



软件部署与交付

清华大学软件学院 刘强





1

软件部署与交付工作

2

持续集成与交付

3

课程实验项目验收

项目接近尾声

当开发阶段接近尾声的时候，项目组的气氛也最高涨！



项目如何交付？

项目交付工作

项目验收交付时，还有三项工作在等着：部署、培训、验收



验收后的项目才正式进入“维护”阶段

软件部署

软件部署是将软件系统部署到客户方的计算机系统上，协助客户准备基础数据，使软件系统顺利上线运行。

- 保证软件符合需求，质量过关
全面做好测试工作（集成测试、功能测试、性能测试）
- 制定部署计划
要发布的代码版本、系统基础设施、数据库创建、基础数据准备
- 准备好程序代码和相关文档
用户手册以及其他系统文档（如需求说明书、设计文档等）

客户培训

在系统部署完成、基础数据准备齐全之后，应该组织客户培训，使其掌握对软件系统的使用和操作。



- 选择合适的培训人员
经验丰富、了解业务和系统
- 准备好培训内容
不要临时抱佛脚
- 制定培训计划
与客户沟通协调，安排时间

项目验收

客户对系统进行验收测试，包括范围核实（用户需求是否全部实现）和质量核实（质量属性是否满足要求）。



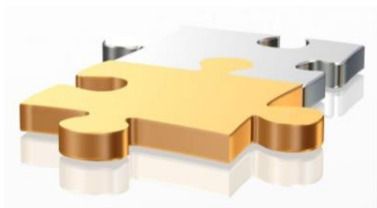
客户在验收报告上签字，一切尘埃落定。对于大中型项目，还会有一个签字验收仪式。

软件部署定义

软件部署是软件生命周期中的一个重要环节，属于软件开发的后期活动，即通过配置、安装和激活等活动来保障软件制品的后续运行。



部署技术影响着整个软件过程的运行效率和投入成本，软件系统部署的管理代价占到整个软件管理开销的绝大部分。



软件配置过程极大地影响着软件部署结果的正确性，应用系统的配置是整个部署过程中的主要错误来源。

软件部署作用

基本目的：支持软件运行，满足用户需求，使得软件系统能够被直接使用。

- 保障软件系统的正常运行和功能实现
- 简化部署的操作过程，提高执行效率
- 同时还必须满足用户在功能和非功能属性方面的个性化需求

通用性
灵活性

可靠性
正确性

过程化
自动化

软件部署模式



面向单机软件的部署模式：包括安装、配置和卸载，该部署模式主要适用于运行在操作系统之上的单机类型的软件。

部署操作的执行功能主要通过脚本编程的方式来实现，以脚本语言编写的操作序列来支持诸如软件的安装和注册。

基本活动：

打包

