

ITIL先锋论坛

# ITIL V3 2011公益讲座

主讲：问静园

13810105539@139.com

神州数码服务交付中心 高级项目经理

2012.3.8 20:00—22:00

BEST PRACTICE

# ITIL®

## A Pocket Guide

2011 EDITION

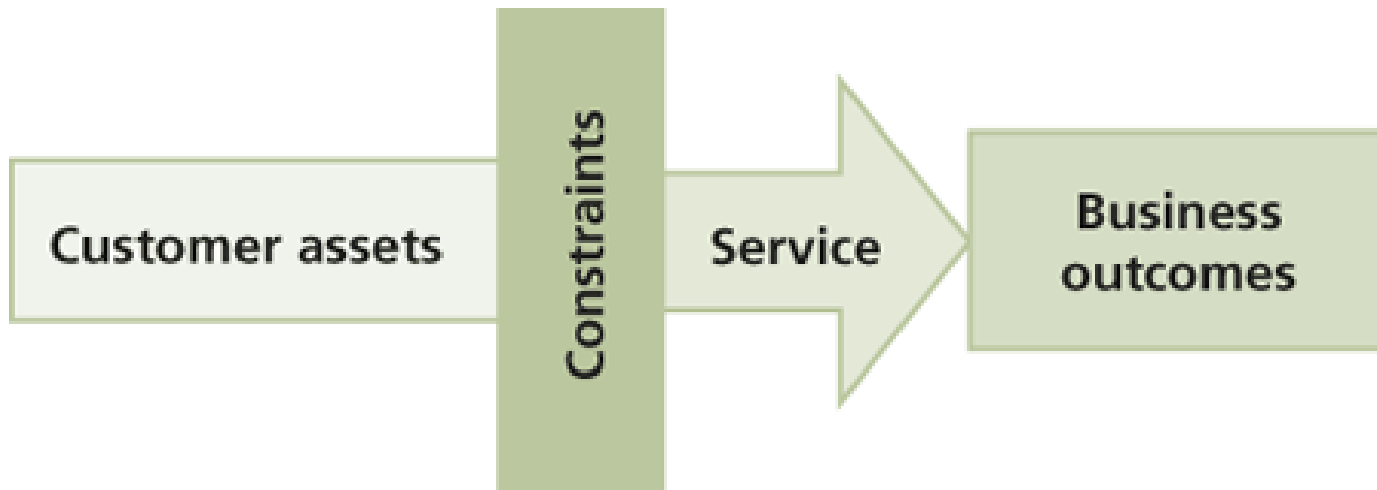


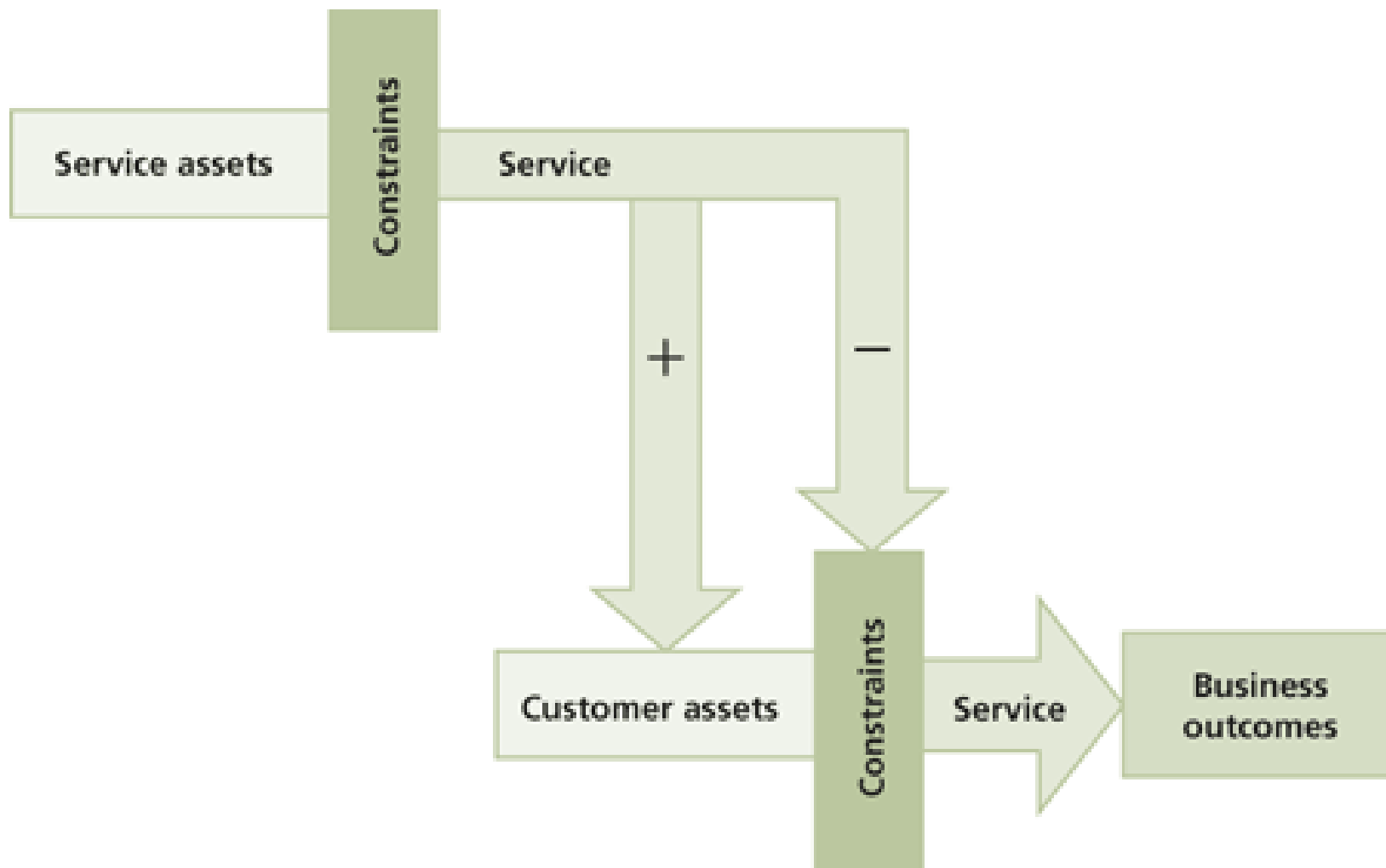
Copyright protected. Use is for Single Users only via a VHP Approved License.  
For information and printed versions please see [www.vanharen.net](http://www.vanharen.net)

- 流程
- 职能
- 程序
  - 执行一个活动或者流程的特殊方法，步骤
  - 一个程序被描述为“HOW”，并且描述“WHO”执行活动
  - 一个程序可以包括来自不同流程的阶段
- 一套工作指导
  - 在程序中的一个或者更多的活动将被详细的执行，包括用什么技术和资源

# 资产

- 客户资产
- 服务资产
- 战略资产





# 组织结构

- **流程所有者**
  - 对流程结果负责
- **流程经理**
  - 对流程的结构负责，报告给流程所有者
- **流程实践者**
  - 定义活动，并报告给流程经理

# 财务管理

对于IT服务的财务管理的职责是管理IT服务提供者的**预算、核算和收费**要求的流程。被用于定量IT服务部门给业务部门的价值。通常回答下面的问题：

- 哪一个服务的成本最高？为什么？

- 我们最大的无效服务在哪里？

## 基本概念

- 利润中心： 外部资金（为外部提供服务所带来的收入）

- 成本中心： 内部资金（同一组织的其他业务单元）

# 资金模型帮助定义IT服务提供者什么时候或如何用资金

- 增量预算
  - 头一个循环期作为预算的基础，循环期可以是月、年等。
- 触发预算
  - 当一种特殊情况或者事件发生时，触发器和变更相关
- 零基预算
  - 忽略过去的经验，从零开始（更多的内部服务者用这种 模型）



