1. 请结合GQM思想解释PSP过程的基本度量元有哪几个?

- GQM
  - 概念层（目标）：目标是为某个特定的对象而定义的。这里的对象是指软件产品、软件过程以及相关的资源等。
  - 操作层（问题）：基于一定的刻画上述目标是否达成或者目标达成的进展情况的模型，使用一系列的问题来定义所研究的对象，然后得出评价或评估特定目标达成进展情况。所选择的问题应当尽量体现质量相关的话题。
  - 量化层（度量）：试图以量化的方式回答上述操作层识别出来的问题。
- PSP基本度量项
  - 度量时间：所属阶段、开始时间、结束时间、中断时间、净时间
  - 度量缺陷：发现日期、注入阶段、消除阶段、消除时间、关联缺陷
  - 度量规模：
    - ➢ 选择的规模度量方式必须反映开发成本
    - ➢ 必须精确定义
    - ➢ 必须能有自动化方法统计
    - ➢ 有助于早期规划
  - 日程（TSP）

2. 产品组件集成策略有哪些？请解释这些策略的优缺点。在此基础上，解释如果要实现高质量集成，可能需要注意哪些方面。

- **大爆炸集成策略**
  该策略将所有已经完成的组件放在一起，进行一次集成。这是一种看起来非常具有吸引力策略。因为这有可能是需要测试用例最少的一种方式。然而，这需要所有待集成的产品组件都具有较高的质量水平，否则，难以定位缺陷位置的缺点会使得该策略消耗很多测试时间。而且，系统越复杂、规模越大，问题越突出。
- **逐一添加集成策略**
  该策略与上述的大爆炸集成策略完全相反，采取一次添加一个组件的方式进行集成。因此其优点就在于很容易定位缺陷的位置，特别在产品组件质量不高的情况下，每次集成之前都有着坚实的质量基础。但是，该方法的缺点也很突出。这可能是需要测试用例最多的一种策略，而且，大量的回归测试也会消耗很多时间。
- **集簇集成策略**
  集簇集成策略是对逐一添加集成策略的改进。简单的随机选择产品组件进行集成并不合理。为了提升测试效率，往往会把有相似功能或者有关联的模块优先进行集成，形成可以工作的组件。然后以组件为单位继续较高层次的集成。此外，这种策略还有一个好处就是，可以尽早获得一些可以工作的组件，有利于其他组件测试工作的开展。但是，这种策略的缺点是过于关注个别组件，而缺乏系统的整体观，不能尽早发现系统层面的缺陷。
- **扁平化集成策略**
  该策略要求尽快构建一个可以工作的扁平化系统。也就是说，优先集成高层的部件，然后逐步将各个组件、模块的真正实现加入系统。这种方式可以尽早发现系统层面的缺陷。然而，该策略的缺陷是为了确保完成的系统，需要大量的打“桩”（stub），即提供一些直接提供返回值的伪实现。这种方式往往不能覆盖整个系统应该处理的多种状态。

需要注意哪些方面？（不知道，大概根据每种策略的缺点说一下吧。。。）

3. 请给予Yield度量指标构建缺陷预测模型，并列举该模型的可能改进方案。

Yield

Phase Yield = 100 * (某阶段发现的缺陷个数) / (某阶段注入的缺陷个数 + 进该阶段前遗留的缺陷个数)

