项目管理

第五讲

鲁赞娜·奇奇安 /乔恩·伯德 / 皮特·贝内特
助教：Alex Elwood、Alex Cockrean、Casper Wang

（使用 N. Walkinshaw 和 R. Craggs 创建的材料）

概述

- 关于测量
- 白盒测量：
  - 代码行数
  - 圈复杂度
- 黑匣子下的测量：
  - 策划扑克
- 软件法：
  - 专利、版权、合同、隐私

测量是质量的核心

·如何规划项目时间和精力？

·为了团队？

·为了客户？

·哪些软件/部分需要更多时间进行测试？

·哪些开发人员应该获得生产力奖金？ ....

“你不能控制什么你无法测量。”

汤姆·德马科，1982

什么是“测量”？

·为对象赋予价值。

·汽车的燃油效率（加仑/英里）

·足球运动员的进球数

·买房的费用

·可以使用这些值作为比较的基础

·最便宜的房子是多少？

·谁是最佳射手？ ·可以使用这些测量和比较来做出更好的决策。

·我应该买哪辆车（例如,给定五辆候选车）

·我应该在我的球队中安排哪位前锋？

– 很难将其归咎于罪孽

– 可恢复性
– 可靠性合规性
– 可分析性
– 便携性合规性

· 例如，软件是高质量的 记住讲座

软件工程中的测量很困难

图 2.4 IS O 9126

· 大多数实体都是
难以衡量
可靠地
· 很难或不可能“确定”单一值

例如，软件质量
(ISO/IEC 25010)：

· 功能适用性
– 功能齐全-
内斯
– 功能正确性
– 功能适当性

· 性能效率
– 时间行为
– 资源利用
- 容量

· 兼容性
– 共存
– 互操作性

· 可用性
– 适当性
– 可实现性
– L 盈利能力
– 操作性
– 用户错误保护
– 用户界面 A esthet-
集成电路
– 无障碍

· 可靠性
– 成熟度
- 可用性
– 容错
– 可恢复性

· 安全
- 保密
- 正直
– N 不可否认性
- 真实性
– 问责制

· 维护能力
– 模块化
– 可重用性
– 可分析性
– 可修改性
– 可测试性

· 便携性
– 适应性
– 可安装性
– 可替换性

图2.5 ISO/IE C 25010

常用指标:大小和复杂性

·开发后... ·维护需要花费多少精力? ·我们应该在哪里进行测试? ·开发需要付出多少努力? ·指标基于源代码（“白盒”）

·开发开始之前... ·模块X 需要多少编程工作量? ·最终产品的预计成本是多少? ·指标基于要求/规范（“黑匣子”）

白盒复杂性指标

一个文件（或一组文件）中的行数

- 易于计算
- 易于理解和解释
- 通常足以满足

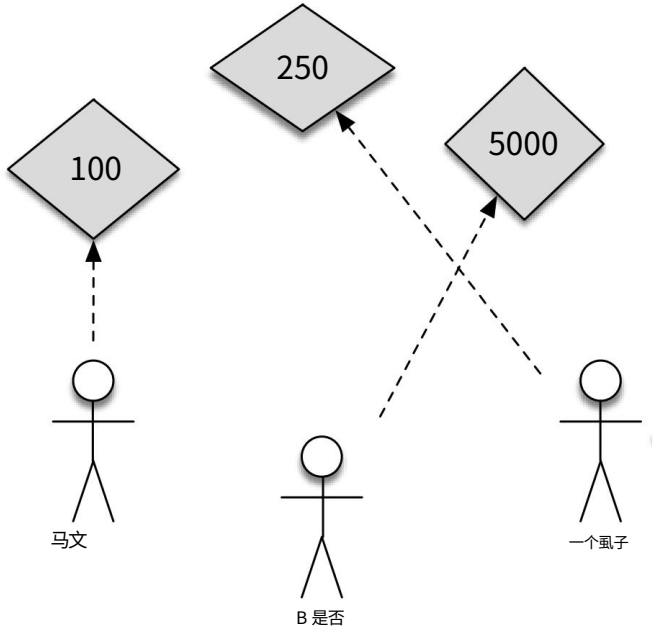
  尺寸的近似测量
- 广泛使用（也许是使用最广泛）的指标
- 评论
- 什么是线？
- 空行
- 并非所有“线”都是平等的
- 忽略逻辑/架构复杂性
- 高度特定于语言

示例:测量程序员专业版

示例:谁是效率最高的程序员?

