

国家 863 重点项目

“高可信软件生产工具与集成环境”技术文档



软件生产线框架规范

Software Production Line Framework Specification

(TRUSTIE-SPL V 2.0)

2009 年 9 月 30 日

发布声明

牵头单位：中国科学院软件研究所

参研单位：国防科技大学计算机学院

北京大学信息科学技术学院

北京航空航天大学计算机学院

中创软件公司

执笔人：高楚舒、王伟、白琳、窦文生、张若定

版本号：2.0

发布时间：2009年9月30日

审核人：王怀民、魏峻

版权声明

本技术规范得到国家 863 重点项目“高可信软件生产工具及集成环境”第一课题“可信的国家软件资源共享与协同生产环境”的资助，版权归“可信的国家软件资源共享与协同生产环境”课题组所有。

本规范在以下条件下可以自由传播：

- (1) 引用需声明；
- (2) 保持本规范的完整性（包括发布声明）；
- (2) 未经课题组许可，任何人不得出版、发行本规范内容。

Trustie 技术框架规范

变更记录

变更版本	日期	A/M/D	原因与修改情况描述	修订人	审核人
V1.0	2009.1.15	A	制定《规范》	高楚舒	
V1.1	2009.5.26	M	对《规范》进行大的结构变更	张若定	
V1.2	2009.6.26	M	结构调整及内容更新	白琳	
V1.3	2009.7.24	M	内容修订	王伟	
V2.0	2009.9.30	M	内容修订以及结构调整	王伟、窦文生、白琳	

注：A - 增加 M - 修改 D - 删节

目 录

第 1 章	引言	1
1.1	背景.....	1
1.2	目的.....	3
1.3	范围.....	3
1.4	读者对象.....	3
1.5	组织结构.....	4
第 2 章	术语和定义	5
2.1	软件生产线.....	5
2.2	软件工具.....	5
2.3	软件制品.....	5
2.4	软件生产线框架.....	5
第 3 章	软件生产线及其组成要素	6
3.1	软件生产线.....	6
3.2	软件生产线组成要素.....	6
3.2.1	软件生产线使用人员.....	7
3.2.2	软件工具.....	7
3.2.3	软件制品.....	8
第 4 章	软件生产线框架	9
4.1	软件生产线元模型.....	9
4.2	软件生产线设计.....	10
4.3	软件生产线装配.....	11
4.4	软件生产线使用.....	13
第 5 章	软件生产线框架参考实现	14
5.1	微内核.....	14
5.2	生产线建模与配置工具.....	14
5.3	生命周期管理.....	15
5.4	工具库.....	15
5.5	任务管理.....	16
5.6	制品管理.....	16
5.7	通信管理.....	16
5.8	个人工作台.....	16
	参考文献	18

第1章 引言

1.1 背景

软件工具是一类用来辅助计算机软件开发、运行、维护、管理、支持等过程中的活动或任务的软件。使用软件工具能节省软件生产开销，提高软件生产率 and 产品质量。随着软件结构的日益复杂，为细化软件开发的分工，开发过程被划分为多个阶段，例如需求分析，设计建模，代码编写，部署监控等阶段。软件工具也被进一步分类，出现了针对各个阶段不同粒度的开发工具，例如 UML 建模工具，调试工具等。众多纷繁的开发工具给用户在选择和集成上都带来了许多问题，软件生产环境应运而生。软件生产环境是一个复杂的系统，包括了用来辅助计算机软件开发、运行、维护、管理、支持等过程中的活动或任务的众多软件。

纵观软件生产环境的发展，可以把它们总结为五个发展阶段：

- 1945-1960 时期，直接运行在裸机上的专用开发工具
- 1960-1980 时期，基于命令行的开发环境 (Command Line Environments , CLE)
- 1980-2000 时期，集成开发环境(Integrated Development Environments , IDE)
- 2000-2004 时期，扩展开发环境(eXtended Development Environments, XDE)
- 2005 年开始兴起的协同开发环境(Collaborative Development Environments, CDE)

最早的软件开发环境就是一种集代码编写、调试、运行等工具于一体的集成开发环境。集成开发环境提供的特性主要聚焦于以语法为导向的编辑器扩展，用于增强用户的开发过程体验。典型的代表为 Smalltalk、第一个 PC 图形化 IDE 工具 Visual Basic、支持增量编译的 Rational 开发环境，微软的 Visual Studio、IBM 的 WebSphere Studio 以及 Sun 的 Studio One 也属于此类。

