软件工程问题集合一览（2022年）

4/25号

1. 什么是软件。
2. 软件可以有哪些分类？
3. 软件的特点有哪些？本质特性有哪几个？
4. 软件危机的现象是什么，为什么会发生软件危机。软件危机现在还存在吗？

4/27号

1. 解决软件危机的途径是什么？
2. 软件工程包含的内容有哪些？
3. 软件工程的基本原理有哪些，请枚举若干。
4. 如何评价软件质量？
5. 瀑布模型包含哪些开发阶段？

5/9号

1. 瀑布模型的优缺点是什么，什么样的开发试用瀑布模型。
2. 快速原型的定义，特点，解决什么问题？（什么情况使用该模型）
3. 增量模型的定义，特点，解决什么问题？（什么情况使用该模型）
4. 敏捷模型解决什么问题？
5. 极限编程（XP）有哪些关键字？
6. 什么是用户故事，试着举例说明。

5/11号

1. Scrum开发模型有哪些关键字？
2. 可行性分析的任务有哪些？
3. 试枚举一些可能触犯法律法规的开发内容；
4. 系统流程图描述什么内容？
5. 什么是数据流图，包含的要素有哪些？

5/16号

1. 如何画数据流图。
2. 画数据流图的注意点。
3. 数据字典包含哪些内容；
4. 什么是人月，人月是什么的单位。“人月”与“人时”的换算关系；
5. 什么是人月神话？
6. LOC是什么的含义，是什么的单位？
7. 估算开发项目的开发时间的LOC方法（步骤）是什么？
8. 如何计算软件开发的生产效率（计算公式）？
9. 项目成本包含哪些内容？

5/18号

1. 需求分析的任务是什么。
2. 需求获取来源有哪些？
3. 需求具体包含的哪些方面的内容。
4. 需求获取有哪些手段。
5. 需求报告的作用是什么？
6. 需求报告的主要内容有哪些？
7. 写作技术文档的注意事项。
8. 对需求报告做评审检查的时候，

应该从那几个基本方面进行评审（需求文档的质量属性有哪些）？

- 正确；无二意；完整；可验证；一致；

1. 如何确保需求规格说明书的质量（验证需求的方法有哪些）？
2. 需求建模时，除了可以使用数据流图数据字典之外，还有哪些需求建模手段？

- E-R图；状态变迁图；以及UML建模工具。

1. 判定表包含哪些要素？（要求：判定表要会画，判定树要能读懂）
2. 需求跟踪矩阵的内容是什么？解决什么问题？

5/23号

1. 什么是checklist？
2. 试举若干配置管理工具的例子。
3. 为什么要进行分解模块化？
4. 总体（概要）设计的内容是什么。

* 将系统划分成模块
* 决定每个模块的功能
* 决定模块的调用关系
* 决定模块的界面，即模块间传递的数据

1. 模块化设计的重要指导思想是什么。

* 分解、信息隐蔽、确保模块独立性

1. 什么是耦合，举例1~2种了解的耦合方式。
2. 为什么不能使用goto语句？可以例外使用goto语句的情况是什么？
3. 为什么不能使用全局变量？

5/25号

1. 控制耦合有什么副作用，如何解决？
2. 标记耦合有什么副作用，如何解决？
3. 模块之间联系的原则是什么？

* 尽量使用数据耦合，少用控制耦合，限制公共耦合的范围，完全不用内容耦合。
* 模块间相互调用时，传递参数最好只有一个。

1. 什么是内聚？
2. 内聚有哪些类型？设计时应追求什么内聚？
3. 模块化的基本原则是什么。

* 高内聚，低耦合。

1. 模块的评价标准。

* 可分解；可组装；可理解；连续性；保护性；

1. 模块的功能可以预测是什么含义？
2. 什么是模块的深度、广度？划分的模块的时候，在深度广度上如何把控？
3. 如何定义一个模块的扇入和扇出？
4. 一个模块的控制域是什么？作用域又是什么？关系是什么？
5. 模块设计时的启发式规则有哪些。

* 低耦合、高内聚；
* 模块规模适中；
* 适当控制深度宽度；
* 降低接口的复杂程度；
* 争取单入单出；
* 模块功能可预测；

**5/30号**

1. 模块的入口和出口是什么含义？
2. 什么是变化流，什么是事务流；
3. 详细设计的内容是什么。
4. 详细设计的工具有哪些。都有哪些优缺点。

* - 程序流程图，盒图、pad图、伪代码。会将代码，或者程序流程图转换为控制流图。

1. 什么是环形复杂度，与算法复杂度有何异同？
2. 如何计算程序的环形复杂度。
3. 环路复杂度对编程有什么指导意义。

**6/01号**

1. 编程语言共有的特性有哪些。
2. 如何选择开发语言。
3. 什么是编码规范。
4. 枚举编码规范的基本要求，以及一些常用的编码规范

**6/06号**

1. 什么是程序内文档。
2. 如何做程序内文档（或者注意点有哪些）。
3. 枚举提升代码效率的方法。（代码、输入输出、数据库访问）
4. 如何确保程序结构清晰、且简单易懂（综合题目）；