通过编写的代码行数来衡量

以代码行数来衡量



这是测量吗

如果没有,为什么

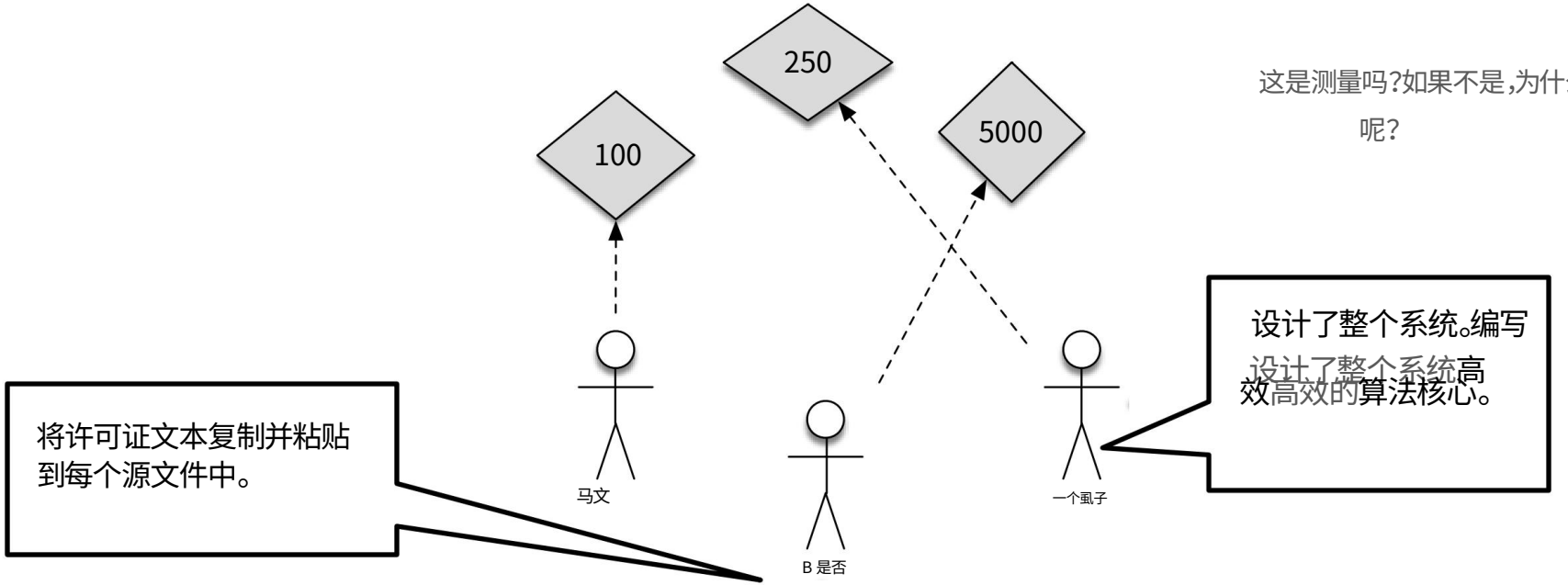
设计整个系统高效算法

示例:测量编程器产品

示例:谁是效率最高的程序员?

- 通过编写的代码行数来衡量

以代码行数来衡量



这是测量吗?如果不是,为什么不呢?

