

产品集成

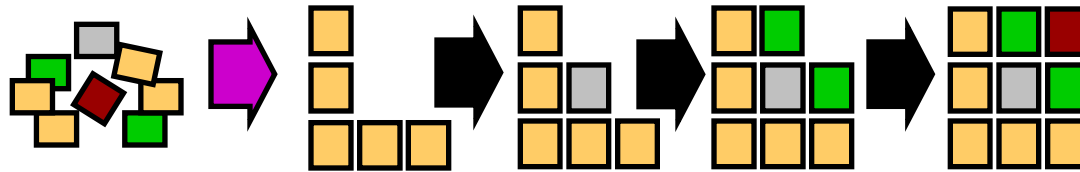
Product Integration

麦哲思科技（北京）有限公司

- 过程域概述
- 特定实践解析
 - 模型原文与参考译文
 - 理解与实施要点
 - 业界案例
- 体系建立要点
- 案例解析

什么是产品集成

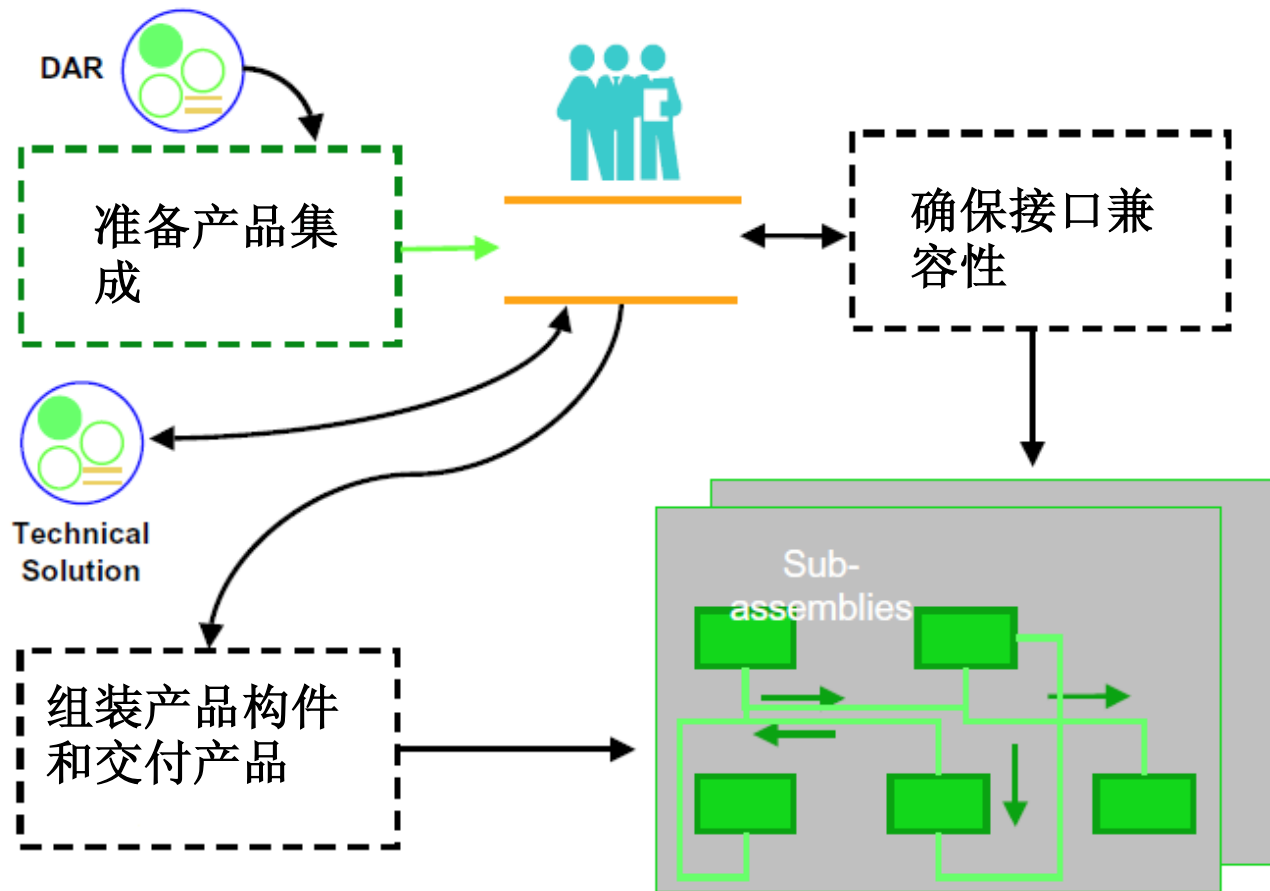
- 集成的逆向是分解
- 在开发的初期, 对产品进行了抽象的产品分解, 并在过程中实现了物理的产品部件的开发, 最终需要物理的将产品部件集成为产品



