
Software Effort Estimation:

软件工作量估算

‘Fake Science’ 伪科学

or

‘Best Practices’最佳实践

Alain Abran

2019 International summit on Development & Management
Strategy & tactics forum on global software development and management
Shanghai (China) July 14-16, 2019

演讲者背景介绍 - Alain Abran

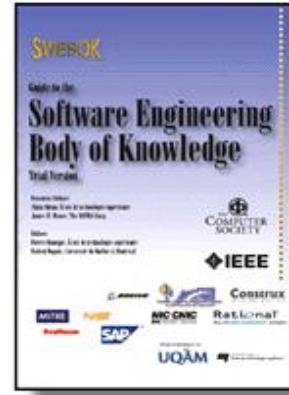
20 years



- Development 开发
- Maintenance 运维
- Process Improvement 过程改进

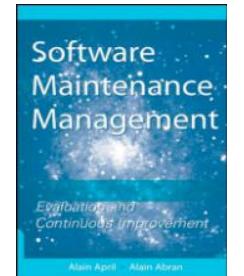
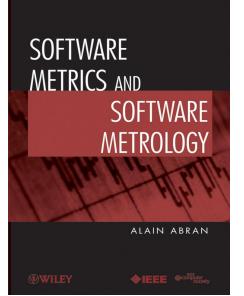
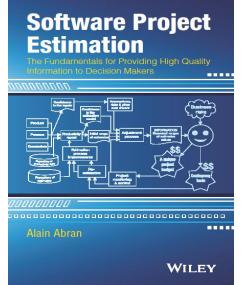


+ 20 years



45 PhD

ISO: 19761,
9126, 25000,
15939, 14143,
19759



Context 背景介绍

For all types and all sizes of software projects, for the very small to the very large organizations:

- development effort must be estimated:
 - ✓ it is a universal problem and it will not disappear!

不论企业的规模如何，不论软件项目的类型和规模如何：
都必须对开发工作量进行估算：这是一个众所周知的难题！

Context 背景

For all types and all sizes of software projects, for the very small to the very large organizations, development effort must be estimated: it is a universal problem and it will not disappear! 不论企业的规模如何，不论软件项目的类型和规模如何：都必须对开发工作量进行估算：这是一个众所周知的难题！

Talk objectives: 演讲主题

1. Which software estimation techniques can you trust? 可以信任哪些软件估算技术？
2. Which ones can provide you with a long-term competitive advantage? 哪些技术可以为你提供长期的竞争优势？

Context: For all types and all sizes of software projects, for the very small to the very large organizations, development effort must be estimated: it is a universal problem and it will not disappear! 不论企业的规模如何，不论软件项目的类型和规模如何：都必须对开发工作量进行估算：这是一个众所周知的难题！

Talk objective 演讲主题：

Which software estimation techniques can you trust and which ones can provide you with a long-term competitive advantage? 可以信任哪些软件估算技术？哪些技术可以为你提供长期的竞争优势？

Talk approach: 演讲内容

Look at some of the widely used estimation techniques to provide you the insights needed to figure out:

- ❖ which ones are merely ‘placebos’ based on ‘fake science’,
- ✓ which ones are based on best professional practices in this field.

介绍一些广泛使用的估算技术，并为你提供一些鉴别方法，
以识别哪些技术只是基于“伪科学”的“安慰剂”，而哪些技术才是基于领域内的最佳专业

List of topics 主题

1. How to recognize Estimating Placebos & Fake Science? 如何识别安慰剂和伪科学?
 2. How to build a competitive advantage? 如何建立竞争力?
 3. My recommendations 我的建议
-

List of topics

- 1. How to recognize Estimating Placebos & Fake Science?**
 2. How to build a competitive advantage?
 3. My recommendations
-