# 预算（Budgeting）

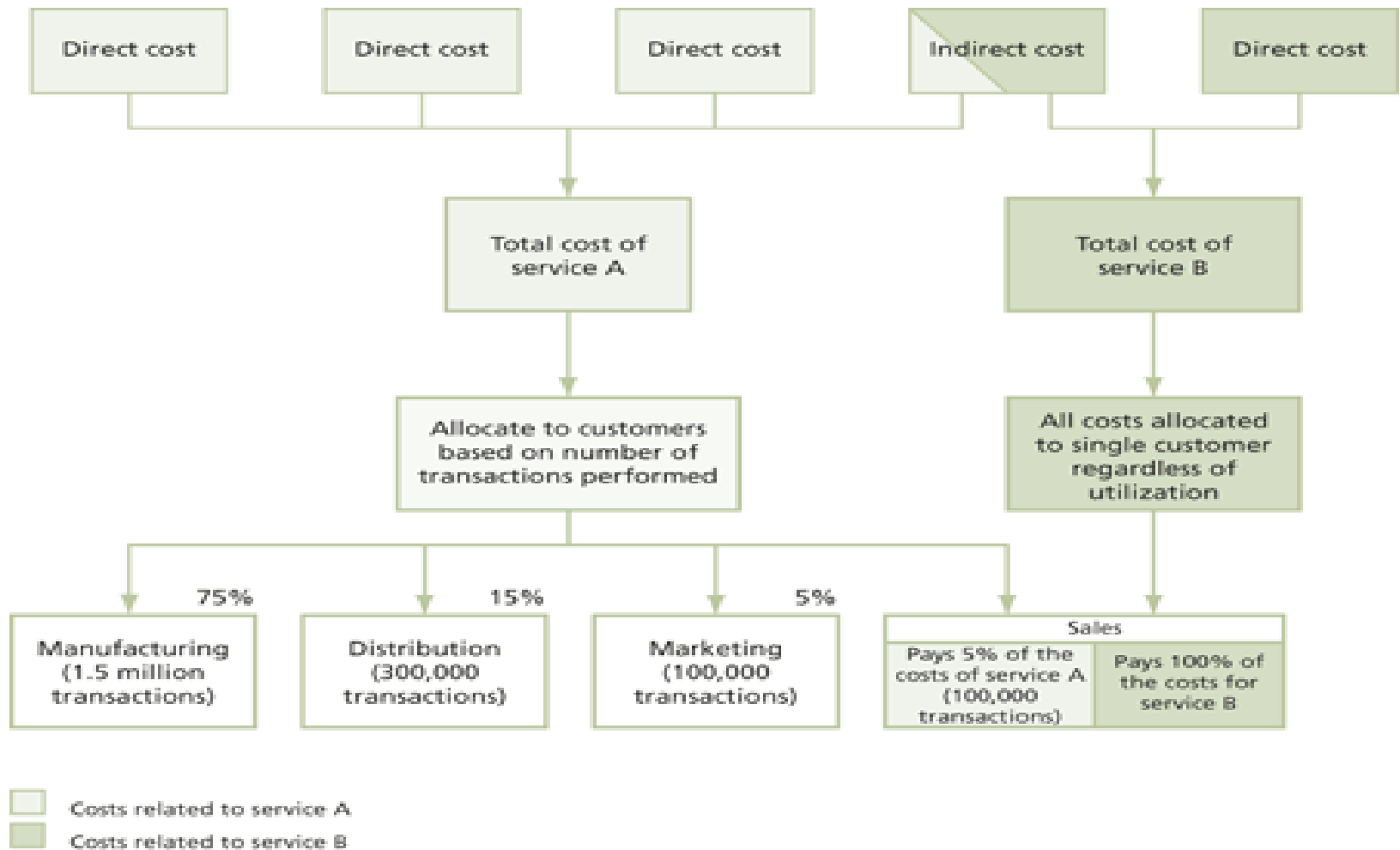
- 预算使得组织可以
  - 预测在某个时间段运行IT需要的资金
  - 保证在任何时间实际的开销可以与预期的花费进行对比
  - 降低超支的风险
  - 确保收入（revenue）可以负担开销(spending)

# 核算方式

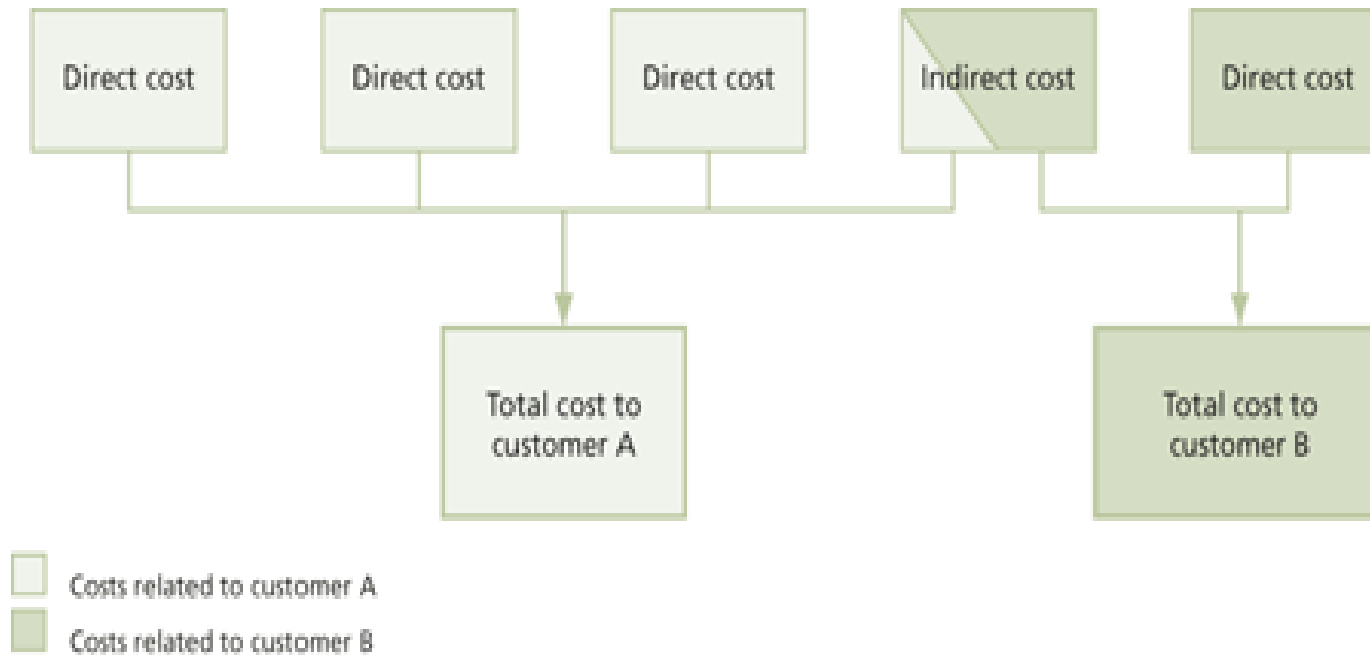
- IT组织的成本
- 服务成本
- 客户成本—很少用这种，比如：一个客户的License
- 位置成本-也很少用，包括了IT服务者与不同地点的客户沟通的成本



