1. - 6种stakeholders

* 从他们的视角，关注的系统的重点是什么
* 在系统开发中扮演的角色
* 系统分析师的职责、需要的技能

1. - 3个building block : DATA PROCESS INTERFACE

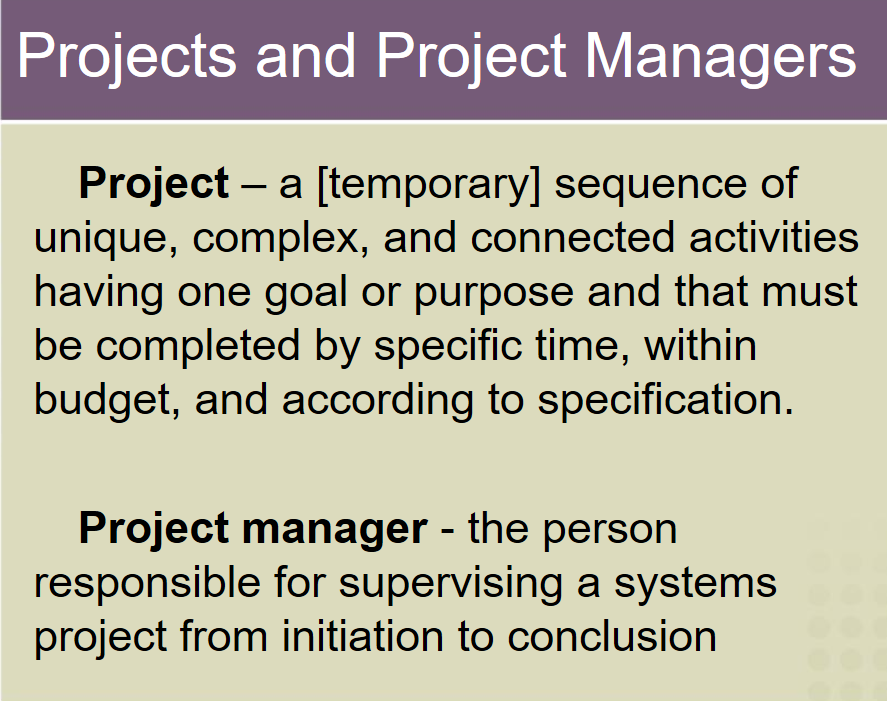
* data information的区别
* front – and back-office的区别（前台系统和后台系统）
* 5类不同系统，稍微了解每一类系统的特点
* 从6个stakeholder的视角，他们关注这3个buinding block的什么地方

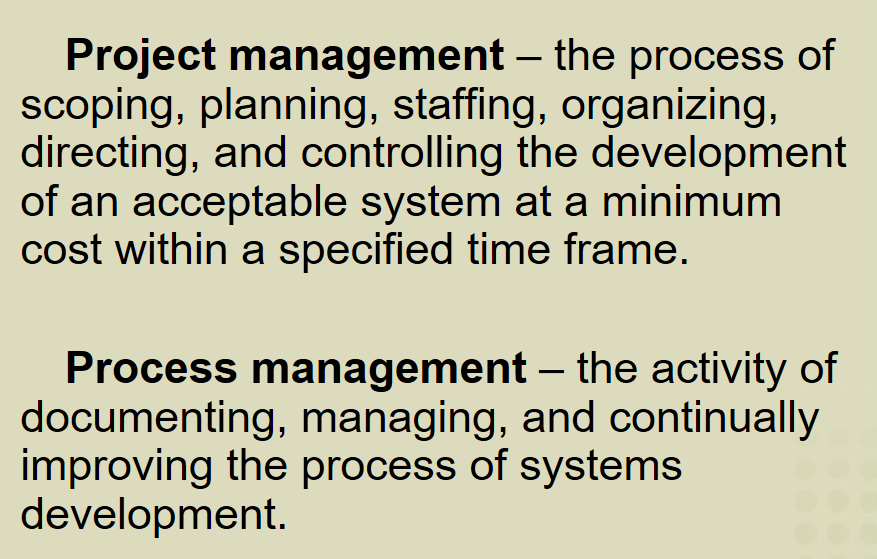
1. 介绍系统开发的基本原则

* CMM
* 系统开发的生存周期
* 系统开发的8条基本原则（很重要）
* (有两个框架：FAST PIECES ，本书从5到15章内容的展开就是围绕FAST框架)
* FAST框架中划分的阶段、每个阶段包含的活动
* 了解PIECES框架这6个单词的内涵
* 系统开发的不同的路线、方法，一些模型。典型的比如生存周期模型
* 了解CASE、ADE是什么

1. 项目管理（计算题就出在项目管理，根据题目画出PERT图然后找出关键路径，根据题目画Gantt图）

* 项目开发失败的原因
* 分清楚什么是project management，什么是process management
* 项目管理的8个功能（在项目管理中要做的8件事情）





* **必须会画PERT图 Gantt图，必须知道这两个图的作用。**Gantt图可以表达出每一个任务的持续时间，并且可以非常清晰的表达出任务之间的重叠关系。PERT图是一个基于网络的图，它能够表达的是不同任务之间的依赖关系，所以在PERT图上要做关键路径分析。**必须要知道什么是关键路径，怎么做关键路径分析。**

