第三讲 可信软件过程构建

高可信软件工程

目录

3.1 可信软件过程

软件过程与软件生命周期

- 软件过程（Software Process）是指一套关于项目的阶段、状态、方法、技术和开发、维护软件的人员以及相关Artifacts（计划、文档、模型、编码、测试、手册等）组成。目前有三种方法：UP（the unified process），The OPEN Process，OOSP(TheObject-Oriented Software Process)。
- 软件过程是指软件整个生命周期，从需求获取，需求分析，设计，实现，测试，发布和维护一个过程模型。一个软件过程定义了软件开发中采用的方法，软件过程还包含该过程中应用的技术——技术方法和自动化工具。过程定义一个框架，为有效交付软件工程技术，这个框架必须创建。软件过程构成了软件项目管理控制的基础，并且创建了一个环境以便于技术方法的采用、工作产品（模型、文档、报告、表格等）的产生、里程碑的创建、质量的保证、正常变更的正确管理。
- 产品的质量源于生产产品的过程质量，可信软件过程是指：合理地配置软件开发的各类资源，在充分理解用户需求的基础上，准确地进行软件分析、软件设计、软件测试及软件实施，最终开发出符合用户期望的软件产品并形成有效过程资源的软件开发过程。

3.1 可信软件过程

3.1.1 可信软件过程管理复杂性

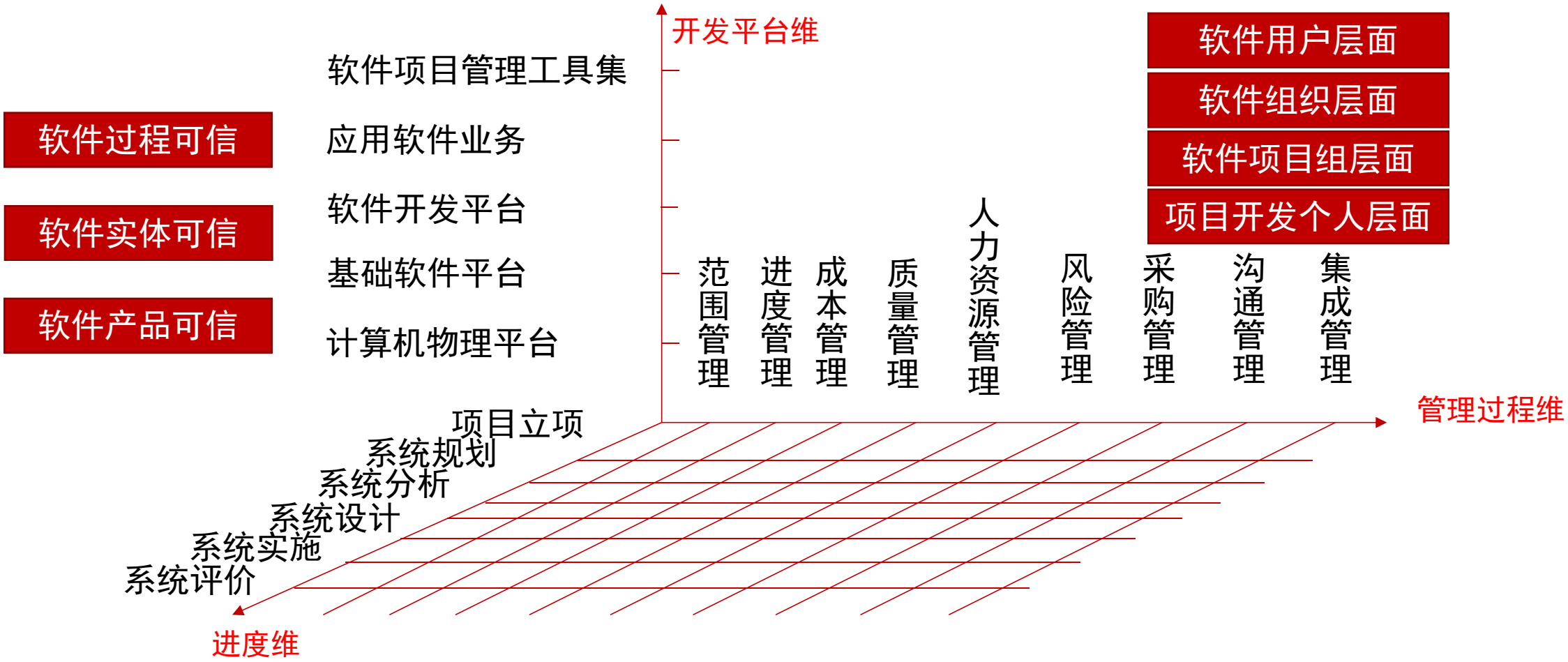
3.1.2 软件开发过程与管理过程

3.1.3 可信软件过程内涵

3.1 可信软件过程

3.1.1 可信软件过程管理复杂性 -- 软件项目开发过程三维模型

可信软件过程管理是一项系统工程，涉及进度维、管理过程维和开发平台维三个维度。



3.1 可信软件过程

3.1.1 可信软件过程管理复杂性 -- 软件项目开发过程三维模型

项目进度维

- 即软件项目生命周期，包括项目立项、系统分析、系统设计和系统运维等7个阶段。在项目开发中，各个阶段和可以设置若干里程碑事件。

软件项目管理过程维

- 根据PMBOK关于项目管理的9个知识领域，包括项目的范围、成本、进度、质量、人力资源、风险、采购、沟通、集成管理9个环节，其中集成管理是协调其它8个项目知识领域的综合管理过程。

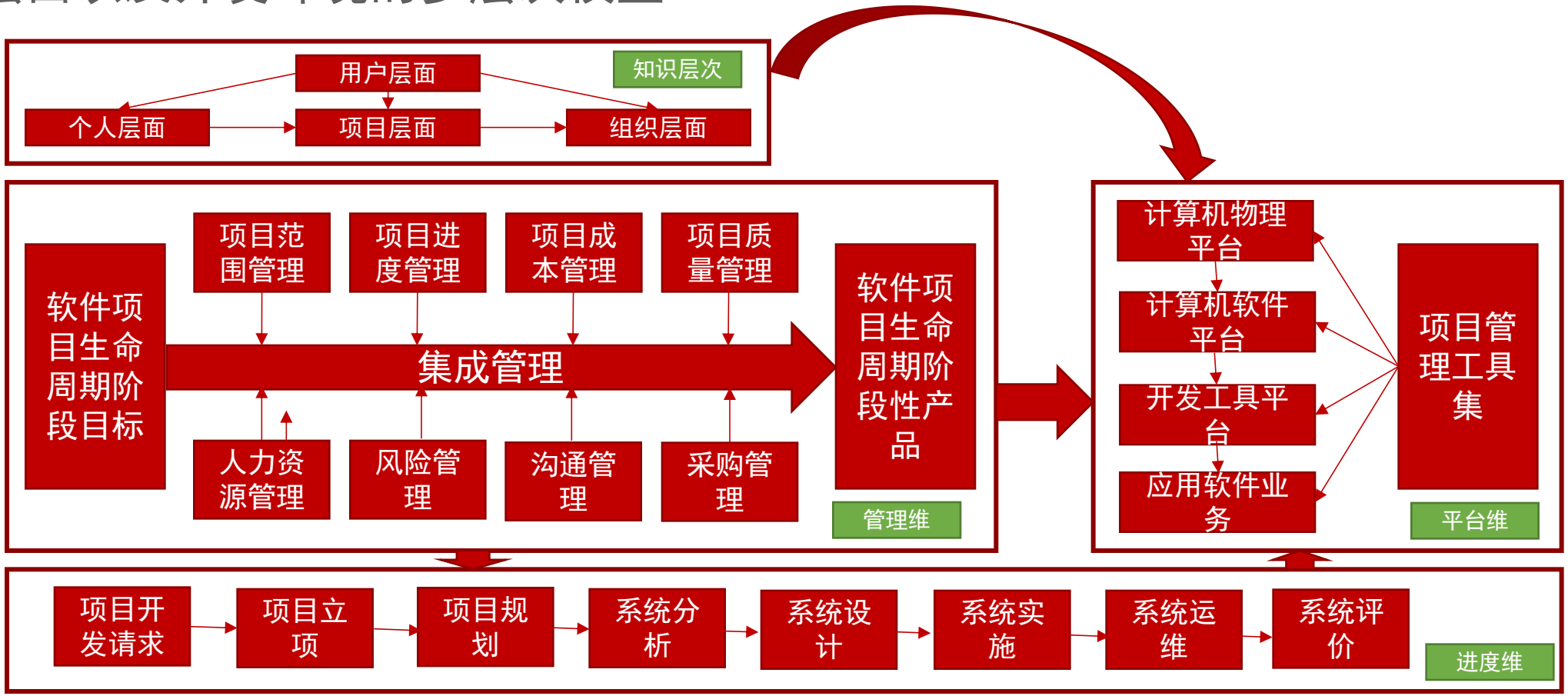
开发平台维

- 包括计算机物理平台、基础软件平台、软件开发平台、软件项目管理工具集等。

3.1 可信软件过程

3.1.1 可信软件过程管理复杂性 -- 软件项目开发过程三维模型

可信软件过程涉及项目管理层面、项目开发个人层面、软件组织层面、用户层面以及开发环境的多层次模型。



3.1 可信软件过程

3.1.1 可信软件过程管理复杂性 – 软件开发过程的主要活动

不同的软件生命周期模型、软件开发过程的阶段划分以及开发方法有所不同，但包含的主要工程和管理活动基本类似，以结构化软件开发方法为例，软件开发过程主要有阶段划分、主要开发活动、各个活动点的特点、活动或里程碑事件结果。

项目生命周期阶段	项目招投标阶段			软件项目开发过程阶段			软件运维阶段	
	项目开发请求	项目立项	项目规划	系统分析	系统设计	系统实施	系统运维	系统评价
主要开发活动	项目相关业务、软硬件、系统评价等	招投标、立项依据，可行性分析	规划方法、功能规划、数据分类等	需求分析。业务流程分析、数据综合查询分析等	功能结构设计、流程设计、代码设计、数据库设计等	数据库、开发平台、程序设计、软件组件、系统测试等	软件验收、鉴定、运维管理等	性能评价、经济评价、综合评价等
活动的特点	非结构化、不规范	定量知识和不定量知识结合	系统宏观规划、定义系统间关系	各部分知识逻辑性较强	系统分析的体现、并具有依赖性的物理模型	逻辑模型的计算机实现	具有较强综合性和随机性	定性和定量
活动结果表现形式	文档	文档	文档	文档、逻辑模型	文档、物理模型	文档、程序源代码	文档、活动文件	文档、测试报告
实例	调研报告	可行性分析方法	项目规划方法	系统数据流图	代码设计方法	测试用例，软件组件	运维报告，历时较长	系统评价方法