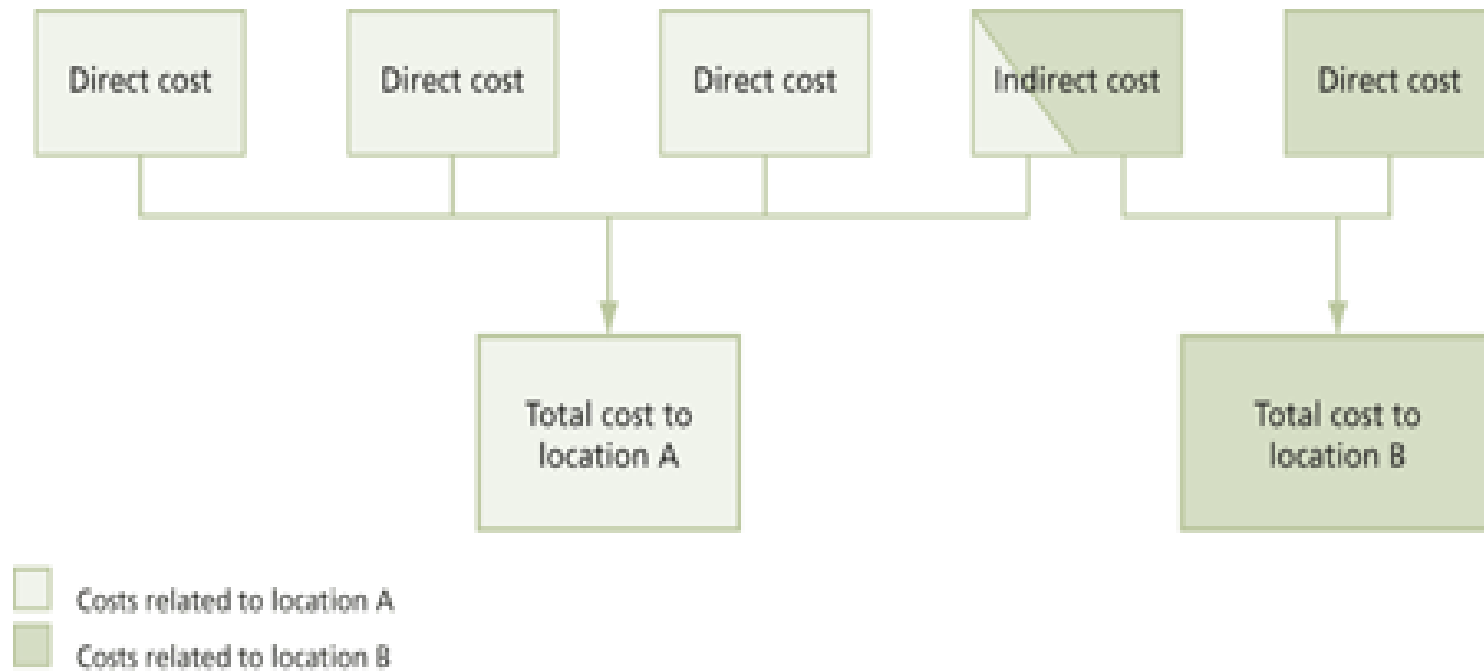
Cost by IT organization



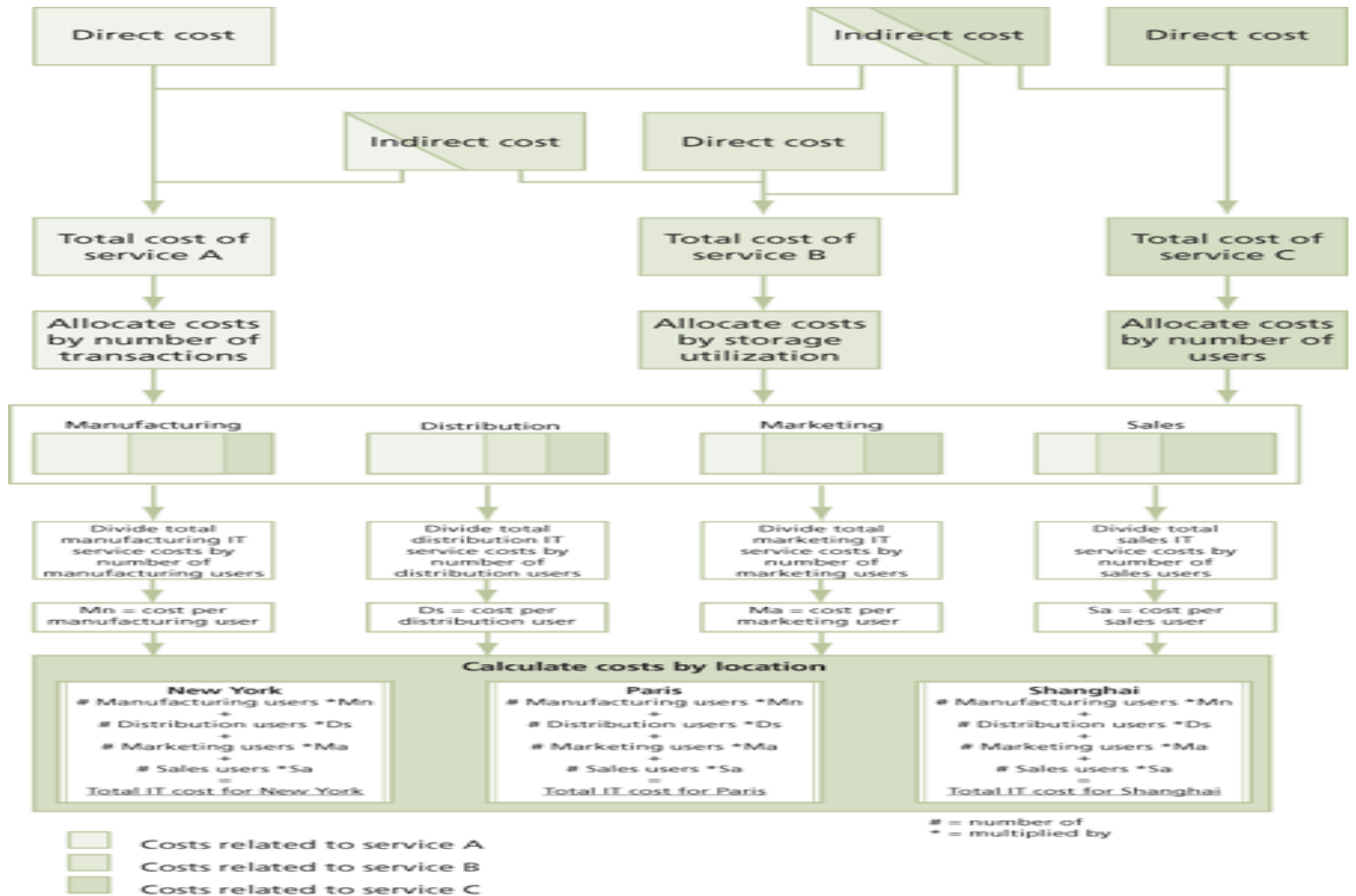
Cost by service



Cost by customer



Cost by location



# 成本类型和元素

类别	成本项目	所属子项
材料	设备成本单元 ECU, Equipment Cost Unit	服务器
		磁盘阵列
		路由器
	软件成本单元 SCU, Software Cost Unit	应用程序采购
		许可授权费用
人力	组织成本单元 OCU, Organisation Cost Unit	支持服务人员
		培训
间接成本	场地成本单元 ACU, Accommodation Cost Unit	机房
		办公室