1. 刻画了完整的系统分析阶段的一个全过程，是复习系统分析阶段的guildline。可以不看第5章，直接看6 7 8 9 章。

系统分析阶段最重要的事情就是确定我们要干什么，用文字或模型描述。

* 按照FAST，系统分析分为四个阶段：初始调查 问题分析 需求分析 决策分析。（需求分析是重点）决策分析中牵扯到可行性分析

1. 系统分析阶段要做的非常非常重要的事情——需求分析

* 需求分为两大类：功能需求、非功能需求，需要知道什么是功能需求、什么是非功能需求
* 那些模型都是刻画功能需求的。非功能性需求用PIECES框架描述。
* Ishikawa(fishbone)(鱼骨图)，用来做问题分析，了解基本概念，不用会画。
* 7种非常重要的发现需求的方法，不用死记硬背，要去理解，能用自己的话说出来意思。
* use case，这是用来描述功能需求的一种方式。可以使用UML用例图把文本描述转换为图的方式。用例本身是描述功能需求的文本形式。第六章和面向对象里都讲到了用例，可以结合起来复习。**用例、用例图一定是考核重点。**

1. 数据建模和分析 描述系统对数据有哪些要求 正式给出了model的概念 在本书里介绍了三种方法：结构化的 信息工程的 面向对象的，这三种方法都是模型驱动的，也就是说他们用来表达需求、设计的，最重要的工具就是model。

* Model分两种：logical physical。系统分析阶段使用的所有模型都是logical model，在设计阶段使用physical model。Logical model表达的是有什么(what)，physical model 表达的是怎么做(how)。
* 描述数据需求的模型——ER图，表达了系统中有哪些数据，以及数据的关联关系。和数据库学的是一样的。需要知道绘制完整ER图的步骤——先绘制上下文的ER图、再绘制key-based的ER图，然后再绘制全属性的ER图，然后再做三范式的验证。注意：三范式验证一定要做。**ER图是必考的。**

1. 用结构化的方法绘制的模型——**数据流图（DFD）一定要会画。根据需求背景的描述，绘制出DFD是必须要掌握的，structured english（结构化英语）不考。**DFD不能代表结构化分析的所有结果，DFD+数据字典才可以。

* 数据流图当中的四个重要的组成元素，每个元素的含义。绘制DFD的步骤：先context、再做use case或event列表、再做功能分解、再针对每个event绘制它的DFD、再把所有event的DFD合在一起形成sysytem diagram。最后一步primitive data flow是可选的。
* 注意：DFD不能代表所有的功能需求。DFD只能看到有多少process以及不同process之间数据的流转情况。但是每个process内部的内容、数据流上流动的内容是什么，DFD 表达不出来，这就需要和DFD相配套的数据字典。数据字典包括：procedural logic（加工逻辑，也就是每一个process中，把输入数据流转变为输出数据流需要遵循的步骤和要求。)和data structure，DFD上只能看见数据流的名字，名字里面包含的内容在DFD看不到，需要用文档单独描述它。
* process是要加工数据，数据的需求在ER图，所以ER图和DFD之间是有关系的，通过CRUD矩阵表示，在这个矩阵上说明DFD上的每一个process对ER图上的每一个实体有什么操作权限。

1. 可行性分析 进行决策分析时的重要手段

* 技术 操作 进度 经济可行性 需要知道每一种可行性衡量什么，经济可行性是需要计算的，**但是这次不考这个计算**。**但是这四种可行性的基本概念、衡量标准是必须要掌握的**。
* cost-benefit（成本效益），**不考算数**，但是怎么算是需要掌握的，投资回报率、净现值、货币现值需要知道的，还有怎么做成本效益分析、哪些是成本、成本有哪几种、哪些是benefit等等需要知道。
* 系统分析结束之后，需要做一个sysytem proposal reports（系统建议书），

1. 进入系统设计阶段 讲了两个系统设计路线

* 了解这两种路线的差异

1. 应用体系结构 对应第8章 把第8章logical的DFD转换成physical的DFD

* 系统架构：集中的（不用管这个）、分布式的
* 将系统划分成5层，这5层在分布的时候，有5种方式，就形成了不同的体系结构的模式，比较经典的是c/s(2层、3层) 、b/s。需要搞清楚这5层是哪5层——数据、数据操作、应用逻辑、显示逻辑、显示层，每一层是干什么的，在分布的时候，这些层怎么分就形成了不同的分布式系统的架构，怎么把一个logical的DFD转换成physical的DFD。

1. 对应第7章 和数据库讲的一样 需要注意参照完整性
2. 13 14 15这三章讲的都是interface，**可能在选择题里面考概念。**

分成三部分讲——输出、输入、GUI

* 输出种类：internal external turnaround 要知道这些是什么
* 输出设计时的基本原则

1. 知道不同控件能用来解决什么输入就行
2. 了解UI设计的基本原则

A．OO ： 考概念 比如什么是类、对象、方法、继承、多态等等

考建模 UML 两个图：**use case图 、类图 必考。**

**可能也考状态图、活动图，这些图的基础概念得知道，比如干什么事要选什么模型、主要构成元素是什么、每个元素的内涵是什么、这个图是对系统的哪些内容建模的。**

B． OO 给对象指定责任，将分析类图转变为设计类图。

划分三类：entity object interface object control object 得知道他们是什么，用来刻画什么样的对象的职责。也需要UML模型。

**总结：**

**选择题可能会考在什么时候选什么模型。部署图、构件图、包图不考。用例图、类图必考**

1. 用例描述表格

