



CMDB**实施经验交流**

- ▶ 循序渐进构建CMDB
- ▶ CMDB演示



阶段 1: 组建项目团队和定义规划项目

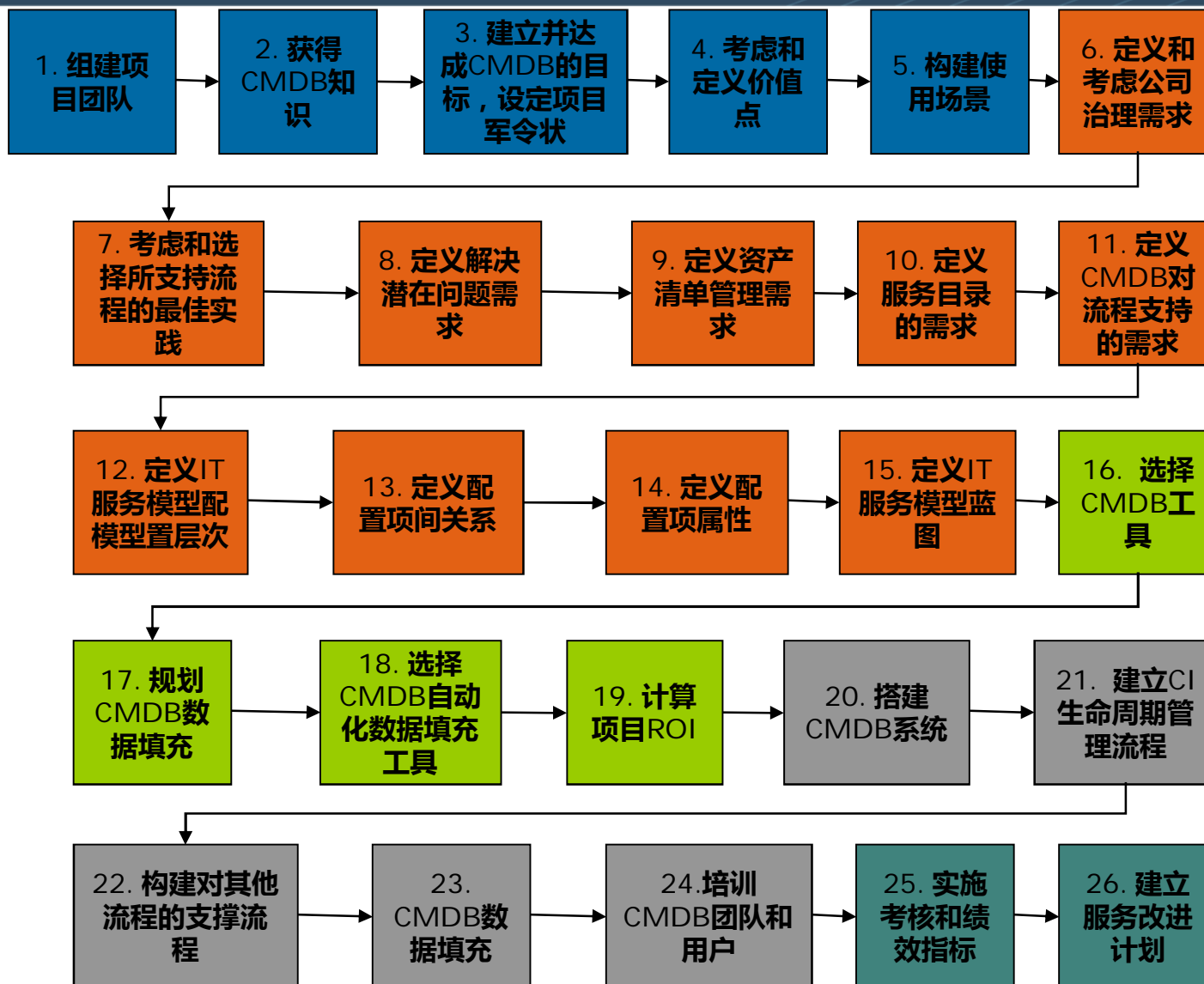
阶段2: 定义需求和建立IT服务模型蓝图

阶段3: 选择CMDB解决方案和工具

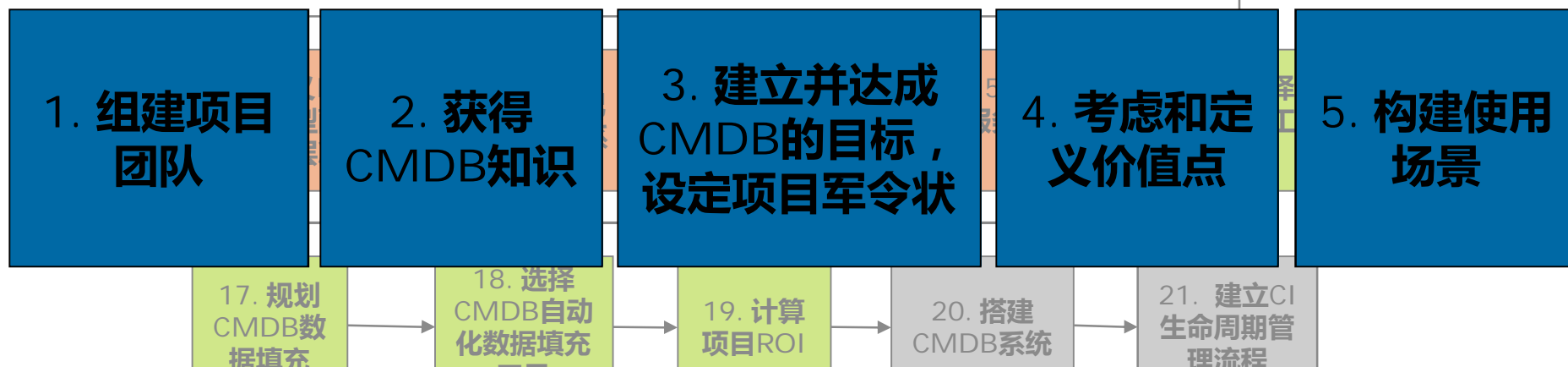
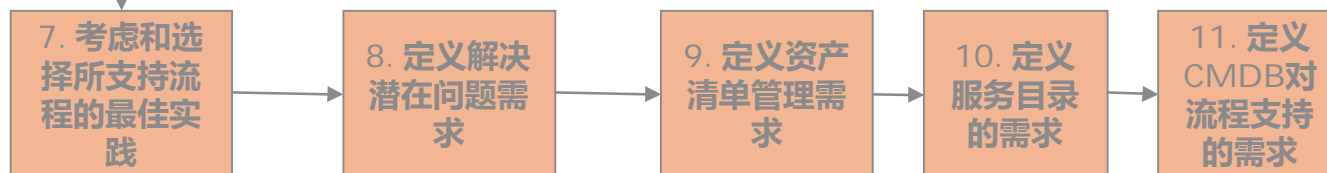
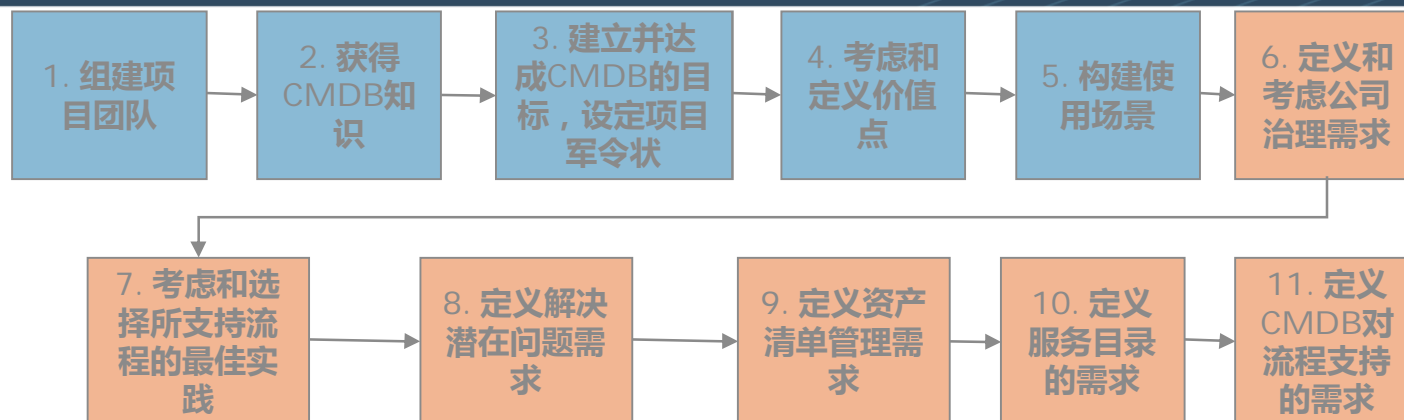
阶段4: 构建和维护CMDB系统

阶段5: 在建设过程中实现和挖掘价值

循序渐进构建CMDB--战术指导



循序渐进构建CMDB--战术指导



里程碑:
项目执行委员会的最高领导对项目的计划、经费和资源审批通过

- ▶ 项目权利机构—项目执行委员会
 - 保证项目目标的按日期按资源实现
 - 最少由以下成员：
 - Executive Sponsor -- 支持本项目的高层领导
 - Key Stakeholders -- 关键的利益干系人们
 - CMDB Owner – CMDB系统管理者
 - Project Manager – 项目经理

- ▶ 选择项目经理
 - 建议具有ITIL Service Manager (Masters) Certification
 - 建议有一定的IT Service Management背景，有数据库项目经验

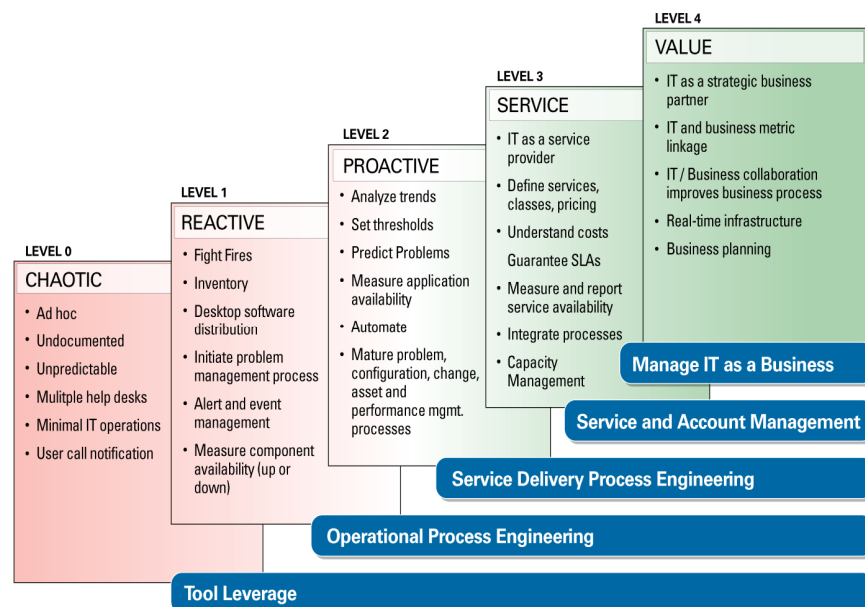
- ▶ 项目团队成员
 - 全职项目组人员
 - 各个技术方面的专家

步骤2 –培训和活动CMDB知识

- ▶ ITIL 意识培训和认证
 - ITIL Foundation
 - ITIL Practitioner – 配置管理
 - ITIL Service Manager
- ▶ 技术培训
 - BMC Atrium CMDB 管理员培训
 - BMC ADDM 管理员培训
- ▶ IT管理成熟度 “配置管理” 部分
 - 完成自评
 - 计划一年之后的级别



