



CS2001

软件工程

2. 个人开发技能

— 软件演化与配置管理

课堂讨论：软件演化

Class Discussion



课堂讨论：软件会因为什么原因发生变化

为了使软件持续保持有用

发现缺陷后进行修正

应客户和用户要求或为保持竞争优势增加新特性

改进系统的性能、可靠性等方面的质量

根据对软件未来发展的展望自主添加新特性

迁移到新的软硬件运行环境上

应用新技术、新硬件等

改进软件本身的可维护性（如可理解性、设计合理性）

一个软件产品的演化之路

- 给家里小孩写了一个简单算术练习软件
- 教学内容加深随之增加了带括号的复杂算式
- 老师看了之后说不错，希望能让全班同学都能用上，因此把单机版改成了Web版
- 老师想看到每次学生的答题时间和错误情况，引入数据库记录答题情况并增加用户登录和账户管理
- 校长希望全校推广，因此引入班级信息及管理功能
- 访问人数增多，服务器不堪重负，增加服务器并修改系统支持负载均衡
- 系统经常不稳定，抱怨渐起，因此迁移到阿里云
- 持续改进和系统运维不堪重负，跟学校讨论商业化运营和推广模式...

演化式开发

Evolutionary Software Development

The evolutionary software development model is based on the idea of rapidly developing an software from start with very raw specification and modifying this according to your softwaree needs.

<https://www.quora.com/What-is-the-evolutionary-software-development-model>

开发与演化进一步融合

开源软件演化示例—fastjson

fastjson: 实现Java对象与Json对象转换

演化问题: 希望支持stomp消息编程中的对象转换



liuzou1991 commented 28 days ago



在使用spring的stomp消息编程时, 当前只能使用 `org.springframework.messaging.converter.MappingJackson2MessageConverter` 处理json到java对象间的转换, 不知道fastjson有没有对这个进行支持的打算

add MappingFastJsonMessageConverter and testcase for #1736 #1737

Merged wenshao merged 2 commits into `alibaba:master` from `kimmking:master` 28 days ago



Conversation 1

Commits 2



Files changed 3



kimmking commented 28 days ago

Contributor



add MappingFastJsonMessageConverter and testcase for #1736



kimmking added some commits 28 days ago



Merge pull request #18 from `alibaba/master` ...

Verified

20510db



add MappingFastJsonMessageConverter and testcase for spring-messaging

✓ 2f45f84

开源软件演化示例—Jersey RESTful



aakhmerov commented on Dec 15, 2017 • edited ▾

Spring 5 went GA on 28.11 it would be nice to provide integration with it.
See <https://spring.io/blog/2017/09/28/spring-framework-5-0-goes-ga> for release notes.

Things to watch out for:

- spring 5 supports java 1.8 or greater only



1

Jersey RESTful: 开源的
RESTful框架，实现了JAX-
RS（JSR 311 & JSR 339）

演化问题：希望支持与
Spring5集成

feat(jersey): add spring 5 support #3737



aakhmerov wants to merge 1 commit into jersey:master from aakhmerov:master



Conversation 2



Commits 1



Files changed 48



aakhmerov commented on Dec 15, 2017

related to #3736

feat(jersey): add spring 5 support ...

5675d80



jansupol commented on Dec 16, 2017

Owner



Thanks for working on this, it would be nice to have the support. However, we cannot merge it right now. First, you would need to sign up an OCA. Second, we shouldn't bring a new dependency in until Jersey is eventually moved to the Eclipse Foundation (increase of legal complexity). But we will be there soon.



aakhmerov commented on Dec 18, 2017 • edited ▾



Hi @jansupol,

thanks for getting back to this pull request. I hope this can be used as a base for future work, meanwhile I will just use dependencies from our local nexus. In any case I realized, that spring 5 integration is not crucial for the project I am currently working on.

Regarding signing OCA, I think this is not an issue and I would do it if required, but if the project is going to be transferred to eclipse Foundation in any case, I guess it doesn't make a lot of sense in any case.

开源软件演化示例—MyBatis



aakhmerov commented on Dec 15, 2017 • edited ▾

Spring 5 went GA on 28.11 it would be nice to provide integration with it.
See <https://spring.io/blog/2017/09/28/spring-framework-5-0-goes-ga> for release notes.

Things to watch out for:

- spring 5 supports java 1.8 or greater only



1

MyBatis: 持久层框架，支持定制化 SQL、存储过程以及高级映射

演化问题: 对Oracle进行持久化操作时，对较长的数据做插入操作会有Bug

feat(jersey): add spring 5 support #3737



aakhmerov wants to merge 1 commit into jersey:master from aakhmerov:master



Conversation 2



Commits 1



Files changed 48



aakhmerov commented on Dec 15, 2017

related to #3736

feat(jersey): add spring 5 support ...

5675d80



jansupol commented on Dec 16, 2017

Owner



Thanks for working on this, it would be nice to have the support. However, we cannot merge it right now. First, you would need to sign up an OCA. Second, we shouldn't bring a new dependency in until Jersey is eventually moved to the Eclipse Foundation (increase of legal complexity). But we will be there soon.



aakhmerov commented on Dec 18, 2017 • edited ▾

Hi @jansupol,

thanks for getting back to this pull request. I hope this can be used as a base for future work, meanwhile I will just use dependencies from our local nexus. In any case I realized, that spring 5 integration is not crucial for the project I am currently working on.

Regarding signing OCA, I think this is not an issue and I would do it if required, but if the project is going to be transferred to eclipse Foundation in any case, I guess it doesn't make a lot of sense in any case.

软件演化对于软件工程的影响

• 软件开发

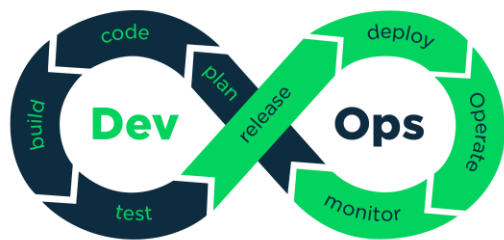
- ✓ 版本管理：记录历史版本及演化过程
- ✓ 设计质量：模块化、松耦合、易扩展
- ✓ 代码质量：易理解、易修改
- ✓ 复用规划：考虑未来类似功能的实现

