

技术解决方案



中国软件评测中心
北京赛迪国软认证有限公司



基本信息

- 中文名称：技术方案
- 英文名称：Technical Solution
- 缩写：TS
- CMMI等级：3级
- 过程类型：工程
- 相关过程域：RD、VER、DAR、REQM、OID



技术解决方案

- 目的：是开发、设计和实现满足需求的解决方案。解决方案、设计结果和实现成品包括产品、产品组件和产品相关的生命周期过程（可能是其中之一或者它们的组合）



技术解决方案

- 技术解决方案过程域适用于产品架构的任何等级，且适用于所有产品、产品组件、产品相关生命周期过程及服务。本过程域专注于下列事项：
 - 评估与选择解决方案（有时称为“设计方案”、“设计概念”或“初步设计”），满足适当的已配置需求；
 - 对选定的解决方案开发详细设计（详细到包括制造、程序制作或实现设计为产品或产品组件所需的信息）；
 - 实现设计成为产品或产品组件



不恰当的技术解决方案

- ❑ 选择了一个无效（或效率底下）的解决方案。
- ❑ 产品不能满足技术性能要求或用户需求。
- ❑ 需要不断地增加测试或返工来解决设计问题。
- ❑ 产品不能提供技术升级支持或功能扩充。



特定目标（Specific Goals）

- ❑ **SG1选择产品组件解决方案**

从备选方案中，选择产品或产品组件解决方案

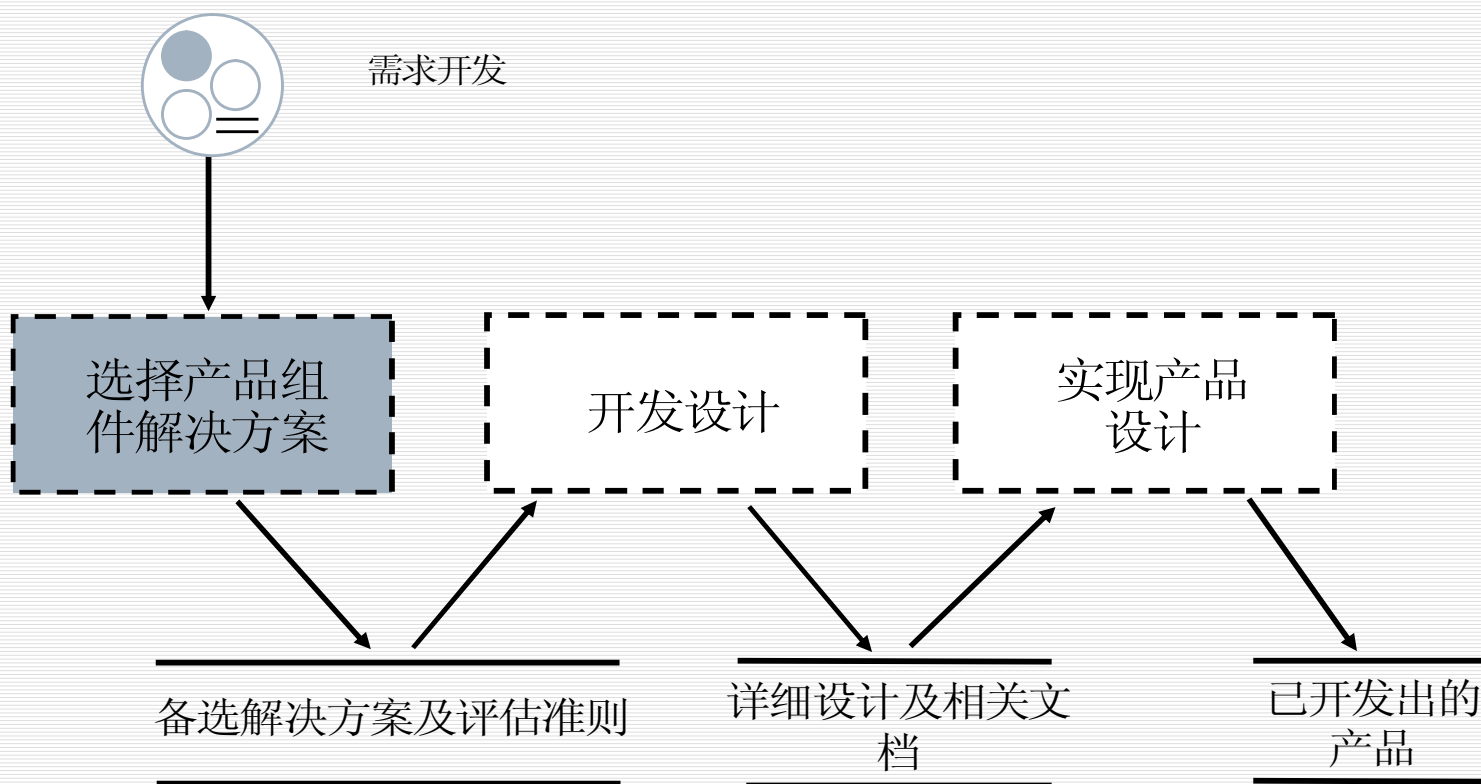