Estimation = a Management Responsibility

估算也是管理层的职责

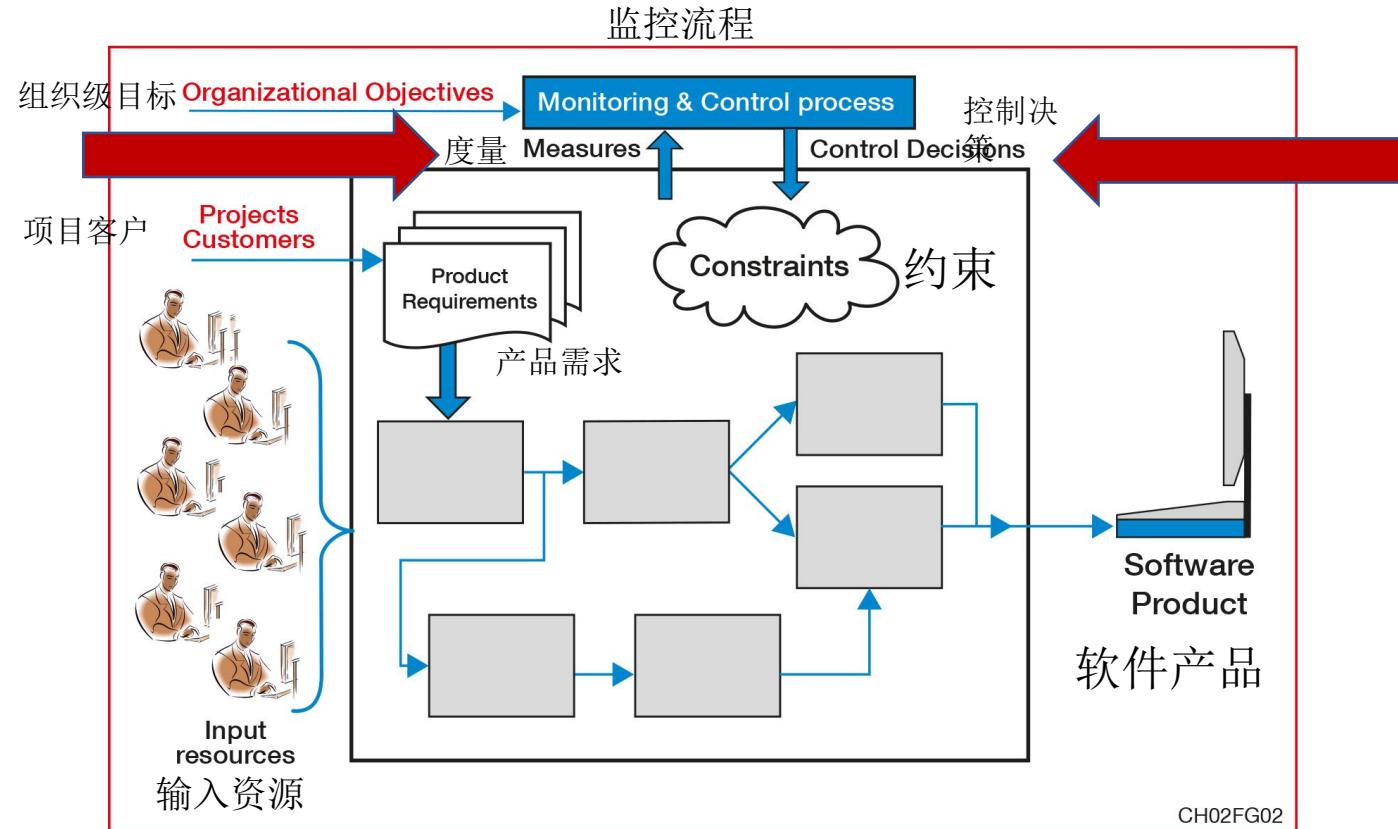


Figure 2.2 A production Process -
the engineering & management perspective.

