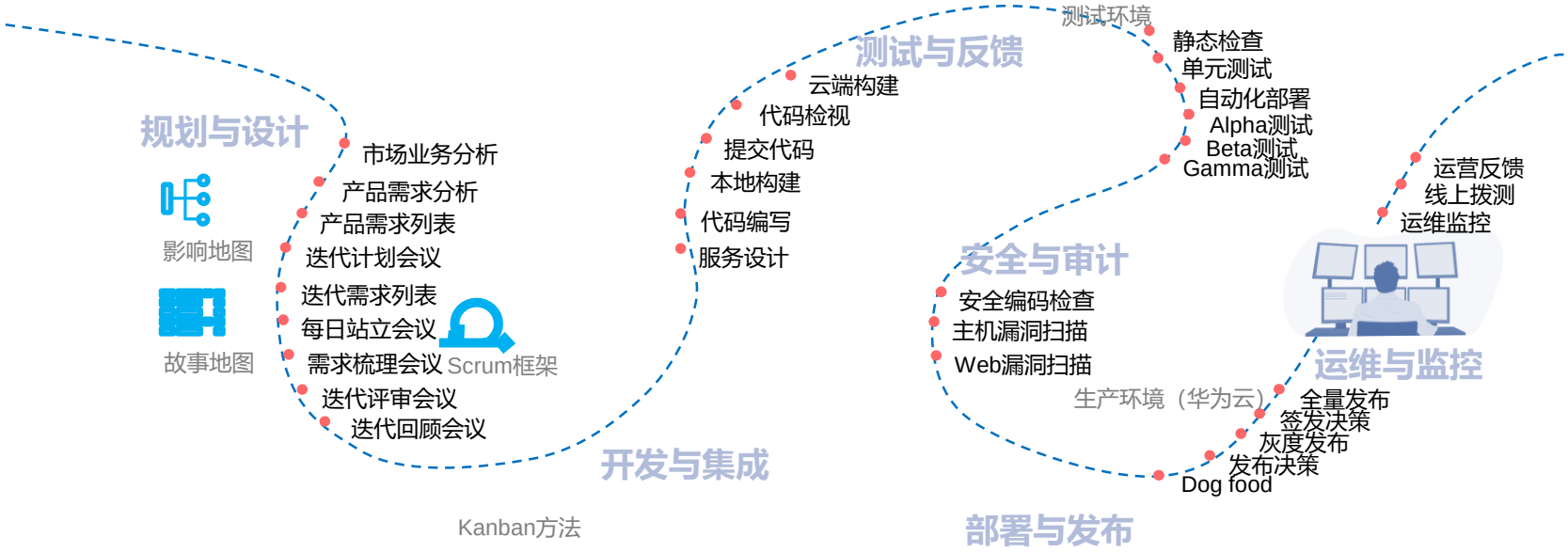
DevOps持续交付实施框架

关注七大领域，持续优化交付粒度，加快交付速度，提升交付质量



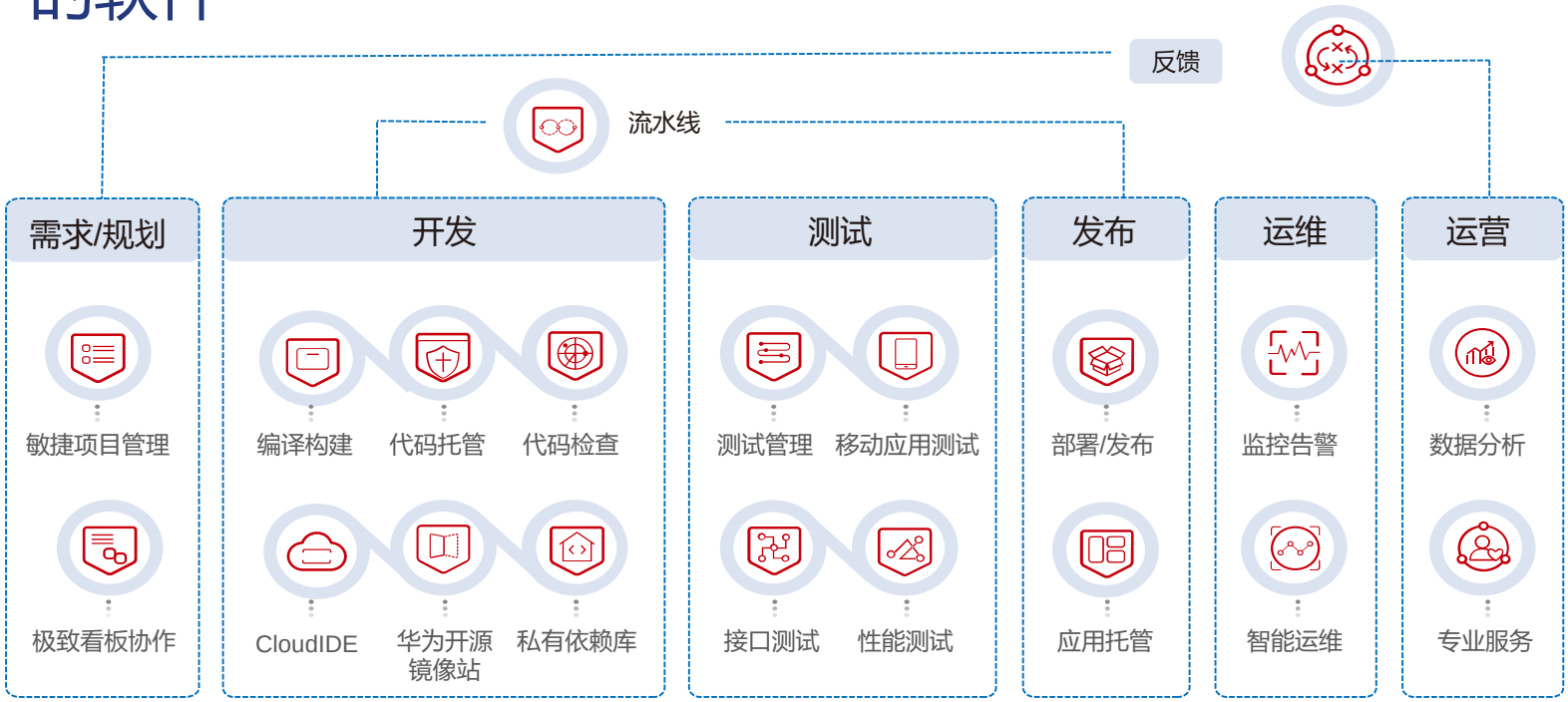
华为云DevCloud HE2E DevOps框架

- 集合业界先进理念，华为30年研发经验，可操作可落地的端到端一站式开发方法论和工具链

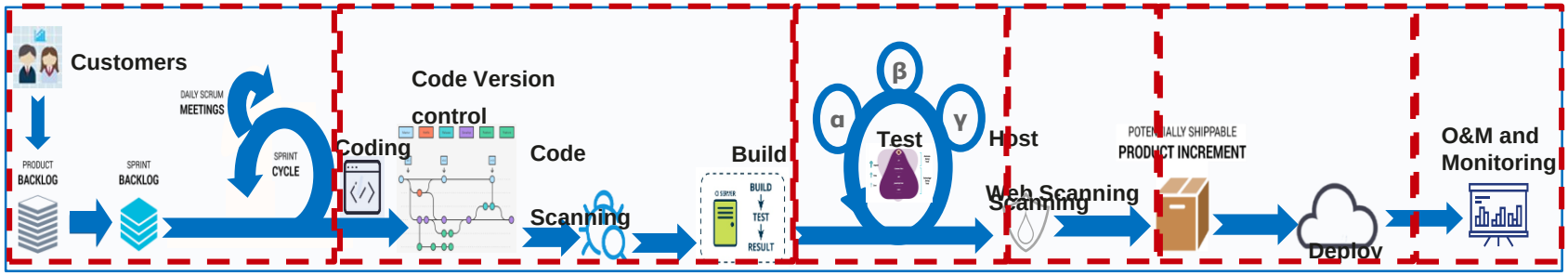


华为云DevCloud - 一站式，全流程，安全可信的DevOps平台