• 软件测试

- ✓ 回归测试：防止修改的副作用

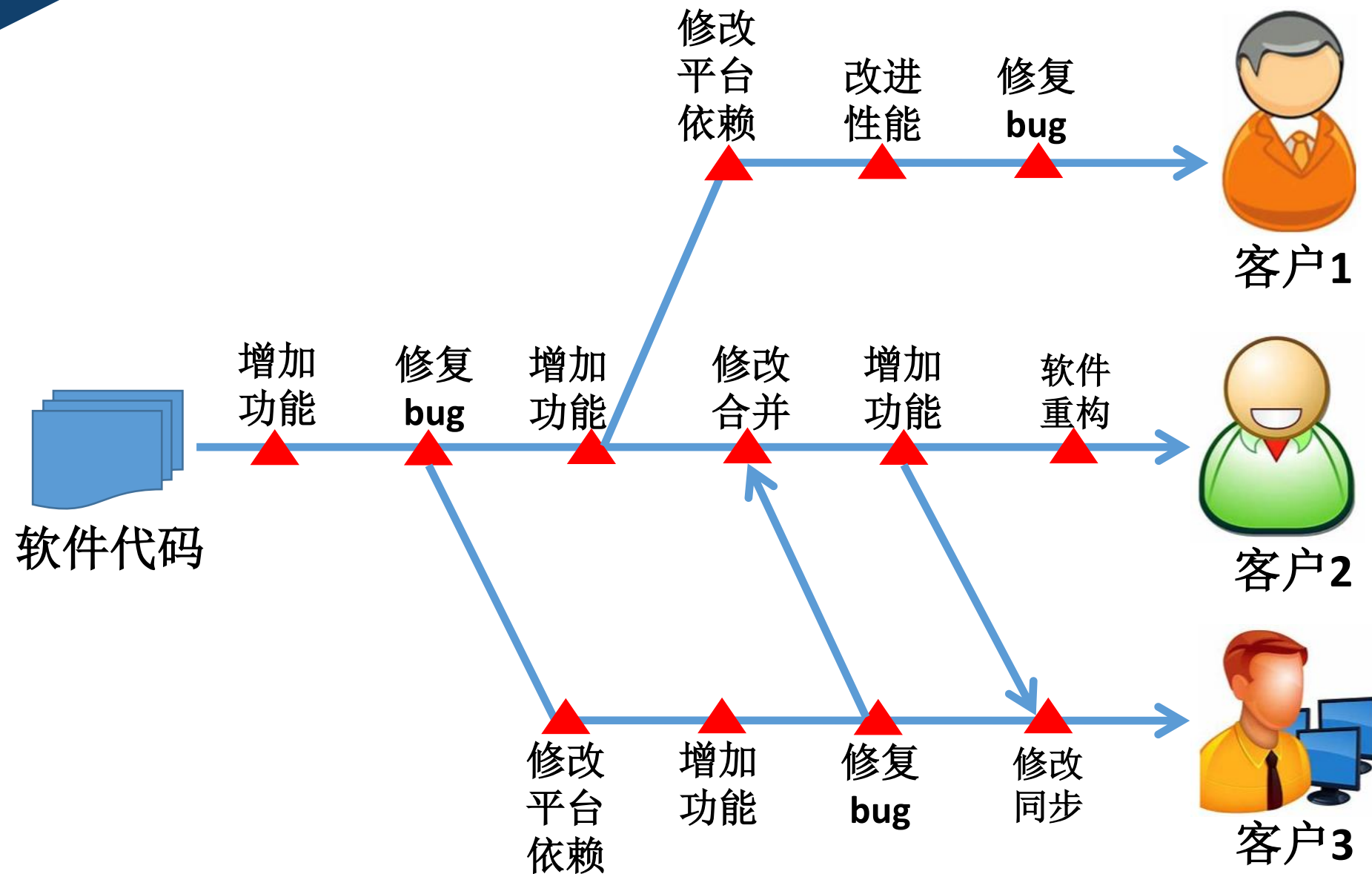
• 软件运维

- ✓ 软件更新：动态升级，更新不停机
- ✓ 系统可伸缩：因应服务请求的变化



DevOps: 开发运维一体化

软件演化与版本



配置管理

正向保证准确发布

反向保证准确回溯



配置项目	标准版 (14.98万元)	豪华版 (16.78万元)	旗舰版 (18.98万元)
LED日间行车灯	—	●	●
车门镀铬把手	—	●	●
驾驶位一键升降	—	●	●
一键启动	—	●	●
倒车雷达	—	●	●
主驾驶位座椅电动调节	—	●	●
USB充电口	—	●	●
皮质座椅	—	●	—
真皮座椅	—	—	●
前排座椅加热/通风	—	—	●
电动天窗	—	●	●
后排出风口	—	●	●
车载导航	—	●	●
定速巡航	—	—	●
方向盘加热	—	—	●
侧安全气囊	—	●	●
驾驶座膝部安全气囊	—	—	●
TPMS胎压监测系统	—	—	●



客户1
标准版



客户2
豪华版



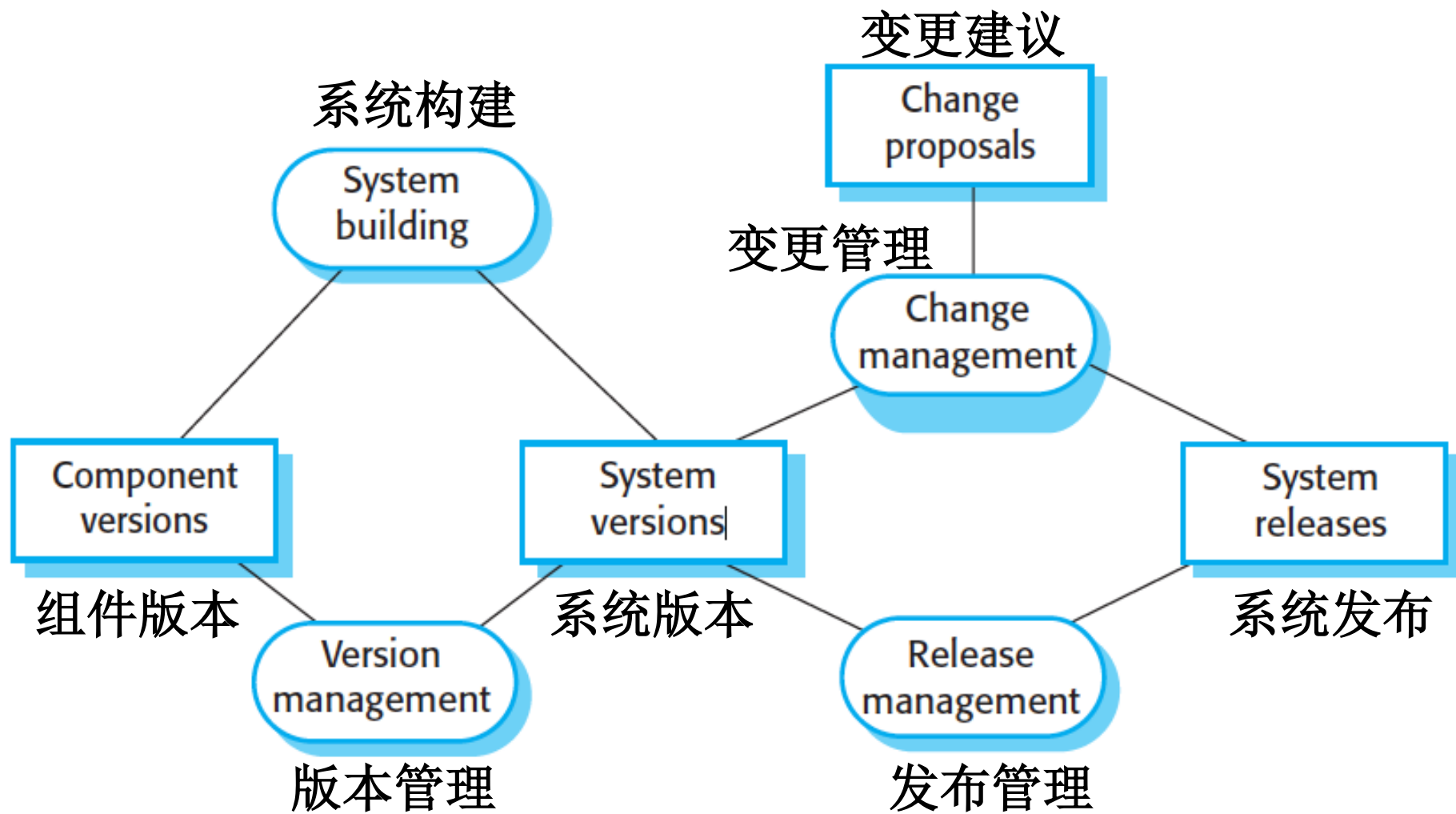
客户3
旗舰版

配置管理

- **版本管理**：跟踪组件的多个版本，确保不同开发者修改不会彼此干涉
- **系统构建**：对程序组件、数据和库进行装配，然后编译链接成可执行系统
- **变更管理**：跟踪变更请求，估算变更开销及其影响，作出变更决定
- **发布管理**：准备对外发布的软件，持续跟踪已经发布供客户使用的系统版本