3.1 可信软件过程

3.1.2 软件开发过程与管理过程

软件产品的生产过程包括项目的开发过程和项目的管理过程

项目的开发过程

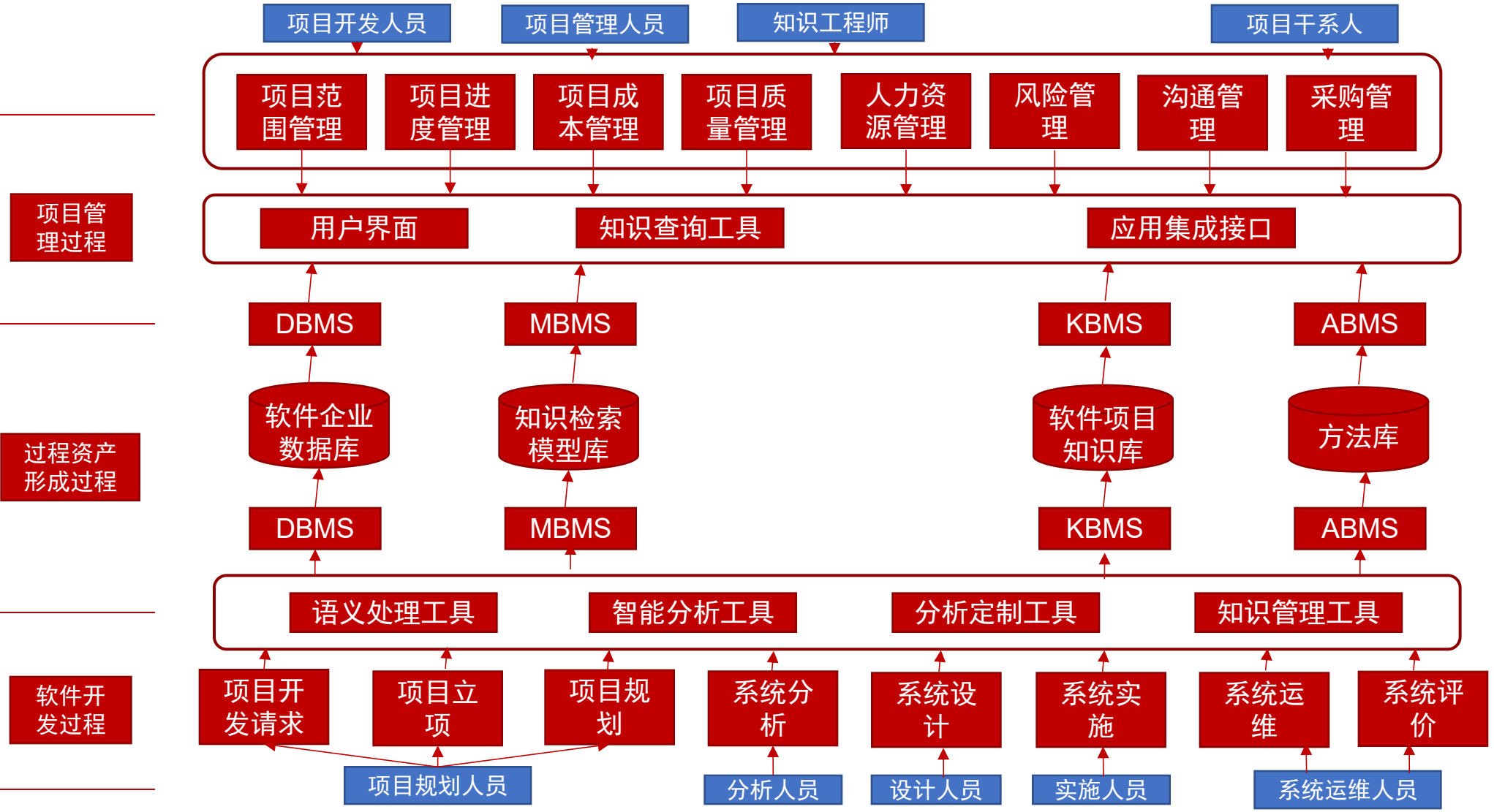
- 项目的开发过程即软件工程的过程（项目的规划、分析、设计、实施等）。工程过程一般将软件开发结构分解成较小的、相对独立的、易于项目团队安排的活动，工程过程的实施一坨软件开发平台

项目的管理过程

- 项目管理过程是对软件工程过程实施进行范围管理、成本管理、进度管理和质量管理等九个知识领域的管理活动计划的安排和考核

3.1 可信软件过程

3.1.2 软件开发过程与管理过程 – 开发过程应与管理过程协调统一

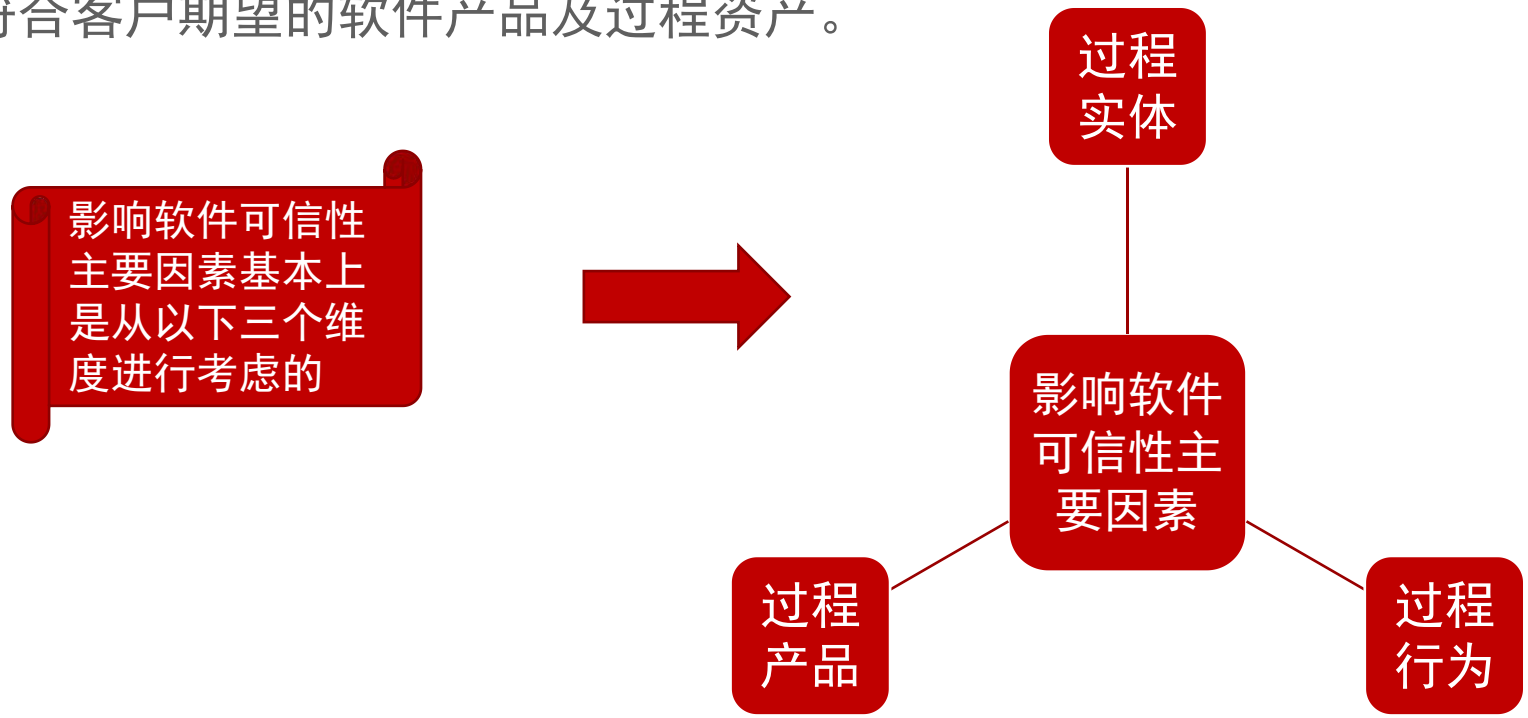


3.1 可信软件过程

3.1.3 可信软件过程内涵及软件过程构成

软件质量源于其生产过程。软件过程能力是指通过执行一个过程达到期望结果的水平，能够产生出一些列可心工作产品的有能力过程，也是生产出满足需求的可心工作产品的信心度，是衡量过程可信的主要因素之一。

目前，CMMI作为在全世界软件企业得到广泛应用的集成软件过程管理模型，提供了覆盖软件开发生命周期所有过程的22个PA（process area），指导软件组织管理和控制软件过程以期得到符合客户期望的软件产品及过程资产。



3.1 可信软件过程

3.1.3 可信软件过程内涵 – 软件过程构成

过程实体

- 过程实体就是负责执行过程的软件开发技术人员和管理人员以及过程资产。
- 相关人员应该有资格或者接受过各种的培训以保证他们有必备的技能、知识以及专业程度。
- 如果不称职人员执行软件过程会导致：提交有缺陷的产品；成本超支、进度延期；留下安全风险；客户满意度下降。

过程行为

- 过程行为是实体行为的体现，是过程活动计划、度量、监控、评审、评价以及执行过程的种种方式。
- 过程行为直接影响过程的可信性。
- 过程行为不可信的危害：无法或难以达到预期效果；过程系一部无法进行；过程行为可信直接影响产品可信。