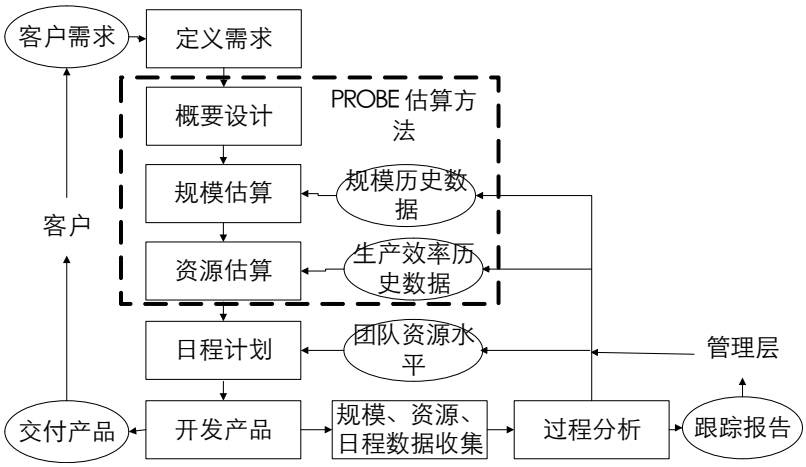
Process Yield = 100 * (第一次编译前发现的缺陷个数) / (第一次编译前注入的缺陷个数)

4. 谈谈你对项目估算的认识，并简要解释应用PROBE方法估算的优缺点。

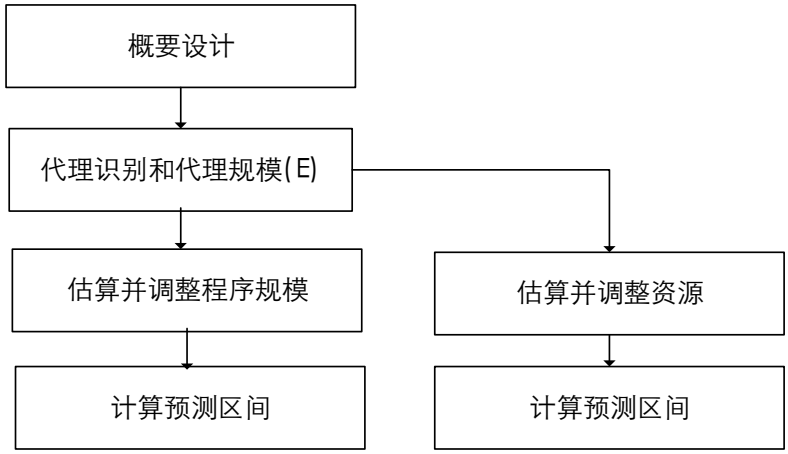
详细见复习讲义

PROBE估算原理：相对大小矩阵

通用计划框架



PROBE估算流程



优缺点：

- **简单方法：**计算简单，但是，不稳定（最小值作为VS、最大值作为VL、中值作为M，VS与M均值作为S、VL与M均值作为L）；
- **正态分布法：**相对稳定，在历史数据基本符合正态分布的情况下，可以给出非常好的相对大小矩阵（均值作为M，计算标准差σ，则S = M- σ ， VS = M-2 σ ， L = M+ σ ， VL = M+2 σ）
- **对数正态分布法：**更加符合人们对于程序的规模的直观感觉，因为大多数人习惯写很多规模很小的程序，少量规模较大的程序（以e为底计算所有数据的自然对数；取对数之后的值的均值作为M，计算相应标准差 σ；S = M- σ ， VS = M-2 σ ， L = M+ σ ， VL = M+2 σ；取反对数）。

5. 请解释PQI指标，如何计算，如何使用。

5个数据乘积（定义成0.0~1.0之间的数值）

- 设计质量：设计的时间应该大于编码的时间
- 设计评审质量：设计评审的时间应该大于设计时间的50%
- 代码评审质量：代码评审时间应该大于编码时间的50%
- 代码质量：代码的编译缺陷密度应当小于10个/千行
- 程序质量：代码单元测试缺陷密度应当小于5个/千行