配置管理是软件开发能力成熟度的基本要求

配置管理活动

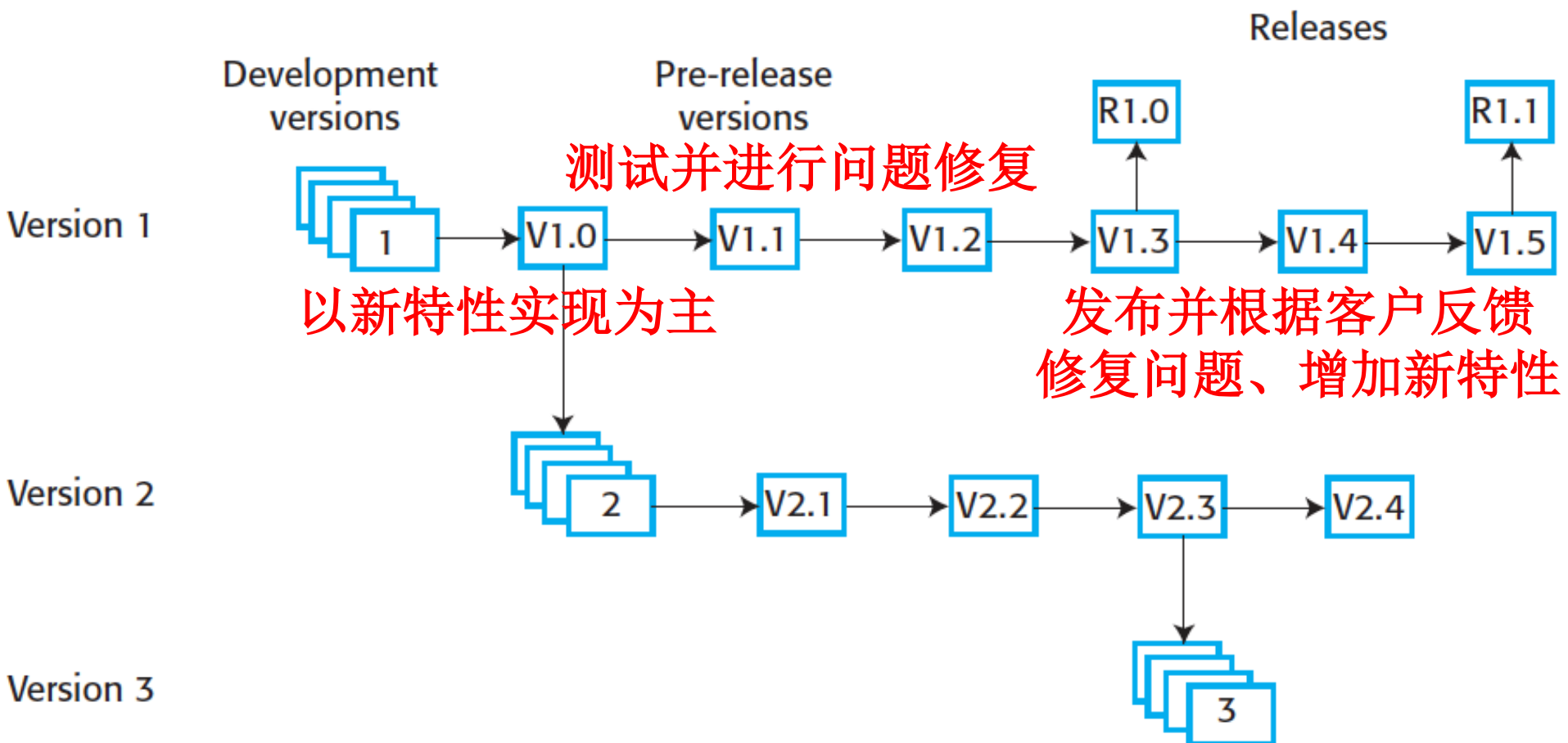


多版本系统开发

开发阶段

系统测试阶段

发布阶段



配置管理相关术语

Baseline 基线

Branching 分支

Codeline 代码线

Configuration (version) control 配置控制

Configuration item 配置项

Mainline 主线

Merging 合并

Release 发布版本

Repository 仓库

System building 系统构建

Version 版本

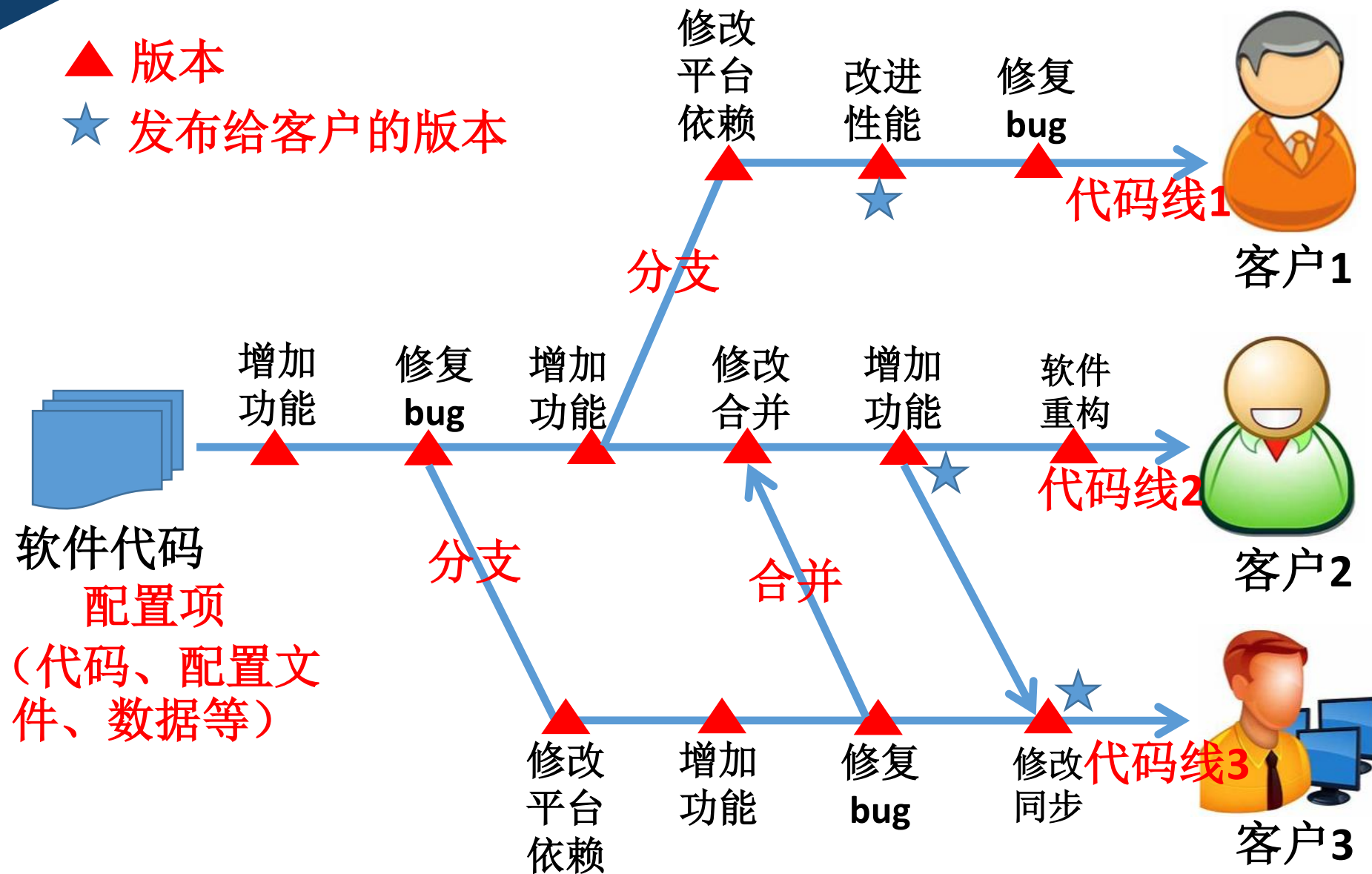
Workspace 工作空间

详见《软件工程》第25章

配置管理相关术语示意

▲ 版本

★ 发布给客户的版本



版本管理 (Version Management)