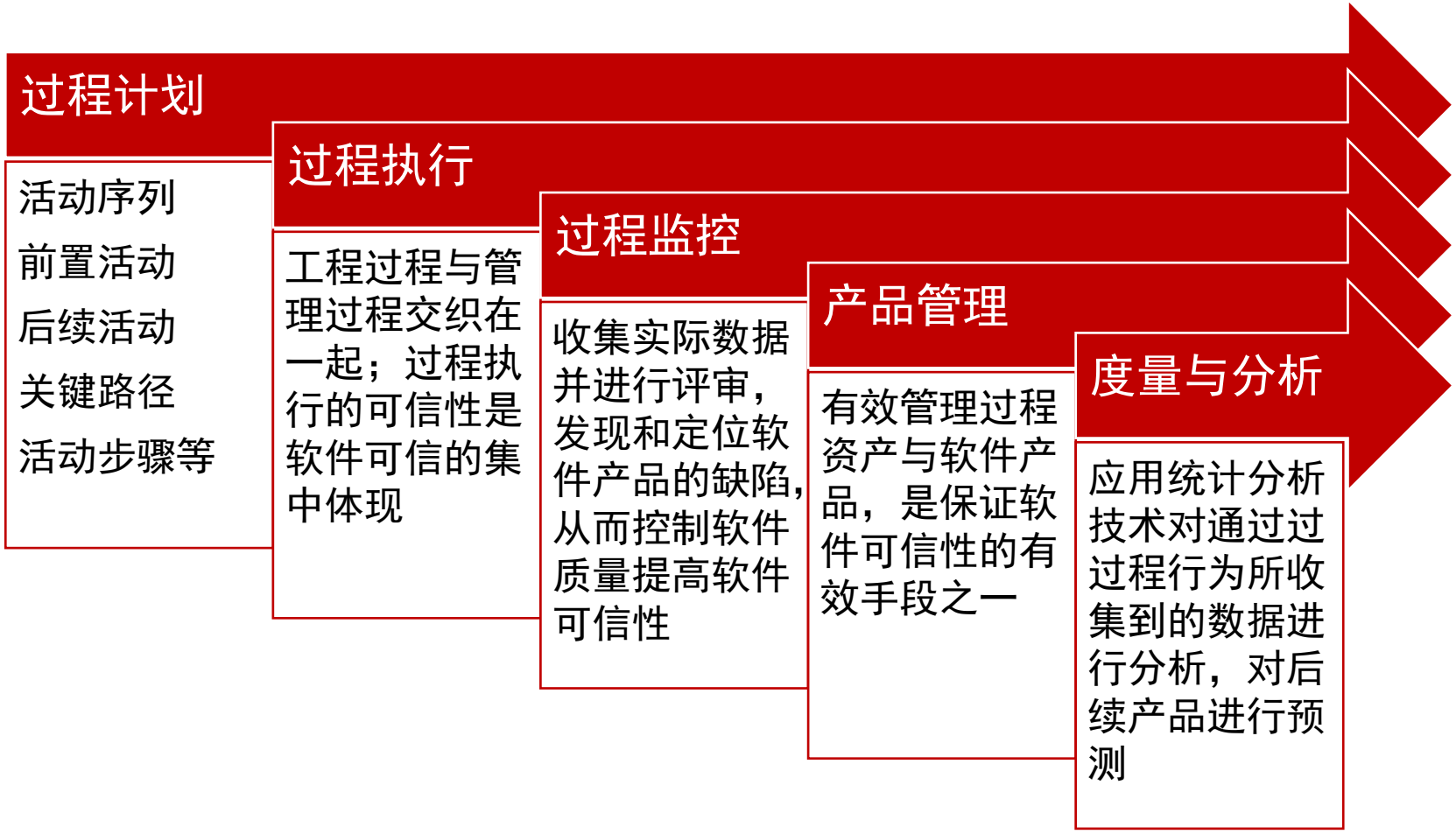
过程产品

- 过程产品就是在开发过程中产生的制品，他可能是一个工作产品，也可能是一个组件。
- 软件产品来源于工作产品并与组件的集成，任何一个不值得信赖的工作产品或组件都将导致交付的产品不可信。

3.1 可信软件过程

3.1.3 可信软件过程内涵 – 软件过程的主要活动

从CMMI涵盖的过程、应用过程和全面质量管理的角度来看，可信软件过程应包含以下阶段：



3.2 可信软件过程建立

3.2.1 可信过程准则

3.2.2 过程可信性模型

3.2.3 可信软件过程管理框架

3.2.4 可信软件过程管理框架下的可信保障过程域

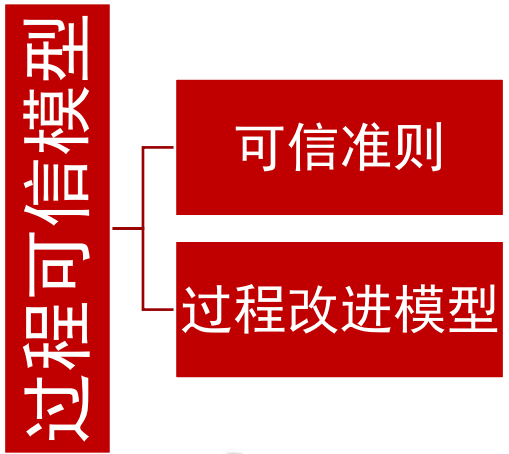
3.2 可信软件过程建立

- 软件过程对产品的可信性影响很大。当计划和管理一个可信过程时，需要从可信过程实体、可信过程行为、可信过程产品三个维度加强过程的管理方式来保证可信过程的构建。

3.2 可信软件过程建立

3.2.1 可信过程准则

- 过程可信模型包括过程可信准则和过程改进模型两部分。
- 过程可信准则指软件开发过程应遵循的一般原则，包括软件开发过程及运行环境的指导性原则：**开发过程可信原则包括需求可跟踪性、设计规约、源码评审等；运行环境可信原则如权限控制、可信路径等、**
- **可信软件方法学（trusted software methodology，TSM）给出了过程可信的详尽可信原则。**
- 过程改进模型是软件组织定义、构建、实施、度量、控制和改进其软件开发过程各类活动的指南。CMMI是业界和学术界公认的较为完善的软件过程改进模型。



3.2 可信软件过程建立

3.2.1 可信过程准则 – TSM

- TSM方法学把软件可信定义为“软件满足既定需求的信心度”，强调软件可信性对贯彻软件开发过程开发生命周期的各类活动的管理决策、技术决策以及既定需求集合的高度依赖型。
- 同时，TSM从软件开发过程的角度提出了供软件开发参考的44条可信原则以及对应需求，在此基础上TSM定义了6个可信级别及相应的评估模型。

3.2 可信软件过程建立

3.2.1 可信过程准则 – TSM

T0	T1	T2	T3	T4	T5		
						审计	确定的软件生命周期活动的记录应该由软件工程环境自动的登记和存储在受保护的存放处
						权限控制	根据清晰定义的安全策略，所有确定的软件生命周期活动应该被软件工程环境自动控制
						可信路径	软件工程环境应该包括一个明确的机制来保证生命周期的活动不会被未授权的方法截获
						识别和授权	一个确定的软件生命周期活动的初始应该有被软件工程环境识别并授权后的人来完成
						配置管理	应建立起一个配置管理系统，系统中应包括关于配置项识别、审核、控制和审计的明确机制和步骤
						环境完整性	对于识别软件工程环境组件的变更应该有一个明确的步骤，如果有需要，恢复环境的完整性
						可信分配	所用的软件从原点转变到目标应该以一个能保证传遍完整性的方式完成

3.2 可信软件过程建立

3.2.1 可信过程准则 – TSM

T0	T1	T2	T3	T4	T5		
						入侵检测	审计跟踪数据应该用来做对软件工程环境周期性获随即的入侵检测
						管理	根据管理文档，由有资格的人员对软件工程环境，软件工具和开发的软件进行维护
						环境和工具选择	软件工程环境和所有的软件工具应当根据一个明确的选择策略来进行选择，选择策略中应该考虑可信等级、成熟度、文档和源码的可获取性等因素
						最小特权	执行生命周期活动的特权应该被分配并维护以保证特权仅分配给有需求的人
						多人控制	生命周期活动的执行需要至少两个或者两个以上有资格的开发人员的认同和参与
						安全政策	所有软件开发者执行开发活动应该遵守明确定义和增强的安全策略
						共享知识	每个软件开发活动组件，包括需求、源码、设计、测试、软件工具、方法和支撑活动等都应该与至少两个人员相关，这些人员应该非常熟悉这些组件的细节、隐含的意义和所考虑的选择方案

3.2 可信软件过程建立

3.2.1 可信过程准则 – TSM

T0	T1	T2	T3	T4	T5		
						软件重用	所有重用的软件应该接受选择，清晰的选择政策应该考虑可信等级、成熟度、文档和可获取的源码
						计划	对于所有软件开发活动的详细设计应该在软件开发计划书中描述，软件开发的管理也应该遵循计划书中所描述的方法
						风险移除	与软件开发活动相关的潜在风险都应该被明确的识别，风险移除策略应该被文档化
						需求分析工具	使用需求分析CASE工具以支持需求规约，一致性检查和文档生成
						需求分析评审	由一个同行评审组对需求分析进行同行评审以保证软件需求分析的完整性，一致性和正确性
						需求分析文档	除了软件需求规约说明书和接口规约说明书，所有帮助理解需求分析过程，重要的需求分析决策的原理等有用的信息都应该被文档化
						形式化需求规约	除了非形式化的需求规约说明以外，需求文档应用一个形式化的框架来规约

3.2 可信软件过程建立

3.2.1 可信过程准则 – TSM

T0	T1	T2	T3	T4	T5		
						需求可跟踪性	对于明确的系统需求或客户端来源，所有的软件需求应保持可跟踪性，并且所有分配给计算机软件配置项的系统需求也应该对于软件需求而言是可跟踪的
						原型方法	所有作为风险移除策略一部分的原型方法应该按照明确的原型计划来执行，在原型计划中应该包括原型设计、开发、测试、文档化和被保护的方式
						原型软件的重用	当原型软件在开发软件中重用时，原型软件应该被充分地文档化、评审和测试以保证可信等级与开发的软件是一致的
						设计工具	在设计中应采用CASE工具以维护设计/需求的跟踪映射关系，并生成设计文档
						设计评审	由一个同行评审组对设计进行同行评审以保证软件设计的完整性，一致性和正确性
						设计文档	除了软件设计说明书和接口设计说明书外，设计活动的特性，考虑到的重要的设计选择项和重要设计理由都应该被文档化
						形式化设计规约	除了非形式化的设计说明以外，设计文档应用一个形式化的框架来完成

3.2 可信软件过程建立

3.2.1 可信过程准则 – TSM

T0	T1	T2	T3	T4	T5		
						设计可跟踪性	设计的各方面和需求应该是互相跟踪的
						源码标准	一个明确定义的源码标准应该在源码活动中被使用
						源码分析	应该使用度量复杂度和风格的工具和步骤来分析所有开发代码
						源码评审	由一个同行评审组对源码进行同行评审以保证软件源码和计算机软件单元测试的完整性、一致性和正确性
						源码文档	源码和软件编码活动的特征应该被文档化
						源码可跟踪性	所有的源码应该对于设计和计算机软件单元测试是可跟踪的，设计对于源码同样如此
						测试策略	所有测试单元、组件和配置项的测试和集成任务都应该包括各种测试策略的规定
						测试职责	软件组件和配置项的测试职责应该交给没有参与被测试的编码和设计活动的独立小组或组织