“封闭的”集成开发环境无法满足不断出现的新开发工具的集成需求。于是，软件开发环境发展到一个“开放的”扩展开发环境阶段，这种开发环境基于可扩

展的体系结构，支持工具的灵活集成。XDE 不仅支持时间上的松耦合开发，也支持空间上的分布开发，并且开始考虑对非软件依赖因素的管理。扩展开发环境提供的特性包括重构支持、需求变更管理、配置管理、建模等。典型的代表为：Eclipse[2]、Rational XDE[3]。其中，Eclipse 是当前开源社区中应用最为广泛的 XDE 平台，它确定了一整套工具接口和模型规范，作为集成框架，可以支持众多的工具集成为一个整体。

软件开发不仅仅是单纯的编码，它是团队开发者相互合作来设计解决方案，构建高质量代码的协同过程，因此以协同开发思想为基础的协同开发环境成为发展趋势。协同开发环境通过整合协同工具，实现无缝的、完整的软件开发。协同开发环境的使用者扩展到了目标软件的行业领域专家、网络与硬件工程师等，使得软件产品相关的所有利益相关方均能配置其感兴趣的属性，在互动的软件开发协作过程中及时沟通并做出对策。协同开发环境提供的特性包括虚拟团队、即时通讯、网络会议、讨论等协同机制。典型的代表为 Collabnet 的 SourceForge[4]、SourceCast[5]等。目前这些协同开发环境提供的功能主要在代码协同方面，包括配置管理、冲突检测等。

软件生产线框架规范是《可信的国家软件资源共享与协同生产环境》课题的重要组成部分。此课题是国家 863 计划“高可信软件生产工具及集成环境”重点项目的主体和核心，旨在突破关键性的高可信软件生产技术、可信软件资源共享支撑技术、软件可信度分级机制以及软件开发协同技术等基础技术；研制覆盖软件开发主要活动和任务的软件工具，建立国家级软件资源库和面向不同技术特征的若干软件生产线，提供开发协同平台，建立“可信的国家软件资源共享与协同生产环境”，以及配套的、可持续发展的运行机制，形成指导本项目其他课题发布软件资源、集成软件工具、开展协同开发的技术规范，从而为构建高可信软件生产工具及集成环境提供核心基础设施[1]。其中，高可信的软件生产平台和集成环境体系结构如图 1-1 所示。

