目录

[第一章 绪论 1](#_Toc399359740)

[1. 嵌入式系统 (P11 P13 P14) 1](#_Toc399359741)

[2. 嵌入式系统的特点 (P21) 2](#_Toc399359742)

[3. 嵌入式处理器分类 (P24) 3](#_Toc399359743)

[4. 嵌入式系统的组成 (P29) 3](#_Toc399359744)

[5. BSP的概念 (P31) 4](#_Toc399359745)

[6. “中国物联网”定义 (P51) 5](#_Toc399359746)

[7. 软件对于嵌入式系统的意义 (P53) 5](#_Toc399359747)

[8. 软件工程环境之软件开发过程及主要模型 (P55) 6](#_Toc399359748)

[第二章 嵌入式软件的开发过程 6](#_Toc399359749)

[1. 研究嵌入式软件开发过程的意义 (P3) 6](#_Toc399359750)

[2. 面向管理过程 (P22) 7](#_Toc399359751)

[3. 嵌入式软件开发过程的特点 (P29) 8](#_Toc399359752)

[4. 过程改善的方法 (P41) 10](#_Toc399359753)

[5. PDCA循环 (P42) 11](#_Toc399359754)

[6. 质量改进范例QIP (P43) 12](#_Toc399359755)

[第三章 开发和测试的标准 13](#_Toc399359756)

[1. ISO 9000系列标准 (P4) 13](#_Toc399359757)

[2. ISO/IEC 12207 (P6) 13](#_Toc399359758)

[3. 能力成熟度模型集成 CMMI (P7) 14](#_Toc399359759)

[4. IEC 61508标准 (P8) 15](#_Toc399359760)

[第四章 与安全有关的软件系统的许可 16](#_Toc399359761)

[1. “许可”的参与者 (P3) 16](#_Toc399359762)

[2. 获得许可证的影响因素 (P9) 16](#_Toc399359763)

[第五章 嵌入式系统的法律问题 18](#_Toc399359764)

[1. 计算机软件保护条例 (P3) 18](#_Toc399359765)

[第六章 嵌入式软件的需求工程 19](#_Toc399359766)

[1. IEEE对需求的分类 (P3) 19](#_Toc399359767)

[2. 嵌入式系统的需求分类 (P4) 20](#_Toc399359768)

[3. RE（需求工程 ） (P5) 20](#_Toc399359769)

[4. RE系统环境 (P7) 20](#_Toc399359770)

[5. RE过程的3个维度 (P12) 22](#_Toc399359771)

[6. RE 过程的5个阶段 (P16) 23](#_Toc399359772)

[7. 复杂嵌入式系统的层次模型 (P26) 25](#_Toc399359773)

[8. 基于场景和目标的系统开发的需求工程 (P39) 27](#_Toc399359774)

[9. 设计举例：数码相机 (P50) 31](#_Toc399359775)

[第七章 软件结构和系统结构 35](#_Toc399359776)

[1. 软件结构 (P2) 35](#_Toc399359777)

[2. 嵌入式系统中与结构相关的特征 (P13) 36](#_Toc399359778)

[3. 嵌入式系统的一般性元结构 (P21) 38](#_Toc399359779)

[4. 嵌入式系统的元结构：组件类 (P2) 38](#_Toc399359780)

[5. 嵌入式系统的元结构：连接器类 (P23) 39](#_Toc399359781)

[6. 嵌入式系统的结构风格 (P39) 40](#_Toc399359782)

[7. X-By-Wire结构 (P40) 40](#_Toc399359783)

[第八章 嵌入式软件测试 41](#_Toc399359784)

[1. 嵌入式软件测试的目的与特征 (P2) 41](#_Toc399359785)

[2. 软件测试技术的多样性 (P3) 41](#_Toc399359786)

[3. 嵌入式软件测试的特殊性 (P10) 42](#_Toc399359787)

[4. RTCA/DO-178B标准 (P12) 42](#_Toc399359788)

[第九章 安全性和可靠性分析技术 43](#_Toc399359789)

[1. 安全性和可靠性的要求背景 (P2) 43](#_Toc399359790)

[2. 安全性和可靠性的应用领域 (P3) 43](#_Toc399359791)

[3. 方法和技术 (P5) 44](#_Toc399359792)

### 第一章 绪论

#### 嵌入式系统 (P11 P13 P14)

**描述1**

嵌入式系统被定义为：以应用为中心、以计算机技术为基础、软件硬件可裁剪、适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。(最常用的描述)

**描述3**

嵌入式系统是计算机技术，通信技术，半导体技术，微电子技术，语音图象数据传输技术，甚至传感器等先进技术和具体应用对象相结合后的更新换代产品，是技术密集，投资强度大，高度分散，不断创新的知识密集型系统。反映当代最新技术的先进水平。

**描述4**

嵌入式系统是一个分散的工业,充满了竞争、机遇与创新,没有哪一个系列的处理器和操作系统能够垄断全部市场。

从某种意义上来说，通用计算机行业的技术是垄断的。占整个计算机行业90%的PC产业，其中80%采用Intel的80x86体系结构，芯片基本上出自Intel，AMD等几家公司。在操作系统和文字处理器方面，Microsoft的Windows及Word占80～90%，凭借操作系统还可以搭配其它应用程序。因此当代的通用计算机工业的基础被认为是由Wintel（Microsoft和Intel 90年代初建立的联盟）垄断的工业。

#### 嵌入式系统的特点 (P21)

1．嵌入式系统通常是面向特定应用的

嵌入式CPU与通用计算机的最大不同就是嵌入式CPU大多工作在为特定用户群设计的系统中，专用于某个或少数几个特定的任务。

2．嵌入式系统功耗低、体积小、集成度高、成本低

通用计算机有足够大的内部空间，具有良好的通风能力，系统中的Pentium或AMD处理器也均配置庞大的散热片和冷却风扇进行系统散热；而许多嵌入式系统却没有如此充足的电能供应，尤其是便携式嵌入设备，即使有足够的电源供应，散热设备的增加也很不方便。由于空间和各种资源相对不足，因此在设计嵌入式系统时应尽可能降低功耗。整个嵌入式系统设计有严格的功耗预算，其硬件和软件也必须围绕这种特性高效率地设计，量体裁衣、去除冗余，最大限度地降低应用成本，力争在同样的硅片面积上实现更高的性能。

3．嵌入式系统具有较长的生命周期

嵌入式系统和具体应用有机地结合在一起，其升级换代也是和具体的产品同步进行，因此嵌入式系统产品一旦进入市场，便具有较长的生命周期。

4．嵌入式系统具有固化的代码

为了提高执行速度和系统可靠性，嵌入式系统中的软件一般都固化在存储器芯片或单片机本身中，而不是存储于磁盘等载体中，系统上电后程序开始执行直至系统关闭，程序是不能被改变的，除非开发人员采用特定的方法才能对程序进行改进并重新写入系统。所以，可以说嵌入式系统的应用软件生命周期也和嵌入式产品一样长。另外，各个行业的应用系统、产品和通用计算机软件不同，很少发生突然性跳跃，嵌入式系统中的软件也因此更强调可继承性和技术衔接性，发展比较稳定。

5．嵌入式系统开发需要专用开发工具和环境

6．嵌入式系统软件需要RTOS开发平台

通用计算机具有完善的操作系统和应用程序接口（API），是计算机基本组成不可分割的一部分，应用程序的开发以及完成后的软件都在操作系统平台上运行，但一般不是实时的。嵌入式系统则不同，其应用程序可以没有操作系统而直接在芯片上运行；但是为了合理地调度多任务，利用系统资源、系统函数以及专家库函数接口，嵌入式系统一般使用实时操作系统RTOS，使系统具有实时约束。用户必须自行选配RTOS开发平台，这样才能保证程序执行的实时性、可靠性，并减少开发时间，保障软件质量。