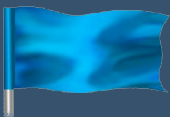
CA ITIL Service
Maturity Assessment



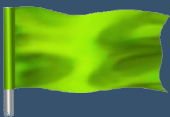
Source: Gartner Research (2005)

步骤3 - 建立并达成CMDB的目标，设定项目军令状

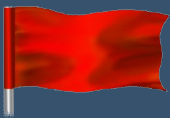
- ▶ 识别使用CMDB的IT部门
- ▶ 访谈利益相关人，汇总他们的关注点和优先级
- ▶ 编写和发布CMDB项目军令状
 - 使用SMART原则来描述；SMART = Specific, Measurable, Achievable, Relevant, Time-Based



2011年3月31日：采集所有基础架构CI和他们的关系，比对真实IT环境，校验所采集数据的正确性，纠正所有不正确的采集数据



2011年5月31日：对所有物理设备数据进行版本控制，形成精确完整的数据版本，并对数据实施配置审计。对此实现CMDB报表功能



2011年7月31日：实现事件管理、问题管理、变更管理对CMDB数据的使用；把软件类数据也纳入版本控制范围



2011年9月30日：实施2套业务系统的服务模型；在变更管理流程中应用这些模型。落实其他系统模型的实施计划

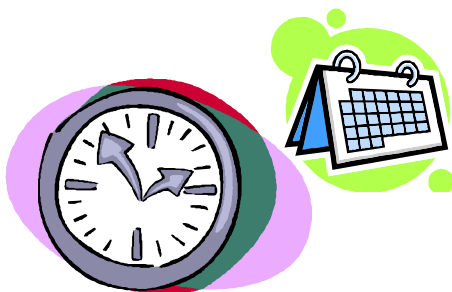


怎么做？

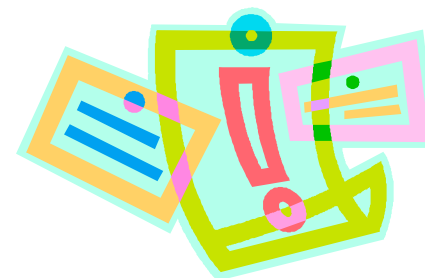
计划



谁参与？



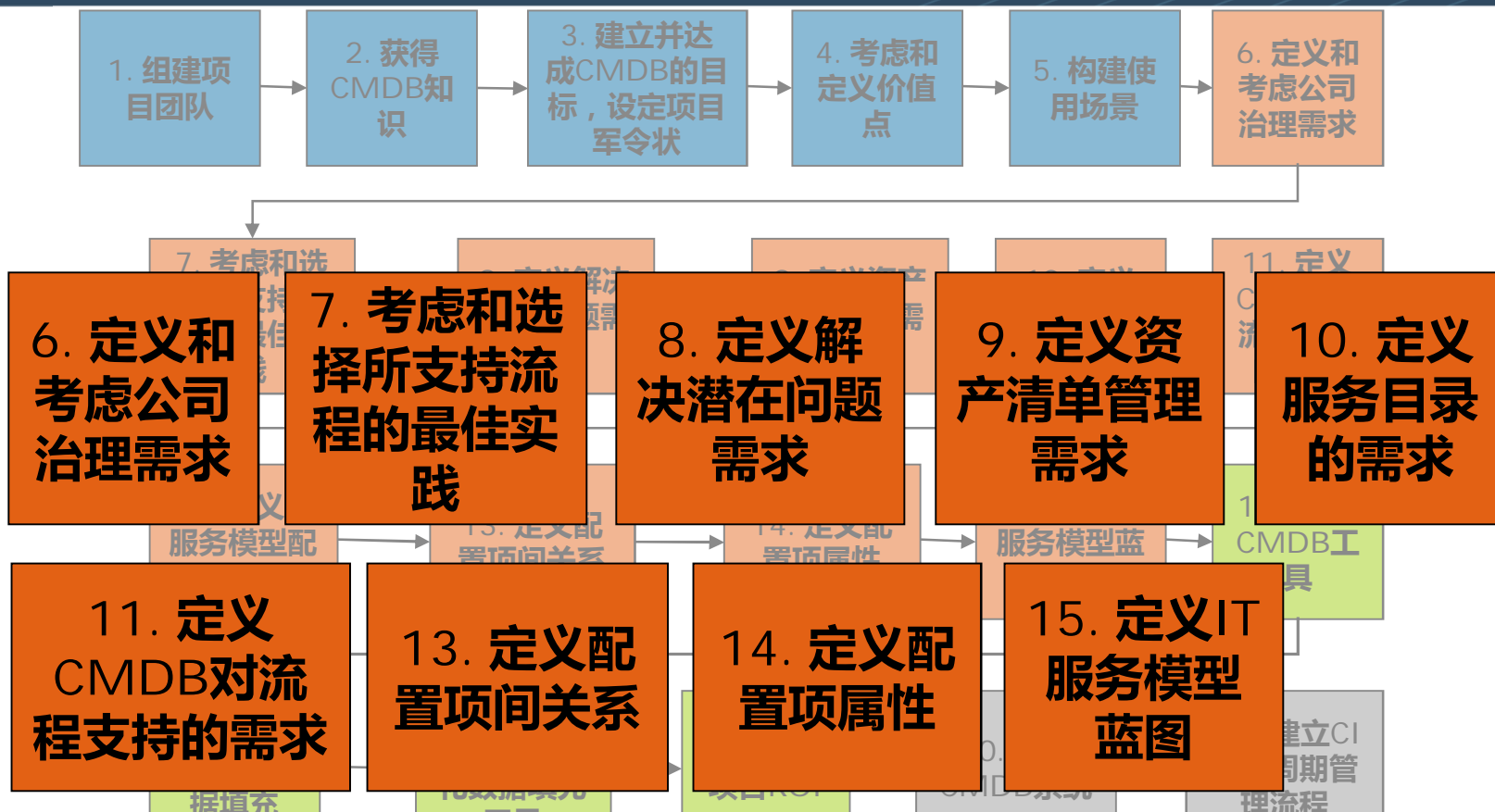
何时做？



做什么？

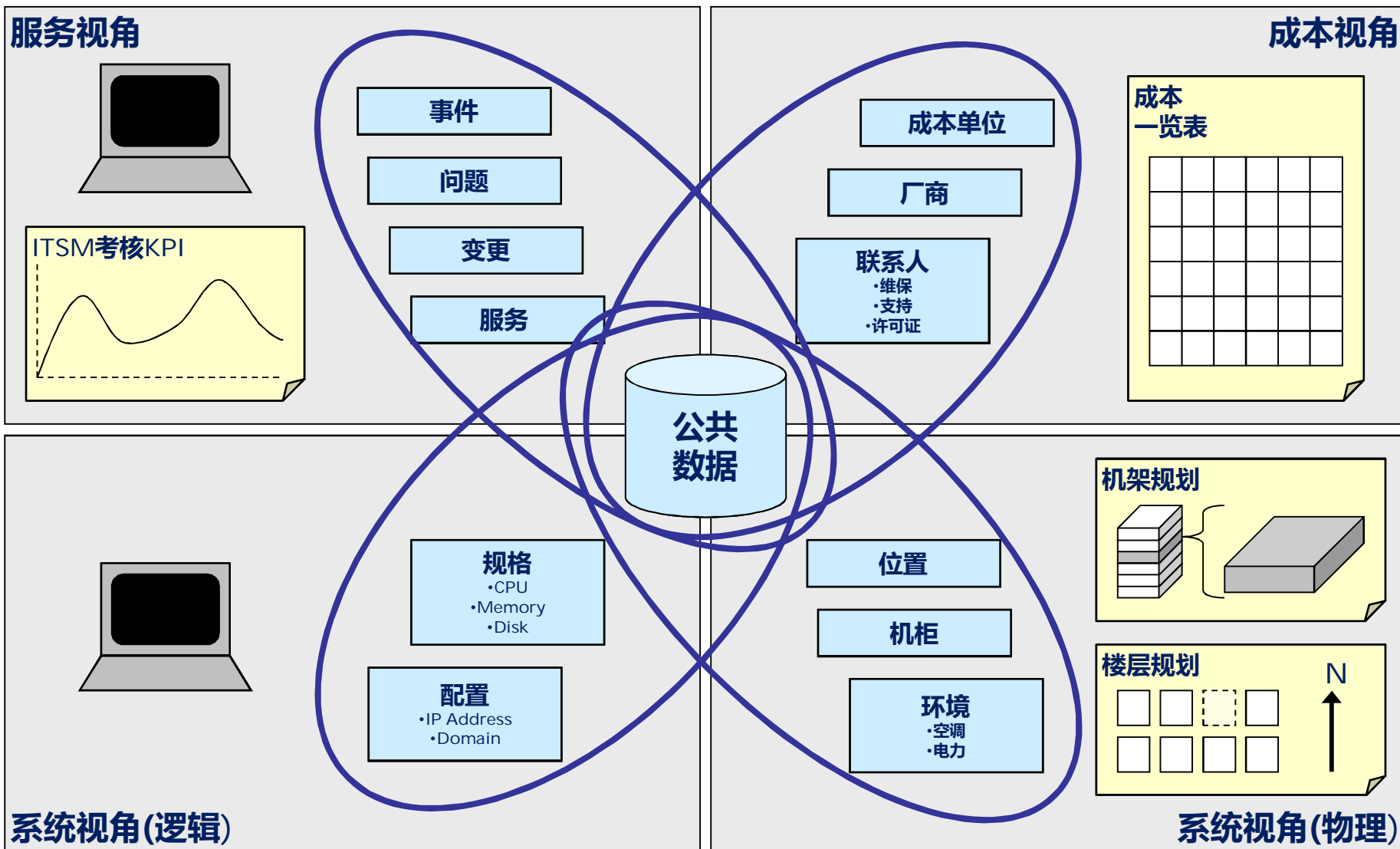
所有项目成功的3个关键因素: 1) 沟通； 2) 沟通； 3)沟通

循序渐进构建CMDB--战术指导



里程碑:
“项目需求文档”和“IT服务模型蓝图”被项目执行委员会审核通过

步骤9 - 定义资产清单管理需求

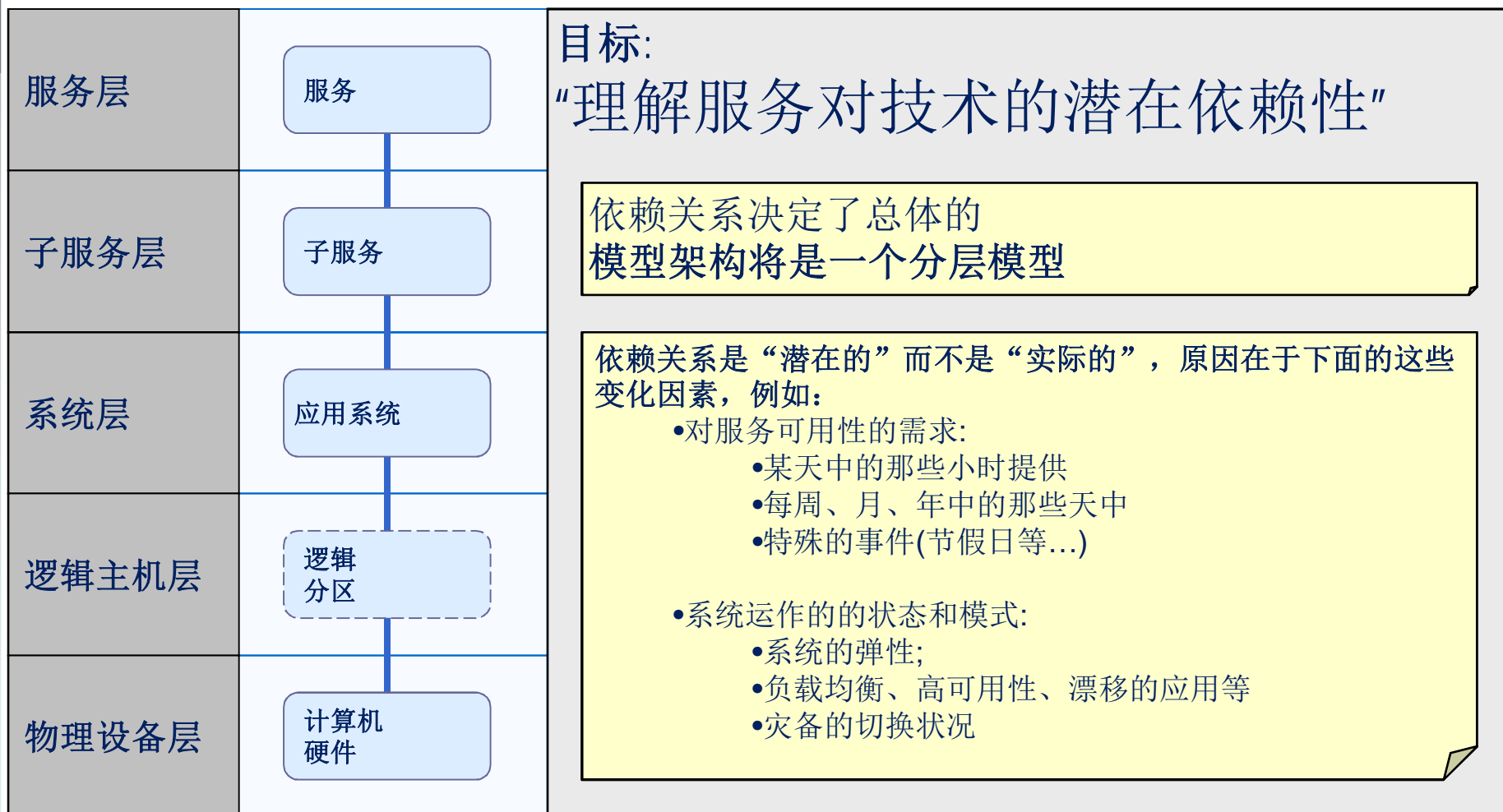


步骤9 - 定义资产清单管理需求（续）

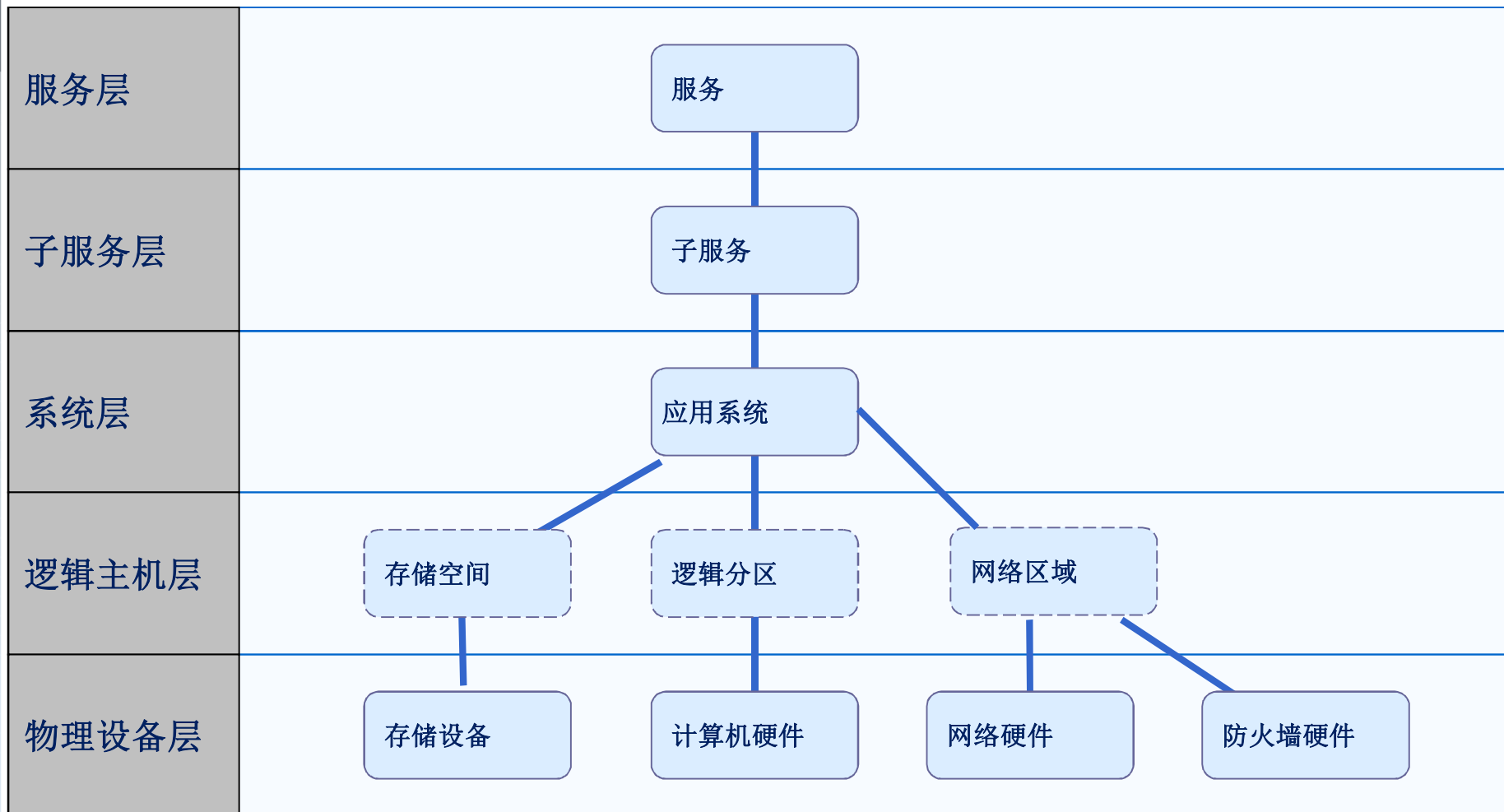
桌面机硬件清单
桌面机软件清单
桌面机操作系统清单
桌面机外设清单
桌面机网络设备清单
桌面机扫描设备清单
桌面机服务器清单
桌面机服务器软件清单
桌面机服务器操作系统清单
桌面机服务器外设清单
桌面机服务器网络设备清单
桌面机服务器扫描设备清单

