

敏捷软件 发展

乔恩·伯德博士

jon.bird@bristol.ac.uk

感谢 Simon Lock 博士为本单元的早期版本开发了许多幻灯片。

图片免版税来自 www.pexels.com

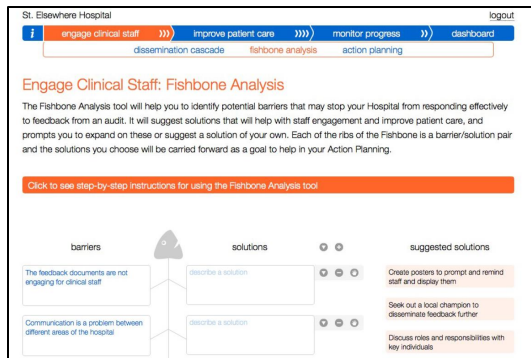
今天的讲座

- 使用瀑布方法开发的数字健康项目，这是上周的焦点
- 软件开发的瀑布方法与敏捷方法
- 敏捷软件开发，包括：极限编程（其中包括结对编程）；测试驱动开发；混乱；和看板
- 使用敏捷方法开发的数字健康项目
- 推荐阅读

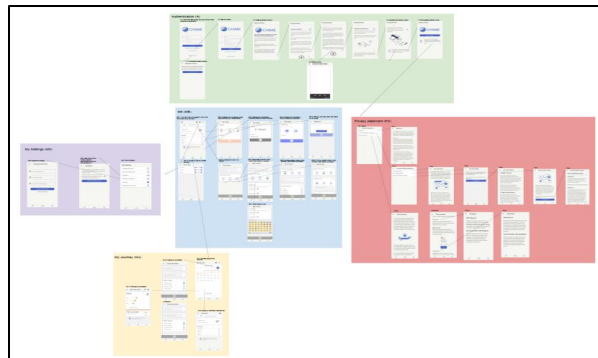


数字健康项目

- 我的研究涉及开发数字技术来解决健康问题
- 我在研究中采用什么软件开发流程？



亲和力



钟声

亲和力-动机

- 英国每年有300万次输血

- 大约20%是不必要的

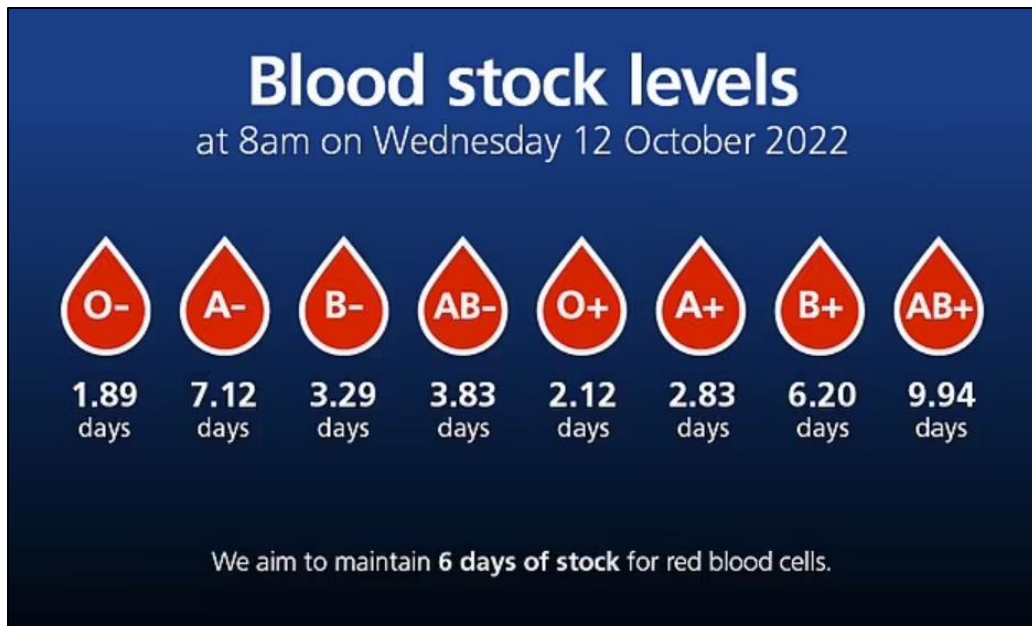
- 血液成分是稀缺资源

- 血液存在健康风险

输血

- 如何减少不必要的血液

输血？



Affinitie – 软件

- 我被邀请加入一个现有的研究项目，开发一个用于输血的网络工具包

- 研究团队由健康心理学家、

统计学家、医生和 NHS 血液和移植

- 目的是帮助输血从业者将输血审核结果传播给

医院里的每个人

St. Elsewhere Hospital logout

[i](#) [engage clinical staff](#) [improve patient care](#) [monitor progress](#) [dashboard](#)