Placebos 安慰剂

Wikipedia Definitions: 维基百科中的定义:



1. “Placebo” in Latin = “I will please” 拉丁语的含义=我会高兴
2. “Any sham medication or procedure ... void of any known therapeutic value”.
American Society of Pain Management Nursing
“任何虚假的药物或程序.....没有任何已知的治疗价值”。美国疼痛管理护理学会

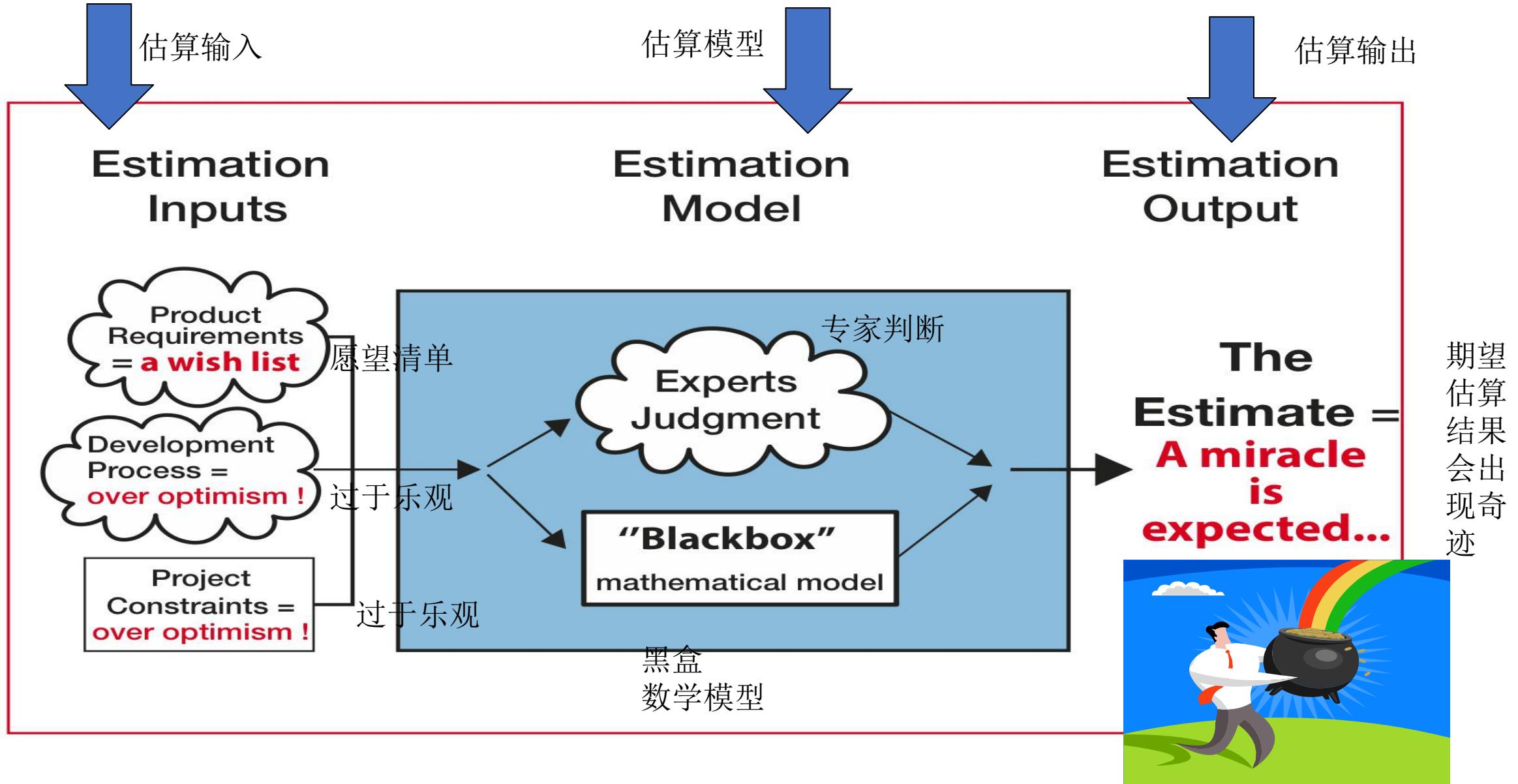


Figure 1.2 Some poor estimation practices observed in industry.

业内一些比较糟糕的估算实践

So-called ‘Expert’ judgment 专家判断

1. Informal, not documented 不正式, 无文档
2. Derived from past experience (subjective recollection of past projects) 根据以往经验 (以往项目的主观经验)
3. Lack of precise quantitative information of independent variables 缺少关于自变量的精确的度量数据
 - Product size 产品规模
 - Cost drivers 成本因子

Can you then expect accurate estimates 关于因变量的精确数据是否可以得到合理的结果?