- 跟踪软件组件或配置项以及使用这些组件的系统的不同版本
- 确保由不同开发者做出的变更不会相互干扰

管理代码线和基线的过程

如果没有版本管理

1#施工图.dwg
1#施工图改.dwg
1#施工图完成版.dwg
1#施工图完成版改1.dwg
1#施工图完成版改2.dwg
1#施工图最终版.dwg
1#施工图最终版改1.dwg
1#施工图最终版改2.dwg
1#施工图最最终版.dwg
1#施工图最最终版改.dwg
1#施工图绝对不改版.dwg
1#施工图绝对不改版改1.dwg
1#施工图绝对不改版改2.dwg

文档库
work



readme.doc



readme -
修改.doc



readme -
副本.doc



readme -
副本
(2).doc



readme -
0709.doc



readme -
0712.doc

代码线与基线

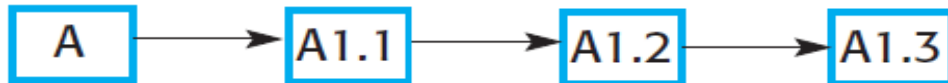
- 代码线（Code line）：源代码（组件、文件）版本的序列，每个版本都是在某个早期版本基础上修改得到的
- 基线（Baseline）：包含组成一个系统的各个组件的特定版本以及所使用的库的描述和配置文件等
- 主线（Mainline）：是由一个原始基线发展而来的系统版本序列

基线的作用

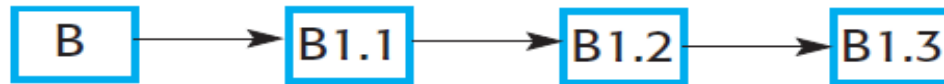
- 基线：系统整体配置的里程碑版本快照
- 基线举例：开发基线、测试基线、发布基线（针对不同客户可以有不同基线）
- 基线的作用
 - ✓ 重现：系统的完整配置版本，例如在接到客户问题报告时重现对应的系统版本
 - ✓ 追踪：建立需求、设计、实现、测试等不同阶段开发制品之间的追踪关系
 - ✓ 报告：以基线为依据报告系统开发的整体状态，例如开发进度、目前实现的需求等

代码线与基线示例

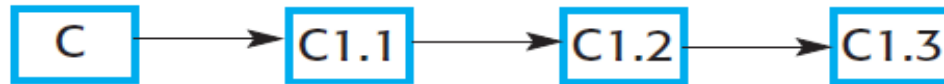
Codeline (A)



Codeline (B)



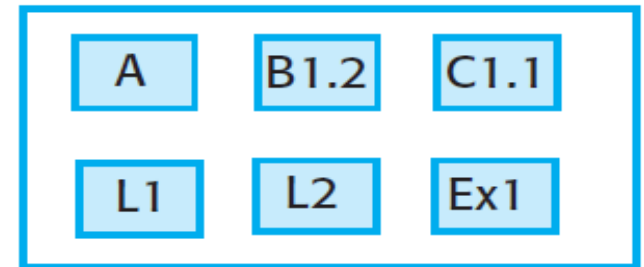
Codeline (C)



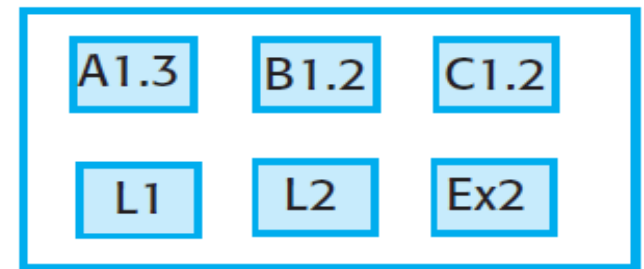
Libraries and external components



Baseline - V1



Baseline - V2



Mainline

版本管理的主要任务

- 版本和发布版本识别
- 变更历史记录
- 支持并行的独立开发
- 支持获取完整的项目制品
- 存储管理

集中式与分布式版本控制系统

- 集中式系统

- ✓ 单个主存储库维护软件组件的所有版本
- ✓ 例如, Subversion

- 分布式系统

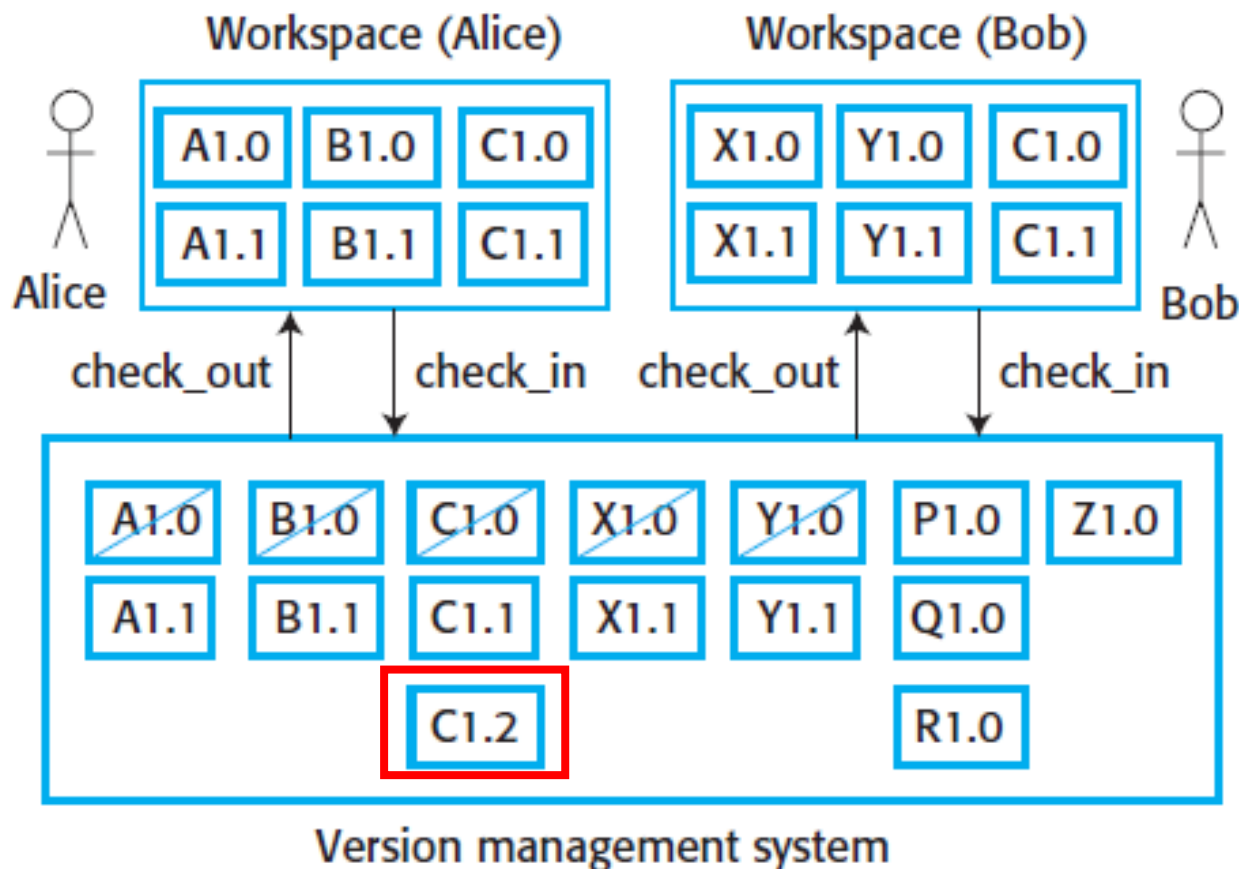
- ✓ 存在多个用于存储组件版本的存储库
- ✓ 例如, Git

版本控制系统 (Version Control System)