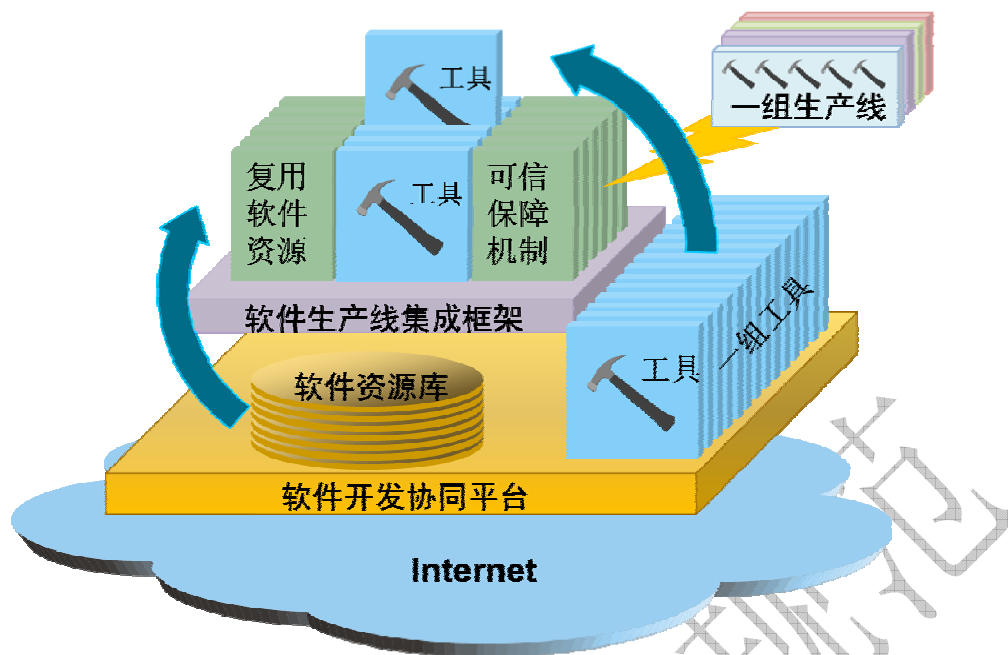


图 1-1 高可信的软件生产平台和集成环境体系结构图

该课题以可信软件生产活动为中心，研究可信软件生产的基本特点，针对工具的集成化、可扩展、协同化开发需求，研究灵活可扩展的软件工具集成框架、生产线数据集成方法和机制，实现生产线与各类工具、软件资源库和开发协同平台之间的集成方法和交互机制，支持生产线的灵活构造，进而形成软件工具与生产线构造的系列国家行业标准，规范软件工具、制品的集成和共享能力[1]。

本规范是“高可信的软件生产平台和集成环境”项目的重要组成部分。

1.2 目的

本规范的目的是提出一种软件生产线元模型，并给出生产线框架的参考实现，指导软件生产线的构造以及基于软件生产线的软件开发实践。

1.3 范围

本规范定义了软件生产线及其组成要素的含义、属性特征，定义和描述了软件生产线元模型，提出了基于元模型的软件生产线框架及其参考实现。

1.4 读者对象

本规范读者对象设定为国家 863 计划“高可信软件生产工具及集成环境”项目组成员，以及其他从事软件生产环境构造、工具集成等工作的研究或技术人员。

1.5 组织结构

本规范按以下方式组织：

第 2 章定义软件生产线的相关概念。

第 3 章介绍软件生产线、软件生产线组成元素及其属性特征；

第 4 章提出软件生产线框架，重点描述软件生产线元模型中所包含的实体以及实体间的关系；

第 5 章提出软件生产线框架参考实现，重点介绍生产线框架的体系结构及其主要功能模块。

Trustie 技术规范

第2章 术语和定义

2.1 软件生产线

软件生产线是按照一定的软件开发方法,将软件开发过程中涉及的人、工具、制品等要素有序组织、相互协作的软件开发环境。(本规范中亦简称“生产线”)。生产线能够提供成套的软件开发支撑,其内在驱动是软件生产过程,目的是生产软件产品。

2.2 软件工具

软件工具是一类用来辅助计算机软件开发、运行、维护、管理、支持等过程中的活动或任务的软件。(本规范中亦简称“工具”)。

2.3 软件制品

软件制品是软件开发过程中产生的模型、代码、文档等与软件相关的实体,是软件工具的产出和作用对象,是软件生产过程的中间产品,或者软件交付的组成部分。(本规范中亦简称“制品”)。

2.4 软件生产线框架

软件生产线框架是用于支撑软件生产线全生命周期活动的基础设施,为生产线的构造、运行提供有效支持。

第3章 软件生产线及其组成要素

3.1 软件生产线

软件生产线是一种基于 Internet 的集成化、可扩展、协同化的软件开发环境，它将软件开发过程中涉及的人、工具、制品等要素，按照一定的软件开发方法，有序组织起来，并使其相互协作。软件生产线能够提供成套的软件开发支撑，其内在驱动是软件生产过程，目的是生产软件制品。

软件生产线的生命周期包括两个阶段：构造和运行。**软件生产线构造**即将软件生产线中人、工具、制品等组成要素有序组织在一起，形成一个统一、一致的整体（即软件生产线）。**软件生产线运行**即生产线各组成要素间相互协作，共同支撑基于软件生产线的软件开发实践。本规范主要关注软件生产线的构造和运行。

软件生产线不仅是人、工具、制品的静态集合，更是一种动态的、能够反映软件开发方法学特征的开发环境。这种动态特性通过软件生产线**过程模型**表达。过程模型是对软件生产线的过程抽象，描述了基于某种软件开发方法的软件开发过程，是反映软件生产线特征的关键。基于过程模型表达机制，本规范将软件生产线刻画为由一系列制品**加工步骤**组成的**加工工序**。软件开发中的人、工具、制品通过加工步骤实现关联，形成软件生产线，如图 3-1 所示：