# 成本分类

**按照投入成本可分为：**

- 资本性支出：资本支出是指固定资产购置或改良成本。例如，服务器淘汰更换或网络系统升级通常属于资本支出范畴。
- 营运支出：营运成本包括 IT 环境运转维护所需开支

**按照可追溯性成本可划分为：**

- 直接成本和间接成本：直接成本是直接归属于某项活动或某个部门的成本费用，间接成本无法直接归属于某个部门客户，可按比例由全体部门分担。

**按照状态成本可分为：**

- 固定成本和变动成本：固定成本是即使未发生任何业务，仍需为保持企业运转而耗用资金，例如、硬件设备维保合同。变动成本将资源使用状况成比例变化，例如、当服务台资源使用频率上升时，客户所需负担的成本也相应提高。

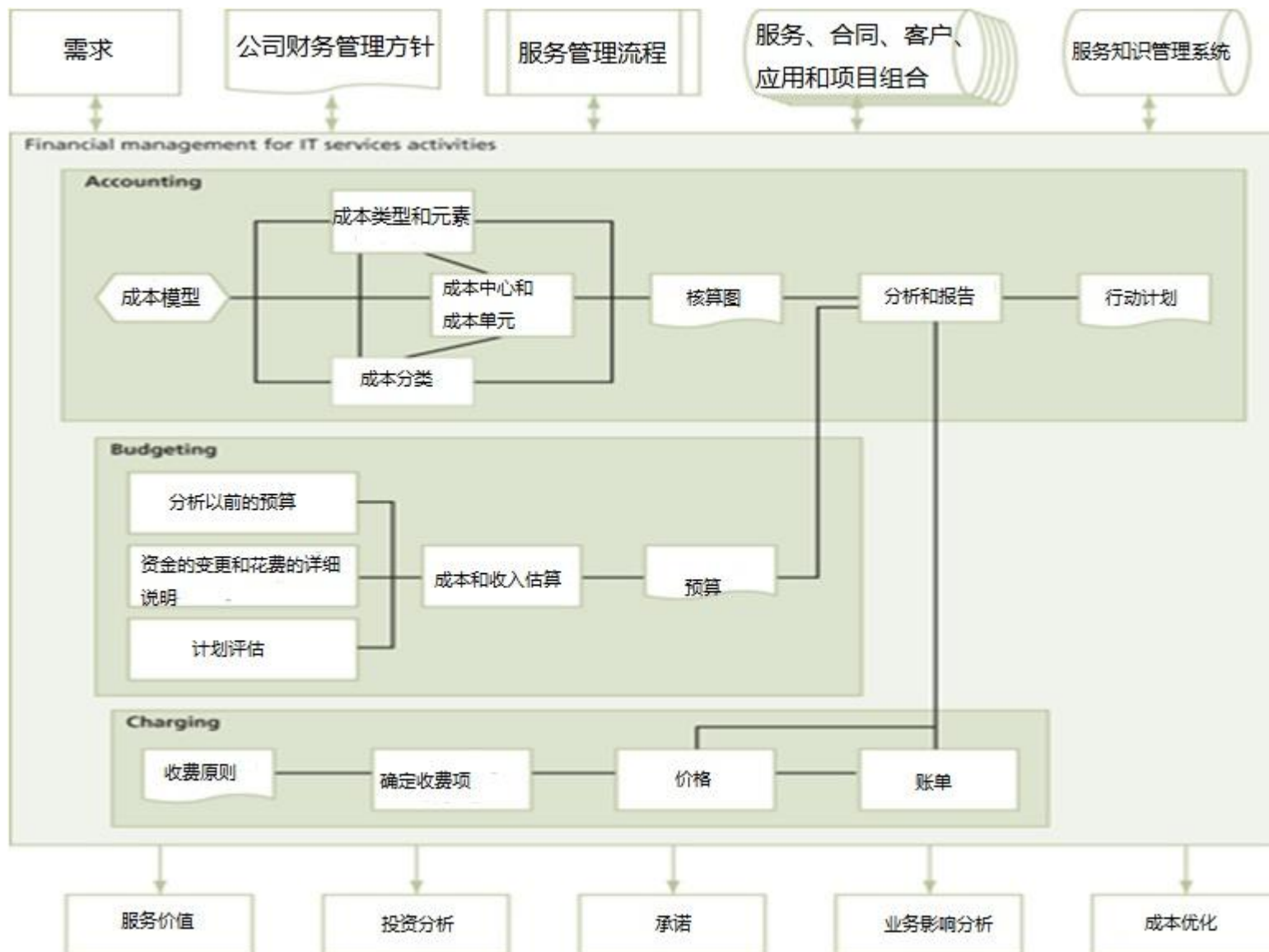
# 收费模式

- 成本
- 成本加
- 现行价格
- 市场价格
- 固定价格
- 分级定义
- 不同的收费

# 账单

- 无账单：内部提供者通过分摊的方式，外部的需要给客户id提供发票
- 信息账单：内部提供者给内部客户提供的账单信息
- 账单：

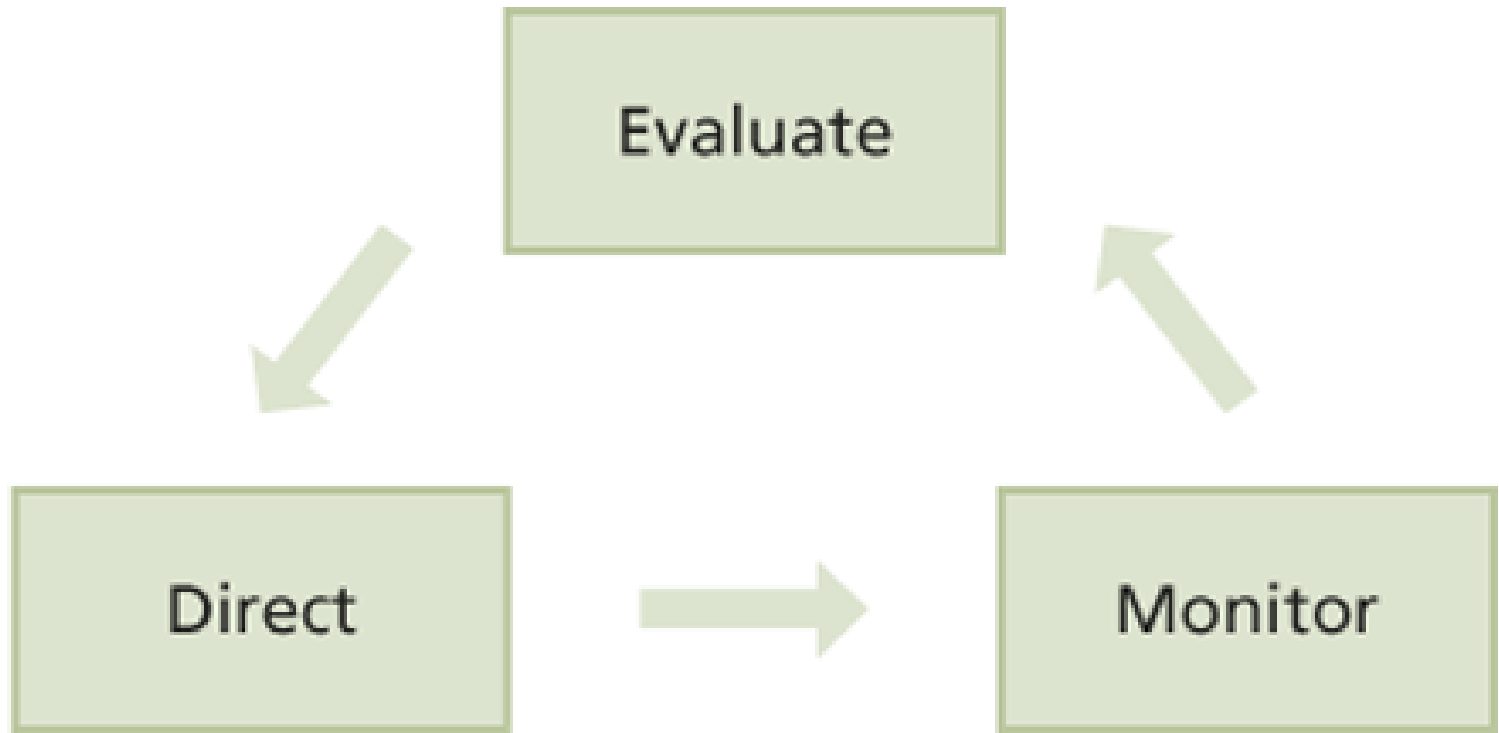
# 财务管理实施流程



# 业务关系管理

- 业务关系管理的目标是建立和维护服务提供者与客户之间的关系。
- 服务提供者的目的是理解客户和他的业务需求。
- 业务关系管理识别客户正在改变的需求，并确保服务提供者能匹配它。业务关系管理也管理客户的期望。
- 指标是客户的满意度。

# 治理



治理的活动

# 云服务的特点

**On demand** Customers can access them whenever and wherever they need them through the internet (or intranet).

**Ubiquitous access** Any client can access these services, even thin clients, because they use standard networking mechanisms and protocols.

**Resource pooling** Cloud services are provided by pooled resources which, because they are based on internet access, can be located anywhere. Physical and virtual devices are dynamically assigned and reassigned according to customer demand. Although customers do not have control over the exact location of these devices, they are able to specify high-level requirements. For example, they can specify that customer data be located in a specific country to comply with data privacy legislation.

**Rapid elasticity** Services can be provisioned to rapidly scale up or down to meet real-time customer demand.

**Measured services** Since cloud services are pay-per-use, it is important that their usage be metered, and that a financial model exists to be able to bill for these services. Resource usage is monitored, controlled and reported.