[dissemination cascade](#) [fishbone analysis](#) [action planning](#)

Engage Clinical Staff: Dissemination Cascade

The Dissemination Cascade tool will help you to identify staff involved in transfusion decision-making. You will be able to indicate who is responsible for giving them feedback documents. Each of the dissemination choices you indicate here will then be carried forward as a goal to help in your Action Planning.

[Click to see step-by-step instructions for using the Dissemination Cascade tool](#)

Transfusion Practitioner informs...

▶ **Hospital Transfusion Committee** ▲ ▼ 🔍

What is disseminated? [select option](#)

How are they informed? [select option](#)

When by? [select date](#)

Named contact? [enter name](#)

▶ **Hospital Transfusion Team** ▲ ▼ 🔍

What is disseminated? [select option](#)

How are they informed? [select option](#)

When by? [select date](#)

Affinitie – 软件开发

- 我们得到了一套明确的要求由研究团队

- 他们已经为输血从业人员开发了一套纸质工具来帮助他们传播血液

输血审核结果

- 软件开发
方法：瀑布

St. Elsewhere Hospital [logout](#)

[i](#) [engage clinical staff](#) >>> [improve patient care](#) >>> [monitor progress](#) >> [dashboard](#)

[dissemination cascade](#) [fishbone analysis](#) [action planning](#)

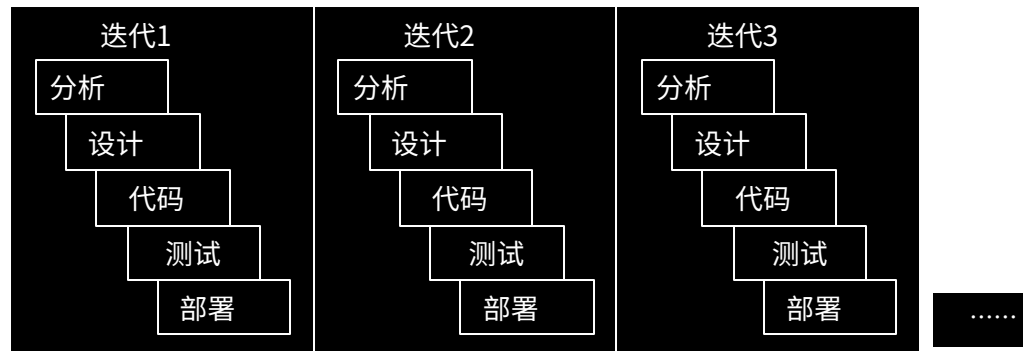
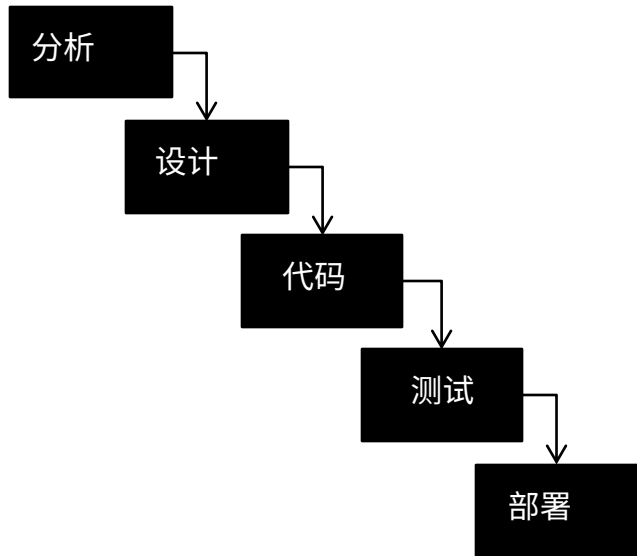
Engage Clinical Staff: Fishbone Analysis

The Fishbone Analysis tool will help you to identify potential barriers that may stop your Hospital from responding effectively to feedback from an audit. It will suggest solutions that will help with staff engagement and improve patient care, and prompts you to expand on these or suggest a solution of your own. Each of the ribs of the Fishbone is a barrier/solution pair and the solutions you choose will be carried forward as a goal to help in your Action Planning.

