



# 软件工程

## 个人技术（1）

Spring 2022, SWUFE

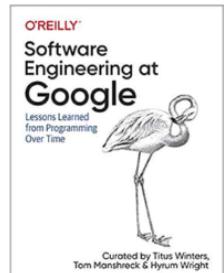
---

## 复习

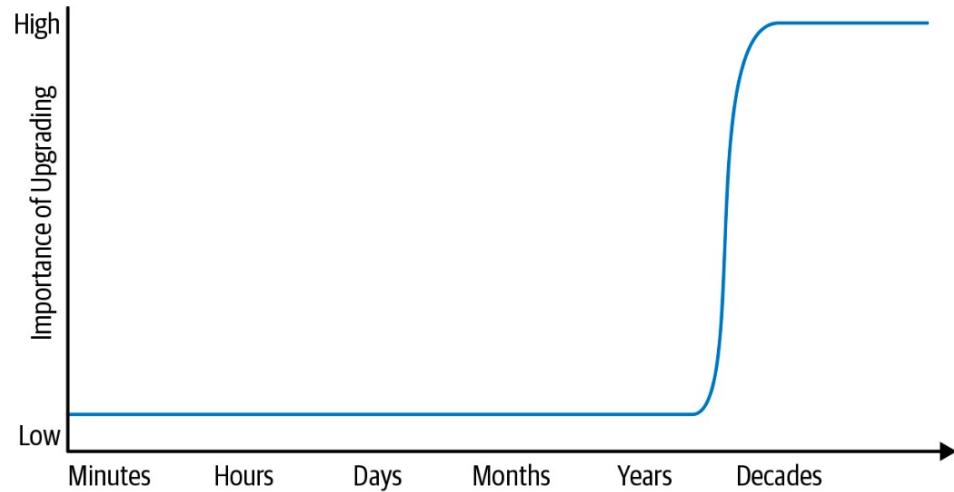
- 软件 = 程序 + 软件工程
- 软件工程的目标：（在预期的时间内）创造“**足够好**”的软件
- 没有银弹

# 谷歌视角下的软件工程

## Software Engineering at Google



作者: Titus Winters / Tom Mansreck / Hyrum K. Wright  
出版社: O'Reilly Media  
副标题: Lessons Learned from Programming Over Time  
出版年: 2020-3-3  
页数: 500  
定价: USD 59.99  
装帧: Paperback  
ISBN: 9781492082798



Software engineering is programming integrated over time.

---

## 个人作业

⚠ 考虑到软件工程的特征，项目需求会反复修改，所以最后需要提交多个版本的交付物。

2022-02-21版本：苹果生态的 AirDrop 在传输文件时非常方便，因此某同学计划模仿实现该功能。目前的方案如下：在桌面端编写一个软件或者网站充当服务器，当手机和它处于同一个局域网时，通过网页的方式发送文件。

---

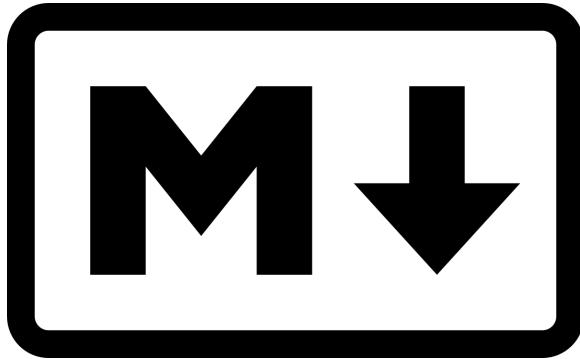
## 前言



组团队开发软件之前，首先要确保团队成员是合格的软件工程师，因此我们需要先完善个人技术，形成**个人软件开发流程**（Personal Software Process, PSP）。