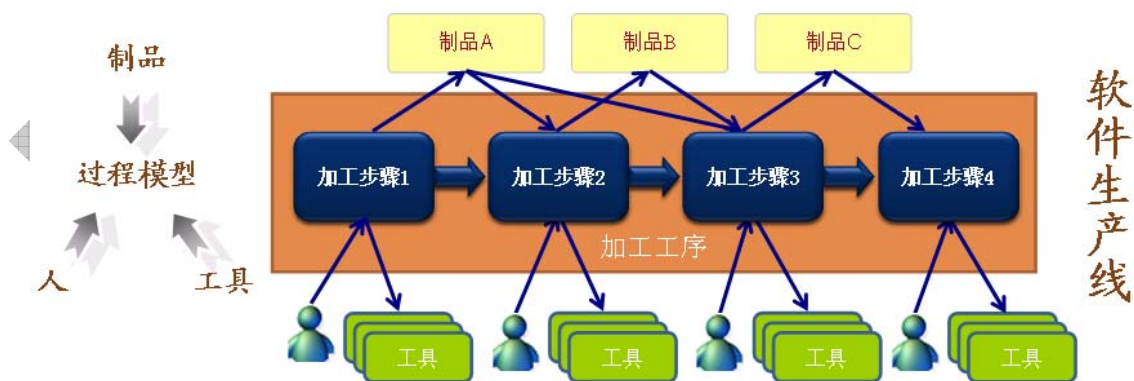


图 3-1 软件生产线概念图

3.2 软件生产线组成要素

软件生产线由三类组成要素构成：软件生产线使用人员、软件工具以及软件

制品。各要素需具备一定的属性特征,以便于实现要素间的有序组织和相互协作,形成统一、一致的软件生产线。

3.2.1 软件生产线使用人员

软件生产线使用人员泛指利用软件生产线实施软件开发相关活动的人,是软件生产线构造人员、软件生产线装配人员和软件生产线用户总称。

- 1、软件生产线构造人员,指根据某类软件特征,制定软件生成过程,其具体职能包括:
 - 设计生产线加工工序
 - 设定生产线相关的工具集;
 - 设定生产线制品集信息;
 - 设定生产线中的角色。
- 2、软件生产线配置人员,指使用具体的软件开发项目信息对软件生产线进行实例化与初始化操作的人,其具体职能包括:
 - 配置本项目相关的工具集;
 - 配置本项目制品集信息(如制品存取位置);
 - 配置本项目开发过程中的角色和用户;
 - 制定和分配软件开发任务。
- 3、软件生产线用户,指基于实例化的软件生产线,实施软件开发实践的人。其具体职能包括:
 - 进行生产活动,完成软件开发任务。

3.2.2 软件工具

软件工具是软件生产线中用以辅助完成软件生产活动的各类软件。在软件生产线中,软件工具不但能被用户直接使用,而且能够通过工具间的相互调用,实现功能组合,减少人工参与,提高软件开发效率。

工具间要实现良好的功能组合,需要具备以下属性特征:

1、封装性

一个工具要提供自己的功能给另一个工具使用,需具备良好的封装性,能够提供独立的功能调用接口,并具备良好的接口描述,方便其它工具的发

现、识别和使用。

2、模块化

工具需具备清晰地模块化特征，一方面便于封装自身功能供其它工具调用，另一方面也便于调用其它工具的功能，将其作为一个子模块，组合在自身的功能集合中。

3.2.3 软件制品

软件制品是软件生产线用户使用软件工具进行软件生产活动的结果产出。就其存在形态而言，软件制品可以是文档、代码、脚本、模型等各种与软件相关的实体；就其使用价值而言，软件制品既可以是软件开发过程中的中间产物，也可以是最终软件交付的组成部分；就其所充当的角色而言，软件制品既是软件工具作用的对象，也是软件工具产出的结果。

要实现软件生产线中软件工具对软件制品的共享，实现软件工具间的数据关联，软件制品需具备以下属性特征：

- 1、互操作性——保障数据表示的一致性，体现为软件生产线中制品间具有统一的模型视图。
- 2、非冗余性——强调软件生产线中制品的冗余程度，体现为软件生产线中统一的制品副本管理和版本控制。