- ❑ **SG2开发设计**

开发产品或者产品组件的设计

- ❑ **SG3实现产品设计**

按照设计实现产品组件和相关的支持文档

SG1选择产品组件解决方案





SG1选择产品组件解决方案

- SP1.1 开发备选方案和制定选择准则
- SP1.2 选择产品组件方案



SP1.1 开发备选方案和制定选择准则

- 备选方案涵盖成本、进度、性能的可接受范围，产品组件需求与设计问题、约束和准则一起用于开发备选方案。
- 选择准则一般包括成本(如时间、人员、费用)、效益(性能、能力、效率)和风险(技术、成本、进度)，备选方案和选择准则可以包括：
 - 成本（包括开发、制造、采购、维护和支持等）
 - 性能



SP1.1 开发备选方案和制定选择准则

- 产品组件和产品相关生命周期过程的复杂度
- 健全的产品操作和使用条件、操作模型、环境以及产品相关生命周期过程的变动
- 产品扩展和升级
- 技术限制
- 对构造方法和原材料的敏感性
- 风险
- 需求和技术的演变
- 报废
- 终端用户和操作者的能力和约束条件
- 商品现货的特征



SP1.1 开发备选方案和制定选择准则

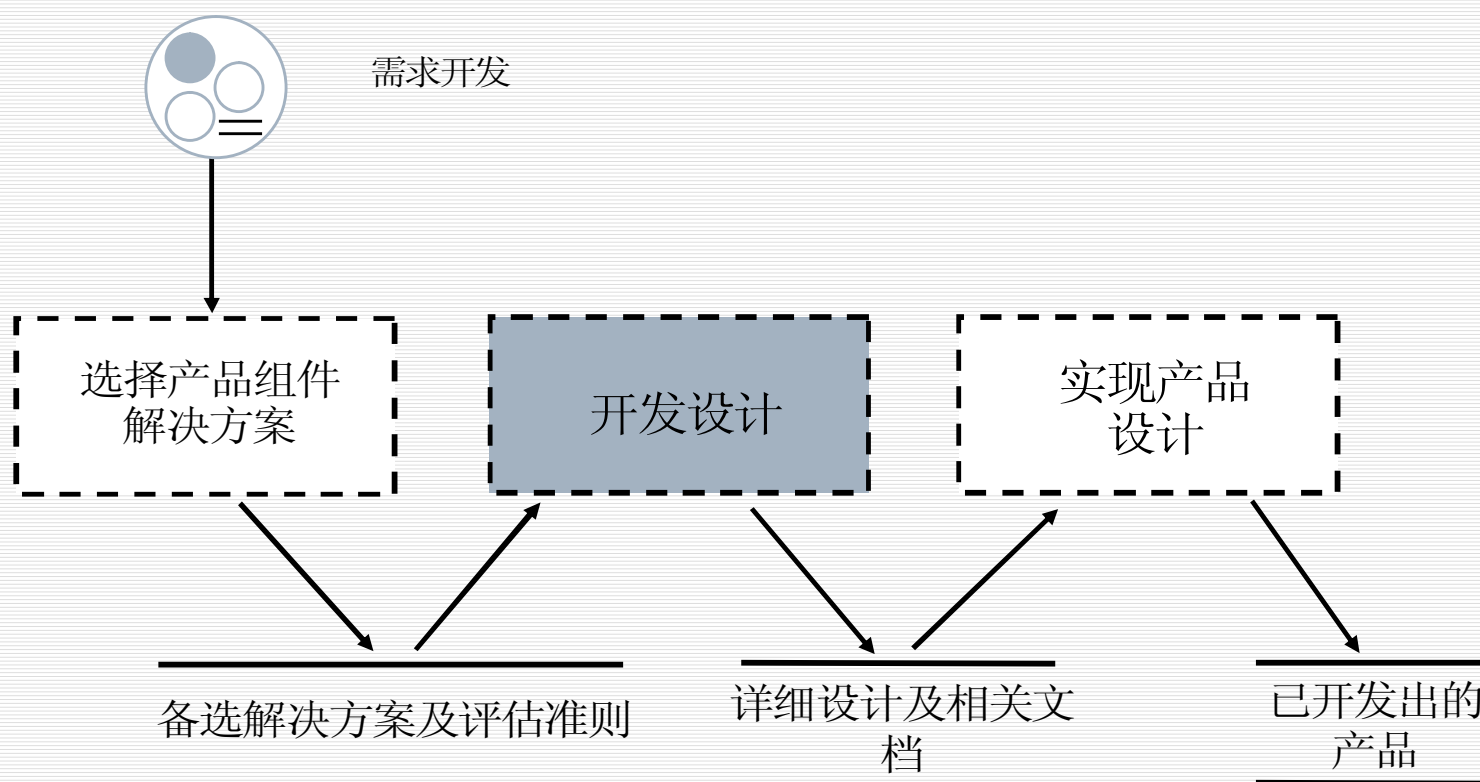
- 用来选择最终解决方案的准则应该提供成本、效益和风险之间的平衡。
- 典型工作成果
 - 备选方案评价准则
 - 新技术评估报告
 - 备选方案
 - 最终方案的选择准则
 - 商品现货的评估报告