3.2 可信软件过程建立

3.2.1 可信过程准则 – TSM

T0	T1	T2	T3	T4	T5		
						可靠性度量	计算机软件组件和计算机软件配置项的测试和领域结果应该用来将观测到软件失败率降低到一个可以接受的程度
						测试工具	软件工程环境应该可以包含一个创造、执行、文化和分析测试完整性的测试床
						测试评审	由一个同行评审组对测试进行同行评审以保证软件测试的完整性、一致性和正确性
						测试文档	除了软件计划测试书、软件测试描述书、软件测试报告以外，软件组件和配置项测试活动的特征也应该被文档化
						测试可跟踪性	所有软件组件和配置项的测试对于需求是可跟踪的，源码和需求对于组件和配置项的测试同样如此
						形式化设计验证	形式化的设计满足它的需求规约
						形式化源码验证	形式化源码验证是证明它的源码满足它的需求和设计

3.2 可信软件过程建立

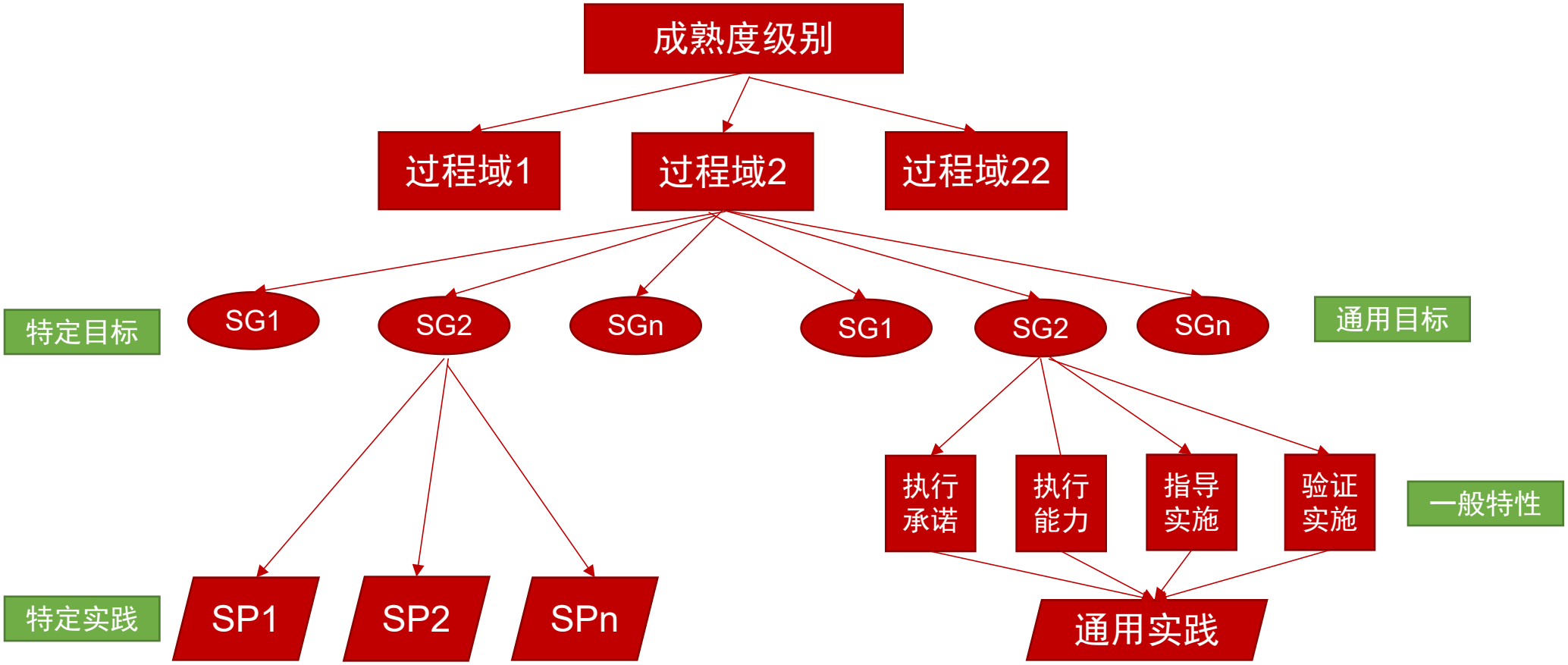
3.2.2 过程可信性模型

- 为了集成TSM中的可信原则、可信度和评价方法，将CMMI关于过程域的定义进行了扩展。
- 过程域是一个软件过程相关实践的聚类，执行这些实践将满足一系列对改善该过程域起到重要作用的目标。
- CMMI模型分为阶段模型和连续模型。

3.2 可信软件过程建立

3.2.2 过程可信性模型 – CMMI阶段模型

- 阶段模型侧重软件组织的整体能力成熟度水平的测度



阶段式模型

1) 阶段式模型

阶段式模型基本沿袭SW-CMM模型框架，仍保持五个“成熟度等级”，但过程域做了一些调整和扩充，如下表 所示：

成熟度等级	过程域
L2可重复级	需求管理　项目计划　配置管理 项目监督和控制　供应商合同管理 度量和分析　过程和产品质量保证
L3已定义级	需求开发　技术解决方案　产品集成 验证　确认　组织级过程焦点 组织级过程定义　组织级培训 集成化项目管理　风险管理　集成化的团队 决策分析和解决方　组织级集成环境
L4 已管理级	组织级过程性能　项目定量管理
L5 优化级	组织级改革和实施　因果分析和解决方案

3.2 可信软件过程建立

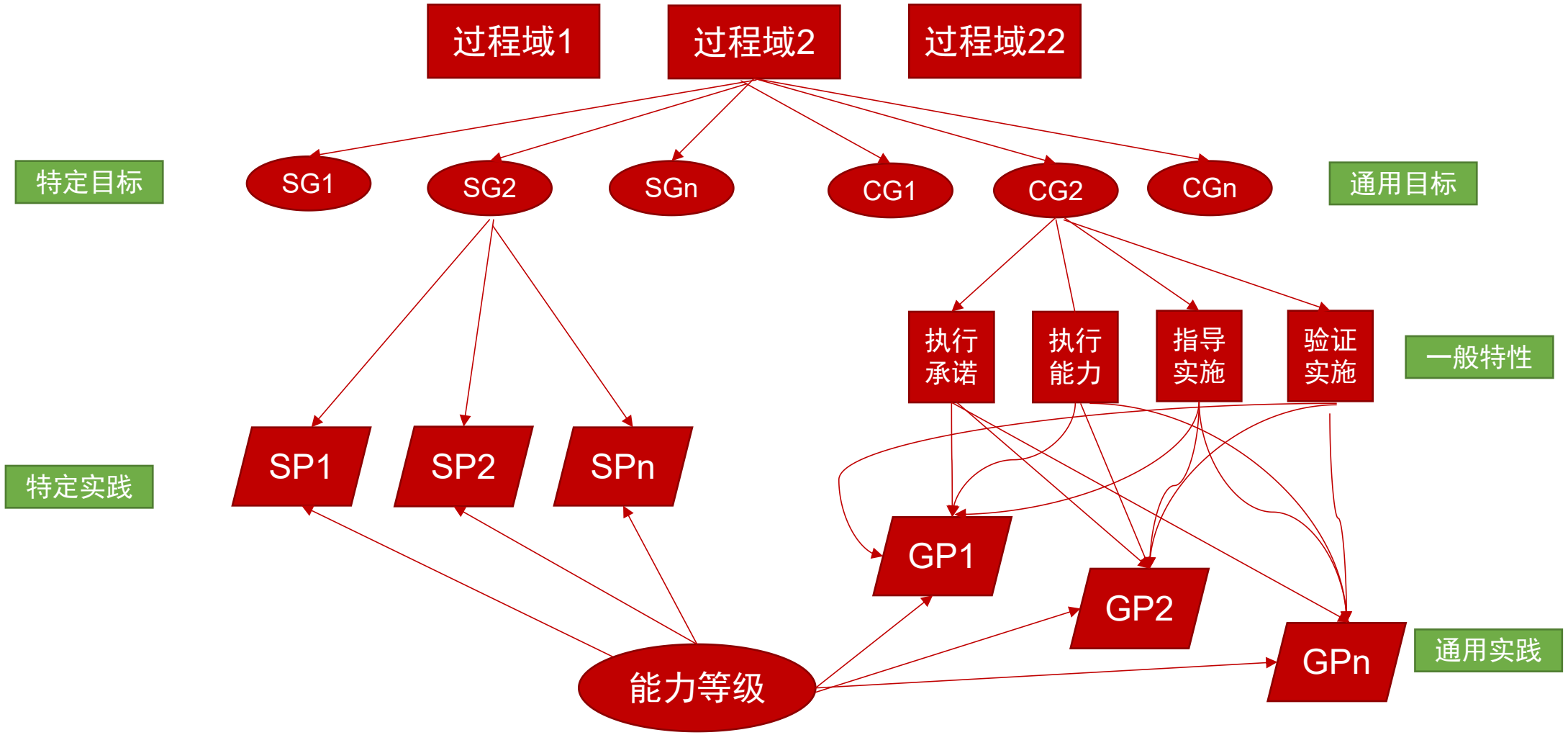
3.2.2 过程可信性模型 – CMMI连续模型

- 连续模型充分考虑不同类型软件组织的过程改进需要和能力成熟度水平评估的需要，针对组织需要的重点过程进行有针对性的能力成熟度水平评估，达到软件组织资源高效配置的目的
- 连续模型根据各过程域的特征将所有过程域按照功能划分为四个过程：

工程过程，过程管理过程、项目管理过程、支持过程。

3.2 可信软件过程建立

3.2.2 过程可信性模型 – CMMI连续模型



连续式模型

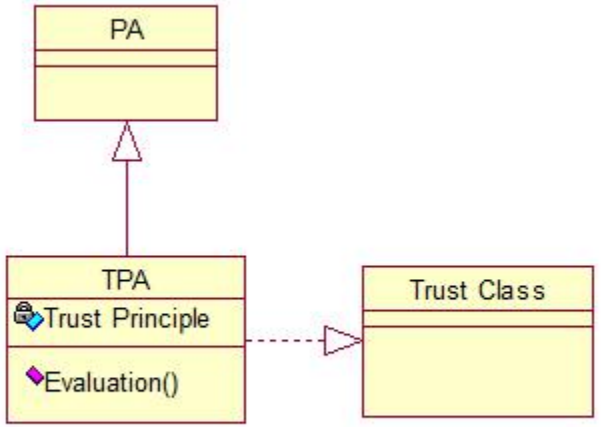
- 连续式模型没有与组织成熟度相关的几个阶段。
- 连续式模型将24个过程域按照功能划分为过程管理、项目管理、工程、支持 四个过程组。