6．请解释配置管理中配置项和产品基线的概念，并设计一个流程对单元测试后已经纳入配置库的代码，修改集成测试中的若干错误后，该如何控制变更。

- 配置项是在配置管理当中作为单独实体进行管理和控制的工作产品集合
- 基线是一个或多个配置项及相关的标识符的代表，是一组经正式审查同意的规格或工作产品集合，是未来开发工作或交付的基础，而且只能经由严格的变更控制程序才能改变。
- 如何控制变更
  - ➢ 跟踪变更请求：1）启动变更请求处理程序，将变更情况保存在变更请求数据库中；2）分析变更建议和所需进行的修改将对工作产品、进度、日程等造成的影响；3）如果变更请求影响到其他基线，则与相关的干系人一起审查这些变更请求，并取得他们的同意；4）跟踪变更申请直到结项
  - ➢ 控制配置项变更：1）确认这些修订已得授权；2）更新配置项；3）将旧基线归档保存，并获取新基线；4）执行审查，确保该变更没有对基线造成意料外的影响；5）上当记录配置项的变更内容和变更理由

7．如果对质量的追求是无止境的，在不考虑资源和成本的前提下，有哪些可能有效的策略?

- 测试+评审
- 质量策略
  - 首先确保基本没有缺陷，然后再考察其他的质量目标。
  - 用缺陷管理来代替质量管理
  - 高质量的产品也就意味着要求组成软件产品的各个组件基本无缺陷
  - 各个组件的高质量是通过高质量评审来实现的

8. 如何制定一份让人无法拒绝的计划，请描述基本步骤和一些注意事项

- 日程计划
  - 制定任务计划和日程计划；前者描述项目所有的任务清单，任务之间的先后顺序、以及每个任务所需时间资源，后者描述了各个任务在日程上的安排，哪天开始哪天结束；制定资源计划，结合任务计划即可定义日程计划
- 质量计划
  - ➢ 项目的质量计划中应当确定需要开展的质量保证活动。
  - ➢ 典型的质量保证活动包括个人评审、团队评审、单元测试、集成测试、系统测试以及验收测试等。
  - ➢ 在质量计划中需要解决的关键的问题是该开展哪些活动，以及这些活动开展的程度，如时间、人数和目标分别是什么。
- 风险计划

  目的是在风险发生前，识别出潜在的问题，以消除潜在问题的负面影响
  - 风险识别
    - ◇ 识别与成本、进度及绩效相关的风险
    - ◇ 审查可能影响项目的环境因素

- ◇ 审查工作分解结构的所有组件，作为风险识别的一部分，以协助确保所有的工作投入均已考虑
- ◇ 审查项目计划的所有组件，作为风险识别的一部分，以确保项目在各方面均已考虑
- ◇ 记录风险的内容、条件及可能的结果
- ◇ 识别每一风险相关的干系人
- ◇ 利用已定义的风险参数，评估已识别的风险
- ◇ 依照定义的风险类别，将风险分类并分组
- ◇ 排列降低风险的优先级

- 风险应对
  识别风险之后，就应当制定相应的风险管理策略，以应对各类风险
- 典型策略：风险转嫁，风险解决，风险缓解

9. 请分别介绍解释Deming、Crosby以及Juran等三位质量管理大师主要贡献以及他们的工作对软件过程和项目管理的借鉴意义。

Deming：

1） 提出质量改进的思想，并被二战后的日本工业界所采用；

2） 提出PDCA循环，被称为“戴明环”（Plan、Do、Check、Action），为最基本的质量和管理工具；

3） 提出十四点原则（树立改进产品和服务的坚定目标；采用新的思维方法；停止依赖检验的办法获得质量；不再凭价格标签进货；坚持不懈地提高产品质量和生产率；岗位培训制度化；管理者的作用应突出强调；排除畏难情绪；打破部门和人员之间的障碍；不再给操作人员提空洞的口号；取消对操作人员规定的工作定额和指标；不再采用按年度对人员工件进行评估；创建积极的自我提高计划制度；让每个员工都投入到提高产品质量的活动中去）。