# 云服务的类型

**Software as a Service (SaaS)**

**Platform as a Service (PaaS)**

**Infrastructure as a Service (IaaS)**



# 服务协调

用于监控每一个流程的文档、原则和标准是正确的；活动执行的及时性；流程之间的冲突；也管理着问题的升级。

For the overall service  
design lifecycle stage

Define and maintain  
policies and methods



Plan design resources  
and capabilities



Coordinate design  
activities



Manage design  
risks and issues



Improve service design

For each design

Plan individual designs



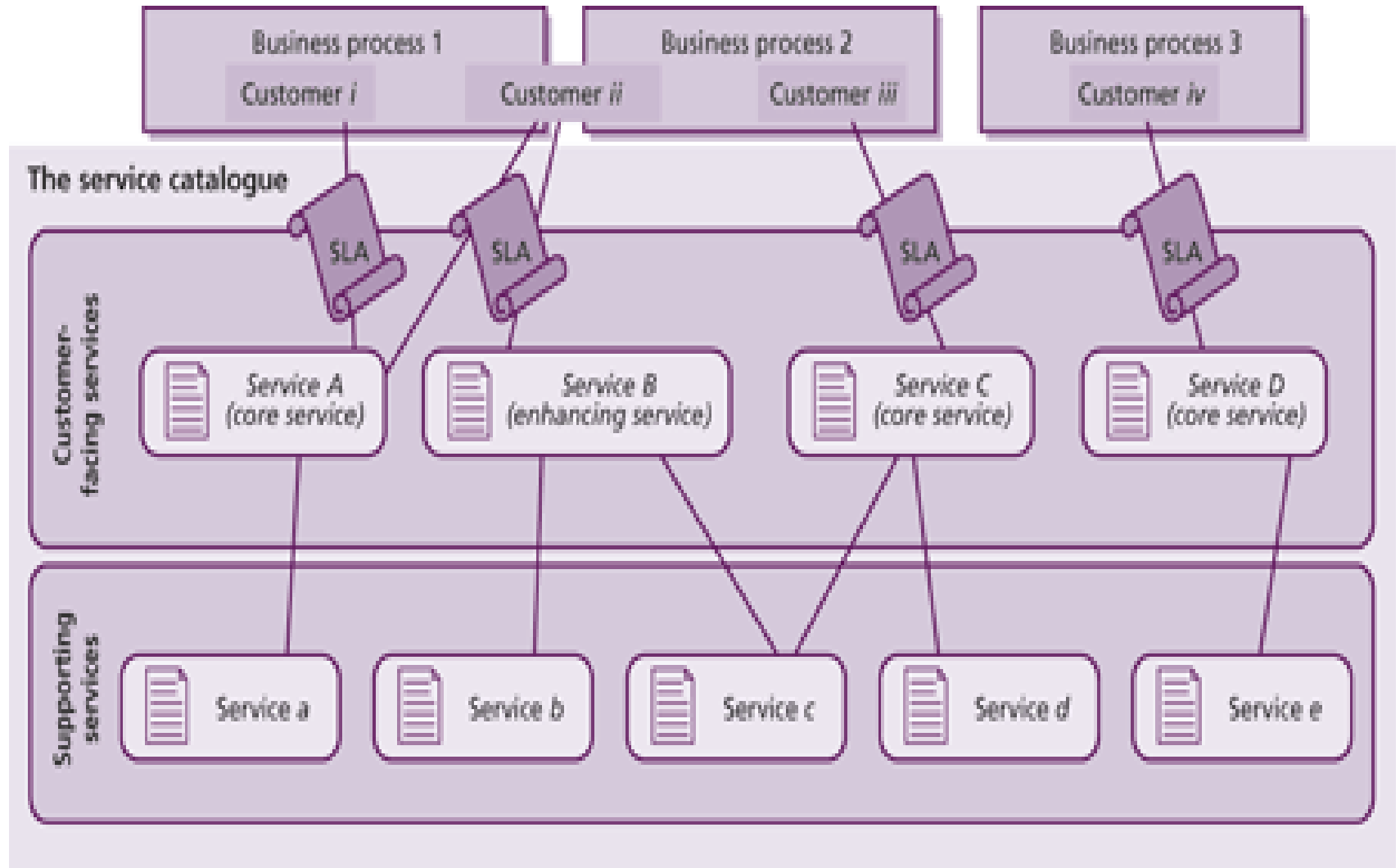
Coordinate individual  
designs



Monitor individual  
designs