集中式系统（检出/检入模式）

个人工作空间

个人工作空间



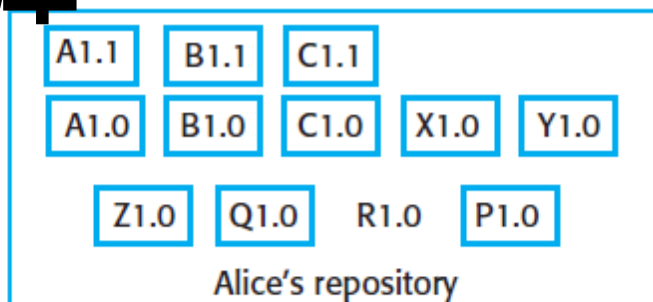
检出单个文件
修改后检入形
成新版本

集中式的版本管理系统

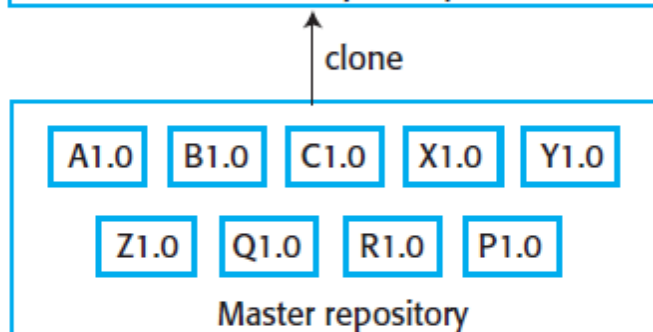
分布式系统（克隆模式）

私有仓库

Alice

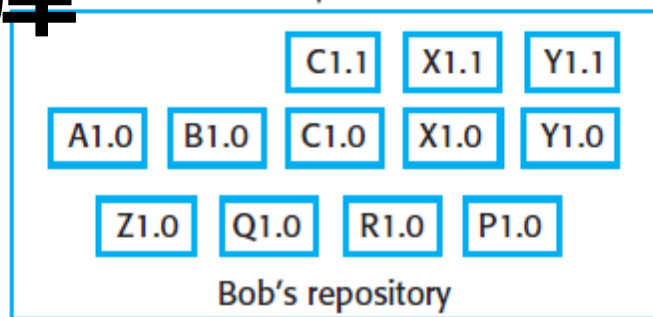


主仓库



私有仓库

Bob



- 1) 克隆项目库
- 2) 本地修改后提交（**commit**），更新私有库
- 3) 将变更推送（**push**）到项目库指定分支
- 4) 将项目库中的更新拉取（**pull**）回本地并与本地修改进行合并