SP1.2 选择产品组件方案

- 依据选择准则，选出“**最佳**”解决方案；相关人员对备选解决方案进行评估和决策，根据评估记录和决策意见，确定最终的解决方案，并以此为基础进行后续活动。
- 典型工作成果
 - 产品组件方案选择决定和理由
 - 需求与产品组件之间对应关系的文档
 - 解决方案、评价准则和选择理由的文档

SG2开发设计





SG2开发设计

开发产品和产品组件的设计

- ☐ SP2.1 设计产品或产品组件
- ☐ SP2.2 建立技术数据包
- ☐ SP2.3 基于规则的接口设计
- ☐ SP2.4 分析“自制、购买或重用”



SP2.1 设计产品或产品组件

- 产品设计包含两个阶段：概要设计和详细设计。**概要设计**建立产品能力和产品架构(包括产品划分、产品组件标识、系统状态与模型、主要的产品内部和全部的外部接口等)，**详细设计**定义产品组件的结构和功能。
- 典型工作成果
 - 产品架构
 - 产品组件设计



SP2.1 设计产品或产品组件-产品架构

- 架构定义是由需求开发过程中的开发架构需求而来，这些需求代表了产品成功的关键品质和性能。架构定义可以包含：
 - 建立功能模块的结构化关联与功能模块内的组件，以及功能模块间的接口规则
 - 建立主要的内部接口和所有的外部接口
 - 识别产品组件以及它们之间的接口
 - 开发产品组件或类和框架
 - 识别主要的重用方法和资源.....



SP2.2 建立技术数据包

□ 技术数据包是向开发者提供的完整产品和组件的描述。技术数据包应该包含下列数据信息（如原料与制造的需求对与软件服务或过程相关的产品组件就没什么用）。

- 产品架构描述
- 分配的需求
- 产品组件的描述
- 产品相关生命周期过程的描述
- 关键产品特性

■ 必需的物理特性和约束



SP2.2 建立技术数据包

- 技术数据包中含大量数据，且这些数据对产品组件的成功开发非常重要，因此建议确定相应的准则来组织数据和选择数据内容，使用产品架构来组织数据抽象视图，使得感兴趣的数据能以恰当和清晰的形式呈现，是一种非常有用的方法。这些视图包括：客户、需求、环境、功能、逻辑、安全、数据、状态/模型、构建和管理。
- 典型工作成果
 - 技术数据包



SP2.3 基于规则的接口设计

- 根据已经建立和维护的准则，设计合适的产品组件接口。
- 接口设计包括
 - 来源
 - 目的地
 - 软件触发点和数据特性
 - 硬件的电子、机械及功能特征



SP2.3 基于规则的接口设计

- 典型工作成果
 - 接口设计规范
 - 接口控制文档
 - 接口描述规则
 - 选取接口设计的理由



SP2.4 分析“自制、购买或重用”