资产管理员&经理

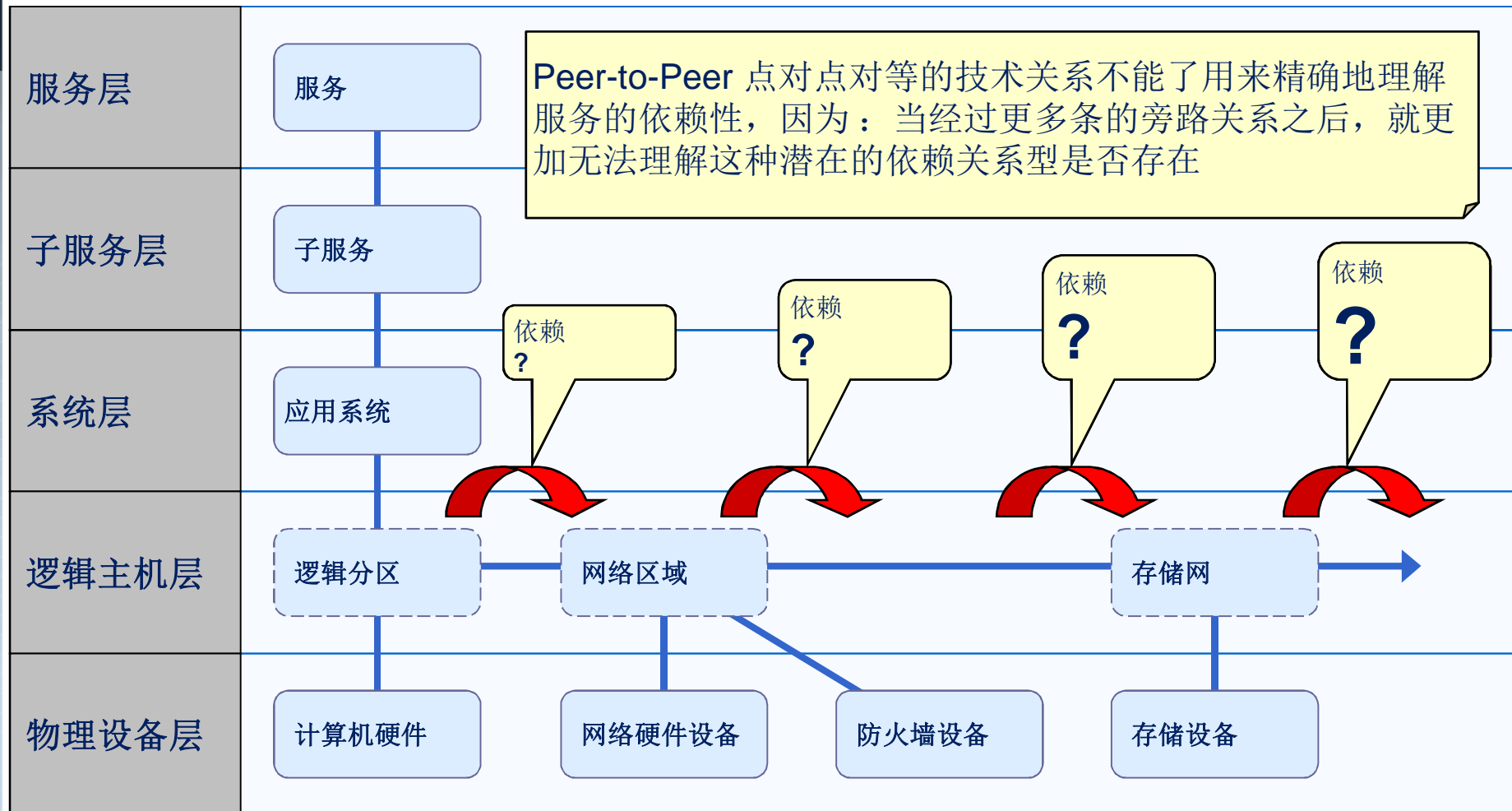
数据中心
环境管理团队和部门



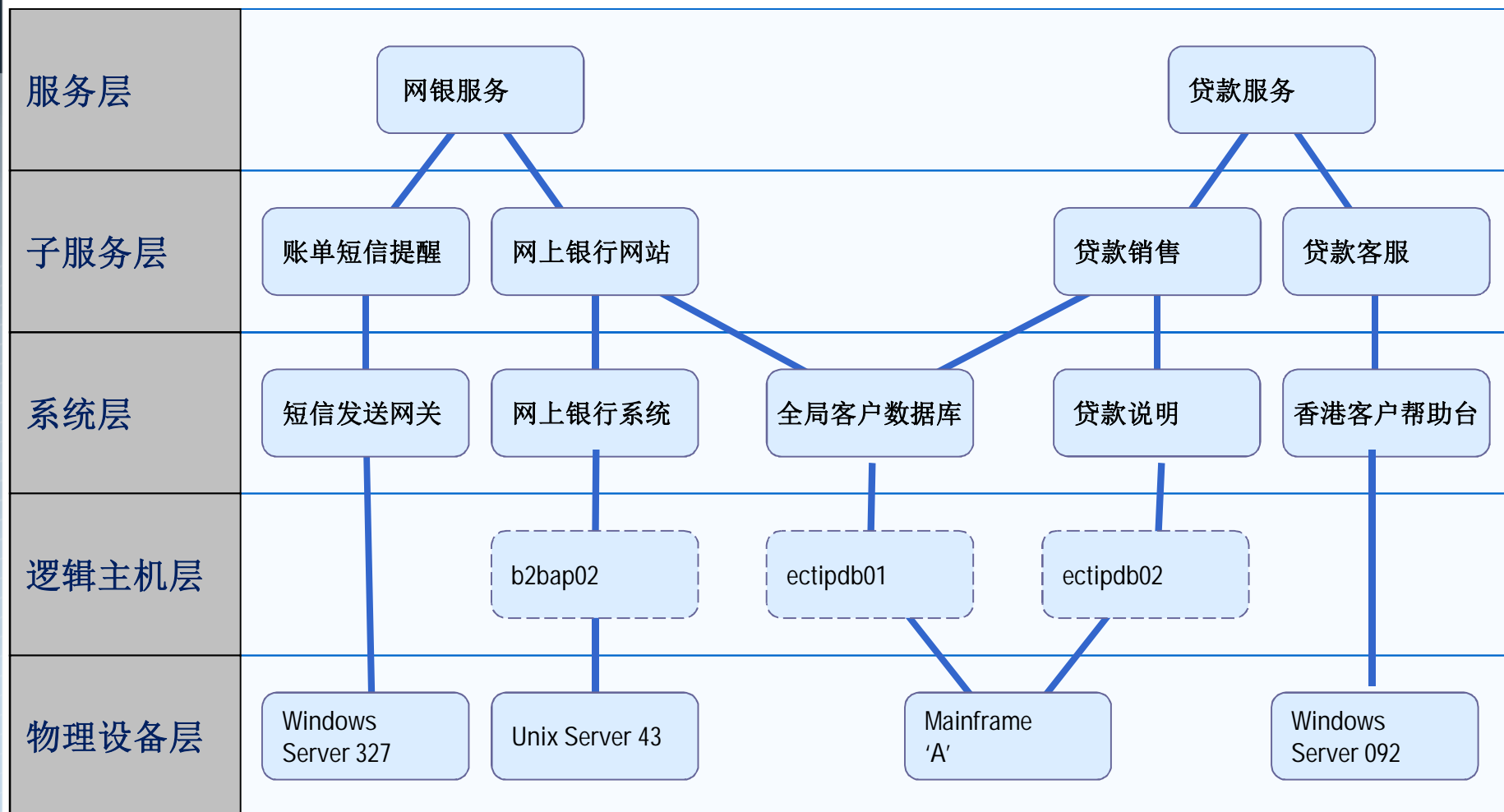
CI类型&结构：增加更多CI类型



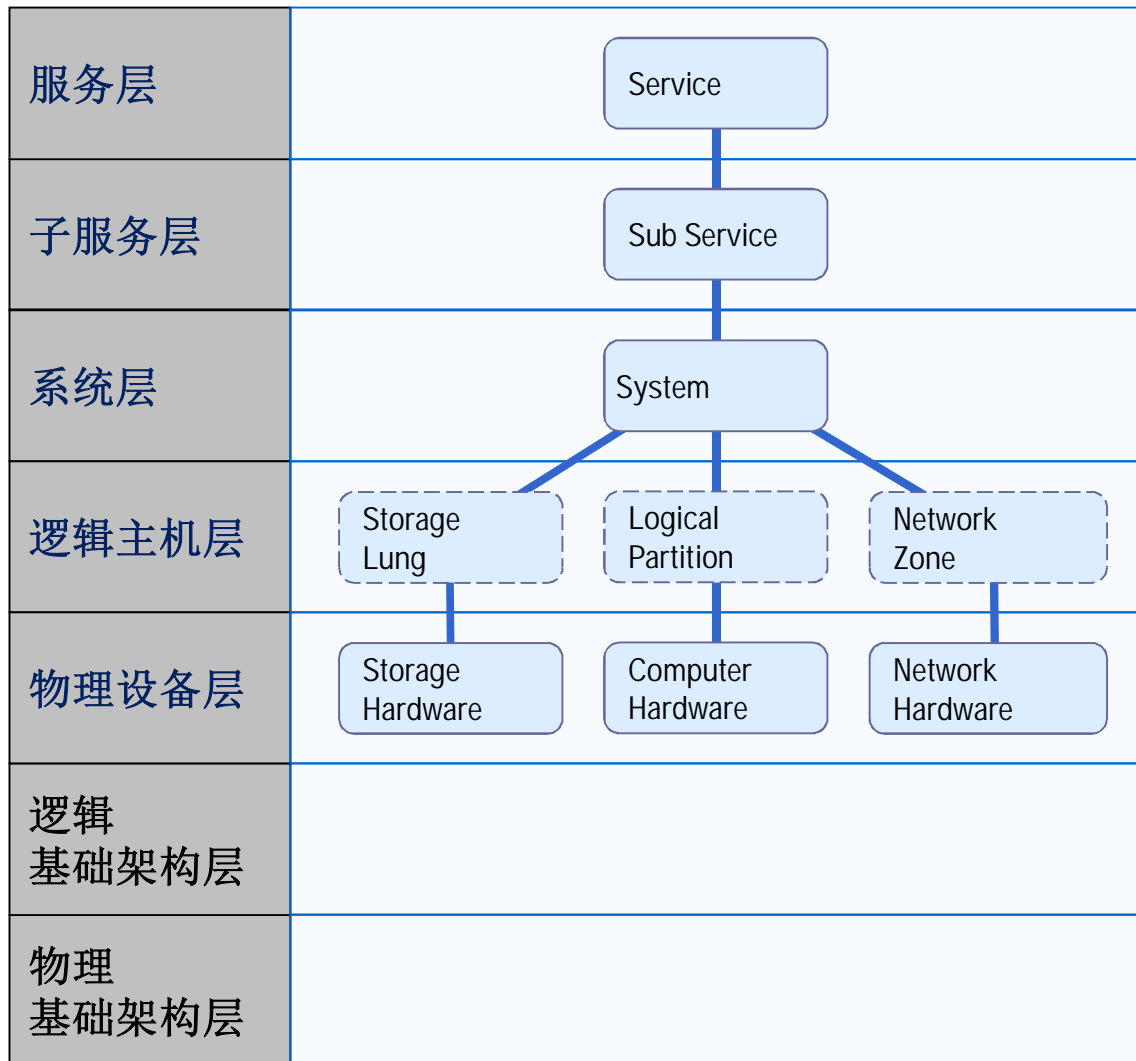
点对点的技术关系 为什么不能被用来理解服务依赖性



步骤9 - 定义IT服务模型蓝图 “应用系统模型实例”



以计算机服务器硬件为中心的模型



优点

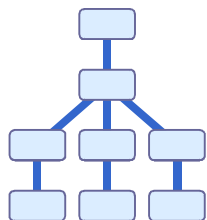
- 有可能和自动化发现工具的数据能配合的更好;
- 不仅能一定程度的反映出系统的依赖关系, 还能提供有限的“计算机硬件为中心”的系统管理视图, 例如一些IT设备对IT设备的关系, 但是都是只能连接到计算机设备上的

缺点

- 结构的复杂性可能导致用户的混淆;
- 由于架构扩展到更大的范围, 反而更难能看到一个总体的视图
- 需要更多的用户培训和指导;
- 计算机硬件和逻辑服务成为向上的一个单点故障

CMDB模型设计和CI类型选择原则：

1) 选择合适的架构类型:



服务管理 = 水平层级模型

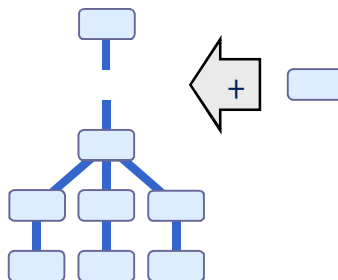


服务管理 =
IT对IT的连接模型

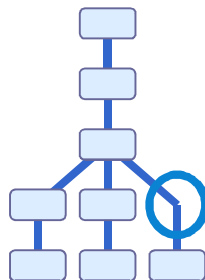


成本管理 =
IT技术到成本和合同关系

2) 在需要的时候添加 / 减少层次 (CI类型):



3) 技术的层面的如果不合理或者无法实现可以先忽略

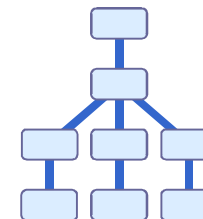


4) 从简单模型开始....



