

- 标准制定：从运行执行任务的方式到所使用的技术，采用标准化定义和约束，从而有效推动信息系统运行相关工作的一致性。
- 资源分配：管理层分配支持信息系统运行的各项能力，包括人力、技术和资源。资源分配应与组织的使命、目标和目的保持一致。
- 过程管理：应测量和管理所有信息系统运行的相关过程，确保过程在时间上和预算目标内被正确和准确地执行。

2. IT 服务管理

IT 服务管理是通过主动管理和流程的持续改进来确保 IT 服务交付有效且高效的一组活动。IT 服务管理由若干不同的活动组成：服务台、事件管理、问题管理、变更管理、配置管理、发布管理、服务级别管理、财务管理、容量管理、服务连续性管理和可用性管理。

(1) 服务台。服务台（Service Desk）是组织体现 IT 服务的重要环节，也是服务干系人体验的重要感知窗口。服务台是服务中与服务干系人沟通和交互的重要界面，负责对服务干系人遇到的问题和需求进行响应和处理；服务台是 IT 服务干系人的“官方”接口和信息发布点，组织内部各个团队之间相互协作的纽带和协调者；服务台对 IT 服务质量及服务干系人体验的管理至关重要，是组织 IT 服务能力持续提升的战略单元。

(2) 事件管理。事件是 IT 服务管理遭遇计划外中断或服务质量出现下降，以及尚未影响服务的配置项故障。事件可能是服务中断、服务速度变慢、软件缺陷以及其他任何组件发生故障。事件管理是 IT 服务中最常见的流程之一，也是 IT 服务必须建立和使用的流程，良好的事件管理必须具备快速解决事件的能力，从而在出现事件时能够尽快恢复服务的正常运作，可以有效提高服务的质量，提升服务干系人满意度。组织应该建立与事件管理过程一致的流程，流程中应该包括：事件受理、分类和初步支持、调查和诊断、解决、进展监控与跟踪、关闭等活动，通过有效执行所定义的活动，能够保障事件响应与处理的效果与效率。

(3) 问题管理。当发生了几个看起来具有相同或相似根本原因的事件时，就会启动问题管理活动。问题管理的总体目标是减少事件的数量和严重性，这种对事件的控制既包括发生事件后的被动性措施，也包括采取主动措施（如：利用系统监控衡量系统运行状况和容量管理等）预防与容量相关的事件发生。与事件管理类似，当确定问题的根本原因时，应制定变更管理和配置管理以进行临时或永久修复。

(4) 变更管理。变更是使一个或更多信息系统配置项的状态发生改变的行动。可见，变更管理的流程更多的是与过程相关，并且重在管理而不是技术，这与事件管理不同，后者建立在技术手段的基础上，强调其管理过程的机械性。变更管理可确保在信息技术环境中执行的所有变更都得到控制和一致化的执行。变更管理的目标是确保使用标准化的方法和程序来高效、及时地处理所有更改，以最大限度地减少与变更相关的事件对服务质量造成的影响，从而改善组织的日常运行。变更管理的主要目的是确保对信息技术环境的所有建议更改都经过适用性和风险管控的审查，并确保变更不会相互干扰，也不会干扰其他计划内或计划外的活动。为了有效，每个干系人都应该审查所有更改，以便正确、全方位地审查每项变更。

(5) 配置管理。配置管理是通过技术或者行政的手段对信息系统的状态进行管理的一系列

活动，这些信息不仅包括信息系统具体配置项信息，还包括这些配置项之间的相互关系。配置项通常包括：硬件详细信息、硬件配置、操作系统版本和配置、软件版本和配置等。配置管理的核心工作是识别、记录、控制、更新配置项信息，主要包含配置管理数据库（Configuration Management Databases, CMDB）的建立以及配置管理数据库准确性的维护，以支持信息系统的正常运行。在IT服务中，配置管理数据库可用于故障定位、问题分析、变更影响度分析、故障分析等，因此，配置管理数据库与真实环境的匹配度和详细度非常重要。

(6) 发布管理。发布管理负责计划和实施信息系统的变更，并且记录该变更的各方面信息。发布是由其实施的变更请求定义的，发布一般是由许多问题修复和IT服务质量改进组成的。发布不仅包括软件方面的变更、硬件方面的变更，同时也包括IT服务管理体系的变更。发布管理通过实施合理的工作程序和严格的监控，保护现有的运营环境和服务不受冲击，负责对软件、硬件、体系发布进行计划、设计、生成、配置和检测，影响范围可能涉及现有的信息系统及其环境、IT用户和组织各分支机构等。

(7) 服务级别管理。服务级别管理就是对IT服务的级别进行定义、记录和管理，并在可接受的成本之下与干系人达成一致的管理过程，通过服务水平协议（Service Level Agreement, SLA）、服务绩效监控和报告的不断循环，持续维护和改进服务质量，以及触发采取行动消除较差服务，从而满足干系人的服务需求。组织需要通过服务目录定义其提供的所有服务和目标。服务目录可被其他文件引用，如SLA，以避免同样的文本和目标被多次重复。服务目录是建立服务干系人预期的关键文件，相关人员都能容易并广泛地获取和阅读。

(8) 财务管理。IT服务财务管理是负责对IT服务运作过程中所有资源进行财务管理的流程，主要活动包括：预算编制、设备投资、费用管理、项目会计和项目投资回报率（Return On Investment, ROI）管理等。财务管理考虑了支持组织目标的IT服务的财务价值。

(9) 容量管理。容量管理用于确认信息系统中有足够的容量满足服务需求。如果信息系统的性能在可接受的范围内，则其具有足够的容量。容量管理不仅仅关注当前需求，还必须考虑未来的需求。容量管理主要活动包括：定期测量、计划变更、战略优化和技术变化等。容量管理由三个子过程组成：业务容量管理、服务容量管理、资源容量管理。

(10) 服务连续性管理。服务连续性管理是一组与组织持续提供服务的能力相关的活动，主要是在发生自然或人为灾难时继续保持服务有效性的活动。服务连续性管理活动分为服务连续性管理的治理、业务影响分析、制订和维护服务连续性计划、测试服务连续性计划、响应与恢复五个过程。

(11) 可用性管理。可用性管理是有关设计、实施、监控、评价和报告IT服务的可用性以确保持续地满足服务干系人的可用性需求的服务管理流程。可用性是指一个组件或一种服务在设定的某个时刻或某段时间内发挥其应有功能的能力，即在约定的服务时段内，IT服务实际能够使用的服务的时间比例。

3. 运行与监控

有效的IT运行要求IT人员按照既定流程和过程理解并正确执行任务。同时，IT运行还强调对人员进行培训，以有效识别异常和错误，并做出正确反应。IT运行的任务常包括：①按照

计划执行作业；②监控作业，并按照优先级为作业分配资源；③重新启动失败的作业和进程；④通过加载或变更备份介质，或通过确保目标的存储系统就绪来优化备份作业；⑤监控信息系统、应用程序和网络的可用性，保证这些系统具备足够的性能；⑥实施空闲期的维护活动，如设备清洁和系统重启等。

IT组织通常制订工作计划，安排定期（每天、每周、每月、每季度等）执行的活动或任务。计划内的活动包括系统承载的活动（如备份）以及人工执行的活动（如访问评审、对账和月末结算）。系统中的计划内活动可以自动或手动调度。大型组织可能具备网络运营中心，也可能具备安全运营中心，这些中心由负责监控相关安全设备、网络、系统和应用程序中的活动的人员组成。在IT运行环境中发生的异常和错误，通常按照IT服务管理体系中的事件管理和问题管理流程进行处理。

1) 运行监控

IT团队应对信息系统、应用程序和基础设施进行监控，以确保它们继续按要求运行。监控工具和系统使IT运行人员能够检测软件或硬件组件何时未按计划运行等。检测和报告的错误类型包括：系统错误、程序错误、通信错误和操作员错误等。IT团队应记录任何意外或异常活动的事件，并基于流程对事件进行管理。

2) 安全监控

组织需要执行不同类型的安全监控，并把安全监控作为其整体策略的一部分，以预防和响应安全事件。组织可能执行的监控类型包括：防火墙策略规则中的例外情况、入侵防御系统的告警、数据丢失防护系统的告警、云安全访问代理的告警、用户访问管理系统的告警、网络异常的告警、网页内容过滤系统的告警、终端管理系统的告警（含反恶意软件）、供应商发布的安全公告、第三方发布的安全公告、威胁情报咨询、门禁系统的告警和视频监控系统的告警等。

4. 终端侧管理

IT团队职能的一个关键环节是它向组织人员提供的服务，以改善他们对IT访问和使用的情况。组织通常使用IT管理工具来促进对用户终端计算机的高效和一致的管理。一般来说，最终用户计算机是“锁定”的，这限制了最终用户可能在其设备上执行的配置更改的数量和类型，包括操作系统配置、补丁安装、软件程序安装、使用外部数据存储设备等，最终用户可能会将此类限制视为不便。但是，这些限制不仅有助于确保最终用户的设备和整个组织的IT环境具有更高的安全性，而且还促进了更高的一致性，从而降低了支持成本。

5. 程序库管理

程序库是组织用来存储和管理应用程序源代码和目标代码的工具。在大多数组织中，应用程序源代码非常敏感。它可能被视为知识产权，并且可能包含算法、加密密钥和其他敏感信息，这些信息应由尽可能少的人员访问。应用程序源代码应被视为信息，并通过组织的安全策略和数据分类策略进行管理。程序库的控制使组织能够对其应用程序的完整性、质量和安全性进行高度控制。程序库通常作为具有用户界面和多种功能的信息系统存在，其中主要功能包括：访问控制、程序签出、程序签入、版本控制和代码分析等。

6. 安全管理

信息安全管理可确保组织的信息安全计划充分识别和解决风险，并在整个运维和服务过程中正常运行。该领域的管理要点详见 4.2.3 节。

7. 介质控制

组织需要采取一系列活动，以确保数字介质得到适当管理，包括对其保护以及销毁不再需要的数据。这些过程通常与数据保留和数据清除过程相关联，以便通过物理和逻辑的安全控制充分保护所需的数据，同时有效丢弃和擦除不再需要的数据。处置不再需要的介质相关的程序，包括擦除该介质上的数据或使该介质上的数据无法以其他方式恢复的所有相关步骤。组织应考虑包含在介质管理、销毁策略和程序范围内的介质主要包括：备份介质、虚拟磁带库、光学介质、硬盘驱动器、固态驱动器、闪存、硬拷贝等。介质清理的策略和程序需要包含在服务提供商的相关要求中，以及记录保存活动以跟踪介质随时间推移的销毁情况。

8. 数据管理

数据管理是与数据的获取、处理、存储、使用和处置相关的一组活动。该领域管理要点见 4.2.1 节。

4.1.5 优化和持续改进

优化和持续改进是信息系统管理活动中的一个环节，良好的优化和持续改进管理活动能够有效保障信息系统的性能和可用性等，延长整体系统的有效使用周期。传统上，优化和持续改进常用的方法为戴明环，即 PDCA 循环。PDCA 循环是将持续改进分为四个阶段，即 Plan（计划）、Do（执行）、Check（检查）和 Act（处理）。

优化和持续改进基于有效的变更管理，使用六西格玛倡导的五阶段方法 DMAIC/DMADV，是对戴明环四阶段周期的延伸，包括：定义（Define）、度量（Measure）、分析（Analysis）、改进/设计（Improve/Design）、控制/验证（Control/Verify）。当第四阶段的“改进”替换为“设计”，“控制”替换为“验证”时，五阶段法就从 DMAIC 转变为 DMADV。

1. 定义阶段

定义阶段的目标包括待优化信息系统定义、核心流程定义和团队组建。

(1) 待优化信息系统定义。该活动关注定义协同的范围、优化目标和目的、系统团队成员和出资人，以及优化时间表和交付成果。待优化信息系统范围与关键业务实践、服务对象交互有关，该定义需要了解信息系统相关的业务。可使用“延伸目标”概念来定义待优化的信息系统。延伸目标是那些超出当前组织结构、资源和技术可预见范围的优化目标。可以帮助超越渐进式改进，重新思考信息系统相关业务、运行或流程，以达到可以实现重大改进的程度。

(2) 核心流程定义。该活动关注定义利益干系人、投入和产出以及广泛的功能。SIPOC (Supplier、Input、Process、Output、Customer) 分析是定义核心流程视图的首选工具。任何一个组织都是一个由提供人、输入、流程、输出，还有服务对象这样相互关联、互动的 5 个部分组成的系统。

(3) 团队组建。该活动重点关注从关键利益干系人群体中确定人员组建高能力团队，对信息系统的问题和收益达成共识。有效的团队形成对于建立利益干系人的支持至关重要。从每个关键利益干系人群体中选出可靠的团队成员，以代表他们在优化和持续改进中的职能或领域。有效的团队通常限制为5~7名参与者。较大的团队更难管理，成员可能会失去对团队的责任感。其他团队成员可能是来自非关键利益干系人组的临时成员，他们仅在需要时参与，例如需要流程专业知识时。

2. 度量阶段

度量阶段目标包括流程定义、指标定义、流程基线和度量系统分析。

(1) 流程定义。流程定义通常使用流程图工具定义度量阶段的流程，以图形方式实现给定信息系统的输入、操作和输出。流程图的目的是帮助人们理解流程，应当尽可能简单，但又不能太简单。当流程图指示太多的决策点时，通常表示可能出现了一个过于复杂的过程，可能会出错。因此，决策点恰恰是信息系统优化的一个潜在改进重点。

