

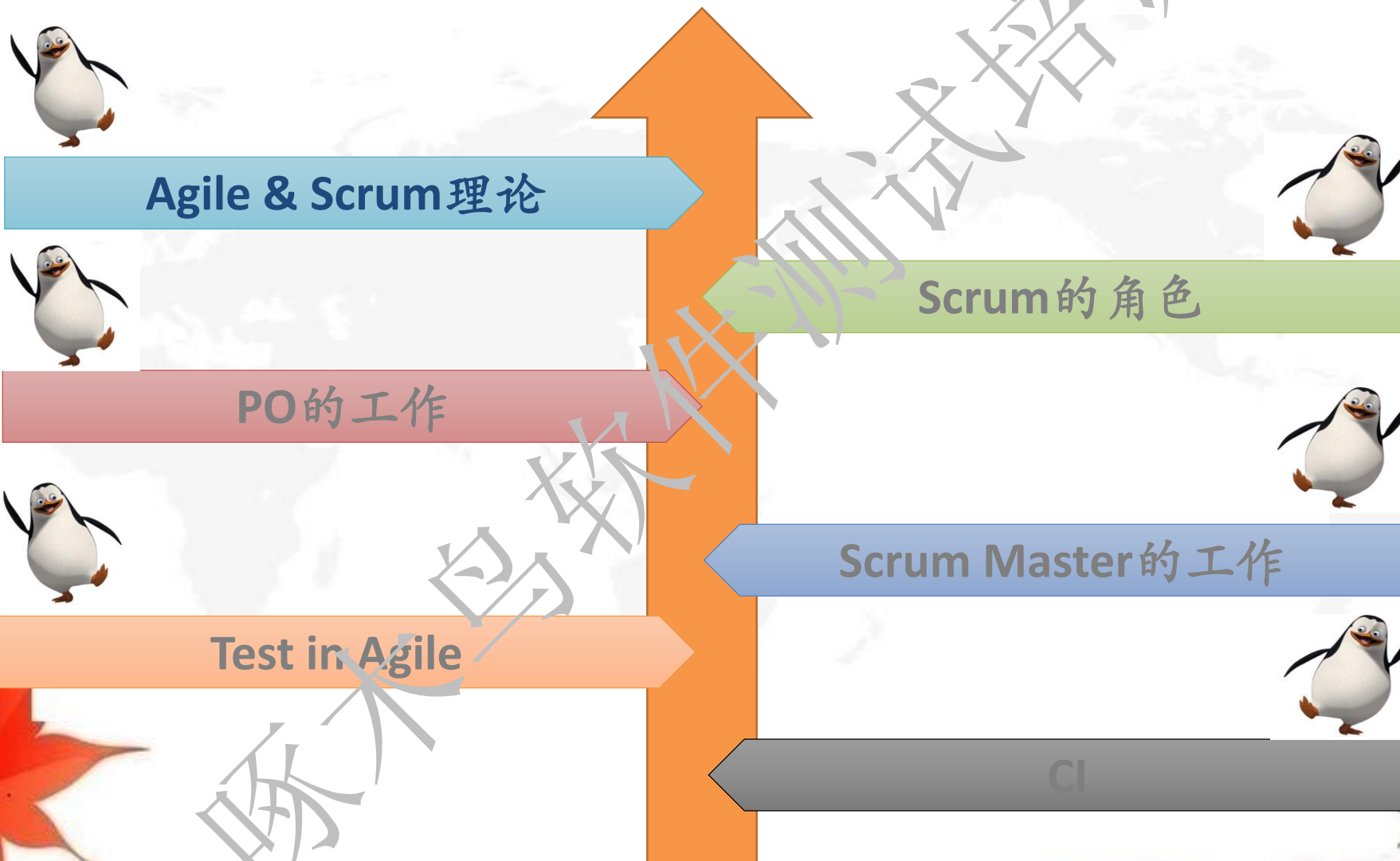


敏捷开发

专业的IT培训专家
顾翔



目录



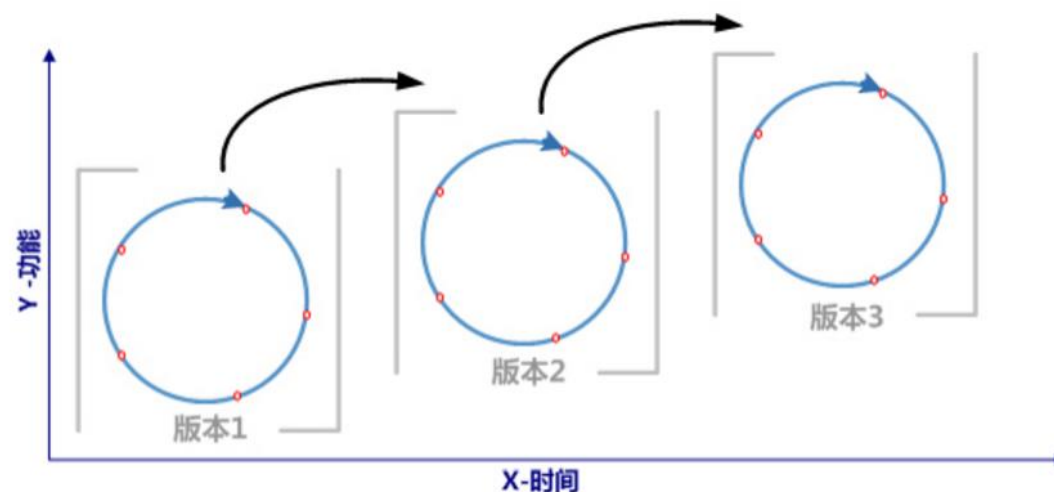
Agile & Scrum理论



敏捷开发(Agile Development)是一种以人为核心、迭代、循序渐进的开发方法。

驱动核心

方法

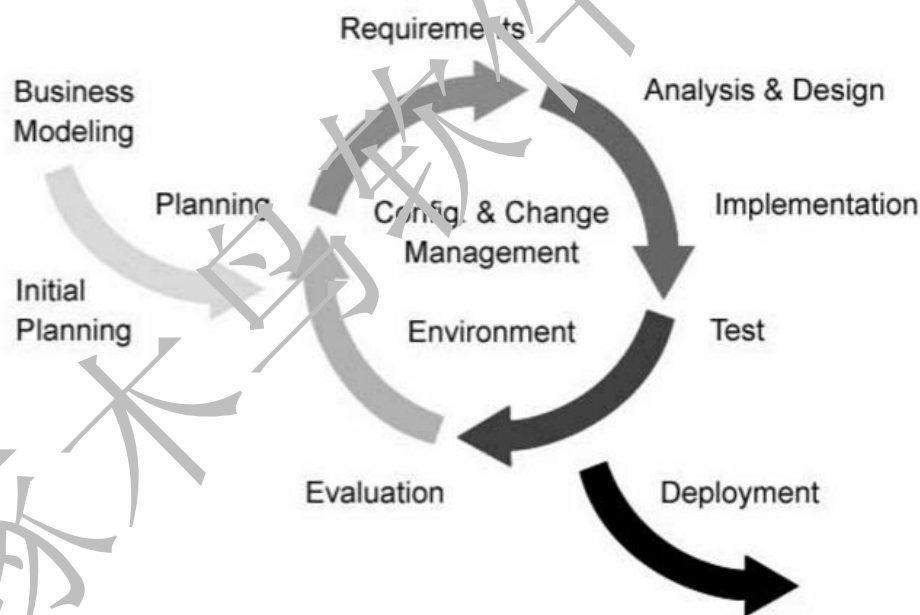


Agile & Scrum理论



迭代

迭代是指把一个复杂且开发周期很长的开发任务，分解为很多小周期可完成的任务，这样的一个月周期就是一次迭代的过程；同时每一次迭代都可以生产或开发出一个可以交付的软件产品



Agile & Scrum理论



Scrum

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；把一个开发流程的名字取名为Scrum，我想你一定能想象出你的开发团队在开发一个项目时，大家像打橄榄球一样迅速、富有战斗激情、人人你争我抢地完成它，你一定会感到非常兴奋的。而Scrum就是这样的—一个开发流程，运用该流程，你就能看到你团队高效的工作。



Agile & Scrum理论



3个角色

- Product Owner
- Scrum Master
- Scrum Team

3个工件

- Product Backlog
- Sprint Backlog
- 燃尽图 (Burn-down Chart)

5个活动

- Sprint计划会议 (Sprint Planning Meeting)
- 每日站会 (Daily Scrum Meeting)
- Sprint评审会议 (Sprint Review Meeting)
- Sprint回顾会议 (Sprint Retrospective Meeting)
- 产品Backlog梳理会议 (Product Backlog Refinement)

5个价值

- 承诺-愿意对目标做出承诺
- 专注-把你的心思和能力都用到你承诺的工作上去
- 开放-Scrum把项目中的一切开放给每个人看
- 尊重-每个人都有他独特的背景和经验
- 勇气-有勇气做出承诺，履行承诺，接受别人的尊重



Agile & Scrum理论



Scrum开发流程中的三大角色



- 产品负责人 (Product Owner)

主要负责确定产品的功能和达到要求的标准，指定软件的发布日期和交付的内容，同时有权力接受或拒绝开发团队的工作成果。



- 流程管理员 (Scrum Master)

主要负责整个Scrum流程在项目中的顺利实施和进行，以及清除挡在客户和开发工作之间的沟通障碍，使得客户可以直接驱动开发。



- 开发团队 (Scrum Team)

主要负责软件产品在Scrum规定流程下进行开发工作，人数控制在5~10人左右，每个成员可能负责不同的技术方面，但要求每成员必须要有很强的自我管理能力和一定的表达能力；成员可以采用任何工作方式，只要能达到Sprint的目标。

Agile & Scrum理论



3个工件



Product Backlog: 产品backlog由所有的功能特性，包括业务功能，非业务功能（技术、架构和工程实践相关），提升点以及缺陷的修复等组成。这些内容也是将来产品版本发布的主要内容。包括：估算、创建者、优先顺序、版本发布、负责人、优先级和可视化。



Sprint Backlog: 先为一个项目写下所有它该具备的显著特性和功能，数量不必很多，做好能保证团队的第一个Sprint有活可干。随着Sprint的进行，生产出可发布的产品增量，客户对产品的直观认识也会随之加深，他们可以据此建议更改或者添加产品Backlog中的任务。



燃尽图 (Burn-down Chart): 见后

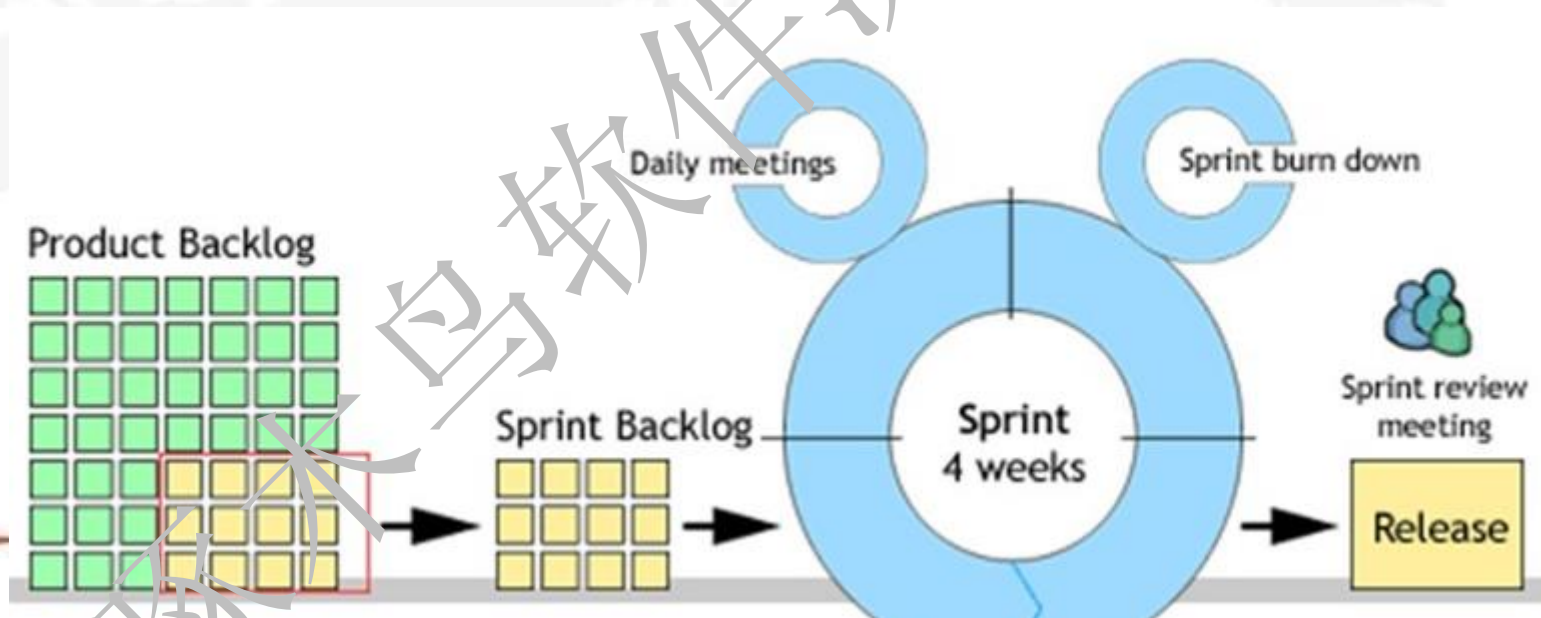


Agile & Scrum理论



Scrum 开发模型

SCRUM 开发模型

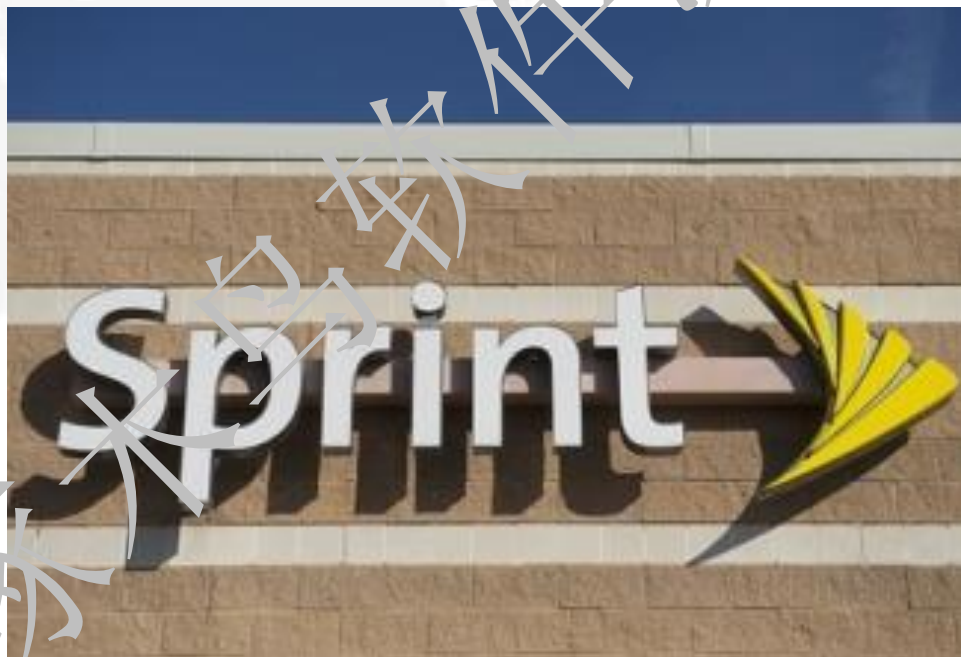


Agile & Scrum理论



Sprint

Sprint是短距离赛跑的意思，这里面指的是一次迭代，而一次迭代的周期是1周到1个月时间，也就是我们要把一次迭代的开发内容以最快的速度完成它，这个过程我们称它为Sprint。



