Lecture 5 Project Management

白盒测试(line of codes & Cyclomatic Complexity(圈度复杂数))
黑盒测试(Planning Poker（规划扑克))
软件法律(Patents, Copyright, Contract, Privacy(专利，版权，合同，隐私)

Measurement is Central to Quality - 质量是衡量的核心

You cannot **control** what you cannot measure." - Tom DeMarco, 1982

What is "Measurement"? (怎么衡量？)

- Attributing values to objects.归因值
- Can use these values as basis for comparison比较基础
-Can use these measurements and comparisons to **make better decisions**. 更好决策

Measurement is Difficult in Software Engineering - 很难衡量(Reason)

- Most entities are difficult to measure reliably实体难衡量
- Difficult or impossible to "pin down" a single value不可能单一值

Usual Metrics: Size and Complexity(大小 & 复杂度)

Before development – effort(需求工作量)，cost(预算)，

Metrics are **based upon** requirements / specification ("black box") – 开发前：黑盒测试

After development – effort maintenance(维护)，test(测试重点)， effort was required for development(后续开发)

Metrics are **based upon** source code ("white box") – 开发后： 白盒测试

Measurement is Difficult in Software Engineering

- Most entities are difficult to measure reliably
- Difficult or impossible to “pin down” a single value

E.g., Software Quality (ISO/IEC 25010):

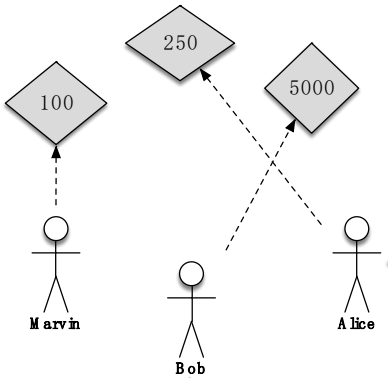
- Functional Suitability
  - Functional Completeness
  - Functional Correctness
  - Functional Appropriateness
- Performance Efficiency
  - Time Behaviour
  - Resource Utilisation
  - Capacity
- Compatibility
  - Co-existence
  - Interoperability
- Usability
  - Appropriateness
  - Realisability
  - Learnability
  - Operability
  - User Error Protection
  - User Interface Aesthetics
  - Accessibility
- Reliability
  - Maturity
  - Availability
  - Fault Tolerance
  - Recoverability
- Security
  - Confidentiality
  - Integrity
  - Non-repudiation
  - Authenticity
  - Accountability
- Maintainability
  - Modularity
  - Reusability
  - Analysability
  - Modifiability
  - Testability
- Portability
  - Adaptability
  - Installability
  - Replaceability

White Box Complexity Metrics – 白盒复杂度衡量标准

Number of lines in a file (or a group of files)文件行数

- Easy to compute计算
- Easy to understand and interpret 理解执行
- Often sufficient for an approximate measure of size 足以对大小测量
- Widely used (perhaps the most widely used) metric 广泛使用
- Comments
- What is a line?
- Blank lines
- Not all “lines” are equal
- Ignores logical/ architectural complexity
- Highly language-specific

Example: Who is the most productive programmer?



Bob: Copied and pasted license text into every source file.
Alice: Designed entire system. Wrote **highly efficient algorithmic** core.
Alice wined!

Cyclomatic Complexity(圈度复杂数)

- Calculated from the **control flow graph**:

  V(G) = E – N + 2P 这玩意儿就是算所谓的独立路径的

  换句话说这个程序可以通过几种不一样的方式实现?

  E – number of edges; 线的数量?