(2) 指标定义。待优化信息系统的定义包括将用于评估流程的指标。选择能够切实提高系统质量、业务绩效和服务对象满意度的指标非常重要。正确选择的指标将为基于数据的决策提供输入，并将成为用于描述信息系统状态的标准化和数据化的语言。度量指标一旦建立，可用于确定影响信息系统的各种因素及其相对重要性，并可比较信息系统不同组件对业务的整体贡献。指标为信息系统的持续改进提供了对质量、成本和进度的重要描述。如何衡量和报告这些情况，以及这些分别对质量敏感、成本敏感和进度敏感的指标，如何与信息系统的关键流程变量和控制相关联，以实现系统范围的持续改进。

(3) 流程基线。当明确了度量指标之后，必须通过基线确定现有系统的能力，以确定当前系统在多大程度上较好地满足了服务对象的要求，并验证定义阶段中确立的信息系统目标达成情况。当系统处于控制优化状态时，可以统计其系统能力，将统计出的系统变异与明确的服务对象要求进行比较。只有在使用基线清晰描述了系统稳定性之后，才能评估系统变异，只有稳定的系统才能预测。当系统指标数据不稳定或不在控制优化中时，可以使用系统性能指标作为粗略估计，将给定周期内观察的系统变化与服务对象要求进行比较。

(4) 度量系统分析。质量始于度量。只有当质量被量化时，才能开始讨论优化和持续改进。度量是根据某些规则将数值分配给被观察到的现象。在对信息系统进行优化和持续改进过程中，需要十分注意度量水平、度量的可靠性与有效性问题。一个良好的度量系统具备特性可包括：

- 准确：应该产生一个“接近”被测量的实际属性的数值。
- 可重复：如果测量系统反复应用于同一物体，则产生的测量价值应彼此接近。
- 线性：测量系统应能够在整个关注范围内产生准确和一致的结果。
- 可重现：当任何经过适当培训的个人使用时，测量系统应产生相同的结果。
- 稳定：应用于相同的项目时，测量系统将来应产生与过去相同的结果。

3. 分析阶段

分析阶段的三个目标包括价值流分析、信息系统异常的源头分析和确定优化改进的驱动因素。

(1) 价值流分析。价值流分析首先定义信息系统使用者眼中相关产品或服务的价值。价值也可以定义为：①组织愿意投资的系统组件；②改变信息系统形式、适合度或功能的活动；

③将业务输入经信息系统转换为输出的活动。

(2) 信息系统异常的源头分析。度量阶段的信息系统异常的来源，提供了信息系统稳定（即控制中）或不稳定（即失控）的证据。首先正确区分这两种类型的变异至关重要，因为每种变异的改进策略不同。对于稳定的信息系统，只有通过对系统进行根本性的更改，才能减少系统内置的常见变异原因。当系统失控时，则必须解决并消除在特定时间段内造成不稳定情况的特殊原因，重新获得稳定的过程，然后可以进行改进。在业务层面，可以分析服务对象数据，以建立服务对象满意度与用于支撑服务对象体验的信息系统组件之间的关系。

(3) 确定优化改进的驱动因素。优化改进的驱动因素是指对信息系统优化影响最大的因素。对于任何信息系统，都可能有许多因素会导致其功能和性能的变化。信息系统改进需要减少其系统或组件的异常，或者将系统衡量的中位线移动到更有利的设置。无论哪种情况，专注于关键的优化改进驱动因素都将有助于信息系统的优化和持续改进。在确定优化改进的驱动因素时，可以使用一些数学分析方法，计算确定关键驱动因素，这些数学分析方法包括相关性与回归分析、最小二乘拟合和残差分析等。

4. 改进 / 设计阶段

改进 / 设计阶段的目标包括：①向发起人提出一个或多个解决方案；量化每种方法的收益；就解决方案达成共识并实施。②定义新的操作 / 设计条件。③为新工艺 / 设计提供定义和缓解故障模式。

(1) 改进 / 设计的解决方案推进。改进 / 设计阶段解决方案的部署可以缩小信息系统当前状态与所需状态之间的差距。实施的方法也必须在此阶段进行验证，以确保达到并保持预期的效果。这个阶段定义了改进和成本降低的相关计划。它通常是成败点，需要团队考虑之前未考虑的因素，并成为变革的真正推动者。此时的管理支持至关重要。

(2) 定义新的操作 / 设计条件。定义阶段中引入的核心流程可用于开发新流程，还可以进行其他实验设计，以确定新信息系统或新系统中新的功能和设计所需的最佳操作条件，以最大或最小化响应。

(3) 定义和缓解故障模式。建立了信息系统的优化和持续改进流程之后，可以评估其故障模式。了解信息系统的故障模式使组织能够定义不同故障的缓解策略，以最大限度地减少故障的影响或发生。这些缓解策略可能会导致新的运行维护过程步骤、最优系统设置或控制策略，以防止信息系统失效；可能是提升信息系统性能，降低信息系统容量损耗。在某些情况下，无法预防故障的情况下，可以制定一种策略来最大限度地减少故障的发生并控制损失。

5. 控制 / 验证阶段

控制 / 验证阶段的目标包括标准化新程序 / 新系统功能的操作控制要素、持续验证优化的信息系统的可交付成果、记录经验教训。

(1) 标准化新程序 / 新系统功能的操作控制要素。当信息系统得到改进，组织需要更好地控制系统，保持进一步改进的能力。管理者必须对改进形成的新方法、新系统运行进行标准化，以维持改进带来的效益。标准化的业务层面控制是保持信息系统优化改进的方法。培训对新系统或优化系统的操作控制能力，是维护已部署改进的关键。

(2) 持续验证优化的信息系统的可交付成果。组织应当将变更的系统组件信息、信息系统状态趋势等内容，对受影响的人员开展培训。当这些人员不仅了解信息系统如何变化，还应了解其产生的原因，以及可能在未来找到进一步改进的方法。

(3) 记录经验教训。随着项目小组完成其活动，必须最终确定和保留项目文档。其中一个关键方面是记录经验教训，如为了更快或更好的结果，可能会做些什么事情。经验对组织中的其他团队有用吗？这种团队总结的另一个重要作用是对他们努力的认可。

4.2 管理要点

信息系统管理涉及系统准备、设计、实施、运行等活动的众多方面，管理重点范围和细致程度随各组织的战略和业务目标的不同而存在差异。从日常管理活动视角来看，各组织关注的管理内容主要聚焦在数据管理、运维管理和信息安全管理等方面的体系化管理。

4.2.1 数据管理

在通常情况下，数据管理是指通过规划、控制与提供数据和信息资产的职能，包括开发、执行和监督有关数据的计划、策略、方案、项目、流程、方法和程序，以获取、控制、保护、交付和提高数据和信息资产价值。国际数据管理协会（DAMA）指出，数据资源管理致力于发展和处理组织数据全生命周期的适当的构建、策略、实践和程序。数据管理框架是对组织的管理平台或者能够产生业务数据的平台产生的数据进行统一的跟踪、协调、管理的功能模型。本部分主要讨论数据管理的基本框架、主要活动和管理要点，使用工程的方法，开展组织数据能力建设与实施，见本书 5.2 节的数据工程。

数据管理能力成熟度评估模型（Data Management Capability Maturity Assessment Model, DCMM）是国家标准 GB/T 36073《数据管理能力成熟度评估模型》中提出的，旨在帮助组织利用先进的数据管理理念和方法，建立和评价自身数据管理能力，持续完善数据管理组织、程序和制度，充分发挥数据在促进组织向信息化、数字化、智能化发展方面的价值，如图 4-6 所示。

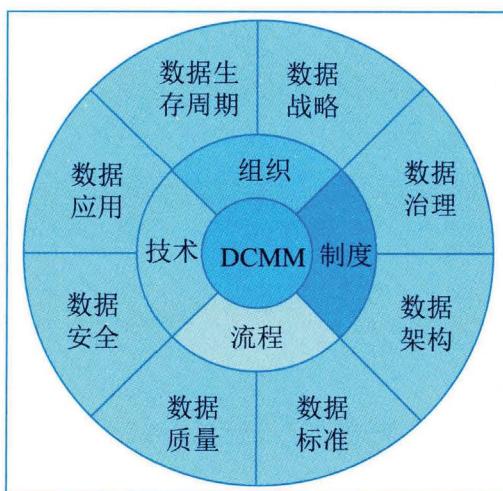


图 4-6 数据管理能力模型

DCMM 定义了数据战略、数据治理、数据架构、数据应用、数据安全、数据质量、数据标准和数据生存周期 8 个核心能力域。

1. 数据战略

组织的数据战略能力域通常包括数据战略规划、数据战略实施和数据战略评估三个能力项。

(1) 数据战略规划。数据战略规划是在组织所有利益相关者之间达成共识的结果。从宏观及微观两个层面确定开展数据管理及应用的动因，并综合反映数据提供方和消费方的需求。数据战略规划主要活动和工作要点包括：

- 识别利益相关者：明确利益相关者的需求。
- 数据战略需求评估：组织对业务和信息化现状进行评估，了解业务和信息化对数据的需求。
- 数据战略制定：主要包括①愿景陈述，其中包含数据管理原则、目的和目标；②规划范围，其中包含重要业务领域、数据范围和数据管理优先权；③所选择的数据管理模型和建设方法；④当前数据管理存在的主要差距；⑤管理层及其责任，以及利益相关者名单；⑥编制数据管理规划的管理方法；⑦持续优化路线图。
- 数据战略发布：以文件、网站、邮件等方式正式发布审批后的数据战略。
- 数据战略修订：根据业务战略、信息化发展等方面的要求，定期进行数据战略的修订。

(2) 数据战略实施。数据战略实施是组织完成数据战略规划后，逐渐实现数据职能框架的过程。实施过程中依据组织数据管理和数据应用的现状，确定与愿景、目标之间的差距；依据数据职能框架制定阶段性数据任务目标，并确定实施步骤。数据战略实施主要活动和工作要点包括：

- 评估准则：建立数据战略规划实施评估标准，规范评估过程和方法。
- 现状评估：对组织当前数据战略落实情况进行分析，评估各项工作开展情况。
- 评估差距：根据现状评估结果与组织数据战略规划进行对比，分析存在的差异。
- 实施路径：利益相关者结合组织的共同目标和实际业务价值进行数据职能任务优先级排序。
- 保障计划：依据实施路径，制定开展各项活动所需的预算。
- 任务实施：根据任务开展工作。
- 过程监控：依据实施路径，及时对实施过程进行监控。

(3) 数据战略评估。组织在数据战略评估过程中需要建立对应的业务案例和投资模型，并在整个数据战略实施过程中跟踪进度，同时做好记录供审计和评估使用。数据战略评估主要活动和工作要点包括：

- 建立任务效益评估模型：从时间、成本、效益等方面建立数据战略相关任务的效益评估模型。
- 建立业务案例：建立基本的用例模型、项目计划、初始风险评估和项目描述，能确定数据管理和数据应用相关任务（项目）的范围、活动、期望的价值以及合理的成本收益分析。

- 建立投资模型：作为数据职能项目投资分析的基础性理论，投资模型确保在充分考虑成本和收益的前提下对所需资本合理分配，投资模型要满足不同业务的信息科技需求以及对应的数据职能内容，同时要广泛沟通以保障对业务或技术的前瞻性支持，并符合相关的监管及合规性要求。
- 阶段评估：在数据工作开展过程中，定期从业务价值、经济效益等维度对已取得的成果进行效益评估。

2. 数据治理

组织的数据治理能力域通常包括数据治理组织、数据制度建设和数据治理沟通三个能力项。

(1) 数据治理组织。数据治理组织需要包括组织架构、岗位设置、团队建设、数据责任等内容，它是各项数据职能工作开展的基础。数据治理组织对组织在数据管理和数据应用行使职责规划和控制，并指导各项数据职能的执行，以确保组织能有效落实数据战略目标。数据治理组织主要活动和工作要点包括：

- 建立数据治理组织：建立数据体系配套的权责明确且内部沟通顺畅的组织，确保数据战略的实施。
- 岗位设置：建立数据治理所需的岗位，明确岗位的职责、任职要求等。
- 团队建设：制订团队培训、能力提升计划，通过引入内部、外部资源定期开展人员培训，提升团队人员的数据治理技能。
- 数据归口管理：明确数据所有人、管理人等相关角色以及数据的归口的具体管理人员。
- 建立绩效评价体系：根据团队人员职责、管理数据范围的划分，制定相关人员的绩效考核体系。

(2) 数据制度建设。为保障数据管理和数据应用各项功能的规范化运行，组织需要建立对应的制度体系。数据制度体系通常分层次设计，遵循严格的发布流程并定期检查和更新。数据制度建设是数据管理和数据应用各项工作有序开展的基础，是数据治理沟通和实施的依据。数据制度建设主要活动和工作要点包括：

- 制定数据制度框架：根据数据职能的层次和授权决策次序，数据制度框架可分为策略、办法、细则三个层次，该框架规定了数据管理和数据应用的具体领域、各个数据职能领域内的目标、遵循的行动原则、完成的明确任务、实行的工作方式、采取的一般步骤和具体措施等。
- 整理数据制度内容：数据管理策略与数据管理办法、数据管理细则共同构成组织数据制度体系，其基本内容包括：①数据策略说明数据管理和数据应用的目的，明确其组织与范围；②数据管理办法是为数据管理和数据应用各领域内活动开展而规定的相关规则和流程；③数据管理细则是为确保各数据方法执行落实而制定的相关文件。
- 数据制度发布：组织内部通过文件、邮件等形式发布审批通过的数据制度。
- 数据制度宣贯：定期开展数据制度相关的培训与宣传工作。
- 数据制度实施：结合数据治理组织的设置，推动数据制度的落地实施。