请列举你曾经拆解但再也无法装配的器具
反思为什么装配失败？

产品集成的目的与目标

- 目的：将产品构件组装为产品，确保集成后的产品运行正确（如，实现了必需的功能与质量属性）并交付产品。



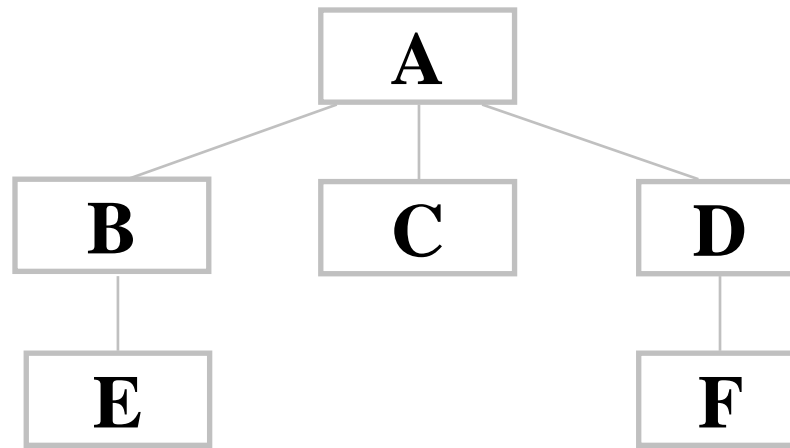
- SP1.1 建立集成策略：建立并维护产品集成策略
- SP1.2 建立产品集成环境：建立并维护必要的环境，以支持产品构件的集成
- SP1.3 建立产品集成规程和准则：为产品构件集成建立和维护规程和准则
- SP2.1 评审接口描述的完整性：评审接口描述的完整性和覆盖
- SP2.2 管理接口：管理内部和外部的接口的定义、设计和产品及产品构件变更
- SP3.1 确认集成用的产品构件已经准备就绪：在进行组装之前，确认组装产品所需的每个产品构件已经恰当地标识、都能按照其描述正常运行，产品构件接口符合接口描述
- SP3.2 组装产品构件：遵循产品集成顺序和可用的规程组装产品构件
- SP3.3 评价组装的产品构件：评价组装的产品构件及接口
- SP3.4 包装并交付产品或产品构件：对组装完成的产品或产品构件打包并把它交付给客户

特定实践解析

- 模型原文
 - Establish and maintain a product integration strategy
- 参考译文
 - 建立并维护产品集成策略
- 原文释义与实施要点
 - 产品集成策略中包含的内容：
 - 一次性集成、增量式集成、每日集成、持续集成？
 - 每次集成的构件有哪些？
 - 接口验证方法：单元测试、接口评审、技术原型、模型？
 - 集成的环境定义
 - 集成测试的工具、设备
 - 集成后的测试重点
 - 集成策略与项目的技术方案、开发顺序相关
 - 确定产品的集成策略也可能需要做DAR

- 业界案例

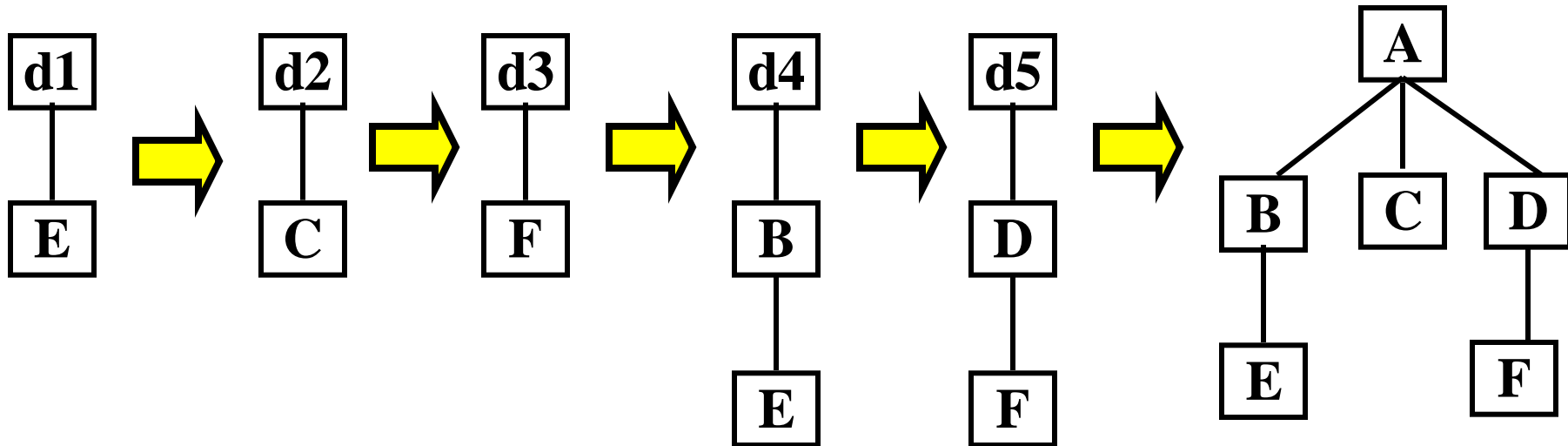
- 对于工期短、项目规模小、模块关系简单的项目，在单元测试完毕后，按一次性集成方式执行集成。
- 在敏捷的实践中强调了每日构建，持续集成。
- 对于模块间关系较复杂的项目，采用增量集成的方式逐步集成，以避免集成问题的集中爆发和延迟发现。
 - 按照子功能和模块的开发完成的顺序来确定集成的顺序
 - 自顶向下集成：从主控模块（主程序）开始沿控制层向下移动，把模块一一组合起来。
 - 自底向上集成：从程序模块结构中最底层的模块开始组装和测试。