7．嵌入式系统开发人员以应用专家为主

嵌入式系统要和各个不同行业的应用相结合的，要求更多的计算机以外的专业知识，其开发人员往往是应用领域专家。

8．嵌入式系统是知识集成系统

嵌入式系统是将计算机技术、半导体技术和电子技术与各个行业的具体应用相结合后的产物，是一门综合技术学科。

#### 嵌入式处理器分类 (P24)

嵌入式系统硬件核心是嵌入式处理器。

据不完全统计，全世界嵌入式处理器的品种数量已经超过1000多种，流行体系结构有30多个，其中8051体系占大多数。

嵌入式处理器可以分为四类：

嵌入式微处理器EMPU（Embedded Microprocessor Unit）

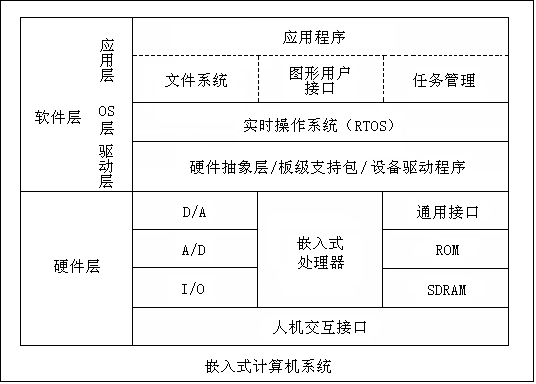
嵌入式微控制器EMCU（Embedded Microcontroller Unit）

嵌入式数字信号处理器EDSP（Embedded Digital Signal Processor）

嵌入式片上系统ESoC（Embedded System on Chip）

#### 嵌入式系统的组成 (P29)

嵌入式系统是面向特定应用的。

实际应用中，绝大多数嵌入式系统是用户针对特定任务而定制的，但就其组成而言，它们一般都是由硬件、软件以及嵌入式系统的开发工具和开发系统3部分组成的，如图所示。 

#### BSP的概念 (P31)

BSP简单地说，就是一段启动代码，和计算机主板的BIOS差不多，但提供的功能相差很大。

BIOS主要负责在电脑开启时检测、初始化系统设备（设置栈指针，中断分配，内存初始化..）、装入操作系统并调度操作系统向硬件发出的指令。

尽管BSP的开始部分和BIOS所做的工作类似，但是BSP还包含和系统有关的基本驱动。

BIOS程序用户不能更改，只能对参数进行修改设置，但BSP可以修改。

BSP的特点：硬件相关性、操作系统相关性

BSP需要完成两部分工作：嵌入式系统的初始化、设备驱动

嵌入式系统的初始化过程：片级初始化（ CPU初始化）、板级初始化（其他器件初始化、软件的数据结构和参数）、系统级初始化（操作系统初始化）。

#### “中国物联网”定义 (P51)

物联网：指的是将无处不在的末端设备和设施，包括具备“内在智能”的传感器、移动终端、工业系统、楼控系统、家庭智能设施、视频监控系统等、和“外在使能”的，如贴上RFID的各种资产、携带无线终端的个人与车辆等等“智能化物件或动物”或“智能尘埃”，通过各种无线和/或有线的长距离和/或短距离通讯网络实现互联互通、应用大集成、以及基于云计算的SaaS营运等模式，在内网、专网、和/或互联网环境下，采用适当的信息安全保障机制，提供安全可控乃至个性化的实时在线监测、定位追溯、报警联动、调度指挥、预案管理、远程控制、安全防范、远程维保、在线升级、统计报表、决策支持、领导桌面等管理和服务功能，实现对“万物”的“高效、节能、安全、环保”的“管、控、营”一体化。

#### 软件对于嵌入式系统的意义 (P53)

软件对于嵌入式系统的意义：

（1）现代系统中的大部分功能是通过软件来实现的。例如，西门子公司在全世界范围内创造的价值，60%是通过软件来实现的。

（2）在很多企业中，软件对于嵌入式系统都具有很大的技术意义和经济意义。

1）对于非软件企业来说，软件是产品不可缺少的组成部分，如汽车、移动电话等企业。

2）对于软件企业来说，软件就是企业的最终产品，如操作系统、文字处理软件等。

（3）在技术上，软件实现更强大的功能。

原先：嵌入式软件取代硬件来实现系统的功能。

现在：通过软件实现硬件不能或很难实现的功能。

例如：嵌入式软件可以提高汽车驾驶的安全性、使电话交谈变得简单容易、尽早诊断疾病、支持大量的人们希望的服务。

#### 软件工程环境之软件开发过程及主要模型 (P55)

软件开发过程：即软件开发模型，是指软件开发全部过程、活动和任务的结构框架，它支配着整个软件开发的工作流程。

软件开发包括需求、设计、编码、测试和维护等阶段。

软件开发特点：软件开发模型能清晰、直观地表达软件开发的全过程，明确规定要完成的主要活动和任务，是软件项目工作的基础。

### 第二章 嵌入式软件的开发过程

#### 研究嵌入式软件开发过程的意义 (P3)

1. 按工程标准开发软件。

嵌入式软件按显式的过程模型来设计，可以改善软件的生产力、质量以及可预测性。

2. 按照成熟度模型评价软件企业。

成熟度模型是对软件开发组织的评价标准，开发过程的显式表达是过程改善的重要前提。

3. 为一个软件开发组织或项目找到合适的开发过程。

对软件项目的开发而言，不存在普遍的开发方法，也不存在最优的开发方法，只有合适的开发方法。开发过程总是与项目目标相协调，与开发环境相适应。

开发环境包括许多方面，比如应用领域、现有的开发技术、开发者的经验、组织框架、预算范围等。

开发过程必须与每个约束条件都相适应。

#### 面向管理过程 (P22)

面向管理过程：为开发满足产品要求的软件或系统，提供一种环境和服务。

**项目规划**

（1）项目规划：是一种“过程”，这个过程根据项目目标和约束条件，选择合适的模型，并将这些模型集成到项目计划之中。

（2）项目计划：对行为、里程碑、中间结果、最终结果进行描述，对人的角色或行为进行分配。

（3）在项目规划时，不同的模型进行相互配合，共同起作用：比如，通过所选择的项目模型、估计模型，对项目总开销和开发时间做出评估。在评估的基础上，就可以确定某些行为的成本和运行时间。之后，就可以进行资源分配。

（4）除了统一的项目计划之外，在项目规划的范围内还可以制定其他的计划，如配置计划、测量计划、成本计划。

**项目管理**

（1）项目管理：是一种“过程”，这个过程按照项目计划，调节和控制项目的运行。

（2）对项目进行的调节和控制，主要集中在费用、时间、功能性内容等方面。

（3）当偏离项目计划时，项目管理采取相应措施，或改变规划。

**质量保障**

（1）质量保障：是一种“过程”，这个过程根据项目的质量要求，对项目阶段或产品进行检验。

（2）质量保障主要考虑：

1）质量方面的情况，如代码的可维护性。

2）可能导致质量问题的因素，如设计中的接口错误。

（3）质量保障过程必须紧密地与项目管理相联系，以便提前发现与项目计划发生偏离的情况，并及时采取相应的措施。

**产品管理**

（1）产品管理：是一种“过程”，这个过程对产品进行管理与供应。

（2）产品管理包括产品的版本、配置以及存取概念的定义、使用。

#### 嵌入式软件开发过程的特点 (P29)

嵌入式系统具有自己的特点，比如软件封装在硬件之中，与物理系统有较强的相互作用，这些特点对嵌入式软件开发过程具有重要的影响。

嵌入式系统的应用环境不同，其特性也不同，因而很难找到对嵌入式软件普遍适用的开发过程。

所以，对于嵌入式软件的开发来说，需要建立或选择针对特定领域的开发过程。

**多学科开发**

（1）嵌入式软件的开发过程是系统工程中的一个子过程，通常由不同学科（如机械工程、电气工程、软件工程）共同参与开发。软件开发与其他学科交织在一起，所以不能孤立地看待嵌入式软件的开发。