(3) 数据治理沟通。数据治理沟通旨在确保组织内全部利益相关者都能及时了解相关策略、

标准、流程、角色、职责、计划的最新情况，开展数据管理和应用相关的培训，掌握数据管理相关的知识和技能。数据治理沟通旨在建立与提升跨部门及部门内部数据管理能力，提升数据资产意识，构建数据文化。数据治理沟通主要活动和工作要点包括：

- 沟通路径：明确数据管理和应用的利益相关者，分析各方的诉求，了解沟通的重点内容。
- 沟通计划：建立定期或不定期沟通计划，并在利益相关者之间达成共识。
- 沟通执行：按照沟通计划安排实施具体沟通活动，同时对沟通情况记录。
- 问题协商机制：包括引入高层管理者等方式，以解决分歧。
- 建立沟通渠道：在组织内部明确沟通的主要渠道，例如邮件、文件、网站、自媒体、研讨会等。
- 制订培训宣贯计划：根据组织人员和业务发展需要，制订相关的培训宣贯计划。
- 开展培训：根据培训计划的要求，定期开展相关培训。

3. 数据架构

组织的数据架构能力域通常包括数据模型、数据分布、数据集成与共享和元数据管理四个能力项。

(1) 数据模型。数据模型是使用结构化的语言将收集到的组织业务运行、管理和决策中使用的数据需求进行综合分析，按照模型设计规范将需求重新组织。数据模型主要活动和工作要点包括：

- 收集和理解组织的数据需求：包括收集和分析组织应用系统的数据需求和实现组织的战略，满足内外部监管，与外部组织互联互通等的数据需求等。
- 制定模型规范：包括模型的管理工具、命名规范、常用术语以及管理方法等。
- 开发数据模型：包括开发设计组织级数据模型、系统应用级数据模型。
- 数据模型应用：根据组织级数据模型的开发，指导和规范系统应用级数据模型的建设。
- 符合性检查：检查组织级数据模型和系统应用级数据模型的一致性。
- 模型变更管理：根据需求变化实时地对数据模型进行维护。

(2) 数据分布。数据分布能力域是针对组织级数据模型中数据的定义，明确数据在系统、组织和流程等方面的分布关系，定义数据类型，明确权威数据源，为数据相关工作提供参考和规范。通过数据分布关系的梳理，定义数据相关工作的优先级，指定数据的责任人，并进一步优化数据的集成关系。数据分布主要活动和工作要点包括：

- 数据现状梳理：对应用系统中的数据进行梳理，了解数据的作用，明确存在的数据问题。
- 识别数据类型：将组织内的数据根据其特征分类管理，一般类型包括主数据、参考数据、交易数据、统计分析数据、文档数据、元数据等类型。
- 数据分布关系梳理：根据组织级数据模型的定义，结合业务流程梳理的成果，定义组织中数据和流程、数据和组织机构、数据和系统的分布关系。
- 梳理数据的权威数据源：对每类数据明确相对合理的唯一信息采集和存储系统。
- 数据分布关系的应用：根据数据分布关系的梳理，对组织数据相关工作进行规范，包括

定义数据工作优先级、优化数据集成等。

- 数据分布关系的维护和管理：根据组织中业务流程和系统建设的情况，定期维护和更新组织中的数据分布关系，保持及时性。

(3) 数据集成与共享。数据集成与共享职能域是建立起组织内各应用系统、各部门之间的集成共享机制，通过组织内部数据集成共享相关制度、标准、技术等方面管理，促进组织内部数据的互联互通。数据集成与共享主要活动和工作要点包括：

- 建立数据集成共享制度：指明数据集成共享的原则、方式和方法。
- 形成数据集成共享标准：依据数据集成共享方式不同，制定不同的数据交换标准。
- 建立数据集成共享环境：将组织内多种类型的数据整合在一起，形成对复杂数据加工处理和便捷访问的环境。
- 建立对新建系统的数据集成方式的检查。

(4) 元数据管理。元数据管理是关于元数据的创建、存储、整合与控制等一套流程的集合。元数据管理主要活动和工作要点包括：

- 元模型管理：对包含描述元数据属性定义的元模型进行分类并定义每一类元模型，元模型可采用或参考相关国家标准。
- 元数据集成和变更：基于元模型对元数据进行收集，对不同类型、不同来源的元数据进行集成，形成对数据描述的统一视图，并基于规范的流程对数据的变更进行及时更新和管理。
- 元数据应用：基于数据管理和数据应用需求，对于组织管理的各类元数据进行分析应用，如查询、血缘分析、影响分析、符合性分析、质量分析等。

4. 数据应用

数据应用能力域通常包括数据分析、数据开放共享和数据服务三个能力项。

(1) 数据分析。数据分析是对组织各项经营管理活动提供数据决策支持而进行的组织内外部数据分析或挖掘建模，以及对应成果的交付运营、评估推广等活动。数据分析能力会影响到组织制定决策、创造价值、向用户提供价值的方式。数据分析主要活动和工作要点包括：

- 常规报表分析：按照规定的格式对数据进行统一的组织、加工和展示。
- 多维分析：各分类之间的数据度量之间的关系，从而找出同类性质的统计项之间数学上的联系。
- 动态预警：基于一定的算法、模型对数据进行实时监测，并根据预设的阈值进行预警。
- 趋势预报：根据客观对象已知的信息对事物在将来的某些特征、发展状况的一种估计、测算活动，运用各种定性和定量的分析理论与方法，对发展趋势进行预判。

(2) 数据开放共享。数据开放共享是指按照统一的管理策略对组织内容的数据进行有选择的对外开放，同时按照管理策略引入外部数据供组织内部使用。数据开放共享是实现数据跨组织、跨行业流转的重要前提，也是数据价值最大化的基础。数据开放共享主要活动和工作要点包括：

- 梳理开放共享数据：组织需要对其开放共享的数据进行全面的梳理，建立清晰的开放共享数据目录。

- 制定外部数据资源目录：对组织需要的外部数据进行统一梳理，建立数据目录，方便内部用户的查询和应用。
- 建立统一的数据开放共享策略：包括安全、质量等内容。
- 数据提供方管理：建立对外数据使用策略、数据提供方服务规范等。
- 数据开放：组织可通过各种方式对外开放数据，并保证开放数据的质量。
- 数据获取：按照数据需求进行数据提供方的选择。

(3) 数据服务。数据服务是通过对组织内外部数据的统一加工和分析，结合公众、行业和组织的需要，以数据分析结果的形式对外提供跨领域、跨行业的数据服务。数据服务是数据资产价值变现最直接的手段，也是数据资产价值衡量的方式之一，通过良好的数据服务对内提升组织的效益，对外更好的服务公众和社会。数据服务的提供可能有多种形式，包括数据分析结果、数据服务调用接口、数据产品或数据服务平台等，具体服务的形式取决于组织数据的战略和发展方向。数据服务主要活动和工作要点包括：

- 数据服务需求分析：需要有数据分析团队来分析外部的数据需求，并结合外部的需求提出数据服务目标和展现形式，形成数据服务需求分析文档。
- 数据服务开发：数据开发团队根据数据服务需求分析对数据进行汇总和加工，形成数据产品。
- 数据服务部署：部署数据产品，对外提供服务。
- 数据服务监控：能对数据服务有全面的监控和管理，实时分析数据服务的状态、调用情况、安全情况等。
- 数据服务授权：对数据服务的用户进行授权，并对访问过程进行控制。

5. 数据安全

组织的数据安全能力域通常包括数据安全策略、数据安全管理、数据安全审计三个能力项。

(1) 数据安全策略。数据安全策略是数据安全的核心内容，在制定的过程中需要结合组织管理需求、监管需求以及相关标准等统一制定。数据安全策略主要活动和工作要点包括：

- 了解国家、行业等监管需求，并根据组织对数据安全的业务需要，进行数据安全策略规划，建立组织的数据安全管理策略。
- 制定适合组织的数据安全标准，确定数据安全等级及覆盖范围等。
- 定义组织数据安全管理的目标、原则、管理制度、管理组织、管理流程等，为组织的数据安全管理提供保障。

(2) 数据安全管理。数据安全管理是在数据安全标准与策略的指导下，通过对数据访问的授权、分类分级的控制、监控数据的访问等进行数据安全的管理工作，满足数据安全的业务需要和监管需求，实现组织内部对数据生存周期的数据安全管理。数据安全管理主要活动和工作要点包括：

- 数据安全等级的划分：根据组织数据安全标准，充分了解组织数据安全管理需求，对组织内部的数据进行等级划分并形成相关文档。
- 数据访问权限控制：制定数据安全管理的利益相关者清单，围绕利益相关者需求，对其

数据访问、控制权限进行授权。

- 用户身份认证和访问行为监控：在数据访问过程中对用户的身份进行认证识别，对其行为进行记录和监控。
- 数据安全的保护：提供数据安全保护控制相关的措施，保证数据在应用过程中的隐私性。
- 数据安全风险管理：对组织已知或潜在的数据安全进行分析，制定防范措施并监督落实。

(3) 数据安全审计。数据安全审计是一项控制活动，负责定期分析、验证、讨论、改进数据安全管理相关的策略、标准和活动。审计工作可由组织内部或外部审计人员执行，审计人员应独立于审计所涉及的数据和流程。数据安全审计的目标是为组织以及外部监管机构提供评估和建议。数据安全审计主要活动和工作要点包括：

- 过程审计：分析实施规程和实际做法，确保数据安全目标、策略、标准、指导方针和预期结果相一致。
- 规范审计：评估现有标准和规程是否适当，是否与业务要求和技术要求相一致。
- 合规审计：检索和审阅组织相关监管法规要求，验证其是否符合监管法规要求。
- 供应商审计：评审合同、数据共享协议，确保供应商切实履行数据安全义务。
- 审计报告发布：向高级管理人员、数据管理专员以及其他利益相关者报告组织内的数据安全状态。
- 数据安全建议：提出数据安全的设计、操作和合规等方面的工作建议。

6. 数据质量

组织的数据质量能力域通常包括数据质量需求、数据质量检查、数据质量分析和数据质量提升四个能力项。

(1) 数据质量需求。数据质量需求是明确数据质量目标，并根据业务需求及数据要求制定用来衡量数据质量的规则，包括衡量数据质量的技术指标、业务指标以及相应的校验规则与方法。数据质量需求是度量和管理数据质量的依据，需要依据组织的数据管理目标、业务管理的需求和行业的监管需求并参考相关标准来统一制定与管理。数据质量需求主要活动和工作要点包括：

- 定义数据质量管理目标：依据组织管理的需求，参考外部监管的要求，明确组织数据质量管理目标。
- 定义数据质量评价维度：依据组织数据质量管理的目标，制定组织数据质量评估维度，指导数据质量评价工作的开展。
- 明确数据质量管理范围：依据组织业务发展的需求以及常见数据问题的分析，明确组织数据质量管理的范围，梳理各类数据的优先级以及质量需求。
- 设计数据质量规则：依据组织的数据质量管理需求及目标，识别数据质量特性，定义各类数据的质量评价指标、校验规则与方法，并根据业务发展需求及数据质量检查分析结果对数据质量规则进行持续维护与更新。

(2) 数据质量检查。数据质量检查是根据数据质量规则中的有关技术指标和业务指标、校验规则与方法对组织的数据质量情况进行实时监控，从而发现数据质量问题，并向数据管理人员进行反馈。数据质量检查主要活动和工作要点包括：

- 制订数据质量检查计划：根据组织数据质量管理目标的需要，制订统一的数据质量检查计划。
- 数据质量情况剖析：根据计划对系统中的数据进行剖析，查看数据的值域分布、填充率、规范性等，切实掌握数据质量实际情况。
- 数据质量校验：依据预先配置的规则、算法，对系统中的数据进行校验。
- 数据质量问题管理：包括问题记录、问题查询、问题分发和问题跟踪。

(3) 数据质量分析。数据质量分析是对数据质量检查过程中发现的数据质量问题及相关信息进行分析，找出影响数据质量的原因，并定义数据质量问题的优先级，作为数据质量提升的参考依据。数据质量分析主要活动和工作要点包括：

- 数据质量分析方法和要求：整理组织数据质量分析的常用方法，明确数据质量分析的要求。
- 数据质量问题分析：深入分析数据质量问题产生的根本原因，为数据质量提升提供参考。
- 数据质量问题影响分析：根据数据质量问题的描述以及数据价值链的分析，评估数据质量对于组织业务开展、应用系统运行等方面的影响，形成数据质量问题影响分析报告。
- 数据质量分析报告：包括对数据质量检查、分析等过程累积的各种信息进行汇总、梳理、统计和分析。
- 建立数据质量知识库：收集各类数据质量案例、经验和知识，形成组织的数据质量知识库。

(4) 数据质量提升。数据质量提升是对数据质量分析的结果，制定、实施数据质量改进方案，包括错误数据更正、业务流程优化、应用系统问题修复等，并制定数据质量问题预防方案，确保数据质量改进的成果得到有效保持。数据质量提升主要活动和工作要点包括：

- 制定数据质量改进方案：根据数据质量分析的结果，制定数据质量提升方案。
- 数据质量校正：采用数据标准化、数据清洗、数据转换和数据整合等手段和技术，对不符合质量要求的数据进行处理，并纠正数据质量问题。
- 数据质量跟踪：记录数据质量事件的评估、初步诊断和后续行动等信息，验证数据质量提升的有效性。
- 数据质量提升：对业务流程进行优化，对系统问题进行修正，对制度和标准进行完善，防止将来同类问题的发生。
- 数据质量文化：通过数据质量相关培训、宣贯等活动，持续提升组织数据质量意识，建立良好的数据质量文化。