Crosby：

1） 提出了“零缺陷”的概念，即第一次就把事情做对。而要做到这一点，就需要把工作放在预防上而不是质量检验上；

2） 提出了质量管理的绝对性：
- 质量就是符合要求，而不是“完美”
- 质量来自于预防，而不是检验
- 质量的标准是“零缺陷”，而不是可接受质量水平
- 质量的衡量标准是“不符合要求的代价”

3） 提出质量改进的基本要素（6C-变革管理的六个阶段）：
- 领悟（Comprehension）——理解质量真谛
- 承诺（commitment）——制定质量策略的决心
- 能力（capability）——教育与培训
- 沟通（communication）——成功的经验文档化、制度化
- 改正（correction）——预防与提高绩效
- 坚持（continuance）——强调质量管理成为一种工作方式

4） 发展质量成熟度的度量。

Juran：

1） 其主编的《质量控制手册》为当今世界质量控制科学的“圣经”，奠定了“全面质量管理TQM”的理论基础（Total Quality Management）；

2） 提出了适用性质量（质量是一种适用性，即产品在使用期间能满足使用者的要求），质量不仅要满足明确的需要，也要满足潜在的需求。这一思想使质量管理范围从生

产过程中的控制进一步扩大到产品开发和工艺设计阶段，即挖掘用户潜在需求；

3） 提出了质量三步曲（质量计划->质量控制->质量改进）；

4） 提出了Juran质量螺旋；

5） 提出了80/20原则，认为有80%的质量问题是由于领导责任引起的。从而将人力与质量管理结合起来。

Shewhart：（补充）

1） 最早将统计控制的思想引入质量管理，为质量改进奠基人；

2） 提出PDS模型（计划-执行-检查 Plan-Do-See），后被戴明进一步发展为PDCA。

Humphrey：（补充）

1）采用Crosby的成熟度度量，提出了软件能力成熟度模型（CMM），对于软件过程管理与改进具有建设性作用。

10. 请结合软件开发特点介绍软件项目管理中自主型团队的必要性。

团队的定义：

一个团队必须包括至少两个成员， 他们为了共同的目标和愿景而努力工作，他们每个人都有明确的角色和相应的职责定义，任务的完成需要团队成员互相依赖和支持。

自主团队的含义：

软件工程师所从事的工作一般称之为复杂的知识工作。在这种性质的工作中，实现软件工程师的自我管理往往可以获得最好的工作效率和质量水平。如果整个团队的所有成员都实现了自我管理，也就形成了所谓的自主团队。

自主团队的特点：

- 自行定义项目的目标；
- 自行决定团队组成形式以及成员的角色；
- 自行决定项目的开发策略；
- 自行定义项目的开发过程；
- 自行制定项目的开发计划；
- 自行度量、管理和控制项目工作；

自主团队的必要性：

自主团队可以形成“胶冻态团队”。在这样的团队中存在一种神奇的力量，这种神奇力量弥漫于该团队做的所有工作；团队成员互相支持，更为重要的是，团队成员在任何时刻都知道应该以怎样的方式帮助别人；团队成员相互信任，有强烈的归属感；团队在适当的知道会聚集在一起，研究现状，讨论策略。

自主团队的形成：

自主团队不是偶然形成的。大部分情况下，在团队建立之初，团队成员往往有着不同的目标；缺乏清晰的角色定义和职责安排；对于待开发的产品只有模糊的概念；团队成员也有着不同的工作习惯和工作方法。经过一段时间的协同工作，团队成员可以慢慢培养团队协作方式，从而逐渐演化成自主团队。

自主团队的外部环境：

1）项目启动阶段获得管理层的支持

- 项目小组应当体现出已经尽最大的可能在满足管理层的需求的工作态度。
- 项目小组应当在计划中体现定期需要向管理层报告的内容。
- 项目小组应当向管理层证明他们所制定的工作计划是合理的。
- 项目小组应当在计划中体现为了追求高质量而开展的工作。
- 项目小组应当在工作计划中允许必要的项目变更。
- 项目小组应当向管理层寻求必要的帮助。