自顶向下, 深度优先:A、B、E、C、D、F

自顶向下, 广度优先:A、B、C、D、E、F

SP1. 1建立集成策略-4



自底向上

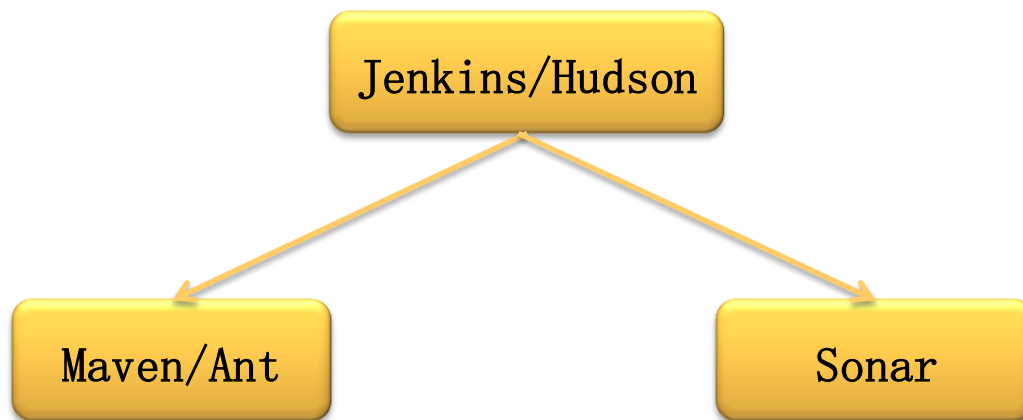
讨论：自顶向下集成和自底向上集成各有什么优缺点？

- 模型原文
 - Establish and maintain the environment needed to support the integration of the product components.
- 参考译文
 - 建立并维护支持产品构件集成所需要的环境
- 原文释义与实施要点
 - 集成环境包括组装的环境与执行集成测试的环境，两个环境可以相同，也可以分别建立。
 - 组装可以在源码服务器上执行，集成测试的环境可以与系统测试的环境相同。
 - 对于嵌入式软件开发，可以使用模拟器模拟运行环境来执行集成。
 - 集成环境可能采购也可能自做或者复用已有的