圈复杂度

·根据控制流图计算：

$$V(G) = E - N + 2P$$

E　边数；

N　节点数量；

P – 程序数（通常为 1）

·通过代码的独立路径数

·独立路径　任何引入至少一个新路径的路径

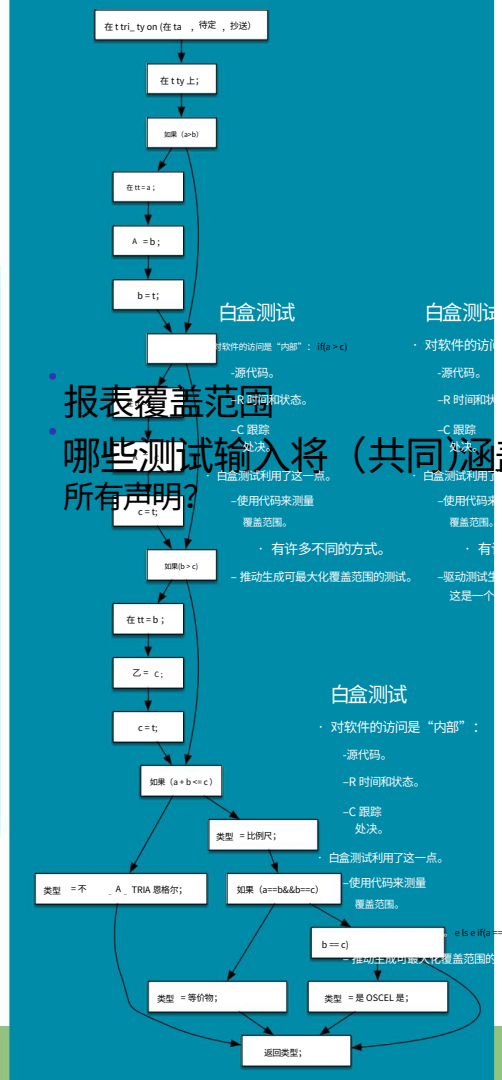
陈述/条件

三角形示例

R的
例子

成就守则 受副视的守则

```
1   int tri_type(int a, int b, int c) {
2       int 类型;
3       如果 (a > b)
4           { intt=a;a=b;b=t; }
5       如果 (a > c)
6           { intt = a;a=c;c=t; }
7       如果 (b > c)
8           {intt=b;b=c;c=t; }
9       如果（a + b <= c）
10          类型= NOT_A_TRIANGLE;
11      别的 {
12          类型=比例尺;
13          如果（a-=b&&b==c）
14  类型 = 等距;
1           否则如果（a == b || b == c）
16  类型 = 等腰;
17          }
18      返回类型;
19      }
```



报表覆盖范围

哪些测试输入将（共同涵盖

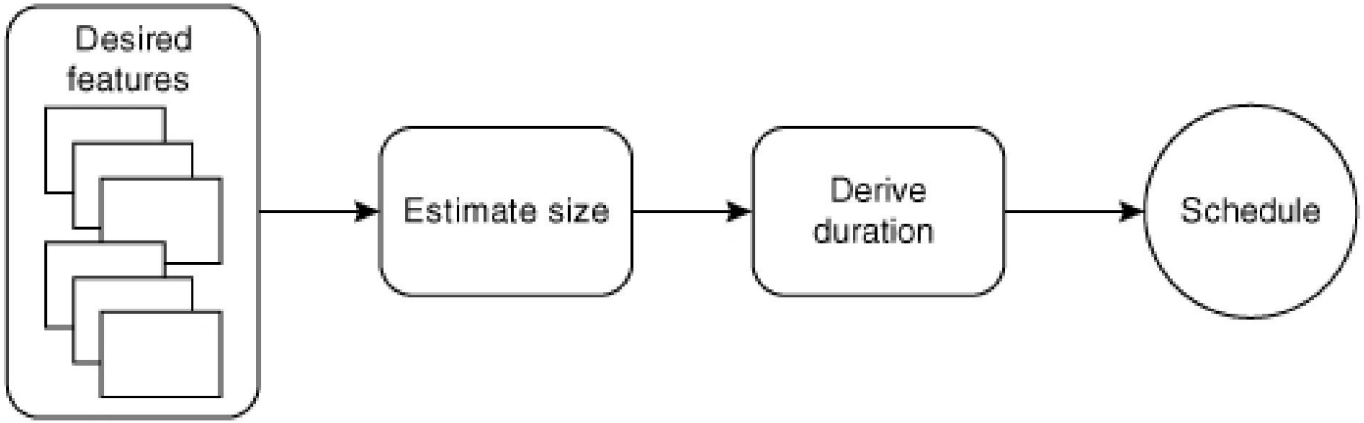
所有声明?