7. 数据标准

组织的数据标准能力域通常包括业务术语、参考数据和主数据、数据元和指标数据四个能力项。

(1) 业务术语。业务术语是组织中业务概念的描述，包括中文名称、英文名称、术语定义等内容。业务术语数据管理就是制定统一的管理制度和流程，并对业务术语的创建、维护和发布进行统一的管理，进而推动业务术语的共享和组织内部的应用。业务术语是组织内部理解数据、应用数据的基础。通过对业务术语的管理能保证组织内部对具体技术名词理解的一致性。业务术语主要活动和工作要点包括：

- 制定业务术语标准：同时制定业务术语管理制度，包含组织、人员职责、应用原则等。
- 业务术语字典：组织中已定义并审批和发布的术语集合。
- 业务术语发布：业务术语变更后及时进行审批并通过邮件、网站、文件等形式进行发布。
- 业务术语应用：在数据模型建设、数据需求描述、数据标准定义等过程中引用业务术语。
- 业务术语宣贯：组织内部介绍、推广已定义的业务术语。

(2) 参考数据和主数据。参考数据和主数据是用于将其他数据进行分类的数据。参考数据管理是对定义的数据值域进行管理，包括标准化术语、代码值和其他唯一标识符，每个取值的业务定义，数据值域列表内部和跨不同列表之间的业务关系的控制，并对相关参考数据的一致、共享使用。主数据是组织中需要跨系统、跨部门共享的核心业务实体数据。主数据管理是对主数据标准和内容进行管理，实现主数据跨系统的一致、共享使用。参考数据和主数据主要活动和工作要点包括：

- 定义编码规则：定义参考数据和主数据唯一标识的生成规则。
- 定义数据模型：定义参考数据和主数据的组成部分及其含义。
- 识别数据值域：识别参考数据和主数据取值范围。
- 管理流程：创建参考数据和主数据管理相关流程。
- 建立质量规则：检查参考数据和主数据相关的业务规则和管理要求，建立参考数据和主数据相关的质量规则。
- 集成共享：参考数据、主数据和应用系统的集成。

(3) 数据元。通过对组织中核心数据元的标准化，可以使数据的拥有者和使用者对数据有一致的理解。数据元主要活动和工作要点包括：

- 建立数据元的分类和命名规则：根据组织的业务特征建立数据元的分类规则，制定数据元的命名、描述与表示规范。
- 建立数据元的管理规范：建立数据元管理的流程和岗位，明确管理岗位职责。
- 数据元的创建：建立数据元的创建方法，进行数据元的识别和创建。
- 建立数据元的统一目录：根据数据元的分类及业务管理需求，建立数据元管理的目录，对组织内部的数据元分类存储。
- 数据元的查找和引用：提供数据元查找和引用的在线工具。
- 数据元的管理：提供对数据元以及数据元目录的日常管理。
- 数据元管理报告：根据数据元标准定期进行引用情况分析，了解各应用系统中对数据元

的引用情况，促进数据元的应用。

(4) 指标数据。指标数据是组织在经营分析过程中衡量某一个目标或事物的数据，一般由指标名称、时间和数值等组成，指标数据管理指组织对内部经营分析所需要的指标数据进行统一规范化定义、采集和应用，用于提升统计分析的数据质量。指标数据主要活动和工作要点包括：

- 根据组织业务管理需求，制定组织内指标数据分类管理框架，保证指标分类框架的全面性和各分类之间的独立性。
- 定义指标数据标准化的格式，梳理组织内部的指标数据，形成统一的指标字典。
- 根据指标数据的定义，由相关部门或应用系统定期进行数据的采集、生成。
- 对指标数据进行访问授权，并根据用户需求进行数据展现。
- 对指标数据采集、应用过程中的数据进行监控，保证指标数据的准确性、及时性。
- 划分指标数据的归口管理部门、管理职责和管理流程，并按照管理规定对指标标准进行维护和管理。

8. 数据生存周期

组织的数据生存周期能力域通常包括数据需求、数据设计和开发、数据运维和数据退役四个能力项。

(1) 数据需求。数据需求是指组织对业务运营、经营分析和战略决策过程中产生和使用数据的分类、含义、分布和流转的描述。数据需求管理过程识别所需的数据，确定数据需求优先级并以文档的方式对数据需求进行记录和管理。数据需求主要活动和工作要点包括：

- 建立数据需求管理制度：明确组织数据需求的管理组织、制度和流程。
- 收集数据需求：需求人员通过各种方式分析数据应用场景，并识别数据应用场景中的数据分类、数据名称、数据含义、数据创建、数据使用、数据展示、数据质量、数据安全、数据保留等需求，编写数据需求文档。
- 评审数据需求：组织人员对数据需求文档进行评审，评审关注各项数据需求是否与业务目标、业务需求保持一致，数据需求是否使用已定义的业务术语、数据项、参考数据等数据标准，干系人对数据需求是否达成共识。
- 更新数据管理标准：对于已有数据管理标准中尚未覆盖的数据需求以及经评出后达到需要变更数据标准的，由数据管理人员根据相关流程更新数据标准，保证数据标准与实际数据需求的一致性。
- 集中管理数据需求：各方数据用户的数据需求应集中由数据管理人员进行收集和管理，确保需求的汇总分析和历史回顾。

(2) 数据设计和开发。数据设计和开发是指设计、实施数据解决方案，提供数据应用，持续满足组织的数据需求的过程。数据解决方案包括数据库结构、数据采集、数据整合、数据交换、数据访问及数据产品（报表、用户视图）等。数据设计和开发主要活动和工作要点包括：

- 设计数据解决方案：包括概要设计和详细设计，其设计内容主要是面向具体的应用系统设计逻辑数据模型、物理数据模型、物理数据库、数据产品、数据访问服务、数据整合

服务等，从而形成满足数据需求的解决方案。

- 数据准备：梳理组织的各类数据，明确数据提供方，制定数据提供方案。
- 数据解决方案质量管理：数据解决方案设计应满足数据用户的业务需求，同时也应满足数据的可用性、安全性、准确性、及时性等数据管理需求，因此需要进行数据模型和设计的质量管理，主要内容包括开发数据模型和设计标准，评审概念模型、逻辑模型和物理模型的设计，以及管理和整合数据模型版本变更。
- 实施数据解决方案：通过质量评审的数据解决方案进入实施阶段，主要内容包括开发和测试数据库、建立和维护测试数据、数据迁移和转换、开发和测试数据产品、数据访问服务、数据整合服务、验证数据需求等。

(3) 数据运维。数据运维是指数据平台及相关数据服务建设完成上线投入运营后，对数据采集、数据处理、数据存储等过程的日常运行及其维护过程，保证数据平台及数据服务的正常运行，为数据应用提供持续可用的数据内容。数据运维主要活动和工作要点包括：

- 制定数据运维方案：根据组织数据管理的需要，明确数据运维的组织，制定统一的数据运维方案。
- 数据提供方管理：建立数据提供的监控规则、监控机制和数据合格标准等服务水平协议和检查手段，持续监控数据提供方的服务水平，确保数据平台和数据服务有持续可用、高质量、安全可靠的数据，数据提供方管理包括对组织的内部和外部数据提供方的管理。
- 数据平台的运维：根据数据运维方对数据库、数据平台、数据建模工具、数据分析工具、ETL工具、数据质量工具、元数据工具、主数据管理工具的选型、部署、运行等进行管理，确保各技术工具的选择符合数据架构整体规划，正常运行各项指标满足数据需求。
- 数据需求的变更管理：数据需求实现之后，需要及时跟踪数据应用的运行情况，监控数据应用和数据需求的一致性，同时对用户提出的需求变更进行管理，确保设计和实施的一致性。

(4) 数据退役。数据退役是对历史数据的管理，根据法律法规、业务、技术等方面需求，对历史数据的保留和销毁，执行历史数据的归档、迁移和销毁工作，确保组织对历史数据的管理符合外部监管机构和内部业务用户的需求，而非仅满足信息技术需求。数据退役主要活动和工作要点包括：

- 数据退役需求分析：向组织管理层、各领域业务用户调研内部和外部对数据退役的需求，明确外部监管要求的数据保留和清除要求，明确内部数据应用的保留和清除要求，同时兼顾信息技术对存储容量、访问速度、存储成本等需求。
- 数据退役设计：综合考虑合规、业务和信息技术需求，设计数据退役标准和执行流程，明确不同类型数据保留策略，包括保留期限、保留方式等，建立数据归档、迁移、获取和清除的工作流程和操作规程，确保数据退役符合标准和流程规范。
- 数据退役执行：根据数据退役设计方案执行数据退役操作，完成数据的归档、迁移和清除等工作，满足法规、业务和技术需要，同时根据需要更新数据退役设计。

- 数据恢复检查：数据退役之后需要制定数据恢复检查机制，定期检查退役数据状态，确保数据在需要时可恢复。
- 归档数据查询：根据业务管理或监管需要，对归档数据的查询请求进行管理，并恢复相关数据以供应用。

9. 理论框架与成熟度

国内外常用的数据管理模型包括：数据管理能力成熟度模型（DCMM）、数据治理框架（Data Governance Institute, DGI）、数据管理能力评价模型（Data Management capability Assessment Model, DCAM）以及数据管理模型（DAMA 定义的模型）等。

(1) 数据管理能力成熟度模型。DCMM 将组织的管理成熟度划分为 5 个等级，分别是：初始级、受管理级、稳健级、量化管理级和优化级。

- 初始级：数据需求的管理主要是在项目级体现，没有统一的管理流程，主要是被动式管理。
- 受管理级：组织意识到数据是资产，根据管理策略的要求制定了管理流程，指定了相关人员进行初步管理。
- 稳健级：数据已被当做实现组织绩效目标的重要资产，在组织层面制定了系列的标准化管理流程，促进数据管理的规范化。
- 量化管理级：数据被认为是获取竞争优势的重要资源，数据管理的效率能量化分析和监控。
- 优化级：数据被认为是组织生存和发展的基础，相关管理流程能实时优化，能在行业内进行最佳实践分享。

(2) 数据治理框架。国际数据治理协会发布了 DGI 数据治理框架，是组织在进行数据治理的操作层面的框架体系，为组织做出决策和采取行动的复杂活动提供的方法，此框架从组织结构、治理规则和治理过程这三个维度提出了关于数据治理活动的 10 个关键通用组件，并在这些要素的基础上构建了数据治理框架，如图 4-7 所示。

(3) 数据管理能力评价模型。企业数据管理协会（EDM）是制定和实施数据标准、最佳实践以及全面培训和认证项目的重要倡导者。基于众多实际案例的经验总结来进行编写数据管理能力成熟度评估模型，提供了用于建立和评估组织数据管理计划的关键维度，主要强调团队协作（流程）、标准执行和资金支持，目前最新 DCAM2.2 版有 4 个组件：①基础组件包含数据战略与业务案例、数据管理流程与资金职能域；②执行组件包含业务和数据架构、数据和技术架构、数据质量管理、数据治理职能域；③分析组件包含数据控制环境职能域；④应用组件包含分析管理职能域。

(4) 数据管理模型。国际数据管理协会 DAMA（DAMA International）在 2018 年发行了 DAMA-DMBOK（数据管理知识体系指南）第 2 版，用于指导组织的数据管理职能和数据战略的评估工作，并建议和指导刚起步的组织去实施和提升数据管理。DAMA-DMBOK2 理论框架由 11 个数据管理职能领域和 7 个基本环境要素共同构成“DAMA 数据管理知识体系”，每项数据职能领域都在 7 个基本环境要素约束下开展工作。DAMA-DMBOK2 能力框架图表 4-4 所示。

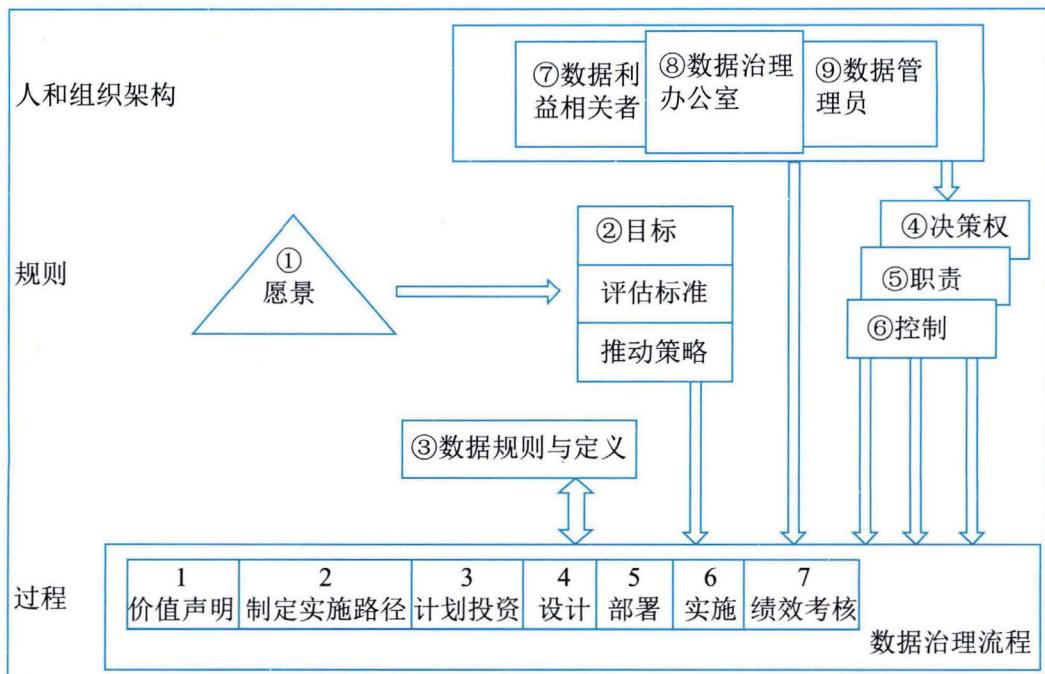


图 4-7 DGI 数据治理框架

表 4-4 DAMA-DMBOK2 职能框架（示意）

数据管理职能	环境要素						
	目标与原则	组织与文化	工具	活动	角色和职责	交付成果	技术
数据治理							
数据架构							
数据建模和设计							
数据存储和操作							
数据安全							
数据集成和互操作							
文档和内容管理							
参考数据和主数据管理							
数据仓库与商务智能							
元数据管理							
数据质量管理							
数据管理职能							