（2）同时，其他学科中也有越来越多的功能依靠嵌入式软件来实现。

（3）相比硬件实现，软件实现具有更高的自由度和设计空间。

（4）开发不同学科（机械、电子、软件、光学等）的应用系统时，要协调不同的开发过程，要对“问题”和“过程”产生共识，特别是对嵌入式软件开发来说，准确地认识和理解周围环境、周边硬件显得尤为重要。

**跨企业开发**

（1）由于专业分工越来越细，一个企业很难完全独立开发一个系统，所以，跨企业开发、企业合作是很现实的问题。例如：在汽车工业中，开发OEM（原始设备制造商）就要与供应商建立密切的联系。

（2）跨企业的开发要明确工作责任、工作分配、接口定义、整合机制。

（3）同步工程是指跨企业开发中任务的并行化，以便缩短开发时间。

（4）随着嵌入式系统的复杂度的上升，必然会有越来越多的企业和开发组织进行协同工作。与汽车供应链一样，会存在多级依赖关系。未来的趋势是，预先做好的元件将更多地应用到嵌入式系统中。

**非功能性特征**

（1）非功能性特征包括实时性、可靠性、容错性、安全性等，它们在嵌入式软件中有着特殊的意义。

例如：实时性在机器人控制中有着重要的意义。可靠性、容错性、安全性在安全要求严格的应用上（如飞机控制）起着重要的作用。

（2）可使用性、人体工程学等一些非功能性特征，在嵌入式环境中的作用比在信息系统中的作用更加重要。

（3）嵌入式系统必须是无差错的：一方面，用户期待一个正确运行的系统；另一方面，软件的更换有时是很昂贵的，因而不会对软件进行维护。

（4）高质量的嵌入式软件是通过提前识别、或系统地避免错误而得到的。因此，错误发现、错误消除、避免发生错误的策略应该是整个开发过程的组成部分。

**经济性**

（1）嵌入式系统市场是信息领域增长最快的市场之一。由此产生了开发时间短、生产成本低的要求。

（2）由于有经济性方面的要求，所以要特别注意开发的可计划性和可控制性。这可以通过定量测量来实现。

（3）通常情况下，生产成本主要是由硬件的价格决定的，这些成本可以通过限制硬件资源（如存储器容量、能量消耗、计算能力方面）得到降低。

（4）软件开发也必须与降低成本相适应，比如采用最优化步骤。

**长生命周期**

（1）长的产品生命周期是多种嵌入式系统的特征。例如：在汽车制造业，大概用3年的时间进行开发，用7年的时间进行生产，用15年的时间进行运营与服务。这个产品的生命周期总计为25年。

（2）在嵌入式系统的开发过程中，必须考虑较长的产品生命周期，特别是开发过程必须与维护过程和服务过程相协调。

（3）在开发嵌入式软件时，要考虑到电子元件的更新。这可以通过标准化的结构、通过与硬件无关的编程来实现。

#### 过程改善的方法 (P41)

改善是打破现状、创造变化、提高实力、将目标放在较现在水准高的地方，拟定达成计划并确实执行。

“过程改善”的目的：增强过程模型的实用性。

在软件领域中，采用了不同的用于过程改善的方法。这类方法提供了发现问题、寻找解决方案的框架。

过程改善的方法：

（1）认证方法：用于验证基本实施过程。“认证”的定义是：“由可以充分信任的第三方证实某一经鉴定的产品或服务符合特定标准或规范性文件的活动” 。比如ISO 9000（质量管理体系标准）。

（2）评估方法：通常额外地提供了改善途径。比如CMMI（能力成熟度集成模型 ）、SPICE（软件过程改进和能力测定，Software Process Improvement and Capability dEtermination，即欧洲和大洋洲流行的国际标准ISO/IEC 15504，比CMMI/CMM更开放）。

（3）连续的改善方法：集中于重要的方面，并提供了一种自我控制、连续的改善。 比如PDCA（PDCA循环 ）、QIP（质量改进范例，Quality Improvement Paradigm）。

#### PDCA循环 (P42)

PDCA循环：戴明（美国质量管理专家）提出，又称为“戴明环”。

P：Plan，计划。确定方针和目标，确定活动计划；

D：Do，实施。具体运作，实现计划中的内容 ；

C：Check，检查。总结执行计划的结果，注意效果，找出问题；

A：Action，改进。对检查的结果进行处理，成功的经验加以肯定，并适当推广、标准化；失败的教训加以总结，以免重现；未解决的问题放到下一个PDCA循环。

PDCA→ PDCA → PDCA → ……

以上四个过程不是运行一次就结束，而是周而复始地进行，一个循环完了，解决一些问题，未解决的问题进入下一个循环，这样阶梯式上升。

PDCA循环是基本的质量管理方法，实际上，也是有效进行任何一项工作的合乎逻辑的工作程序。



#### 质量改进范例QIP (P43)

**行为1 特征化**

（1）确定项目的特征。

（2）利用这些特征，可以识别合适项目的模型，比如采用什么样的过程模型、质量模型

**行为2 定义目标**

（1）用可测试的形式定义项目的目标。

（2）为了能够测试，必须有针对目标的所有维度的度量值和额定值，比如物体、视角、质量等特征

**行为3 制定项目计划**

通过选择模型来制定项目计划：

（1）选择有利于识别特征和目标的模型

（2）选择与质量模型相符的模型

（3）选择支持的方法和工具（开发和测量）

**行为4 执行**

（1）按照所选择的计划执行项目。

（2）包括获取所有度量值的实际值，以及在比较实际值与理论值的基础上建立项目团队的反馈。

**行为5 分析**

在项目结束之后，进行调查，分析所使用的模型展现了哪些优点和缺点，为进一步修改基本模型奠定基础

**行为6 修改基本模型**

（1）在对项目认识的基础上修改存在的模型。

（2）不同的环境可能有不同的模型，因此可能会产生模型等级。

（3）修改模型的目的：提高模型在未来项目中重复使用的可能性。

### 第三章 开发和测试的标准

#### ISO 9000系列标准 (P4)

ISO： International Organization for Standardization，国际标准化组织，是一个全球性的非政府组织，是国际标准化领域中一个十分重要的组织。 ISO成立于1947年，中央秘书处设在瑞士的日内瓦。

中国是ISO的正式成员，代表中国的组织是中国国家标准化管理委员会。

ISO 9000：是指质量管理体系标准，它不是指一个标准，而是一种标准的统称。ISO 9000是ISO发布的12000多个标准中最畅销、最普遍的产品。

ISO 9001：“质量管理体系要求”，2008 版是 ISO 9001:2008 ，对应的国标是GB/T 19001-2008 。 ISO 9001从框架上给出了质量管理体系的要求，在各个方面做了规定，包括公司政策、监督、书面计划、从产品咨询到产品提供、产品的监督、顾客的满意度等等。

#### ISO/IEC 12207 (P6)

IEC： International Electrotechnical Commission，国际电工委员会，是非政府性国际组织，成立于1906年，是世界上成立最早的国际标准化机构。总部设在日内瓦。