Agile & Scrum理论



Scrum开发

1. 我们首先需要确定一个**Product Backlog**（按优先顺序排列的一个产品需求列表），这个是由Product Owner 负责的；
2. Scrum Team根据Product Backlog列表，**做工作量的预估和安排**；
3. 有了Product Backlog列表，我们需要通过**Sprint Planning Meeting**（Sprint计划会议）来从中挑选出一个Story作为本次迭代完成的目标，这个目标的时间周期是1~4个星期，然后把这个Story进行细化，形成一个Sprint Backlog；
4. **Sprint Backlog**是由Scrum Team去完成的，每个成员根据Sprint Backlog再细化成更小的任务（细到每个任务的工作量在2天内能完成）；
5. 在Scrum Team完成计划会议上选出的Sprint Backlog过程中，需要进行**Daily Scrum Meeting**（每日站立会议），每次会议控制在15分钟左右，每个人都必须发言，并且要向所有成员当面汇报你昨天完成了什么，并且向所有成员承诺你今天要完成什么，同时遇到不能解决的问题也可以提出，每个人回答完成后，要走到黑板前更新自己的Sprint burn down（Sprint燃尽图）；

Agile & Scrum理论



Scrum开发

6. 做到**每日集成**，也就是每天都要有一个可以成功编译，并且可以演示的版本；很多人可能还没有用过自动化的每日集成，其实TFS就有这个功能，它可以支持每次有成员进行签入操作的时候，在服务器上自动获取最新版本，然后在服务器中编译，如果通过则马上再执行单元测试代码，如果也全部通过，则将该版本发布，这时一次正式的签入操作才保存到TFS中，中间有任何失败，都会用邮件通知项目管理人员；
7. 当一个Story完成，也就是Sprint Backlog被完成，也就表示一次Sprint完成，这时，我们要**进行 Sprint Review Meeting（演示会议）**，也称为评审会议，产品负责人和客户都要参加（最好本公司老板也参加），每一个Scrum Team的成员都要向他们演示自己完成的软件产品（这个会议非常重要，一定不能取消）；
8. 最后就是**Sprint Retrospective Meeting（回顾会议）**，也称为总结会议，以轮流发言方式进行，每个人都要发言，总结并讨论改进的地方，放入下一轮Sprint的产品需求中。



Agile & Scrum理论



Product Backlog

产品需求

产品 BACKLOG (示例)					
ID	Name	Imp	Est	How to demo	Notes
1	存款	30	5	登录, 打开存款界面, 存入 10 欧元, 转到我的账户余额界面, 检查我的余额增加了 10 欧元。	需要 UML 顺序图。目前不需要考虑加密的问题。
2	查看自己的交易明细	10	8	登录, 点击“交易”, 存入一笔款项。返回交易页面, 看到新的存款显示在页面上。	使用分页技术避免大规模的数据库查询。和查看用户列表的设计相似。

Imp: 重要性; Est: 大致相当于一个“理想的人天 (man-day)”



Agile & Scrum理论



每日的站立会议

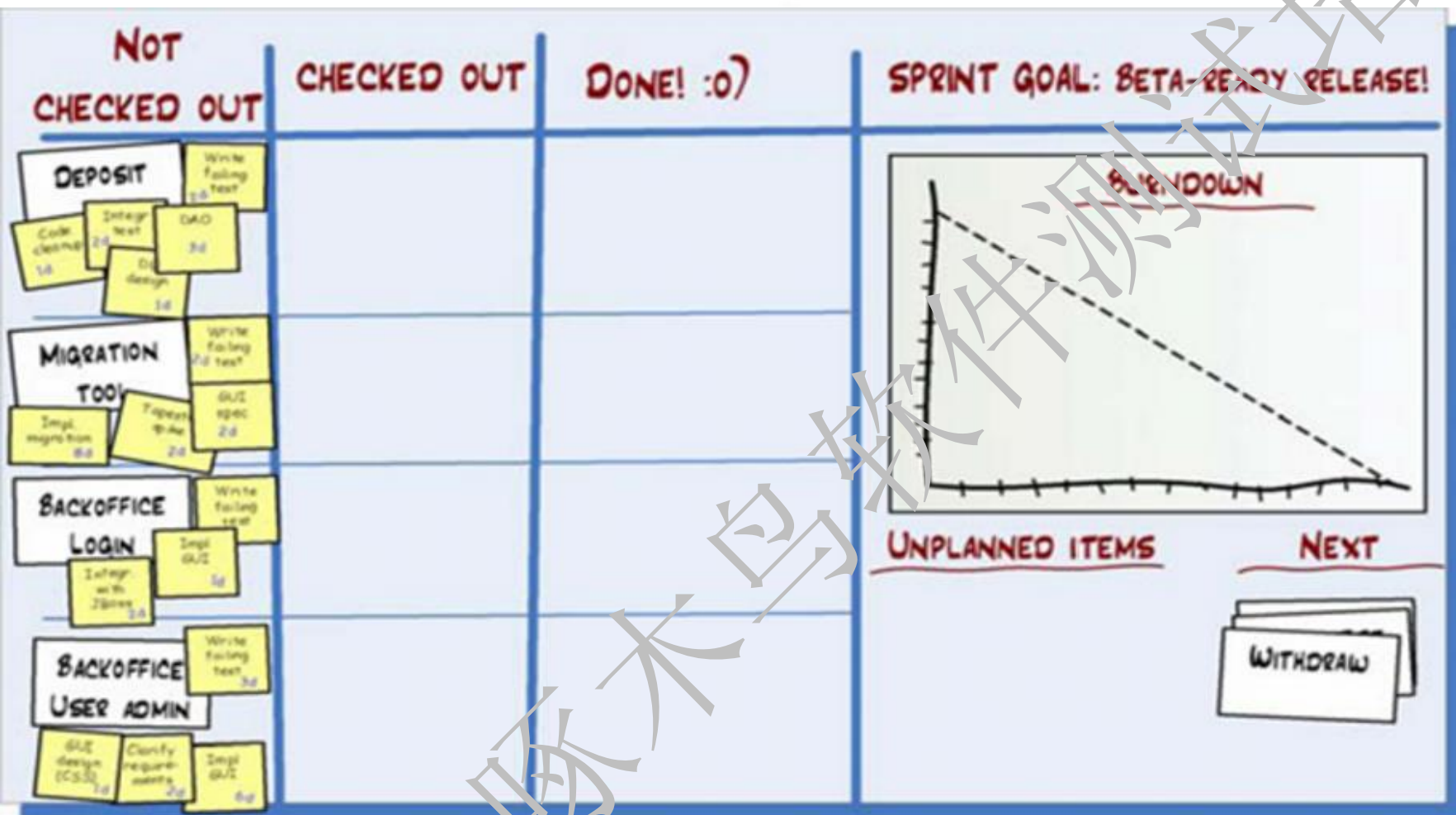
昨天做了什么？
今天准备做什么？
工作中有什么问题？



Agile & Scrum理论



任务看板

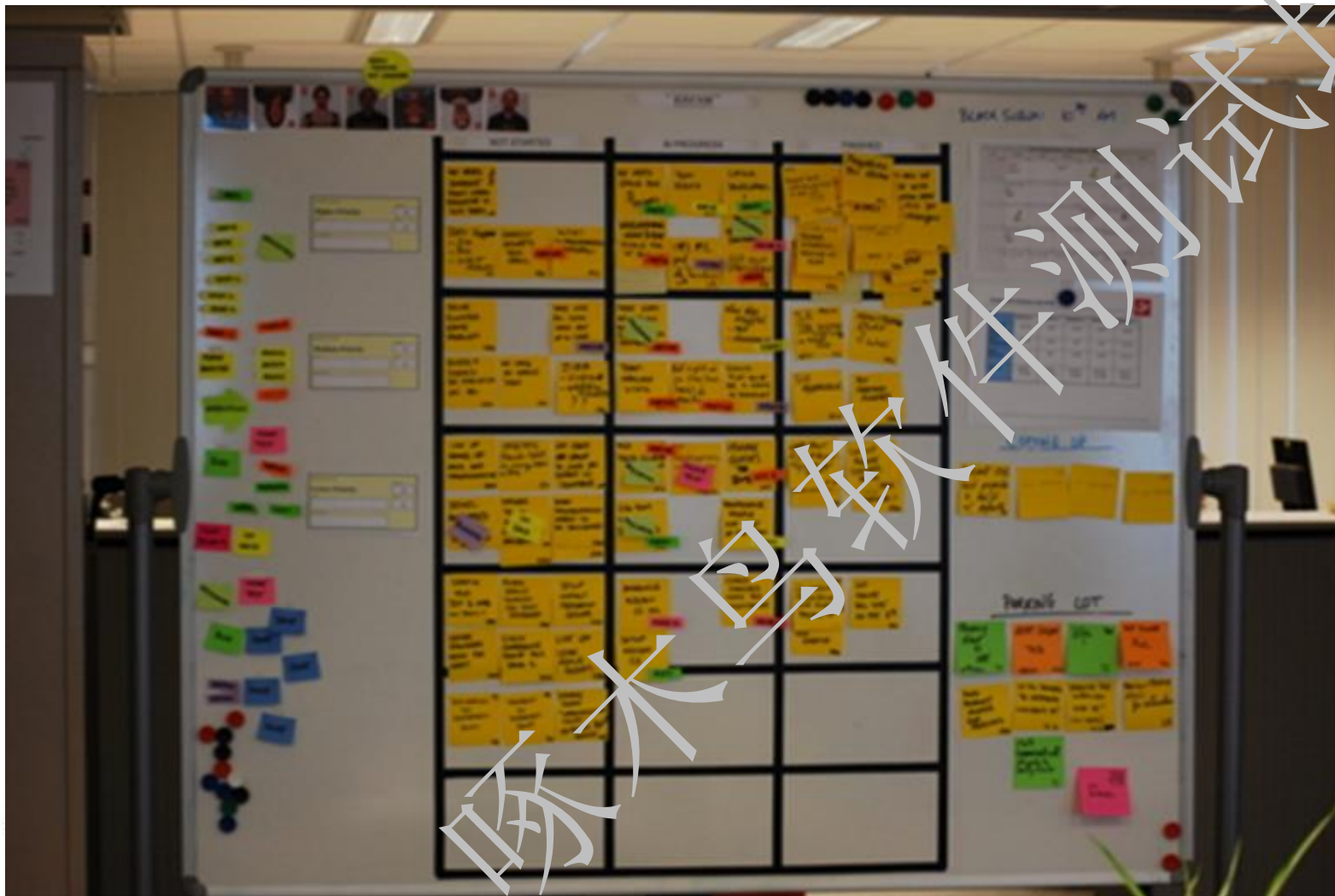


任务看板包含 未完成、正在做、已完成的工作状态，假设你今天把一个未完成的工作已经完成，那么你要把小卡片从未完成区域贴到已完成区域。

Agile & Scrum理论



任务看板

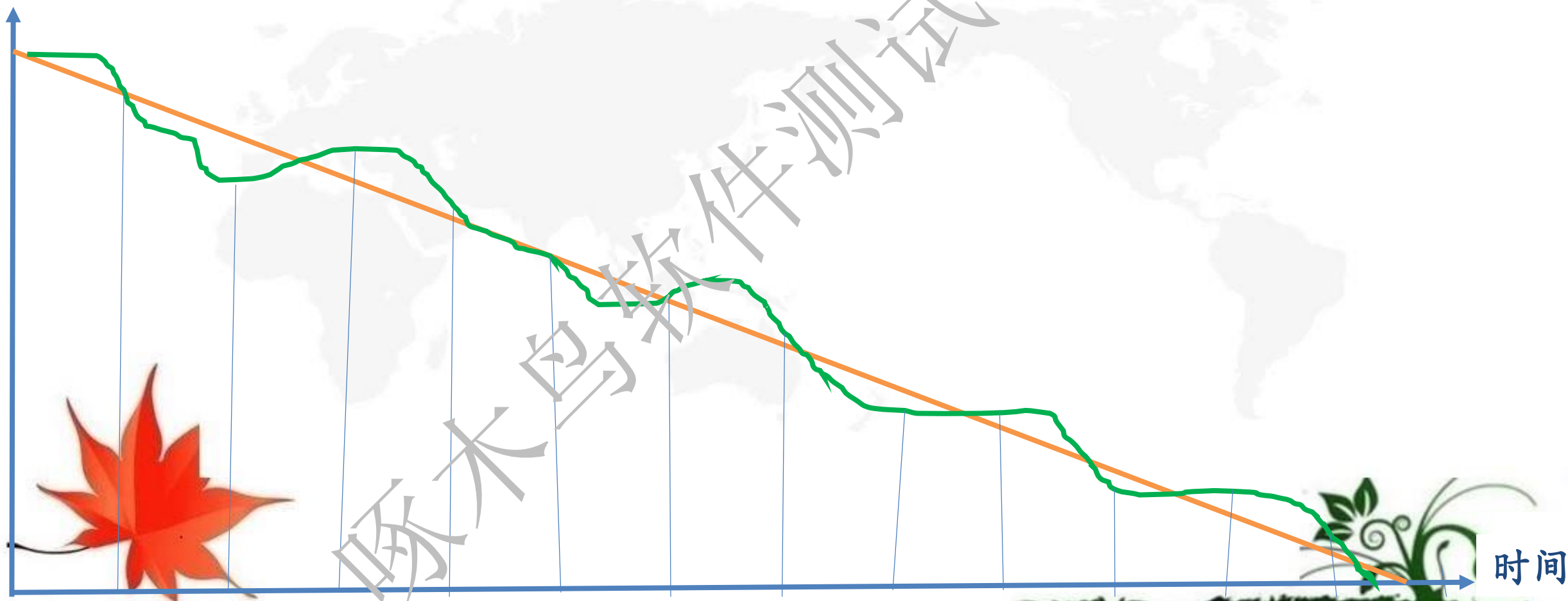


Agile & Scrum理论



燃尽图

计划工作量 (mh)



Agile & Scrum理论



计划纸牌



比如A程序员开发一个功能，需要5个小时，B程序员认为只需要半小时，那他们各自取相应的牌，藏在手中，最后摊牌，如果时间差距很大，那么A和B就可以讨论A为什么要5个小时...