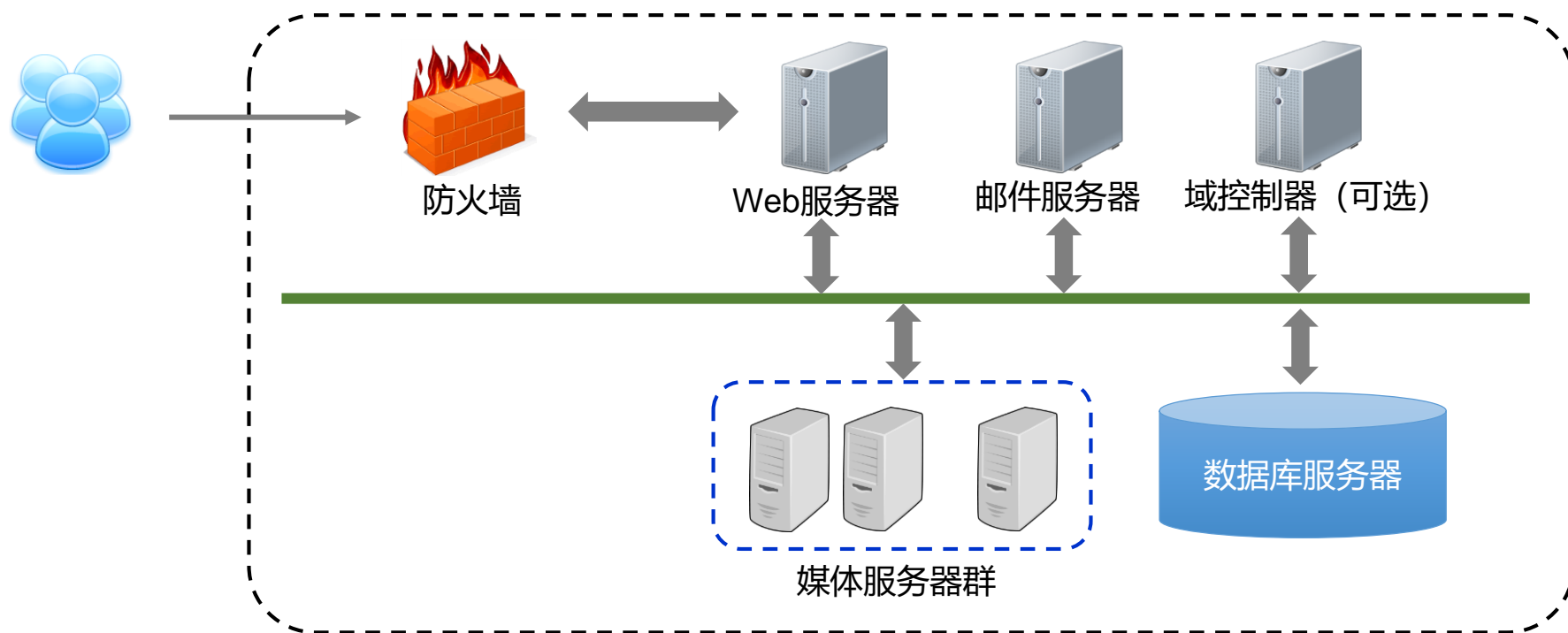
安装

更新

激活

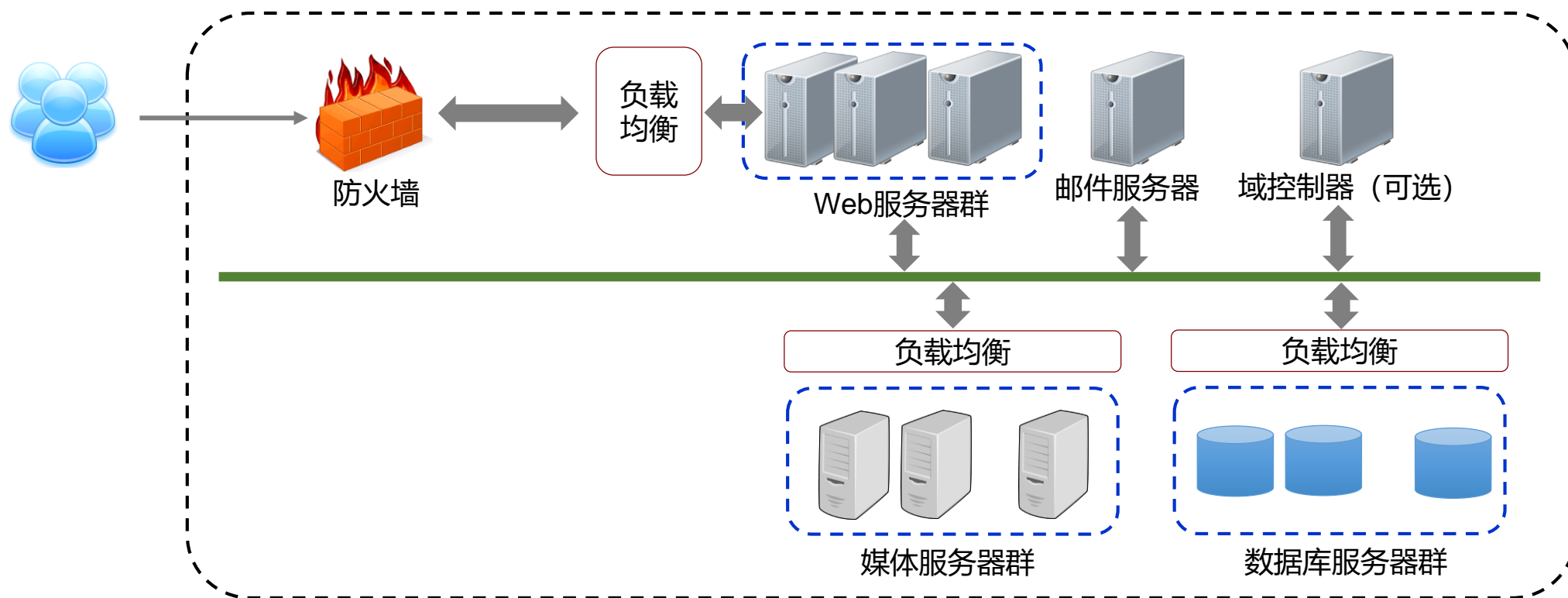
软件部署模式

集中式服务器应用部署：适用于用户访问量小（500人以下）、硬件环境要求不高的情况，诸如中小企业、高校在线学习、实训平台等。



软件部署模式

集群式服务器应用部署：适用于并发用户访问量大（10000以上）、对系统稳定性和性能要求高的分布式平台部署。





1

软件部署与交付工作

2

持续集成与交付

3

课程实验项目验收

软件交付的演进历程

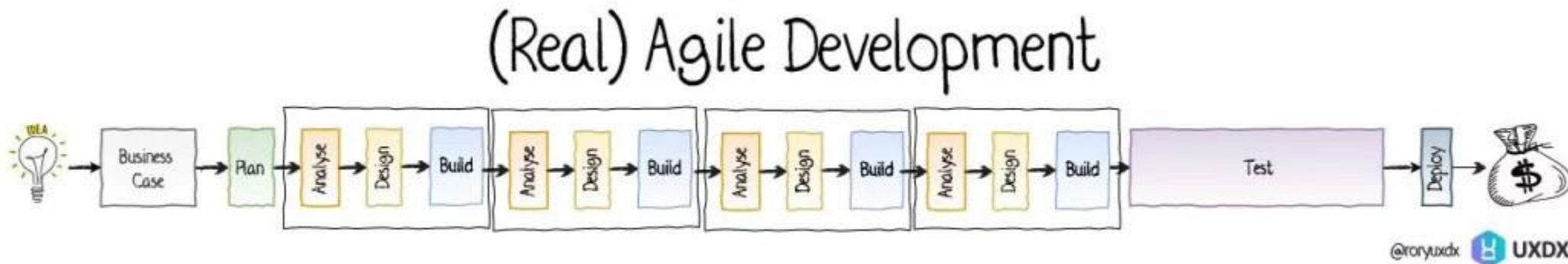


软件开发的一些关键假设：

<https://blog.uxdx.com/the-evolution-of-software-delivery/>

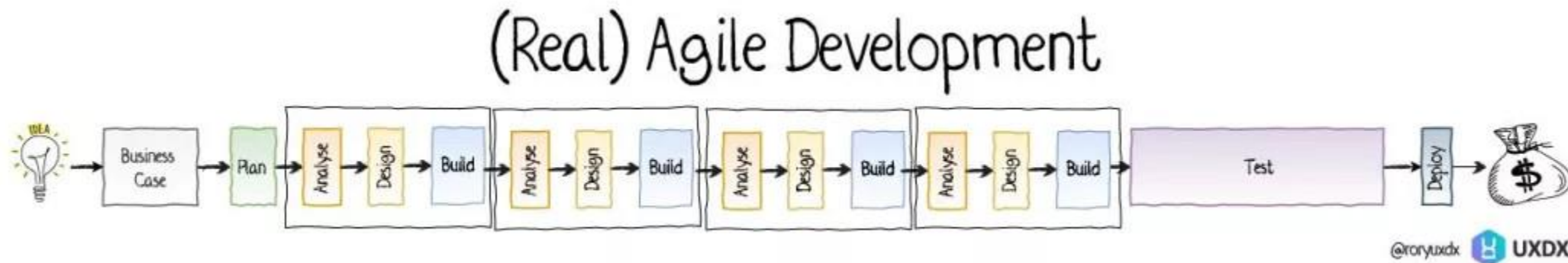
- 软件开发需要很长的时间
- 软件发布不会很频繁
- 软件构建之后很难进行更改，所以要确保第一次把事情做对
- 软件开发需要很多不同的、成本高昂的技能集