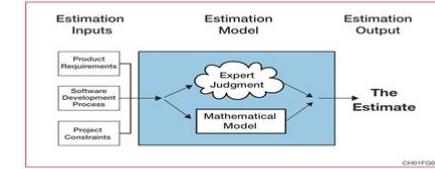


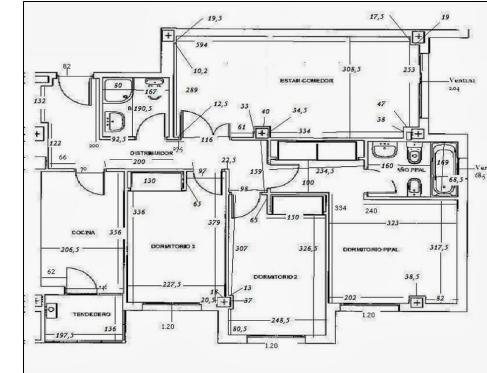
Figure 1.1 Common view of an estimation process.



<http://www.rsc.org/chemistryworld/Issues/2007/March/RidingRAERollercoaster.asp>

Engineering 工程学&数学

- ✓ Detailed observation of past projects
对于过往项目的细致观察
 - ✓ Quantitative data collection
量化数据的收集
 - ✓ Analysis of the impact of individual variables (one at a time)
单一变量影响的分析
 - ✓ Selection of relevant samples
样本点的选择
 - ✓ Statistical analyses
统计学分析
 - ✓ Very careful extrapolation to similar or other contexts
仔细推断（模型可用的）
其他相似的场景

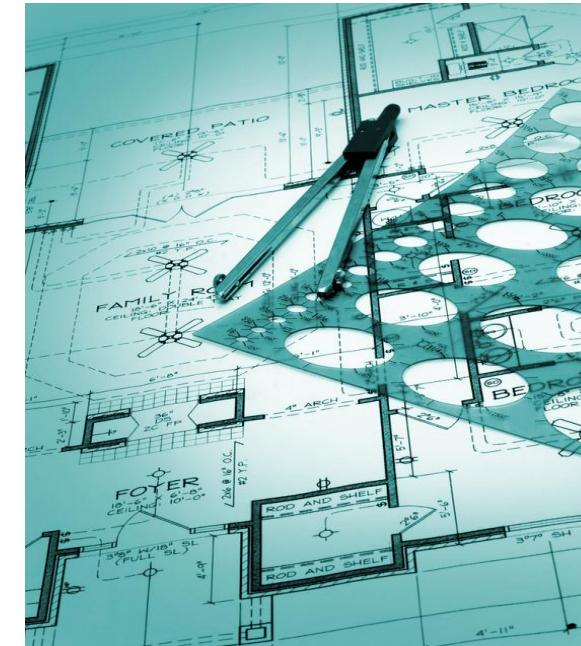


Which one is your estimation process?

你们公司的估算流程是哪一种？



Or?