边数 = 27
节点数量 = 22

V = 27 – 22 + 2 = 7

黑盒复杂性指标

评估敏捷项目

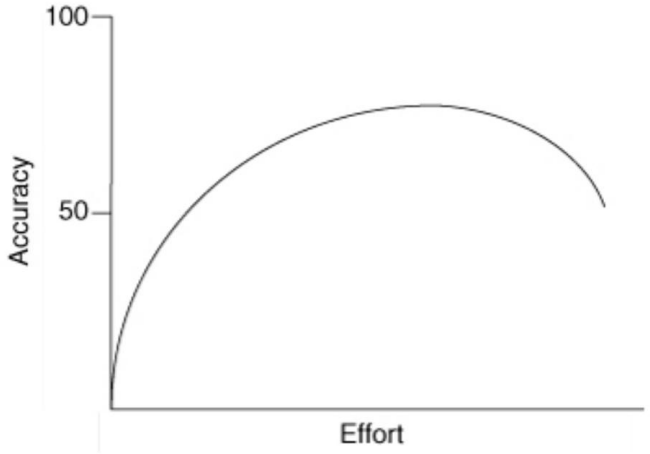


图片来自： Mike Cohn的《敏捷估算和规划》

层数（尺寸估计）

- 一种非正式的、灵活的“尺寸测量”单位
  - 通常估计为 1-10
- 在冲刺计划会议上从整个团队中得出评估结果
- 基于“大众的智慧”的理念
  - 各组的集体估计（即一个故事所需的努力）是比个人的估计更好