SP1.2 建立产品集成环境-2

- 业界案例
 - 设立环境小组，对各个项目的集成环境统一建立和维护
 - 编制测试环境搭建和操作手册
 - 使用测试环境清单检查集成环境的准备情况

环境类别	详细环境清单	责任人	当前状况
硬件环境			✓
			×
网络环境			✓
			×
软件环境			✓
			×

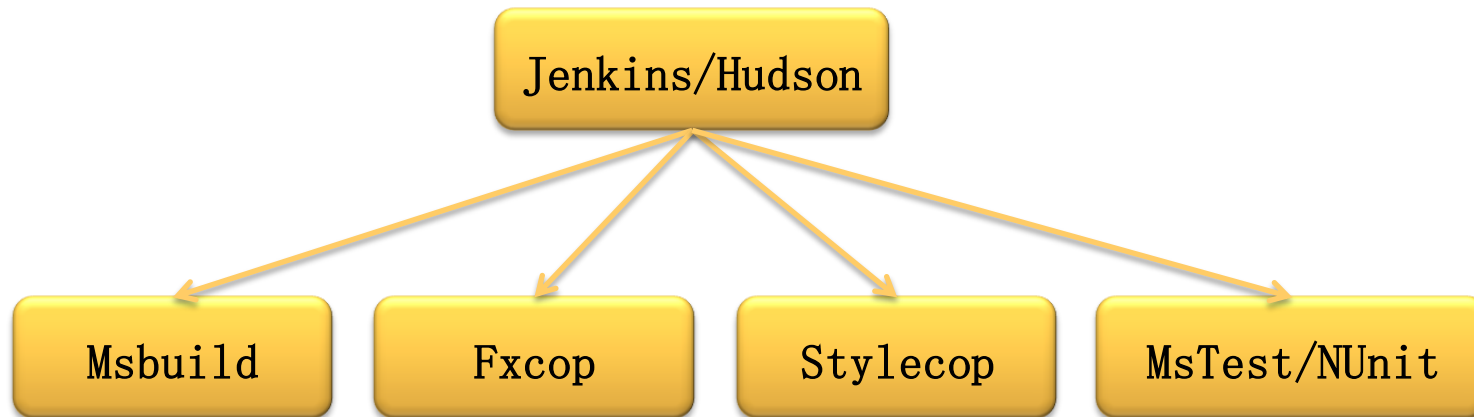


Jenkins/Hudson是目前比较流行的CI引擎，特点是简单易用,扩展性强。

构建工具可以用Ant与Maven。

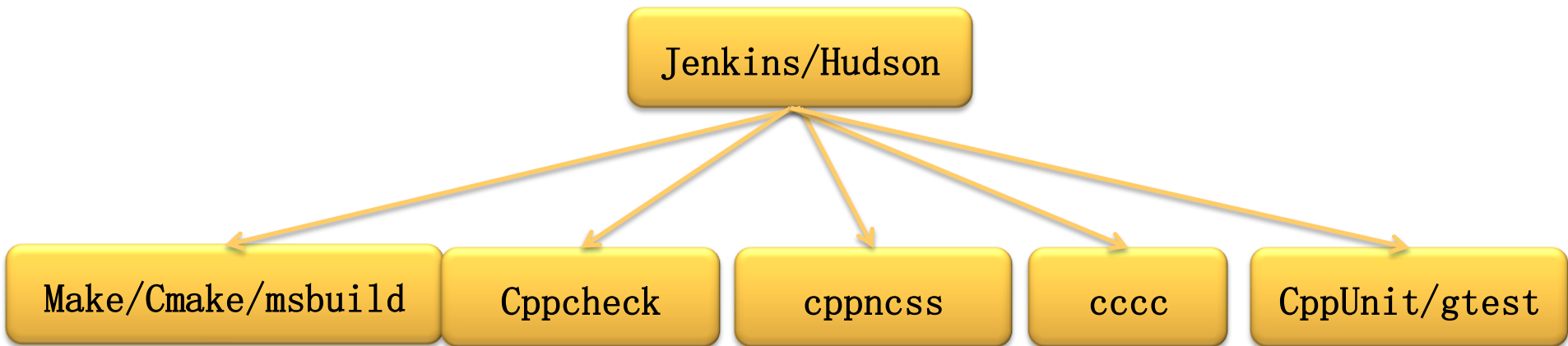
Sonar是一个很强大的QA工具，本身集成了Junit、覆盖率、代码规范和静态结构分析工具。

.Net下的持续集成平台



Fxcop是代码逻辑检查工具，
Stylecop是编码规范检查工具。
单元测试可以用MsTest或者Nunit.

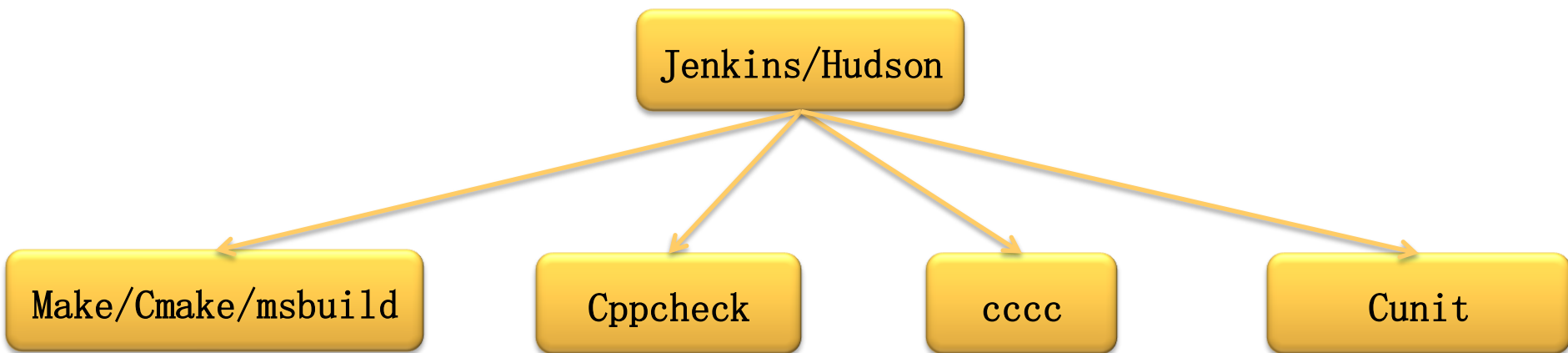
C++下的持续集成平台



如果是Linux平台下的C++，主流的构建工具是make。

如果是windows平台的可以使用msbuild。

cppcheck是静态结构分析工具，cppncss是代码圈复杂度和代码行统计工具，单元测试可以用CppUnit或者Google test(gtest).



如果是Linux平台下的C，主流的构建工具是make。

如果是windows平台的可以使用msbuild。

cppcheck是静态结构分析工具，cccc是代码圈复杂度和代码行统计工具，单元测试可以用CUnit

- 模型原文
 - Establish and maintain procedures and criteria for integration of the product components.
- 参考译文
 - 为产品构件集成建立和维护规程和准则
- 原文释义与实施要点
 - 集成的规程包括：增量迭代的次数、编译与链接的脚本、集成测试的重点、在每次集成时执行的其他评价
 - 集成的准则包括：集成进入与退出的准则、集成测试的用例和环境的要求等
 - 可以单独编写一个集成策略或计划文档，也可合并项目的主计划中

- 业界案例
 - 集成的一般步骤
 - 1. 确保开发人员提交了最新的版本
 - 2. 取得要集成的代码
 - 3. 编译、链接、制作安装包
 - 4. 安装并执行测试
 - 5. 标记集成通过后的版本
 - 6. 通知开发/测试人员
 - 集成完成的准则示例：
 - 成功地执行了测试计划中规定的所有集成测试；
 - 修正了所发现的错误；
 - 测试结果通过了评审；

- 模型原文
 - Review interface descriptions for coverage and completeness.
- 参考译文
 - 评审接口描述的完整性和覆盖度
- 原文释义与实施要点
 - 产品的接口包括产品构件之间的接口以及与产品集成环境的接口
 - 评审接口描述的完整性：
 - 有哪些类接口？
 - 每类的接口是否都描述了？
 - 是否所有的接口需求都实现了？
 - 是否所有的接口都遵循接口描述规范？
 - 如何测试这些接口？