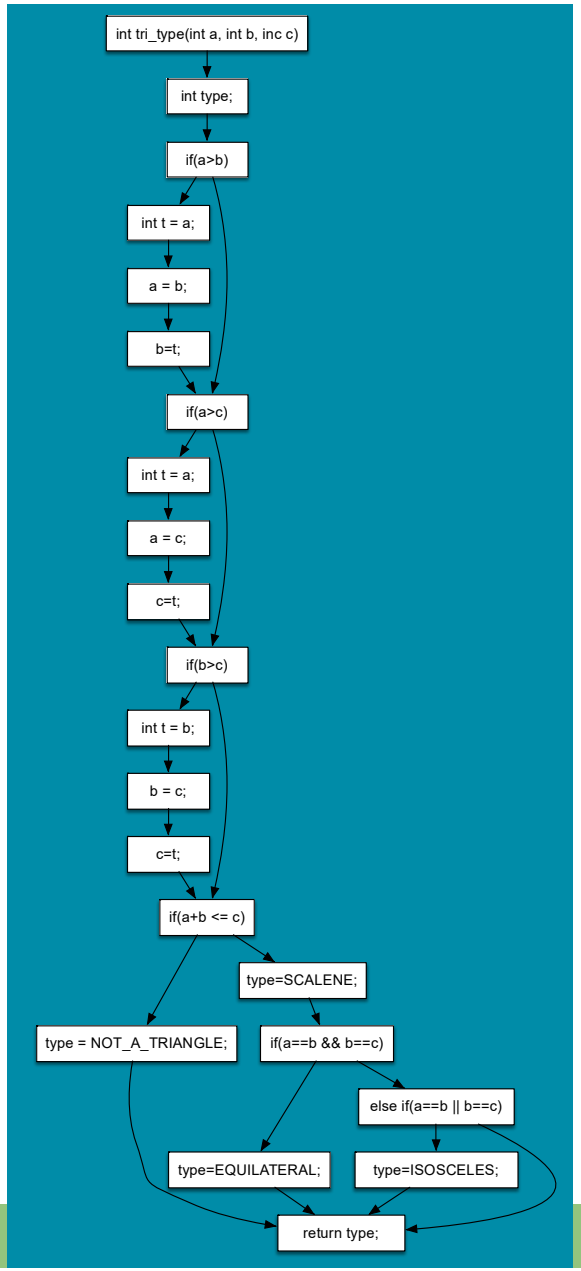
  N – number of nodes; 节点/基本块

  P – number of procedures (usually 1) 一个进入一个出去，一般是2

- Number of independent paths through the code 计算所有的独立路径
- Independent path – any path that introduces at least one new statement/condition (引入至少一个新的条件/代码块? -> 一个新的独立路径)

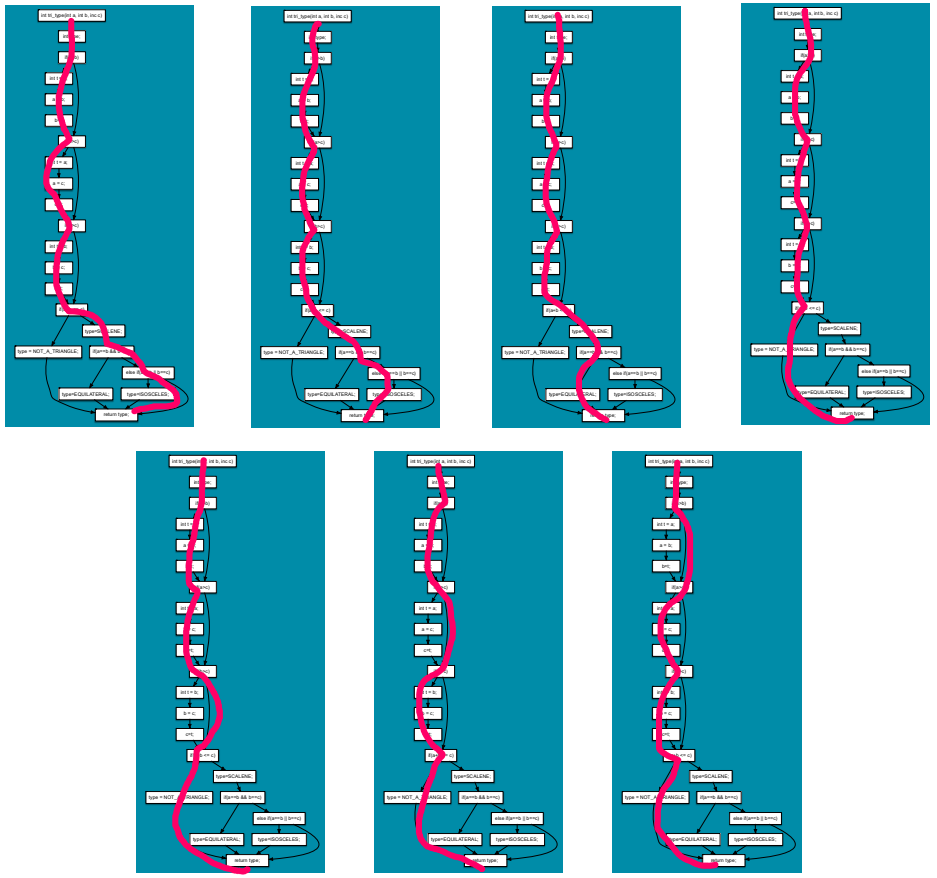
Triangle Example

```
1    int tri_type(int a, int b, int c) {
2      int type;
3      if (a > b)
4        { int t = a; a = b; b = t; }
5      if (a > c)
6        { int t = a; a = c; c = t; }
7      if (b > c)
8        { int t = b; b = c; c = t; }
9      if (a + b <= c)
10       type = NOT_A_TRIANGLE;
11     else {
12       type = SCALENE;
13       if (a == b && b == c)
14         type = EQUILATERAL;
15       else if (a == b || b == c)
16         type = ISOSCELES;
17     }
18   return type;
19   }
```