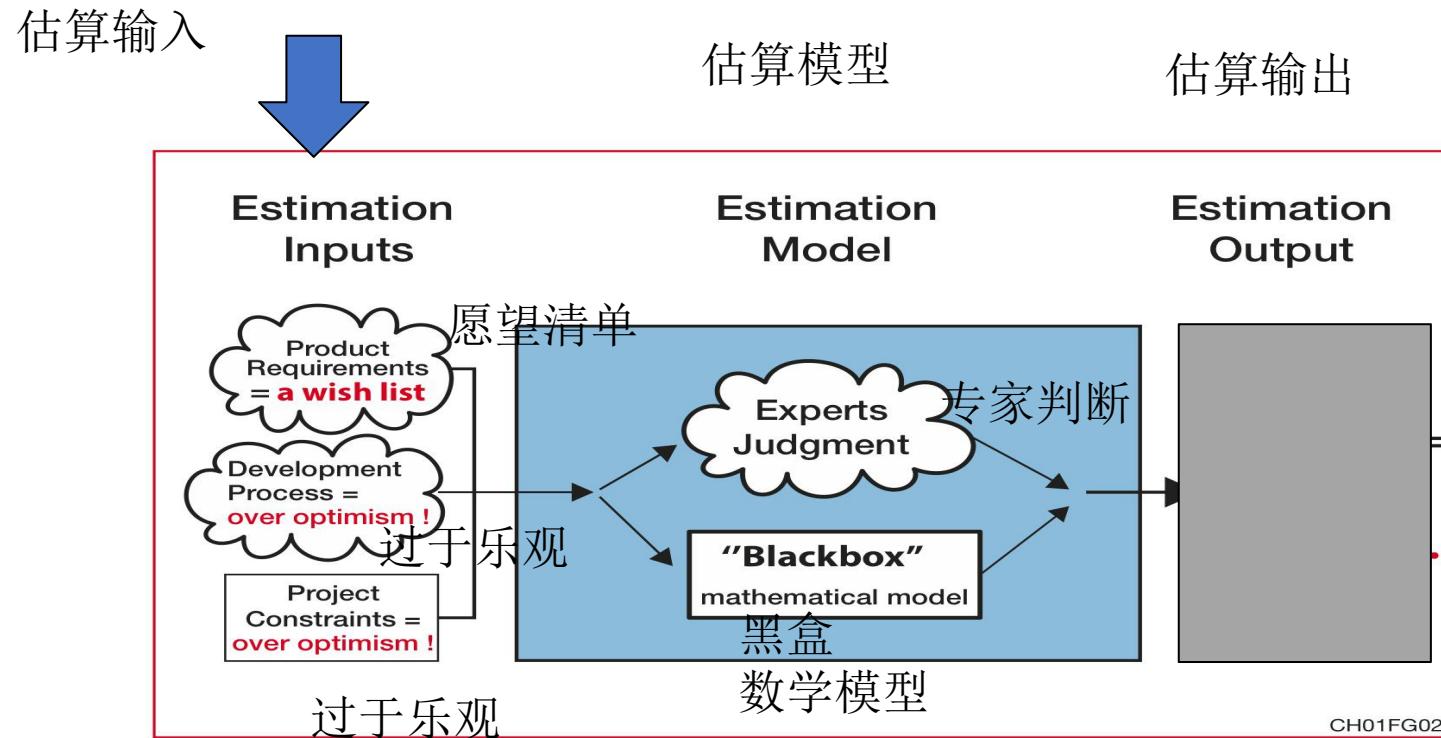
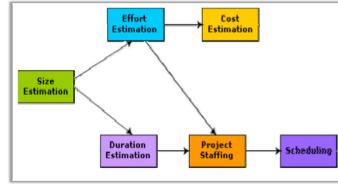


Figure 1.2 Some poor estimation practices observed in industry.