分布式版本控制系统的优势

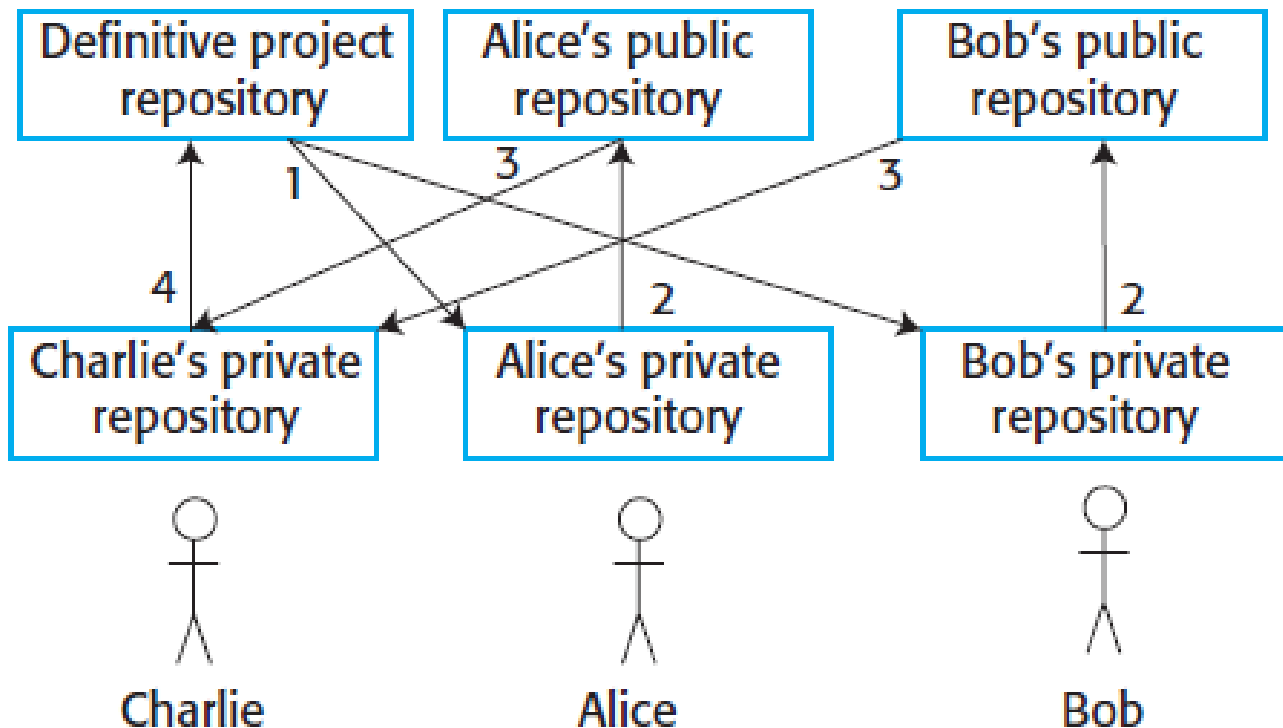
- 提供了仓库备份机制
- 允许离线工作（本地提交）
- 完善的本地项目支持

开源项目开发：社会化协作

1. 开发参与者克隆项目库
2. 开发参与者推送本地变更到个人公共库
3. 集成管理员将变更拉到本地库中进行测试
4. 集成管理员确定接受修改后更新最终项目库

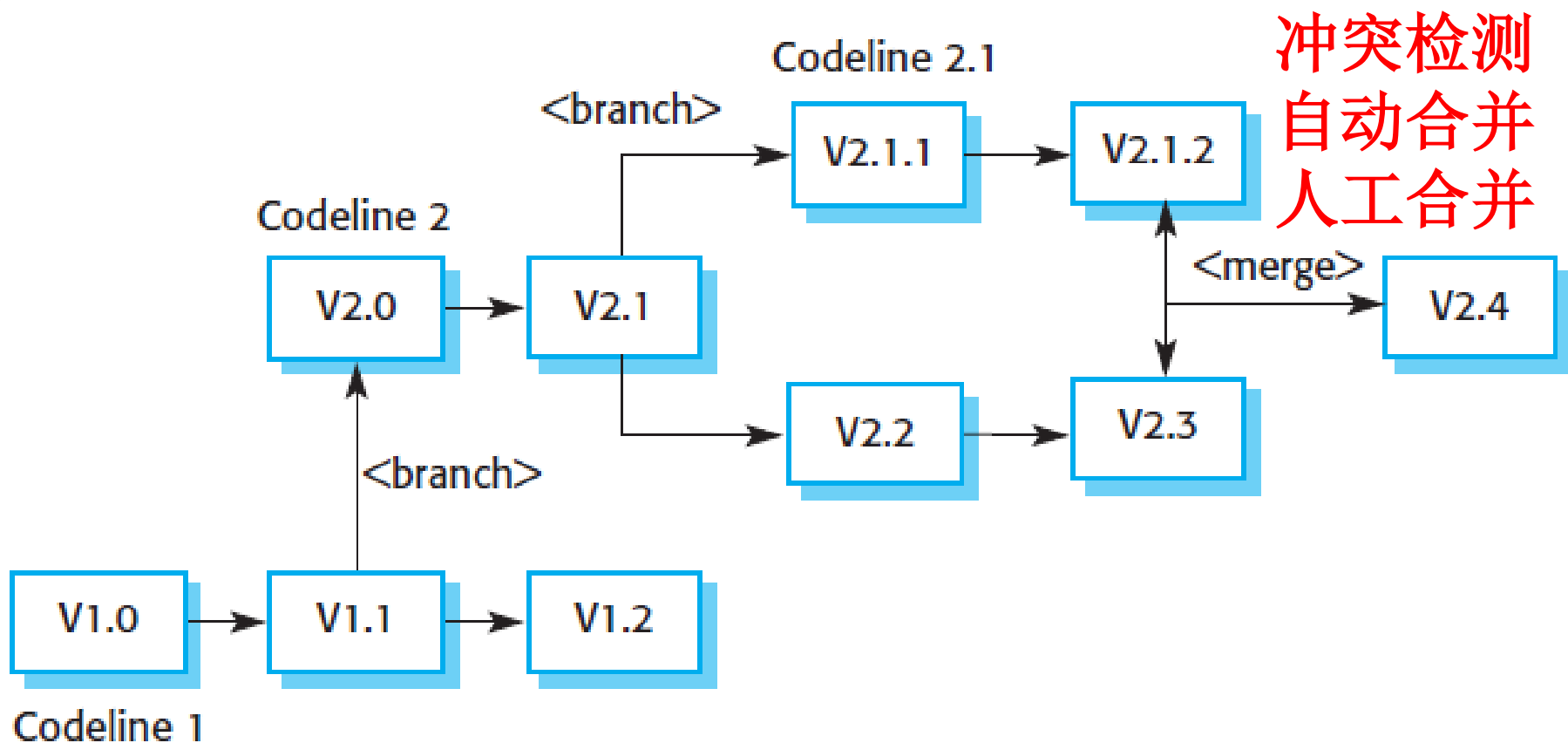


github
SOCIAL CODING



集成管理者 开发参与者 开发参与者

分支与合并



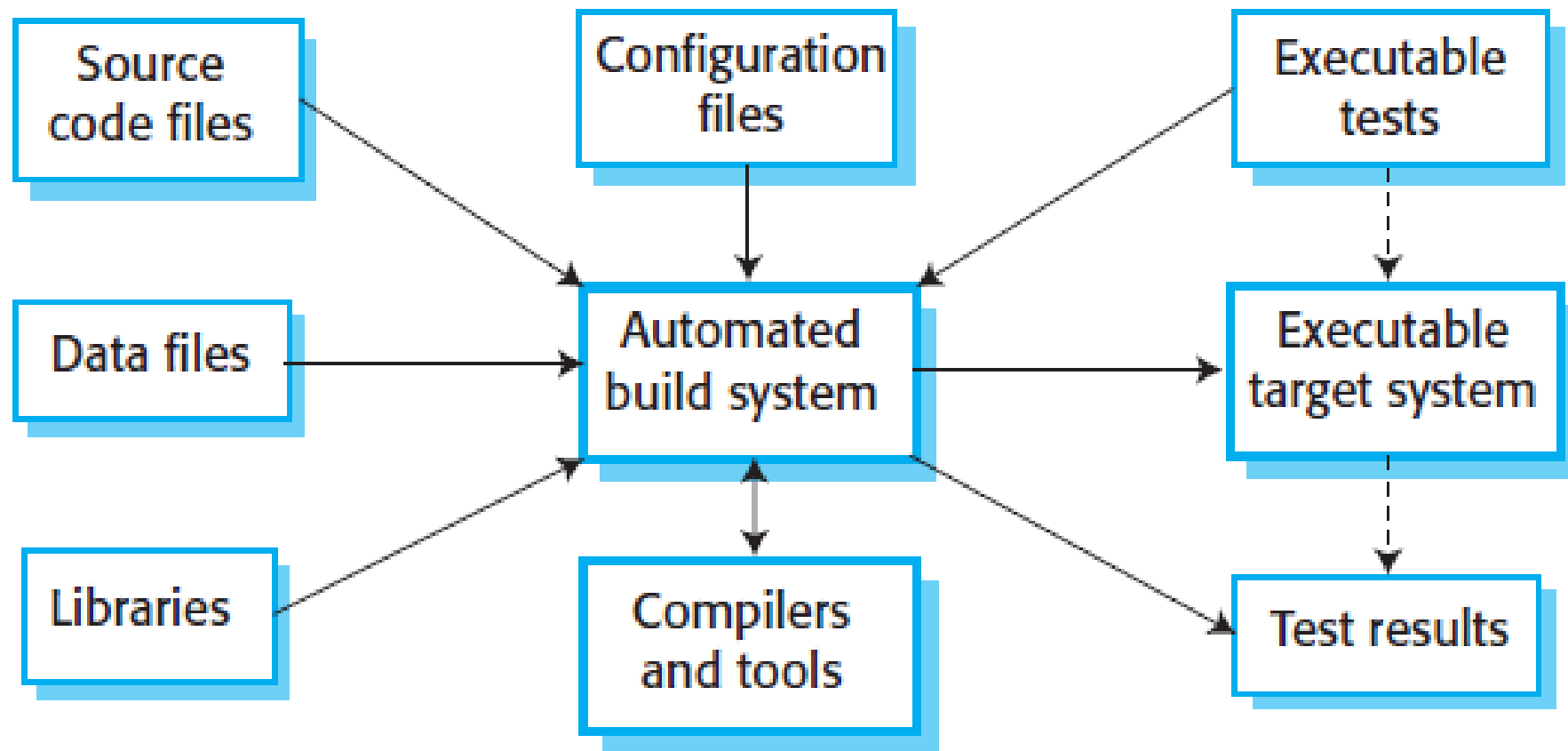
允许不同的开发者在不同分支上独立工作，减少干扰

例如，一个探索性的代码修改不会破坏可运行的系统版本

系统构建 (System Building)