[Click to see step-by-step instructions for using the Fishbone Analysis tool](#)

The interface displays a fishbone diagram with three ribs. The left rib is labeled 'barriers' and contains two text boxes: 'The feedback documents are not engaging for clinical staff' and 'Communication is a problem between different areas of the hospital'. The right rib is labeled 'solutions' and contains two text boxes: 'describe a solution' and 'describe a solution'. The far right section is labeled 'suggested solutions' and contains three text boxes: 'Create posters to prompt and remind staff and display them', 'Seek out a local champion to disseminate feedback further', and 'Discuss roles and responsibilities with key individuals'. There are navigation icons (back, forward, search) between the barriers and solutions sections.

瀑布式生命周期与敏捷生命周期



什么是敏捷软件开发？

- 敏捷是一种软件开发思维方式
- 2001 年冬天，17 位软件开发人员在犹他州的一个滑雪胜地会面，起草了一份宣言，概述了开发文档驱动软件的替代方法

当时的发展进程

- 宣言简洁，提出了软件开发的四个关键价值观

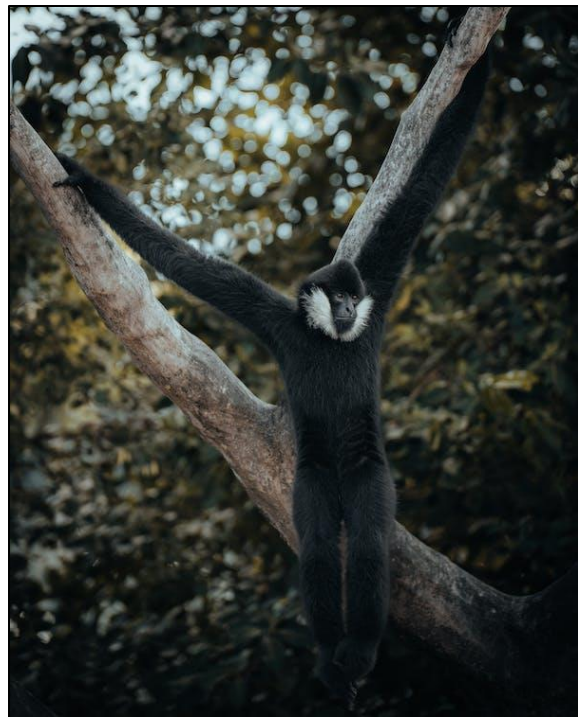


敏捷软件开发宣言

我们通过实践并帮助他人开发软件，从而发现更好的软件开发方法。通过这项工作，我们认识到：

个体和互动超越**流程和工具** **工作软件**过于全面的文档 **客户协作**合同谈判 **应对变化**过度遵循计划

也就是说，虽然右边的物品有价值，但我们更看重左边的物品



你应该如何阅读宣言？

- 这些价值观是有目的的挑衅性的，以便让人们思考软件开发
- 敏捷宣言是**不是**提议我们忽略软件开发的各个方面，例如流程、工具和文档
- 相反，敏捷宣言希望人们考虑进行软件开发各方面的替代方法