- 业界案例
 - 在各阶段评审的检查单中设立接口评审的检查项检查接口的完整性。
 - 进行单独的接口技术评审检查接口的完整性。
 - 通过建立与维护接口与需求之间的跟踪矩阵来确保接口的完整性。
 - 制定接口描述的规范和准则。
 - 提供接口描述的最佳实践。

- 场景描述：
请将您手上的水笔拆分为4部分：笔帽；笔芯；笔套（后半）；笔套（前半）。
- 任务：
根据下面给出的场景，参考PI PA：
列举并描述这4部分之间的接口及接口规格

- 模型原文
 - Manage internal and external interface definitions, designs, and changes for products and product components
- 参考译文
 - 管理内部和外部的接口定义、设计和产品及产品构件变更
- 原文释义与实施要点
 - 接口需求驱动集成产品构件所必要的接口开发
 - 在开发产品的早期开始产品和产品构件接口的管理
 - 接口的定义和设计不仅影响产品构件和外部系统，而且也影响验证和确认环境
 - 接口的管理包括在整个产品的生命周期里，维护接口的一致性，以及解决接口的冲突、不符合和变更问题
 - 接口包括内部产品构件之间、内部与外部系统之间、产品与集成环境、验证及确认环境之间的接口
 - 来自供应商的采购的产品与其他产品或产品构件之间的接口的管理对于项目的成功至关重要

- 业界案例
 - 在开发的全生命周期管理接口，包括接口需求的管理和开发；接口需求的设计和实现；接口的评审；接口的变更控制等等。
 - 接口的相关产物纳入配置管理，并分配相应的权限以便于存取。
 - 接口的变更管理，走正式变更管理和发布的流程，由CCB控制。
 - 建立接口管理一览表，包括各个产品构件与外部环境的接口关系、不同产品构件之间的接口关系、产品构件内的接口关系。

- 模型原文

- Confirm, prior to assembly, that each product component required to assemble the product has been properly identified, behaves according to its description, and that the product-component interfaces comply with the interface descriptions.

- 参考译文

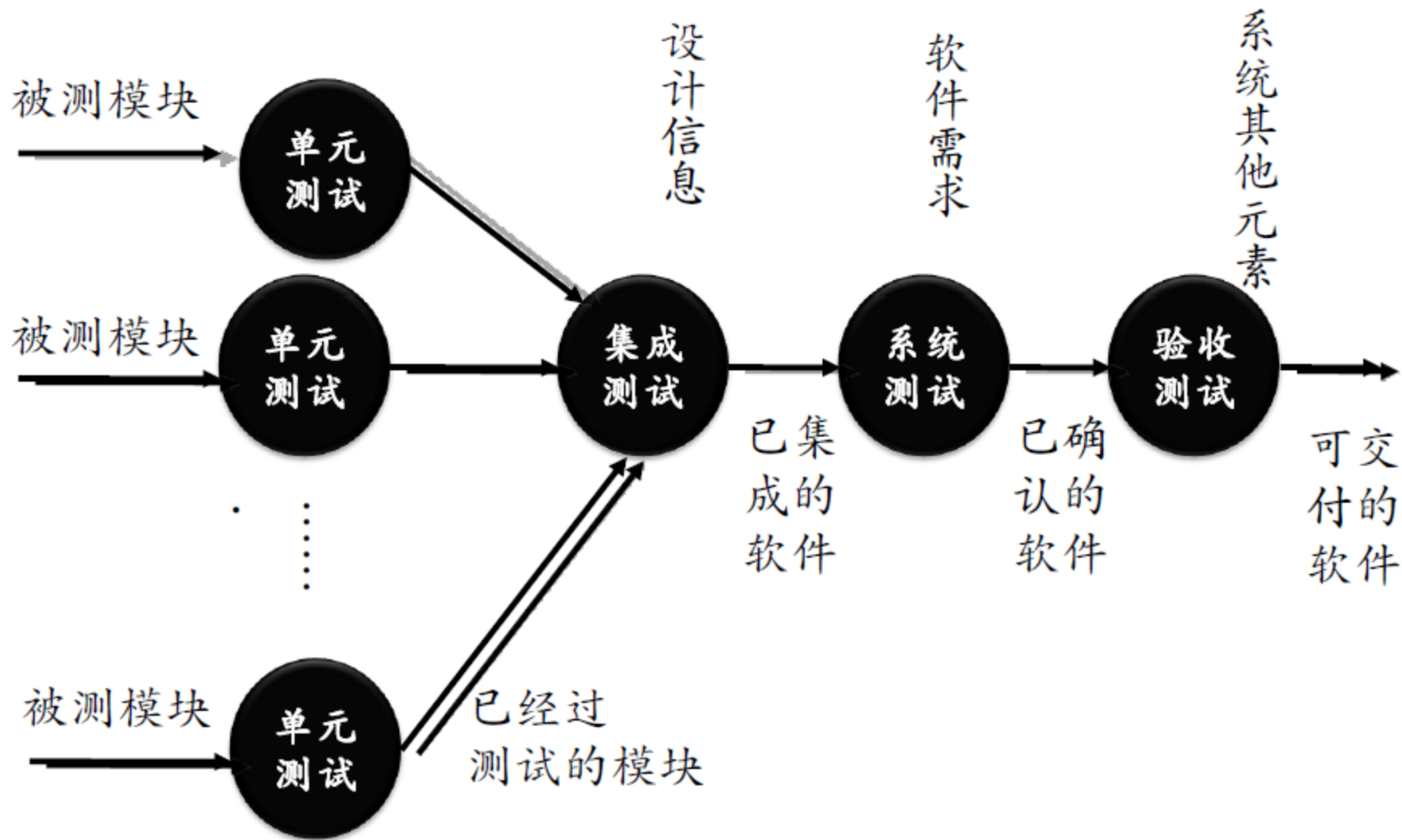
- 在进行组装之前，确认组装产品所需的每个产品构件已经恰当地标识、都能按照其描述正常运行，产品构件接口符合接口描述。