4.2.2 运维管理

IT 运维是组织 IT 服务中关键的一种类型。随着组织 IT 建设的不断深入和完善，信息系统运维已经成为了各行各业各组织管理者和 IT 团队普遍关注的问题。IT 运维是指采用 IT 手段及方法，依据服务对象提出的服务级别要求，对其所使用的 IT 系统运行环境、业务系统等提供的综合活动。

1. 能力模型

国家标准 GB/T 28827.1《信息技术服务 运行维护 第1部分 通用要求》定义了IT运维能力模型，该模型包含治理要求、运行维护服务能力体系和价值实现，如图4-8所示。治理要求是为实现运行维护服务绩效、风险控制和服务合规性的组织目标，提出的关于最高管理层领导作用及承诺的能力体系建设要求。运行维护服务能力体系（MCS）是组织依据运行维护服务方针和目标，策划并制定运行维护服务能力方案，确保组织交付的运行维护服务内容符合相关规定，并满足质量要求，对运行维护服务交付过程、结果以及运行维护服务能力体系进行监督、测量、分析和评审，以实现运行维护服务能力的持续提升。价值实现是组织结合业务对信息系统的网络化、数字化和智能化要求，识别内部和外部用户对服务的需求或期望，定义多样化的服务场景，并通过服务能力、要素、活动的组合完成服务的提供，直接或间接地为服务需求方和利益相关者实现服务价值。

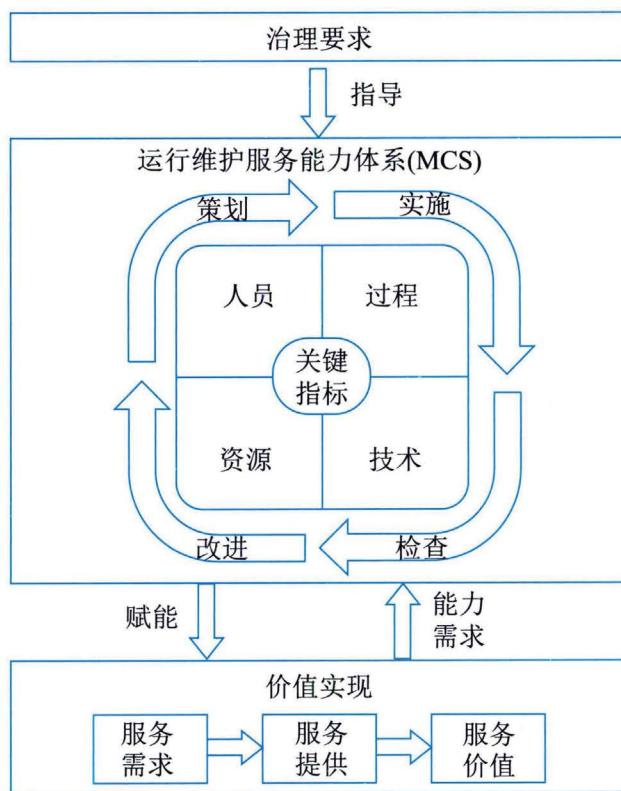


图 4-8 IT 运维能力模型图

1) 能力建设

组织需要考虑环境的内外部因素，在治理要求的指导下，根据服务场景，识别服务能力需求，围绕人员、过程、技术、资源能力四要素，策划、实施、检查和改进运行维护能力体系，向各种服务场景赋能，通过服务提供实现服务价值；并针对能力建设、人员、过程、技术、资源建立关键指标；还需要定期评价运行维护服务能力成熟度，衡量能力水平差距，以持续提升运行维护服务能力。

在治理层面，最高管理层应依据组织治理目标，提出运行维护服务能力管理治理要求，以

确保实现运行维护服务绩效、风险控制和服务合规性。

在能力管理方面，运维能力管理是面向运维全生命期的总体能力管控机制，分为策划、实施、检查和持续改进四个阶段，各阶段交替循环，实现运维能力持续性地螺旋式上升的管理目标。这需要组织：①周期性的（如按年度）面向外部的用户需求以及内部的合规要求和成本约束等，对运维能力进行总体策划，包括服务目录的建立和维护，组织架构和管理制度的确立，并形成年度运维能力管理计划，确保运维目标的可实施性；②细化能力管理计划为具体的实施计划（通常按部门进行任务分解），并落地执行；③定期（如按半年度或季度）跟踪和检查实施计划的执行情况，并进行适时评估、优化和调整；④对IT运维能力管理的达成情况进行总结分析，并持续改进。实现按PDCA的方式实施能力管理，进而提升整体服务能力。

在能力管理过程中，组织需要首先明确能力管理团队的组成，并明确这些团队成员的职责范围与分工，根据组织IT运维的内外部环境、技术发展现状、运维各利益干系人的诉求、能力体系覆盖范围、管理者的作用、资金投入、人才保障、基础设施设备的情况、安全以及质量体系的基础等因素，实施能力策划活动，并明晰周期性的能力管理计划、能力指标等，在策划过程中需要明确策划的输入、输出、审批以及变更控制等；同时抓好能力计划实施的计划管理、协调管理、记录管理以及成果管理等，做到实施过程记录的“线条”证据化；并设立专门的检查组织，明确检查方法，并按照确定的计划实施检查；还需要建立适合于组织的改进机制，以及确保改进活动的有效实施。通常来说，能力管理不是分管运维的高级管理者或者主管运维的负责人单个岗位的工作，需要人力资源、技术研发、质量监督等等多方面的人员共同参与。

在价值实现方面，组织需要在不同的服务场景中识别服务需求，通过服务提供，满足用户需求，实现服务价值：

- 服务需求：识别服务需求并遵循能力管理的要求对服务场景进行完整的策划。
- 服务提供：配置符合能力要素要求且和服务场景相适宜的人员、过程、技术和资源，并遵循能力管理的要求实施服务提供。
- 服务价值：将运行维护服务能力体系输出的服务能力应用到服务场景中，通过服务成果、成本控制、风险控制实现服务价值。

2) 人员能力

在任何组织当中，人力资源都是组织的核心竞争力之一。因此绝大部分组织对人员相关的建设和管理都非常重视，无论是人员的容量、技能、工作绩效等方方面面，都是组织关注的重点。组织人员能力建设聚焦在从知识、技能和经验维度选择合适的人，从人员管理和岗位职责维度明确做适合的事，目的是指导IT运维团队根据岗位职责和管理要求“选人做事”。

结合IT运维工作的特点，运维人员一般分为管理类、技术类和操作类三种人员岗位，管理类主要负责运维的组织管理，技术类主要负责运维技术建设以及运维活动中的技术决策等，操作类主要负责运维活动的执行等。

为了保证人员能力满足运维服务的要求，组织依据运维能力策划要求，进行人员能力策划、岗位结构、人员储备、人员培训、绩效管理和能力评价等管理活动。

对运维的人员能力建设，通常还需要考虑：

- 面向IT运维所有干系人需求，建立人员需求规划；
- 基于人员需求计划，制定人员招聘、培训、储备和考核机制并实施；
- 定义IT运维人员岗位，根据工作内容不同，划分管理岗、技术岗、操作岗，并对每个岗位梳理工作职责，同时定义岗位的任职要求，包括知识、技能及经验要求等方面。

组织应按人员能力计划，进行运行维护人员能力评价，至少应包括：

- 建立运行维护服务对应岗位的等级评价标准；
- 建立运行维护服务团队和人员能力评价机制；
- 实施团队和人员能力评价；
- 依据评价结果对人员能力进行持续改进，需要时调整人员能力计划。

3) 资源能力

资源主要由人员、过程和技术要素中被固化下来的能力转化而成，人员、过程和技术要素在知识、服务管理、工具支撑等方面的能力被固化下来，同时又对人员、过程和技术要素提供有力的支撑和保障，进而形成资源能力中的知识库、服务台、备件库以及运行维护工具，资源能力确保IT运维能“保障做事”。

IT运维资源是为了保证IT运维的正常交付所依存和产生的有形及无形资产。该表述最后的落脚点是资产，这就区别于广义的资源概念，广义的资源是指组织拥有的物力、财力、人力等各种物质要素的总称。

组织在建设资源能力过程中，要充分重视自主知识、技术和业务流程的固化工作，从而充分发挥经验的沉淀，尤其要关注一线人员的技术资源化，从而保证质量的同时提高效率和效能，建议组织可以定期收集一下一线人员针对资源的意见和建议，从而及时补充必要的资源，保持组织的运维能力的优化提升。

组织应根据运维能力策划要求和特定服务场景的需求，按需建立和管理运行维护工具、服务台、备件库、最终软件库、服务数据和服务知识等，以满足不同服务场景的服务需求。实现与人员、过程和技术结合，保证资源能力满足价值实现过程中服务提供的需求。

4) 技术能力

组织需要通过自有核心技术的研发和非自有核心技术的学习，持续提升IT运维过程中发现问题和解决问题的能力，在提升IT运维效率方面是重点考虑的要素，技术要素确保IT运维能“高效做事”。

在实施IT运维过程中，可能面临各种问题、风险以及新技术和前沿技术应用所提出的新要求，组织需要根据服务对象要求或技术发展趋势，具备发现和解决问题、风险控制、技术储备以及研发、应用新技术和前沿技术的能力。

“早发现，早解决”一直是IT运维的一个重要原则，技术作为提高效率的基本因素，其在该领域中起着至关重要的作用。需要说明一点，这里的技术不单纯指IT技术，而是涵盖IT技术在内的所有IT运维技术，包括工作手册、思维方法等。从分类上来说，运维技术聚焦在发现问题的技术和解决问题的技术两大领域。

组织应根据运维能力策划要求，实施技术管理、技术研发和技术成果应用等活动，保证技

术能力满足不同服务场景下的服务要求，包括运维服务能力长期发展的需求、治理、预期效益等，实现其服务价值。

5) 过程

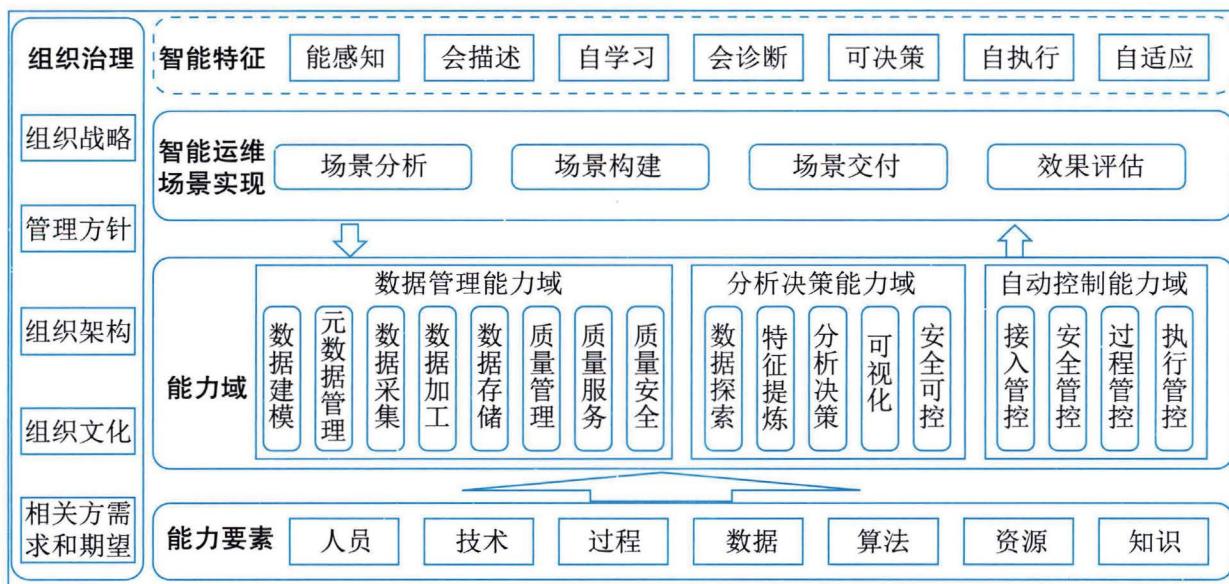
组织通过过程的制定，把人员、技术和资源要素以过程为主线串接在一起，用于指导IT运维人员按约定的方式和方法，确保IT运维能“正确做事”。

过程又称流程，是为达到特定的价值目标而由不同的人分别共同完成的一系列活动。活动之间不仅有严格的先后顺序限定，而且活动的内容、方式、责任等也都必须有明确的安排和界定，以使不同活动在不同岗位角色之间进行转手交接成为可能。活动与活动之间在时间和空间上的转移可以有较大的跨度。而狭义的业务流程，则认为它仅仅是与客户价值的满足相联系的一系列活动。