敏捷是由程序员为程序员创建的

编码员喜欢

编写高质量代码

从他们的待办事项清单上划掉一些事情

通过向客户展示工作软件来给他们留下深刻印象

编码员不喜欢

编写大量文档

提前承诺最终设计

进行微观管理

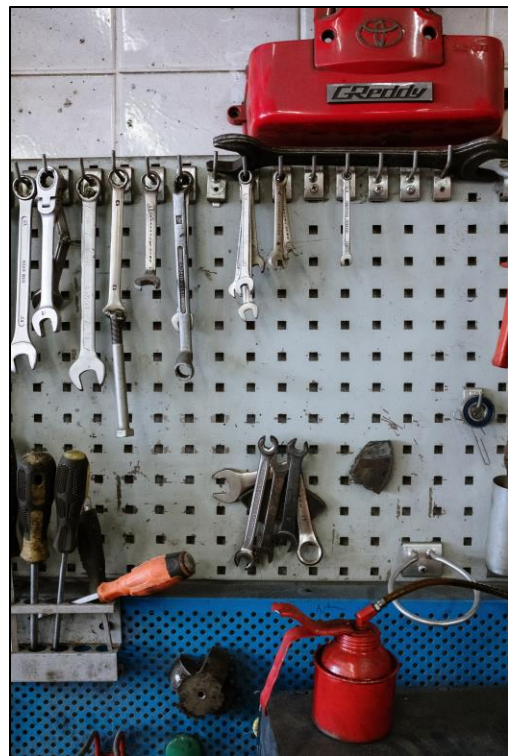
努力完成大而不可更改的最后期限

十二条敏捷原则

满足客户的需求	满足编码员的需求
最重要的是让客户满意（通过尽早且持续地交付工作软件）	以稳定、可持续的速度工作（没有英雄般的努力）
拥抱变化（即使是在开发周期的后期）	依靠自组织团队
每天与客户合作	团队定期反思他们的表现
使用面对面的交流	进度是通过生成的工作代码量来衡量的
经常交付工作软件	持续关注卓越技术
	尽量减少不必要的工作量
	围绕积极进取的个人建立团队

敏捷方法

- 有多种方法可以遵循敏捷价值观和原则
- 不同的公司选择不同的方法
- 流行的方法包括：
 - 极限编程（XP）（两位共同创建者是宣言的签署人）
 - 测试驱动开发（创建者是签名者）
 - 看板
 - Scrum（两位共同创建者是签署者）
- 我们将介绍每种方法的一些关键实践，我们认为这些实践将对您在本单元和夏季项目中有用
- 阅读中有一些方法的链接



如果其中任何一条适用于您，请举手

- 你的代码结构复杂且“复杂”
- 您主要靠自己工作
- 在小组项目中，您只负责自己的代码
- 您以自己的风格编写代码
- 有些周末你工作，所以有些“通宵工作”
- 你的代码在“英勇爆发”中发展



极限编程精神

- **简单的设计：**用最简单的方式实现功能
- **可持续的步伐：**努力是持续且可控的
- **编码标准：**团队遵循商定的风格和格式
- **集体所有制：**每个人都拥有所有代码
- **全团队方法：**每个人都包含在一切之中



极限编程实践

- **结对编程：** 三个臭皮匠顶个诸葛亮
- **测试驱动：** 确保代码正确运行
- **小版本发布：** 经常交付并获取客户的反馈
- **持续集成：** 确保系统正常运行
- **重构：** 当事情变得混乱时重组系统



更详细的结对编程

代码由两个程序员在一台机器上编写：

- 这**舵**使用键盘和鼠标并进行编码
- 这**战术家**思考影响和潜在问题
- 沟通对于结对编程的工作至关重要
- 结对编程促进项目沟通
- 两人并不“拥有”该代码 - 任何人都可以更改它
- 配对可以（并且应该）随时演变
- 所有代码在编写时都会经过审查
- 这**战术家**非常适合推荐重构



结对编程的影响

研究评估了结对编程的影响并确定了许多好处

单个程序员每月 77 行源代码，而结对编程每月 175 行源代码

[詹森，2005]

开发时间成本增加 15%，但提高了设计质量，减少了缺陷，降低了人员配置风险，增强了技术技能，改善了团队沟通，并且在统计显着水平上被认为更令人愉快。

[科伯恩和威廉姆斯，2000]



测试驱动开发的更多细节

- 测试是在任何代码之前编写的，它们驱动所有开发
- 程序员的工作是编写代码以通过测试
- 如果某个功能没有经过测试，那么它就不会被实现
- 测试是系统的要求



测试驱动开发的好处

●代码覆盖率

我们可以确定编写的所有代码都至少有一个测试，因为如果没有测试，代码就不会存在

●简化调试

如果测试失败，那么我们知道它一定是由上次更改引起的

●系统文档

测试本身是文档的一种形式，因为它们描述了代码应该做什么



Scrum

- Scrum 是一种项目管理方法
- 一些关键概念是：
- **Scrum-A**起来整个团队的每日会议
- **敏捷大师**-队长
- **短跑**-短而快速的开发迭代
- **产品积压**-待办事项 需要完成的工作清单
- **产品拥有者**-客户（或其代表）