连续式分组	过程域		
过程管理	组织级过程焦点	组织级过程定义	
	组织级培训	组织级过程性能	
	组织级改革和实施		
项目管理	项目计划	项目监督和控制	
	供应商合同管理	集成化项目管理	
	风险管理	集成化的团队	项目定量管理
工 程	需求管理	需求开发	技术解决方案
	产品集成	验证	确认
支 持	配置管理	度量和分析	过程和产品质量保证
	决策分析和解决方案		组织级集成环境
	因果分析和解决方案		

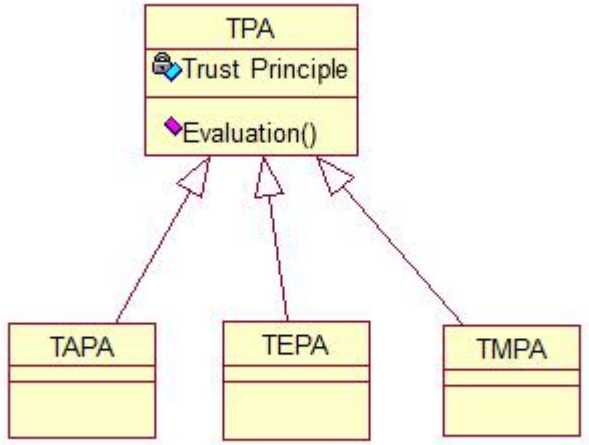
3.2 可信软件过程建立

3.2.2 过程可信性模型 – 可信软件过程域

- 对TSM的可信原则（trust principle，TP）与CCMI中过程域进行了映射，每个确定的PA上都附上一些列相关的可信原则并扩展到一个相对应的可信过程域（trustworthy process area，TPAs）。创建新的TPA，将不能映射到任何CMMI过程域上的可信原则包含进来。



TPA的结构



TPA的扩展结构：包括过程实体、过程行为和过程产品

3.2 可信软件过程建立

3.2.2 过程可信性模型 – 可信软件过程域

可信保障过程域

- 可信保障过程域(trusted assurance process area，TAPA)是为了保证在软件开发项目中，尤其在分布式网络开发环境下，合适的软件过程实体能够按照合适的步骤执行特定软件实践
- 包括：配置管理、过程和质量保证、度量和分析、决策分析和决议、原因分析和决议、供应商协议管理、集成项目管理、组织培训、组织过程性能和组织创新部署

可信监控过程域

- 将CMMI中的管理和支撑过程被扩展到可信监控过程域（TMPAs）中
- 包括项目监督和控制、风险管理、量化项目管理、项目计划、组织过程焦点和组织过程定义

可信工程过程域

- 可信工程过程域（TEPAs）的提出是为了包含更加严格的活动过程以保证工作产品满足可信需求
- 包括需求管理、需求开发、技术方案、产品集成、验证和确认六个过程域

可信过程域	CMMI过程域	备注
可信工程过程域	需求管理	CMMI工程过程域
	需求开发	
	技术方案	
	产品集成	
	验证	
	确认	
可信保障过程域	配置管理	CMMI支持过程域
	过程和质量保证	
	质量和分析	
	决策分析和决议	
	原因分析和决议	
	供应商协议管理	CMMI项目管理过程域
	集成项目管理	
	组织培训	CMMI过程管理过程
	组织过程性能	
	组织创新部署	
可信监控过程域	项目监督和控制	CMMI项目管理过程域
	风险管理	
	量化项目管理	
	项目计划	
	组织过程焦点	CMMI过程管理过程域
	组织过程定义	

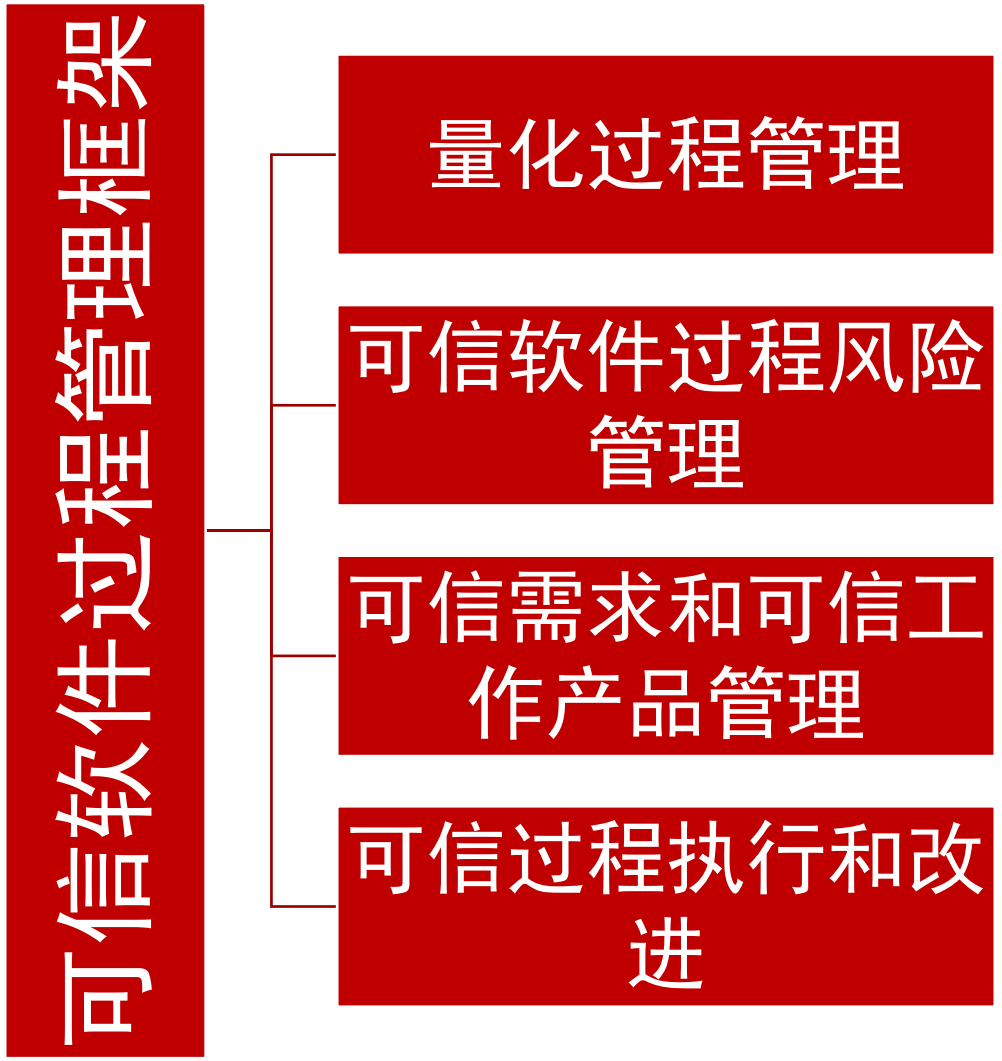
3.2 可信软件过程建立

3.2.3 可信软件过程管理框架

- 可信软件过程建立的基础是通用软件过程已经被定义和管理，并且与特定的TPA（如TPAs、TMPAs、TEPAs）进行了恰当的绑定和部署。因此，过程可信模型可以用来对这些部署和评估提供指南。
- 参考一些与信息安全可信或安全相关的过程标准和模型，继承了可信原则、可信等级和可信实践，扩展了CMMI应用。
- 此外，借鉴了TSM的思想开发一系列可信等级定义、因子和评价标准，以使得过程可信模型支持特定应用领域的可信评估。
- 建立TPA是计划和加强可信软件开发的基础。面向过程的可信管理框架通过以下四个保障特性来保证软件可信性。