第4章 软件生产线框架

软件生产线是面向特定软件开发方法的，用于支撑软件开发实践的，集成化、可扩展、协同化的软件开发环境。软件生产线的生命周期分为构造与运行两个阶段：**1、软件生产线构造阶段**，即按照一定的软件开发方法，设计软件生产线过程模型，并通过过程模型将生产线中的人、工具、制品等要素有序组织起来，形成以过程模型为核心的软件生产线模型；**2、软件生产线运行阶段**，即对设计好的软件生产线模型实例化，配置相关的软件开发项目信息，形成支撑该软件项目开发的软件生产线，并在生产线的指导下进行软件开发实践。

软件生产线的核心是软件生产线元模型，基于生产线原模型构造软件生产线，并支持软件生产线的运行。

4.1 软件生产线元模型

软件生产线过程模型是对特定软件开发方法的软件开发过程的抽象，表现为由一系列制品加工步骤组成的加工工序。**软件生产线模型**以过程模型为核心，通过过程模型将生产线中人、工具、制品等要素有机联系在一起，是对软件生产线要素及其关系的抽象。

本规范定义软件生产线元模型及其过程元模型如图 4-1 所示：

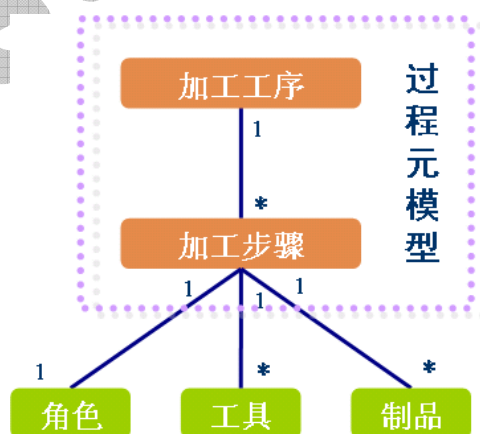


图 4-1 软件生产线元模型及软件生产线过程元模型

加工工序与加工步骤

加工工序是一组用于描述软件开发过程的软件生产活动序列，是对软件生产

线的过程抽象。

加工步骤是加工工序中的一道“工序”环节，是对一类软件生产活动的抽象，例如“开发”步骤、“测试”步骤等。

加工工序定义了一组有序的加工步骤。

加工步骤与角色、工具、制品

加工步骤定义本步骤相关的角色信息、工具集信息，以及制品集信息，实现软件生产线中人、工具、制品间的关联。

角色是对软件生产线用户在利用软件生产线实施软件开发实践的过程中所承担的职能的定义，例如“开发人员”、“测试人员”等。

加工步骤通常由一种职能角色的用户担当完成,例如在某生产线的“开发”环节，各“开发人员”需要进行多个软件功能模块的开发活动，在“测试”环节，“测试人员”需要进行各功能模块的单元测试以及集成测试等活动。加工步骤需要对应一个特定的软件工具集、一个特定的输出制品集，以及一个或多个可能的输入制品集（加工工序中的第一个加工步骤通常没有输入制品集）。软件工具集中包含了完成该加工步骤所需的工具；输出制品集中包含了该加工步骤所产出的制品结果，输入制品集中包含了该加工步骤所需的软件制品。

角色与工具

角色通过加工步骤与工具相关联。软件生产线使用人员根据自身角色，利用加工步骤中定义的工具集信息，获取指定工具集中的工具，下载并安装，以此形成个人开发环境，实施软件生产活动。