- 集华为研发实践、前沿研发理念、先进研发工具于一体，使能软件企业/开发者简单高效地向最终用户交付有价值的软件

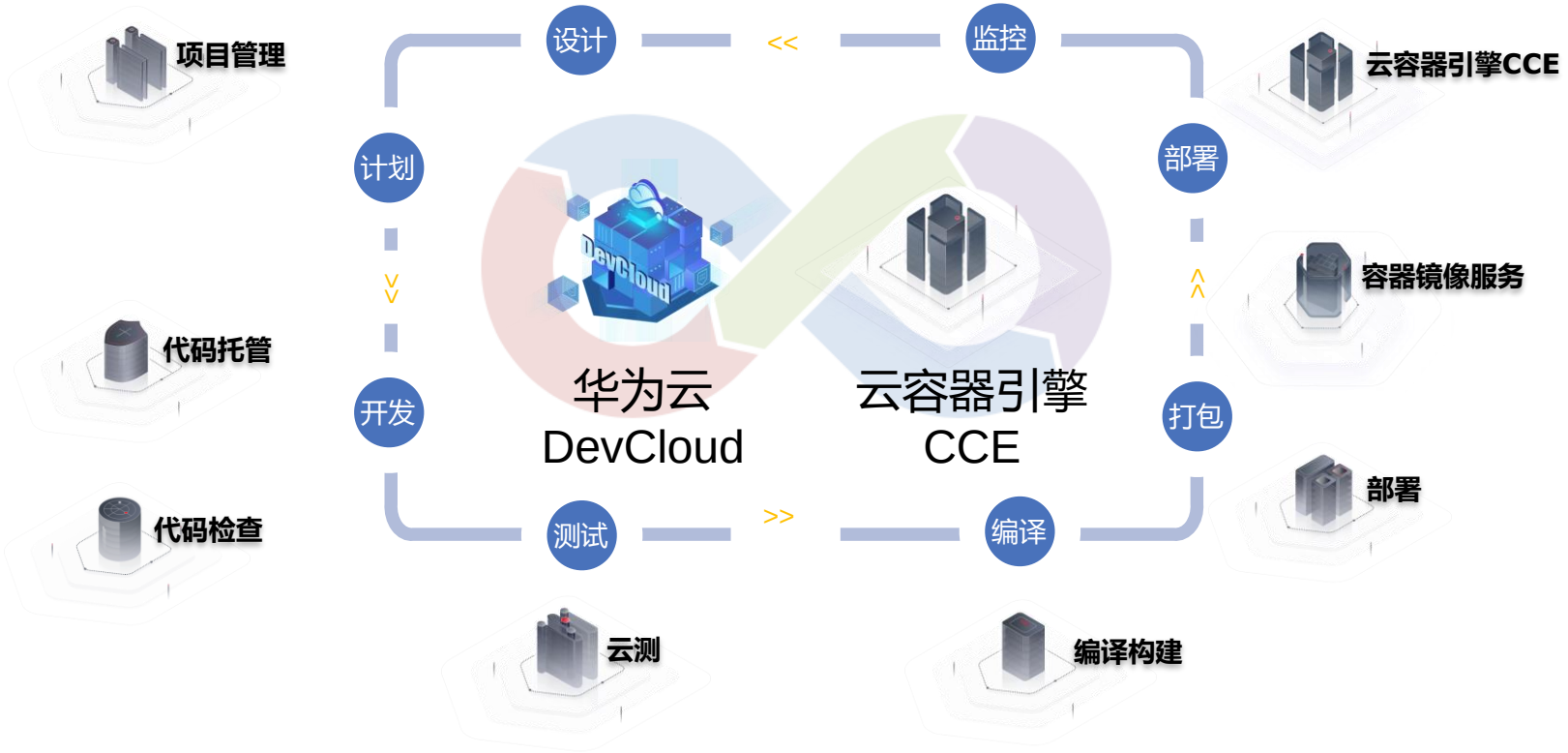


端到端工具链服务 - 涵盖软件生命周期环节



持续设计和规划	持续开发和集成	持续测试与反馈	持续安全与审计	持续部署与发布	持续运维与监控
2种项目流程（Scrum和看板）； 自定义的研发作业流； 14种度量看板； 6+华为实践模板WIKI；	10+种语言和20+种框架； CloudIDE支持100+种编程语言，30秒极速启动，对接多种代码仓库； 开源镜像站支持100+开源组件、工具操作系统加速下载，全国CDN，国内唯一Maven Central镜像站。	云测支持REST、Swagger等多种接口免编码功能测试，利用云端弹性资源快速模拟3K+用户并发性能测试； 2万+真机，利用精准图像和控件识别技术测试移动APP兼容性。	主机漏洞扫描进度高、支持多种通用Liunx系统及华为自研EulerOS系统。 Web漏洞扫描，零环境搭建成本、一键完成扫描。 提供清晰简洁的扫描报告和修复建议。	可视化编排流水线，支持并行/串行、人工卡点、质量门禁、构建/检查/测试/部署等多任务类型、分层分级流水线等； 利用统一调度引擎、云上弹性资源能力实现百万级并发任务调度能力。	全方位覆盖业务、应用、和基础设施层监控； 指标、日志、性能、告警等多维度关联分析。