- 原文释义与实施要点
 - 恰当地标识：通过了配置审计。
 - 按照其描述正常运行：通过了单元测试。
 - 接口符合接口描述：通过了评审。
- 业界案例
 - 对于代码量较少的项目，集成测试前单元测试的语句覆盖率必须达到100%。
 - 无法100%单元测试的，采用单元测试和代码评审结合的方法，确保两者的覆盖率之和达到100%。
 - 相关人员根据集成测试就绪检查单检查产品构件的准备情况。
 - 设立集成测试前的里程碑评审确保产品构件的充分准备。

- 模型原文
 - Assemble product components according to the product integration sequence and available procedures.
- 参考译文
 - 遵循产品集成顺序和可用的规程组装产品构件
- 原文释义与实施要点
 - 一般在项目组中指定某个人负责集成产品
 - 应按照集成策略、集成规程、集成准则执行集成活动
 - 在实践中通常是：边集成，边测试
 - 集成的层次
 - 1个人负责的几个模块，这些模块之间的集成
 - 2个人负责的模块之间的集成
 - 1个小组于另1个小组之间的集成
 - 1个项目组与另1个项目组之间的集成

- 模型原文
 - Evaluate assembled product components for interface compatibility.
- 参考译文
评价组装的产品构件及接口
- 原文释义与实施要点
 - 执行集成测试
 - 在集成策略里定义集成测试结束的准则
- 业界案例
 - 产品集成与集成测试交替进行
 - 持续集成，代码编写一点就测试一点
 - 集成测试的对象：集成之后的构件接口测试、构件性能测试、功能测试、人机界面测试。
 - 对于复用或采购的构件，集成测试主要侧重接口测试
 - 可以选择系统测试的case作为集成测试的case。

单元测试、集成测试与系统测试的辨析



单元测试、集成测试与系统测试的辨析

名称	测试对象	侧重点	参照物	充分性的评价方法	时机	测试方法	测试执行者	投入工作量	发现的缺陷个数
单元测试	软件的最小单元，如函数、方法等	逻辑的正确性	详细设计、源程序	代码、分支等覆盖率	软件中的基本组成单位完成后，边开发边测试	白盒测试、动态测试	一般是开发人员		
集成测试	软件的模块、子系统	接口的正确性	概要设计、详细设计	接口覆盖率	软件系统集成过程中，边集成，边测试	黑盒测试、功能测试、白盒测试等	开发人员与测试人员		
系统测试	系统	需求的满足性	产品需求	用户场景覆盖率	系统开发完成后，交付客户之前	黑盒测试、功能测试、非功能测试等	测试人员		
验收测试	系统	需求的满足性	客户需求	需求覆盖率	交付客户后，正式投入使用之前	黑盒测试、功能测试、非功能测试等	客户		

练习：

- 场景描述：
请将您手上的水笔拆分为4部分：笔帽；笔芯；笔套（后半）；笔套（前半）。
- 任务：
根据下面给出的场景，参考PI PA：
列举对组装好的水笔可以从哪些方面进行评估

- 模型原文
 - Package the assembled product or product component and deliver it to the customer.
- 参考译文

对组装完成的产品或产品构件打包并把它交付给客户
- 原文释义与实施要点
 - 软件的打包、培训、安装、交付行为
 - 交付后客户要确认
 - 客户可能执行验收测试
 - 问题要记录并跟踪关闭
- 业界案例
 - 通过制作光盘交付产品
 - 通过FTP等共享机制交付产品
 - 制作安装视频指导交付产品的使用
 - 产品交付前执行里程碑评审进行全面检查
 - 产品交付前执行配置审计确保交付产物的一致性和完整性
 - 产品交付前执行部分的验收测试case检查交付物
 - 产品交付后客户需要签字确认