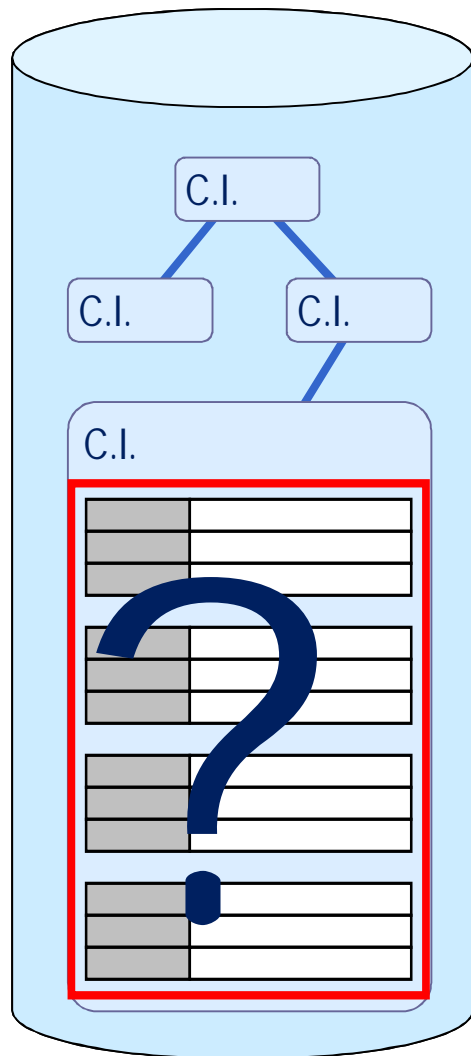
业务系统, 软件 & 服务器

5) 然后发展到复杂的模型...



增加存储, 网络, etc..

属性选择的四C法则 – 帮助来识别CI属性



CORE 核心

这些属性与CI对象本身无关，例如：唯一编码和名称等

CAPABILITY 能力

这些属性用来支持某个管理流程，这些数据能真的支持流程中的流转和决策

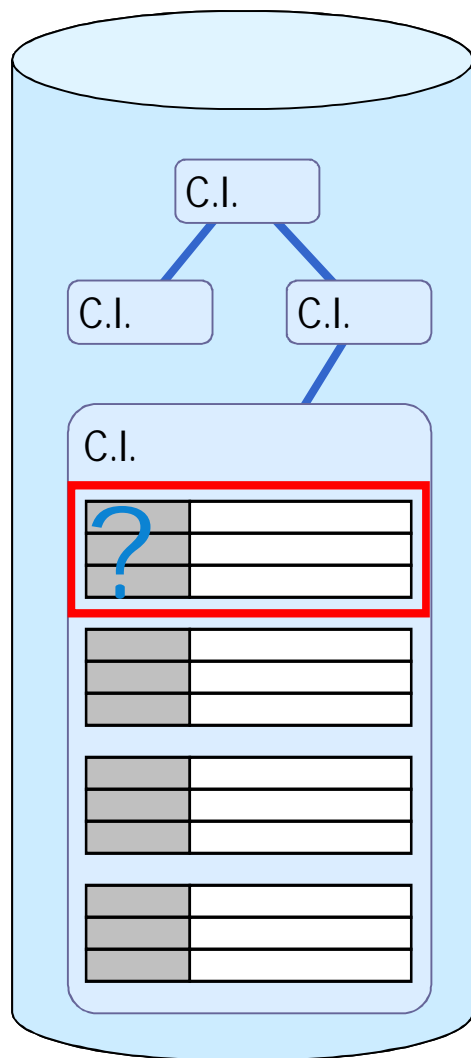
CONTROL 控制

这些属性用来控制CMDB的数据本身，例如修改审计记录等

CONTEXT 详细

这些属性是附加的可选属性，用来帮助用户能更容易的理解和使用CMDB数据，用来实现有限的分析能力

什么是核心属性？



CORE 核心

这些属性与CI对象本身无关，例如：唯一编码和名称等

CAPABILITY 能力

这些属性用来支持某个管理流程，这些数据能真的支持流程中的流转和决策

CONTROL 控制

这些属性用来控制CMDB的数据本身，例如修改审计记录等

CONTEXT 详细

这些属性是附加的可选属性，用来帮助用户能更容易的理解和使用CMDB数据，用来实现有限的分析能力

核心属性选择范围和标准 – 名称、类型和状态

C.I.

Core

Capability

Control

Context

名称:	APPSERVER27
资产编号:	LTSB12345
序列号:	65H4385454
生产IP:	11.101.2.33

名称& 标识

包括所使用的任何统一名称或者标识（逻辑或者物理的）。例如：如果系统管理员远程访问一个服务器时，往往使用服务器的IP地址更多，而不是这个服务器的主机名。

C分类(Level 1):	硬件
T类型 (Level 2):	处理单元
I条目 (Level 3):	服务器
产品名称 (Level 4):	P-SERIES (RS6000)

类型/ 分类

包括一些详细分类属性，这描述了这个配置对象从粗到细的各级分类名称，这种分类通常被定义为3层或者多层

生命周期:	运行中
-------	-----

状态

包括一些状态属性，要来描述CI的生命周期状态，注意这个状态值需要包括各种系统和成本视角的情况

核心属性 – 三层分类设计原则和示例

Type (Level 1):	Type (Level 2):	Type (Level 3):	Type (Level 4):
硬件	处理单元	服务器	P595
硬件	处理单元	服务器	SUN-Fire454
硬件	处理单元	桌面机	
硬件	处理单元	手持电脑	
硬件	FIREWALL		CCB 产品分类参考范例
SOFTWARE	EXECUTABLE	OPERATING SYSTEM	
SOFTWARE	EXECUTABLE	APPLICATION	
SOFTWARE	COMPONENT	LOGICAL PARTITION	P-SERIES
GROUP	SYSTEM	GOLD	
GROUP	SERVICE	CATEGORY A	

设计原则:

从最粗的找到需要找的开始到最细的条目, 这样让用户也能很方便的找到所需要的分类条目

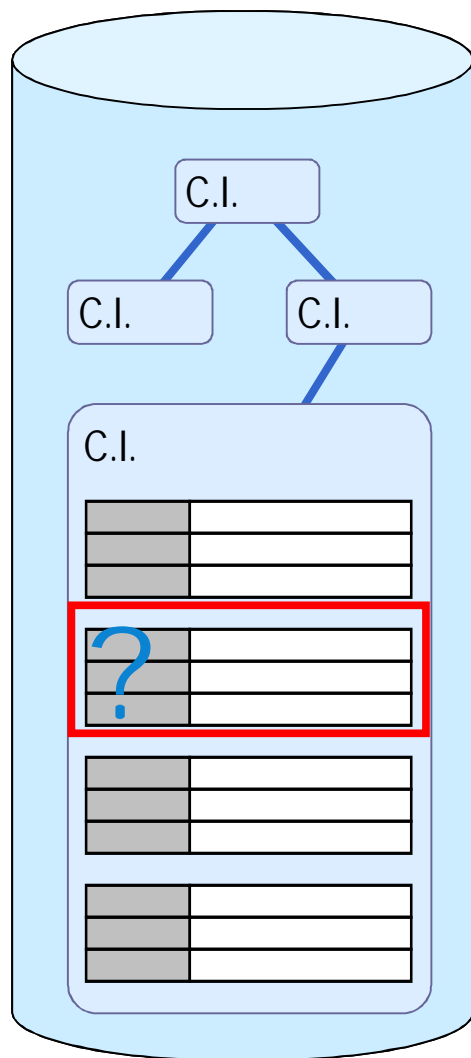
设计原则:

具体的条目可以有这方面负责的人或者有需求的人提议, 然后讨论通过。例如: 不同的条目可能有不同的运维团队关系和负责和提议, 这样让他们能各自找到关系的CI。

设计原则:

尽量避免某个条目重复出现的次数

什么是能力属性？



CORE 核心

这些属性与CI对象本身无关，例如：唯一编码和名称等

CAPABILITY 能力

这些属性用来支持某个管理流程，这些数据能真的支持流程中的流转和决策

CONTROL 控制

这些属性用来控制CMDB的数据本身，例如修改审计记录等

CONTEXT 详细

这些属性是附加的可选属性，用来帮助用户能更容易的理解和使用CMDB数据，用来实现有限的分析能力

流程中所需的能力属性

ITIL
职责定义:

配置
管理

‘为事件、问题、和变更&
发布管理提供唯一配置参
考’

‘为其他服务管理流程提
供精确配置信息’

‘对所有IT资产负责’