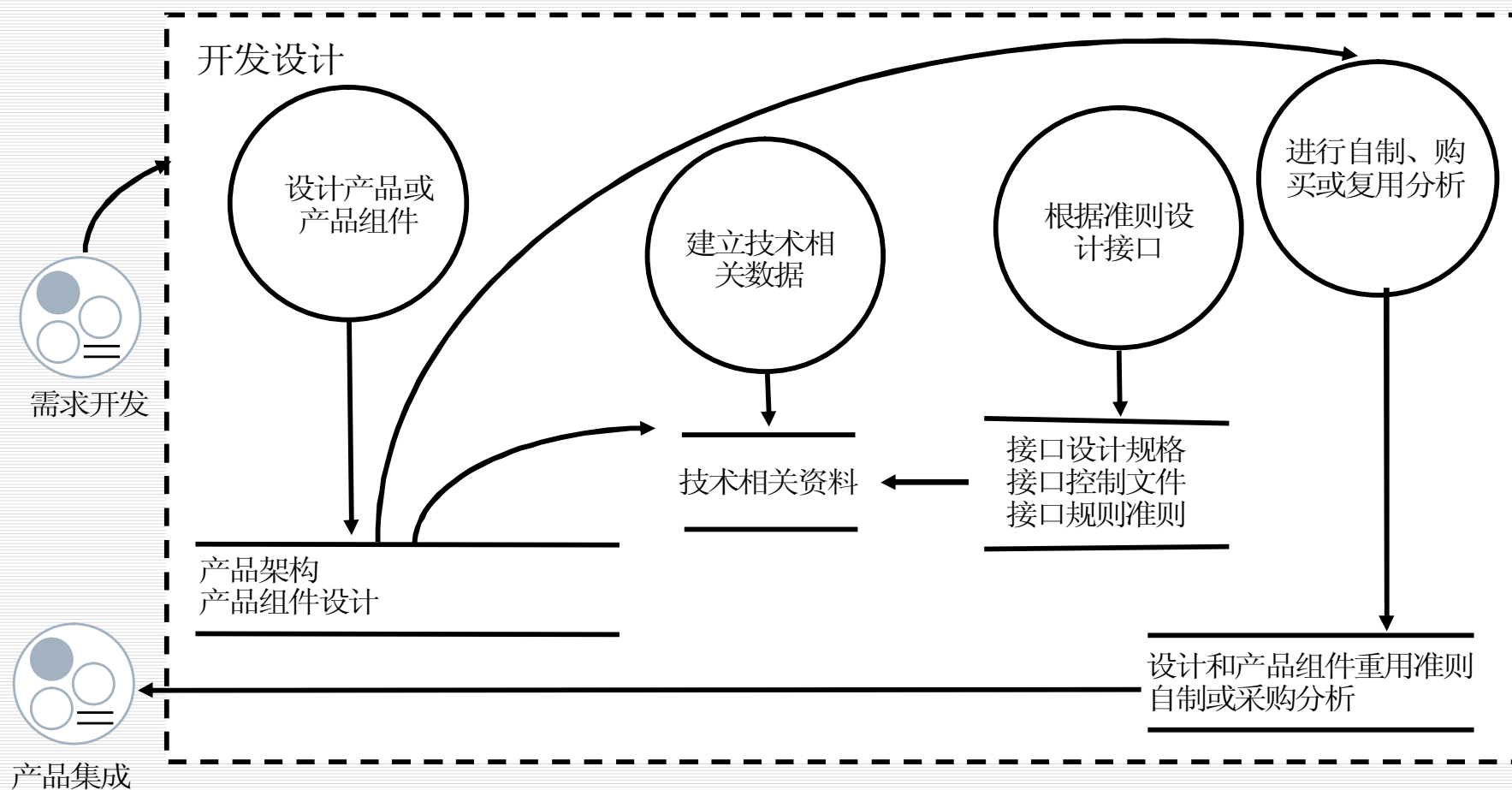
- 根据对项目需求的分析，决定产品组建采用自己制作、外购还是重用以前的部件。
- 典型工作成果
 - 设计和组件重用原则
 - 自制和外购原则
 - 选购商用组件的指导原则