工具与制品

工具通过加工步骤与软件制品相关联，利用加工步骤中定义的软件制品集信息，获取输入制品，经过加工，将结果输出为输出制品。

4.2 软件生产线设计

根据 4.1 中生产线元模型中定义的概念及关联关系，生产线设计流程如图 4-2 所示：

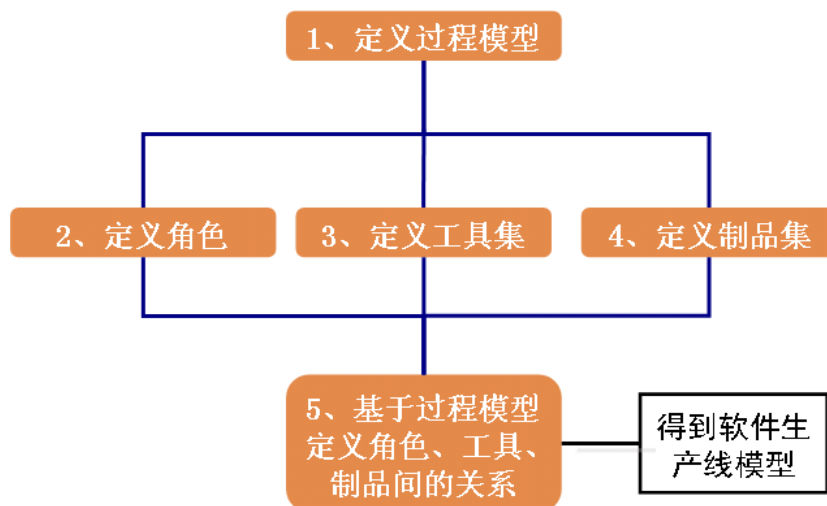


图 4-2 软件生产线模型设计流程

- 1、定义反映特定软件开发方法学特征的软件生产线过程模型；
- 2、定义软件开发过程中涉及的角色；
- 3、定义软件开发过程中使用的工具集，一个加工步骤对应一个工具集；
- 4、定义软件开发过程中产生的制品集，一个加工步骤对应一个输入制品集及一个输出制品集；
- 5、基于软件生产线过程模型，定义角色、工具、制品间的关联关系，形成软件生产线模型。

4.3 软件生产线装配

软件生产线模型刻画了软件开发的过程，以及开发过程中人、工具、制品之间的关系，是一个通用的、可设计形成面向特定软件开发方法的生产线模型。软件生产线装配是对软件生产线模型的实例化与初始化操作，即针对特定的软件开发项目，配置相关的项目信息，形成支撑该项目开发的软件生产线。

软件生产线装配阶段需要配置的信息包括：

1、任务

任务是软件生产线过程模型中软件生产活动的实例，它描述了软件生产活动的具体执行人员、活动内容、活动的目标和要求，以及所受到的约束条件等。一个加工步骤可能被分解为一个或多个任务。（见图 4-3）

2、用户

用户是软件生产活动的参与者，是任务的实际执行者。一个任务可以分配给

多个用户共同完成，同时，一个用户也可以承担多项任务。（见图 4-3）

- 3、角色（可选）
- 4、工具集（可选）
- 5、制品集（可选）

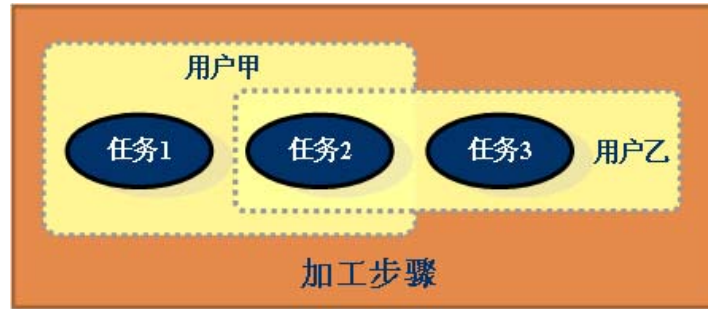


图 4-3 加工步骤任务分解与分配

软件生产线装配流程如图 4-4:

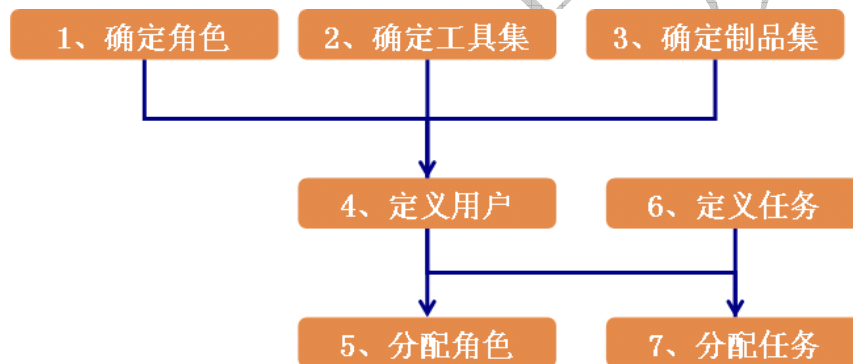


图 4-4 软件生产线装配流程

- 1、确定软件开发过程中涉及的角色（可使用软件生产线模型定义的默认值，也可根据实际项目需要重新设置）
- 2、确定软件开发过程中使用的工具集（可使用软件生产线模型定义的默认值，也可根据实际项目需要重新设置）
- 3、确定软件开发过程中产生的制品集（可使用软件生产线模型定义的默认值，也可根据实际项目需要重新设置）
- 4、定义参与软件开发活动的用户
- 5、为用户分配角色
- 6、定义软件开发过程中的任务
- 7、为用户分配任务