---

## 1. Markdown



Markdown是一种简洁的标记语言，流行于写博客和代码文档等场景。



# VSCode

week1 > ① README.md > **本周作业** > **题目二 (10分)**

1 本周主要学习了现代软件工程的含义。

2

3 **# 本周作业**

4 **## 题目一 (10分)**

5 使用你不熟悉的语言 (Scala, Rust, C#, Kotlin) 写一个简单的“Hello [你的名字]”程序。要求写出环境搭建的过程。

6

7 **## 题目二 (10分)**

8 托尼在20世纪60年代设计Algol W语言的时候引入了 Null Reference，对后面的编程影响很大，他称这是一个“10亿美元的错误” ([Billion Dollar Mistake] (<https://www.infoq.com/presentations/Null-References-The-Billion-Dollar-Mistake-Tony-Hoare/>)）。请结合代码简述上面你使用的语言采用了什么更好的办法处理Null。

Preview README.md ×

本周主要学习了现代软件工程的含义。

本周作业

### 题目一（10分）

使用你不熟悉的语言（Scala, Rust, C#, Kotlin）写个简单的“Hello [你的名字]”程序。要求写出环境搭建的过程。

### 题目二（10分）

托尼在20世纪60年代设计Algol W语言的时候引入了Null Reference，对后面的编程影响很大，他称这是一个“10亿美元的错误”（[Billion Dollar Mistake](#)）。请结合代码简述上面你使用的语言采用了什么更好的办法处理Null。



我们以标准的Markdown语法为主，  
并兼顾GitHub的MD语法

README.md

questions for week 1

1 hour ago

README.md



本周主要学习了现代软件工程的含义。

## 本周作业

### 题目一（10分）

使用你不熟悉的语言（Scala, Rust, C#, Kotlin）写个简单的“Hello [你的名字]”程序。要求写出环境搭建的过程。

### 题目二（10分）

托尼在20世纪60年代设计Algol W语言的时候引入了Null Reference，对后面的编程影响很大，他称这是一个“10亿美元的错误”（Billion Dollar Mistake）。请结合代码简述上面你使用的语言采用了什么更好的办法处理Null。



## 采用MD的网站

**Title**  
Be specific and imagine you're asking a question to another person

---

e.g. Is there an R function for finding the index of an element in a vector?

---

**Body**  
Include all the information someone would need to answer your question

---

  
[Links](#) [Images](#) [Styling/Headers](#) [Lists](#) [Blockquotes](#) [Code](#) [HTML](#) [Tables](#) [More](#)

[Hide formatting tips](#)

---

# 采用MD的网站



week0\_陈中普\_20191001

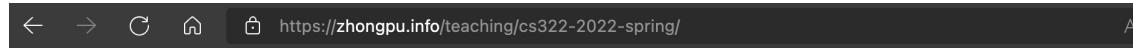
## 题目一

Java ▾

Wrap code

```
1 public class Main {  
2     public static void main(String[] args) {  
3         System.out.println("Hello 陈中普");  
4     }  
5 }
```

# 采用MD的网站



Zhongpu

Publications    Teaching



Zhongpu Chen

PhD

## Spring 2022 (Software Engineering)

### Lecture Location and Time

From Week 1 to 17, every Monday (Period 1 to 3).



- Room H304, Yide Building (颐德楼)
- Room I101 (for Lab)

### Syllabus

---

## 1.1 斜体， 加粗



I am an *italic*, and I am a ****Bold****.

I am also a *italic*.

I am also a **bold**.

建议：斜体使用`*text*`，  
加粗使用`**text**`。



## 1.2 代码



The keyword `pass` in Python means *\*doing nothing\**.

```
```python
def foo():
    pass
```
```

单个变量使用`code`。多行  
代码使用 `'''`

Code

'''

---

## 1.3 引用



Richard Feynman said,

> Even if you know all those names for it, you still  
know nothing about the bird.

引用使用 >

## 1.4 标题

一级标题使用#，  
二级标题使用##，  
以此类推。

```
# 这是MarkDown的例子

## 斜体，粗体
I am an _italic_, and I am a **Bold**.

I am also a *italic*.

I am also a __bold__.

## 代码

The keyword `pass` in Python means *doing nothing*.