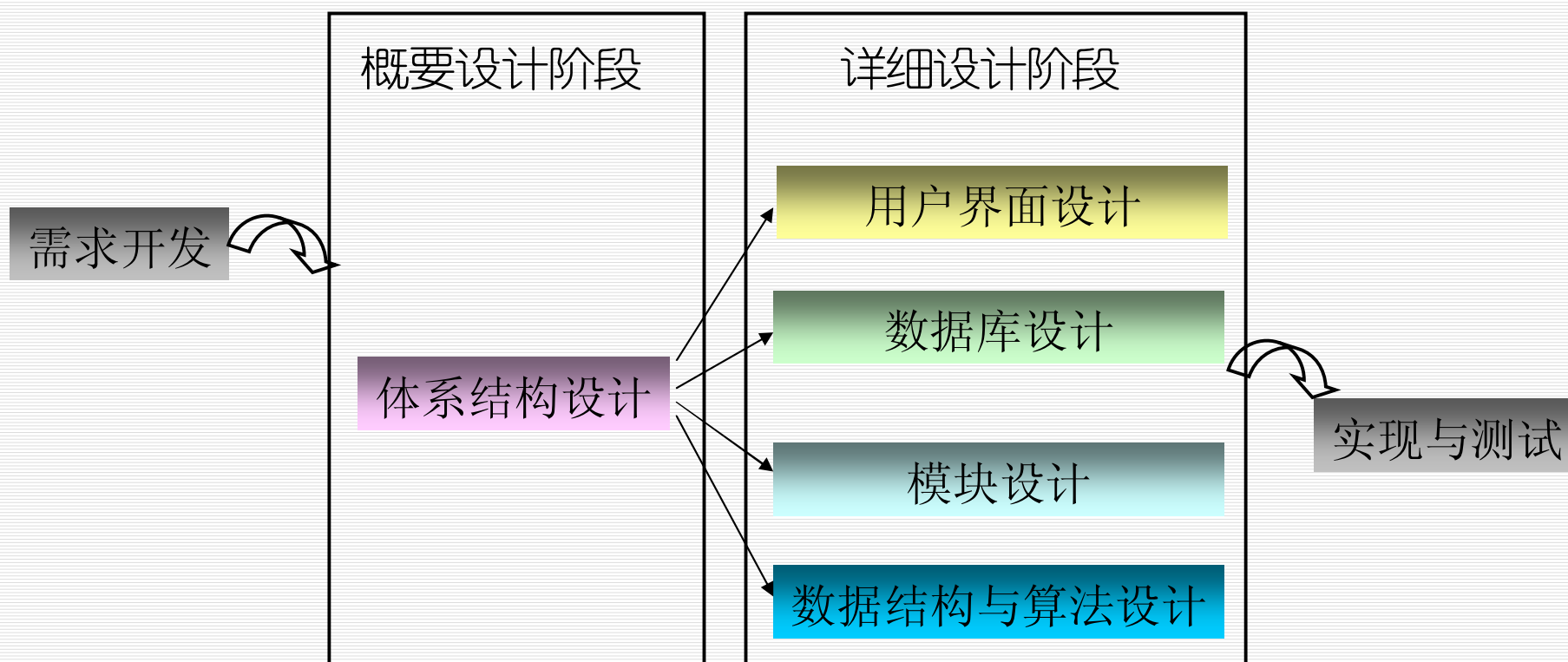
SP2.4 分析“自制、购买或重用”

- 影响决策的因素有：
 - 产品或服务提供的功能以及这些功能如何适应项目的需要
 - 可用的项目资源和技术
 - 选购商用组件的指导原则
 - 内部开发与外购成本的比较
 - 关键的交付日期和集成日期
 - 战略商业联盟，包含高级的商业需求
 - 可用产品的市场调研，包括商品现货
 - 可用产品的功能与品质
 - 潜在供应生的技术和能力
 - 对核心竞争力的影响
 - 有关外购产品的许可证、保证书、权责和限制
 - 产品可用性
 - 所有权问题
 - 风险缓解