软件交付的演进历程



- 设计可以随着迭代过程中的学习进行更改，因此只做每次迭代所需的设计就会有利于减少可能的返工。
- 架构师定义好实践、原则和强制措施，确保设计符合指定的约束条件，而不是定义出一个固定的架构。
- 开发工程师被赋予更多的责任，需要针对问题给出最佳解决方案，而不仅仅是按照需求规格说明书进行软件构建。

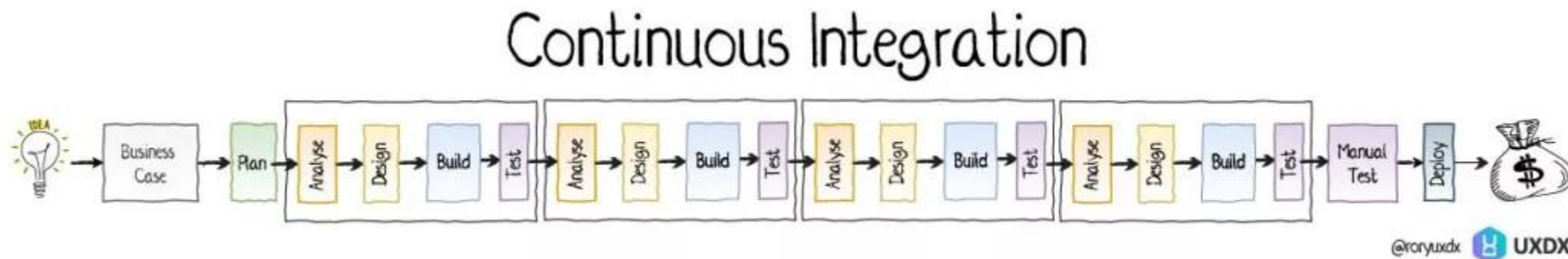
软件交付的演进历程



软件开发的一些关键假设：

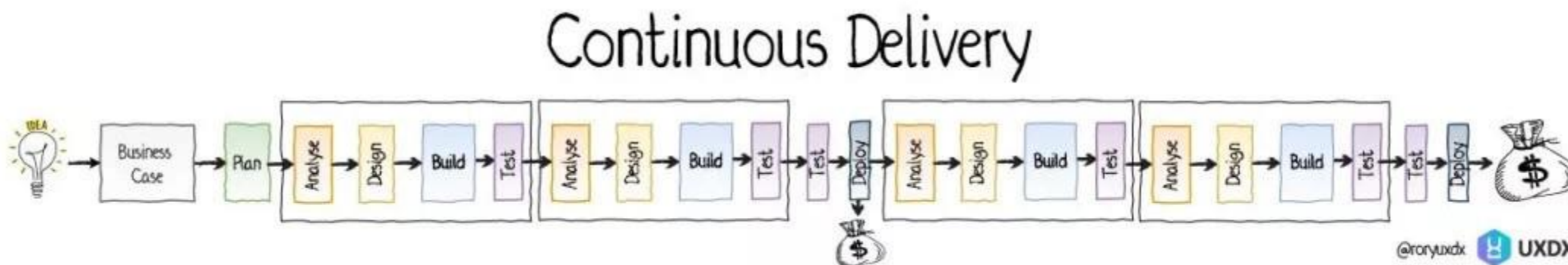
- 软件开发需要很长的时间
- 软件发布不会很频繁
- 前期无法预知最终产品，因此将大型开发分解为小部分，并在项目进行中不断学习
- 软件开发需要很多不同的、成本高昂的技能集
- 开发团队负责在一定的边界条件下进行设计