```python
def foo():
    pass
```

## 引用

Richard Feynman said,
> Even if you know all those names for it, you still know nothing about the bird.
```

---

## 1.5 无序列表



I have used several operating systems, including

- MacOS
- Windows
- Ubuntu
- Manjaro

无序列表使用 \* 或  
+ 或 -

---

## 1.6 有序列表



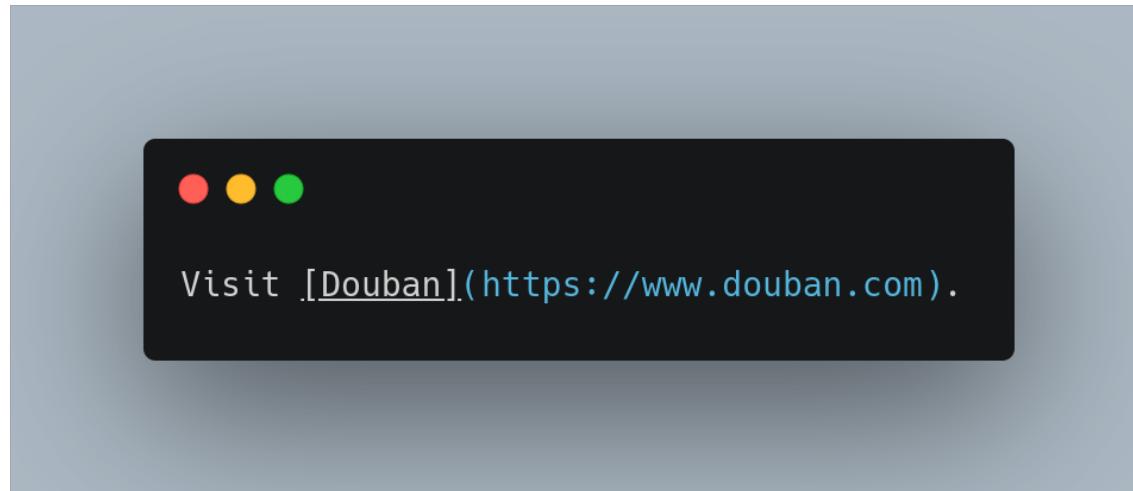
I can speak several languages, including