Review designs and  
ensure handover of  
service design package



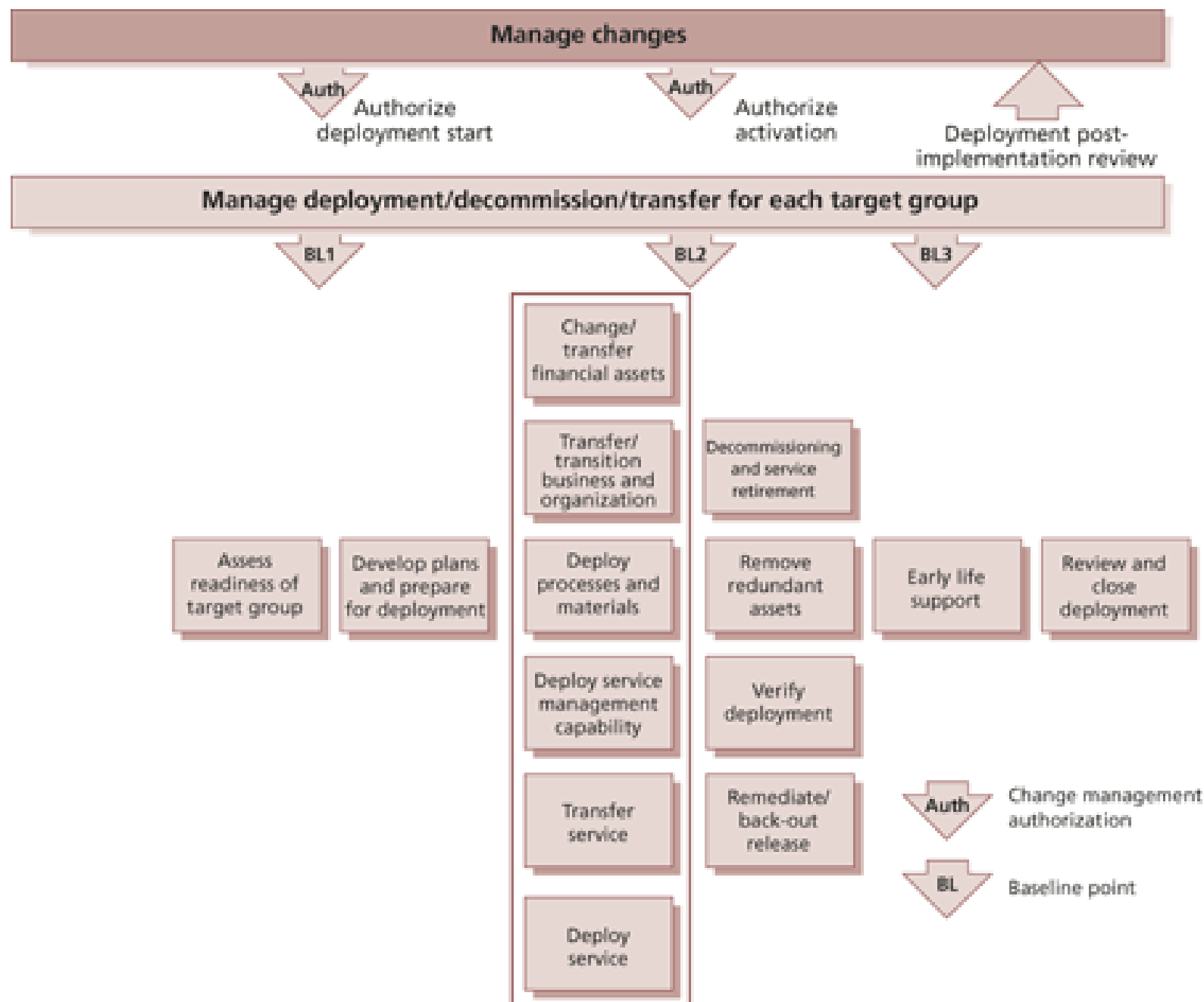
Types of service in a service catalogue

## **The change proposal should include:**

A high-level description of the new, changed or retired service, including business outcomes to be supported, and utility and warranty to be provided

A full business case including risks, issues and alternatives, as well as budget and financial expectations

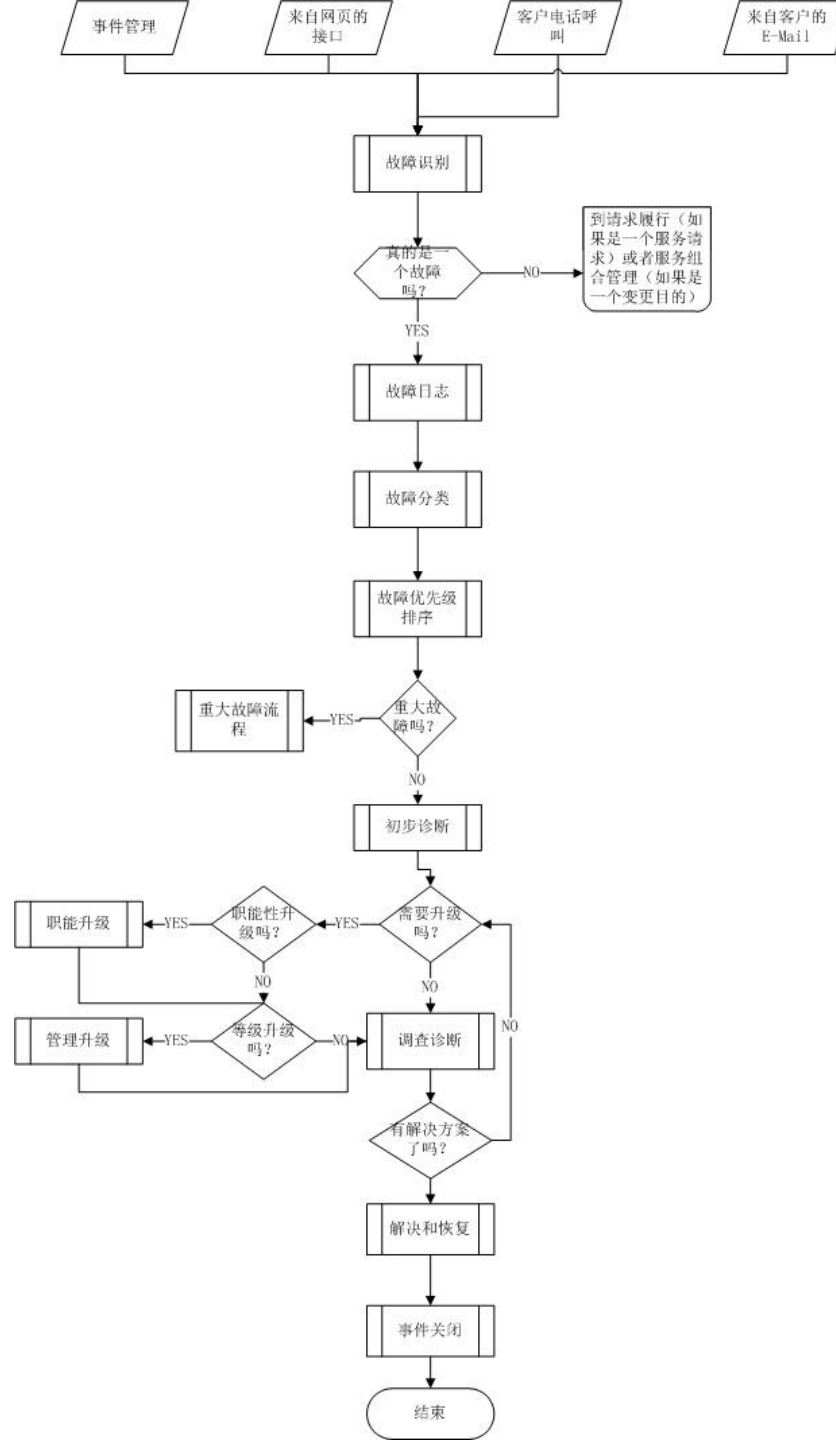
An outline schedule for design and implementation of the change.