4.4 软件生产线使用

通过软件生产线的设计与装配，已经形成支撑项目开发实践的软件生产线。生产线使用人员使用该软件生产线，组装个人工作平台，并完成软件开发任务。软件生产线使用流程如图 4-5 所示。

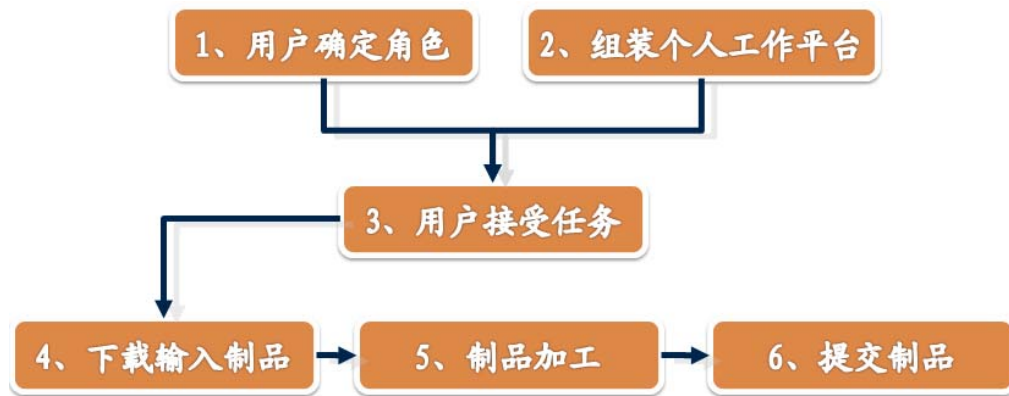


图 4-5 软件生产线的使用流程

生产线使用具体流程如下：

- 1、生产线用户获得自己角色信息；
- 2、生产线用户根据自己的角色组装个人工作平台；
- 3、生产线用户接收生产线任务；
- 4、生产线用户根据任务的要求下载输入制品；
- 5、生产线用户加工制品，完成开发任务；
- 6、生产线用户提交加工后的软件制品。

第5章 软件生产线框架参考实现

软件生产线框架为是支持软件生产线构造、运行的基础环境。根据第4章提出的软件生产线框架，本章节给出以下生产线框架的参考实现。图5-1描述了软件生产线框架参考实现架构。