1947年ISO成立后，IEC曾作为电工部门并入ISO，但在技术上、财务上仍保持其独立性。

根据1976年ISO与IEC的新协议，两个组织都是法律上独立的组织，IEC负责有关电工、电子领域的国际标准化工作，其他领域则由ISO负责。

ISO/IEC 12207：“信息技术--软件生命周期过程”，2002年针对软件发布

ISO/IEC 12207将“过程”分为3部分：基本过程、支持过程、内务过程

（1）基本过程：包括获取、提供、开发、维护和操作过程等。

（2）支持过程：包括记录、配置管理、质量保证、验证、生效、回顾、审计、问题解决等。验证和回顾是与测试、测试手段连接在一起的，并且需要对软件做出精确的解释，

（3）内务过程：包括管理机制、基础结构、改进和培训等。

#### 能力成熟度模型集成 CMMI (P7)

CMMI ：Capability Maturity Model Integration，能力成熟度模型集成。

SEI：Software Engineering Institute，软件工程研究所。鉴于软件对国防装备和现代化战争的重要作用，1984年美国国防部建立了软件工程研究所。研究所设立在卡内基-梅龙大学，对外的名称是卡内基-梅龙软件工程研究所。

SEI（软件工程研究所）研究的内容包括软件采办、软件体系结构和生产线、过程改进和性能测定、安全性、系统互操作性和可信性。

SEI（软件工程研究所）是一个服务于军队同时面向社会的研究机构，是软件工程领域国际顶级的研究机构，它所开发的软件能力成熟度模型（CMM）系列，在全球范围得到普遍的认可和应用。

CMMI的目的 ：帮助软件企业对软件工程过程进行管理和改进，增强开发与改进能力，从而能按时地、不超预算地开发出高质量的软件。

CMMI的目标要素：质量、时间表、最低成本。

CMMI为改进一个组织的各种过程提供了一个集成化框架，新的集成模型框架消除了各个模型的不一致性，减少了模型间的重复，增加透明度和理解，建立了一个自动的、可扩展的框架。因而能够从总体上改进组织的质量和效率。

#### IEC 61508标准 (P8)

IEC 61508 ：电气/电子/可编程电子安全相关系统的功能安全。

IEC 61508是国际电工委员会(IEC) 制定的适合安全性相关系统的国际标准，该标准预计会成为应用软件领域的标准和标准框架。标准提出一种软件生命周期的方法学，包括风险评估、设计、集成、测试、修正、维护和安全管理。

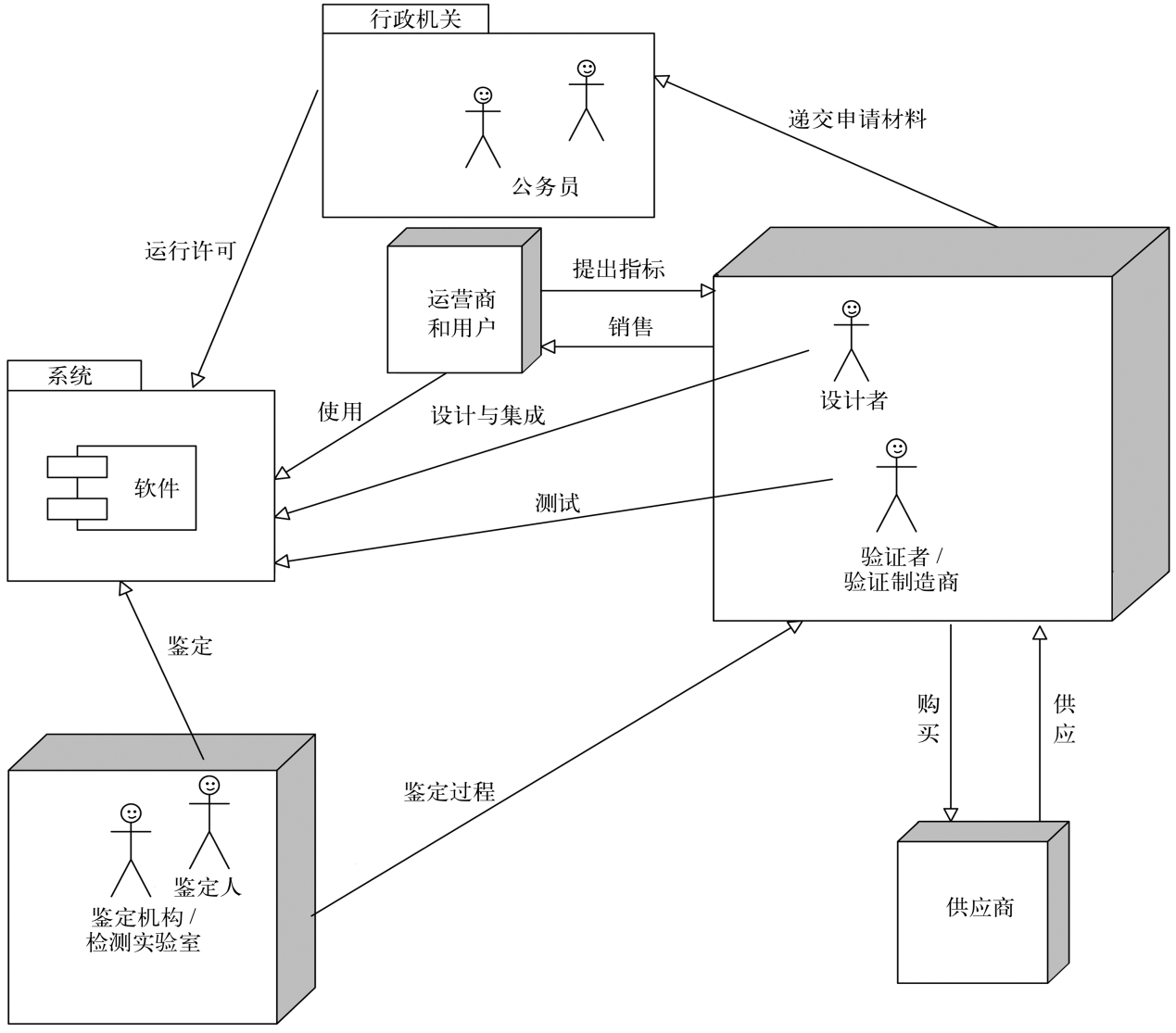
完整的标准由7部分组成，第1～4部分是标准的，这些标准中的要求应该能够得到检验和验证，第5～7部分是非标准的。

其中第3部分定义软件需求和陈述软件的安全生命周期，包括验证和确认，为每个安全级别的软件推荐合适的工具和方法。

IEC 61508-1：第1部分，一般要求  
IEC 61508-2：第2部分，安全相关系统的要求，硬件  
IEC 61508-3：第3部分，软件要求  
IEC 61508-4：第4部分，定义和缩写词  
IEC 61508-5：第5部分，安全完整性等级的测定方法  
IEC 61508-6：第6部分，第2、3部分的应用指南  
IEC 61508-7：第7部分，技术和措施概述

### 第四章 与安全有关的软件系统的许可

#### “许可”的参与者 (P3)



#### 获得许可证的影响因素 (P9)

**系统的复杂度**

（1）系统的复杂度直接影响许可和鉴定过程的复杂度。

（2）复杂系统的许可过程需要更多的时间和占用更多的资源。系统越复杂，所涉及的人员或组织就越多。

（3）对于复杂的系统来说，需要明确规定责任和任务。

**规划和约定**

（1）从一开始，就要制定一份申请许可证的步骤、计划。

（2）通常，在申请许可证的计划书中至少应包括下列内容：

1. 必须检测哪些项目？根据哪些标准和法律？以何种形式？

2. 现有的检测报告哪些可用？哪些项目需要重新检测？

3. 为保证及时完成检测，什么时候必须拥有哪些文件？

4. 检测报告必须在什么时候、以何种方式递交给行政机构？

5. 检测需要用到哪些资源？资源可以是实施检测的检验人，也可能是以下一些情况：

(1) 测试环境，使用环境。

(2) 测试工作人员，或者协助人员（制造商或运营商的员工）。

(3) 气候环境：是否需要在特定的环境下进行检测？如果需要，那么这个检测可能受不同季节的影响。

6. 谁来承担测试的费用？

7. 测试期间会产生哪些文档？文档的内容是什么？

8. 使用何种语言来编制文档？应计划好翻译文档所需的时间和资源。

9. 必须进行哪些商谈？这些商谈在何时何地举行？商谈的目的是什么？如果申请许可证的过程涉及不同的组织和众多的参与者，那么找出一个大家都能接受的商谈日期是比较困难的，必须事先考虑。