将软件组件、外部库、配置文件等编译和链接成一个完整、可执行的程序

需要集成系统构建工具和版本控制工具，使得构建过程可以从版本库中获取组件版本



构建系统主要功能

- 按需生成构建脚本
- 版本管理系统集成
- 最小化再编译
- 可执行系统创建
- 测试自动化
- 构建及测试结果报告
- 文档生成

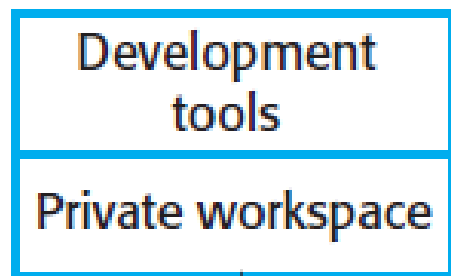
构建脚本

- 定义了所要构建的系统
 - ✓ 组件及其版本信息
 - ✓ 组件间依赖关系
 - ✓ 编译和链接工具的版本信息
- 包括配置详细描述，可以使用配置描述语言

开发、构建、部署

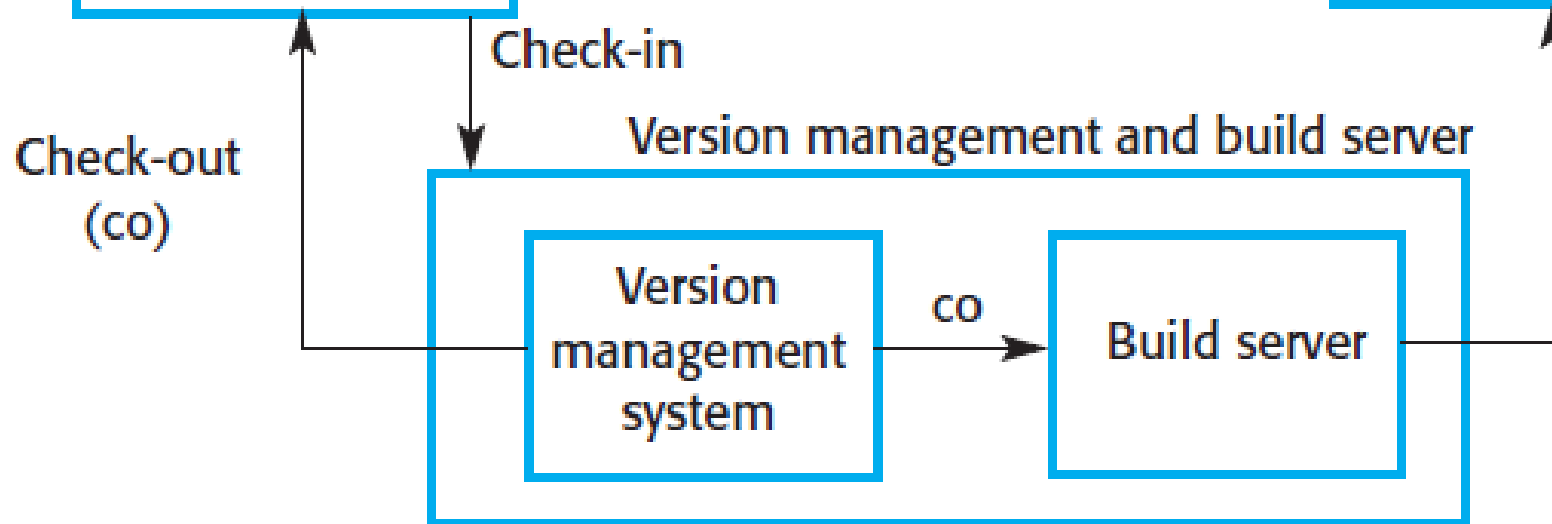
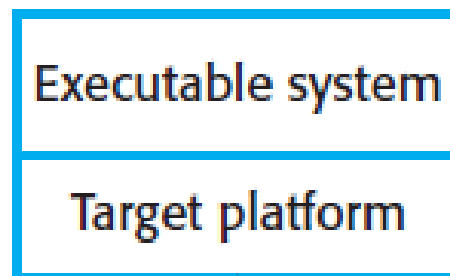
本地开发环境