3.2 可信软件过程建立

3.2.3 可信软件过程管理框架



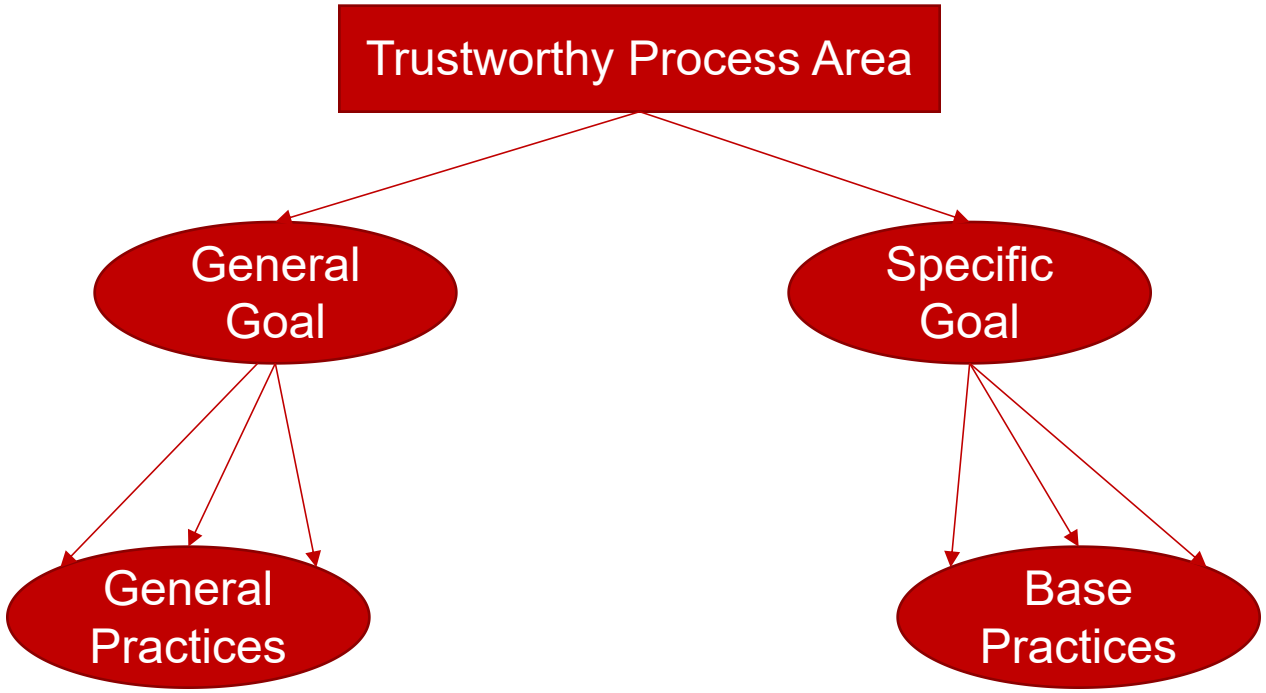
3.2 可信软件过程建立

3.2.4 可信软件过程管理框架下的可信保障过程域

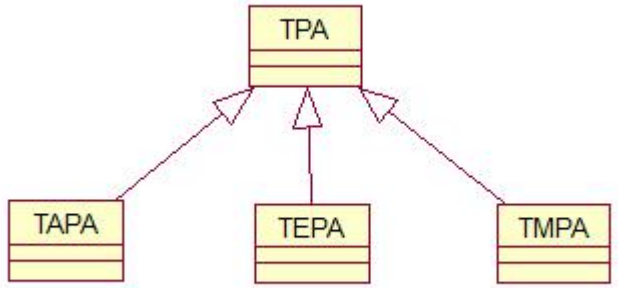
- 为了满足过程实体可信、过程行为可信和过程产品可信的三维可信要求，可信保障过程域在扩展现有的软件过程改进模型或者框架基础上，提供更为正确而可信的活动和实践来保证软件可信性目标。
- 在软件开发过程中有着各种各样并行的和交互的活动与实践。CMMI模型将其分为四类，这些过程都有着不同的成熟度级别，不同类型的过程相互支持，反映了组织的软件过程能力成熟度。
- 在可信软件开发过程中，产品质量方面的可信性也是在开发过程中通过每一个活动和实践逐步得以获取，一步步演化并最终确保可信的。这些保证可信的活动和实践得以抽象和归结到可信过程域（trustworthy process area，TPAs），这个属于扩展自CMMI的过程域，如下图（a）所示。
- 与过程能力成熟度类似，软件过程可信性也是一个有着多重级别的概念，如可信级0~5。不同的软件过程可信性级别要求不同的可信过程域的集合，如下图（b） 所示。

3.2 可信软件过程建立

3.2.4 可信软件过程管理框架下的可信保障过程域



图（a）



图（b）

3.2 可信软件过程建立

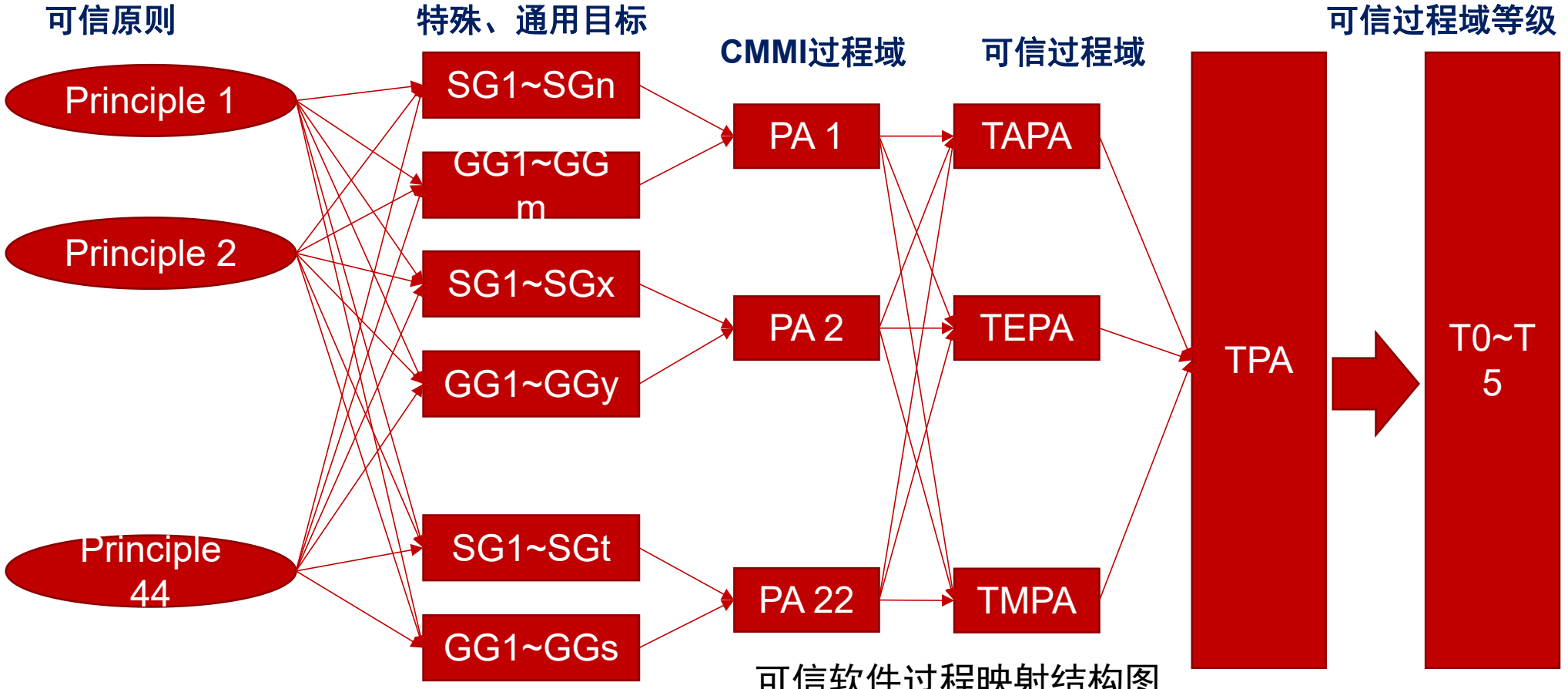
3.2.4 可信软件过程管理框架下的可信保障过程域

- 为了建立可信软件过程模型和度量模型，以现有的模型或是标准作为基础，将可信过程与可信原则的映射，建立一个可信过程模型，为组织开发可信软件提供实践指导，并为软件过程可信度分级提供评价基础。
- TSM中44个可信原则，涵盖软件开发过程的各个方面，在充分考虑过程行为、过程产品和过程实体软件过程三个维度的具体要求，根据每个过程域的属性特征映射一系列相关的可信原则并扩展到一个相对应的可信过程域。为评价软件过程的可信度，定义了6个可信级别（T0~T5），每个可信级别对不同的可信原则有不同的要求，形成可信保障过程域、可信监控过程域和可信工程过程域，再根据TSM基本原理确定软件过程等级。

3.2 可信软件过程建立

3.2.4 可信软件过程管理框架下的可信保障过程域

- 通过将TSM中的44个可信原则（principle1 principle）和CMMI过程域特殊目标和通用目标（SG/GG）中特殊实践（SP）和通用实践（CP）进行映射，并为每对映射的SP与TP列出一系列的基于可信原则的度量指标，度量该实践活动是否满足该可信原则，从而将TSM可信原则、可信分级与评估方法和可信过程域TPA集成起来，最终建立可信软件的过程模型与度量模型，以及可信软件分级模型。



可信软件过程映射结构图

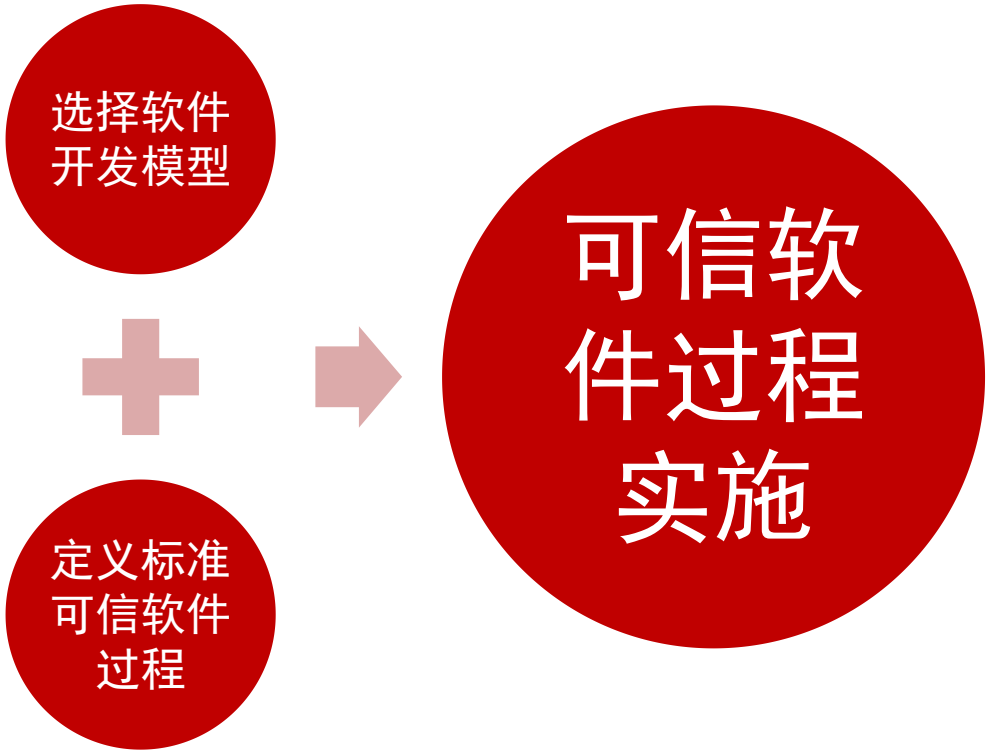
3.3 可信软件过程实施

- 3.3.1 可信软件过程定义
- 3.3.2 可信软件过程剪裁
- 3.3.3 可信软件过程执行
- 3.3.4 可信软件过程度量
- 3.3.5 可信软件过程改进