**人为因素**

（1）尽管申请许可证有一套形式上的规定，但这个过程始终是由人来参与的，从而导致除了技术因素外，人的感觉也是不能忽略的。

（2）每个人都有自己的任务和目标，理解他人有助于降低风险。应考虑您的同伴正处于什么样的环境之中。这些环境包括个人环境、经济形势、职业发展目标、政治或经济的边界条件。

（3）申请人（制造商、运营商）与行政机关或鉴定人之间的相互信任很重要。如果缺乏信任，那么检测过程的花费将会显著上升。行政机关和鉴定人会理所当然地详细询问，以确定所有的材料和陈述是否都是正确无误的。这会带来大量的额外开支和时间上的延迟。因此，改善这种信任关系是十分必要的。

**风险处理**

（1）一般情况下，法律、规范除了包括产品本身的要求外（如产品的功能、外观、文档），还包括了安全性、可靠性方面的要求。

（2）但是，合同仅仅说明了遵守某个规范。

（3）为了达到合同规定的某个规范的安全标准，花费在相应的设计与检验方面的费用可能有很大的不同。因此，在合同签订之前，就必须明确应该满足哪些要求。

**企业秘密的保护**

（1）由于许可过程可能检验工序和技术，所以许可过程可能会带来专门技术或企业秘密外泄的危险。

（2）申请者应与检测机构、鉴定人签订相应的秘密保护协议。但秘密保护协议也存在被破坏的可能，因此，可以出于自身利益的考虑，可以不公开某些企业秘密。

（3）选择可靠的鉴定人，对于企业秘密的保护也有着十分重要的意义。

### 第五章 嵌入式系统的法律问题

#### 计算机软件保护条例 (P3)

在1991年，我国发布了《计算机软件保护条例》，在《著作权法》修订后，国务院于2001年12月20日审议通过了新的《计算机软件保护条例》，自2002年1月1日起实施。

新的《计算机软件保护条例》的规定与《伯尔尼公约》和《与贸易有关的知识产权协议》的规定保持一致。

伯尔尼公约 ：保护文学和艺术作品伯尔尼公约。1878年，法国作家雨果（代表作：巴黎圣母院、悲惨世界）在巴黎主持召开了一次重要的文学大会，建立了一个国际文学艺术协会。1883年，该协会将一份国际公约草案交给瑞士政府。 1886年9月9日，该草案在伯尔尼举行的第三次大会上被通过，定名为《保护文学和艺术作品伯尔尼公约》，简称《伯尔尼公约》。《伯尔尼公约》的产生，标志着国际版权保护体系的初步形成。 1992年，中国成为该公约的第93个成员国。

与贸易有关的知识产权协议：Agreement on Trade-Related Aspects of Intellectual Property Rights，是世界贸易组织（WTO）的知识产权国际公约，简称TRIPS协议，是世界范围内知识产权保护领域中涉及面广、保护水平高、保护力度大、制约力强的一个国际公约。

### 第六章 嵌入式软件的需求工程

#### IEEE对需求的分类 (P3)

* **IEEE对需求进行了不同类别的划分：**

（1）功能性需求：系统或系统元件必须具有的功能。

（2）设计需求：系统或系统元件设计时提出的需求。

（3）接口需求：一种边界条件（如协议），这种需求涉及系统之间、元件

之间、系统与元件之间的相互作用。

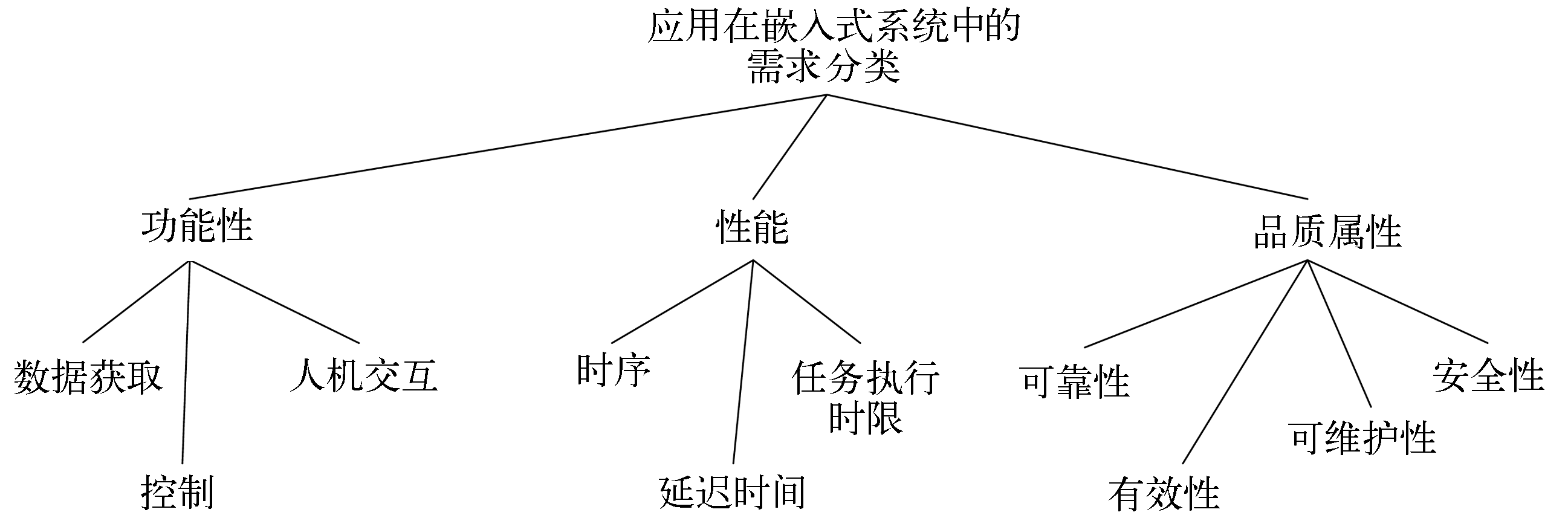
（4）性能需求：功能性需求的运行情况，如运行时间、存储要求、精确度。

（5）物理需求：系统或系统元件的物理特性，如形状、大小、重量。

（6）品质属性：系统的品质特性，如可靠性、安全性、可维护性、移动性。

通常把（2）～（6）归入“非功能性需求”，但，大部分的“非功能性需求”最终都能通过功能性需求来实现。

#### 嵌入式系统的需求分类 (P4)



#### RE（需求工程 ） (P5)

需求工程是随着计算机的发展而发展的：

（1）在计算机发展的初期，软件规模不大，软件开发所关注的是代码编写，需求分析很少受到重视。

（2）后来软件开发引入了生命周期的概念，需求分析成为软件生命周期的第一个阶段。

（3）随着软件系统规模的扩大，需求分析在整个软件开发与维护过程中越来越重要，直接关系到软件的成功与否。人们逐渐认识到需求分析不再仅限于软件开发的最初阶段，它贯穿于系统开发的整个生命周期。