Agile & Scrum理论



自组织与非自组织



Agile & Scrum理论

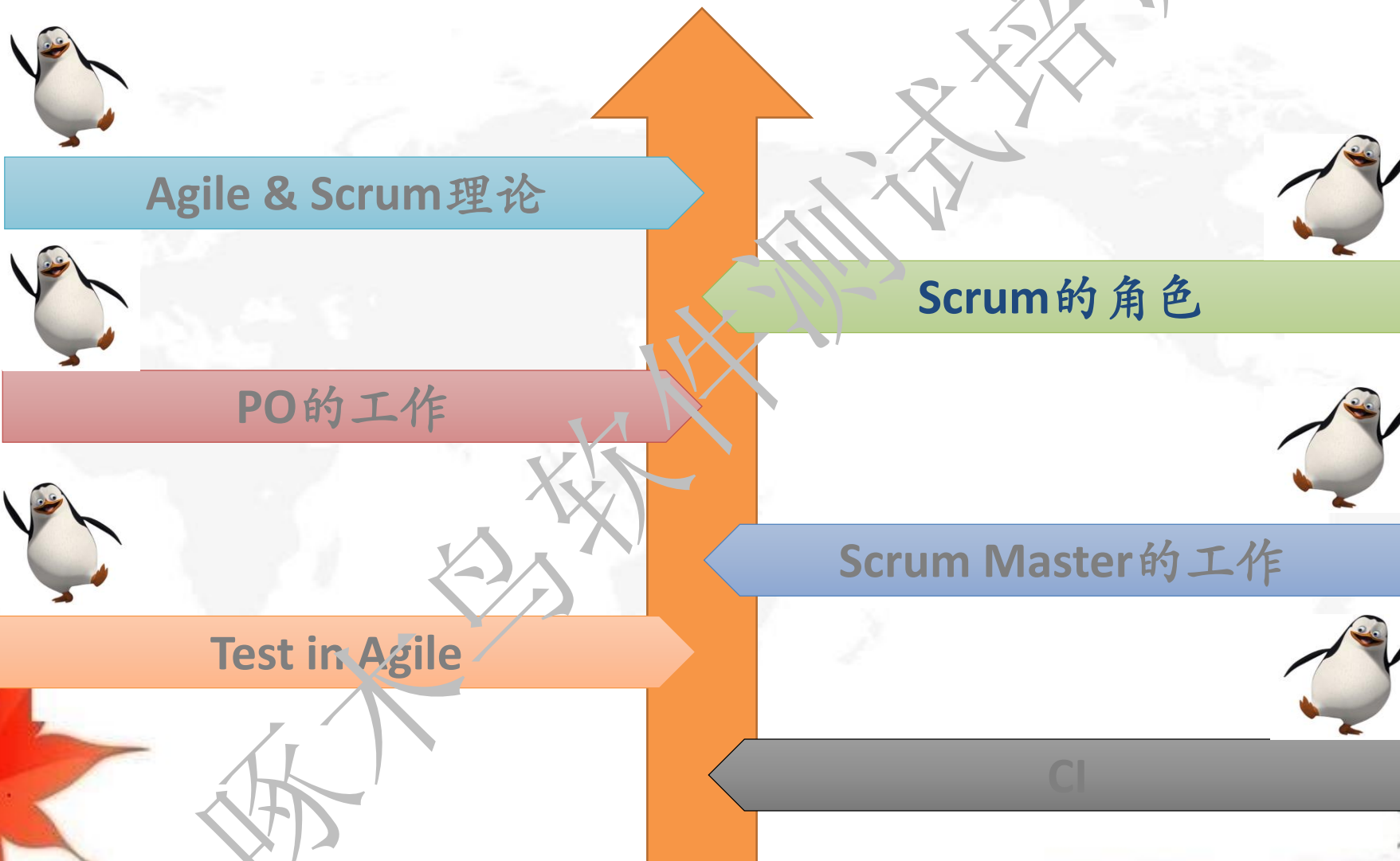


敏捷开发的4句宣言

- 个体与交互 胜过 过程与工具
- 可以工作的软件 胜过 面面俱到的文档
- 客户协作 胜过 合同谈判
- 响应变化 胜过 遵循计划



目录



Scrum的角色



Product Owner (PO)

Product Owner角色定义

确定产品的方向和愿景，定义产品发布的内容、优先级及交付时间，为产品ROI

(profitability of product) 负责。是维护产品需求清单 (product backlog) 的人，代表利益相关者的利益。



Scrum的角色



Product Owner (PO)

Product Owner工作职责

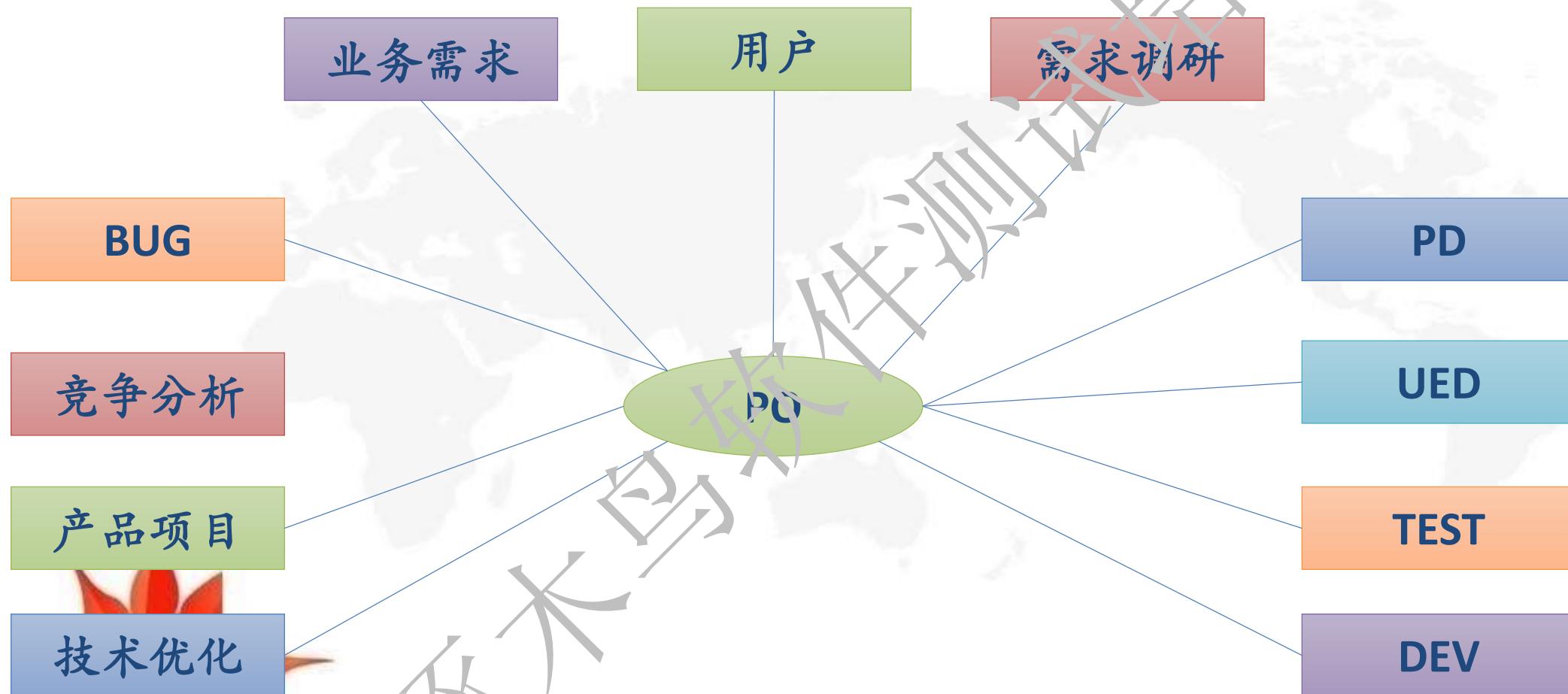
负责最大化产品以及开发团队工作的价值。主要职责如下：

1. 确定产品的功能；
2. 决定发布的日期和发布内容；
3. 为产品的ROI负责；
4. 根据市场价值确定功能优先级；
5. 每个sprint中，根据需要调整功能和优先级（每个sprint开始前调整）；
6. 接受或拒绝开发团队的工作成果；
7. 参与Scrum Planning Meetings（Sprint计划会议），Sprint Review Meeting（Sprint评审会）和Sprint Retrospective Meeting（Sprint回顾会议）

Scrum的角色



Product Owner (PO)



Scrum的角色



Scrum Master (SM)

Scrum Master角色定义

是团队的导师和组织者，与Product Owner紧密合作，及时为团队成员提供帮助。促使team按照scrum方式运行，为Scrum过程负责的人。

Scrum Master并非团队的领导（因为团队是自我组织的），而是一个负责屏蔽外界对开发团队干扰的角色。Scrum Master是规则的执行者，他是Scrum团队中的服务型领导。



Scrum的角色



Scrum Master (SM)

Scrum Master工作职责

确保scrum被理解和正确使用并使得Scrum的收益最大化。主要职责如下：

1. 保证团队资源合理利用；
2. 保证各个角色及职责良好协作；
3. 解决团队开发中的障碍；
4. 作为团队和团队外部的接口，协调解决沟通中的问题；
5. 保证开发过程按计划进行，组织Scrum Planning Meetings（Sprint计划会议），Daily Stand-up Meeting（每日站会），Sprint Review Meeting（Sprint评审会）和 Sprint Retrospective Meeting（Sprint回顾会）。

Scrum的角色



Scrum Master (SM)



Definition of Done



Scrum的角色



Team member

Testing

- Test case
- Test script
- Test execute
- Test report

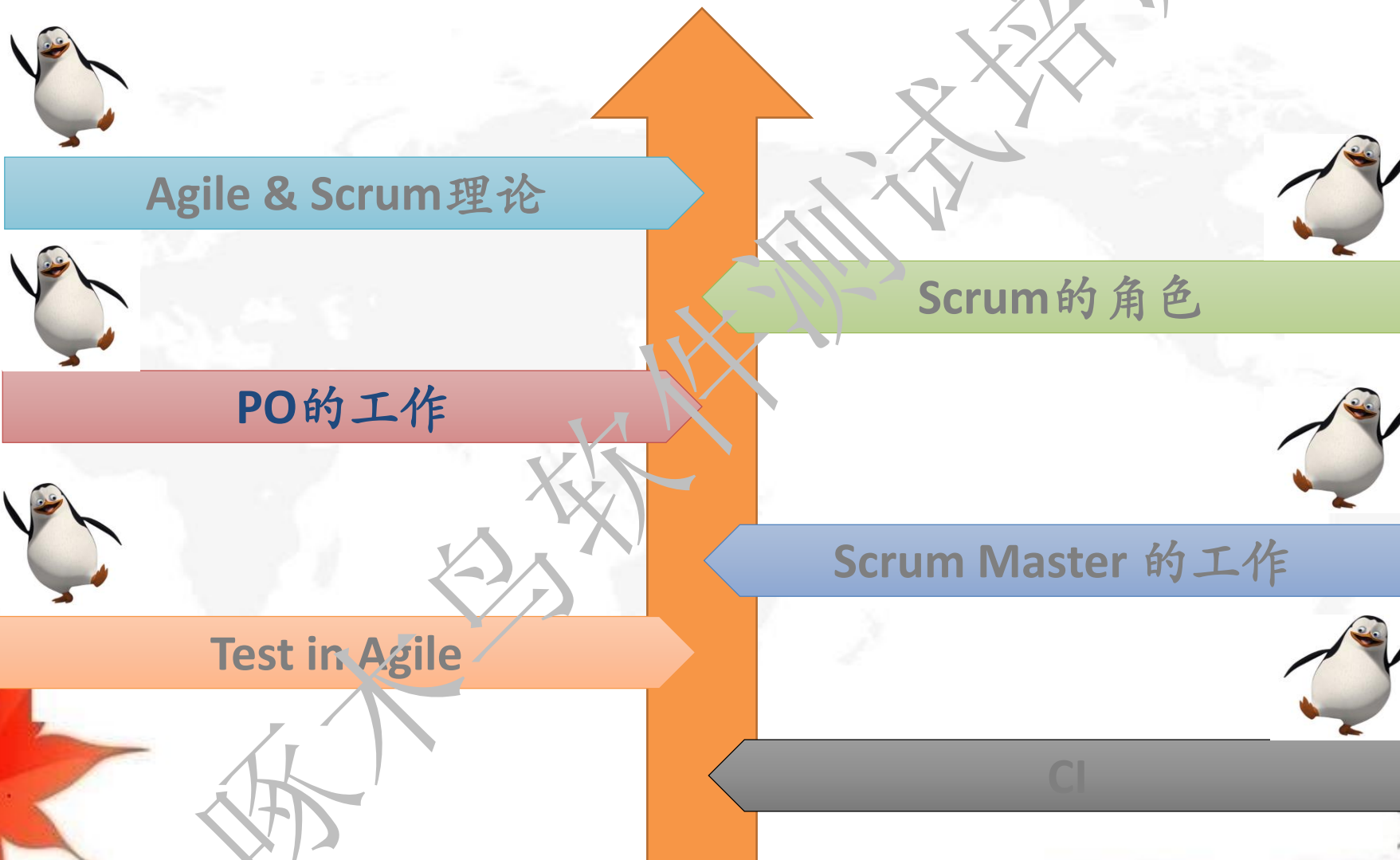
BUG

Coding

- Design
- Product coding
- Unit test



目录



PO的工作



Product Owner 职责

1. 确定产品的功能。
2. 决定发布计划（发布日期和发布内容）。
3. 为产品的profitability of the product (ROI)负责。
4. 根据市场价值确定功能优先级。（即根据功能客户的价值/投资回报率来确定优先级）
5. 在Sprint调整功能和调整功能优先级。
6. 接受或拒绝接受开发团队的工作成果。

PO的工作



Product Owner主要工作

1. 搜集用户故事
2. 记录用户故事
3. 需求讨论
4. 参与估算和Sprint计划会议
5. 维护产品列表
6. 验收产品，参与Sprint评审会议



PO的工作



案例

1超级用户

1.1 给卖家开设账户、删除账户

1.2 添加、删除商品类别 **注意没有修改**

2卖家

2.1 登录，修改登录密码，退出

2.2 添加,修改，删除商品

3买家：

3.1 注册，登录，修改密码，修改注册信息，退出

3.2 查询商品，商品预览

3.3 查看商品详细信息

3.4 加入购物车，查看购物车，修改购物车中产品数量，将商品移除购物车

3.5 产生订单，确认订单，评价与评分

PO的工作



User Story

1 超级用户

1.1 登录 (US1) 、给卖家开设账户(US2)、删除卖家账户(US3)

1.2 添加商品类别 (US4)、删除商品类别 (US5) 注意没有修改

2 卖家

2.1 登录(US6)、修改登录密码(US7)、退出(US8)、冻结营业(US9)

2.2 添加商品(US10)、修改商品(US11)、删除商品(US12)

3 买家:

3.1 注册(US13)、登录(US14)、修改密码(US15)、修改注册信息(US16)、退出(US17)

3.2 查询商品(US18)、商品预览(US19)

3.3 查看商品详细信息(US20)

3.4 加入购物车(US21)、查看购物车(US22)、修改购物车中产品数量(US23)、将商品移除购物车(US24)

3.5 产生订单(US25)、确认订单(US26)、评价与评分(US27)



PO的工作



User Story细化

US1: 超级用户登录系统;

US2: 超级用户登录后给卖家设置登录名、密码;

US3: 超级用户登录后可以删除卖家账号 (卖家必须没有订单);

US4: 超级用户登录后设置商品类别, 包括服装、食品、家电等;

US5: 超级用户登录后删除设置商品类别 (商品类别下必须没有商品);

US6: 卖家根据超级用户设置的密码登录系统 (不考虑验证码);

US7: 卖家登录后可以修改密码, 修改密码字段包括原密码、当前密码、当前密码确认;

US8: 卖家登录后退出系统;

US9: 卖家登录后设置, 冻结营业。对于冻结营业卖家, 只能处理已生成的订单, 不可以产生新订单;



PO的工作



- US10: 卖家登录系统后添加商品，以下字段为必填项：商品名称、商品类别、简介、详情、价格、图片(1-4张)；
- US11: 卖家登录系统后添加商品后，可以修改商品信息；
- US12: 卖家登录系统后添加商品后，可以删除商品信息(删除后的信息不影响已下单的客户)；
- US13: 买家注册信息，以下字段为必填项：姓名、账号、密码、密码确认、收货地址(1-n个)、微信账号；
- US14: 买家登录系统；
- US15: 买家登录后可以修改密码，修改密码字段包括原密码、当前密码、当前密码确认；
- US16: 买家登录后可以修改不包括密码的用户信息；
- US17: 买家登录后退出系统；
- US18: 买家登录后，可以通过关键字和商品类别组合查询商品；

PO的工作



- US19：买家登录后展示最新添加的20样商品预览，查询后显示查询的商品预览；
- US20：买家通过商品预览，点击商品后，显示商品的所有信息，包括商品信息和所有对该商品的评分评价；
- US21：买家通过商品详情加入购物车；
- US22：买家进入购物车后可以查看购物车内容，进行数量编辑；
- US23：买家进入购物车后可以修改已放入购物车内产品的数量；
- US24：买家进入购物车后可以移除已放入购物车内的产品；
- US25：买家可以通过商品详情或者购物车生成订单；
- US26：买家可以选择配送地址，确认后生成订单；
- US27：买家可以查询自己的订单，收到商品后确认，并且给出评分(包括商品相符程度、物流速度、商品质量)评价。

PO的工作



制作Product Backlog

US1: 超级用户登录系统

描述: 超级用户是系统最高权限的用户，他主要负责给卖家建立登录账号以及建立产品目录。其用户名为root，密码存在XML文件里面。超级用户登录界面只需要给出密码文本框即可。UI要求为L级别。

创建者: Jerry

优先顺序: 1

版本发布: Sprint2

负责人: (Stand up meeting决定)

优先级: H

估算: (Plan Meeting决定)

PO的工作



制作Product Backlog

US2:

...

US3:

...

US4:

...

US5:

...

US6:

...

US7:

...

US8:

...

US9:

...

US10

: ...

US11

: ...