SG2开发设计

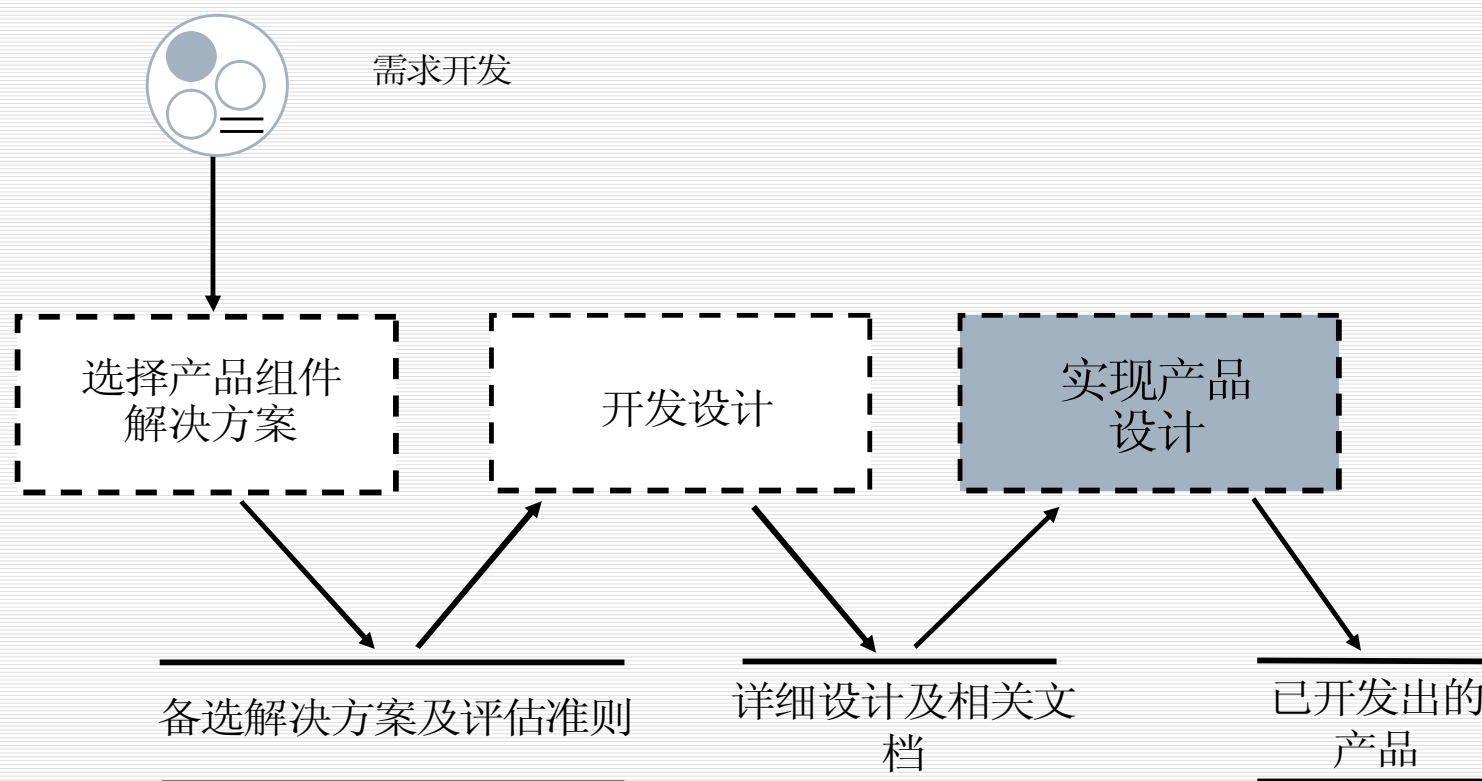


软件设计过程



软件设计过程示意

SG3实现产品设计





SG3实现产品设计

- ☐ SP3.1 实施设计
- ☐ SP3.2 编写产品支持文档



SP3.1 实施设计

- 一旦设计完成，就要把它作为产品组件予以实现。
实现特性举例如下：
 - 软件已编码
 - 数据已文档化
 - 服务已文档化
 - 电子和机械部件已制作
 - 产品独特的制造过程已置于实际操作中
 - 过程已记录
 - 设施已建造
 - 原材料已生产
 - 典型工作成果：已经实现的设计
-



SP3.1 实施设计-子实践

- 1.使用有效的方法实现产品组件。软件编码的方法有：
 - 结构化程序设计
 - 面向对象程序设计
 - 自动产生程序代码
 - 软件代码重用
 - 使用适合的设计模式