Development system



目标系统（运行）环境

Target system



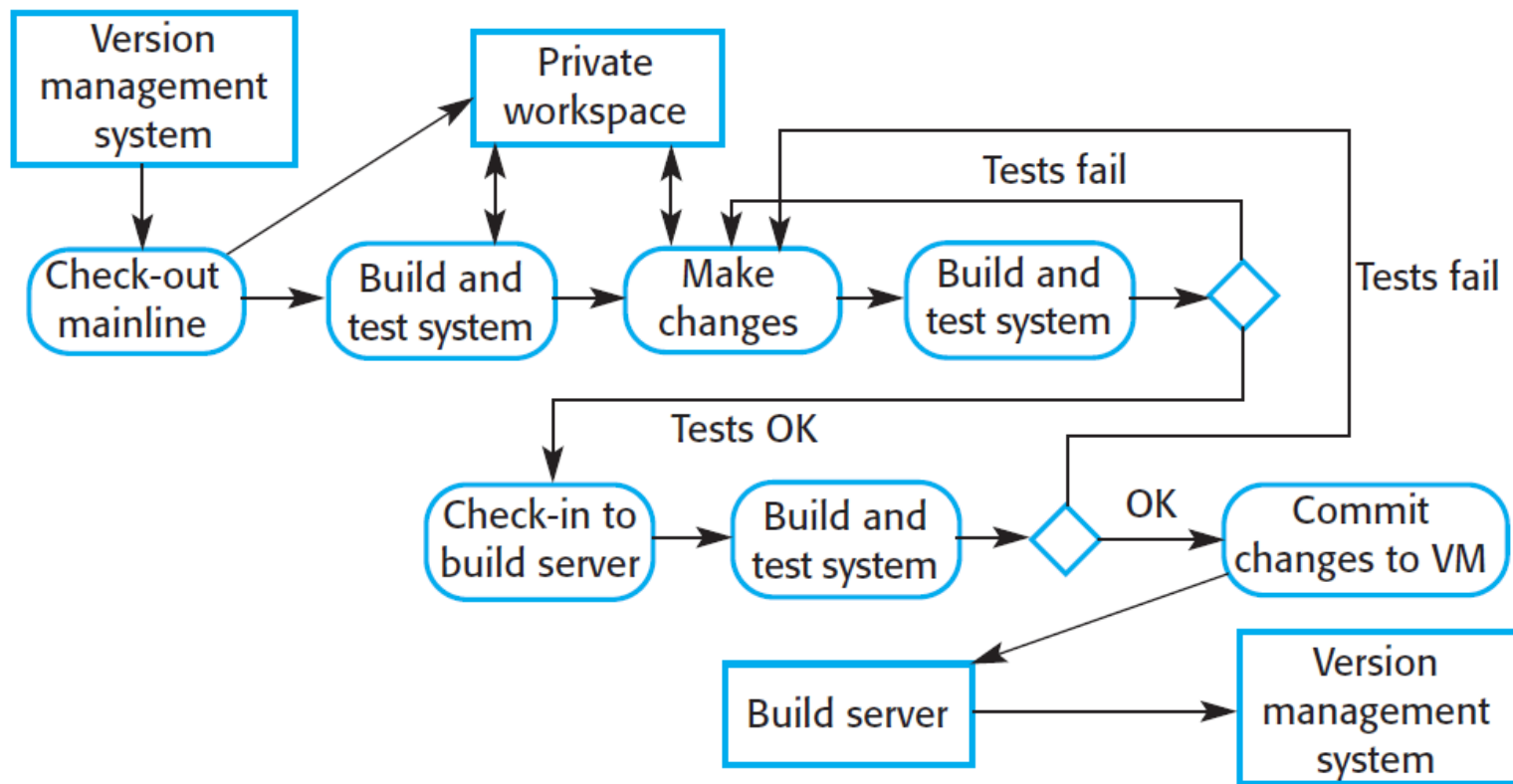
版本管理和构建服务器

持续集成

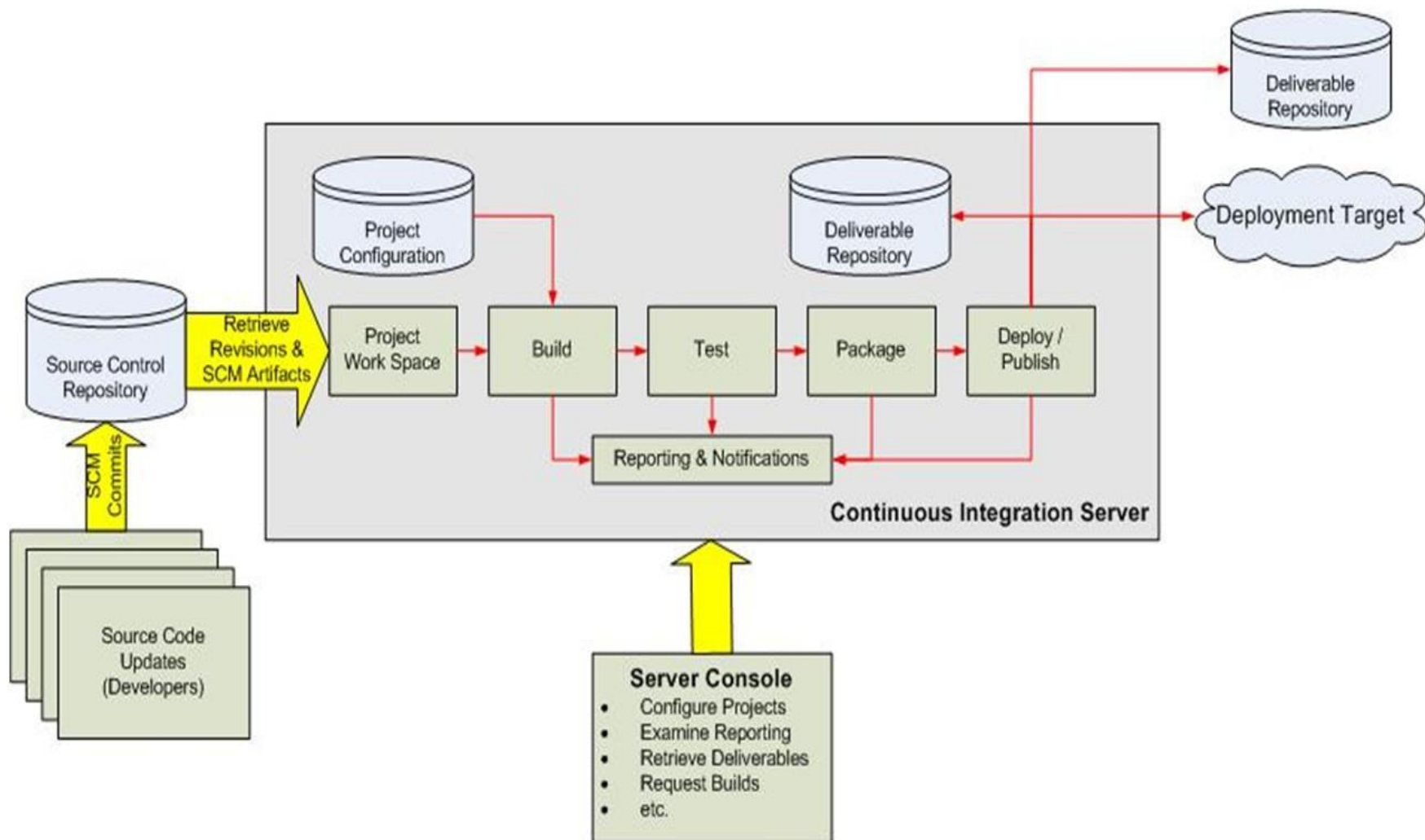
敏捷开发实践的重要组成部分：基本思想是在对源代码做出小的改变后频繁重建主线版本

确保主线版本可工作：一旦构建失败必须马上修复

频繁的质量反馈：代码度量、缺陷检测、自动化测试等



持续集成系统的典型结构



相关工具

Jenkins: 基于Java的持续集成工具



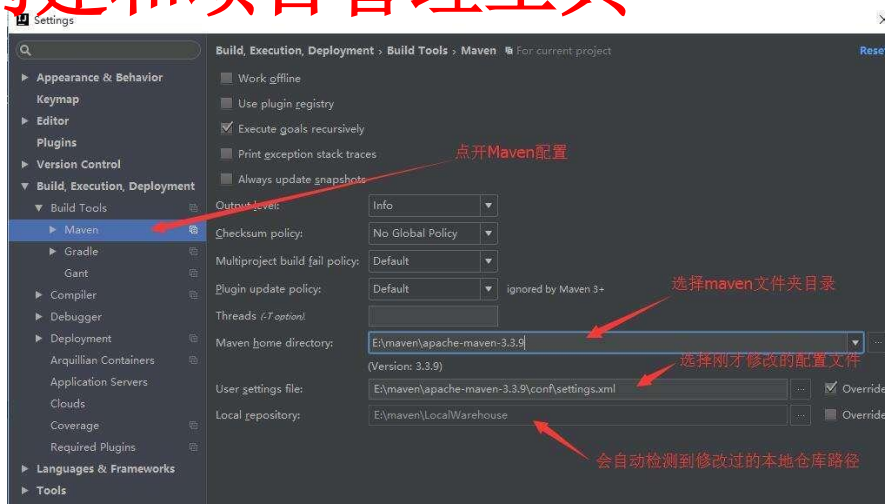
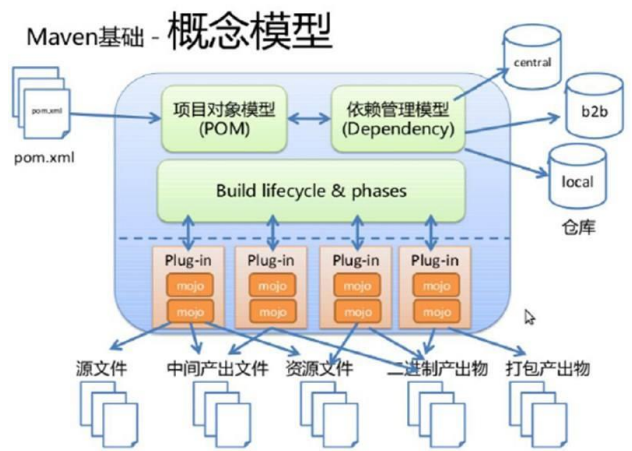
The screenshot shows the Jenkins dashboard. On the left is a sidebar with navigation links: 新建 (New), 用户 (Users), 任务历史 (Task History), 系统管理 (System Management), My Views, Credentials, and New View. The main area displays a table of build jobs. Below the table are sections for '构建队列' (Build Queue) and '构建执行状态' (Build Execution Status).

S	W	名称 ↓	上次成功	上次失败	上次持续时间
		apiknowledgegraph	没有	无	无
		ggg	没有	无	无
		jichaofdu	没有	25 分 - #3	1 毫秒
		testmaven	没有	3 分 22 秒 - #4	5.5 秒

构建队列: 队列中没有构建任务

构建执行状态: 1 空闲, 2 空闲

Maven: 程序构建和项目管理工具



The screenshot shows the 'Maven' settings in an IDE. The 'Maven' section is expanded, showing various configuration options. Red arrows point to specific settings with labels:

- 点开Maven配置 (Click on Maven configuration)
- 选择maven文件夹目录 (Select Maven file folder)
- 选择刚才修改的配置文件 (Select the configuration file just modified)
- 会自动检测到修改过的本地仓库路径 (It will automatically detect the modified local repository path)

构建脚本示例

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5
6     <groupId>fdse.microservice</groupId>
7     <artifactId>ts-travel-service</artifactId>
8     <version>1.0</version>
9     <packaging>jar</packaging>
10
11     <name>ts-travel-service</name>
12     <description>ts-travel-service</description>
13
14     <properties>
15         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
16         <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
17         <java.version>1.8</java.version>
18     </properties>
19
20     <dependencies>
21         <dependency>
22             <groupId>org.springframework.boot</groupId>
23             <artifactId>spring-boot-starter-data-mongodb</artifactId>
24         </dependency>
25     </dependencies>
26
27     <build>
28         <plugins>
29             <plugin>
30                 <groupId>org.springframework.boot</groupId>
31                 <artifactId>spring-boot-maven-plugin</artifactId>
32             </plugin>
33         </plugins>
34     </build>
35
36 </project>
```

项目所属组织的名称
项目版本
打包方式（jar等）等信息

项目的名称与描述

项目的属性，
比如编码方式
或者Java版本

本项目使用的库文件依赖

Build的配置，比如使用的插件

Maven的POM（项目对象模型）文件

阅读建议

- 《软件工程》第25章“配置管理”（25.1–25.2）

✓ 支持视频：<http://software-engineering-book.com/videos/software-management/>

快速阅读后整理问题
在QQ群中提出并讨论

CS2001

软件工程

End

2. 个人开发技能
— 软件演化与配置管理