（4）80年代中期，形成了软件工程的子领域-需求工程。90年以后，需求工程成为研究的热点之一。

定义：需求工程是指采用已证实有效的技术、方法进行需求分析，确定客户需求，帮助分析人员理解问题并定义目标系统的所有外部特征的一门学科。它通过合适的工具和记号系统地描述待开发系统，以及系统的行为特征和相关约束，形成需求文档，并支持用户需求的持续改进。

#### RE系统环境 (P7)

系统：由一些具有特定功能、以一定的规律相互联系的物体所组成的总体。

系统环境：是指系统所遭受的所有外界条件及其影响的综合。系统环境直接或间接地影响着一个系统及其开发过程。

系统边界：为了限制所研究问题涉及的范围，用系统边界把所研究的系统与影响系统的环境区分开。

系统的需求，原则上只能在一个具体的环境下被定义，也就是说，没有具体环境就不存在需求（即没有背景的要求）。

系统环境可以划分为4个领域，即从4个方面考察 。

**对象领域**

（1）对象领域确定了系统的所有操作对象，这些操作对象可能涉及系统本身，也可能涉及系统环境，

（2）例如：执行器、传感器，以及这些器件的更新时间间隔、精确度，对系统起决定性作用的基本物理定律。

**使用领域**

（1）使用领域涉及在系统使用过程中产生的一些问题。

（2）例如：系统与系统之间、系统与使用者之间的接口，系统支持的相互作用模型，人机交互，使用目的，使用场景，系统带来的价值。

**IT系统领域**

（1）IT系统领域涉及在系统的研发及应用过程中所面临的政策、制度、法律、法规、要求、约束等方面的一些问题。

（2）例如：将要用到的软、硬件的使用规定，硬件的局限性，现有IT架构带来的种种约束，售后服务，与地域无关的IT安全原则及质量标准，IT基础设施。

**系统开发领域**

（1）系统开发领域涉及在系统开发过程中需要考虑的一些问题。

（2）不同于其他三个领域，系统开发领域是在系统开发时才出现的，主要涉及技术开发层面。

（3）例如：开发方法，开发标准，描述语言，设计工具等。

#### RE过程的3个维度 (P12)

RE过程可以用3个维度来刻划。

**内容维**

（1）内容维：代表系统需求的内容，即，在RE过程的某个阶段，对系统需求达到的理解程度。

（2）RE早期：从系统的模糊设想开始，产生初步需求。

（3）RE后期：开发人员尽可能全面地了解系统的需求，对每个需求进行充分的描述和定义。

**表示维**

（1）表示维：代表系统需求的记录和表达方式。

（2）RE早期：使用信息描述方法来说明需求，如自然语言、草图。

（3）RE后期：使用正式的表达方法描述需求。

**协调维**

（1）协调维：代表在系统需求建立过程中参与人员的协调程度。

（2）RE早期：参与人员对系统需求有不同的见解和想法。考虑到参与人员的基础知识水平，为了有利于协调过程，有时使用非正式的描述语言。

（3）RE后期：参与人员对系统需求的细节达成高度一致，并尽量避免工程后期的利益分歧，这种利益分歧常常导致方案的修改或变更，从而产生巨大开销。

#### RE 过程的5个阶段 (P16)

RE 过程中执行的操作可以划分成五类。

**需求获取**

（1）目标：通过各种信息确定系统的需求。通过对现有系统的观察及对任务的分析，开发、捕获和修订系统的需求。

（2）除了对系统需求进行分析外，还可以采用创造性、团队性的方法（如收集创造性意见或卡片分类法） ，也可以采用基于目标和场景的处理方法。

**需求协商**

（1）目标：找出参与人员之间存在的系统需求的分歧，并找到消除分歧的解决方案。

（2）例如：消除分歧的一个基本要素是找出所有可能的替代方案，以及支持或反对每个方案的理由。

（3）重要的成功因素：让所有的相关人员参与分歧的评判和分歧的解决过程。

**需求描述**

（1）目标：使用某种适当的语言及模型，对系统的需求进行详细的形式化的描述。

（2）选择适当的语言是重要性的：使用适当的语言，能使用户确认系统需求的正确性和有效性，能够支持体系结构的开发和实现。

**需求检验**

（1）目标：检验需求的一致性、完整性、有效性、正确性，以及参与人需求意见的协调程度。

（2）在这里，“有效性的检验” 表示人为进行的调查，例如，关于顾客或用户需求的调查。

（3）“需求检验”阶段检验了所记录的需求在原则上是否正确。它确保没有忽略需求，而且考虑到系统即使在以后使用，这些需求也是正确的。

**需求管理**

（1）目标：在系统的整个生命周期中，对系统需求进行连续一致的管理。支持系统的需求演进，比如像需求变化和可跟踪性等问题。

（2）可以将需求的管理过程模型化，例如基于信息模型的管理。

（3）使用数据库时，“需求”通常进行版本化。

#### 复杂嵌入式系统的层次模型 (P26)

**“系统需求”层**

（1）“系统需求”层：需求层。最上层，定义了整个系统的需求。

（2）也可以根据实际情况，从某个专业角度安排这个层次的结构，不必局限于参考结构。

**“系统总体结构”层**

（1）“系统总体结构”层：设计层。对系统以功能模块网的形式进行描述。

（2）功能模块：在逻辑上属于系统（或系统某功能）的某个任务，是系统需求的子集。

（3）该层从技术角度对功能模块进行定义，并不对它进行软/硬件的划分。

（4）功能模块借助通信路径形成网络，并具有进行相互通信的接口。需要说明的是，这些接口在物理上是否具有可实现性并不重要，模块之间的通信信道只是用于确定系统的总体结构。

**“对功能模块的需求”层**

（1）“对功能模块的需求”层：需求层。在上一层“系统总体结构”中定义了一些功能模块，本层就是将整个系统的需求对应归入到这些模块中去。

（2）在确定功能模块需求的时候，可以根据需要对“系统需求”进行细化。“对功能模块的需求”可以是功能模块内部的功能需求，也可以是功能模块外部的行为或功能需求。

（3）本层也可能产生对上一层“系统总体结构”的新的需求。

**“硬件/软件总体结构”层**

（1）“硬件/软件总体结构”层：设计层。对系统的功能进行软/硬件部分的划分，是系统功能的细化。

（2）在划分过程中，除了要考虑“系统总体结构”外，还要考虑“对功能模块的需求”。

（3）本层对软/硬件部分进行抽象定义，即，对模块功能进行原则性的软/硬件划分，而具体的确定，如通信总线、传感器、执行器、具体的软/硬件接口等的确定，并不在这个层次上进行。

**“对硬件/软件的需求”层**

（1）“对硬件/软件的需求”层：需求层。以“功能模块需求”为依据，详细描述“软/硬件需求”及“硬件/软件总体结构”

（2）本层对功能模块需求进行相应的细化。

（3）本层也可能产生对上一层“硬件/软件总体结构”的新的需求。

**“软件分配”层**

（1）“软件分配”层：设计层。将软件划分到具体的硬件部件中，即，确定软件到具体硬件的调配。

（2）这种划分可能产生2方面的影响：

1）产生新的软件需求，例如，根据具体硬件产生的需求变化。

2）这种划分使硬件需求具体化。

**“详细的软/硬件需求”层**

（1）“详细的软/硬件需求”层：需求层。结合软件的分配，重新定义和细化软/硬件需求。

（2）至此，详细描述了软/硬件需求，同时，对软/硬件接口也进行了约束性的定义。

（3）“层次模型”可以根据专业领域或系统复杂度进行优化或简化。

#### 基于场景和目标的系统开发的需求工程 (P39)

SEGOS-RE：基于场景和目标的系统开发的需求工程。

SEGOS-RE技术是波尔教授的软件系统工程工作组提出的。通过这种技术，软件的使用变得多样化，并根据已有经验得到改善和发展。

SEGOS-RE组成：由5个部分行为组成，TA-1～TA-5，这些行为在三个RE协同设计过程中得到执行。

**环境分析(TA-1)**

（1）TA-1：环境分析。

（2）目标：对系统环境及其主要特性进行识别和记录。

（3）除了4个系统环境领域，还要注意已经在更高的抽象层次中定义的系统环境。

环境分析的目的：认识系统环境及其基本作用，并进行记录。需要考虑对象领域、使用领域、IT系统领域、系统开发领域等所有4个系统领域。

可以采用的方式：邀请相关领域的专家参与所有相关活动。

分析的内容： 4个系统领域涉及的对象，特别是，在对象领域中观察到的现象，以及采用哪些传感器、执行器。例如：通过传感器，监控哪些物理量（如速度、方位、温度等），使用哪些控制功能来控制执行器。