1. Mandarin
2. English
3. Japanese

有序列表使用 [1.](#)，以此类推

---

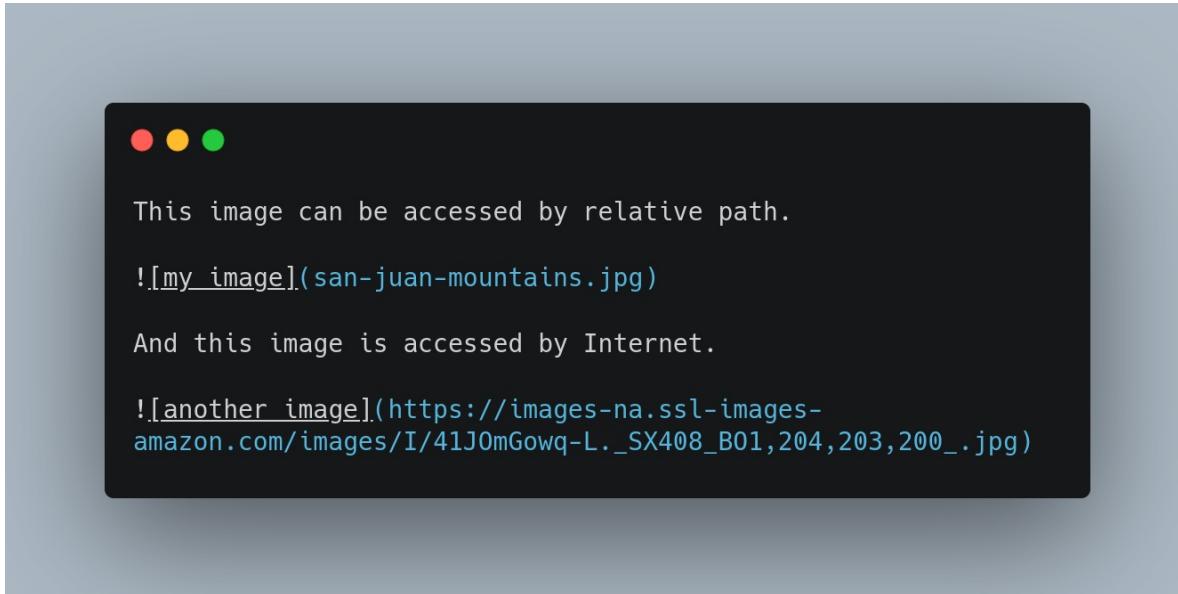
## 1.7 超链接



超链接使用 `[text](link_address)`

---

## 1.8 图片



图片使用 `![text](image_path)`

---

## 1.9 其他扩展

- 表格
- Emoj
- 字体颜色
- ...

|                         |                |                   |
|-------------------------|----------------|-------------------|
| 🎀 :bowtie:              | 😊 :smile:      | 😂 :laughing:      |
| 😳 :blush:               | 😃 :smiley:     | 😌 :relaxed:       |
| 😏 :smirk:               | 😍 :heart_eyes: | 😘 :kissing_heart: |
| 😚 :kissing_closed_eyes: | 😳 :flushed:    | 😌 :relieved:      |

<http://pan.swufe.edu.cn:80/link/D91A0D7F401F89B05C59E3FB4BBE2F1E>

---

## 课堂任务（**10分**）

下载 VSCode，从下面任意一个熟悉的话题，使用Markdown（至少用7类不同元素）图文并茂地介绍：

- 二分搜索算法
- HTTP
- 关系数据库的数据类型
- SQL语句
- Java的泛型
- C++ STL的迭代器
- Python的虚拟环境

---

## 2. Maven/Gradle

Java的项目构建工具





## 依赖管理

- 在开发Java EE项目时，你的程序需要连接MySQL或者PostgreSQL。你应该怎么办？



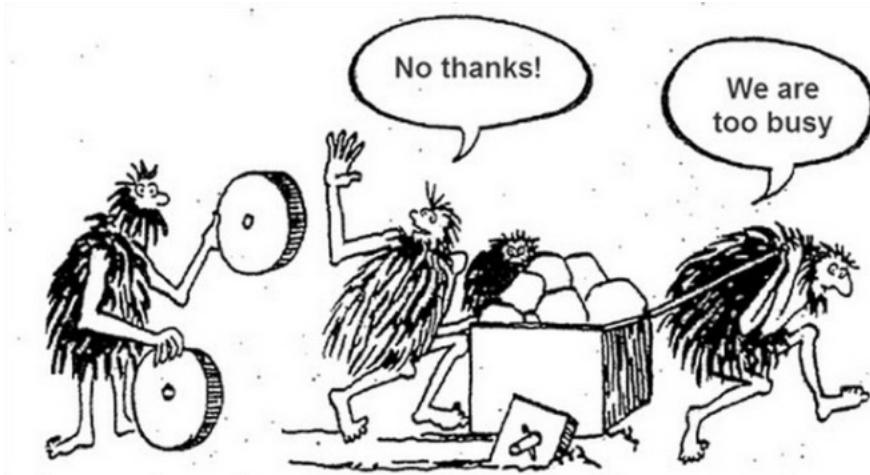
```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.28</version>
</dependency>
```

implementation 'mysql:mysql-connector-java:8.0.28'

---

## 依赖管理

Never reinvent the wheel!

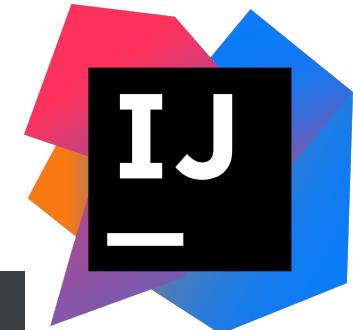


# Java的标准工程结构

```
my-app
|--- pom.xml
`--- src
    |--- main
    |   `--- java
    |       `--- com
    |           `--- mycompany
    |               `--- app
    |                   `--- App.java
    `--- test
        `--- java
            `--- com
                `--- mycompany
                    `--- app
                        `--- AppTest.java
```