软件交付的演进历程



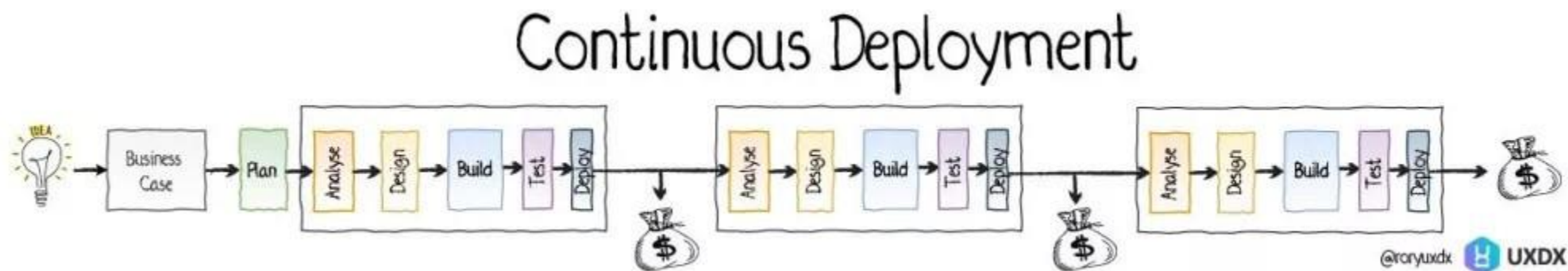
- 互联网时代的开发需要快速响应用户的变化，频繁地将软件的新版本进行交付。
- 自动化测试让人们可以更频繁地验证客户的想法，它带来的益处超过了其成本。自动化测试周期让团队在整个开发过程中，可以持续开发“可发布”的代码。
- 100% 的自动化测试是个很难达到的目标，因此在软件发布前还会有小量的手动验收测试，但其质量已经足够验证解决方案。

软件交付的演进历程



- 持续集成的好处在于代码在任何阶段都是可部署的，但是人们仍然需要一些时间在环境间移动代码并完成部署步骤。
- 在理想情况下，团队应该在每次迭代之后，发现是否有非功能性需求的问题需要解决。但是，手动部署占用时间很长，因此部署也应该自动化。