SP3.1 实施设计-子实践

□ 2.遵循恰当的标准与准则。

编码标准：

- 程序设计语言标准
- 变量命名规则
- 可接受的语言程序结构
- 软件产品组件的结构和层次
- 程序代码的注释和格式



SP3.1 实施设计-子实践

编码准则：

- 模块化
- 清晰
- 简单
- 结构化
- 程序代码的注释和格式（例如不使用**Goto**语句，单一出、入口）
- 可维护性



SP3.1 实施设计-子实践

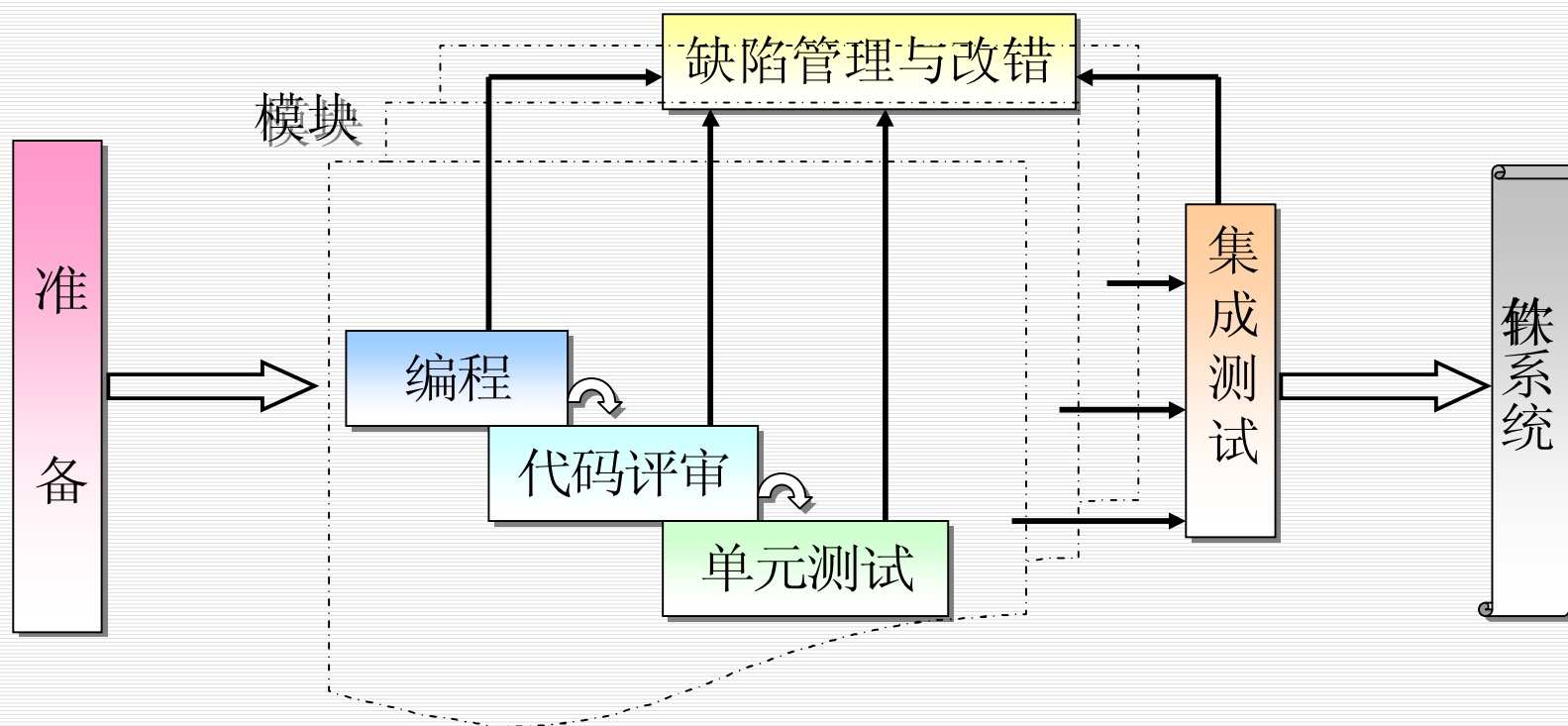
- ☐ 3.对选定的产品组件执行同行评审。
- ☐ 4.对产品组件执行适当的单元测试。

单元测试的方法：

- ☐ 语句覆盖测试
 - ☐ 分支覆盖测试
 - ☐ 断言覆盖测试
 - ☐ 路径覆盖测试
 - ☐ 边界值测试
 - ☐ 特殊值测试
- ☐ 5.必要时对产品组件进行修订。