基于华为云DevCloud和云容器引擎的DevOps流水线



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

敏捷项目管理

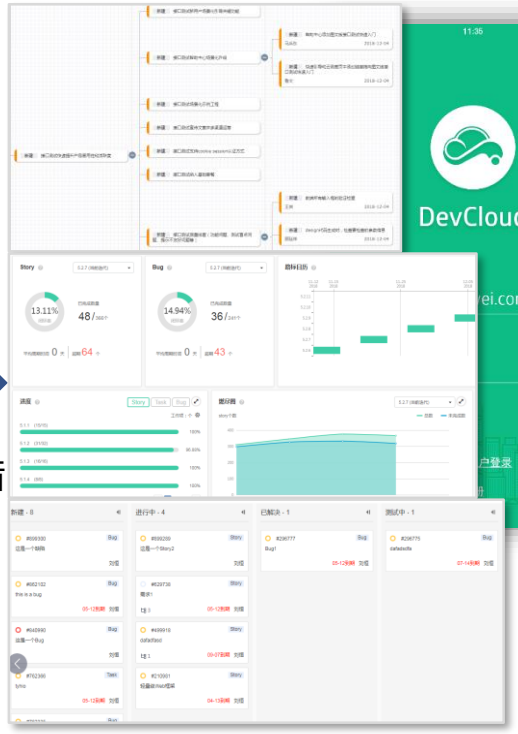
丢需求

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

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

代码托管

注重安全

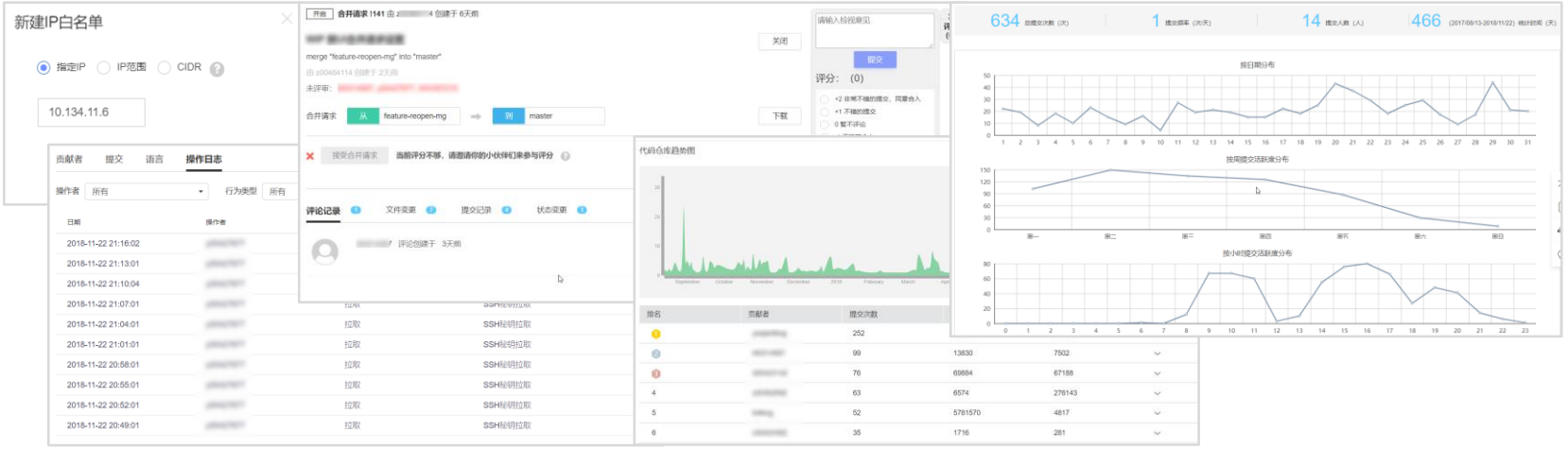
- 支持IP白名单设置，确保接入安全；
- 支持访问日志，方便审计；
- 支持加密存储，防范代码泄露；
- 支持Git-Crypt，扩展加密存储能力；
- 安全防护墙，阻止用户上传敏感信息。