项目估算中的准确性与工作量



图片来自： Mike Cohn的《敏捷估算和规划》

规划扑克

·整个团队都参与其中

·每个成员都有一组编号
牌

·数字遵循斐波那契数列

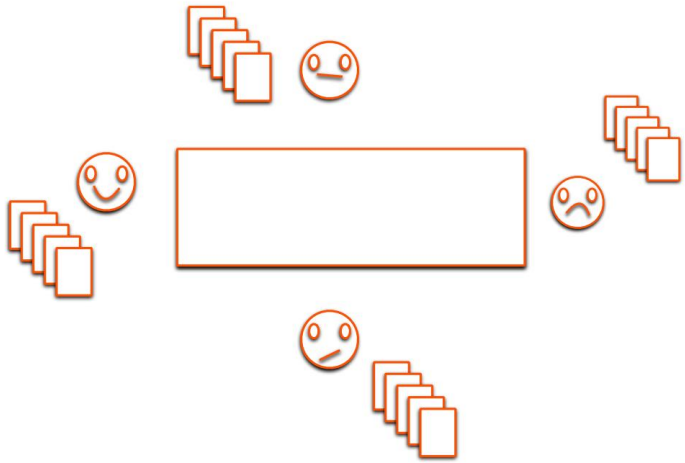
· 1,3,5,8,13,20,... ·较大的
任务变得更难以准确估计

·低值 - 实施起来很简单 ·高值 - 实施起来很困难

·每个成员还被赋予一个“？”卡片

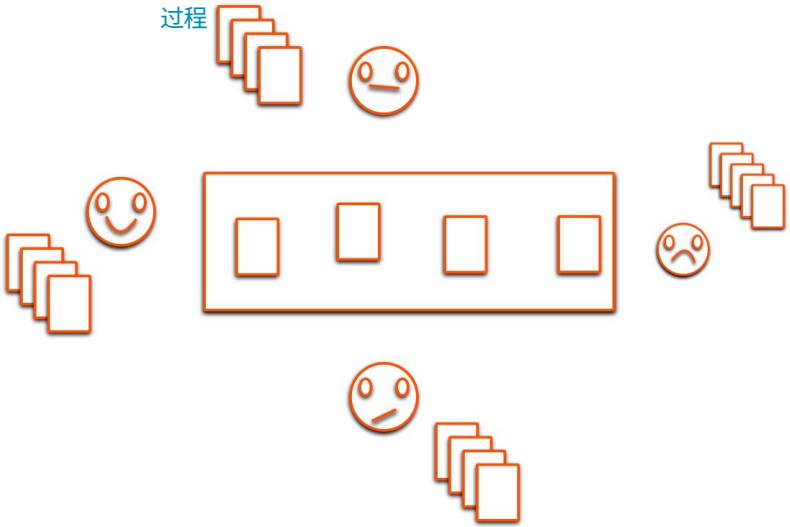
过程

过程

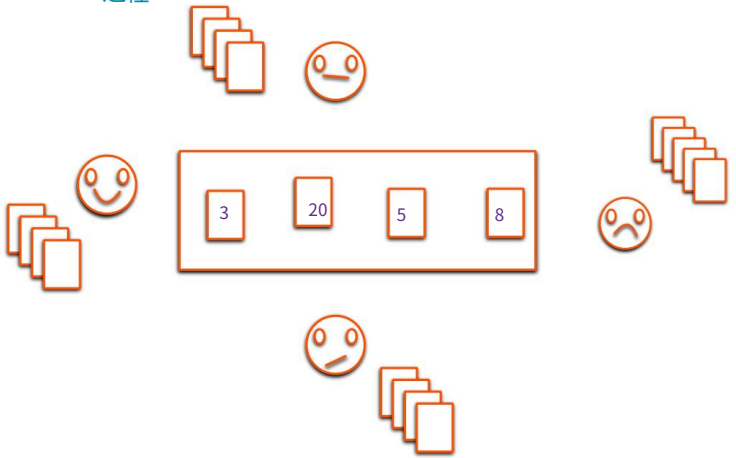


规划扑克:流程

过程

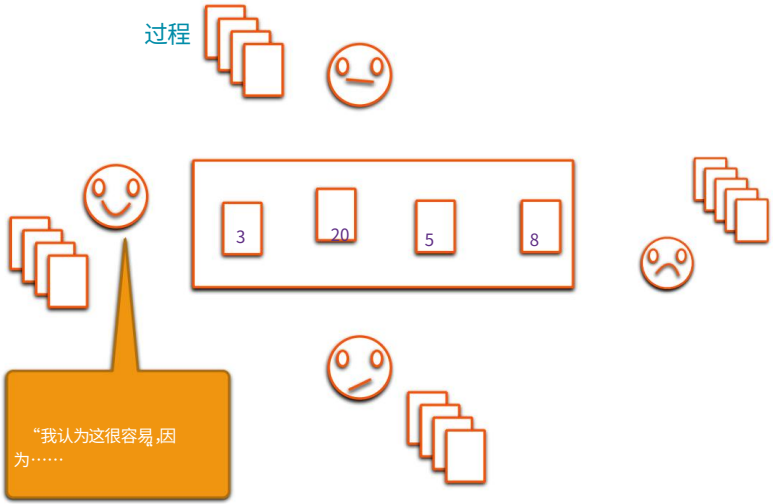


过程



规划扑克:流程

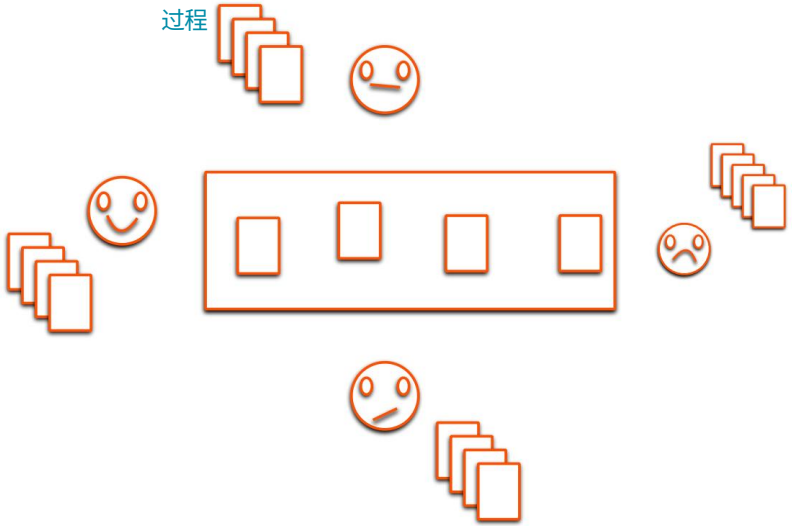
过程



过程

规划扑克:流程

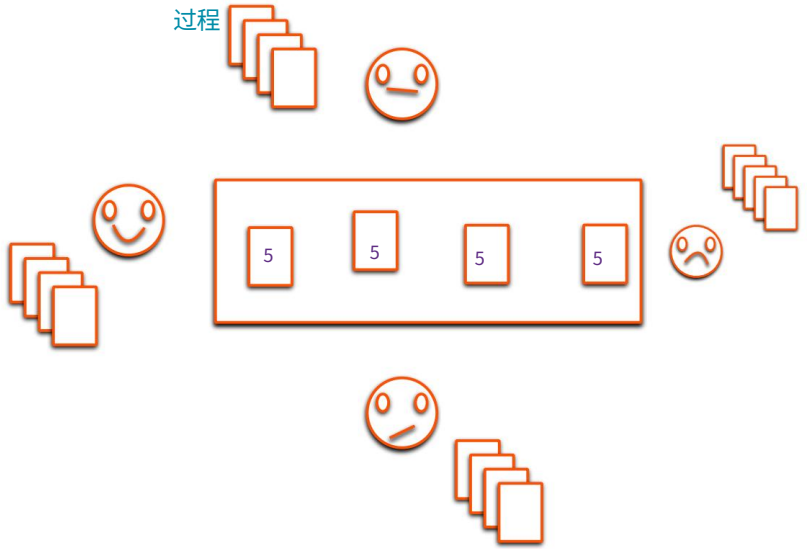
过程



循环最多重复 3 次迭代（以避免无限循环！）

循环重复最多 3 次迭代（以避免无限循环！）

过程



团队速度

·每个冲刺实现的（估计)故事点数量。 ·可以从之前的冲刺中派生出来。