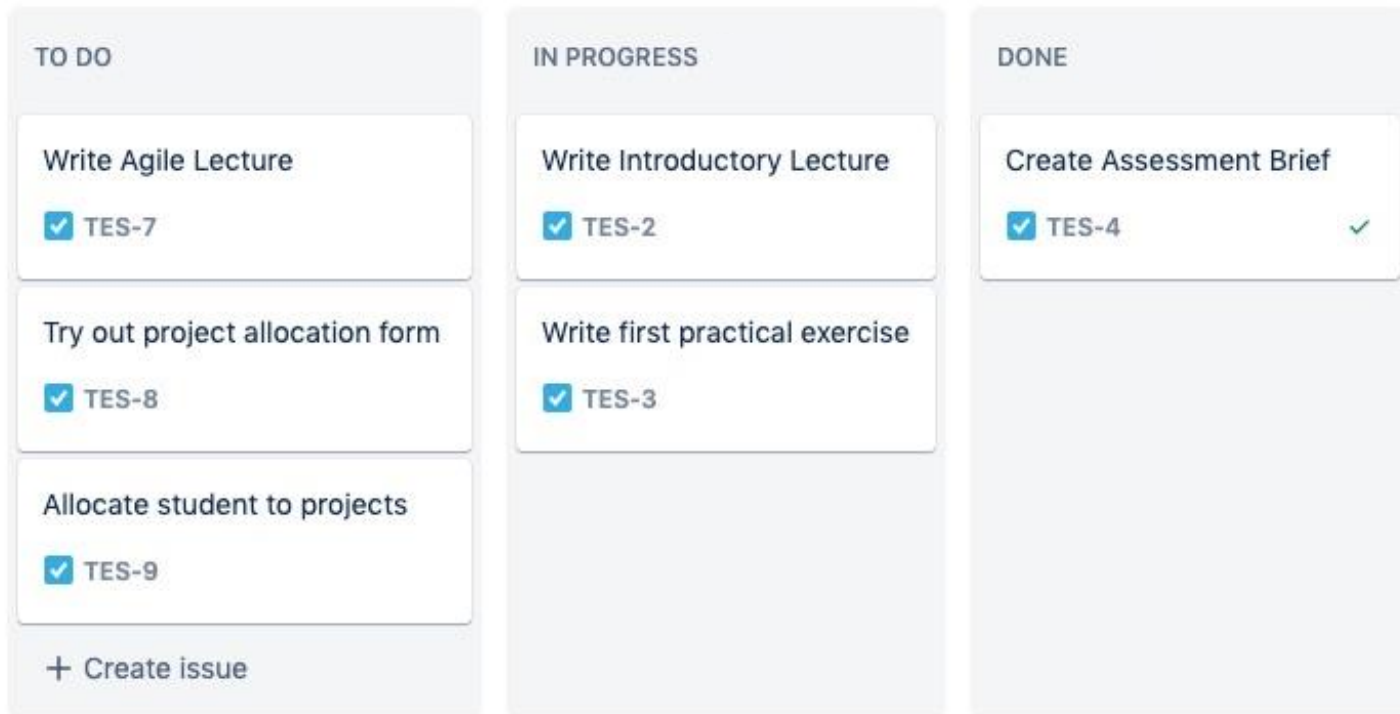


看板

- 该概念源自看板方法，该方法于 2007 年首次定义，但源自丰田在 20 世纪 50 年代为准时制造而开发的调度系统
- 在日语中，“看板”的意思是“视觉板”或“标志”
- 它基本上是一个灵活的“待办事项”列表工具
- 从“待办事项”到“已完成”的各种状态的问题进展
- 它最初是作为白板上的便利贴实现的
- 各种数字工具现在可以实现相同的功能，例如 Jira



Jira 看板



柱（泳道）

- 通常使用三列
- 但 Jira 允许您自定义布局
- 例如，您可能有以下列：
 - 积压
 - 正在验证中
 - 等待整合
- 做对你的团队有用的事情，但确保你有一个“完成”栏

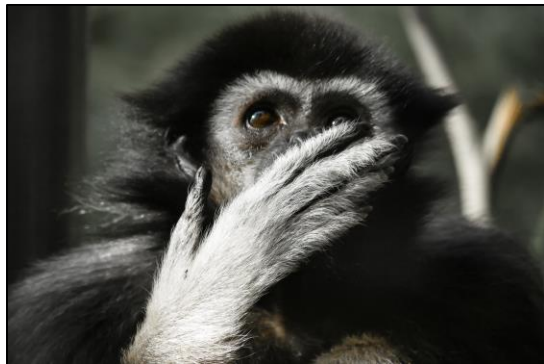


敏捷的问题

- 很难起草具有法律约束力的合同 - 从来没有提前写出完整的规范
- 当您有一个干净的开始并且不受以前工作的限制时，有利于绿地开发。然而，对于涉及改善和改变的棕地开发来说，它并不是那么有效。