ITIL
服务管理流程

Service Support

Service Desk

Change Mgt

Incident Mgt

Problem Mgt

Release Mgt

Configuration Mgt

Service Delivery

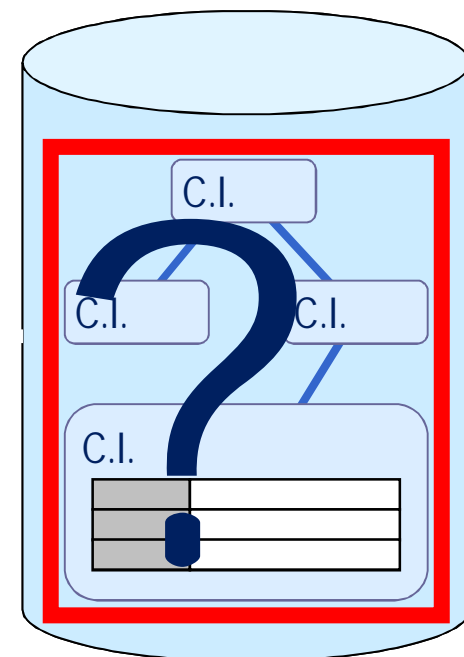
Service Level Mgt

Availability Mgt

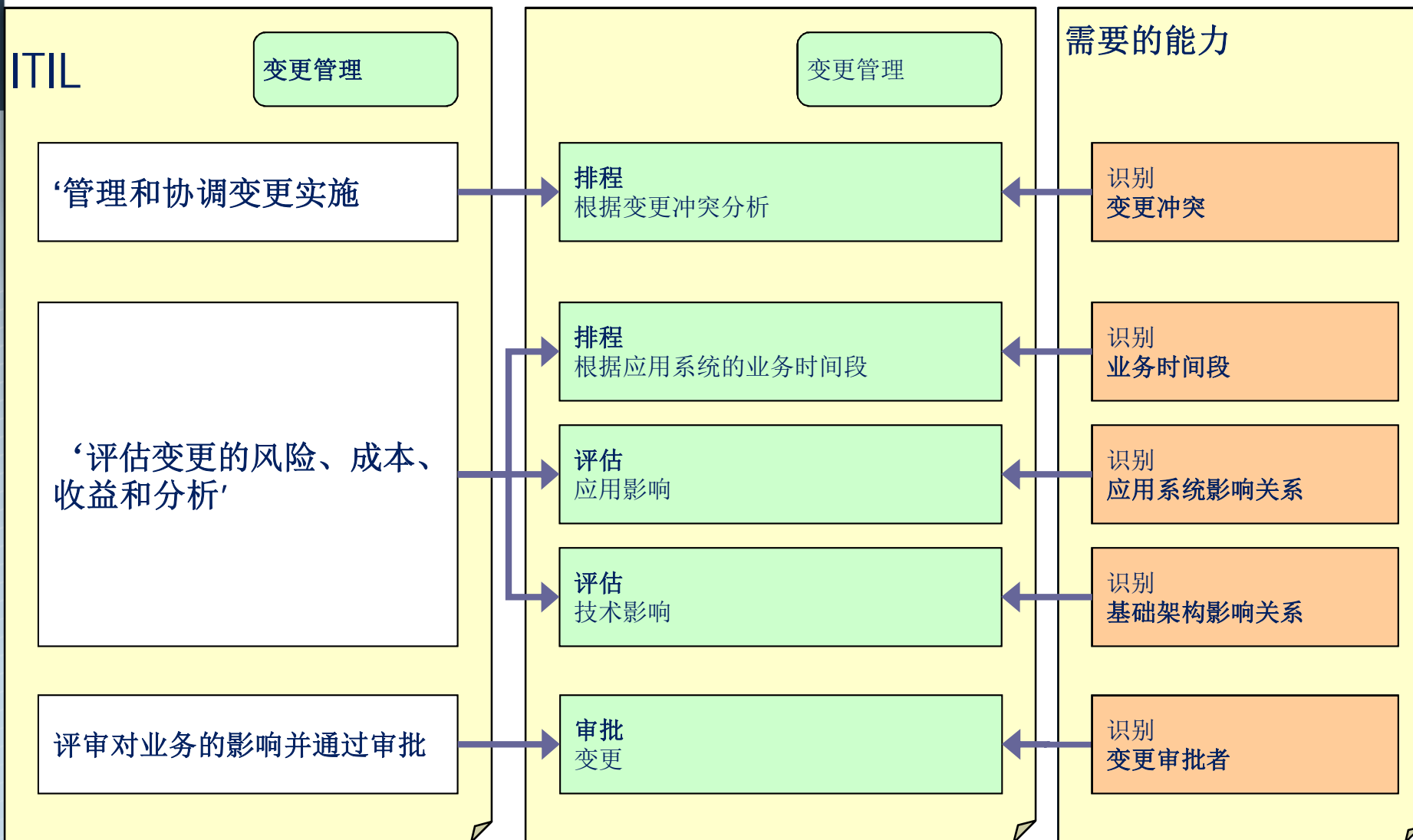
Capacity Mgt

IT Service Continuity Mgt

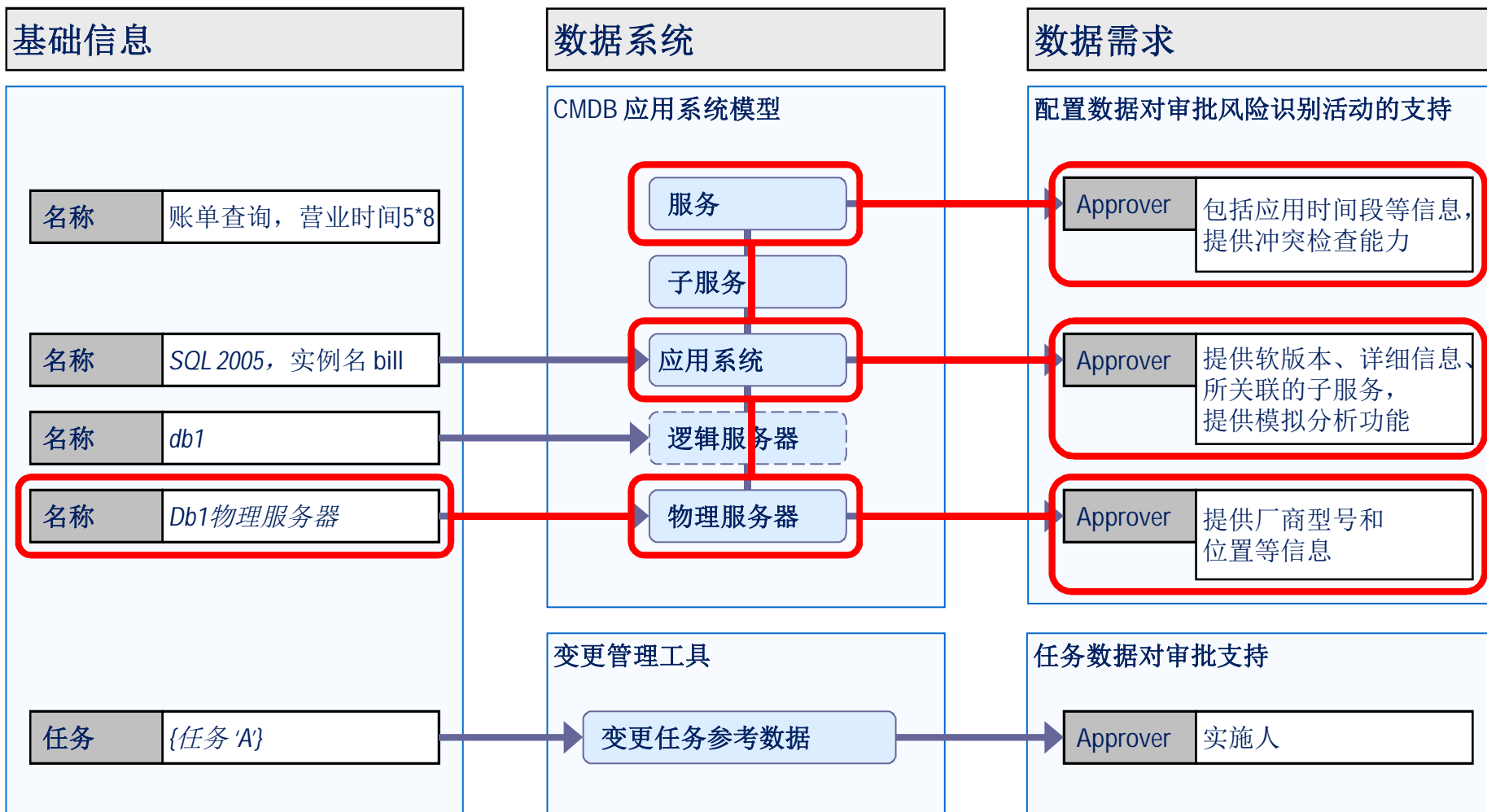
IT Financial Mgt



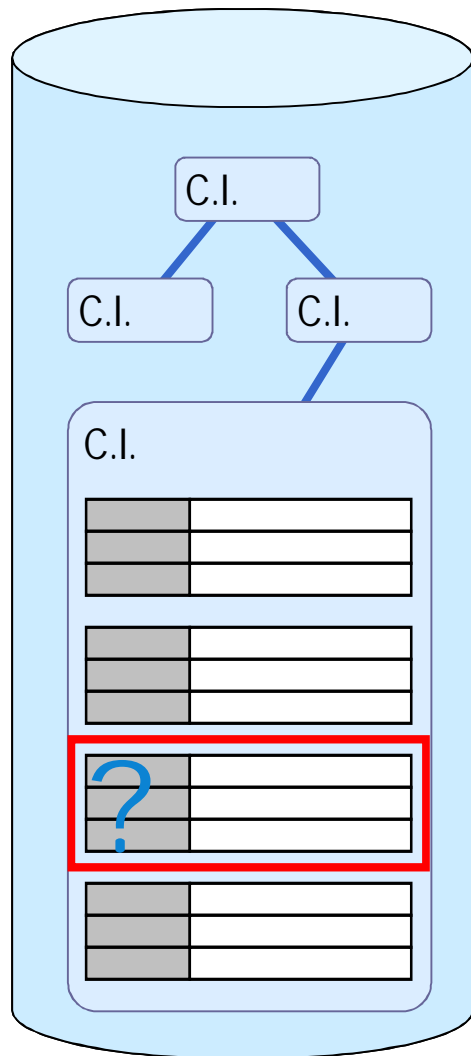
变更的目标 – 识别出的所需要的能力



变更对CMDB的需求



什么是控制属性？



CORE 核心

这些属性与CI对象本身无关，例如：唯一编码和名称等

CAPABILITY 能力

这些属性用来支持某个管理流程，这些数据能真的支持流程中的流转和决策

CONTROL 控制

这些属性用来控制CMDB的数据本身，例如修改审计记录等

CONTEXT 详细

这些属性是附加的可选属性，用来帮助用户能更容易的理解和使用CMDB数据，用来实现有限的分析能力

控制属性 – 记录新增、修改和审计记录

C.I.

Core

Capability

Control

Context

CI创建

用户ID:	创建时间
TIVOLI FEED	23/03/05 03:27

CI修改

用户ID:	Modified Date:	Attribute:	Old Value:	New Value:	
FRED SMITH	24/03/05 10:14	STATUS	UNKNOWN	ACTIVE	▲
FRED SMITH	24/03/05 10:15	TYPE		P-SERIES	■
FRED SMITH	14/06/07 15:34	STATUS	ACTIVE	INACTIVE	■
					▼

CI审计

用户ID:	最好审计日期:	审计状态:
PETE JONES	17/05/07	PENDING

C.I.

Core

Capability

Control

Context

CI创建

User ID:	Creation Date
TIVOLI FEED	23/03/05 03:27

设计原则

确保系统自动产生和维护这两个字段

设计原则

在查询条件中可以有选择的考虑这些属性：例如：

- 用户ID
- 日期范围;
- 审计日期

CI修改

User ID:	Modified Date:	Attribute:	Old Value:	New Value:
FRED SMITH	24/03/05 10:14	STATUS	UN	
FRED SMITH	24/03/05 10:15	TYPE		
FRED SMITH	14/06/07 15:34	STATUS		

设计原则

有选择性的宣传需要审计修改记录的字段，保证审计结果有用，并容易搜索。

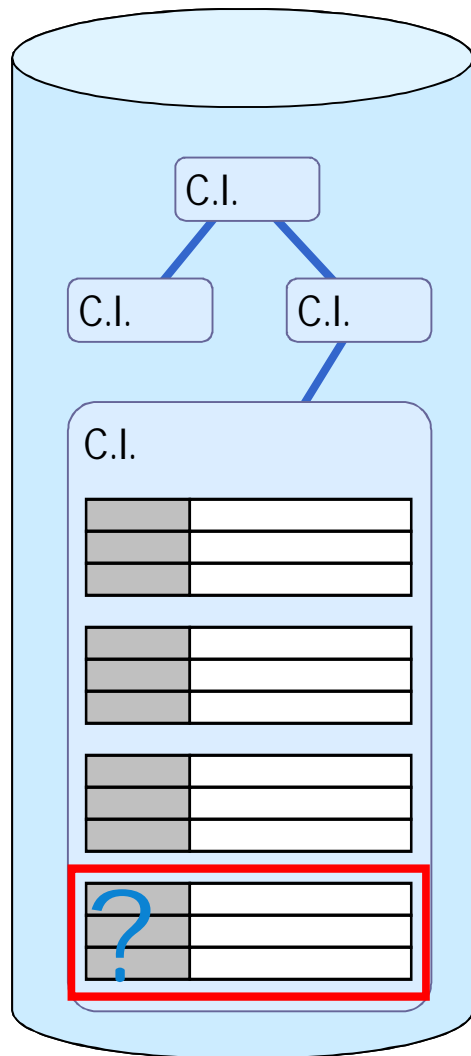
CI审计

User ID:	Audit Date:	Audit Status:
PETE JONES	17/05/07	PENDING

推荐的审计属性:

- 核心属性 – 必须包括这些属性
- 能力属性 – 建议排除一部分;
- 详细属性 – 建议排除一部分

什么是详细属性？



CORE 核心

这些属性与CI对象本身无关，例如：唯一编码和名称等

CAPABILITY 能力

这些属性用来支持某个管理流程，这些数据能真的支持流程中的流转和决策

CONTROL 控制

这些属性用来控制CMDB的数据本身，例如修改审计记录等

CONTEXT 详细

这些属性是附加的可选属性，用来帮助用户能更容易的理解和使用CMDB数据，用来实现有限的分析能力

C.I.