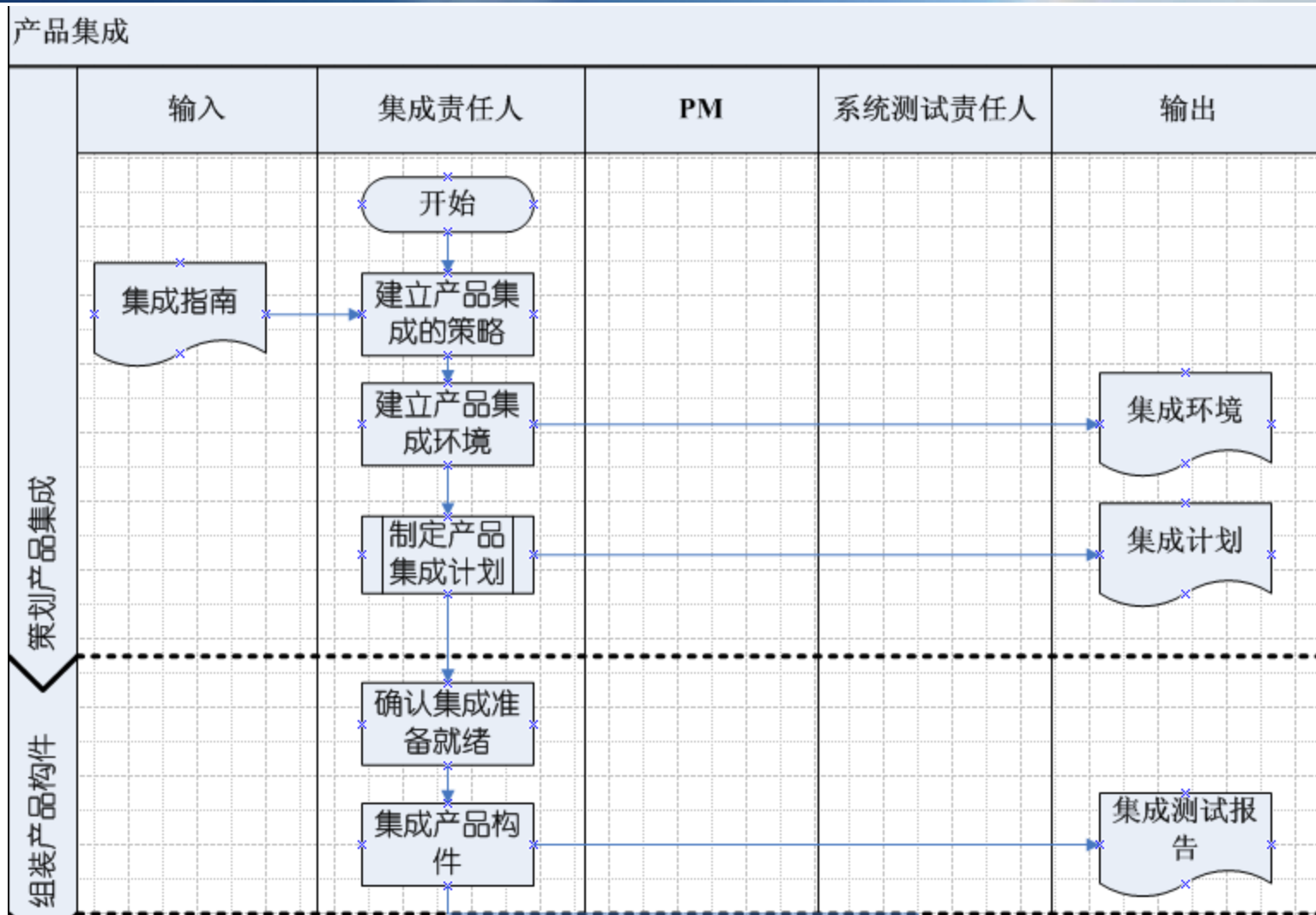
- 产品交付检查单：
 - 是否按照CM过程从基线库中生成产品？
 - 产品发布前是否执行了基线审计？
 - 媒介的标签是否正确，至少应显示出名称、发布日期和正确版本号？
 - 版本描述文档、手册是否同产品在一起？版本描述文档是否描述的是正确的版本？
 - 是否正式审查了版本描述文档，是否有相应的记录？
 - 是否正式审批了用户手册，是否有相应的记录？
 - 是否有一个交付物的分发列表？
 - 是否有具体的记录和处理分发问题的方法？是什么？
 - 分发问题是否由具体的人员处理？是谁？

总体的思路	实践	实践定义的要点	直接证据最小集
<p>1 可以分成2个过程进行定义： 集成与集成测试流程 产品打包交付流程 也可以只写一个流程。</p> <p>2 本过程的必须文档： 集成计划</p> <p>3 集成的方式包括了持续集成、阶段性集成、一次性集成多种方式</p>	SP1.1 确定产品集成顺序	在制定集成计划时，定义分几次集成、每次集成哪些模块、集成的环境要求	产品集成计划
	SP1.2 建立产品集成环境		对产品集成环境的定义 或者实际的产品集成环境
	SP1.3 建立产品集成规程和准则	要定义每次集成的进入与退出准则以及编写集成的脚本，确定集成的具体步骤与方法	产品集成计划
	SP2.1 评审接口描述的完整性	此实践在设计流程中可以体现	接口评审的记录（通常包含在设计评审报告中）
	SP2.2 管理接口	此实践在设计流程及变更流程中可以体现	接口的定义与描述
	SP3.1 确认集成用的产品构件已经准备就绪	执行了单元测试或代码走查、构件纳入了配置库	单元测试报告或代码走查报告
	SP3.2 组装产品构件	集成	产品
	SP3.3 评价组装的产品构件	集成测试	集成测试报告
	SP3.4 打包/包装并交付产品或产品构件		交付清单或验收报告