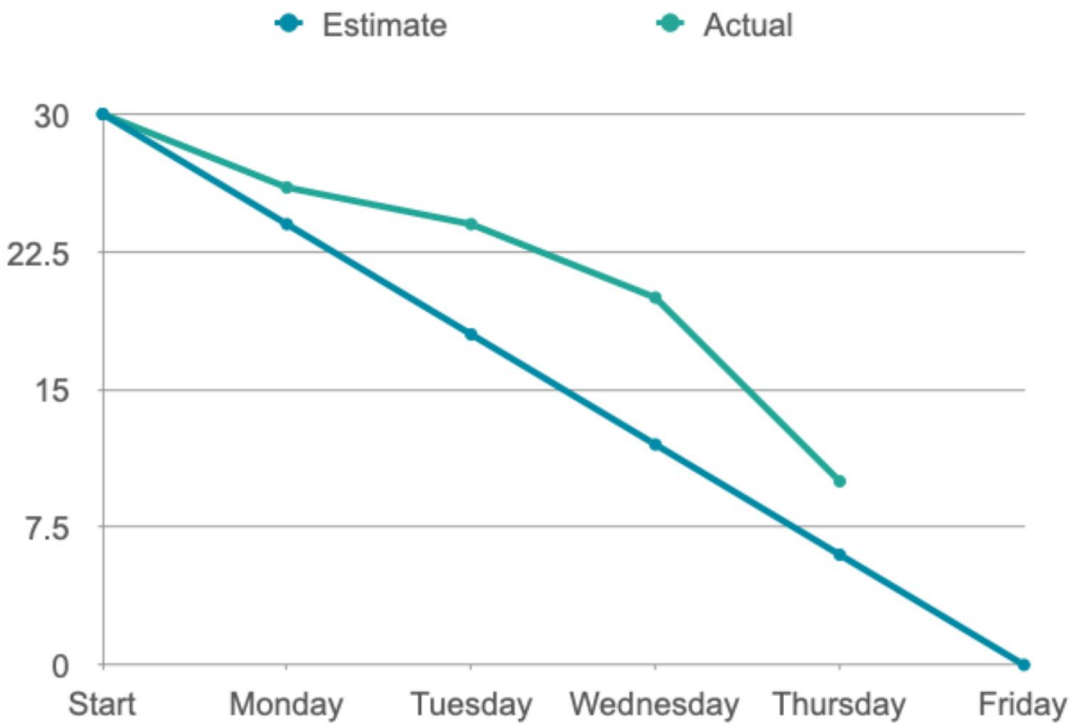
·例如,之前x冲刺实现的平均分。

·可用于估计：

·完成项目所需的时间。

·冲刺中可以完成的目标故事数量。

燃烧图表



软件法:专利、版权、合同、隐私

专利法

政府许可证,授予在一定期限内的权利,特别是排除他人制造、使用或销售发明的权利

·政府授予

·阻止他人利用您的发明

·使用寿命20年

发明必须

·是新的

·具有创造性（不是明显的改进）

·具备工业应用能力

社交网络”

马克·扎克伯格是否侵犯了专利?

·未获得专利

·这个想法并不新鲜,社交网络在此之前就已经存在

版权

·创作者拥有表演、复制、改编其作品的专有权。 ·其他人都必须获得许可（并且可能需要付费） · “文学、戏剧、音乐和艺术作品”包括软件 ·自动拥有（未授予） ·在作者去世后持续70 年（很多例外）

这会以两种不同的方式影响软件： ·应用程序的非法副本（盗版）！ ·使用别人的代码/UI 设计/等。在您的应用程序中（不是“想法”，而是其他人创建的实际“东西”（代码、设计、文档））

版权盗窃？

不：

- 获得许可（获得许可证）
- 处于“合理使用”范围内（例如用于研究或审查）
- 使用“开源”软件
- 独立地自己创造类似的东西

- 不能使用“显而易见”的代码文案撰写的