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。

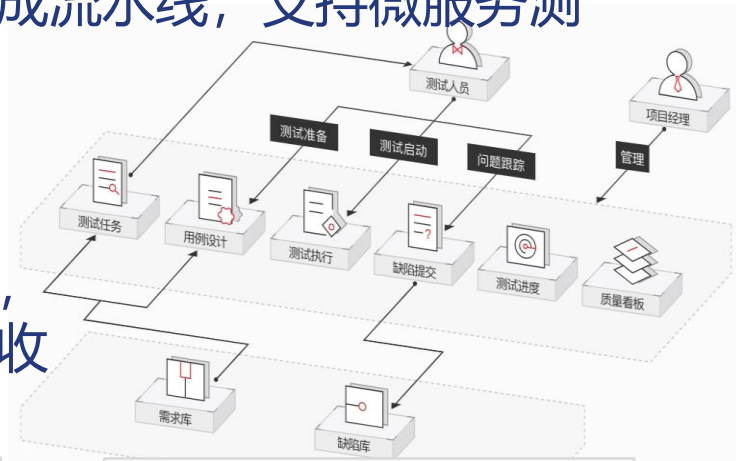
贡献可视化

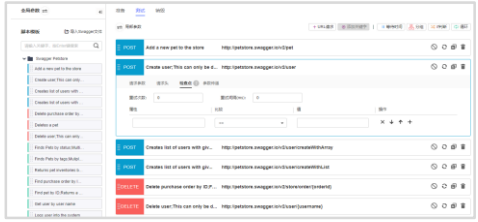
- 代码语言类型统计；
- 代码增长趋势图；
- 小组贡献量排名；
- 提交活跃度分布；
- 提交质量统计。



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，
    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