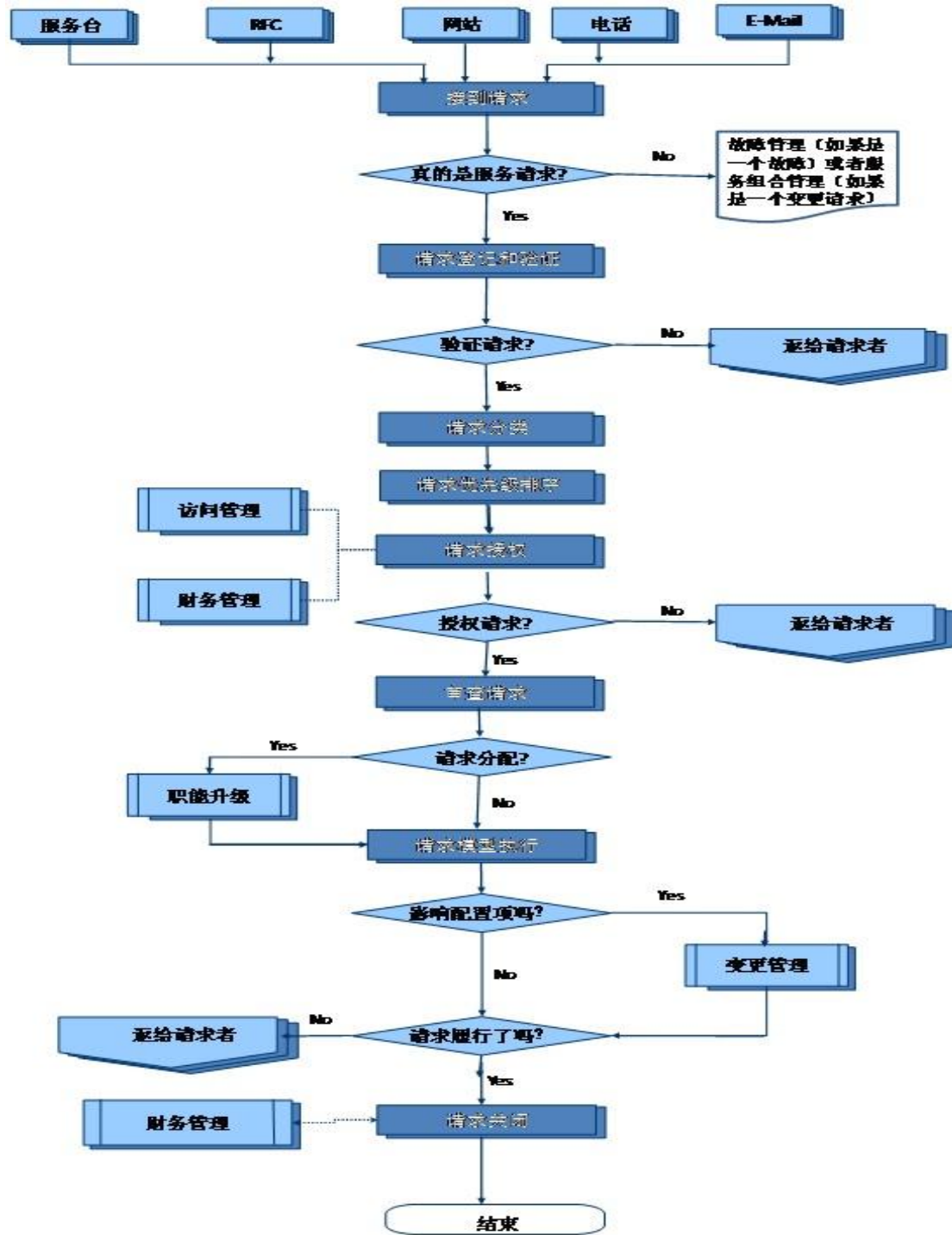
A **configuration item** (CI) is a service asset that needs to be managed in order to deliver an IT service. All CIs are service assets, but many service assets are not configuration items. Examples of configuration items are a server or a software licence. Every CI must be under the control of change management.

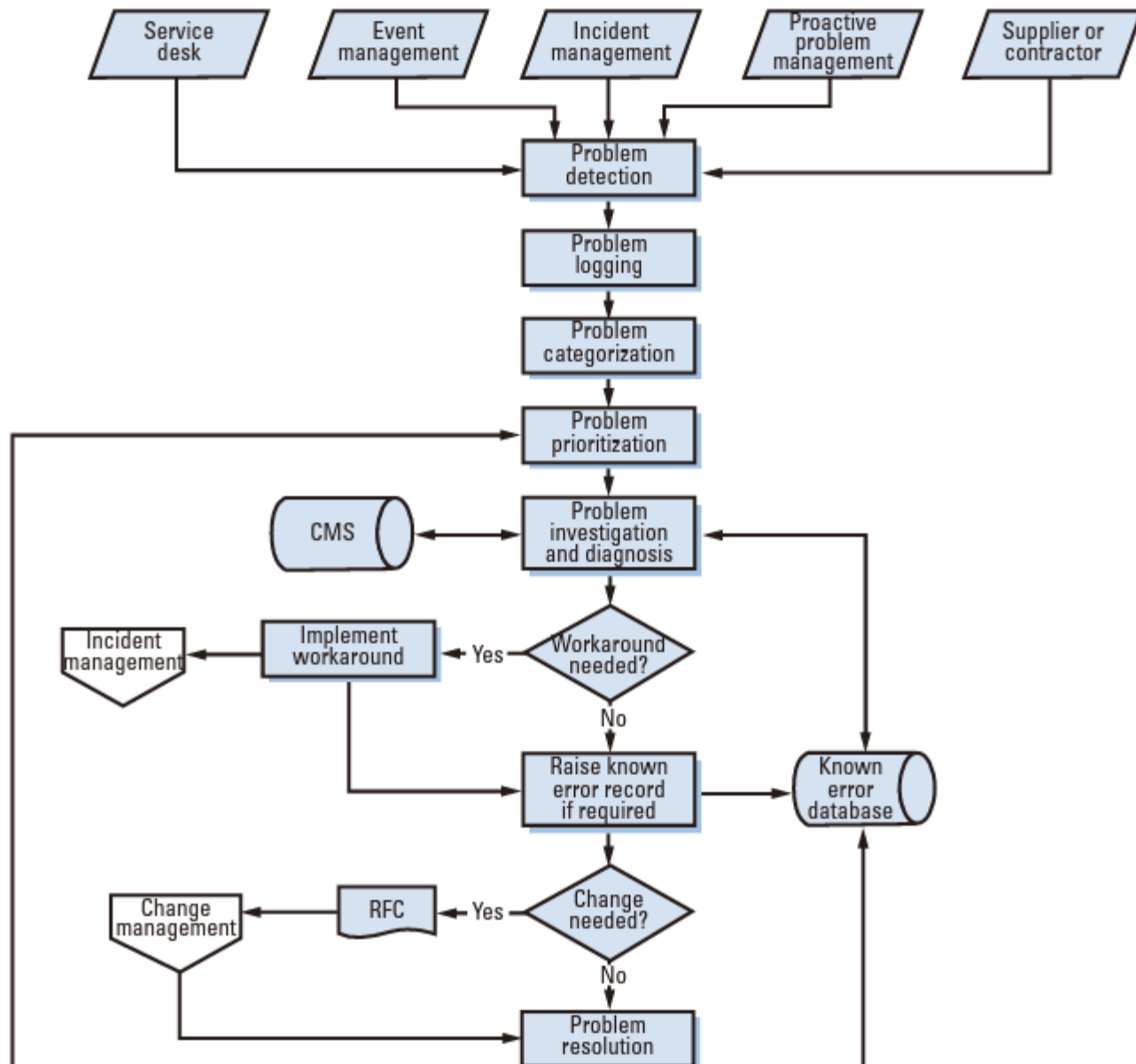
A **configuration record** is a set of attributes and relationships about a CI. Configuration records are stored in a configuration management database (CMDB) and managed with a configuration management system (CMS). It is important to note that CIs are not stored in a CMDB; configuration records describe CIs that are stored in the CMDB.