pom.xml是我们学习的重点：  
Project Object Model

# 例子



IntelliJ IDEA

New Project

Project SDK: azul-11 Azul Zulu version 11.0.13

Create from archetype

New Project

Name: maven-test

Location: ~/github/swufe-se/week2/maven-test

**Artifact Coordinates**

GroupId: org.swufe.se

The name of the artifact group, usually a company domain

ArtifactId: maven-test

The name of the artifact within the group, usually a project name

Version: 1.0-SNAPSHOT

Java

Maven

Gradle

Android

IntelliJ Platform Plugin

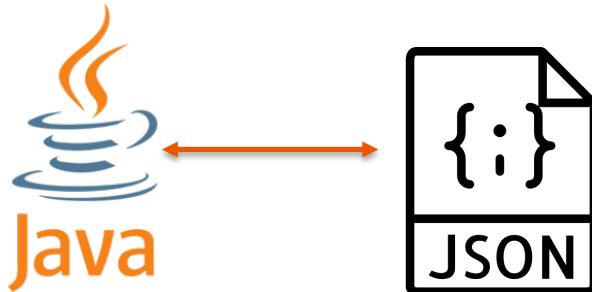
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4           xsi:schemaLocation="http://maven.apache.org/POM/4
5 <modelVersion>4.0.0</modelVersion>
6
7 <groupId>org.swufe.se</groupId>
8 <artifactId>maven-test</artifactId>
9 <version>1.0-SNAPSHOT</version>
10
11 <properties>
12     <maven.compiler.source>11</maven.compiler.source>
13     <maven.compiler.target>11</maven.compiler.target>
14 </properties>
15
16 </project>
```

简单的pom.xml

## 使用JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write.

```
{name:"John", age:31, city:"New York"}
```



# Gson

← → ⌂ ⌂ https://mvnrepository.com/search?q=Gson

## MVNREPOSITORY

Gson

Repository

- Central 377
- JCenter 101
- Sonatype 89

**Gson » 2.9.0**

Gson

Found 527 results

Sort: [relevance](#) | [popular](#) | [newest](#)

License	Apache 2.0
Categories	JSON Libraries
Date	(Feb 11, 2022)
Files	<a href="#">pom</a> (7 KB) <a href="#">jar</a> (243 KB) <a href="#">View All</a>
Repositories	Central
Used By	<a href="#">17,170 artifacts</a>

[Maven](#) [Gradle](#) [Gradle \(Short\)](#) [Gradle \(Kotlin\)](#) [SBT](#) [Ivy](#) [Grape](#) [Leiningen](#)

```
<!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
<dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.9.0</version>
</dependency>
```

# Maven设置镜像地址

<https://developer.aliyun.com/mvn/guide>

```
settings.xml ×
Users > zhongpu > .m2 > settings.xml > settings
1  <settings>
2      <mirrors>
3          <mirror>
4              <id>aliyunmaven</id>
5              <mirrorOf>*</mirrorOf>
6              <name>ali-public</name>
7              <url>https://maven.aliyun.com/repository/public</url>
8          </mirror>
9      </mirrors>
10 </settings>
```

## 添加依赖

```
<dependencies>
    <!-- https://mvnrepository.com/artifact/com. -->
    <dependency>
        <groupId>com.google.code.gson</groupId>
        <artifactId>gson</artifactId>
        <version>2.9.0</version>
    </dependency>
</dependencies>
```



## 使用依赖