US12

: ...

US13

: ...



PO的工作



制作Product Backlog

US13: 买家注册信息

描述: 卖家注册登录系统信息, 以下字段为必填项: 姓名、账号、密码、密码确认、收货地址(1-n个)、微信账号。另外还有QQ号、生日、Email为非必填项。当用户信息填写错误的时候, 页面用AJAX技术显示错误信息。收货地址(1-n个)可以在另外一个页面中填写。

创建者: Jerry

优先顺序: 2

版本发布: Sprint3

负责人: (Stand up meeting决定)

优先级: M

估算: (Plan Meeting决定)

PO的工作



制作Product Backlog

US14

: ...

US15

: ...

US16

: ...

US17

: ...

US18

: ...

US19

: ...

US20

: ...

US21

: ...

US22

: ...

US23

: ...

US24

: ...

US25

: ...

US26

: ...

US27

: ...



PO的工作



进度

- Spring1: 系统架构设计、数据库结构设计、业务学习、HTML5 & CSS3学习
- Spring2: US1-US12, 开发、单元测试、功能测试、系统测试
- Spring3: US13-US20, 开发、单元测试、功能测试、系统测试
- Spring4: US21-US27, 开发、单元测试、功能测试、系统测试, 总体测试, 总体总结

2 个Function team: Galaxy Team: WEB DEV, Shark Team: 安卓APP开发

1个 Integration Team: 负责CI, 版本控制, 系统集成, 非功能性测试



PO的工作



Function Team Sprint Backlog (SPRINT3)

US13 DEV & UT	US13 Test Case	US13 Test Execute	US14 DEV & UT	US14 Test Case	US14 Test Execute
...	US20 DEV & UT	US20 Test Case	US20 Test Execute	CI 问题定 位	Sprint2 回 归测试
复测	Bug修复	Sprint3、 Sprint2 探 索式功能 测试			



PO的工作



Integration Team Sprint Backlog (SPRINT3)

Sprint2版本
Preferment
Test环境搭
建

Sprint2版本
Preferment
Test数据准
备

Sprint2版
本负债测
试

Sprint2版
本负载测
试

Sprint2版
本强度

Sprint2版
本大数据
容量测试

Sprint2版
本用户友
好性测试

Sprint2版
本安全测
试

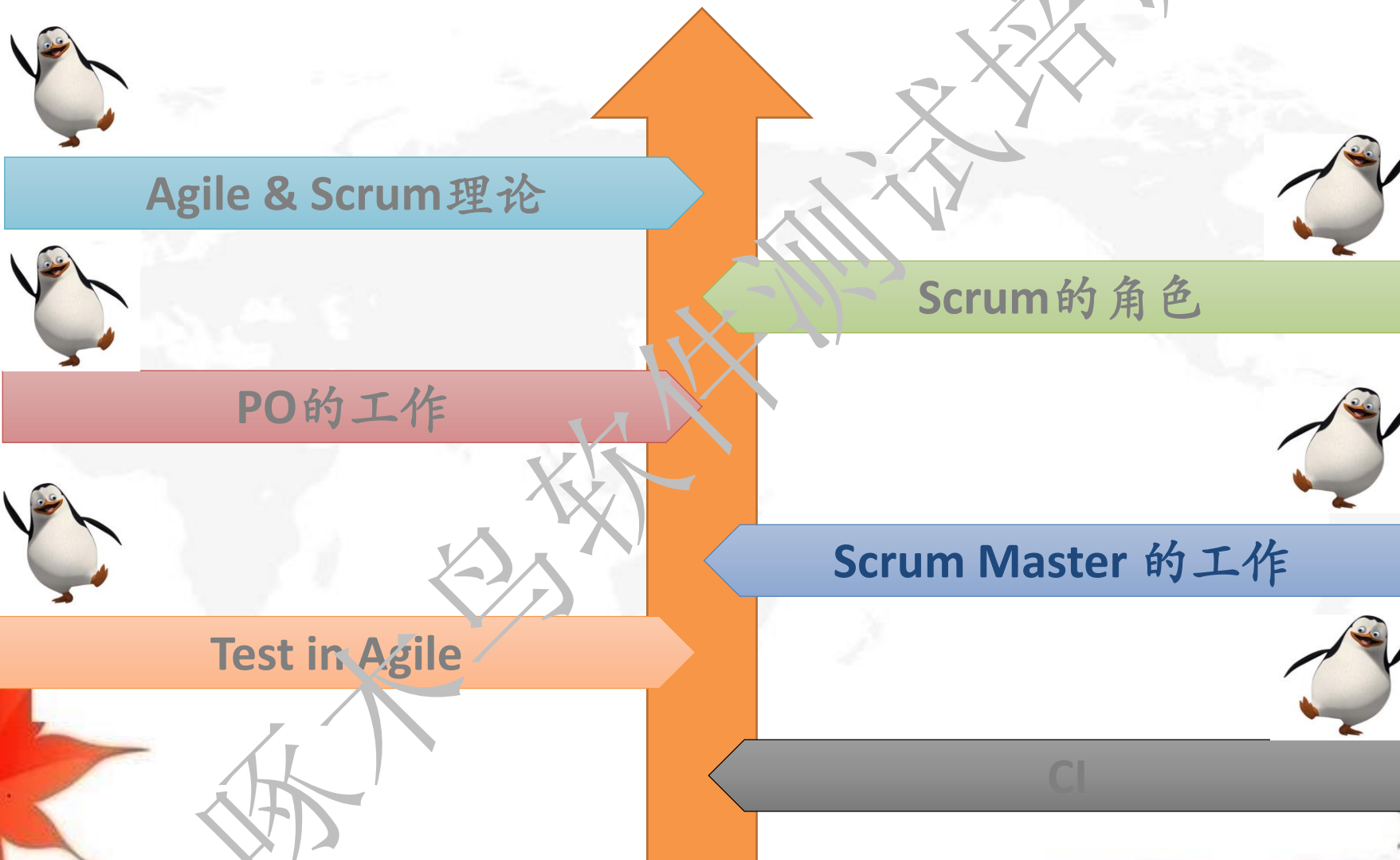
Sprint3 版
本控制

Sprint3 CI
控制

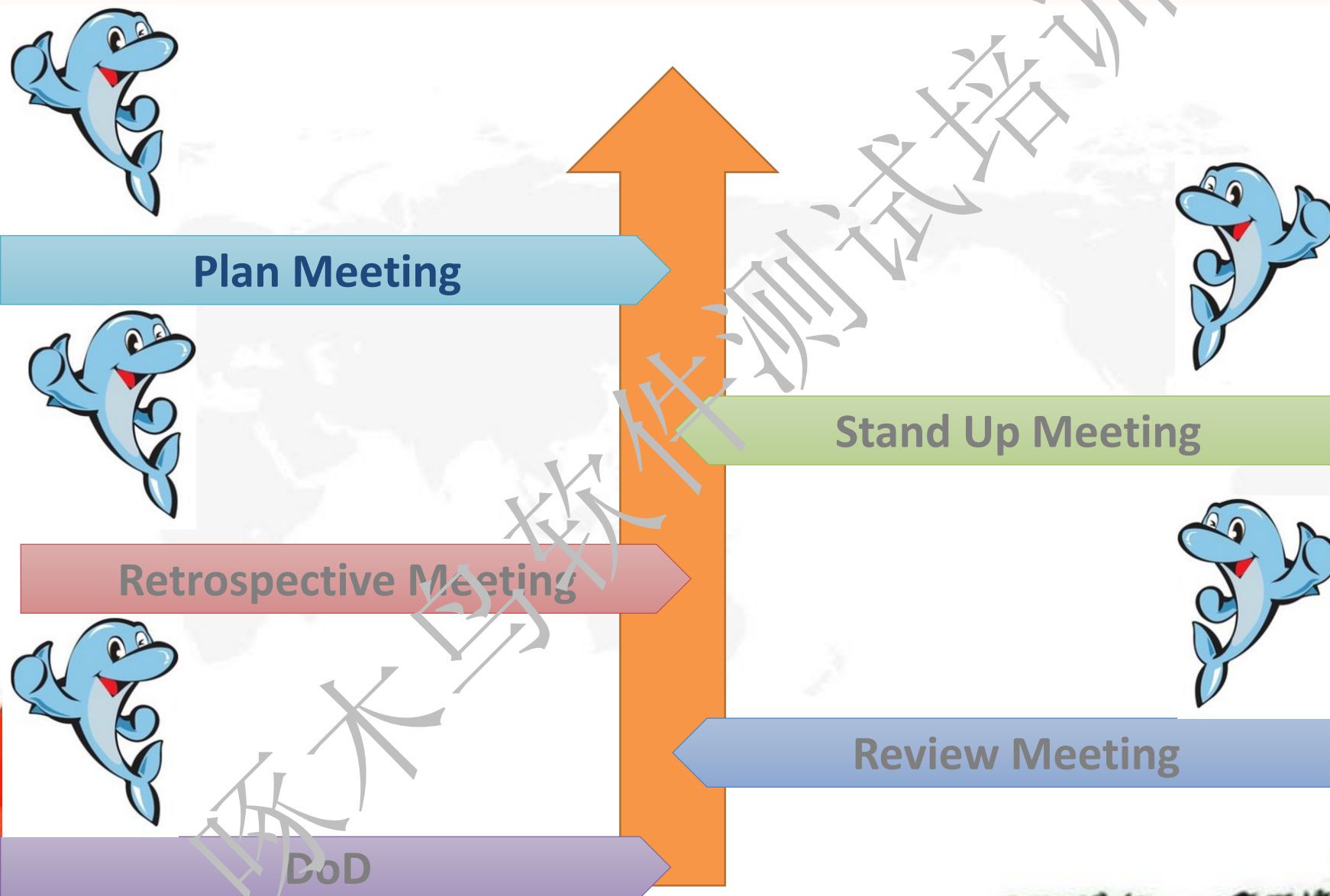
...



目录



Scrum Master的工作



Plan Meeting



会议内容

1. PO会拿出拆分好的Story，并且按照优先级排列。
2. PO会开始对优先级最高的Story进行讲解，包括接收条件等。
3. Scrum团队的所有人在Scrum Master的号令下开始举牌。
4. 由Scrum Master将达成一致Point记录在Story边上。
5. 重复上述步骤至BA准备的所有的Story都评完了。
6. 由Scrum Master根据经验将需要纳入本Sprint的Story贴到白板上，或者登记在管理软件中。

Plan Meeting



斐波那契数列

斐波那契数列指的是这样一个数列

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,

233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368。

这个数列从第2项开始，每一项都等于前两项之和。

在SCRUM出牌采用：1, 2, 3, 5, 8, 13, 21...（5表示5ManHours做完，一般不允许超过21）。



Plan Meeting



案例分析

Spring3(第三周): US13-US20

US13: 买家注册信息, 以下字段为必填项: 姓名、账号、密码、密码确认、收货地址(1-n个)、微信账号; 测试用例执行

PO讲解

出牌

决定

A: 8

B: 8

C: 13

让B来说位什么用8
可以完成US13开发,
争取B说服A、C, 或者A、C说服B

Plan Meeting



案例分析

Spring3(第三周): US13-US20

US13: 买家注册信息, 以下字段为必填项: 姓名、账号、密码、密码确认、收货地址(1-n个)、微信账号; 测试用例设计

PO讲解

出牌

决定

A: 5

B: 3

C: 8

每个人诉说自己意见, 争取最后达成。



Plan Meeting



案例分析

Spring3(第三周): US13-US20

US13: 买家注册信息, 以下字段为必填项: 姓名、账号、密码、密码确认、收货地址(1-n个)、微信账号;
测试用例执行

PO讲解

出牌

决定

A: 1

B: 1

C: 1

意见达成



Plan Meeting



案例分析

Spring3(第三周): US13-US20

	To Do	Doing	Done
US13: 买家注册信息, 以下字段为必填项: 姓名、账号、密码、密码确认、收货地址(1-n个)、微信账号; 开发与UT 8MH	US13: 买家注册信息, 以下字段为必填项: 姓名、账号、密码、密码确认、收货地址(1-n个)、微信账号; 测试用例设计 5MH	US13: 买家注册信息, 以下字段为必填项: 姓名、账号、密码、密码确认、收货地址(1-n个)、微信账号; 测试用例执行 1MH	

接下来继续US14-US20



Plan Meeting



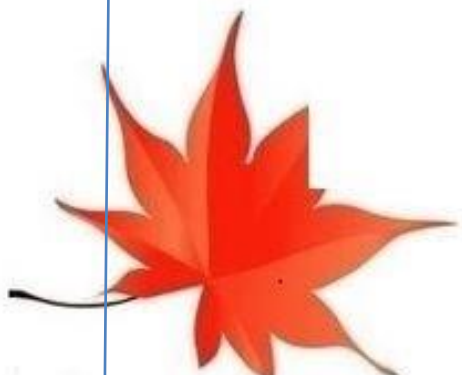
案例分析

Spring3(第三周): US13-US20

最后定义出来的开发时间为443 MH,比如一周工作五天,一天8个小时,团队人员5.5人(PO承担0.5个研发任务),有一位员工计划这个Sprint请2天假(16MH)。可以估算时间 $(443-16)\text{MH}/8\text{H}/5.5\text{M}=9.9$ 天,约不到2周。

427

days



Scrum Master的工作



Stand Up Meeting



内容

- 昨天我完成了什么？（把Task从doing move to done，或者减少MH）
- 今天我要去做什么？（把Task 从To Do move to doing）
- 在开发过程中遇到了什么阻力？

散会后SM：

- 绘制燃尽图
- 总结问题，组间协调或者找高级经理协调

Plan Meeting



案例分析

To Do

USX: .	USX: .	USX: .	USX: .	USX: .	USX: .	...
XMH	XMH	XMH	XMH	XMH	XMH	...
USX: ...	USX: ...	USX: ...	USX: ...	USX: ...	USX: ...	...
XMH	XMH	XMH	XMH	XMH	XMH	...
USX: USX:	USX: USX:	USX: USX:	USX: USX:	USX: USX:	USX: USX:	...
XMH XMH	XMH XMH	XMH XMH	XMH XMH	XMH XMH	XMH XMH	...

Doing

US14: 买家
登录系统
开发与UT
A 13MH

US14: 买家
登录系统;
测试用例设计
B 5MH

Done

US13: 买家...
测试用例设计
1MH

US13: 买家...
测试用例设计
5MH

US13: 买家...
测试用例设计
8MH



A stylized illustration of a person with dark hair and a white collared shirt under a maroon sweater. The right side of their face is covered by a white bandage with visible stitches. The background is a solid light blue.