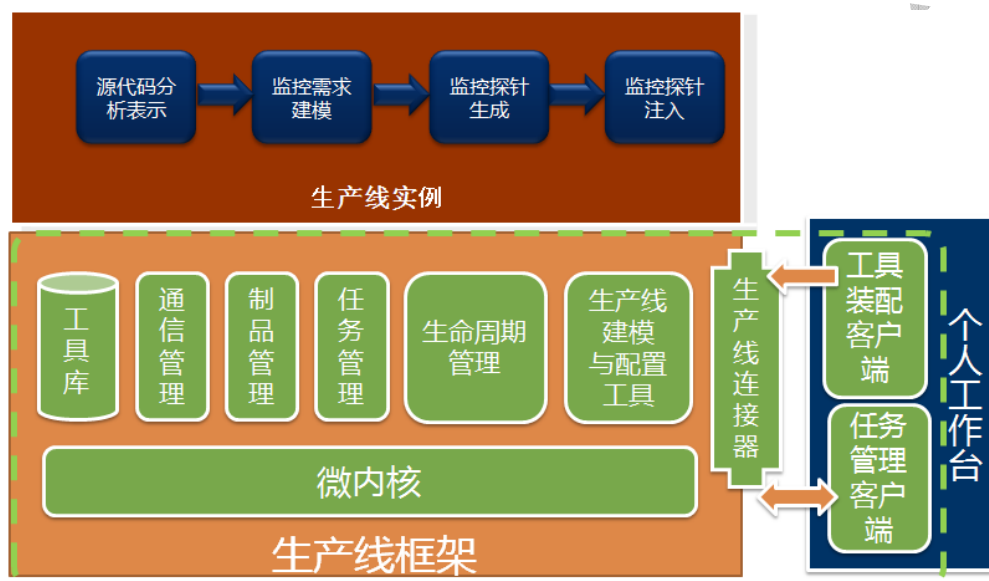


图5-1 软件生产线框架参考实现架构

5.1 微内核

为了解决集成环境的可扩展性问题，工业界引入一种“微内核+扩展体”的插件体系结构。在该体系结构中，“扩展体”作为外围设备按照一定的规范编写并运行在微内核之上，微内核负责“扩展体”的组织、管理和运行。该体系结构中的“扩展体”是一种结构化构件，它将代码（和/或文档）添加至系统并以结构化方式描述，能够在不对原系统作修改和编译的情况下将该功能添加至系统。

本规范提出的软件生产线框架参考实现参照微内核体系结构，框架提供的各类服务以“扩展体”方式发布。

5.2 生产线建模与配置工具

生产线建模与配置工具提供图形化的界面，辅助生产线设计/配置人员构造软件生产线模型，包括定义加工工序及其加工步骤，定义生产线中的角色和工具，定义加工步骤与角色、工具、制品之间的对应关系，最终生成描述软件生产线模

型的配置文件。

配置文件采用 XML 文件描述以下信息：

角色<roles>，定义了软件生产线中所涉及的用户角色；

工具集<toolsets>，定义了软件生产线所包含的全部工具集，包括工具集名称、描述信息、URL 资源地址等属性；

加工工序<spl_process>，定义软件生产线的加工工序，由一组加工步骤所组成；

加工步骤<spl_step>，定义加工步骤名称、描述信息、所关联的角色、所关联的工具集，以及所关联的输入制品集、输出制品集等属性。

以下是流程制导软件生产线的配置文件示例：



5.3 生命周期管理

生命周期管理提供软件生产线模型的运行时实例管理，提供生产线配置信息的查询和管理接口。

5.4 工具库

软件生产线中的工具按照软件生产线框架规范，以一定的形式存放于工具库

中。工具库提供工具的添加、删除、查询等功能。

5.5 任务管理

任务管理系统负责制定、调度和管理软件开发任务。在软件开发过程中，项目管理人员制定各个加工步骤的软件开发任务，描述任务的内容，设置任务的相关属性（如输入/输出制品集地址信息），制定任务的要求（如任务的完成时间），指定完成任务的项目成员。任务管理系统同时需要支持任务的修改、删除、查询等。

5.6 制品管理

制品管理系统负责存储、提取、更新软件开发过程中所产生的软件制品，支持制品在整个软件开发过程中的共享。

- 制品存储：将软件开发过程中所产生的制品存储在加工步骤中指定的输出制品存放位置。
- 制品提取：按照加工步骤中指定的输入制品存储地址，获取软件开发所需的制品。
- 制品更新：将新版本的软件制品替换原有的旧版本的软件制品。

5.7 通信管理

通信管理模块为软件生产线中的工具提供多种形式的访问机制。生产线中的工具在控制集成方面满足*提供*和*使用*两种调用关系。一个工具可以将自身的功能以服务的形式提供给另一个工具使用；反之，一个工具可以使用另一个工具所提供的服务，并且当该服务不可用或不满足需要时，能够方便、灵活地替换为其他工具的服务。

5.8 个人工作台

工具装配客户端

软件生产线用户利用工具装配客户端实现软件生产线模型的实例化。

在软件开发过程中，生产线用户根据自身的角色，通过工具装配客户端，从生产线工具库中下载并安装本角色相关的开发工具（在加工步骤中定义），以此形成本用户的软件开发环境，即个人工作台。

任务管理客户端

任务管理客户端从任务管理系统中获取本用户相关的软件开发任务, 根据任务描述的制品位置信息(输入/输出位置), 提取待加工制品; 在制品加工完成后, 根据制品位置信息存储制品。

Trustie技术规范

参考文献

- [1] Trustie课题组, 可信的国家软件资源共享与协同生产环境项目申请书.
- [2] Eclipse home <http://www.eclipse.org>
- [3] RAD7.0 product home <http://www-306.ibm.com/software/awdtools/developer/application/>
- [4] SourceForge home <http://www.sourceforge.com/>
- [5] SourceCast home <http://www.sourcecast.org/>

Trustie技术规范