组织需要结合服务场景与运维能力策划要求，设计过程框架，明确各过程之间的关系和接口，制定服务级别、服务报告、事件、问题、变更、发布、配置、可用性和连续性、系统容量、信息安全等管理过程的目标、活动和考核指标，支撑服务过程的规范化管理和服务价值实现。

2. 智能运维

中国电子工业标准化技术协会发布的团体标准T/CESA 1172《信息技术服务 智能运维 通用要求》，给出了智能运维能力框架，包括组织治理、智能特征、智能运维场景实现、能力域和能力要素，其中能力要素是构建智能运维能力的基础。组织需在组织治理的指导下，对智能运维场景实现提出能力建设要求，开展智能运维能力规划和建设。组织通过场景分析、场景构建、场景交付和效果评估四个过程，基于数据管理能力域提供的高质量数据，结合分析决策能力域做出合理判断或结论，并根据需要驱动自动控制能力域执行运维操作，使运维场景具备智能特征，提升智能运维水平，实现质量可靠、安全可控、效率提升、成本降低。智能运维能力框架如图4-9所示。



(1) 能力要素。智能运维的能力要素主要包括:

- 人员: 运维团队需要熟悉IT运维领域的业务活动与流程, 掌握自动化、大数据、人工智能、云计算、算法等技术, 具备一定的智能运维研发能力。
- 技术: 技术通常包括统一的标准和规范、开放的基础公共资源与服务、数据与流程及服务的互联互通等。
- 过程: 智能运维定义的过程需要具备清晰界定人机界面, 能够充分发挥智能化优势, 实现过程优化, 并考虑权限控制、风险规避。
- 数据: 运维组织需要加强数据治理, 保证数据质量, 规范数据接口。运维应用需要围绕数据进行采集、加工、消费, 提升运维智能化水平。
- 算法: 可以聚焦在异常检测、根因分析、故障预测、知识图谱、健康诊断、决策分析等方面, 具备有穷性、确切性、有效性等特点。
- 资源: 组织在数据管理能力域数据服务中, 对于资源管理, 至少应根据不同场景要求, 配置开放共享服务管理所需要的算力、带宽、存储等。
- 知识: 知识通常包括运维技术方案及方法与步骤、运维的经验沉淀、运维对象的多维度描述、运维数据的智能挖掘结果等。

(2) 能力平台。智能运维能力平台通常具备数据管理、分析决策、自动控制等能力。其中, 数据管理能力用于采集、处理、存储、展示各种运维数据。分析决策能力以感知到的数据作为输入, 做出实时的运维决策, 驱动自动化工具实施操作。自动控制根据运维决策, 实施具体的运维操作。

(3) 能力应用。以运维场景为中心, 通过场景分析、能力构建、服务交付、迭代调优四个关键环节, 可以使运维场景具备智能特征。根据复杂程度, 运维场景分为单一场景、复合场景和全局场景。

- 场景分析: 是指从业务或IT本身接收对新服务或改进服务的需求, 场景需求分析从业务需求、用户需求以及系统需求, 不同层次阶段进行不同方式、内容以及侧重点的需求调研。
- 能力构建: 是指基于运维场景分析的结果和目标要求, 应用赋能平台中适合运维场景数据特点的加工处理能力、系统性设计数据的处理流程, 构建符合特定运维场景需求的智能运维解决方案。
- 服务交付: 是指制订详细的交付计划, 准备必要的资源, 评估可能存在的风险并明确规避方案, 完善交付实施过程, 通过服务交付检查确保运维场景的智能特征符合策划要求。
- 迭代调优: 是指通过持续的迭代对智能运维场景的优化, 确保投入符合智能运维具体场景的规划目标渐进式达成。

(4) 智能运维需具备若干智能特征, 智能特征包括:

- 能感知: 指具备灵敏、准确地识别人、活动和对象的状态的特点。
- 会描述: 指具备直观友好地展现和表达运维场景中各类信息的特点。

- 自学习：指具备积累数据、完善模型、总结规律等主动获取知识的特点。
- 会诊断：指具备对人、活动和对象进行分析、定位、判断的特点。
- 可决策：指具备综合分析，给出后续处置依据或解决方案的特点。
- 自执行：指具备对已知运维场景做出自动化处置的特点。
- 自适应：指具备自动适应环境变化，动态优化处理的特点。

4.2.3 信息安全管理

当前社会已经进入了数字时代，其突出的特点表现为信息的价值在很多方面超过其信息处理设施包括信息载体本身的价值，例如一台计算机上存储和处理的信息价值往往超过计算机本身的价值。另外，现代社会的各类组织，包括政府、企业，对信息以及信息处理设施的依赖也越来越大，一旦信息丢失或泄密、信息处理设施中断，很多组织的业务也就无法运营了。新时代对于信息的安全提出了更高的要求，对信息安全的内涵也不断进行延伸和拓展。

1. CIA 三要素

CIA三要素是保密性（Confidentiality）、完整性（Integrity）和可用性（Availability）三个词的缩写。CIA是系统安全设计的目标。保密性、完整性和可用性是信息安全最为关注的三个属性，因此这三个特性也经常被称为信息安全三元组，这也是信息安全通常所强调的目标。信息安全已经成为一门涉及计算机科学、网络技术、通信技术、密码技术、信息安全技术、应用数学、数论和信息论等多种学科的综合性学科。从广义上来说，凡是涉及网络上信息的保密性、完整性、可用性、真实性和可核查性的相关技术和理论都属于信息安全的研究领域。

CIA有其局限性。CIA关注的重心在信息，虽然这是大多数信息安全的核心要素，但对于信息系统安全而言，仅考虑CIA是不够的。信息安全的复杂性决定了还存在其他的重要因素。CIA给出了一个信息系统整体安全模型框架，能帮助信息化工作人员在制定安全策略时形成思路，但这并不是所有需要考虑的策略。CIA可以作为规划、实施量化安全策略的基本原则，但是我们也应该认识到它的局限性。

2. 信息安全管理体

信息系统安全管理是对一个组织机构中信息系统的生存周期全过程实施符合安全等级责任要求的管理，主要包括：

- 落实安全管理机构及安全管理人员，明确角色与职责，制定安全规划；
- 开发安全策略；
- 实施风险管理；
- 制订业务持续性计划和灾难恢复计划；
- 选择与实施安全措施；
- 保证配置、变更的正确与安全；
- 进行安全审计；
- 保证维护支持；

- 进行监控、检查，处理安全事件；
- 安全意识与安全教育；
- 人员安全管理等。

在组织机构中应建立安全管理机构，不同安全等级的安全管理机构逐步建立自己的信息系统安全组织机构管理体系，参考步骤包括：①配备安全管理人员。管理层中应有一人分管信息系统安全工作，并为信息系统的安全管理配备专职或兼职的安全管理人员。②建立安全职能部门。建立管理信息系统安全工作的职能部门，或者明确设置一个职能部门监管信息安全工作，作为该部门的关键职责之一。③成立安全领导小组。在管理层成立信息系统安全管理委员会或信息系统安全领导小组，对覆盖全国或跨地区的组织机构，应在总部和下级单位建立各级信息系统安全领导小组，在基层至少要有一位专职的安全管理人员负责信息系统安全工作。④主要负责人出任领导。由组织机构的主要负责人出任信息系统安全领导小组负责人。⑤建立信息安全保密管理部门。建立信息系统安全保密监督管理的职能部门，或对原有保密部门明确信息安全保密管理责任，加强对信息系统安全管理重要过程和管理人员的保密监督管理。

3. 网络安全等级保护

国家市场监督管理总局、国家标准化管理委员会宣布网络安全等级保护制度2.0相关的若干国家标准正式发布，并于2019年12月1日开始实施。“等保1.0”体系以信息系统为对象，确立了五级安全保护等级，并从信息系统安全等级保护的定级方法、基本要求、实施过程、测评工作等方面入手，形成了一套相对完整的、有明确标准的，且涵盖了制度与技术要求的等级保护规范体系。然而，随着网络安全形势日益严峻，“等保1.0”体系逐渐难以持续应对不容乐观的网络安全新时代，于是“等保2.0”体系应运而生。

等保2.0将“信息系统安全”的概念扩展到了“网络安全”，其中所谓“网络”是指由计算机或者其他信息终端及相关设备组成的按照一定的规则和程序对信息进行收集、存储、传输、交换、处理的系统。

1) 安全保护等级划分

GB/T 22240《信息安全技术 网络安全等级保护定级指南》定义了等级保护对象，为网络安全等级保护工作直接作用的对象，主要包括信息系统、通信网络设施和数据资源等。根据等级保护对象在国家安全、经济建设、社会生活中的重要程度，以及一旦遭到破坏、丧失功能或数据被篡改、泄露、丢失、损毁后，对国家安全、社会秩序、公共利益以及公民、法人和其他组织的合法权益的侵害程度等因素，等级保护对象的安全保护等级分为以下五级：第一级，等级保护对象受到破坏后，会对相关公民、法人和其他组织的合法权益造成损害，但不危害国家安全、社会秩序和公共利益；第二级，等级保护对象受到破坏后，会对相关公民、法人和其他组织的合法权益产生严重损害或特别严重损害，或者对社会秩序和公共利益造成危害，但不危害国家安全；第三级，等级保护对象受到破坏后，会对社会秩序和公共利益造成严重危害，或者对国家安全造成危害；第四级，等级保护对象受到破坏后，会对社会秩序和公共利益造成特别严重危害，或者对国家安全造成严重危害；第五级，等级保护对象受到破坏后，会对国家安全造成特别严重危害。

2) 安全保护能力等级划分

GB/T 22239《信息安全技术 网络安全等级保护基本要求》规定了不同级别的等级保护对象应具备的基本安全保护能力。

第一级安全保护能力：应能够防护免受来自个人的、拥有很少资源的威胁源发起的恶意攻击、一般的自然灾害，以及其他相当危害程度的威胁所造成的关键资源损害，在自身遭到损害后，能够恢复部分功能。

第二级安全保护能力：应能够防护免受来自外部小型组织的、拥有少量资源的威胁源发起的恶意攻击、一般的自然灾害，以及其他相当危害程度的威胁所造成的重要资源损害，能够发现重要的安全漏洞和处置安全事件，在自身遭到损害后，能够在一段时间内恢复部分功能。

第三级安全保护能力：应能够在统一安全策略下防护免受来自外部有组织的团体、拥有较为丰富资源的威胁源发起的恶意攻击、较为严重的自然灾害，以及其他相当程度的威胁所造成的主要资源损害，能够及时发现、监测攻击行为和处置安全事件，在自身遭到损害后，能够较快恢复绝大部分功能。

第四级安全保护能力：应能够在统一安全策略下防护免受来自国家级别的、敌对组织的、拥有丰富资源的威胁源发起的恶意攻击、严重的自然灾害，以及其他相当危害程度的威胁所造成的资源损害，能够及时发现、监测发现攻击行为和安全事件，在自身遭到损害后，能够迅速恢复所有功能。

第五级安全保护能力：略。

4.3 本章练习

1. 选择题

(1) _____ 不属于信息系统架构模式。

- A. 集中式架构
- B. 分布式架构
- C. 企业信息化架构
- D. 面向服务的架构

参考答案: C

(2) _____ 不属于 IT 运维能力的关键指标。

- A. 人员
- B. 技术
- C. 过程
- D. 问题

参考答案: D

(3) _____ 不属于信息系统咨询设计的范畴。

- A. 战略咨询
- B. 技术咨询
- C. 管理咨询
- D. 财务咨询

参考答案: D

(4) 智能运维场景实现的关键审核要点应围绕_____。

- A. 质量可靠、安全可控、效率提升、成本降低的四个运维目标
- B. 场景分析、场景构建、场景交付、效果评估四个关键过程

- C. 数据管理、分析决策、自动控制三个能力域
- D. 能感知、会描述、自学习、会诊断、可决策、自执行、自适应七个特征

参考答案: B

(5) _____就是确保所传输的数据只被其预定的接收者读取。

- A. 保密性
- B. 可靠性
- C. 可用性
- D. 完整性

参考答案: A

(6) _____不属于服务质量构成的因素。

- A. 服务生产质量
- B. 服务要素质量
- C. 服务生产质量
- D. 服务感知质量

参考答案: D

2. 思考题

(1) 请给出信息系统规划的战略三角关系，并阐述信息系统战略应如何与业务和组织战略保持一致？

参考答案: 略

(2) 请详细阐述运维能力模型的主要内容。

参考答案: 略

(3) 请阐述一下IT服务管理的主要活动，及这些活动的主要管理要点。

参考答案: 略

第5章 信息系统工程

信息系统工程是用系统工程的原理、方法来指导信息系统建设与管理的一门工程技术学科，它是信息科学、管理科学、系统科学、计算机科学与通信技术相结合的综合性、交叉性、具有独特风格的应用学科。当前信息系统工程的主要任务是研究信息处理过程内在的规律，以及基于计算机、互联网和云计算等信息技术手段的形式化表达和处理规律。其基本概念、原理和方法对实际分析、设计、开发、运行和服务一个信息系统，从理论、手段、方法、技术等多方面提供了一套完整、科学、实用的研究与工程体系，具有十分重要的应用价值，对信息系统建设有着重要的理论指导意义。

5.1 软件工程

软件工程是指应用计算机科学、数学及管理科学等原理，以工程化的原则和方法来解决软件问题的工程，其目的是提高软件生产率、提高软件质量、降低软件成本。电气与电子工程师协会（Institute of Electrical and Electronics Engineers, IEEE）对软件工程的定义是：将系统的、规范的、可度量的工程化方法应用于软件开发、运行和维护的全过程及上述方法的研究。