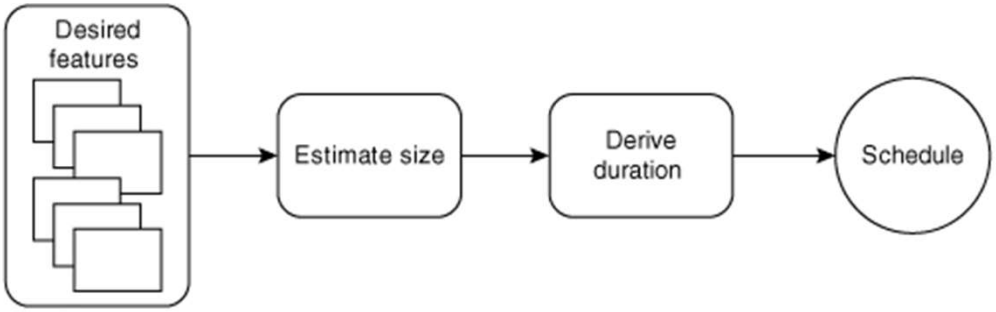
Number of Edges = 27
Number of Nodes = 22

V = 27 – 22 + 2 = 7

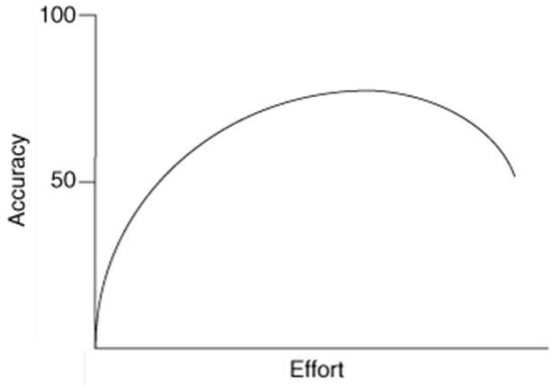


Black Box Complexity Metrics – 黑盒复杂度衡量标准

评估Agile项目开发流程：



Accuracy vs Effort in Project Estimation准确度和估算



Story Points – 评估用户故事(user stories)复杂度(大小评估？)
An informal, agile unit of “size measurement”
Usually an estimate from 1-10

Derive an estimate from the whole team at sprint planning meetings 团队合作推导出估算值

Based on the idea of the “Wisdom of the Crowds”
The collective estimate of groups (i.e., of effort required for a story) is better than the estimate of an individual）众人之智

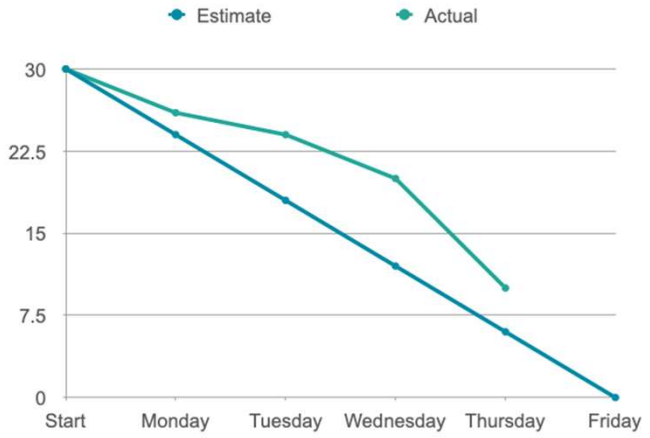
Planning Poker(规划扑克)

1.The **whole team** is involved
2. Each member is given a set of numbered cards
  (Numbers follow the **Fibonacci sequence** 1,3,5,8,13,20,...)
Larger tasks become harder to estimate in exact terms
Low values - trivial to implement 低 -> 简单
High values - difficult to implement 高 -> 难
Each member is also given a “?” card 有张？卡片
打牌，出小的觉得简单+原因解释；出大的觉得难+原因解释
最大只能打三轮（避免无限循环）

Team Velocity（团队速度—冲刺爽）

- Number of (estimated) story points implemented per sprint.
- Can be derived from previous sprints.(从以前的冲刺推导)
e.g., Average points implemented from previous x sprints.
  (以前X个冲刺的平均点数)
- Can be used to estimate:
  - Time required to complete project. 所需总时间
  - Target number of stories that can be completed in a sprint.
    (在一个冲刺中可以完成的目标故事数量。)