- 

- 

- 昨天研究了JunitXMH就可以协助A
- 今天准备开发买家登录后可以修改密码...确认
- 没有遇到什么问题

开发 Do

Doing

Done

US14: 买家
登录系统
开发与UT

А 15 W H H

US14: 买家 登录系统; 测试用例设 计

B \$1M

US13: 买家... 测试用例执行

1MH

US15: 买家登 录后可以修改 密码...确认 测试用例设计

C 13MH

US15: 买家 登录后可以 修改密码...确 认 开发与UT

C 13MH

US13: 买家... 测试用例 计

5MH

US13: 买家注册...
开发与UT

8MH

Plan Meeting



案例分析

427

上一个工作日消耗：13个MH,进度滞后，从燃尽图上也可以看出。

今天CI发现10个没有通过的，有3个分配给我们小组。

SM今日工作：

- 协商A尽快掌握JUnit技巧，C协助；
- 看看别的小组的Bug处理流程，是否使用工具或者Excel表？
- 分别让A、B、C三人处理3个没有处理的case（在Plan是要预留一些时间给处理CI没有处理的case）；

days

Scrum Master的工作



Retrospective Meeting



介绍

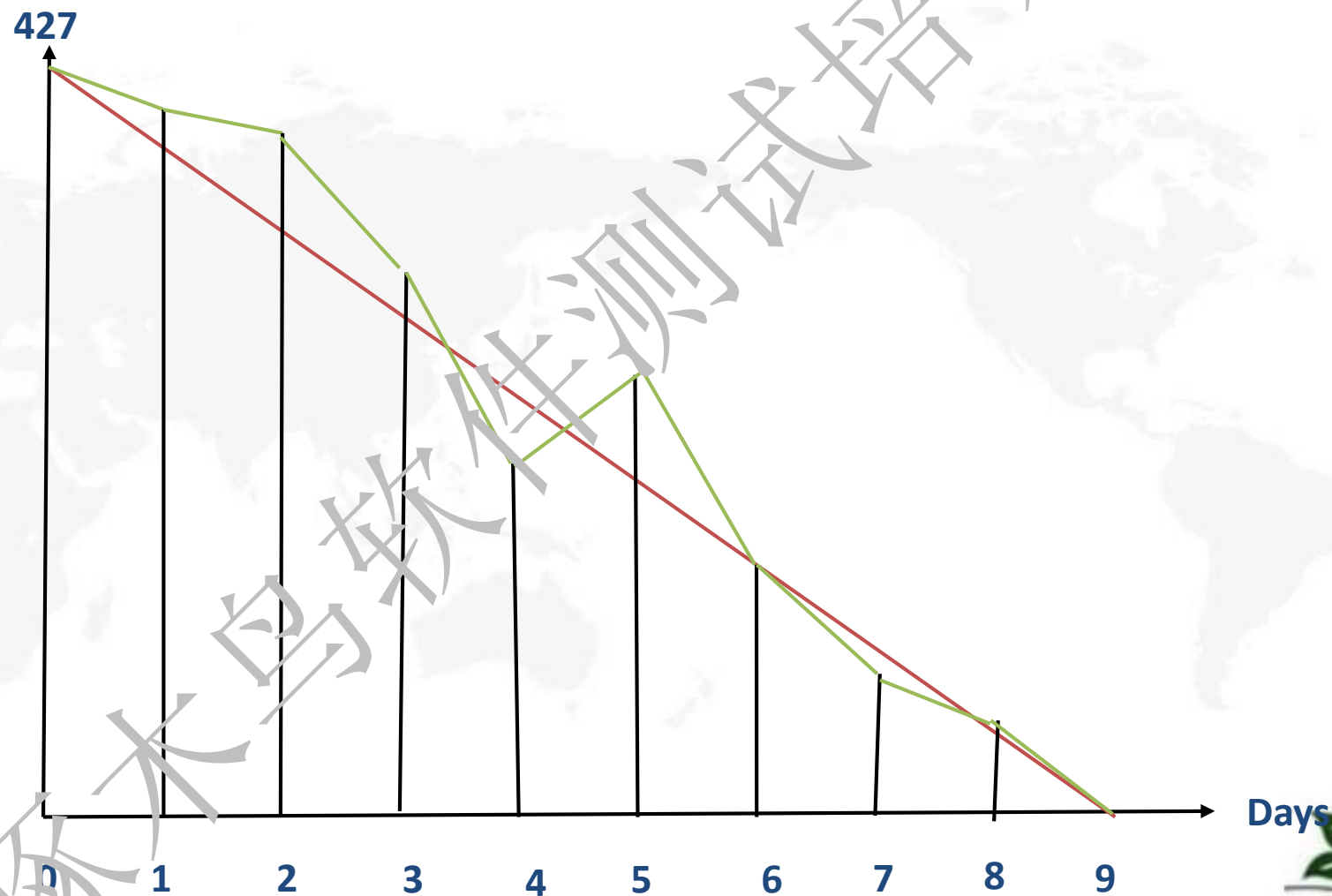
所谓Retrospective，就是指回顾，反省。Scrum流程是步步迭代的，而其迭代的不仅仅是你们开发的产品，也是你们这个team的工作方式。因此每个sprint之后回顾一下走过的这一段，包括从Plan， Design & Coding， Testing， 到Demo等等的方方面面。

1. 我们上个sprint哪里做的比较好
2. 我们上个sprint哪里做的不好
3. 列出来之后，然后讨论可能的解决方案。并最好分为三类：
 - Start doing
 - Stop doing
 - Continue doing

Retrospective Meeting



案例分析



Retrospective Meeting



案例分析



这个Sprint我认为整体把握的比较好，主要的问题是团队第一次接触Junit，建议下次如果有什么新技术最好Sprint前安排1到2天的学习时间。另外对于代码不要采取强制的Check Out,一个代码允许多人修改，Merge的时候仔细些就可以。



整个测试基本上还算可以，就是希望开发在把代码放入代码版本库里进行更加仔细的对以前的单元测试用例进行测试，这个Spring定位的几个CI没有通过的80%是由于这次新代码引入的。



A说的是对的，我用了一天的时间掌握了Junit工具。另外我认为测试在发现问题的同时可否将缺陷描述的清晰些，最好配上图片或者对应的log日志，便于我们修改问题。

Retrospective Meeting



案例分析

Galaxy Scrum Team 软件开发宣言

1. 新代码Check in到代码库中必须对以前所有的模块进行好单元测试;
2. 测试通过Excel提交缺陷, 缺陷务必详细, 使得研发容易定位, 必要的时候配上图片及Log日志
3. 有新技术(研发、业务)引入, Spring前预留相应的Warm Up时间
4. 不允许强制性的Check Out



Retrospective Meeting



QA Retrospective Meeting

Q: 领导要参加，怎么办？

A: 我们欢迎领导（特别是负责研发的副总）参加，希望他可以通过这个机会了解我们下一步会如何改进，为什么要改进，同时也更有机会获得管理层的支持

Q: 会议达成的共识要不要以书面形式写下来？

A: 要，同时要在下一个sprint中不断强化改进的结论，回顾会议的结论，SM会发minutes给Team、PO和负责研发的副总；最好选出团队列出的最好的3点和最需要改进的3点，写在一张大白纸上，贴在团队的工作板旁边。

Q: 以MSN形式开回顾会议，有没有人实践过？

A: 最好有个会议室，MSN可能会花费很长时间，而且文字表达总是缺乏生动，Scrum的一大特点就是Face-face。在一个案例中，PO在美国，team在中国，在回顾会议中，我们用Webeye和电话会议的方式来保证双方共同参加

Retrospective Meeting



QA Retrospective Meeting

Q: 是不是强制要求每个人都要发言?

A: 要求每个人都发言。SM会事先将sprint过程中的主要事件节点和遇到的issue在白板中标出，同时也要求team成员列出做的最好的和需要改进的，具体：Retrospective会议的时候，发POSTIT给每个团队成员（包括Team Leader），给大家15-20分钟自己写下自己觉得过去迭代中做到好的事情，还有做的不好（或者待改进的事情），写完贴到白板上汇总，大家一条一条过，并且进行讨论。最后选出TOP 3 GOOD THINGS和TOP 3 NEED IMPROVEMENT。当然，做完了项目总有一些感慨的，随便说说也是非常欢迎的。

Q: 时间长度上面，真的需要3小时吗?

A: 1~2个小时有效，可以控制不跑题。也有一些情况，全体成员结束之后，PM和leader又会讨论更长的时间。

Q: Scrum Master在回顾会议时有没有特别不恰当的行为，应当避免的?

A: 最后一个发言，否则下属会被影响

Scrum Master的工作



Retrospective Meeting



QA Retrospective Meeting

Q: 是不是强制要求每个人都要发言?

A: 要求每个人都发言。SM会事先将sprint过程中的主要事件节点和遇到的issue在白板中标出，同时也要求team成员列出做的最好的和需要改进的，具体：Retrospective会议的时候，发POSTIT给每个团队成员（包括Team Leader），给大家15-20分钟自己写下自己觉得过去迭代中做到好的事情，还有做的不好（或者待改进的事情），写完贴到白板上汇总，大家一条一条过，并且进行讨论。最后选出TOP 3 GOOD THINGS和TOP 3 NEED IMPROVEMENT。当然，做完了项目总有一些感慨的，随便说说也是非常欢迎的。

Q: 时间长度上面，真的需要3小时吗?

A: 1~2个小时有效，可以控制不跑题。也有一些情况，全体成员结束之后，PM和leader又会讨论更长的时间。

Q: Scrum Master在回顾会议时有没有特别不恰当的行为，应当避免的?

A: 最后一个发言，否则下属会被影响

Scrum Master的工作



Review Meeting



常见的迭代DoD条款有：

在**迭代结束前**给Product Owner演示并接受评价的会议，并根据反馈结果，提出新的产品

Backlog

参与人员：产品经理、Product Owner、Scrum Master、团队所有成员

会议时长：1-4小时，视演示内容而定

主要是检验迭代成果，检查是否完成迭代计划中的迭代目标，有可能的话要求用户参与测试流程，并得到用户对产品的认可，鼓励用户自己进行测试设计和进行破坏性测试，充分暴露产品的设计和功能问题。

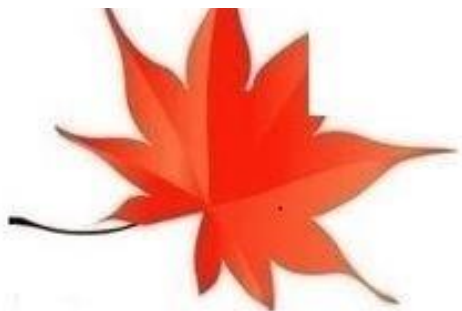
由Scrum Master来推进会议进程，Product Owner记录用户反馈，根据结果维护产品backlog，一般在迭代结束前做一次。

DoD



每日DoD，典型条款有：

- 搭建每日构建环境，晚上自动静态代码检查、编译、部署和测试，每日修复前一日构建和测试发现的缺陷和问题。
- 下班前必须Check in当天书写的代码。
- 当天的代码必须在当天或者第2天邀请同伴进行Code Review。
- 采用TDD，凡是检入的功能代码必须要有对应的单元测试（严格采用TDD）



DoD

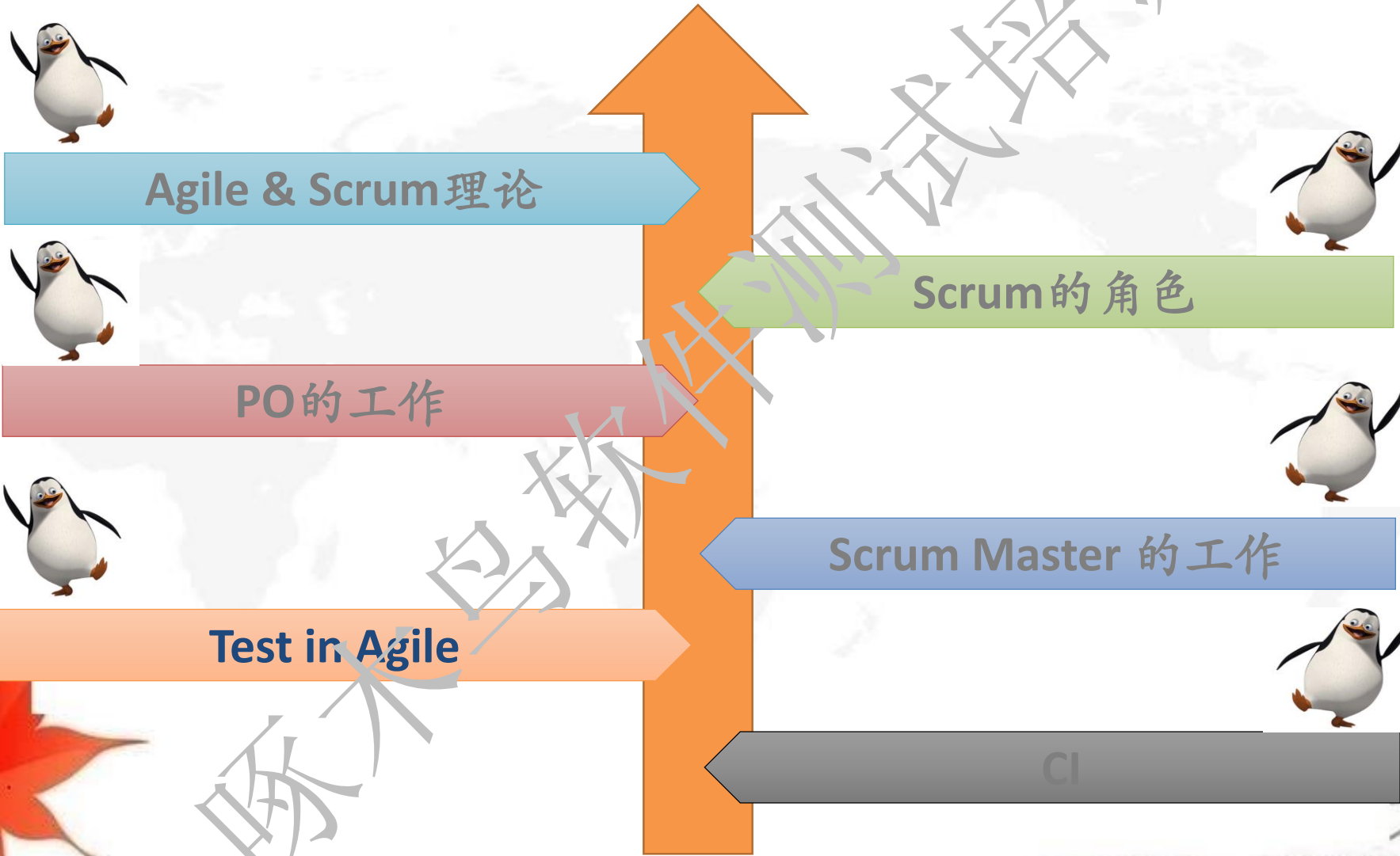


用户故事（或者用例）的DoD 典型条款有：

- 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
- 某些用户故事得到对应的自动化测试用例
- 用户故事得到用户代表试用并初步认可



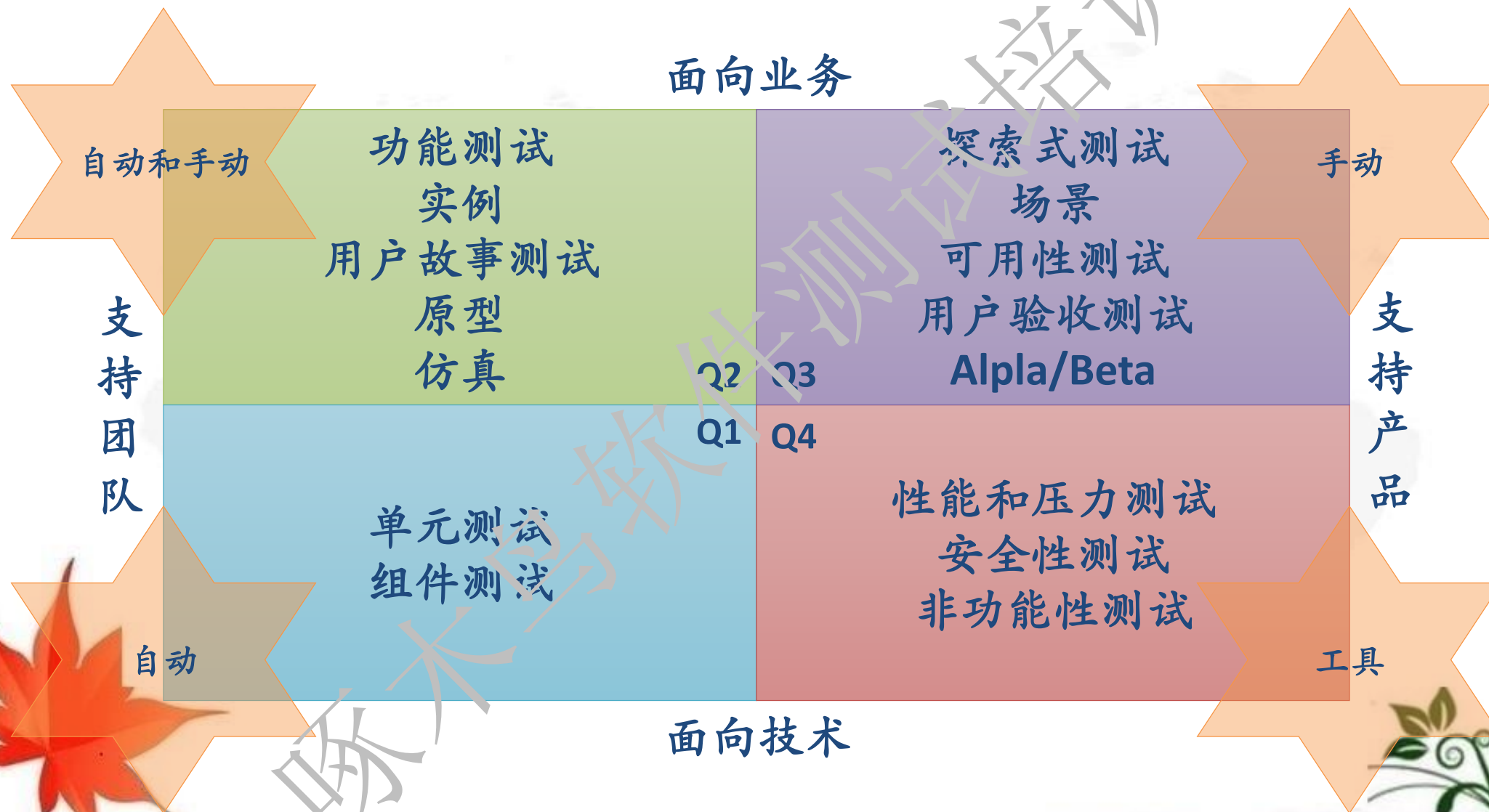
目录



Test in Agile



测试四项象限



Test in Agile



支持团队的面向技术测试



第一象限测试的意义

- 尽可能做好的软件
- 编写过程中漏掉的缺陷越多，交付的效率越慢

无单元测试



有单元测试



自动检查修改
是否引入错误



支持团队的面向技术测试



TDD介绍

- 先有单元测试代码，再有产品代码

```
void testAdd1(){
    add(3);
    add(5);
    assertEquals(8, calculator.getResult());
}
void testAdd2(){
    add(-3);
    add(-5);
    assertEquals(-8, calculator.getResult());
}
void testAdd3(){
    add(-3);
    add(3);
    assertEquals(0, calculator.getResult());
}
```

```
void testAdd4(){
    add(65534);
    add(1);
    assertEquals(null, calculator.getResult());
}
void testAdd5(){
    add(1);
    add(65534);
    assertEquals(null, calculator.getResult());
}
void testAdd6(){
    add(65534);
    add(-65534);
    assertEquals(0, calculator.getResult());
}
```