软件工程由方法、工具和过程三个部分组成：①软件工程方法是完成软件工程项目的技术手段，它支持整个软件生命周期；②软件工程使用的工具是人们在开发软件的活动中智力和体力的扩展与延伸，它自动或半自动地支持软件的开发和管理，支持各种软件文档的生成；③软件工程中的过程贯穿于软件开发的各个环节，管理人员在软件工程过程中，要对软件开发的质量、进度和成本进行评估、管理和控制，包括人员组织、计划跟踪与控制、成本估算、质量保证和配置管理等。

5.1.1 架构设计

软件架构为软件系统提供了一个结构、行为和属性的高级抽象，由构件的描述、构件的相互作用（连接件）、指导构件集成的模式以及这些模式的约束组成。软件架构不仅指定了系统的组织结构和拓扑结构，并且显示了系统需求和构件之间的对应关系，提供了一些设计决策的基本原理。

软件架构虽脱胎于软件工程，但其形成同时借鉴了计算机架构和网络架构中很多宝贵的思想和方法。近年来，软件架构已完全独立于软件工程，成为计算机科学的一个最新的研究方向和独立学科分支。软件架构研究的主要内容涉及软件架构描述、软件架构风格、软件架构评估和软件架构的形式化方法等。解决好软件的复用、质量和维护问题，是研究软件架构的根本目的。

1. 软件架构风格

软件架构设计的一个核心问题是能否达到架构级的软件复用，也就是说，能否在不同的系

统中使用同一个软件架构。软件架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（Idiomatic Paradigm）。架构风格定义了一个系统“家族”，即一个架构定义、一个词汇表和一组约束。词汇表中包含一些构件和连接件类型，而约束指出系统是如何将这些构件和连接件组合起来的。架构风格反映了领域中众多系统所共有的结构和语义特性，并指导如何将各个构件有效地组织成一个完整的系统。

Garlan 和 Shaw 对通用软件架构风格进行了分类，他们将软件架构分为：①数据流风格。数据流风格包括批处理序列和管道/过滤器两种风格。②调用/返回风格。调用/返回风格包括主程序/子程序、数据抽象和面向对象，以及层次结构。③独立构件风格。独立构件风格包括进程通信和事件驱动的系统。④虚拟机风格。虚拟机风格包括解释器和基于规则的系统。⑤仓库风格。仓库风格包括数据库系统、黑板系统和超文本系统。

2. 软件架构评估

软件架构设计是软件开发过程中的关键一步。没有一个合适的架构而要有一个成功的软件设计几乎是不可想象的，尤其是针对庞大且复杂的系统来说。不同类型的系统需要不同的架构，甚至一个系统的不同子系统也需要不同的架构。架构的选择往往成为系统设计成败的关键。但是，怎样才能知道为系统所选用的架构是否恰当呢，如何确保按照所选用的架构能顺利地开发出成功的软件产品呢，要回答这些问题并不容易，因为它受到很多因素的影响，需要专门的方法来对其进行评估。软件架构评估可以只针对一个架构，也可以针对一组架构。在架构评估过程中，评估人员所关注的是系统的质量属性。

在分析具体架构评估方法之前，我们先来了解两个概念，分别是敏感点（Sensitivity Point）和权衡点（Trade-off Point）。敏感点是一个或多个构件（或之间的关系）的特性，权衡点是影响多个质量属性的特性，是多个质量属性的敏感点。例如，改变加密级别可能会对安全性和性能产生非常重要的影响。提高加密级别可以提高安全性，但可能要耗费更多的处理时间，影响系统性能。如果某个机密消息的处理有严格的时间延迟要求，则加密级别可能就会成为一个权衡点。

从目前已有的软件架构评估技术来看，可以归纳为三类主要的评估方式，分别是基于调查问卷（或检查表）的方式、基于场景的方式和基于度量的方式。这三种评估方式中，基于场景的评估方式最为常用。

基于场景的方式主要包括：架构权衡分析法（Architecture Trade-off Analysis Method, ATAM）、软件架构分析法（Software Architecture Analysis Method, SAAM）和成本效益分析法（Cost Benefit Analysis Method, CBAM）中。在架构评估中，一般采用刺激（Stimulus）、环境（Environment）和响应（Response）三方面来对场景进行描述。刺激是场景中解释或描述项目干系人怎样引发与系统的交互部分，环境描述的是刺激发生时的情况，响应是指系统是如何通过架构对刺激做出反应的。

基于场景的方式分析软件架构对场景的支持程度，从而判断该架构对这一场景所代表的质量需求的满足程度。例如，用一系列对软件的修改来反映易修改性方面的需求，用一系列攻击性操作来代表安全性方面的需求等。这一评估方式考虑到了所有与系统相关人员对质量的要求，

涉及的基本活动包括确定应用领域的功能和软件架构之间的映射，设计用于体现待评估质量属性的场景，以及分析软件架构对场景的支持程度。

不同的系统对同一质量属性的理解可能不同，例如，对操作系统来说，可移植性被理解为系统可在不同的硬件平台上运行，而对于普通的应用系统而言，可移植性往往是指该系统可在不同的操作系统上运行。由于存在这种不一致性，对一个领域适合的场景设计在另一个领域内未必合适，因此，基于场景的评估方式是特定于领域的。这一评估方式的实施者一方面需要有丰富的领域知识，以对某一质量需求设计出合理的场景；另一方面，必须对待评估的软件架构有一定的了解，以准确判断它是否支持场景描述的一系列活动。

5.1.2 需求分析

软件需求是指用户对新系统在功能、行为、性能、设计约束等方面期望。根据 IEEE 的软件工程标准词汇表，软件需求是指用户解决问题或达到目标所需的条件或能力，是系统或系统部件要满足合同、标准、规范或其他正式规定文档所需具有的条件或能力，以及反映这些条件或能力的文档说明。

1. 需求的层次

简单地说，软件需求就是系统必须完成的事以及必须具备的品质。需求是多层次的，包括业务需求、用户需求和系统需求，这三个不同层次从目标到具体，从整体到局部，从概念到细节。

质量功能部署（Quality Function Deployment, QFD）是一种将用户要求转化成软件需求的技术，其目的是最大限度地提升软件工程过程中用户的满意度。为了达到这个目标，QFD 将软件需求分为三类，分别是常规需求、期望需求和意外需求。

2. 需求过程

需求过程主要包括需求获取、需求分析、需求规格说明书编制、需求验证与确认等。

1) 需求获取

需求获取是一个确定和理解不同的项目干系人的需求和约束的过程。需求获取是一件看上去很简单，做起来却很难的事情。需求获取是否科学、准备充分，对获取出来的结果影响很大，这是因为大部分用户无法完整地描述需求，而且也不可能看到系统的全貌。因此，需求获取只有与用户的有效合作才能成功。常见的需求获取方法包括用户访谈、问卷调查、采样、情节串联板、联合需求计划等。

2) 需求分析

在需求获取阶段获得的需求是杂乱的，是用户对新系统的期望和要求，这些要求有重复的地方，也有矛盾的地方，这样的要求是不能作为软件设计基础的。一个好的需求应该具有无二义性、完整性、一致性、可测试性、确定性、可跟踪性、正确性、必要性等特性，因此，需要分析人员把杂乱无章的用户要求和期望转化为用户需求，这就是需求分析的工作。

需求分析对已经获取到的需求进行提炼、分析和审查，以确保所有的项目干系人都明白其含义并找出其中的错误、遗漏或其他不足的地方。需求分析的关键在于对问题域的研究与理解。

为了便于理解问题域，现代软件工程方法所推荐的做法是对问题域进行抽象，将其分解为若干个基本元素，然后对元素之间的关系进行建模。

使用结构化分析（Structured Analysis, SA）方法进行需求分析，其建立的模型的核心是数据字典。围绕这个核心，有三个层次的模型，分别是数据模型、功能模型和行为模型（也称为状态模型）。在实际工作中，一般使用实体关系图（E-R 图）表示数据模型，用数据流图（Data Flow Diagram, DFD）表示功能模型，用状态转换图（State Transform Diagram, STD）表示行为模型。E-R 图主要描述实体、属性，以及实体之间的关系；DFD 从数据传递和加工的角度，利用图形符号通过逐层细分描述系统内各个部件的功能和数据在它们之间传递的情况，来说明系统所完成的功能；STD 通过描述系统的状态和引起系统状态转换的事件，来表示系统的行为，指出作为特定事件的结果将执行哪些动作（例如，处理数据等）。

面向对象的分析（Object-Oriented Analysis, OOA）的基本任务是运用面向对象的（Object-Oriented, OO）方法，对问题域进行分析和理解，正确认识其中的事物及它们之间的关系，找出描述问题域和系统功能所需的类和对象，定义它们的属性和职责，以及它们之间所形成的各种联系。最终产生一个符合用户需求，并能直接反映问题域和系统功能的 OOA 模型及其详细说明。OOA 模型包括用例模型和分析模型，用例是一种描述系统需求的方法，使用用例的方法来描述系统需求的过程就是用例建模；分析模型描述系统的基本逻辑结构，展示对象和类如何组成系统（静态模型），以及它们如何保持通信，实现系统行为（动态模型）。

3) 需求规格说明书编制

软件需求规格说明书（Software Requirement Specification, SRS）是需求开发活动的产物，编制该文档的目的是使项目干系人与开发团队对系统的初始规定有一个共同的理解，使之成为整个开发工作的基础。SRS 是软件开发过程中最重要的文档之一，对于任何规模和性质的软件项目都不应该缺少。

在国家标准 GB/T 8567《计算机软件文档编制规范》中，提供了一个 SRS 的文档模板和编写指南，其中规定 SRS 应该包括范围、引用文件、需求、合格性规定、需求可追踪性、尚未解决的问题、注解和附录。

另外，国家标准 GB/T 9385《计算机软件需求说明编制指南》也给出了一个详细的 SRS 写作大纲，由于该标准年代久远，一些情况已经与现实不符，可以考虑作为 SRS 写作的参考之用。

4) 需求验证与确认

资深软件工程师都知道，当以 SRS 为基础进行后续开发工作，如果在开发后期或在交付系统之后才发现需求存在问题，这时修补需求错误就需要做大量的工作。相对而言，在系统分析阶段，检测 SRS 中的错误所采取的任何措施都将节省相当多的时间和资金。因此，有必要对于 SRS 的正确性进行验证，以确保需求符合良好特征。需求验证与确认活动内容包括：

- SRS 正确地描述了预期的、满足项目干系人需求的系统行为和特征；
- SRS 中的软件需求是从系统需求、业务规格和其他来源中正确推导而来的；
- 需求是完整的和高质量的；

- 需求的表示在所有地方都是一致的；
- 需求为继续进行系统设计、实现和测试提供了足够的基础。

在实际工作中，一般通过需求评审和需求测试工作来对需求进行验证。需求评审就是对 SRS 进行技术评审，SRS 的评审是一项精益求精的技术，它可以发现那些二义性的或不确定性的需求，为项目干系人提供在需求问题上达成共识的方法。需求的遗漏和错误具有很强的隐蔽性，仅仅通过阅读 SRS，通常很难想象在特定环境下系统的行为。只有在业务需求基本明确，用户需求部分确定时，同步进行需求测试，才可能及早发现问题，从而在需求开发阶段以较低的代价解决这些问题。

3. UML

统一建模语言（Unified Modeling Language, UML）是一种定义良好、易于表达、功能强大且普遍适用的建模语言，它融入了软件工程领域的新思想、新方法和新技术，它的作用域不限于支持 OOA 和 OOD（Object-Oriented Design，面向对象设计），还支持从需求分析开始的软件开发的全过程。从总体上来看，UML 的结构包括构造块、规则和公共机制三个部分，如表 5-1 所示。

表 5-1 UML 的结构

部分	说明
构造块	UML 有三种基本的构造块，分别是事物（Thing）、关系（Relationship）和图（Diagram）。事物是 UML 的重要组成部分，关系把事物紧密联系在一起，图是多个相互关联的事物的集合
规则	规则是构造块如何放在一起的规定，包括为构造块命名；给一个名字以特定含义的语境，即范围；怎样使用或看见名字，即可见性；事物如何正确、一致地相互联系，即完整性；运行或模拟动态模型的含义是什么，即执行
公共机制	公共机制是指达到特定目标的公共 UML 方法，主要包括规格说明（详细说明）、修饰、公共分类（通用划分）和扩展机制四种

1) UML 中的事物

UML 中的事物也称为建模元素，包括结构事物（Structural Things）、行为事物（Behavioral Things，也称动作事物）、分组事物（Grouping Things）和注释事物（Annotational Things，也称注解事物）。这些事物是 UML 模型中最基本的 OO 构造块，如表 5-2 所示。

表 5-2 UML 中的事物

建模元素	说明
结构事物	结构事物在模型中属于最静态的部分，代表概念上或物理上的元素。UML 有七种结构事物，分别是类、接口、协作、用例、活动类、构件和节点
行为事物	行为事物是 UML 模型中的动态部分，代表时间和空间上的动作。UML 有两种主要的行为事物。第一种是交互（内部活动），交互是由一组对象之间在特定上下文中，为达到特定目的而进行的一系列消息交换而组成动作。交互中组成动作的对象的每个操作都要详细列出，包括消息、动作次序（消息产生的动作）、连接（对象之间的连接）；第二种是状态机，状态机由一系列对象的状态组成

(续表)