软件交付的演进历程



功能发布控制：

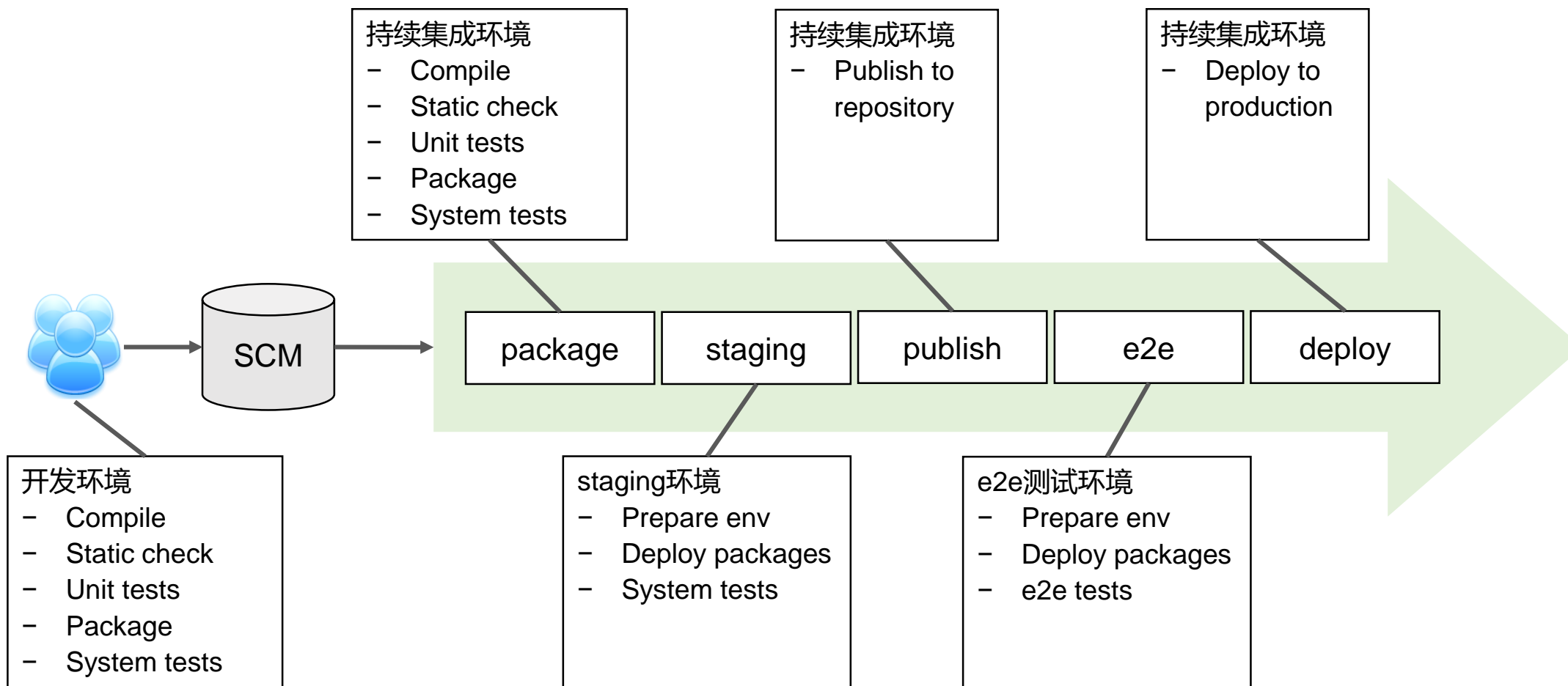
- 标识后面开发的新功能，这样可以部分发布已完成的功能，但让它们处于关闭状态。用户意识不到这些新功能，这些代码不会对其它部分造成破坏。
- 将功能置于一个动态的转换中，我们可以开始在测试环境下进行 A/B 测试和多元测试。我们可以动态打开或关闭某个功能，因此可以在生产环境进行用户验收测试。

软件交付原则



- 为软件的发布创建一个可重复且可靠的过程
- 将几乎所有的事情自动化（构建、部署、测试、发布）
- 把所有的东西都纳入版本控制（需求文档、测试脚本、自动化测试用例、网络配置脚本、部署脚本、数据库创建、升级、回滚和初始化脚本、库文件、应用程序依赖的软件集、工具链及技术文档等）
- 找到流程中最痛苦的事情，并提交频繁地进行。如果集成最痛苦，那应在开始阶段就不断进行集成和测试；如果发布痛苦，每次提交并通过自动化测试后就进行发布
- 用户故事只有已发布才算完成，交付成果属于每个成员，交付前每个成员都为其负责
- 持续改进交付过程：整个团队召开回顾会议，提出改进方向及方法，每个改进点应该同一个人负责跟踪，确保改进被执行，下一次回顾会议汇报结果。

产品发布系统



持续集成与交付的过程

- **提交：**开发者向代码仓库提交代码
- **测试：**代码仓库对commit操作配置钩子（hook），只要提交代码或者合并进主干，就会运行自动化测试

测试的种类：

- 单元测试：针对函数或模块的测试
- 集成测试：针对整体产品的某个功能的测试，又称功能测试
- 端对端测试：从用户界面直达数据库的全链路测试

说明：

- 第一轮至少要跑单元测试
- 通过第一轮测试，代码可以合并进主干，系统可以达到交付要求

持续集成与交付的过程

- **构建：**将源码转换为可运行代码，包括安装依赖、配置各种资源等
常见的构建工具：Jenkin, Travis, Codeship, Striber
- **第二轮测试：**第二轮是全面测试，单元测试和集成测试都会跑，有条件的话也要做端对端测试。所有测试以自动化为主，少数无法自动化的测试用例则人工执行。
说明：新版本的每一个更新点都必须测试到。如果测试的覆盖率不高，进入后面的部署阶段后，很可能会出现严重的问题。
- **部署：**通过了第二轮测试，当前代码就是一个可以直接部署的版本（artifact）。将这个版本的所有文件打包（可使用docker容器技术打包成镜像）存档，发到生产服务器，生产服务器将从docker-registry 拉取镜像进行部署。