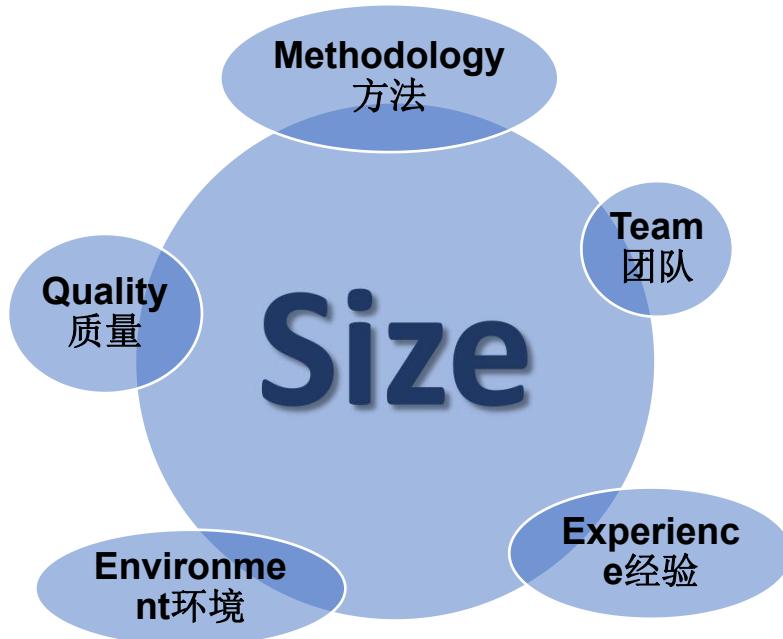
3

业内一些比较糟糕的估算实践

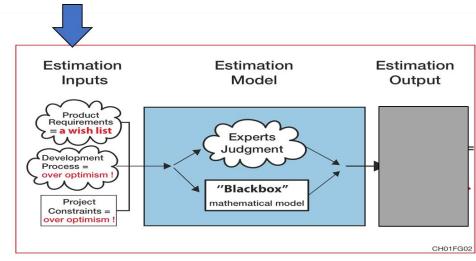
规模：主要变量 Size: Dominant variable



The right measurement tools ? 正确的度量工具



1. Story Points 故事点
2. Use Case Points 用例点
3. SNAP Points SNAP 点
4. Lines of Code 代码行
5. IFPUG Function Points IFPUG 功能点
6. COSMIC Function Points COSMIC 功能点





Measurement in real life! 实际生活中的度量

可再现**Repeatability**:

- different individuals, in different contexts, at different times, & following same measurement procedures **will obtain the same measurement results.** 不同的人，不同的背景，不同的时间，按照同样的度量步骤，得到相同的度量结果

Measurement results: 度量结果

- obtained with minimal judgment. 通过最少的判断获得
- results auditable. 结果经得起审计

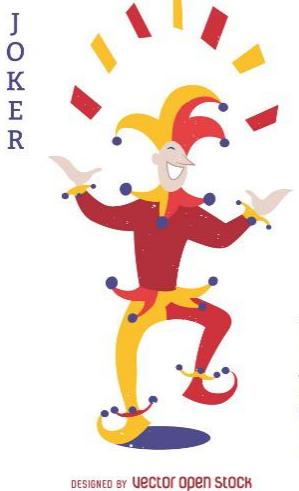
• **Accountability....! 可靠性**

Planning Poker & **Story Points** in Agile:

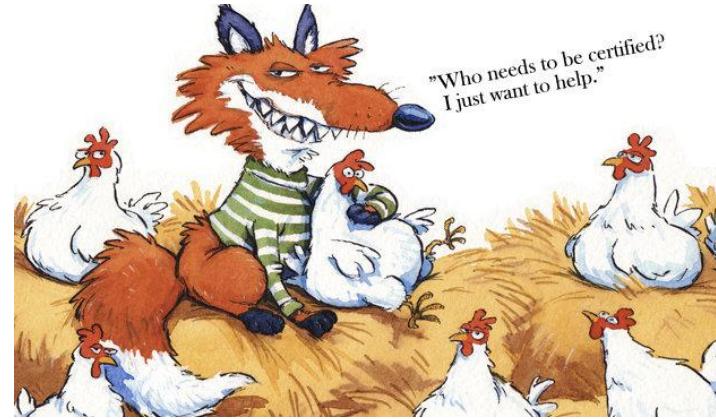
Do they meet measurement criteria? 敏捷中的策划扑克法和故事点：它们满足度量准则吗？



- Repeatability: 可再现
 - different individuals, in different contexts, at different times, & following the same measurement procedures will **NOT** obtain the same measurement results.
 - 不同的人，不同的背景，不同的时间，尽管按照同样的度量步骤，不会得到相同的度量结果
- Measurement results: 度量结果
 - obtained with minimal judgment. 通过最少的判断获得
 - results auditable. 结果经得起审计