然后再去写
产品代码、
使得这6个
cases均通过

支持团队的面向技术测试



第一象限测试的工具

- 源代码控制
- IDE
- 构建工具 (ant、Nant、Maven)
- 构建自动化工具 (CruiseControl、CruiseControl.ne、CruiseControl.rb、AnthillPro、Bamboo、BuildBeat、Team City& Pulse)
- 单元测试工具 (xUnit TestNG Abbot SWTBot EasyMock Ruby/Mock)



Test in Agile



支持团队的面向业务测试



第二象限测试与软件质量

内部质量



外部质量



使用质量



软件质量

第一象限

第二象限

支持团队的面向业务测试



需求的产生

用户故事



实例、测试



交流



需求



支持团队的面向业务测试

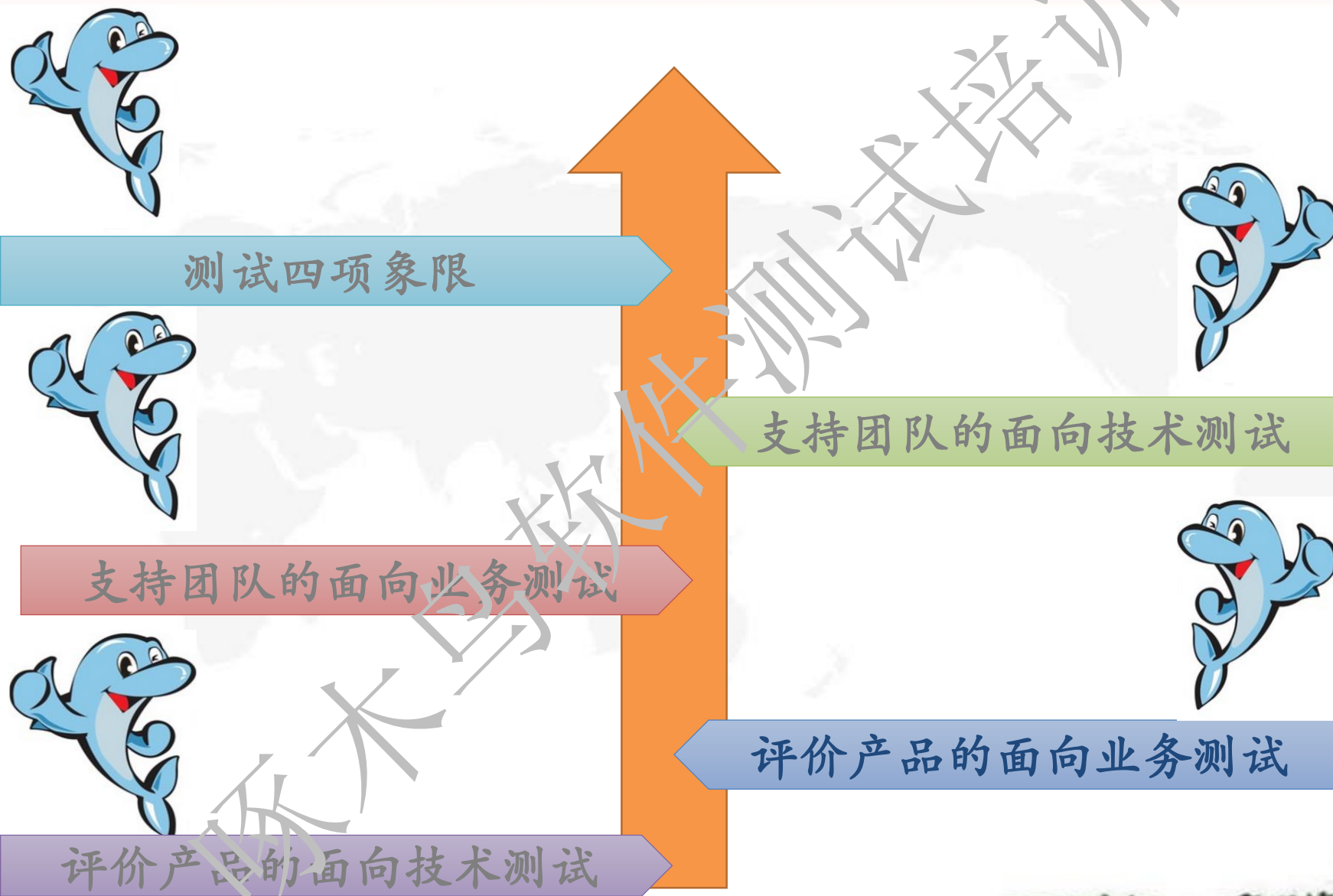


第二象限的测试工作

- 功能测试用例设计
- 编写自动化测试脚本
- 执行自动化测试脚本
- 手工执行手工测试用例执行
- 复测与回归测试



Test in Agile



评价产品的面向业务测试



场景测试

肥皂剧测试：Hans Buwlada[2003]

张小姐听她的闺蜜说她在一家电子商务网站上买了一套床上用品，感觉非常好，张小姐也想买一套。回家吃完晚饭后，她注册了这个网站的个人信息。然后进行了登录，添加了收货信息和支付信息。然后通过分类信息找到了这套床上用品，她把床上用品放入了购物车中。同时她看见了一条推荐的裤子，非常喜欢，进去看了一下详情，可是她通过聊天工具与店商老板聊了聊后得知没有适合她的尺寸的裤子。这个时候张小姐的妈妈打电话过来，正好谈及舅舅的孙子需要一种奶粉，在商场里面没有买到。挂了电话后，张小姐通过这个网上的产品搜索竟然查到了这种奶粉。选择了一个客户评价相对比较好的商店，选择了奶粉，加入了购物车。已经是晚上10点多钟了，于是张小姐决定结束本次网上购物的活动，她进入购物车，选择了床上用品和奶粉，提交了订单，并且准备通过支付宝付款，可是正在付款的时候网络Wi-Fi出现了故障，重新启动Wi-Fi，网络通畅后，还可以重新付款，并且告之付款成功的信息。

三天后床上用品和奶粉都收到了，奶粉很好，她在网上提交了收货的信息，并且给了好评。但是床上用品不太令她满意，布料边角有些地方加工比较粗糙，于是她用手机拍下了部分粗糙的边角图片，上传到网上，与商家协商后，决定退货。商家答应了，张小姐把床上用品快递给了卖家，卖家收到货后把钱退还给了张小姐。

评价产品的面向业务测试



场景测试

1. 注册用户基本信息：在注册的时候，我试图用数据库中已有的ID进行注册，系统给出了友好的提示，没有问题。在输入密码的时候，没有告诉我密码的强弱程度，对于这点，我认为应该报告给产品经理；
2. 注册完毕，进行登录操作，我试图采用SQL注入法，系统已经被屏蔽；
3. 登录进去以后，我修改了基本信息，添加了收货信息和支付信息。在收货信息填写中我试图采用CSS注入法，结果发现开发工程师在这里没有作屏蔽，这个需要作为一个Bug；
4. 通过分类信息，我找到床上用品，进入详情，放个购物车，没有发现问题；
5. 我请另外一位同事作为服装市场的卖家，我进入他的店铺，用我们自己开发的聊天工具进行了聊天的功能测试，除了速度反应有些慢(作为性能Bug处理)，功能倒是没有发现什么问题；
6. 在搜索功能中，我输入了‘%’，见鬼，所有的商品信息都被显示出来了，这个需要作为一个Bug进行处理；
7. 在搜索功能中，输入“惠氏婴儿奶粉”，找到了一个厂商的奶粉，进入详情页面，放入购物车；

评价产品的面向业务测试



场景测试

8. 进入购物车，把奶粉个数改为两份，然后生成订单。在订单中我又发现了一个大的Bug，奶粉的价格仍旧按照一份计算；
9. 我选择了收货地址信息，进行付款操作，操作过程中，我断掉了网；
10. 我把网重新启动后，重新提交了订单，竟然发现付款成功，但是金额为0，一个大Bug被发现；
11. 为了测试确认收货和退货功能，我不得不重新选择了那个床上用品和奶粉，再次生成订单，付款；
12. 付款成功后，我确认了奶粉的订单，并且给了个好评；
13. 我故意把一个exe文件改为jpg文件，进入退货模块，把“jpg文件”上传到网上，非常好的是系统判断了上传上去的jpg文件格式非法；
14. 我用手机拍下了一张图片，重新上传，并且成功了；
15. 我又开了一个浏览器窗口，用床上用品卖家的账号登录，确定了退货；
16. 在张女士的浏览器窗口，检查到了床上用品的钱被退了回来。

评价产品的面向业务测试



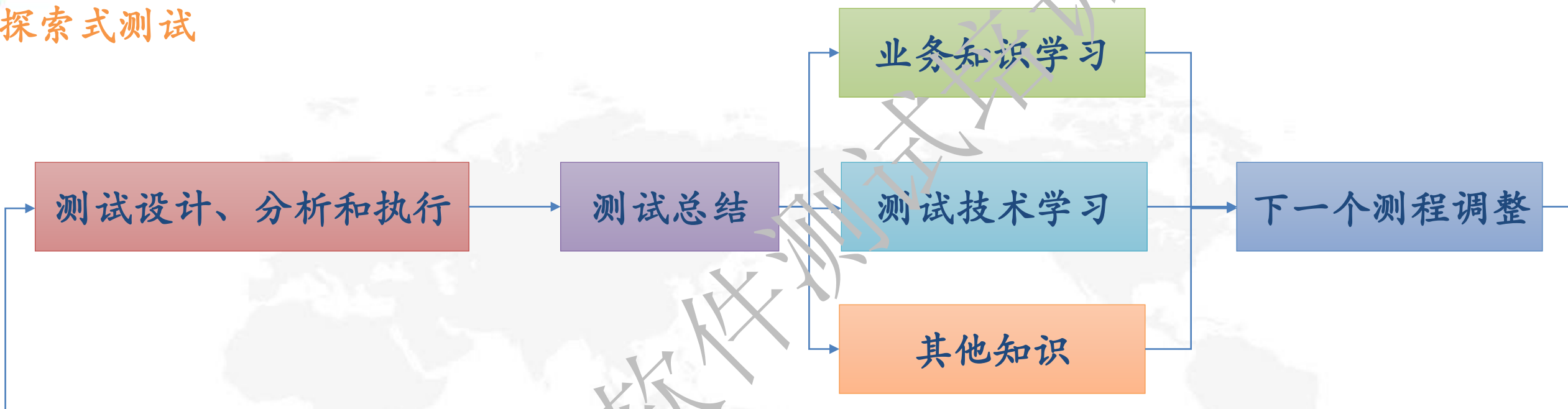
探索式测试

“探索式软件测试”是软件测试专家Cem Kaner博士在1983年提出的，并受到语境驱动软件测试学派（Context Driven Testing School）的支持。随着近年来敏捷开发的出现，探索式软件测试的理论由于符合快速提交的理念，也被重新提出，并且受到了广泛地重视。其实我们以前在软件测试中也有意无意地使用到探索式软件测试中的一些方法，就是没有把它升华到一个理论的高度。探索式软件测试（Exploratory Testing）：是一种自由的软件测试风格，强调软件测试工程师展开软件测试学习、软件测试设计、软件测试执行和软件测试结果评估等活动，以持续优化软件测试工作。

评价产品的面向业务测试



探索式测试



- 测试分析，设计，执行属于一个测程，一个测程一般在0.5-3小时；
- 测程完毕根据测试结果进行经验总结；
- 根据测程的反馈信息进一步学习相关业务，测试及其他知识；
- 根据测试结果及其学习情况计划下一个测程策略。



评价产品的面向业务测试



可用性测试

用户需求&角色测试

导航测试

研究竞争对手



评价产品的面向业务测试



用户界面的背后

API测试

接口测试



评价产品的面向业务测试



测试文档

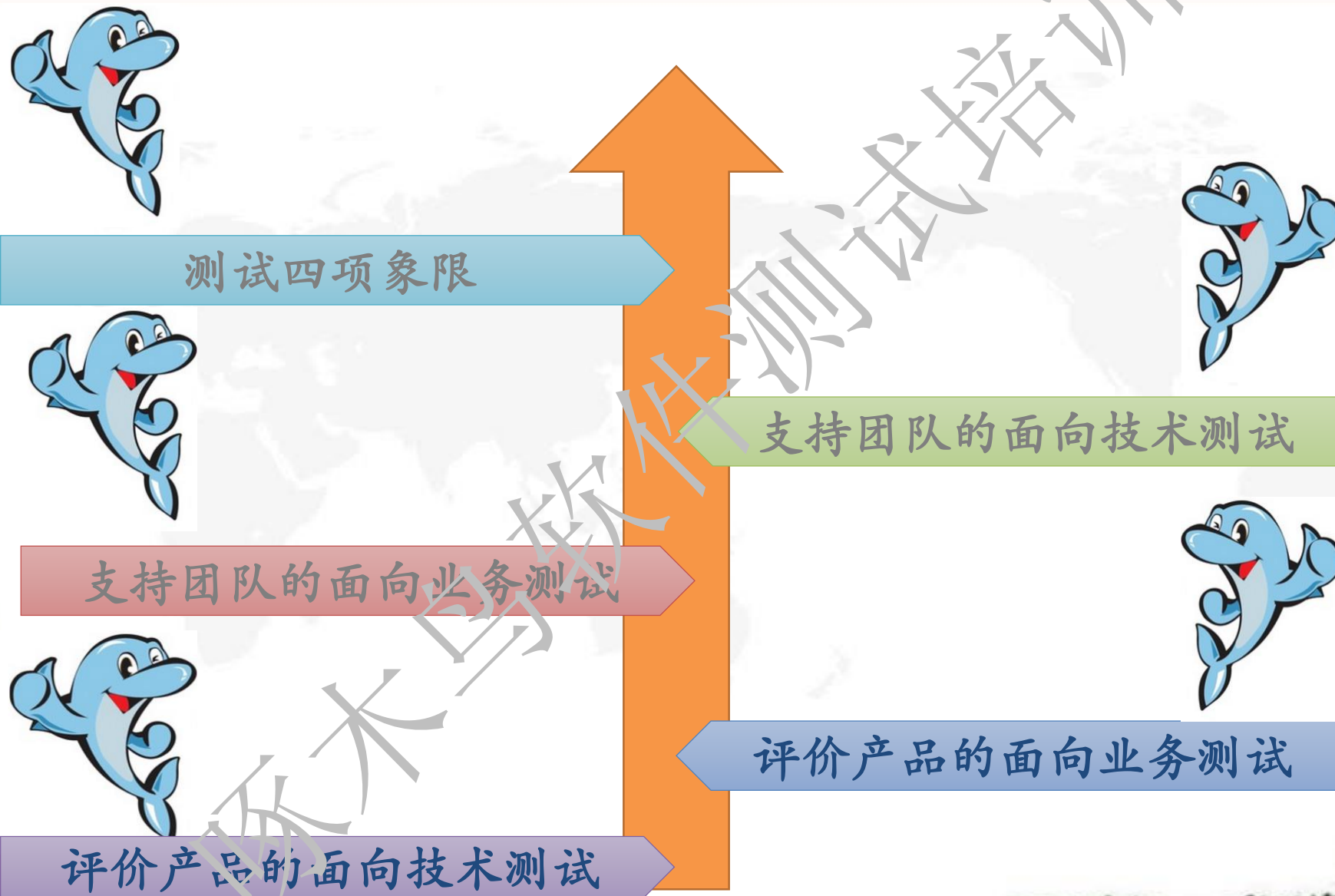
用户手册

广告

...