Burn Charts



Software Laws: Patents, Copyright, Contract, Privacy（专利，版权，合同，隐私）

Patent Law（专利法）

A government **license** giving a right for a set period, especially to **exclude others** from **making, using, or selling an invention**

- Granted by **the government**
- to stop others **exploiting your invention**
- Lasts **20** Year ｜Inventions Must
- be **new 新**
- be an **inventive step** (not an obvious improvement) 创造性
- capable of **industrial application 工业应用**

Copyright（版权）

Creator has exclusive rights to perform, copy, adapt their work.
Everyone else must get Permission (and possibly pay)
"literary, dramatic, musical and artistic works" includes software
Automatically owned (not granted) 自动获得
Lasts 70 years after authors death (lots of exceptions) 死后70年
This **affects** software in **2** different ways:
1.Illegal Copies of Applications (Piracy) !
2.Using someone else's **code/UI design/etc**. in your application (Not the "idea" but the actual **"stuff"** (code, design, documents) created by someone else)

Did **Mark Zuckerberg** infringe a patent?

- No patent was granted
- The idea was not new, social networks existed before this

没有侵权。

Copyright Theft?

No:
Get permission (obtain a licence)
Be within "fair use" (e.g. for study or review)
Use "open source" software
Create something similar yourself, independently
"Obvious" code can't be copywrited

Yes:
Displaying an image from another page
Using code found on the internet
Copying Windows 95 for your friends

Contract Law合同法

Employer contracts usually **force** an employee to:
1. Not work for anyone else
2. Hand over any ideas (Intellectual Property)
3. Not disclose company secrets (**Non-disclosure-agreements保密协议**) (even after you stop working for them)

Data Protection 数据保护

8 Principles of Data Protection:

Any company storing "personal data" must make sure it is:
1. fairly and lawfully processed (consent, contractual and legal obligations, public interest, ...) 公平合法
2. processed for **limited purposes**;为限定目的
3. adequate, relevant and **not excessive**;**适当、相关且不过度**
4. **accurate** and, where necessary, kept up to **date**; 准确&及时
5.not kept longer than necessary;不超过必要期限
6. processed in accordance with the **data subject's rights**; **数据主体的权利进行处理**
7. secure;安全性
8.**not transferred to countries without adequate protection 不转移到保护不足的国家**

Did Mark Zuckerberg infringe copyright?
Maybe
- but there is **no evidence** he copied
- it it's **not fair use**
- it wasn‘t OSS 非开源软件
- he saw the code **so didn't invent** it himself

Did Mark Zuckerberg **break contract**?
Probably Not
- there was **no written contract**
- he did **not disclose any secrets** about the other project

不同地区如何数据保护：
UK : Data Protection Act
EU : Data Protection Directive
US : a "patchwork" of state and national laws

Review

1.How can we **measure complexity**?

Lines of files(代码行数) and Cyclomatic Complexity (圈度复杂数). (V(G)=E-N+2P)

2. Why do we use **black box options**?

1.Simplicity2.Time-saving3.Cost-effectiveness4.Risk mitigation5.Focus on core competencies

总的来说，黑匣子选项为用户提供了一种便捷、高效且具有成本效益的方式，可以访问复杂的技术或功能，而无需大量的技术专业知识或资源。

3.What is a **patent** – 定义考察

A government **license** giving a right for a set period, especially to **exclude others** from **making, using, or selling an invention.**

4.What is the **difference** between patent and copyright?

1.**保护对象**：专利保护的是**发明性、创新性的实用性**发明或设计，如新的产品、工艺或方法;版权保护的是创作的**原创性**作品，包括文学作品、音乐作品、艺术作品、软件代码等。

2.**保护范围**：专利保护给予专利持有人在一定时间内对其发明或设计的独占权，防止他人未经许可制造、使用、销售或引入同类产品;版权保护给予版权持有人对其作品的一系列权利，包括复制、发行、展示、表演、修改和衍生等权利。

3.**申请方式**：专利需要向**专利局**提交专利申请，并经过审查、授予和**登记等程序后**才能获得保护;版权通常在创作作品时即**自动产生**，无需特别申请或登记，但在一些国家，如美国，可以通过注册版权来获得额外的保护和便利。

4.**保护期限**：专利20年左右;版权的保护期限较长，一般为**作者生命加70年**，在一些国家有所不同。

5.**保护要求**：专利需要满足**新颖性、非显性、工业应用性**等法定要求;版权通常自作品创作完成时即**自动产生，无需满足特定**的法定要求