1

软件部署与交付工作

2

持续集成与交付

3

课程实验项目验收

课程实验项目交付

具体安排:

- 12月19日 (周四) 课堂上各小组进行汇报和展示
 - 每个小组汇报时长为10分钟, 包括PPT汇报和系统演示
 - 注意突出自己的特点和长处
- 1月5日 (周日) 在网络学堂提交最终成果
 - 包括源代码库链接、开发过程文档压缩包、交付文档、演示PPT、产品介绍演示视频、个人总结等
 - 在服务器上部署各小组开发的系统

课程实验项目交付

开发过程文档包括但不限于以下部分：

- 需求原型与交互设计文档
- 团队会议纪要
- 代码编写规范
- 软件测试报告
- 开发过程产生的其他文档（如果有的话）

说明：按照已有资料提交，不要事后补交



课程实验项目交付



交付文档包括但不限于以下内容：

- 交付产品：服务器IP或域名、小程序码（如有的话）、管理员账号和密码
- 产品目标：所开发的系统功能和性能方面的要求
- 开发组织管理：过程管理、人员分工、开发环境、配置管理等
- 系统设计：前端交互、后端模块、接口规范、数据库设计等
- 重点和难点问题及其解决方法，以及核心算法（如有的话）等
- 测试总结：功能测试方法以及缺陷汇总、性能测试方法与结果
- 系统部署：部署方法与规范等

课程实验项目交付

评价标准：

- | | |
|--------------|-----|
| • 软件设计与代码质量 | 20分 |
| • 系统功能与界面设计 | 20分 |
| • 产品测试与部署 | 20分 |
| • 团队协作与代码管理 | 20分 |
| • 课堂汇报与演示 | 10分 |
| • 交付文档 | 10分 |
| • 个人贡献度：规则待定 | |

说明：

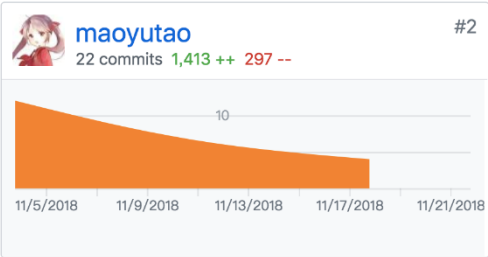
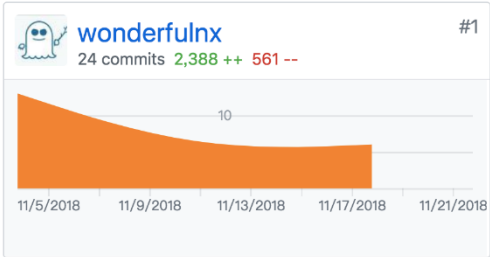
- 开发工作的衡量和评价主要针对每个小组自己做的部分
- 产品测试与部署以是否达到上线提测为衡量标准，系统可以成功部署运行
- **在达到上线提测的前提下**，对于技术方案和创意设计方面表现突出的产品，可以获得最多5分的加分。