已经确定的环境因素、它们之间的相互关系、它们与系统之间的关系，可以记录在一个环境图中。

**目标认定(TA-2)**

（1）TA-2：目标认定。

（2）目标：获取目标，对目标进行有效性证明，并记录，形成文档。

（3）除了4个系统环境领域，还要注意已经在更高的抽象层次中建立的目标。

目标：就是系统要实现的成果、要达到的目的。包括功能目标、质量目标、成本目标、风险目标、管理目标等。

目标的认定：目标的认定可以与环境分析并行地进行，也可以在环境分析之后进行。目标可以通过多种方式来确定，比如：环境模型、出资人之间的对话、分析已有文件记录、分析已有成品系统等。

除了系统的目标外，还要考虑系统开发过程的目标，比如，提高某个具体的开发行为的效果或效率。

为了充分地确定目标，可以联合使用目标建模方法和场景建模方法，目标和场景具有较大的关联性：

1. 场景提供了目标。场景记录了所希望的、具体的目标的实现，使目标变得更容易理解。

2. 场景能识别新的上层目标。可以通过目标实现来获得新的上层目标。

3. 场景能认定新的、细节化的目标。可以通过场景记录的目标实现过程得出新的细节化目标。

4. 场景能修订已有目标。比如，只有部分目标能够实现，或者，所记录的目标之间的关系被证明是错误的，此时需要修订目标。

**场景认定(TA-3)**

（1）TA-3：场景认定。

（2）目标：对系统的具体使用过程进行获取、记录，并形成文档。

（3）除了4个系统环境领域，还要注意已经在更高的抽象层次中定义的场景及其优化。

场景：描述了可能发生在几个对象之间、系统与环境之间的行为。定义场景的原型仍然是对象领域、使用领域、IT系统领域、系统开发领域等4个环境领域。

“对象领域”的场景：描述了如何对“对象领域”的信息进行更新。

“使用领域”的场景：记录了系统所希望的使用情况。

“IT系统领域”的场景：描述了系统应该支持的、所希望进行的维护程序。

“系统开发领域”的场景：描述了系统开发可以采用的模型、工具。

场景认定的依据：

1. 已被认定的目标。

2. 环境分析的结果。

3. 成品系统的使用实例（用例）。

4. 成品系统的用户文件、系统文件。

5. 专家，以及所有参与人。

场景的获得、记录、有效性检验，首先应观察系统的正常行为；然后，再对可能出现的系统错误、意外行为进行详细描述。如有必要，可以使用“错误实例”来记录不希望的使用方式或者错误的使用方式。

**需求的推导(TA-4)**

（1）TA-4：需求的推导。

（2）目标：推导出每个抽象层次的需求。

（3）这些需求可以根据目标、场景、环境来获得。

目的：产生确认的需求，并形成规范性文本。整个系统的需求可以从TA-1～TA-3的行为中获得。

需求的获取，来自于以下几个方面：

1. 从场景中获取需求：分析所用场景，获取系统的需求。需求可以由一系列的场景步骤推导出来。多个场景可能对同一个需求进行定义，这种情况下，产生的需求就会与两个以上的场景联系起来。

优点：需求来自于场景，反过来，需求也可以嵌入到某个场景环境中去，这使得需求、以及需求之间的关系变得更容易理解。

2. 从目标中获取需求：步骤TA-2（目标认定）中定义的目标，特别是质量目标，是需求的另一个主要来源。

优点：所有重要的质量需求都可以从相应的质量目标中推导出，而且这些需求能达到一定的完备性。从质量目标中推导出需求，可以较好地实现需求与目标的匹配。

3. 从环境对象中获取需求：除了目标和场景外，步骤TA-1（环境分析）中定义的环境对象也可以作为获取需求的来源。

**系统总体结构设计(TA-5)**

（1）TA-5：系统总体结构设计。

（2）目标：定义每个抽象层次的解决方案的结构。

（3）另外，还要对TA-1～TA-4获得的结果进行调整，进行相应的匹配。

系统总体结构设计：简明定义功能模块，不进行软/硬件部分的区分。

功能模块可以与步骤TA-4（需求的推导）确定的系统需求在内容上捆绑、集成。

功能模块之间的相互作用可以由步骤TA-3（场景认定）定义的场景得出。针对每个场景，认定并记录功能模块之间的相互作用。所有功能模块之间的相互作用确定后，根据需要，可以进行适当的抽象。在抽象的过程中，在不同场景中得出的、相同或相近的相互作用归纳为一般性的相互作用。

可以通过步骤TA-2（目标认定）的目标和步骤TA-1（环境分析）的系统环境，对功能模块及其相互作用进行有效性的检验，同时进行功能模块相互作用的匹配和调整，特别是与系统环境的接口。

#### 设计举例：数码相机 (P50)

### 第七章 软件结构和系统结构

#### 软件结构 (P2)

软件结构：软件结构描述了构成软件系统的各个组件，以及各个组件之间的相互作用。

软件结构的基本要素：组件、连接器、软件系统。

#### 嵌入式系统中与结构相关的特征 (P13)

元结构：描述组件的类型和连接器的类型。一个元结构是元元结构的一个实例。

嵌入式系统的元结构由元元结构推导得出，这一推导过程由嵌入式系统的特性决定。

嵌入式系统中与结构相关的特征主要有7个：设备集成、环境相关、实时性、反应特性、并行性、安全性、互异性。

**设备集成**

一个嵌入式系统就是一个计算机应用系统，它是一个仪器或设备的固定构件。

**环境相关**

（1）嵌入式系统与控制环境紧密相关，它需要控制的环境是由大量的物理过程组成的。

（2）这些物理过程可以是化学的、机械的、电子的或液压的过程。

（3）启动服务器、显示器也可以归类为环境。

（4）这些物理过程必须能够通过传感器来观察，或者，通过执行器施加影响。

**实时性**

嵌入式系统与物理过程相连接，所以嵌入式系统必须严格满足实时性要求。

**反应特性**

（1）嵌入式系统必须时刻准备好，要对周围环境中的异步和周期性事件做出及时反应。

（2）过程和过程控制必须持久处于激活状态

**并行性**

（1）很多的外部过程是并行运行的，因此也产生了对过程控制的相应要求。

（2）一般来说，每个过程对应一个控制组件，因此也产生了控制组件之间的相互协作问题。

（3）组件之间的这种相互协作决定了系统的软件质量

**安全性**

（1）对嵌入式系统的外部环境来说，错误可能具有致命的影响，因此，在结构中，必须设计相应的安全组件和容错组件。

（2）这些组件必须与嵌入式系统中的其他组件一起协同工作。

**互异性**

（1）嵌入式系统不同的组成部分，具有不同的优先级，不同的组成、原理和工作方式。

（2）所以，系统设计必须克服互异性的限制，必要时，可以特殊设计连接器。

#### 嵌入式系统的一般性元结构 (P21)

通过对嵌入式系统特征的讨论，得出以下用于嵌入式系统的一般性元结构，如图所示。

#### 嵌入式系统的元结构：组件类 (P2)

（1）嵌入式系统的组件类由嵌入式系统的共性特征得出

（2）嵌入式系统的组件类由一般性元结构的类推导得出

（3）组件类的名称依据相应的嵌入式系统特征选定

#### 嵌入式系统的元结构：连接器类 (P23)

（1）嵌入式系统的连接器类由嵌入式系统的特征导出，由元结构的类推导得出。

（2）通常，不同类组件或相同类组件之间的耦合存在一个连接器类

#### 嵌入式系统的结构风格 (P39)

结构风格：元结构也称为结构风格，是描述某一特定应用领域中系统组织方式的惯用模式。结构风格定义一个词汇表和一组约束。词汇表中包含一些组件类和连接器类，这组约束指出系统是如何将这些组件和连接器组合起来的。