Core

Capability

Control

Context

生产IP:	10.47.28.127
CPU 数量:	1
CPU 速度:	333 Mhz

详细属性

从IP地址（如果IP为测试网IP），CPU数量和CPU速度能看出这个机器更像一个非关键对测试机，而非重要的对生产服务器。

设计原则

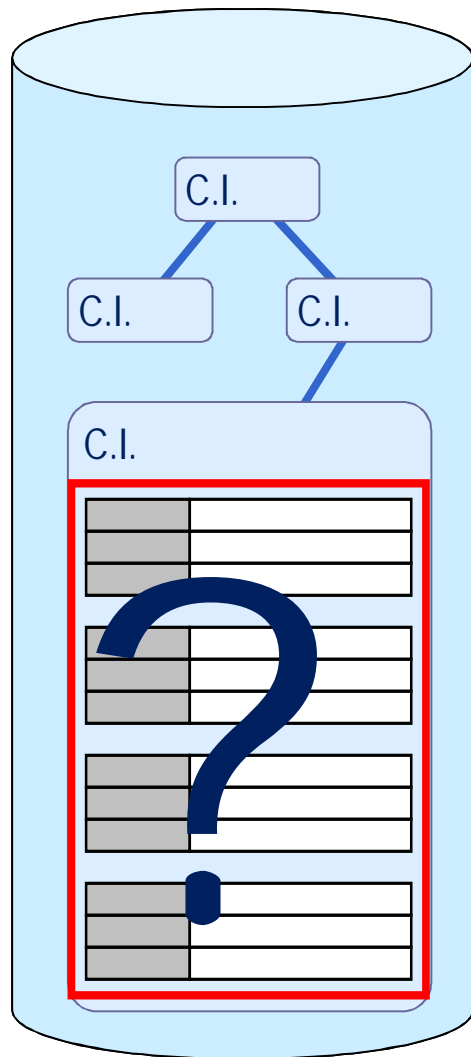
低价值的无用信息不应该存放在**CMDB**中。考虑属性是否的使用者和使用频率。考虑属性的数据源，考虑是否可自动发现，考虑更新维护的工作量。