建模元素	说明
分组事物	分组事物是UML模型中组织的部分，可以把它们看成是个盒子，模型可以在其中进行分解。UML只有一种分组事物，称为包。包是一种将有组织的元素分组的机制。与构件不同的是，包纯粹是一种概念上的事物，只存在于开发阶段，而构件可以存在于系统运行阶段
注释事物	注释事物是UML模型的解释部分

2) UML中的关系

UML用关系把事物结合在一起，主要有四种关系，分别为：

- 依赖（Dependency）：依赖是两个事物之间的语义关系，其中一个事物发生变化会影响另一个事物的语义。
- 关联（Association）：关联描述一组对象之间连接的结构关系。
- 泛化（Generalization）：泛化是一般化和特殊化的关系，描述特殊元素的对象可替换一般元素的对象。
- 实现（Realization）：实现是类之间的语义关系，其中的一个类指定了由另一个类保证执行的契约。

3) UML 2.0 中的图

UML 2.0 包括 14 种图，如表 5-3 所示。

表 5-3 UML 2.0 中的图

种类	说明
类图（Class Diagram）	类图描述一组类、接口、协作和它们之间的关系。在 OO 系统的建模中，最常见的图就是类图。类图给出了系统的静态设计视图，活动类的类图给出了系统的静态进程视图
对象图（Object Diagram）	对象图描述一组对象及它们之间的关系。对象图描述了在类图中所建立的事物实例的静态快照。和类图一样，这些图给出系统的静态设计视图或静态进程视图，但它们是从真实案例或原型案例的角度建立的
构件图（Component Diagram）	构件图描述一个封装的类和它的接口、端口，以及由内嵌的构件和连接件构成的内部结构。构件图用于表示系统的静态设计实现视图。对于由小的部件构建大的系统来说，构件图是很重要的。构件图是类图的变体
组合结构图（Composite Structure Diagram）	组合结构图描述结构化类（例如，构件或类）的内部结构，包括结构化类与系统其余部分的交互点。组合结构图用于画出结构化类的内部内容
用例图（Use Case Diagram）	用例图描述一组用例、参与者及它们之间的关系。用例图给出系统的静态用例视图。这些图在对系统的行为进行组织和建模时是非常重要的

(续表)

种类	说明
顺序图 (Sequence Diagram, 也称序列图)	顺序图是一种交互图 (Interaction Diagram), 交互图展现了一种交互, 它由一组对象或参与者以及它们之间可能发送的消息构成。交互图专注于系统的动态视图。顺序图是强调消息的时序的交互图
通信图 (Communication Diagram)	通信图也是一种交互图, 它强调收发消息的对象或参与者的结构组织。顺序图和通信图表达了类似的基本概念, 但它们所强调的概念不同, 顺序图强调的是时序, 通信图强调的是对象之间的组织结构 (关系)。在 UML 1.X 版本中, 通信图称为协作图 (Collaboration Diagram)
定时图 (Timing Diagram, 也称计时图)	定时图也是一种交互图, 它强调消息跨越不同对象或参与者的实际时间, 而不仅仅只是关心消息的相对顺序
状态图 (State Diagram)	状态图描述一个状态机, 它由状态、转移、事件和活动组成。状态图给出了对象的动态视图。它对于接口、类或协作的行为建模尤为重要, 而且它强调事件导致的对象行为, 这非常有助于对反应式系统建模
活动图 (Activity Diagram)	活动图将进程或其他计算结构展示为计算内部一步步的控制流和数据流。活动图专注于系统的动态视图。它对系统的功能建模和业务流程建模特别重要, 并强调对象间的控制流程
部署图 (Deployment Diagram)	部署图描述对运行时的处理节点及在其中生存的构件的配置。部署图给出了架构的静态部署视图, 通常一个节点包含一个或多个部署图
制品图 (Artifact Diagram)	制品图描述计算机中一个系统的物理结构。制品包括文件、数据库和类似的物理比特集合。制品图通常与部署图一起使用。制品也给出了它们实现的类和构件
包图 (Package Diagram)	包图描述由模型本身分解而成的组织单元, 以及它们之间的依赖关系
交互概览图 (Interaction Overview Diagram)	交互概览图是活动图和顺序图的混合物

4) UML 视图

UML 对系统架构的定义是系统的组织结构, 包括系统分解的组成部分, 以及它们的关联性、交互机制和指导原则等提供系统设计的信息, 包括 5 个系统视图:

- 逻辑视图: 逻辑视图也称为设计视图, 它表示了设计模型中在架构方面具有重要意义的部分, 即类、子系统、包和用例实现的子集。
- 进程视图: 进程视图是可执行线程和进程作为活动类的建模, 它是逻辑视图的一次执行实例, 描述了并发与同步结构。
- 实现视图: 实现视图对组成基于系统的物理代码的文件和构件进行建模。

- 部署视图：部署视图把构件部署到一组物理节点上，表示软件到硬件的映射和分布结构。
- 用例视图：用例视图是最基本的需求分析模型。

另外，UML 还允许在一定的阶段隐藏模型的某些元素，遗漏某些元素，可不保证模型的完整性，但模型逐步地要达到完整和一致。

4. 面向对象分析

OOA 的基本任务是运用 OO 方法，对问题域进行分析和理解，正确认识其中的事物及它们之间的关系，找出描述问题域和系统功能所需的类和对象，定义它们的属性和职责以及它们之间所形成的各种联系。最终产生一个符合用户需求，并能直接反映问题域和系统功能的 OOA 模型及其详细说明。OOA 模型独立于具体实现，即不考虑与系统具体实现有关的因素，这也是 OOA 和 OOD 的区别之所在。OOA 的任务是“做什么”，OOD 的任务是“怎么做”。

面向对象分析阶段的核心工作是建立系统的用例模型与分析模型。

1) 用例模型

结构化分析（Structured Analysis, SA）方法采用功能分解的方式来描述系统功能，在这种表达方式中，系统功能被分解到各个功能模块中，通过描述细分的系统模块的功能来达到描述整个系统功能的目的。采用 SA 方法来描述系统需求，很容易混淆需求和设计的界限，这样的描述实际上已经包含了部分的设计在内。因此，系统分析师常常感到迷惑，不知道系统需求应该详细到何种程度。一个极端的做法就是将需求详细到概要设计，因为这样的需求描述既包含了外部需求也包含了内部设计。SA 方法的另一个缺点是分割了各项系统功能的应用环境，从各项功能项入手，很难了解到这些功能项如何相互关联来实现一个完整的系统服务。

从用户的角度来看，他们并不想了解系统的内部结构和设计，他们所关心的是系统所能提供的服务，这就是用例方法的基本思想。用例方法是一种需求合成技术，先获取需求并记录下来，然后从这些零散的要求和期望中进行整理与提炼，从而建立用例模型。在 OOA 方法中，构建用例模型一般需要经历四个阶段，分别是识别参与者、合并需求获得用例、细化用例描述和调整用例模型，如表 5-4 所示。其中前三个阶段是必须的。

表 5-4 构建用例模型的阶段

阶段	说明
识别参与者	参与者是与系统交互的所有事物，该角色不仅可以由人承担，还可以是其他系统和硬件设备，甚至是系统时钟。参与者一定在系统之外，不是系统的一部分。可以通过下列问题来帮助系统分析师发现系统的参与者：谁使用这个系统，谁安装这个系统，谁启动这个系统，谁维护这个系统，谁关闭这个系统，哪些（其他的）系统使用这个系统，谁从这个系统获取信息，谁为这个系统提供信息，是否有事情自动在预计的时间发生
合并需求获得用例	参与者都找到之后，接下来就是仔细地检查参与者，为每一个参与者确定用例。 ①要将获取到的需求分配给与其相关的参与者，以便可以针对每个参与者进行工作，而无遗漏； ②在合并之前，要明确为什么合并，知道了合并的目的，才可能选择正确的合并操作； ③将识别到的参与者和合并生成的用例，通过用例图的形式整理出来，以获得用例模型的框架

(续表)

阶段	说明
细化用例描述	用例建模的主要工作是书写用例规约(Use Case Specification),而不是画图。用例模板为一个给定项目的所有人员定义了用例规约的结果,其内容至少包括用例名、参与者、目标、前置条件、事件流(基本事件流和扩展事件流)和后置条件等,其他的还可以包括非功能需求和用例优先级等。
调整用例模型	<p>在建立了初步的用例模型后,还可以利用用例之间的关系来调整用例模型。用例之间的关系主要有包含、扩展和泛化。利用这些关系,把一些公共的信息抽取出来,以便于复用,使得用例模型更易于维护。</p> <ul style="list-style-type: none"> ● 包含关系:当可以从两个或两个以上的用例中提取公共行为时,应该使用包含关系来表示它们。其中这个提取出来的公用用例称为抽象用例,而把原始用例称为基本用例或基础用例 ● 扩展关系:如果一个用例明显地混合了两种或两种以上不同的场景,即根据情况可能发生多种分支,则可以将这个用例分为一个基本用例和一个或多个扩展用例,这样使描述可能更加清晰 ● 泛化关系:当多个用例共同拥有一种类似的结构和行为的时候,可以将它们的共性抽象成为父用例,其他的用例作为泛化关系中的子用例。在用例的泛化关系中,子用例是父用例的一种特殊形式,子用例继承了父用例所有的结构、行为和关系

2) 分析模型

前文从用户的观点对系统进行了用例建模,但捕获了用例并不意味着分析的结束,还要对需求进行深入分析,获取关于问题域本质内容的分析模型。分析模型描述系统的基本逻辑结构,展示对象和类如何组成系统(静态模型),以及它们如何保持通信,实现系统行为(动态模型)。

为了使模型独立于具体的开发语言,系统分析师需要把注意力集中在概念性问题上而不是软件技术问题上,这些技术的起点就是领域模型。领域模型又称为概念模型或简称为域模型,也就是找到那些代表事物与概念的对象,即概念类。概念类可以从用例模型中获得灵感,经过完善将形成分析模型中的分析类。每一个用例对应一个类图,描述参与这个用例实现的所有概念类,而用例的实现主要通过交互图来表示。例如,用例的事件流会对应产生一个顺序图,描述相关对象如何通过合作来完成整个事件流,复杂的备选事件流也可以产生一个或多个顺序图。所有这些图的集合就构成了系统的分析模型。

建立分析模型的过程大致包括定义概念类,确定类之间的关系,为类添加职责,建立交互图等,其中有学者将前三个步骤统称为类-责任-协作者(Class-Responsibility-Collaborator,CRC)建模。类之间的主要关系有关联、依赖、泛化、聚合、组合和实现等,它们在UML中的表示方式,如表5-5所示。

表5-5 类之间的关系

关系	图例	描述
关联		关联提供了不同类的对象之间的结构关系,它在一段时间内将多个类的实例连接在一起。关联体现的是对象实例之间的关系,而不表示两个类之间的关系。其余的关系涉及类元自身的描述,而不是它们的实例。对于关联关系的描述,可以使用关联名称、角色、多重性和导向性来说明

(续表)

关系	图例	描述
依赖		两个类 A 和 B，如果 B 的变化可能会引起 A 的变化，则称类 A 依赖于类 B。依赖可以由各种原因引起，例如，一个类向另一个类发送消息，一个类是另一个类的数据成员，一个类是另一个类的某个操作参数等
泛化		泛化关系描述了一般事物与该事物中的特殊种类之间的关系，也就是父类与子类之间的关系。继承关系是泛化关系的反关系，也就是说，子类继承了父类，而父类则是子类的泛化
共享聚集		共享聚集关系通常简称为聚合关系，它表示类之间的整体与部分的关系，其含义是“部分”可能同时属于多个“整体”，“部分”与“整体”的生命周期可以不相同。例如，汽车和车轮就是聚合关系，车子坏了，车轮还可以用；车轮坏了，可以再换一个新的
组合聚集		组合聚集关系通常简称为组合关系，它也是表示类之间的整体与部分的关系。与聚合关系的区别在于，组合关系中的“部分”只能属于一个“整体”，“部分”与“整体”的生命周期相同，“部分”随着“整体”的创建而创建，也随着“整体”的消亡而消亡。例如，一个公司包含多个部门，它们之间的关系就是组合关系。公司一旦倒闭，也就没有部门了
实现		实现关系将说明和实现联系起来。接口是对行为而非实现的说明，而类中则包含了实现的结构。一个或多个类可以实现一个接口，而每个类分别实现接口中的操作

5.1.3 软件设计

软件设计是需求分析的延伸与拓展。需求分析阶段解决“做什么”的问题，而软件设计阶段解决“怎么做”的问题。同时，它也是系统实施的基础，为系统实施工作做好铺垫。合理的软件设计方案既可以保证系统的质量，也可以提高开发效率，确保系统实施工作的顺利进行。从方法上来说，软件设计分为结构化设计与面向对象设计。

1. 结构化设计

结构化设计（Structured Design, SD）是一种面向数据流的方法，它以 SRS 和 SA 阶段所产生的 DFD 和数据字典等文档为基础，是一个自顶向下、逐步求精和模块化的过程。SD 方法的基本思想是将软件设计成由相对独立且具有单一功能的模块组成的结构，分为概要设计和详细设计两个阶段，其中概要设计又称为总体结构设计，它是开发过程中很关键的一步，其主要任务是将系统的功能需求分配给软件模块，确定每个模块的功能和调用关系，形成软件的模块结构图，即系统结构图。在概要设计中，将系统开发的总任务分解成许多个基本的、具体的任务，而为每个具体任务选择适当的技术手段和处理方法的过程称为详细设计。根据任务的不同，详细设计又可分为多种，例如，输入 / 输出设计、处理流程设计、数据存储设计、用户界面设计、安全性和可靠性设计等。