3.3 可信软件过程实施

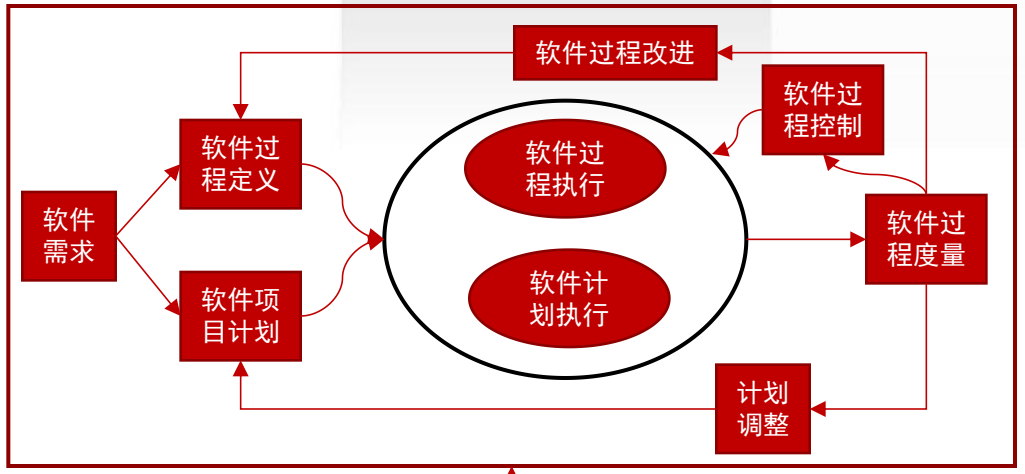
3.3.1 可信软件过程定义

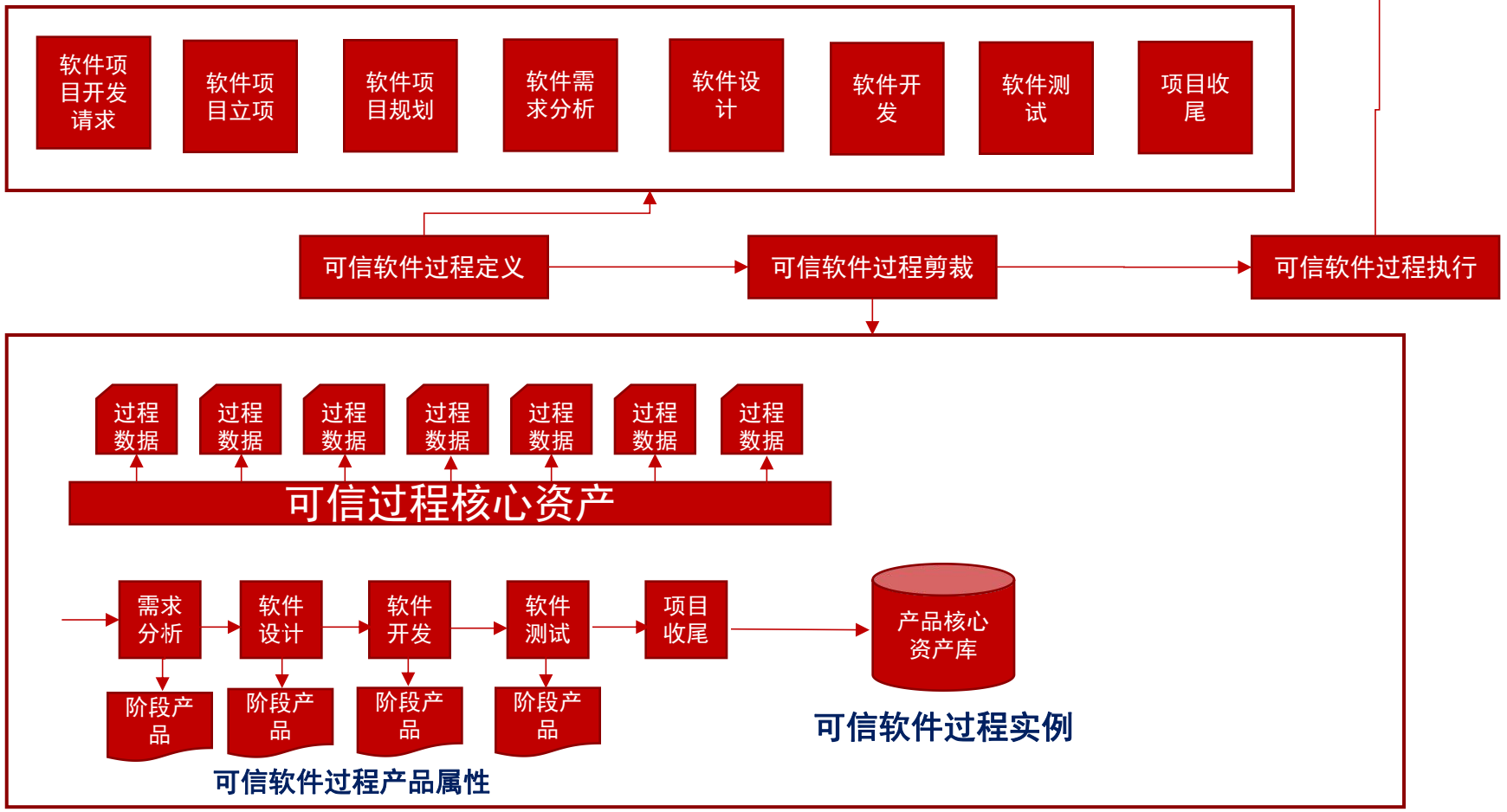
- 可信软件过程是基于可信原则和能力成熟度模型集成的过程域的最大集合，包括软件开发过程管理的方方面面，对于特定的软件组织和特定的软件开发项目，组织应根据不同软件开发模型进行必要的可信过程定义。



3.3 可信软件过程实施

3.3.1 可信软件过程定义





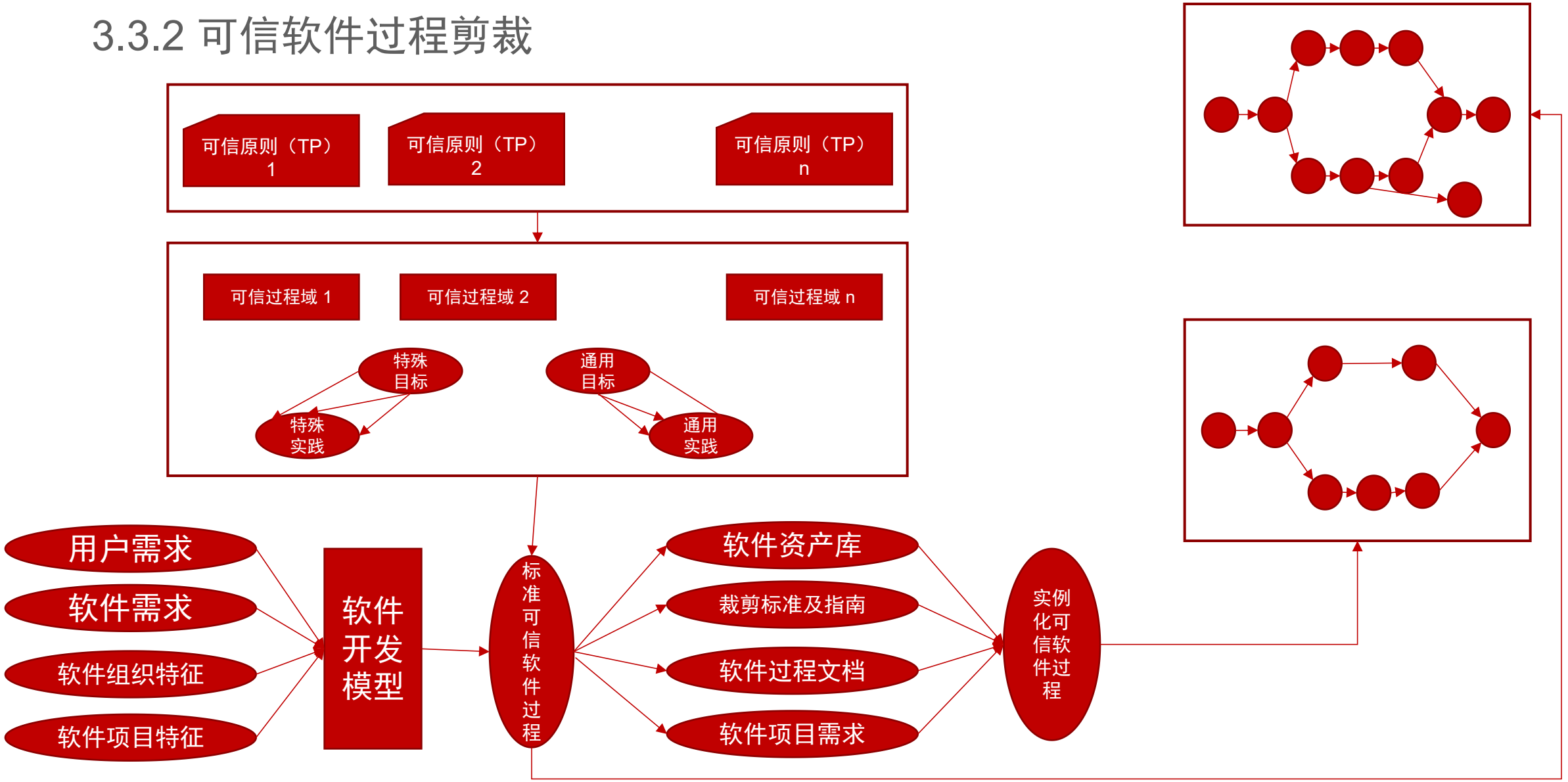
3.3 可信软件过程实施

3.3.2 可信软件过程剪裁

- 可信标准软件过程是在具体的软件开发模型基础上定义的适合同类项目开发的综合过程，由于同类项目存在差异，所以在开发过程中为确保软件具有高可信性和可操作性，软件组织应根据用户需求、软件需求、不同软件项目特征、组织自身特点和技术成熟度确定软件开发模型，根据软件开发模型选择标准可信软件过程模型，根据组织现有的过程资产库、软件项目需求以及软件过程裁剪标注指南确定项目开发需要的实例化可信软件过程。