设计原则 – 定义好管理范围

详细属性很多，所以经常会被完全填充，为了避免这个情况，一定要定义好**CMDB**的范围。

判断是否这个属性被某个管理流程所需要？如果不是，这个属性需要排除在**CMDB**之外，属性的请求者则可能需要把请求提给其他的管理系统，如监控或者自动化等。

属性选择的四C法则 – 帮助来识别CI属性



CORE 核心

这些属性与CI对象本身无关，例如：唯一编码和名称等

CAPABILITY 能力

这些属性用来支持某个管理流程，这些数据能真的支持流程中的流转和决策

CONTROL 控制

这些属性用来控制CMDB的数据本身，例如修改审计记录等

CONTEXT 详细

这些属性是附加的可选属性，用来帮助用户能更容易的理解和使用CMDB数据，用来实现有限的分析能力

项目实施提示：使用命名标准和规范



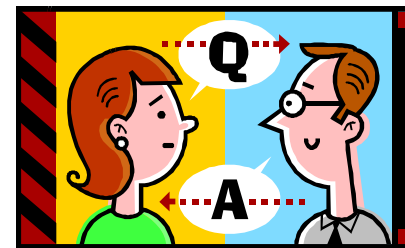
现存的标准和规范



常识



协商一致



参与



企业组织结构

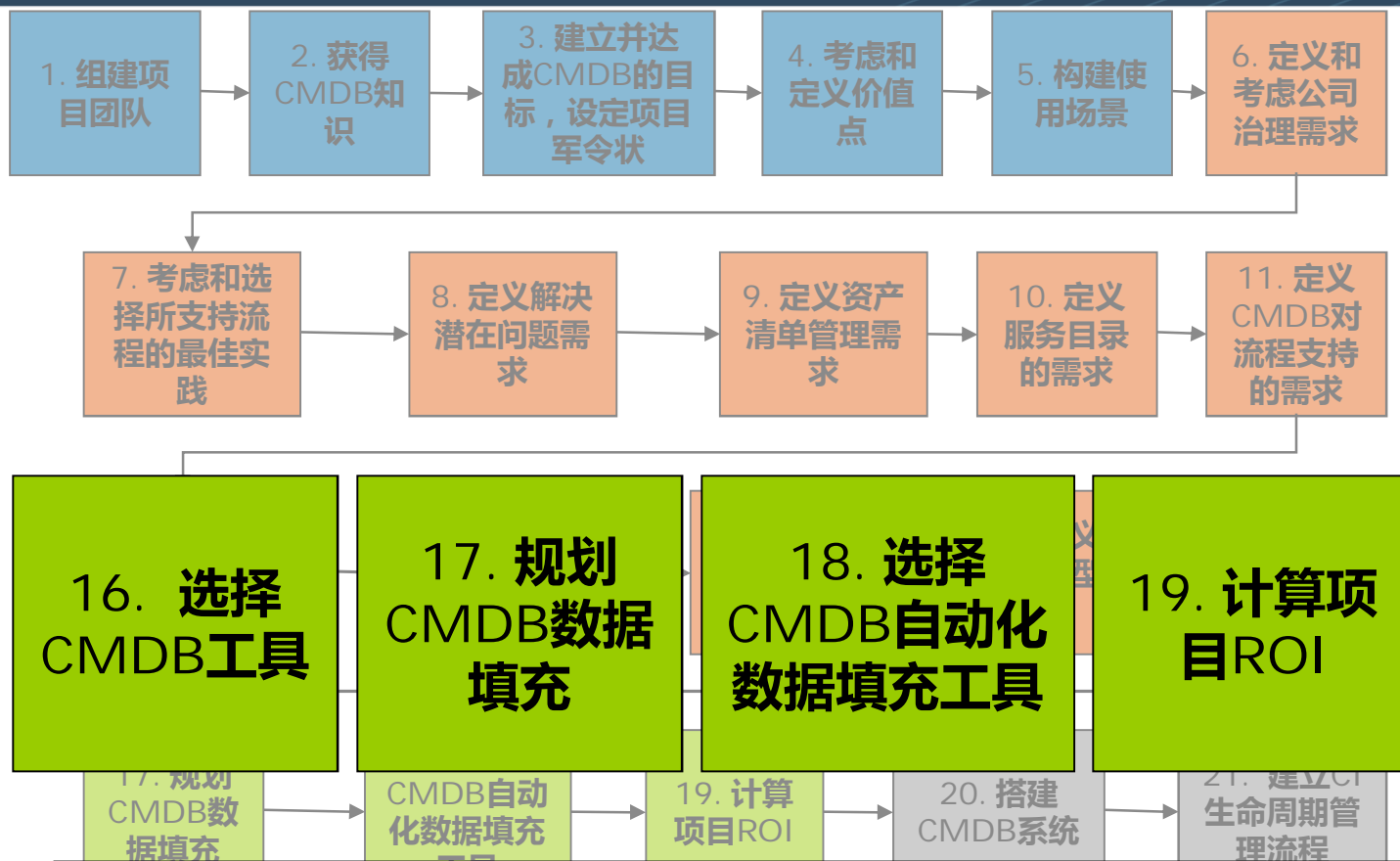


位置

项目实施提示：选择合理的推广方式

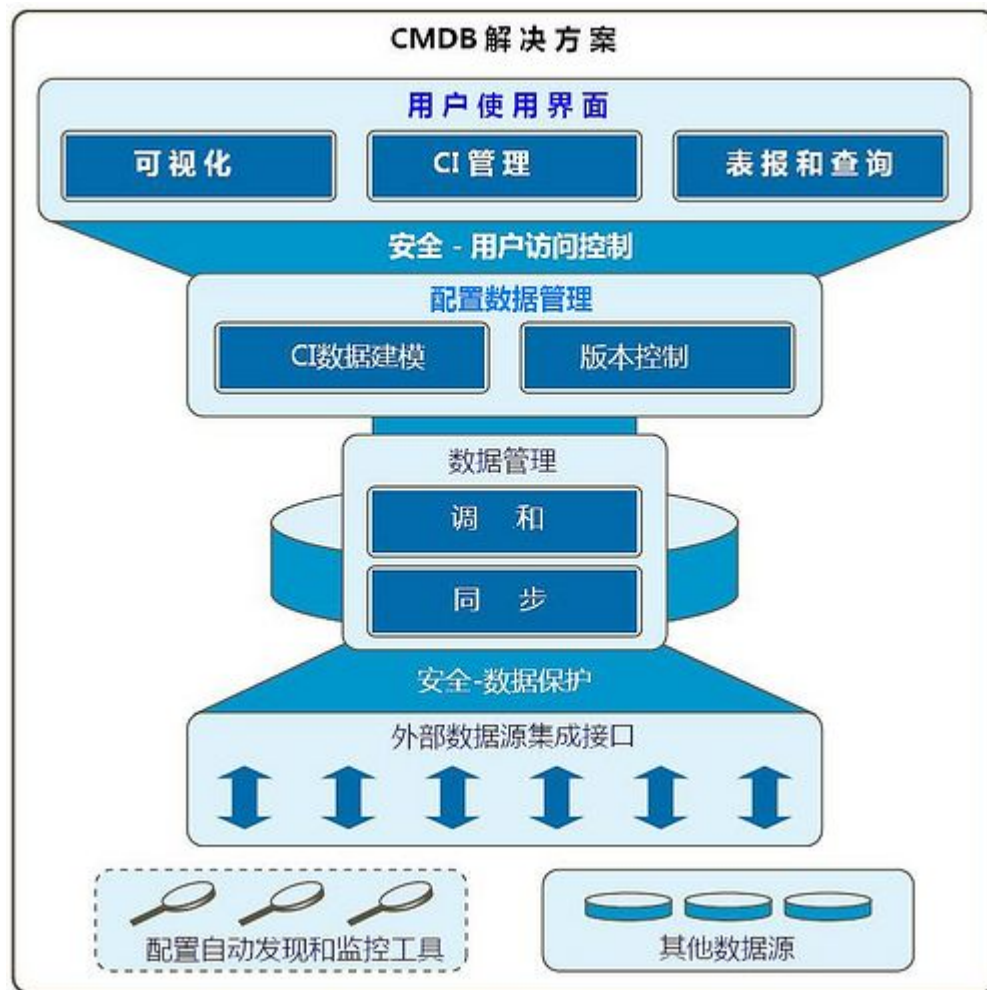


循序渐进构建CMDB--战术指导



里程碑:
CMDB产品工具已经采购完毕, 并且准备好实施CMDB

步骤16 – 选择CMDB工具



标准	权重
功能	20%
健壮性	10%
战略	10%
用户体验	5%
市场形象	10%
定价结构	15%
厂商支持	5%
公司稳定性	15%
成功客户	10%

步骤16 – 规划CMDB数据填充

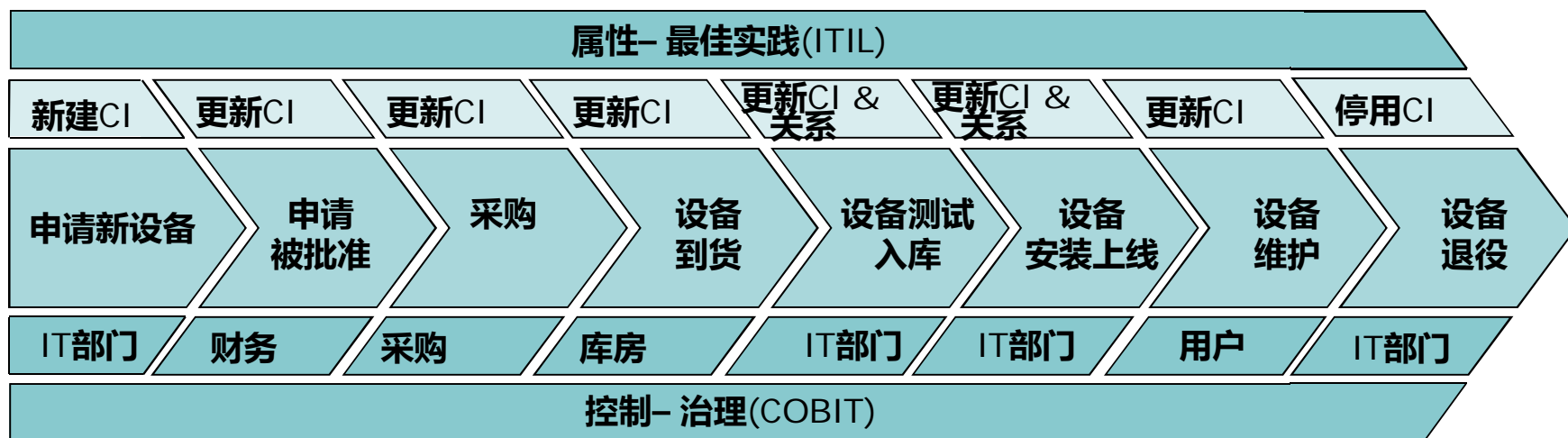
1	Foundation Discovery						Configuration Discovery			
2	Fields	Windows w/ WMI	Solaris (SSH / telnet)	HP-UX (SSH / telnet)	Linux (SSH / telnet)	AIX (SSH / telnet)	Windows w/ WMI	Windows w/o WMI	Solaris	HP-UX
3	Category: Machine									
4	- Name	X	X	X	X	X	X	X	X	X
5	- Manufacturer	X	X	X	X	X	X	X	X	X
6	Dedicated (type of device)	X	X	X	X	X				
7	- Description	X	X	X	X	X	X	X	X	X
8	- Mac / Machine ID						X	X	X	X
9	- Model	X	X	X	X	X	X	X	X	X
10	- Domain	X	X	X	X	X	X	X	X	X
11	- Serial number	X	X	X	X	X	X		X	X
12	- Scan time	X	X	X	X	X	X	X	X	X
13	- Number of processors						X		X	X
14	- Type						X	X		
15	- Insert time						X	X	X	X
16	- WMI version						X			