2）在项目进展过程中获得管理层的支持

- 项目小组应当严格遵循定义好的开发过程开展项目开发工作。
- 项目小组应当维护和更新项目成员的个人计划和团队计划。
- 项目小组应当对产品质量进行管理。
- 项目小组应当跟踪项目进展，并定期向管理层报告。
- 项目小组应当持续地向管理层展现优异的项目表现。

2014重点整理：

- 质量管理策略
  首先确保基本没有缺陷，然后再考察其他的质量目标。
  用缺陷管理来代替质量管理
  高质量的产品也就意味着要求组成软件产品的各个组件基本无缺陷
  各个组件的高质量是通过高质量评审来实现的
- 理由，为什么有效?
  提高生产效率，通过关注每个组件的质量，往往可以避免在集成测试和系统测试消除大量缺陷，减少消除代价，提升生产效率
- 评审质量指标
  - ➢ Yield
    Phase Yield = 100 * (某阶段发现的缺陷个数) / (某阶段注入的缺陷个数 + 进该阶段前遗留的缺陷个数)
    Process Yield = 100 * (第一次编译前发现的缺陷个数) / (第一次编译前注入的缺陷个数)
  - ➢ A/FR
    PSP质检成本/PSP失效成本，越高越好，期望值是2
  - ➢ PQI
    5个数据乘积：设计质量 * 设计评审质量 * 代码评审质量 * 代码质量 * 程序质量
  - ➢ 评审速度
    代码评审速度小于200 LOC/小时，文档评审速度小于4 Page/小时
  - ➢ DRL
    以某个测试阶段（一般为单元测试）每小时发现的缺陷数为基础，其他阶段每小时发现缺陷数与该测试阶段每小时发现的缺陷的比值就是DRL
- 质量计划
  - ➢ 项目的质量计划中应当确定需要开展的质量保证活动。
  - ➢ 典型的质量保证活动包括个人评审、团队评审、单元测试、集成测试、系统测试以及验收测试等。
  - ➢ 在质量计划中需要解决的关键的问题是该开展哪些活动，以及这些活动开展的程度，如时间、人数和目标分别是什么。

  风险计划
  目的是在风险发生前，识别出潜在的问题，以消除潜在问题的负面影响
  - 风险识别
    - ◇ 识别与成本、进度及绩效相关的风险
    - ◇ 审查可能影响项目的环境因素
    - ◇ 审查工作分解结构的所有组件，作为风险识别的一部分，以协助确保所有的工作投入均已考虑
    - ◇ 审查项目计划的所有组件，作为风险识别的一部分，以确保项目在

各方面均已考虑
    - 记录风险的内容、条件及可能的结果
    - 识别每一风险相关的干系人
    - 利用已定义的风险参数，评估已识别的风险
    - 依照定义的风险类别，将风险分类并分组
    - 排列降低风险的优先级
  - 风险应对
    识别风险之后，就应当制定相应的风险管理策略，以应对各类风险
    典型策略：风险转嫁，风险解决，风险缓解

- 计划、管理是产生的启发
- 估算与计划的差异
  - 估算主要用来估算待开发程序的规模和所需资源
  - 质量计划用于保证项目的质量
  - 包含的活动不同等等
- 如何做一份无可拒绝的计划
  制定任务计划和日程计划；前者描述项目所有的任务清单，任务之间的先后顺序、以及每个任务所需时间资源，后者描述了各个任务在日程上的安排，哪天开始哪天结束；
  制定资源计划

- 过程定义度量
  PSP基本度量项
  - 度量时间：所属阶段、开始时间、结束时间、中断时间、净时间
  - 度量缺陷：发现日期、注入阶段、消除阶段、消除时间、关联缺陷
  - 度量规模：
    - 选择的规模度量方式必须反映开发成本
    - 必须精确定义
    - 必须能有自动化方法统计
    - 有助于早期规划
  - 日程（TSP）