3.3 可信软件过程实施

3.3.2 可信软件过程剪裁



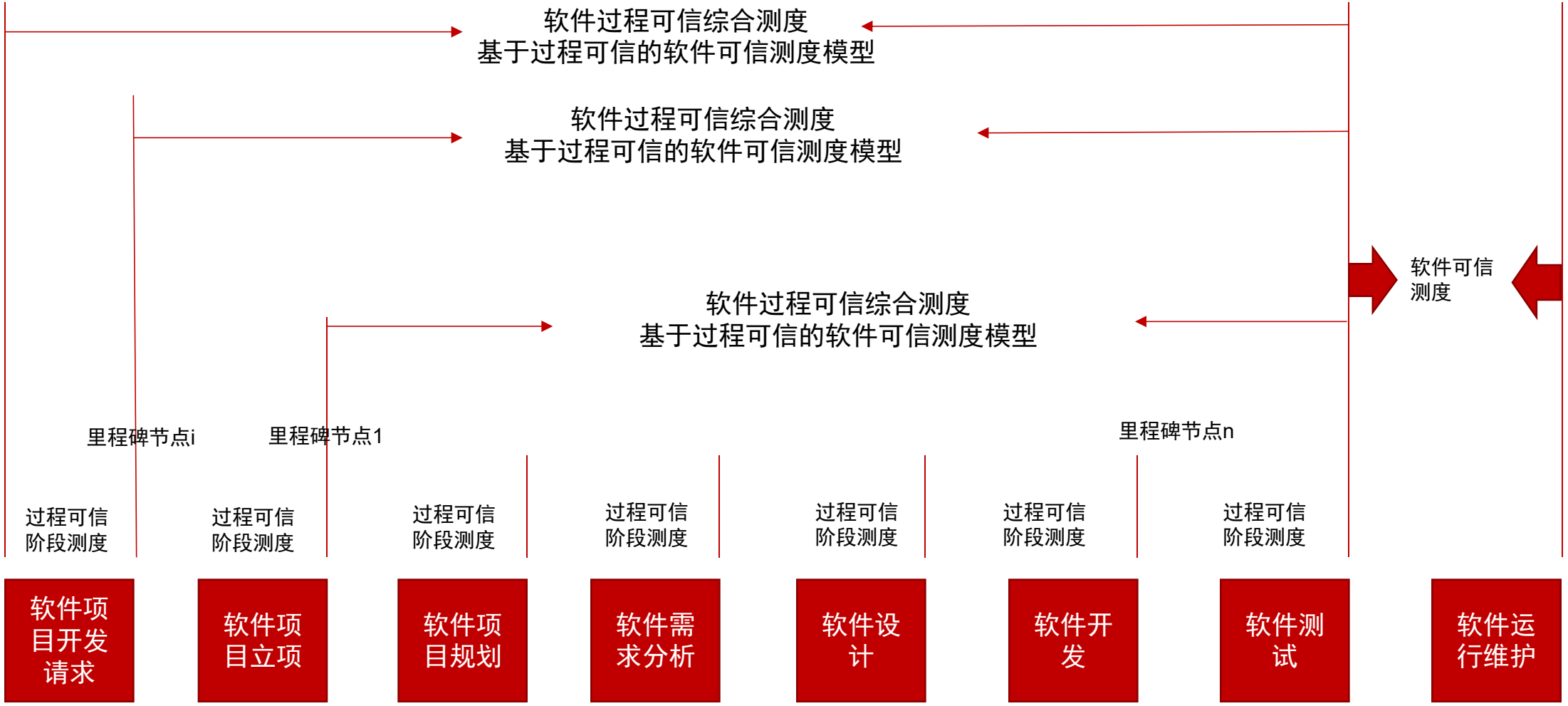
3.3 可信软件过程实施

3.3.3 可信软件过程执行

- 标准软件过程模型是软件开发的通用模型或子过程模型。软件组织在针对某一特定项目需要明细过程中更详细的过程信息，如可用资源、过程资产、可重用工作产品等。
- 软件过程是由一些列相互制约的活动序列构成，随着软件项目开发过程的推进，软件的不确定性逐渐明晰，因此，需要对当前生命周期阶段的后续阶段的可信度进行重新测度，并重新制定软件过程改进策略。
- 软件组织应建立动态软件可信度测度和过程改进机制，当前生命周期阶段或里程碑事件结束后，应重新测度后续过程的可信度并识别过程的风险。
- 采用迭代式的软件过程可测信度。

3.3 可信软件过程实施

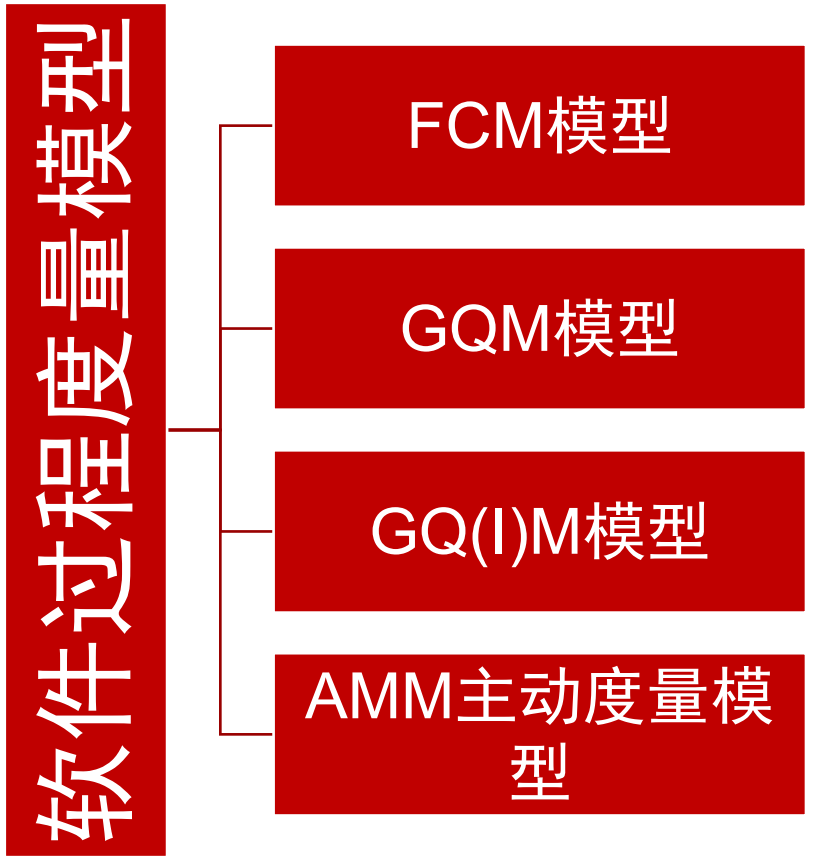
3.3.3 可信软件过程执行



3.3 可信软件过程实施

3.3.4 可信软件过程度量

- 1958年Rubey和Hurtwick提出了软件度量概念，旨在通过有效度量科学的评价软件过程质量，加强对软件开发过程控制和管理，合理地组织和分配资源，制订切实可行的软件开发计划，达到以较低成本获得高质量软件的目的。
- Boehm与1976年提出了软件质量度量的层次模型，认为对软件属性不能仅有定性的研究，还必须有定量研究。
- McCall等于1977年将软件质量分解至能够度量的层次，提出FCM（factor criteria metric）模型。
- 1984年美国教授Victor R. Basili教授提出基于目标的GQM模型。
- 2002年Wolfhart Woethert将GQM模型进一步细化为GQ(I)M（goal question indicator measure）。



3.3 可信软件过程实施

3.3.4 可信软件过程度量

- FCM模型

通过一种分层结构建立面向用户的质量要求与面向软件属性的准则和度量之间的关系，通过对软件属性的度量来反映软件质量特性。但是，不同软件的质量要素不同，则评价准则不同，所以，该模型有局限性。

- GQM模型

是一种基于目标的自上而下的度量定义方法，属目标驱动的度量模型。包括Goal、Question、和Metric三个层次。缺乏过程中间的问题，无法综合地反映过程能力成熟度。

- GQ（1)M模型

在GQM的基础上增加了指示层，该模型根据问题直接确定度量和数据元素，有助于达到目标的可度量性和相关指示器有机协调。缺乏反映过程的整体情况。

- AMM主动性度量模型

是一种多层次、多维度过程度量模型，包括许多过程属性和指标，可以分析过程及其产品是否可信。

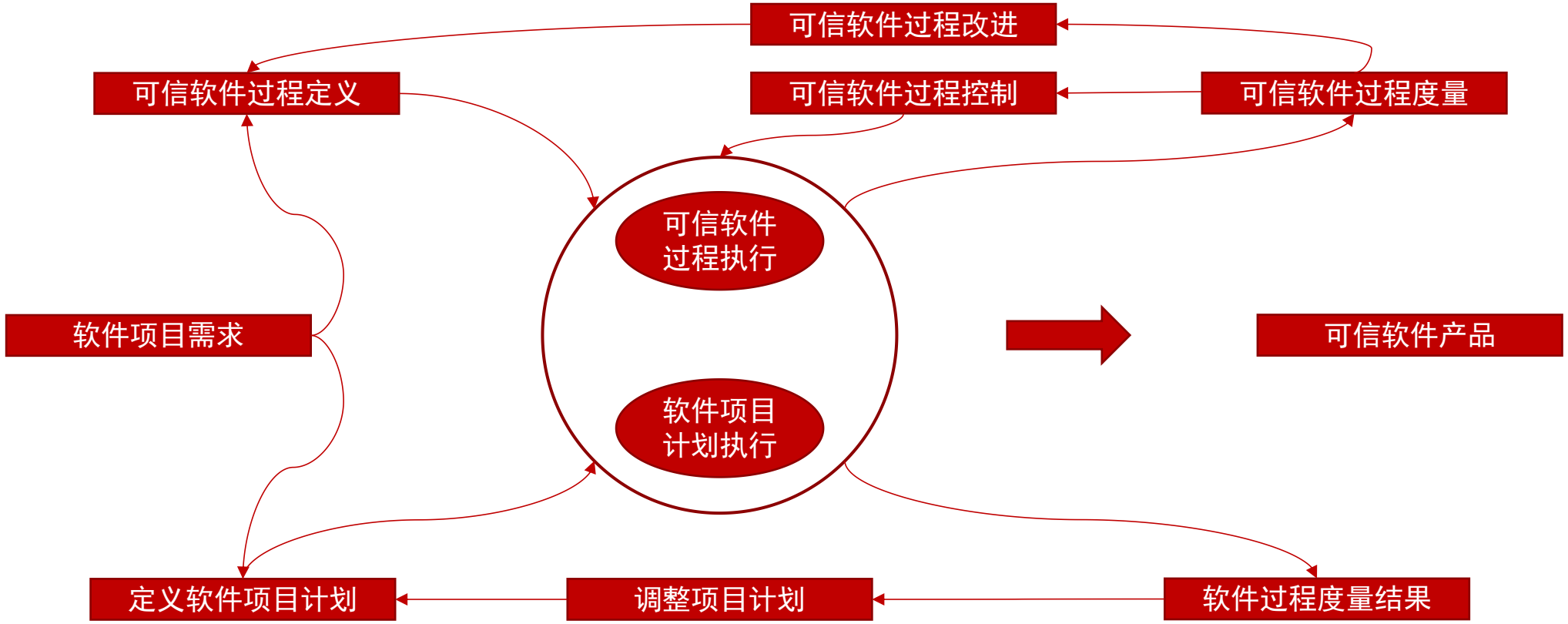
- 软件过程度量的相关标准

GB/T16260-2006、ISO/IEC9126

3.3 可信软件过程实施

3.3.5 可信软件过程改进

- 软件过程是软件组织最复杂和最重要的业务流程。可信软件过程改进（software process improvement， SPI）是根据组织实际情况以及组织选定的过程改进模型和改进方法，指导软件组织对其软件工程过程和项目管理过程进行改进。



可信软件过程改进流程

谢谢大家！