```
public class Main {  
    public static void main(String[] args) {  
        Person person = new Person("Bob", 18, "Chengdu");  
        Gson gson = new Gson();  
        System.out.println(gson.toJson(person));  
    }  
}
```

思考：查阅Java EE中**Servlet**的依赖。  
为什么它有个**scope**，含义是什么？

---

---

## 课堂练习

Java中的Set不能有重复元素，但是有时候我们需要“重复且无序”（即数据结构中的Bag）。**Google Guava**扩展了Java的标准库，其中提供了MultiSet能够满足上述需求。

使用Maven新建一个工程，添加Guava作为依赖，并编写简单的测试代码。参考<https://www.baeldung.com/guava-multiset>

---

### 3. 单元测试

绝大多数软件都是由多人完成的，大家的工作相互有依赖关系。那就会出现一个人写的**模块**被另一个人写的**模块**调用的情况。很多误解、疏忽都发生在两个**模块**之间。

如何能让自己写的模块尽量无懈可击？

单元测试（Unit Test）就是一个很有效的解决方案。

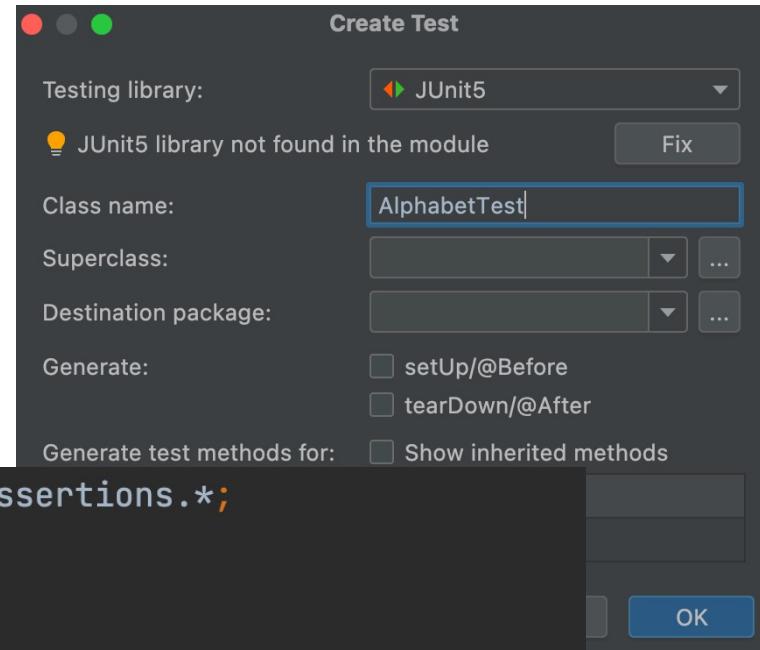
Testing



## 一个例子

```
public class Alphabet {  
    private static boolean isVowel(int c) {  
        return c == 'a'  
            || c == 'e'  
            || c == 'i'  
            || c == 'o'  
            || c == 'u';  
    }  
  
    public static int countVowels(String s) {  
        return (int) s.toLowerCase().chars()  
            .filter(Alphabet::isVowel).count();  
    }  
}
```

# 新建测试



```
1 import static org.junit.jupiter.api.Assertions.*;
2
3 class AlphabetTest {
4
5     @org.junit.jupiter.api.Test
6     void countVowels() {
7         assertEquals(Alphabet.countVowels(s: "apple"), actual: 2);
8     }
9 }
```

## 测试覆盖率

The screenshot shows an IDE interface with a dark theme. At the top, there's a toolbar with various icons. Below it, a navigation bar displays 'AlphabetTest.countVowels' and other tabs. The main area has a title '(unit-test) ×'. On the left, there's a code editor window showing Java code. On the right, a coverage report is displayed with the title 'Coverage: AlphabetTest.countVowels'. The report states '100% classes, 75% lines covered in 'all classes in scope''. It includes a table with columns: Element, Class, %, Method, %, and Line, %. The table lists several packages and classes, with 'Alphabet' at the bottom having 100% class coverage, 66% method coverage, and 75% line coverage.

Element	Class, %	Method, %	Line, %
apple			
com			
images			
java			
javax			
jdk			
META-INF			
netscape			
org			
sun			
toolbarButtonGraphics			
Alphabet	100% (1/1)	66% (2/3)	75% (3/4)

```
public static int countConstants(String s) {  
    return s.length() - Alphabet.countVowels(s);  
}
```

对于真正的「软件」，单元测试  
不可或缺。

尽量使得代码覆盖率达到100%。

---



# 总结

- 文档 : Markdown
- 构建 : Maven
- 测试 : 单元测试