SP3.1 实施设计-编码

■ 编码与测试流程图





SP3.2 编写产品支持文档

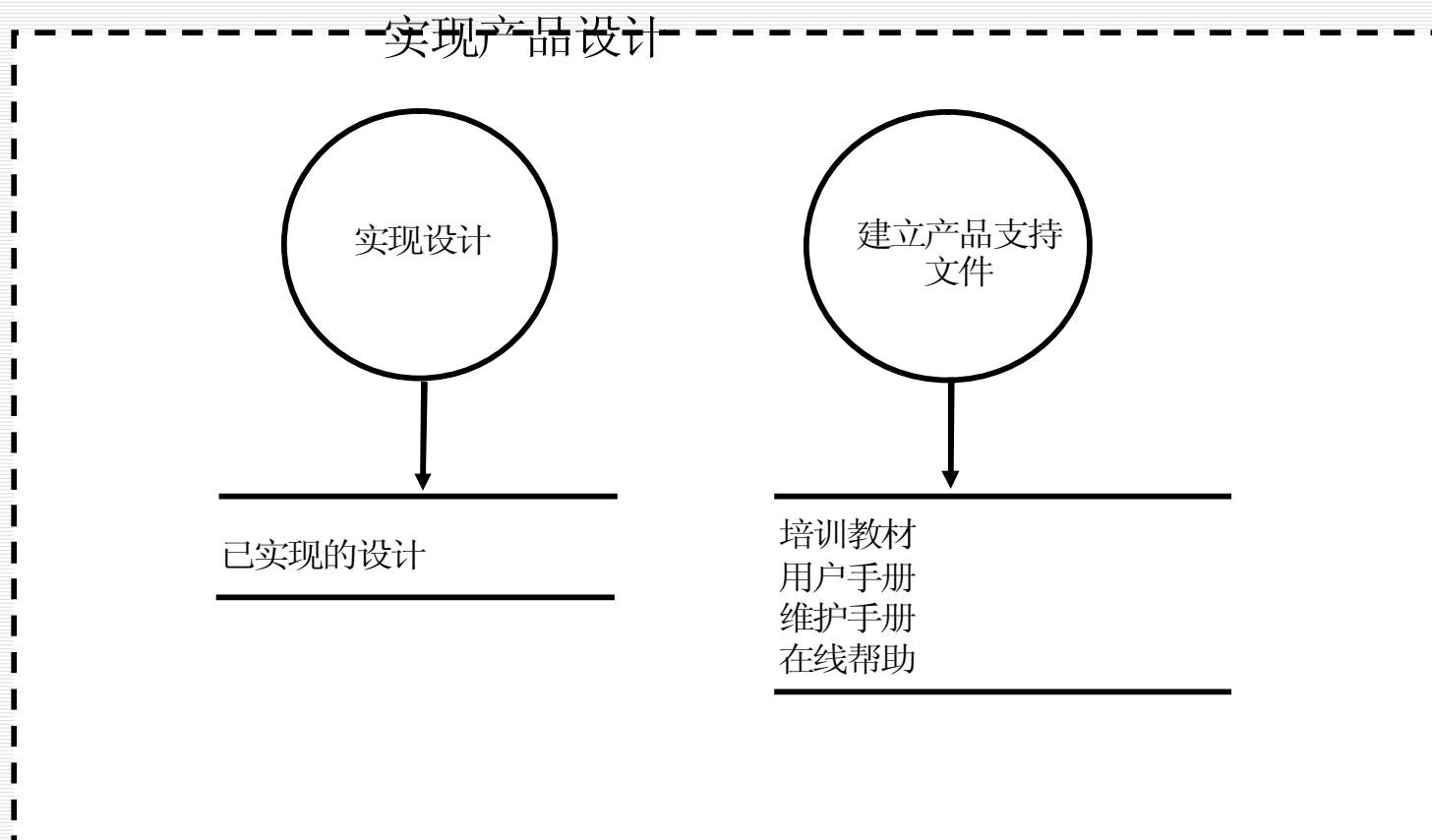
- 开发并维护用于产品安装、操作和维护的相关文档
- 典型工作成果
 - 最终用户培训材料
 - 用户手册
 - 操作手册
 - 维护手册
 - 在线帮助



SP3.2 编写产品支持文档

- 文档制作的标准举例：
 - 与指定的文字处理器兼容
 - 可接受的字体
 - 分页、章节和段落的编码
 - 与指定的手册风格一致
 - 缩写的使用
 - 安全分级标识
 - 国际化的需求

技术解决方案





TS----总结

- ☐ 备选方案
- ☐ 概要设计说明书
- ☐ 详细设计说明书
- ☐ 数据库设计说明书
- ☐ 用户手册
- ☐ 帮助文档
- ☐

谢谢！