结构风格反映了领域中众多系统所共有的结构和语义特性，并指导如何将各个模块和子系统有效地组织成一个完整的系统。

结构风格的概念给嵌入式系统的发展带来了很多好处。人们在需求工程阶段弄清嵌入式系统所属的结构风格，然后以这种结构风格对嵌入式系统进行发展。

#### X-By-Wire结构 (P40)

X-By-Wire线控技术：Safety Related Fault Tolerant Systems in Vehicles，汽车的安全相关性容错系统。

By-Wire：“wire”是电线的意思，“by-wire”可理解为线控。

X：X代表汽车中各个系统，代表用于汽车电子控制的系统组件，大的如：发动机、悬架、转向机等，小的如：油门、门锁等。

例如：线控转向Steering-By-Wire，线控制动Brake-By-Wire，线控悬架Suspension-by-wire、线控驱动Drive-by-wire 、线控换挡Shift-by-wire、线控操纵Steer-by-wire。这些组件通过“Wire”相耦合。

线控技术是从应用于飞机驾驶控制上的Fly-By-Wire发展而来。该技术利用传感器将驾驶者输入的信号传递到中央处理器，通过中央处理器的控制逻辑发送信号给相应的执行机构，完成驾驶者的相关操作。这样可取代传统的机械结构，实现对汽车各种运动的电子线控。

X-By-Wire目标：线控技术的最终发展目标是汽车的集成化控制，它将汽车的各个系统相互结合、相互作用、共享传感器的数据，更好地发挥各系统的作用，实现复杂且安全的汽车功能，获得最佳的整车性能，提高车辆的操纵性、稳定性、安全性和智能化，最终实现无人驾驶。

### 第八章 嵌入式软件测试

#### 嵌入式软件测试的目的与特征 (P2)

嵌入式软件测试目的：质量特征的正确性检查。嵌入式软件对正确性有着更高的要求，因为错误会带来巨大的风险。

质量特征通常分为两类：

（1）功能性的质量特征

（2）非功能性的质量特征

应用中，测试技术应具有以下特征：

（1）易于使用，不需要特殊的前提条件。

（2）提供尽可能完整的结果。

（3）提供尽可能可靠的结果。

（4）经济，资金和时间的投入应在可接受的范围内。

（5）灵活，遇到瓶颈时，能够对所进行的工作进行相应的调整。

（6）测试技术所使用的基础设施（如工具）应可用。

#### 软件测试技术的多样性 (P3)

应用领域不同，测试要求也不同：

（1）在可能造成人身伤害的安全性要求严格的软件开发中，测试结果的稳定性比其他目标更重要。

（2）在安全性要求不严格的软件系统中，更看重经济性和实用性。

测试技术有多种，各有侧重，没有哪种测试技术是最优的：

（1）在要求严格的软件开发（如轨道技术和航空技术）中，形式化技术占主导地位。

（2）在常规软件的开发中，动态测试技术、静态分析技术能够更好地折中实现不同的需要。

（3）动态测试更容易实现，考虑所有的测试环境的影响，有专业工具支持。

#### 嵌入式软件测试的特殊性 (P10)

（1）嵌入式软件的动态测试需要考虑操作环境的影响，这对于实时性测试尤为重要，通常使用专门的测试系统：硬件在回路系统。

（2）动态测试是一种随机的方法，所以，对于高可靠性的系统，动态测试几乎不能进行安全性分析。

（3）除动态测试外，嵌入式软件的测试也可以使用静态分析和形式化技术。由于各种测试方法都有不足，所以，在进行要求严格的软件测试时，可以混合使用几种方法。

（4）对于控制软件，除了功能的完整测试外，还需要进行响应时间测试和特定环境功能测试。

#### RTCA/DO-178B标准 (P12)

RTCA：Radio Technical Commission for Aeronautics，美国航空无线电技术委员会。

DO-178B是RTCA提出的一个航空工业的软件标准，是美国联邦航空局（FAA）推荐的软件适航认证指导性文件。

RTCA/DO-178B的标准名称：Software Considerations in Airborne Systems and Equipment Certification，航空系统和设备认证的软件注意事项，用于建立一套开发人员、安装人员和用户在使用计算机技术设计航空系统及设备时遵从的软件要求。

对于所有的机载软件，均适用1992年版的DO-178B标准。作为最老的也是最成熟的软件安全性标准，对其他行业的软件安全性标准影响巨大。

该标准对每一级软件的开发、测试与验证进行了详细描述。

### 第九章 安全性和可靠性分析技术

#### 安全性和可靠性的要求背景 (P2)

在技术范围内，用户通常要求开发者给出证明，以便说明系统的安全性和可靠性是否达到给定的期望目标。

在安全性领域里，新产品上市前，通常应该得到官方的许可。从这个角度说，安全性和可靠性是产品能否在经济上取得成功的一条准则，在这些方面有缺陷，特别是造成系统的停机或事故，会带来高额的赔偿，还会严重损害制造商的形象。

例如：

（1）2003年8月北美大停电事件：波及美国1/4地区，关闭了九座核电站，持续29小时，经济损失大概300亿美元。停电原因：配电网设计不合理。一家发电厂（美国电力私有化）突然出现了故障，电闸自动掉下来，其他电厂马上自动增加发电量进行支援，这些电厂本来就处于饱和状态，由于一下子超负荷运转，电厂全部发生跳闸进行自我保护。结果在短短9秒钟之内，美国7州和加拿大1省出现了灾难性的多米诺效应。

（2）1998年6月德国埃舍德城际高速列车事故：列车脱轨撞上陆桥，数百人丧生，经济损失大概2000万欧元。

（3）2011年7月温州动车追尾脱轨事故：40人丧生，中断行车32小时，直接经济损失约2亿元人民币。

#### 安全性和可靠性的应用领域 (P3)

对安全性和可靠性均有要求的应用领域：

（1）汽车领域，包括交通管理、安全设备，如信号、信号塔等。

（2）航空、航天飞行，包括航线和安全设备。

（3）轨道交通领域。

（4）其他载货、载人设备，如电梯。

（5）核能。

（6）化工行业。

（7）自动化行业、机器人。

（8）医药行业。

在上述领域中，可能会出现对人类或环境的危险。因此，对于那些与安全性相关的系统，在危险事故方面提出了很高的可靠性要求，如，故障率约为109-107/工作小时。

对已有的系统必须证明它达到了安全性和可靠性要求。

#### 方法和技术 (P5)

介绍几种安全性和可靠性的分析方法和技术 ，包括：

（1）危险和可操作性研究（HAZOP）

（2）预先危险分析（PHA）

（3）功能危险分析（FHA）

（4）失效模式效应分析（FMEA）

（5）故障树分析法（FTA）

（6）潜在通路分析（SCA）

...