Test in Agile



评价产品的面向技术测试



ISO225000（替代ISO 9126）软件质量模型（内部质量，外部质量）



评价产品的面向技术测试



信息安全性:指的是软件产品保护**信息**和**数据**的能力, 以使**未授权的人员**或者**系统不能阅读或者修改**这些**信息**和**数据**, 而不拒绝授权人员或者系统对它们的访问。

信息安全性测试的关注点

1. 对应用程序/数据进行**未授权的复制**;
2. **未授权的访问控制**;
3. 出入域溢出导致的**缓存区溢出**;
4. **服务拒绝**, 阻止用户与应用程序的交互;
5. 在网络上**窃听**数据传获取敏感信息;
6. 破解保护**敏感信息**的加密代码;
7. **逻辑炸弹/复活节彩蛋**。

评价产品的面向技术测试



信息安全性分类

- 与**用户**接口相关；
- 与**文件**系统相关；
- 与**操作系统**相关；
- 与**外部软件**相关。

信息安全性测试方法

- 使用工具来**创建系统概况或网络图**；
- 使用多种工具**进行漏洞扫描**；
- 获得信息研制“**攻击方案**”；
- 根据**安全专家（白帽子黑客）**的建议进行多种攻击。

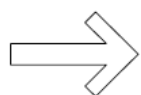
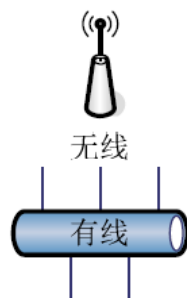
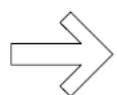
参考：James Whittaker写的《How to Break Software Security》(Addison Wesley, 2003)

评价产品的面向技术测试



案例分析：信息安全性

开发计划
市场策略
销售计划
测试报告
年度规划
员工信息



公司网络

家庭网络

公共热点

1. 硬盘加密;
2. 屏幕保护;
3. 防病毒软件;
4. 防火墙;
5. Windows下的共享文件;
6. Windows用户帐户密码;
7. Windows Service Pack等级;
8. 安全性补丁;
9. 远程桌面连接;



评价产品的面向技术测试



案例分析

某公司开发一套网上答题的系统，全部题目为单项选择题，可以选择A B C D四种中的任意一项，每一周评选最高得分者，且给他增加一定的点分。为了防止网友对所有题一次性多次选择某同一个答案，比如对所有题都选择D，程序里做了控制，如果连续5次选择同一个答案的网友被自动踢出答题系统。程序经过严格测试后上线使用。可是上线不到四个月，发现每周最高得分者均为一个叫张先生，查看其答案，竟然所有题目都答成B，让所有开发者很是奇怪。两周后公司的开发经理在网站群聊中找到了这位张先生，张先生告诉开发经理，你们系统在前端JavaScript做了控制，但是在后端JavaBean中没进行控制，所以他使用Delphi绕过前端，编了一个死循环，24小时发送答案D给每一题目。

评价产品的面向技术测试



案例分析：肉机



评价产品的面向技术测试



CSS注入

文章标题

输入

SQL注入

文章标题

查找

Select * from table where title like '%ver%'

ver is %

Select * from table where title like '%%%'

Select count(*) from user where name=\$name and password=\$password

\$name="Jerry", \$password="123456 or 1=1"

Select count(*) from user where name=jerry and password=123456 or 1=1

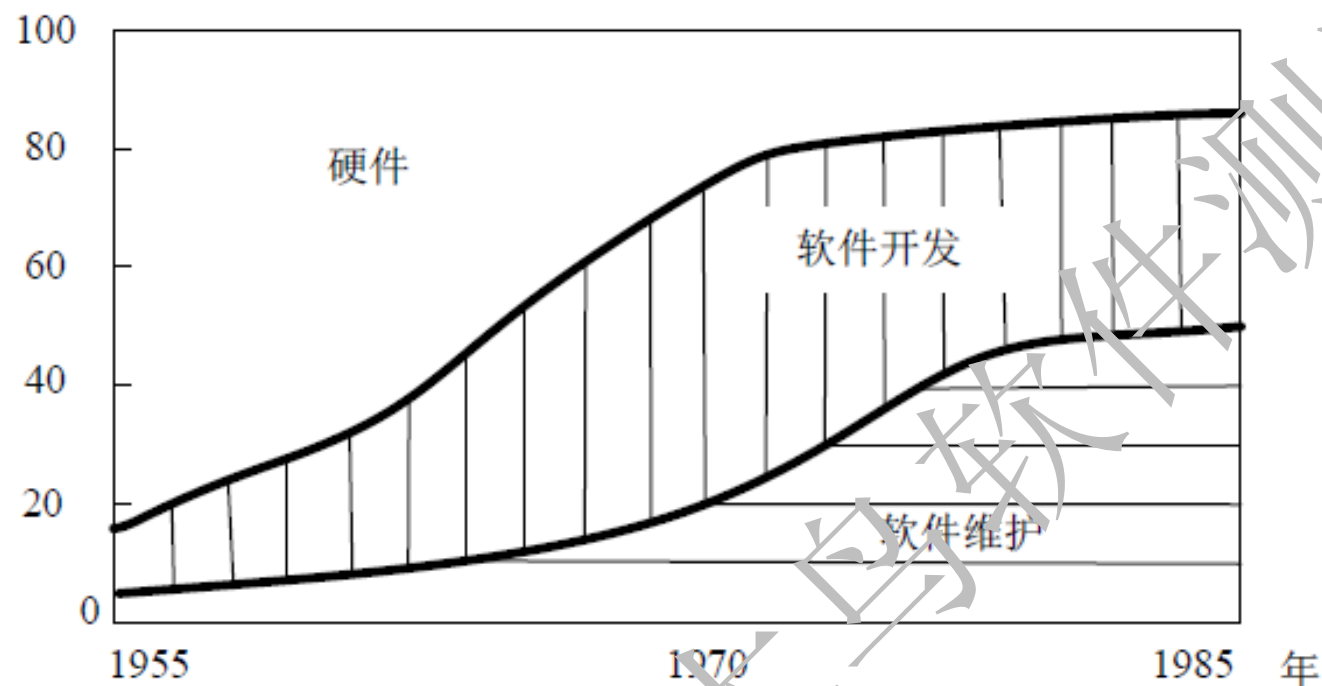


评价产品的面向技术测试



可维护性：可维护性指的是软件产品可被修改的能力，修改包括纠正、改进或者软件对环境、需求和功能规格说明变化的适应能力。

总成本的百分比



Robert W. Noel, DoD Standards: Hindrance or Help to Software Maintainability?, IEEE, 1986

评价产品的面向技术测试



代码可维护性：代码要留出一些接口，便于二次开发，多些变量，少些hardcode等等

(code Review的方式)

```
A:=FALSE;↵
WHILE NOT A DO↵
  IF B.C=B.J THEN↵
    B.E:=B.E+D.F;↵
    IF G.EOF THEN↵
      A:=TRUE;↵
    ELSE↵
      ReadBlock(G,D);↵
    END;↵
  ELSE↵
    WriteBlock(h,B);↵
    ReadBlock(I,B);↵
  END;↵
END;↵
```



```
EndOfUpdate:=FALSE;↵
WHILE NOT EndOfUpdate DO↵
  IF UserRec.id=UserRec.UpdateNumber THEN↵
    UserRec.Used:=UserRec.Used+UpdateRec.ResourcesUsed;↵
    IF UpdateFile.EOF THEN↵
      EndOfUpdate:=TRUE;↵
    ELSE↵
      ReadBlock(UpdateFile,UpdateRec);↵
    END;↵
  ELSE↵
    WriteBlock(NewUsersFile,UserRec);↵
    ReadBlock(UserFile,UserRec);↵
  END;↵
END;↵
```



评价产品的面向技术测试



打好的补丁运行在源程序上是否可以运行?

补丁

源程序



评价产品的面向技术测试



相容性指的是软件产品与一个或者更多的**规定系统之间进行交互的能力**。互操作性测试用于验证软件产品或者应用程序在各种指定的目标环境下是否可以正常工作。主要包括：

1. 硬件
2. 软件
3. 中间件
4. 操作系统...

互操作性测试

输入的互操作

输出的互操作

自适应性

评价产品的面向技术测试



南向接口，北向接口

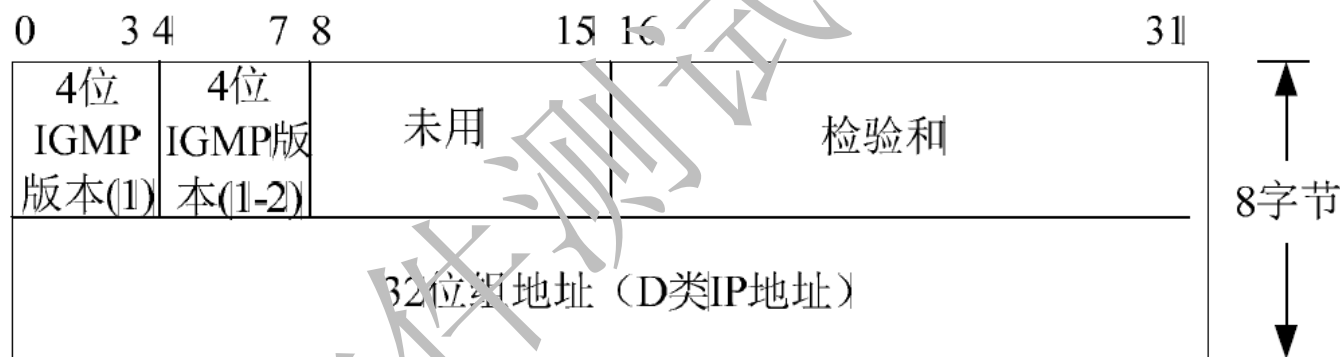


评价产品的面向技术测试



案例分析

输入与输出的互操作性测试



IGMP v1 (RFC 1112)

IGMP v2 (RFC 2236)

IGMP v3 (RFC 3376)

互操作的自适应能力

1, 10M/100M/1000的速率的自适应能力;

2, IGMP v1(RFC 1112), IGMP v2(RFC 2236)和IGMP v3(RFC 3376)的自适应能力,

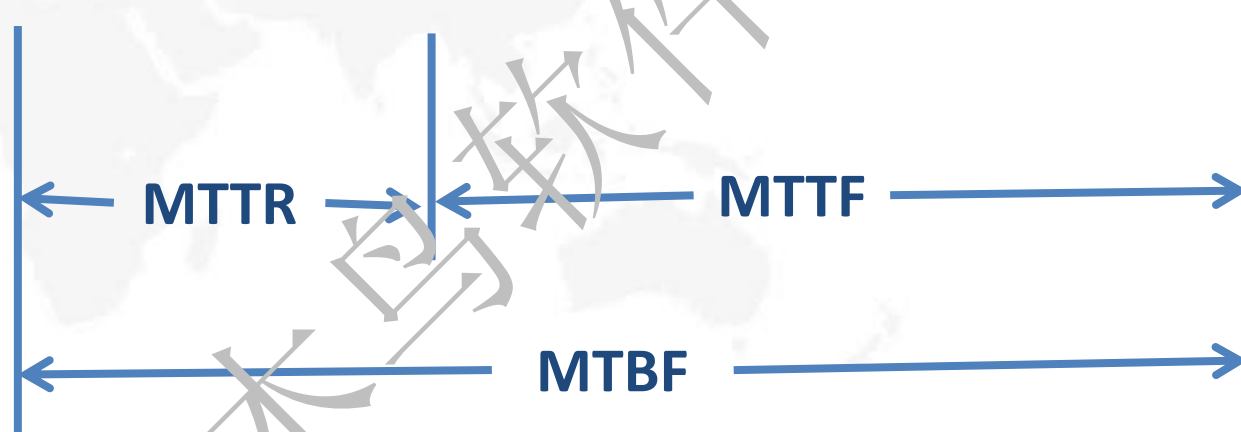
评价产品的面向技术测试



可靠性测试的目的之一是对软件成熟度在时间上的统计度量指标进行监控，并将其与既定目标相比较。例如：

1. 平均**失效间隔时间**MTBF;
2. 平均**修复时间**MTTR;
3. 平均**失效前时间**MTTF;

监控得到的数值在时间上的分布，可以表示为可靠性增长模型。



$$\text{MTBF} = \text{MTTR} + \text{MTTF}$$

评价产品的面向技术测试



可靠性测试的3个子质量特性

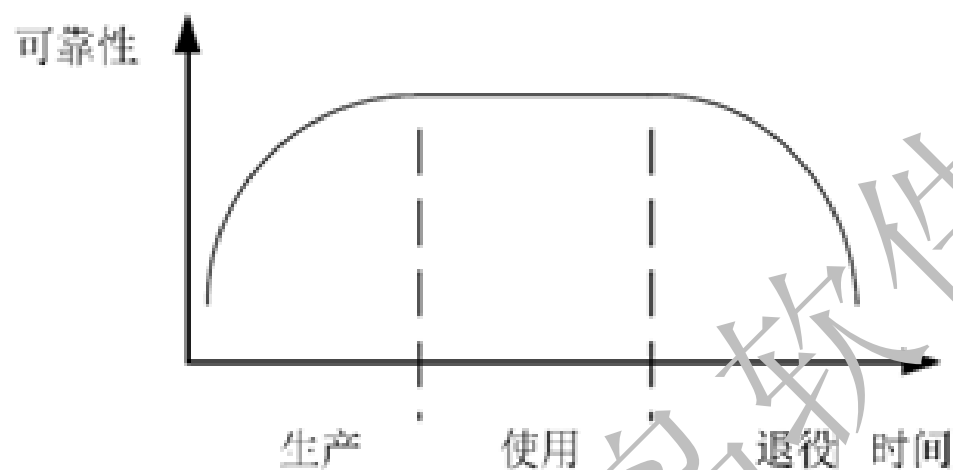
1. 成熟性 (Maturity) : 软件产品为避免由于软件中的故障或者缺陷而导致失效的能力。
2. 容错性 (Fault Tolerance) : 在软件出现故障或者会违反其指定接口的情况下, 软件产品维持规定的性能级别的能力。
3. 易恢复性 (Recoverability) : 在失效发生的时候, 软件产品重建规定的性能级别并恢复受直接影响的数据的能力。