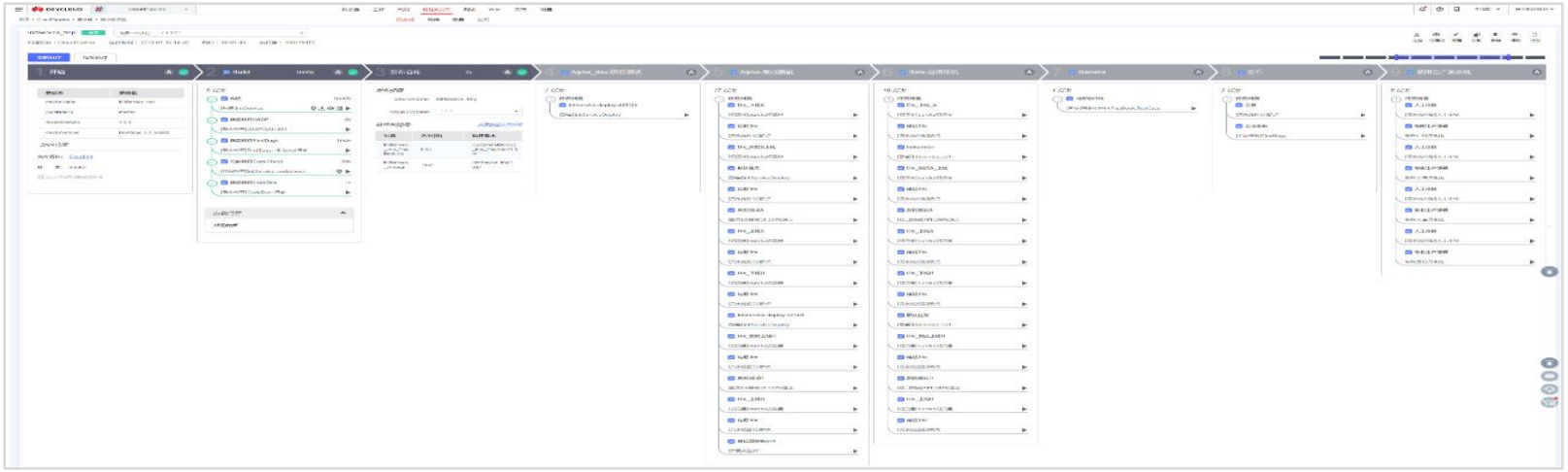


持续集成和持续部署

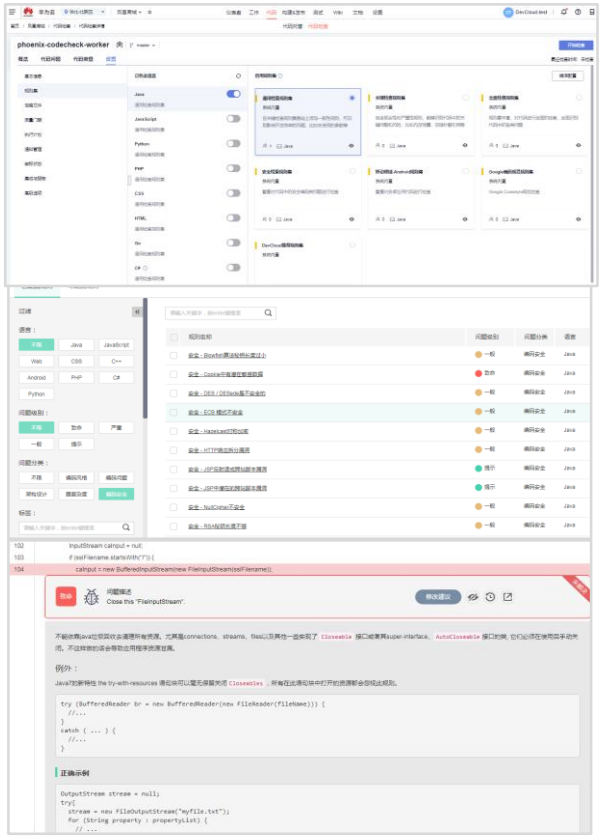
- 使用华为云DevCloud内置的CI/CD能力，持续交付价值

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



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8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。