是的：

- 显示另一张图像页
- 使用在互联网上找到的代码
- 为您的朋友复制Windows 95

社交网络”

马克·扎克伯格是否侵犯了版权？

也许 ·但
没有证据表明他复制了 ·这不属于合理使用

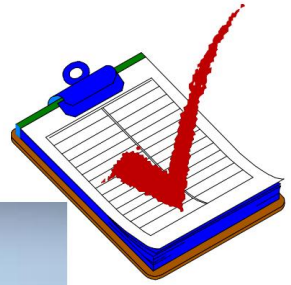
·这不是 OSS ·他看到了代码，所以不是自己发明的

合同法

雇主合同通常迫使雇员：

- 不为其他人工作
- 移交任何想法（知识产权）
- 不泄露公司机密（保密协议） （即使在您停止为他们工作之后）

社交网络”



马克·扎克伯格违约了吗？

可能不会

·没有书面记录

合同

·他没有透露其他项目的任何秘密

数据保护

·英国:数据
保护法
·欧盟:日期
保护
指示
·美国:a
国家的“拼凑”和
国家法律

数据保护 8 项原则:

任何存储“个人数据”的公司都必须确保其: ·公平、合法地处理(同意、合同和法律义务、公共利益……)

·出于有限目的进行处理;
·充分、相关且不过分;
·准确,并在必要时保持最新; ·保存时间不得超过必要的时间;

·根据数据主体的权利进行处理;
·安全;
不转移到没有充分保护的国家

审查

·我们如何衡量复杂性？ ·为什么我们使用黑盒选项？ ·什么是专利 ·专利和版权有什么区别？ ·我们从社交网络中了解到契约的哪些内容？