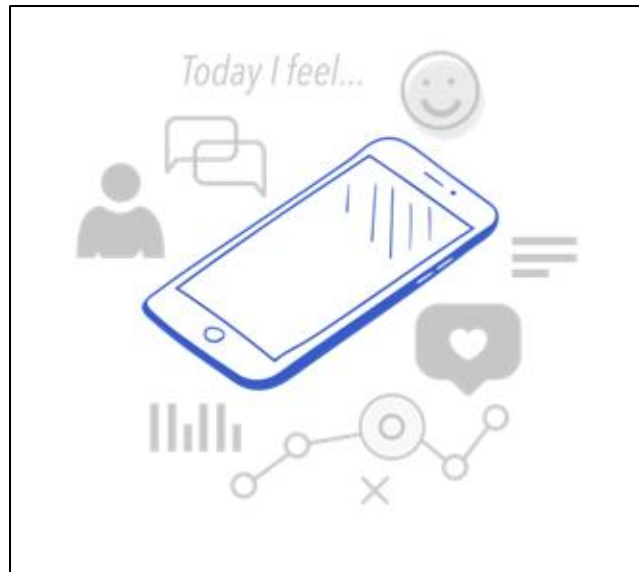
维护遗留系统。

- 对于小型共处一地的团队来说效果很好，但是对于大型分布式开发呢？
- 依赖于团队中开发人员的知识，但如果他们不在（假期、疾病、人员流动）怎么办？



CHIME-- 动机

- 跟踪健康和生活方式数据可以帮助人们管理长期健康期限条件
- 与医疗保健共享这些数据专业人士可以改善临床决策



CHIME-软件

- Chime 是一款专为人们设计的应用程序
感染艾滋病毒

(艾滋病毒感染者)

- 它是由研究人员开发的

英国数量

大学在

与合作

特伦斯·希金斯

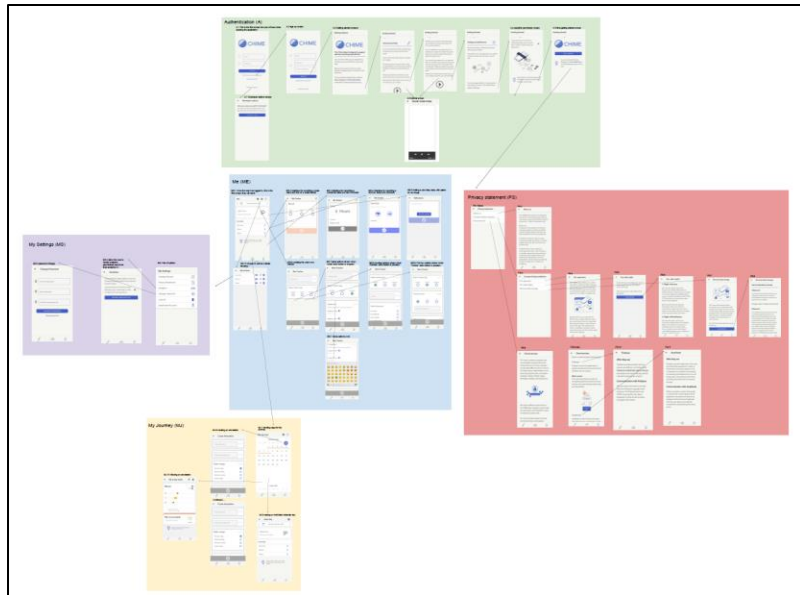
Trust，英国领先的艾滋

病慈善机构



CHIME——软件开发

- 我的项目管理了两名在 UoB 开发应用程序代码的程序员
- 我们最初根据该项目其他研究人员确定的要求设计并构建了一个原型应用程序
- 该应用程序由利益相关者评估，他们确定了更多需求
- 然后，我们进行了一系列四次为期两周的冲刺，并在每次冲刺结束时向其他研究人员展示工作代码
- 该应用程序随后由 PLHIV 进行评估
- 软件开发方式：**敏捷**



阅读

- 讲座中提到的两篇关于评估结对编程的研究论文

[RW Jensen \(2005\) 结对编程经验, 超载, 13\(65\)](#)

[A. Cockburn 和 L. Williams \(2001\) 结对编程的成本和收益。检查了极限编程。G. Succi 和 M. Marchesi, AddisonWesley: 223-243。](#)

- 对敏捷宣言和方法的良好概述 [敏捷方法概述](#)

- 看板方法 <https://kanbanize.com/kanban-resources/getting-started/what-is-kanban>

- 单元测试

[Java 中的单元测试](#)

程序员可以追求的敏捷程度