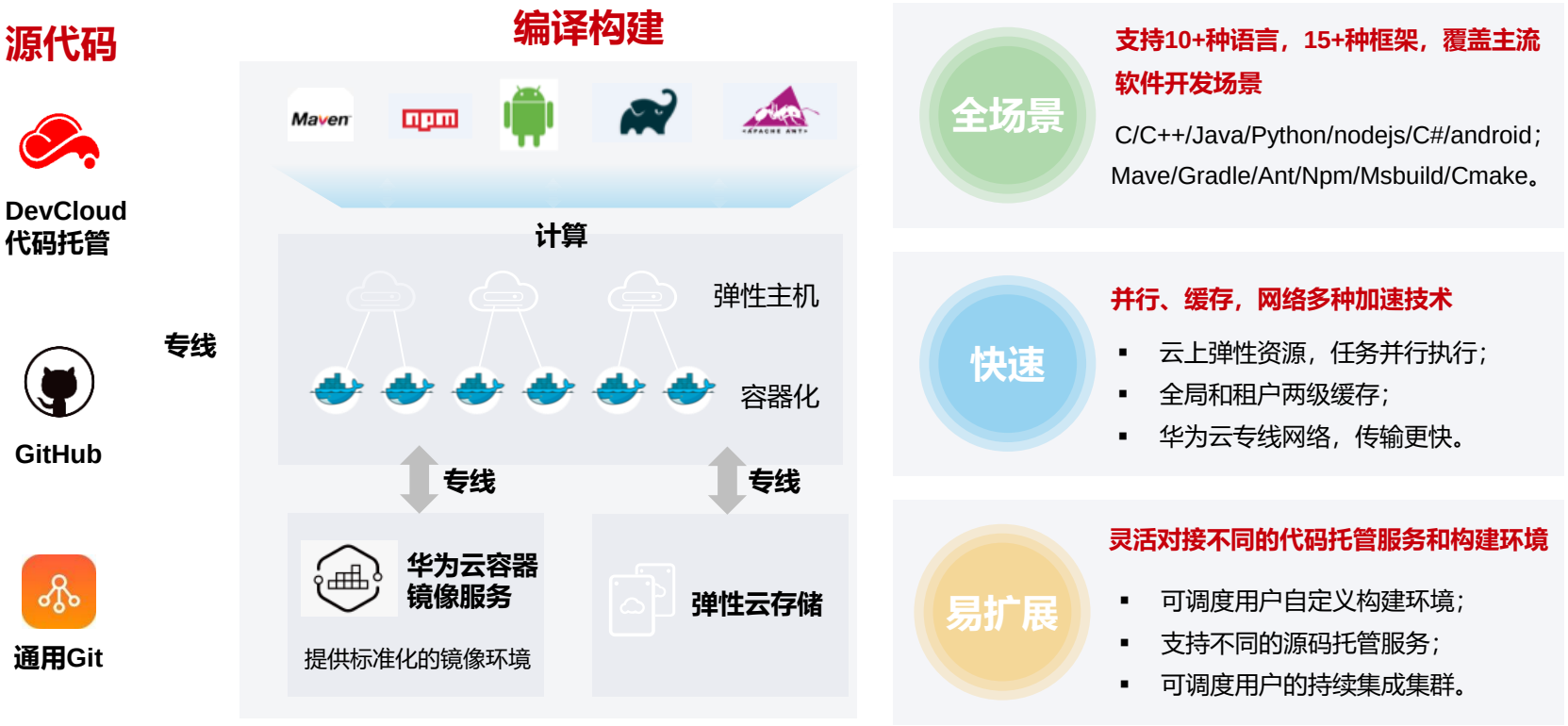
专业

- 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。

编译构建 - 一站式的持续集成



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

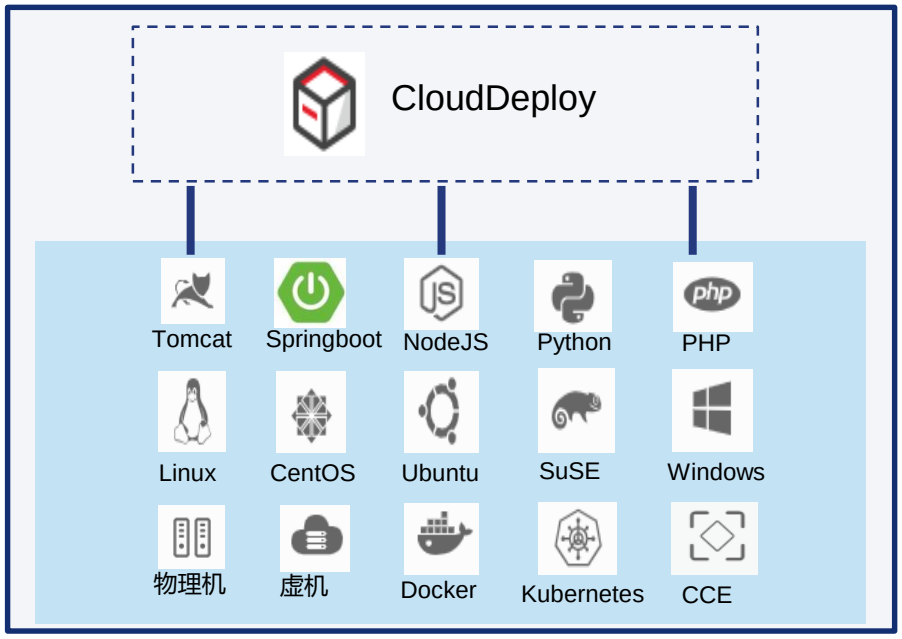
- 云上弹性资源，任务并行执行；
- 全局和租户两级缓存；
- 华为云专线网络，传输更快。