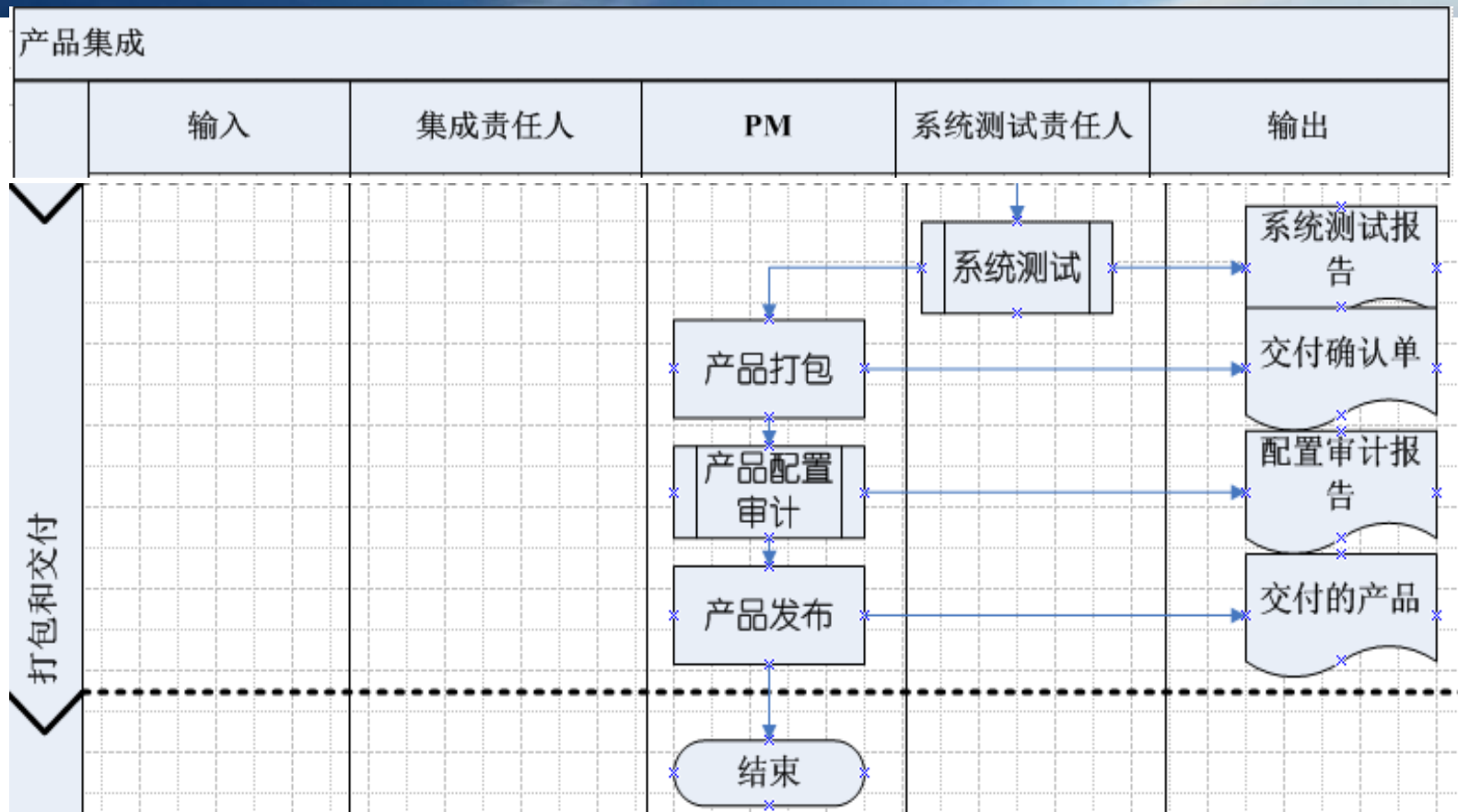
案例解析

- 制作产品集成计划、策略和规程
- 建立并维护产品集成环境
- 执行集成
- 集成测试
- 打包并交付产品和产品构件

产品集成工作流程-1



产品集成工作流程-2



PI工作流程

1. 建立产品集成的策略

集成负责人根据概要设计说明书和产品集成指南，识别待集成的产品构件和接口，确定集成策略、集成的次数、集成的步骤、集成的进入和推出准则等。

2. 建立产品集成环境

集成负责人确定集成的环境，包括软硬件的详细配置和描述。产品集成环境可以通过自己开发获得，也可由已有环境获得。

3. 制定产品集成计划

集成负责人将上述内容文档化为《产品集成计划》，产品集成计划需要通过项目组的评审。

1. 确认集成准备就绪

集成负责人对产品构件进行检查，确认产品构件是否满足集成的进入准则，集成的最小进入准则为产品构件通过了单元测试。

2. 集成产品构件

当满足产品集成进入准则时，集成负责人依据《产品集成计划》进行集成，并执行集成测试以评价其集成工作，主要包括：

- 集成之后的构件接口测试。
- 集成之后的基本功能测试。

集成负责人将集成测试通过后的代码形成测试基线，制作《集成测试报告》并提交给项目组，准备进行系统测试。

1. 执行系统测试

测试组执行系统测试，参见系统测试过程定义。

2. 产品打包

系统测试通过后，项目组对软件产品进行分类打包，并填写《项目交付确认单》。

3. 产品配置审计

项目经理根据《项目交付确认单》组织对交付产品进行配置审计，审计其完整性及一致性，并签字确认。

4. 产品发布

《项目交付确认单》经各方审计确认后，配置管理员从配置库中标识发布基线，并提取发布基线的软件产品，根据需要制作产品介质。

交付物提交客户后，由客户签字确认。

Q&A
谢谢!