学生项目管理示例

Contributions to master, excluding merge commits



Contributions to master, excluding merge commits



迭代一完成

Closed 2 minutes ago 100% complete

迭代一基本完成，可以做简单的预约操作

☐

0 Open

8 Closed

<input type="checkbox"/>	<div>前端房间管理相关页面的编写</div> <div>#17 by wonderfulnx was closed 3 minutes ago</div>
<input type="checkbox"/>	<div>苏总负责管理端前端开发（纳鑫辅助）</div> <div>#16 by wonderfulnx was closed 4 minutes ago</div>
<input type="checkbox"/>	<div>王老板负责微信小程序前端开发（毛总辅助）</div> <div>#14 by wonderfulnx was closed 8 minutes ago</div>
<input type="checkbox"/>	<div>数据库测试数据脚本</div> <div>#11 by maoyutao was closed 2 hours ago</div>
<input type="checkbox"/>	<div>苏总负责管理端前端开发（纳鑫辅助）</div> <div>#15 by wonderfulnx was closed 2 days ago</div>
<input type="checkbox"/>	<div>纳鑫—数据库完善和redis研究</div> <div>#7 by wonderfulnx was closed 2 days ago</div>
<div><div>≡</div><input type="checkbox"/></div>	<div>管理端API</div> <div>#10 by maoyutao was closed 2 days ago</div>

Edit milestone

New issue

Dependencies	Dependents
--------------	------------

These dependencies are defined in TsingE-Backend’s manifest files, such as [requirements.txt](#).

Dependencies defined in requirements.txt 7		
>	<div><div></div>django / django</div>	~> 2.1.1
>	<div><div></div>kraiz / django-crontab</div>	~> 0.7.1
>	<div><div></div>niwinz / django-redis</div>	~> 4.9.0
>	<div><div></div>graphql-python / graphene-django</div>	~> 2.2.0
	<div><div></div>PyMySQL / mysqlclient-python mysqlclient</div>	~> 1.3.13
>	<div><div></div>gusibi / python-weixin</div>	~> 0.3.2
>	<div><div></div>andymccurdy / redis-py redis</div>	~> 2.10.6

1 微信小程序前端开发



- 🔔

王老板负责微信小程序前端开发（毛总辅助）

#14 opened by wonderfulnx

Merge pull request #2 from wonderfulnx/manage_api ...

wonderfulnx committed 3 hours ago

Merge branch 'master' of github.com:wonderfulnx/TsingE-Backend into m... ...

wonderfulnx committed 3 hours ago

chore(model): change redis version, add to requirements

wonderfulnx committed 3 hours ago

fix(view): changed repeat rule from int to string

wonderfulnx committed 4 hours ago

fix(view): fix spell error

wonderfulnx committed 9 hours ago

feat(view): add support for multi change for permission and price

wonderfulnx committed 10 hours ago

adjust(controller):session age

maoyutao committed 12 hours ago

Commits on Nov 19, 2018

chore(view):add python-weixin

wonderfulnx committed 2 days ago

3 管理端前端开发



- 🟢

前端以表格形式呈现琴房占用情况

#20 opened by wonderfulnx

- 🔔

前端房间管理相关页面的编写

#17 opened by wonderfulnx

- 🔔

苏总负责管理端前端开发（纳鑫辅助）

#16 opened by wonderfulnx

feat(view): finished most of the piano room operation

wonderfulnx committed 3 hours ago

fix(*): wipe data before starting query or mutation

SteveYurongSu committed 6 hours ago

feat(OrderUser.vue): query / mutation

SteveYurongSu committed 6 hours ago

feat(UserBlacklist.vue): query / mutation

SteveYurongSu committed 7 hours ago

feat(UserQuery.vue): query / mutation

SteveYurongSu committed 7 hours ago

feat(view): finish room price and room privilege, but without mutatio... ...

wonderfulnx committed a day ago

Commits on Nov 20, 2018

feat(view): add v-for, not done yet

wonderfulnx committed a day ago

feat(*): OrderUser.vue / add some .gql files

SteveYurongSu committed 2 days ago

fix(OrderDetail.vue OrderToday.vue): show user description correctly

SteveYurongSu committed 2 days ago

7 后端开发



- 🔔

后端房间权限和价格mutation的修改

#18 opened by wonderfulnx

- 🔔

小程序端API

#13 opened by maoyutao

- 🔔

django定时任务

#12 opened by maoyutao

- 🔔

管理端API

#10 opened by maoyutao

- 🔔

数据库测试数据脚本

#11 opened by maoyutao



谢谢大家!

THANKS