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。

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。

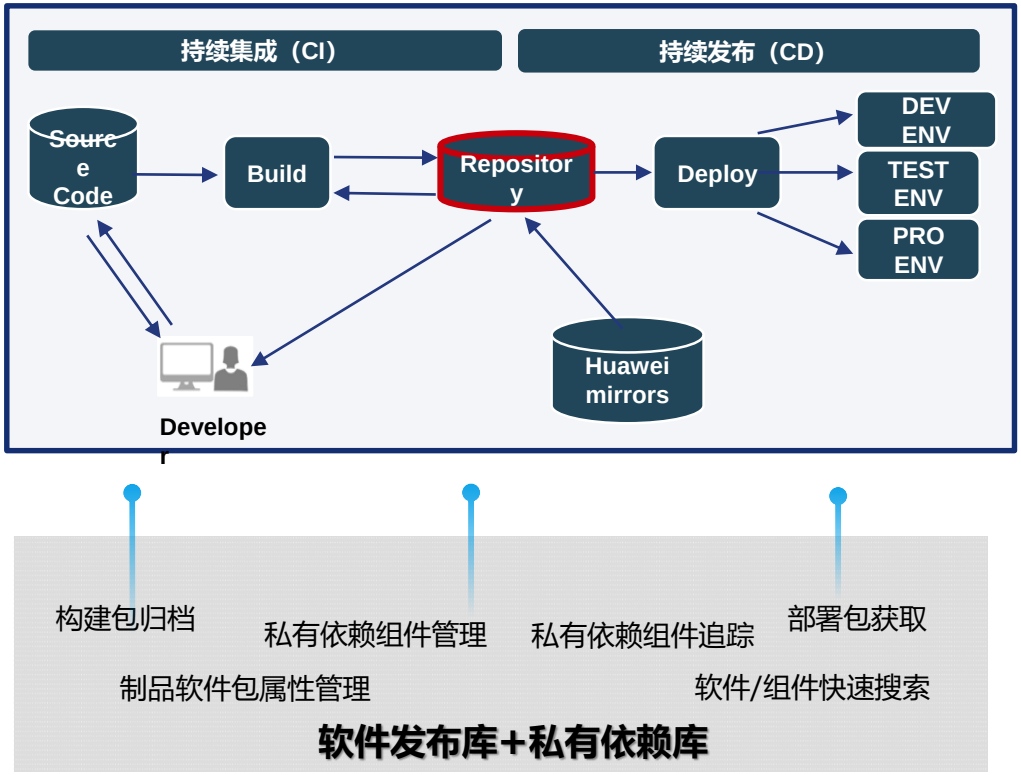
支持多种技术栈应用的部署

- 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。

发布 - 软件包资产的可视化管理和追溯



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。

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、……

- 国外开源组件**下载慢；**
- 国内镜像站**内容不全面；**
- 异地协作**难以保证多地下载**速度；
- 开发对各包管理工具**配置不了解；**
- 开源组件**来源有安全风险；**
- 国内镜像站**不能保证长期维护和稳定。**

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。

容器化云环境CCE

学习用户行为，持续改进产品体验

CCE内置的集群监控能力，确保应用稳定高效运行

反馈 <<

持续反馈

监控 <<

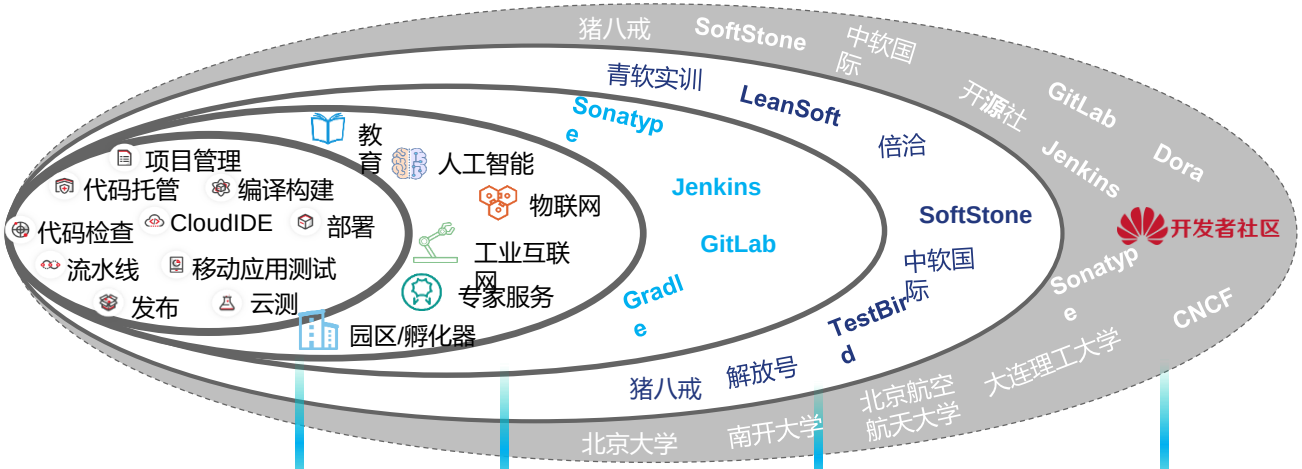
k8s集群管理

容器镜像管理

容器镜像仓库SWR提供简单易用、安全可靠的镜像管理

部署

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

合作伙伴　方案伙伴　高校　开发者社区开源合作

培训认证　专家服务　园区

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。