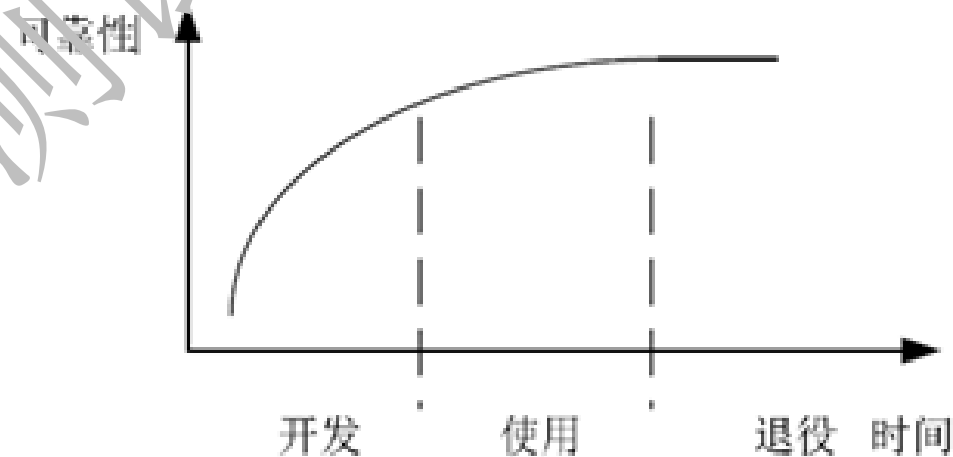
评价产品的面向技术测试



可靠性曲线分析



硬件的可靠性变化



软件的可靠性变化



评价产品的面向技术测试



案例分析：故障恢复测试

产生故障方法：

- 服务器断电；
- 客户机断电；
- 网络断网；

恢复后检查：

- 程序功能是否正常？
- 数据库及表中的字段是否正常？
- 故障发生中正在交易的事务如何？是否下笔交易仍旧可以进行、
- 铁路、飞机、航天等系统发生故障进入安全模式

评价产品的面向技术测试



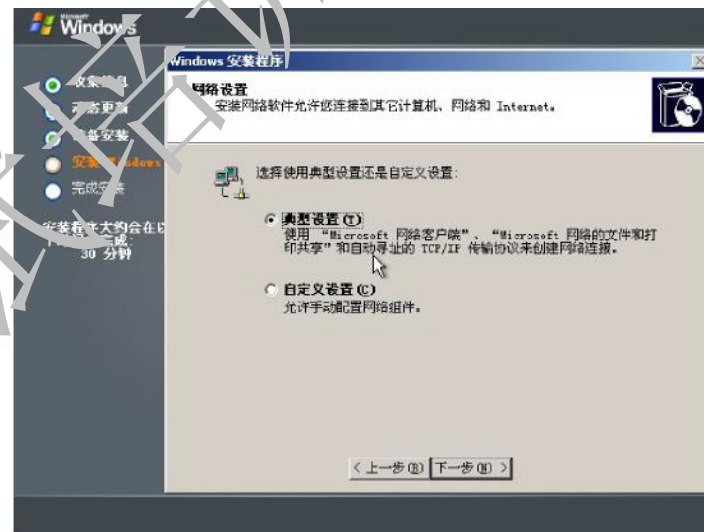
1. 可移植性测试通常和软件移植到某个特定的运行环境中的难易程度相关，包括第一次建立或从现有环境移植到另一个环境。测试类型包括
2. 可安装性测试
3. 共存性/兼容性测试
4. 适应性测试
5. 可替换性测试

评价产品的面向技术测试



案例分析：可安装性可以检查：

1. 正确处理安装过程中所出现的失败；
2. 测试不完全的安装/卸载能否完成；
3. 测试是否可以识别无效的硬件平台或操作系统配置；
4. 验证是否可以成功地进行软件卸载（卸载数据文件必须干净）



评价产品的面向技术测试



案例分析：

某网络设备A想移植网络设备到网络设备B中，发现错误，经过排查，结论是网络设备A IP地址用的是用户地址序列（高位在前，地位在后）；而网络设备B IP地址用的是网络地址序列（低位在前，高位在后）

192.168.0.8->C0.A8.00.08

用户地址序列:C0A80008

网络地址序列:0800A8C0

评价产品的面向技术测试



案例分析

某用户把系统从阿里云迁移到百度云，他需要考虑迁移过来的系统是否可以正常运行。

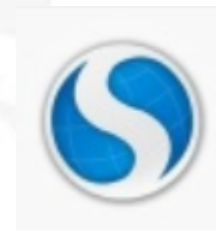


评价产品的面向技术测试



兼容性

- 数据库兼容性
- 操作系统兼容性
- 浏览器兼容性
- 插件兼容性
- 上下版本兼容性
- ...



评价产品的面向技术测试



效率（性能）的子质量特性：

1. **性能测试**：要评价系统或组件的性能是否和具体的性能需求一致；
2. **负载测试（LOAD测试）**：通过**增加负载**来评估组件或系统的性能的方法；
3. **压力测试**：评估系统处于或**超过预期负载**时系统的运行一段时间；（一般为最大负载**80%**，运行**48小时**）

评价产品的面向技术测试



案例分析：性能测试问题

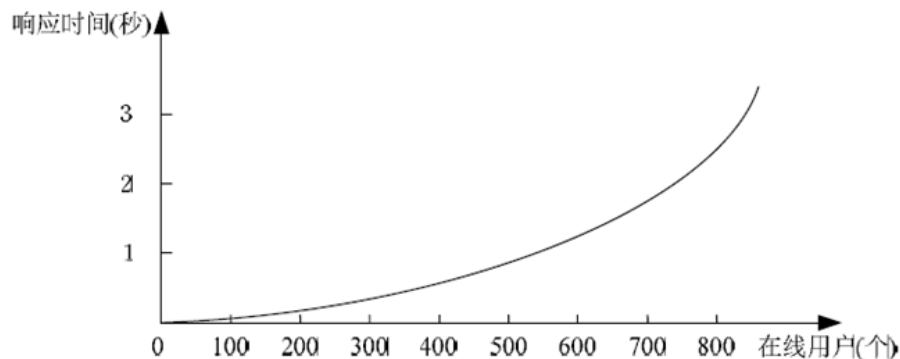
页面加载时间	愿意等待的用户比例
10秒	84%
15秒	51%
20秒	26%
30秒	5%

B. M. Subraya, Integrated Approach to Web Performance Testing: A Practitioner's Guide, IGI Global, 2006

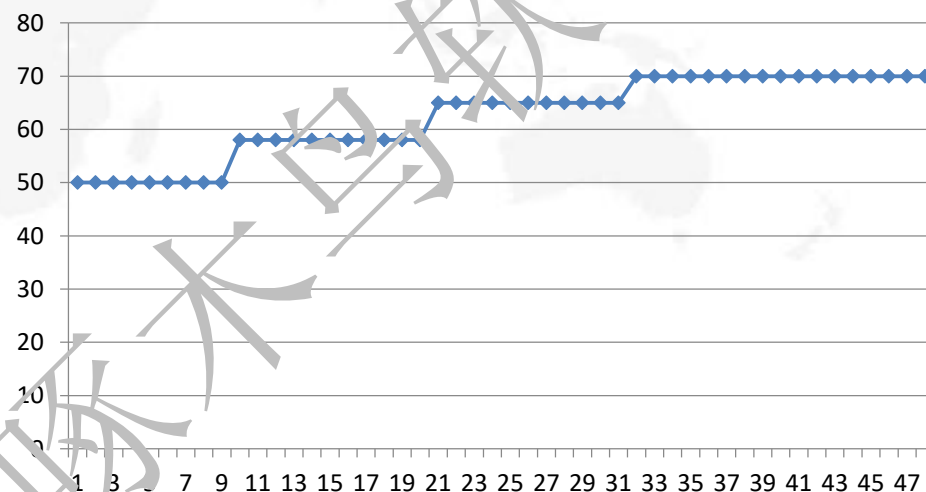
2/5/10



评价产品的面向技术测试



案例分析： 负载测试

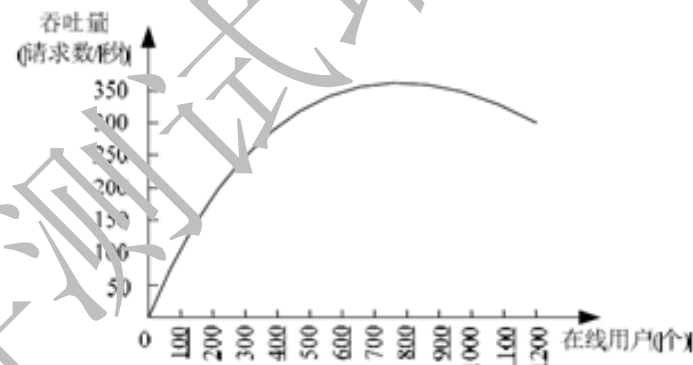


案例分析： 压力测试



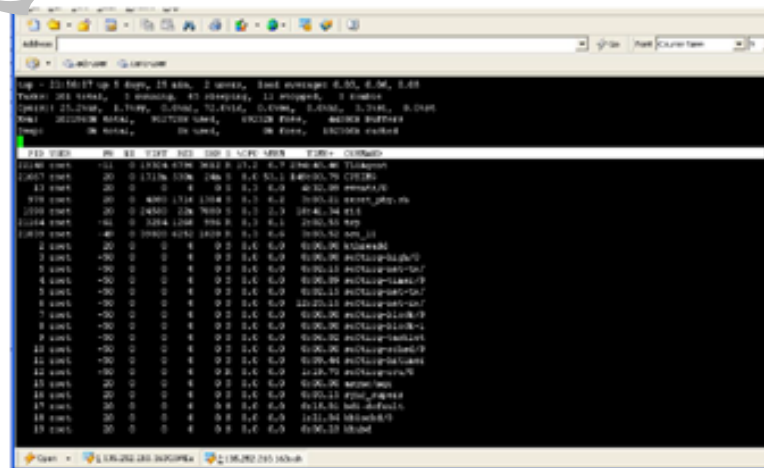


案例分析：效率测试度量



1. 响应时间
2. 带宽、吞吐量、时延、丢包率、点击量/秒
3. 资源使用情况

- CPU
- Memory
-



评价产品的面向技术测试



性能测试的范围定义：

- 和整个开发团队一起确认性能测试的范围；
- 系统中被频繁使用的功能，调用的接口等（80/20原则）；
- 系统中涉及到大量数据库读写的功能；
- 大量读写系统缓存的功能；
- 感觉速度慢的功能；

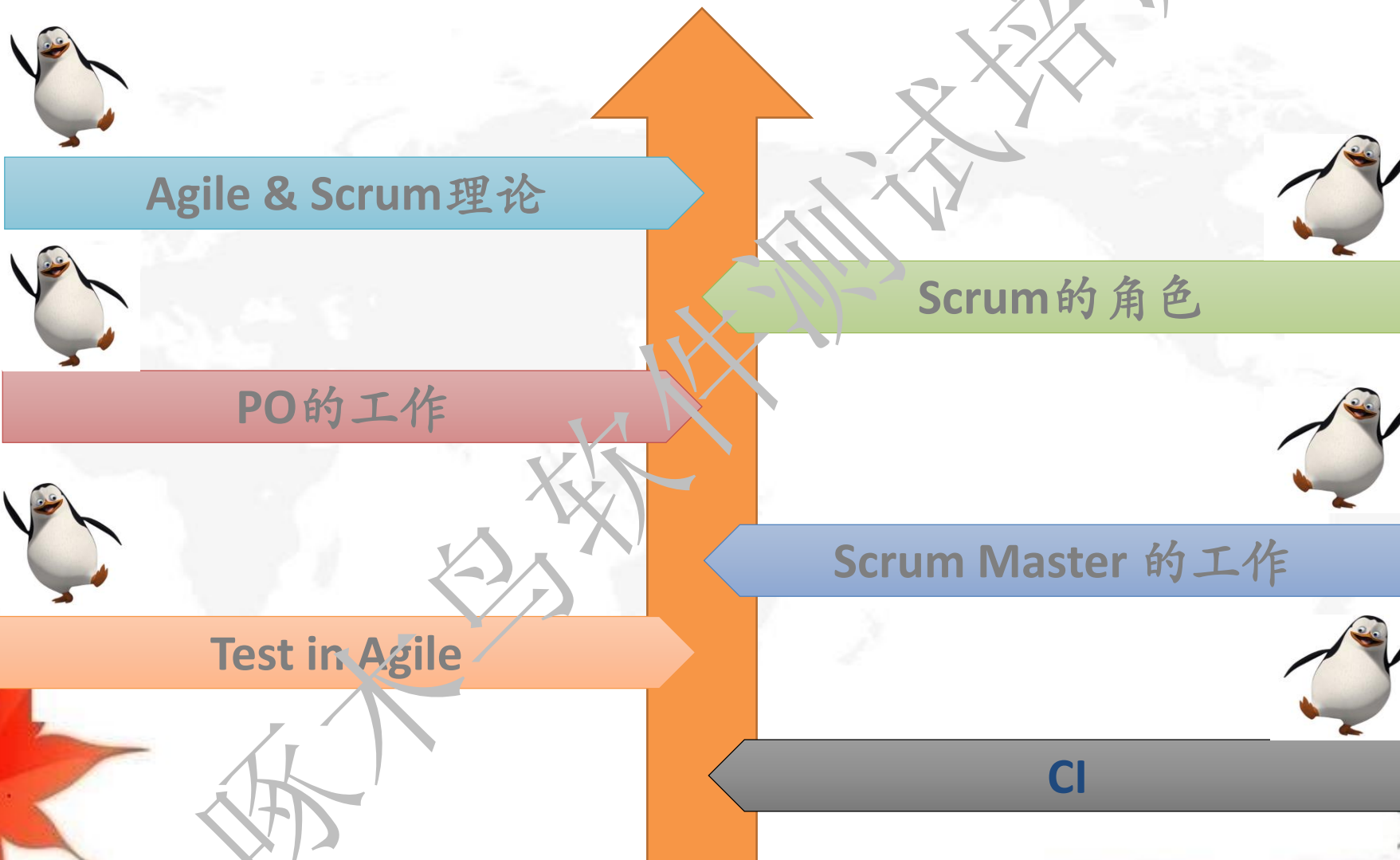
评价产品的面向技术测试



案例分析：网站查询新闻效率测试

- 性能测试：正常情况下查询，返回时间；
- 负载测试：
 - 保持1000条新闻记录，将查询并发从10000开始，每次增加1000，查看响应时间，CPU，内存情况；
 - 保持1000人并发，将新闻个数从100000条开始，每次增加10000条，查看响应时间，CPU，内存情况。
- 压力测试：
 - 保持80%的最高并发数，1000条新闻记录，运行48小时，持续监测响应时间，CPU，内存情况；
 - 保持80%的最高新闻记录，1000个并发，运行48小时，持续监测响应时间，CPU，内存情况；
 - 保持40%的最高并发数，40%的最高新闻记录，运行48小时，持续监测响应时间，CPU，内存情况。

目录





环境要求

- **无人值守**：自动化程度非常高
- **适应不同项目的需求**：可以定制或配置环境
- **管理成本低**：比如虚拟化主机
- **On-Demand按需分配**：动态调度资源
- **支持相应的开发流程和配置管理**





步骤



CI



构建工具

JAVA: Ant, Maven, Gradle(Groovy),

Buidr





集成管理工具

- Jenkins/Hudson (2011更名)
- Travis CI (支持DSL,Github,RESTful,Chef等)
- CruiseControl(.net,.rb)

知名工具Jenkins: <http://jenkins-ci.org/>





在CI中测试分量

- 持续检查
- 持续构建
- 持续验证
- 持续部署
- CI基础设施
- 持续报告



- 自动代码静态测试
- 自动单元动态测试(如JUnit)
- 自动部署SUT (System Under Test)
- 自动BVT (Build Verification Test)
- 自动生成测试报告





自动部署

- 使用Jenkins的插件
- 调用Maven完成自动部署
- 不用插件，使用自己定义脚本



结束



Thank You

顾翔

啄木鸟软件测试培训网

QQ: 2025344

微信号: xianggu0625

Email: xianggu625@126.com

网站: www.3testing.com

微信公众号: 见二维码