6/16号

1. 什么是软件缺陷，试举一些**软件缺陷**的例子。

* 程序异常终止；
* 未达到需求说明书指明的要求；
* 出现了需求说明书中指明不会出现的错误；
* 功能超出了需求说明书指明的范围；
* 未达到需求说明书未指明但是应达到的要求；
* 难以理解，不易使用，运行速度缓慢；

1. 如何确保（保障）软件质量？

静态测试+动态测试

1. 为什么应该通过静态测试要找出30%~70%的bug（为什么需要静态测试而不是单纯依靠动态测试，而且还对静态测试bug摘出有目标比例要求）？

不能把大量bug遗留到测试阶段，会导致遗留给用户更高比例的bug，而且bug应该尽早发现修改（在静态检测阶段进行发现和修改），这样修改代价小，反之修改代价会大。

1. 测试的目的；
2. 什么是回归测试
3. 软件测试的准则有哪些？这些准则如何进行应用？
4. 为什么说：测试发现的错误中的80%很可能是由程序中20%的模块造成的。
5. 什么是白盒测试，什么情况需要进行白盒测试？
6. 为什么需要白盒测试，只进行功能测试是否可以？
7. 白盒测试都包含哪几种基本技法
8. 满足条件覆盖的测试用例，是否一定满足判定覆盖？
9. 什么是黑盒测试？黑盒测试的基本技法有哪些？

6/21号

1. 什么是基本路径，如何寻找基本路径？
2. 遇到循环语句，如何进行测试？
3. 什么是驱动模块，什么是桩模块？为什么要打桩和做驱动？
4. 自顶向下集成测试，需要构筑驱动模块还是桩模块？
5. 什么是黑盒测试，什么情况进行黑盒测试？
6. 除功能测试之外，系统测试还需要包含哪些测试（至少枚举3条）。
7. 什么是回归测试？为什么要进行回归测试？
8. α测试和β测试是什么含义？

6/23号

1. 程序运行发现了错误，如何定位错误、确认问题点？（综合问题，答案在维护和调试里面）

* **确认故障问题发生的软件和操作步骤**
* **-　发生时日**
* **-　该当制品名、版本**
* **-　OS版本**
* **-　现象的内容**

**发生时的现象、事前的操作内容、发生后的操作（确认是否有log的写入）、**

**再现性的有无**

* **-　确认其他运行的SW**
* **-　ＨＷ的器械名称（打印机名、网络）**
* **-　ＬＯＧ文件（event log、其它log）**
* **在对应的软件代码里面定位排错**
* **在输出储存器内容**
* **打印语句**
* **使用debug工具进行跟踪运行**
* 定位的策略：
* **试探法**
* **回朔法**
* **对分法**
* **归纳法**

1. 修改错误代码的注意点有哪些？能枚举至少若干条。

* **设计思想的一致性**
* **修改的处所尽可能的少，不要比需要修改的地方多**
* **尽量不使用共享系统中的已有变量，而使用局部量**
* **不要建立公用子程序，而建立各自独立的子程序**
* **坚持修改后的复审**
* **建立修改文档**
* **注意修改相应的文档**

1. 维护的种类有哪些。
2. 什么是再工程。
3. 什么是逆向工程。
4. 软件项目管理的内容包含哪些。至少枚举3条。
5. 甘特图一般用来描述什么内容？试举例进行绘制。
6. 作为一个合格的软件开发人员的条件有哪些，试枚举5条以上。
7. 什么是SQA？
8. 试枚举评价软件质量的方面。

* 正确性；
* 健壮性
* 可理解性
* 可维护性
* 可移植性
* 效率
* 可用性
* 安全性

1. 软件配置管理（SCM）的对象有哪些。

* 文档
* 代码
* 数据

1. 软件配置**管理**的目标

* **使软件的变更和修改可以更容易被适应，**
* **减少当变更必须发生时所需花费的工作量。**

1. 软件配置**管理**的内容。

* 标识变化；
* 控制变化；
* 确保适当地实现了变化；
* 向需要知道这类信息的人报告变化。

1. 配置管理工具举例。

* Svn
* Git
* Sourcesafe

1. 什么是基线，什么是里程碑
2. CMM各个级别的典型标志。

* 5：**最好的实践方式的确立和继续的过程改善**
* 4：**过程、成果物测定值被收集、被分析、定量的过程控制**
* 3：**组织过程的标准被确立，应用于管理和工程活动中**
* **2：可确立基本的项目管理、成本、进度获得了管理**
* **1：项目的成功依从于个人的努力（混沌）**

1. 如何确保软件的质量（综述题目）

* 正确的开发方式（需求获取、设计、编码这些环节的注意点，自行展开）
* 保障手段1：分阶段的评审（静态测试）
* 保障手段2：分阶段的测试（动态测试），确保代码覆盖、功能覆盖、非功能测试要注意不要遗漏，性能测试很重要。