17	- Aliases	X	X	X	X	X	X		X	
18	- Boot time						X	X	X	X
19	- Memory	In Memory category	In Memory category	In Memory category	In Memory category	In Memory category			X	X
20	DHCP Use	X	X	X	X	X				
21										
22	Category: OS									
23	- Manufacturer	X	X	X	X	X	X	X	X	X
24	- Product	X					X	X	X	X
25	- Version	X	X	X	X	X	X	X	X	X

Figure 17.2. Example of a CI mapping blueprint between CIs and data sources

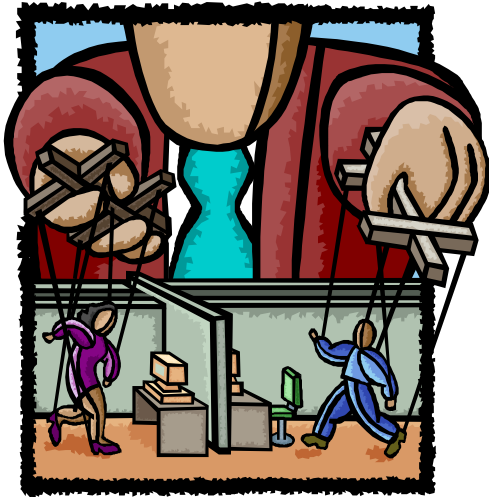
循序渐进构建CMDB--战术指导



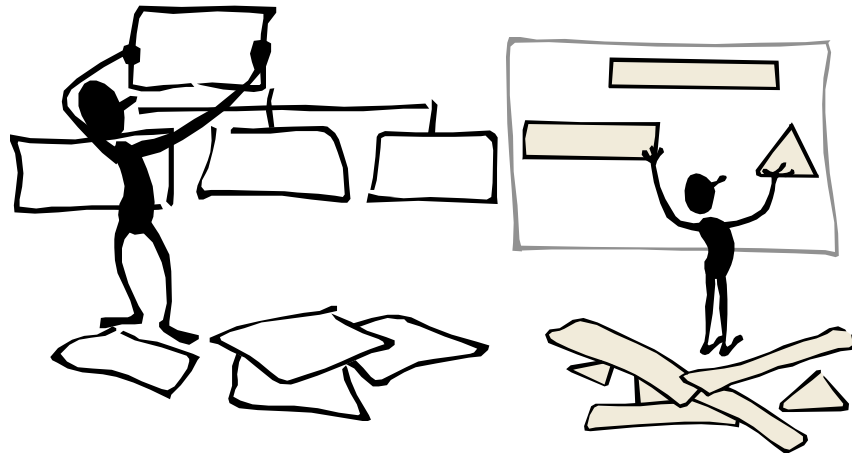
步骤16 –建立CI生命周期管理流程



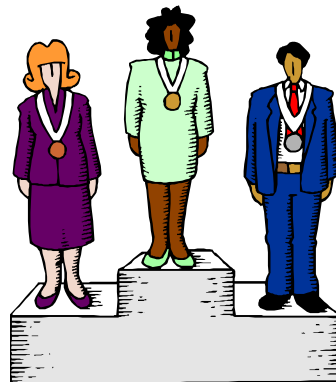
项目实施提示：准备企业文化的变革



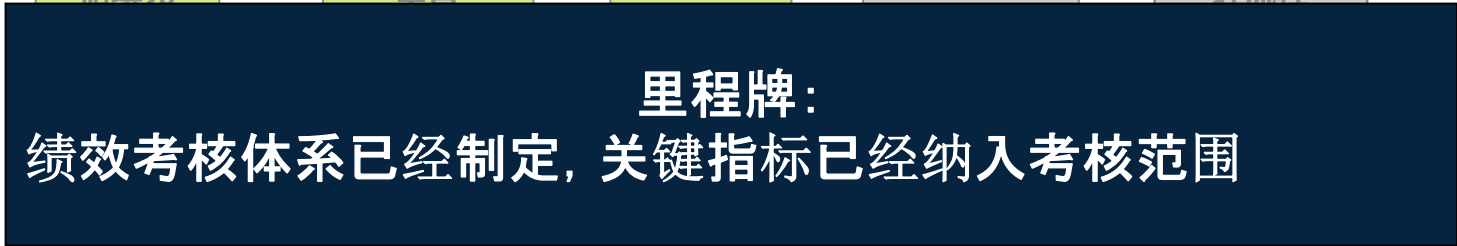
理念



期望- 参与 - 透明度



鼓励措施



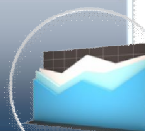
- 配置项数据和物理配置项之间存在不一致的数量，以配置项总数的百分比来表达

精确性



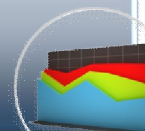
- 配置管理系统与外部集成和联邦的数据源的数量，除以IT所使用到所有数据源的总数

完整性



- 配置管理系统中，与至少一个IT服务有关系的配置项的总数除以总配置项数的百分比

有效性



- 在执行资产审计方面的工作量和执行效率

效率



- 带CI更新的变更单数量
- 带CI更新的事件单数量
- 配置项负责人属性有值的

其他



步骤26 – 数据质量控制：参考目标

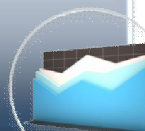
- CMDB上线9个月内 目标为10%
- CMDB上线12个月内 目标为5%

精确性



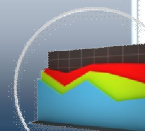
- 6各月以内达到80%，12个月内增加到90%

完整性



- 在12个月内相对基线有30%的提高（前置条件SLM）

有效性



- 在12个月内对照基线有30%的提高

效率



- 24个月内达到100% (ALL)

其他



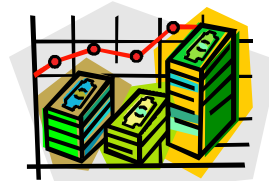
项目实施提示：取得管理层认可



符合项目规划



清晰的交付结果



以结果导向，聚焦成果



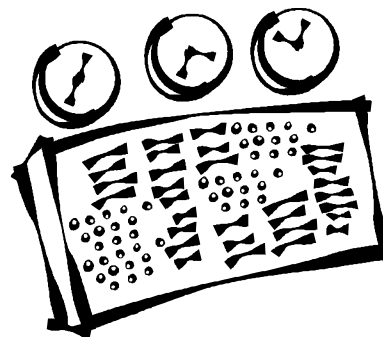
项目进度沟通



保证项目计划日期



嘉奖&庆祝
(quick) wins



可考核、可控制的流程
通过应用绩效考核指标