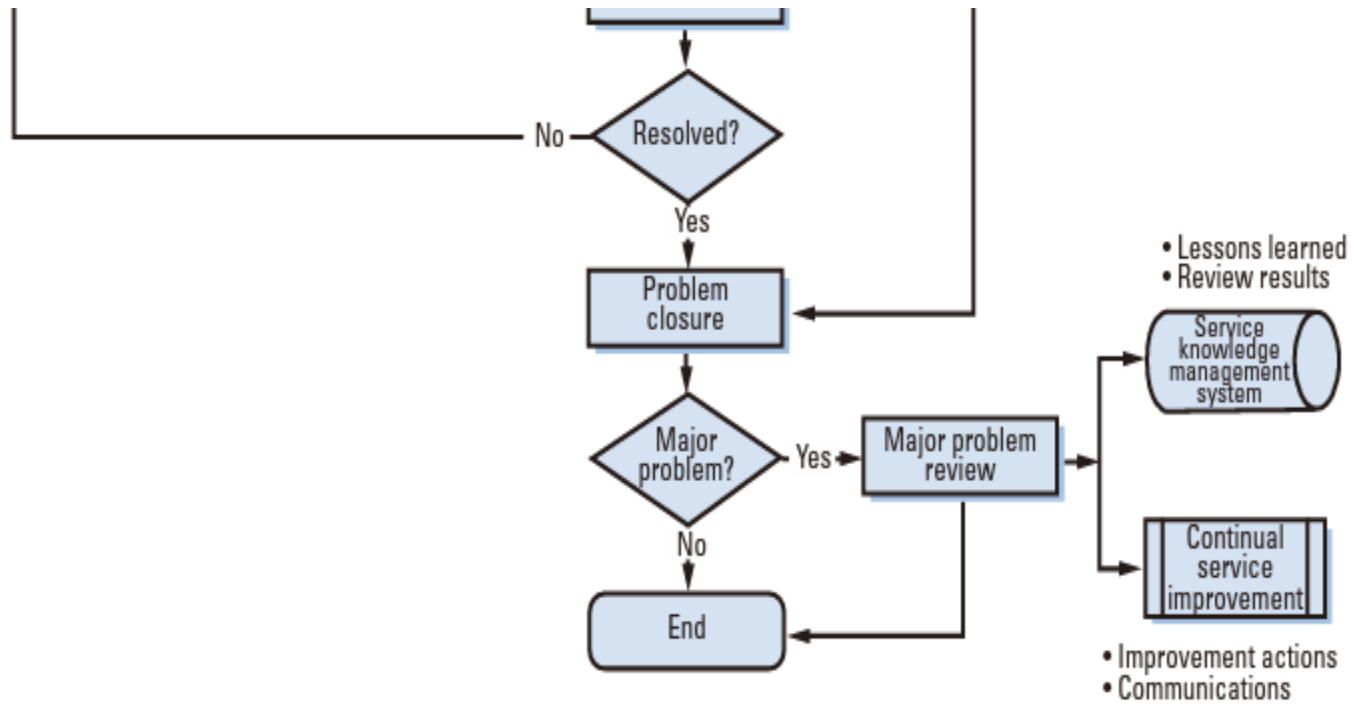
The **service knowledge management system** (SKMS) is a set of tools and databases that are used to manage knowledge, information and data. Many configuration items are available in the form of knowledge or information, and these are typically stored in the SKMS – for example, a service level agreement, a report template or a definitive media library. The SACM process is not responsible for managing the SKMS. Some items in the SKMS will be owned and managed by the SACM process, but others will be owned and managed by other processes or people.



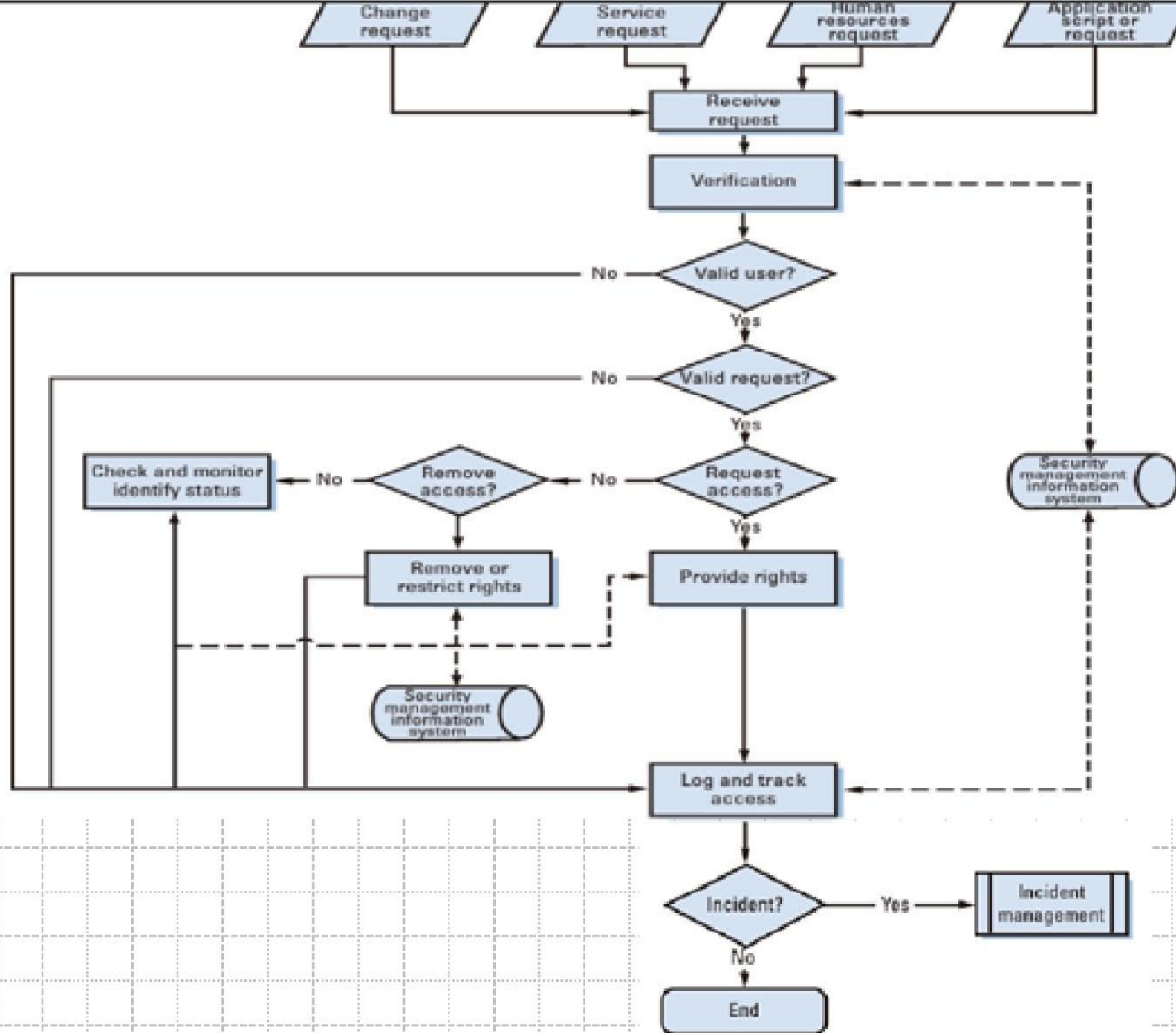








- *Proactive problem management* – Identifying and solving problems and known errors before further incidents related to them can occur. Proactive problem management includes the identification of trends or potential weaknesses. It is initiated by service operation, but usually driven by CSI.



# 应用管理与应用开发

to *application development*, which is mainly concerned with the one-time activities for requirements, design and build of applications, *application management* covers the entire ongoing lifecycle of an application, including requirements, design, build, deploy, operate and optimize.