Story Points 故事点



- **Unaccountability....! 不可靠**

Velocity 速率



Km per hour
每小时公里数

**Velocity
in Agile?**
敏捷中的速率

Actual hours 实际工时 = % 没有单位 = **percentage %**
Estimated hours 估算工时

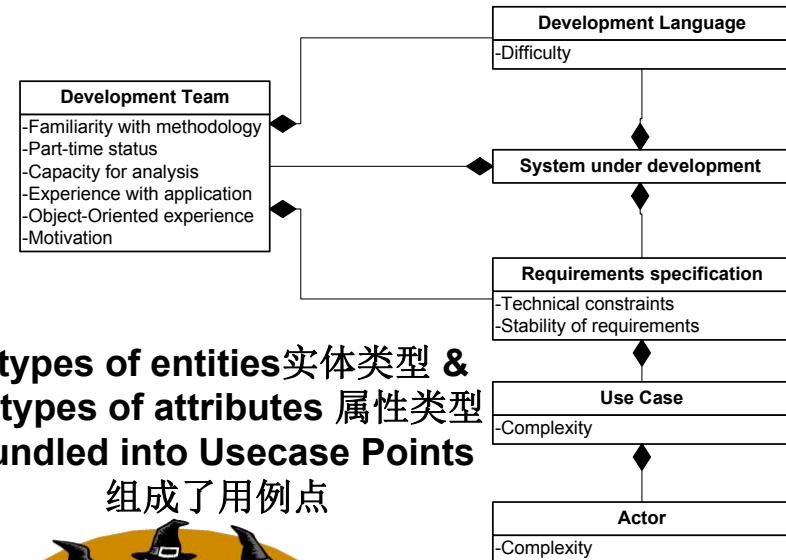
- **Unaccountability....! 不可靠**



Usecase Points用例点

Table 1: Entities, Attributes, and Measurement Rules

实体	属性	Measurement rule 度量规则
Entity	Attribute 属性	
Actor 用户	Complexity (of actor) 用户的复杂度	The type of complexity (simple, average, or complex) of the interaction between the actor and the system 用户与系统交互的复杂度类型 (简单, 中等, 复杂)
Use case 用例	Complexity (of use case) 用例的复杂度	The type of complexity (simple, average, or complex) measured in the number of transactions 通过业务数量度量的复杂度类型 (简单, 中等, 复杂)
	Relevance of the technical quality requirements 技术质量需求的相关性	The level of relevance (from 0 to 5) of each of the 13 known non-functional qualities 13个非功能质量需求的相关程度 (0-5) ,
Development team 开发团队	Stability of the requirements 需求稳定性	The level of stability (from 0 to 5) of the functional and non-functional requirements 功能和非功能需求的稳定性 (0-5)
	Familiarity with the methodology 对方法的熟悉程度	The level (from 0 to 5) of skills and knowledge of the development methodology in use for the project 项目所需的开发技能和知识的等级 (0-5)
	Part-time status 兼职状态	The level (from 0 to 5) of part-time staff on the team 团队中的兼职人员等级 (0-5)
	Analysis capability 分析能力	The level (from 0 to 5) of analysis capabilities of the development team with respect to project needs 开发团队针对项目需求的分析能力级别(0-5)
	Application experience 系统用的经验	The level (from 0 to 5) of team experience with the application domain of the system 团队对系统应用的经验等级 (0-5)
	Object-oriented experience 面向对象的经验	The level (from 0 to 5) of team experience with object-oriented design 团队关于面向对象设计的经验等级 (0-5)
Motivation 积极性		The level (from 0 to 5) of team motivation 团队的积极性能等级 (0-5)



5 types of entities 实体类型 &
11 types of attributes 属性类型
Bundled into Usecase Points

组成了用例点



It fails primary school
maths! 小学数学不及格

Fake Science
伪科学

Sizing Non-Functional Requirements: 度量非功能性需求

SNAP Points

	分类	子概念	权重
	Category	Sub-concepts for the classification	SNAP weights basis
数学运算	Data Entry Validation	Nesting level complexity 嵌套等级复杂度	2,3,4 * number of DETs
	Logical operations	Control flow complexity 控制流复杂度	4,6,10 * number of DETs
	Mathematical operations	Control flow complexity 控制流复杂度	4,6,10 * number of DETs
数据格式	Data formatting	Transformation complexity 变换复杂度	2,3,5 * DETs
内部数据移动	Internal data movements	Internal boundaries crossed DET transferred 内部边界	5 * (# of internal boundaries crossed)+2*(#DETss)
数据配置功能	Functionality by data config.	Complexity 复杂性	3,4,6 * Number of records 记录
UI变更	UI Changes	UI types complexity UI类型复杂度	2,-4 * DETs
帮助方法	Help methods	Help types 帮助类型	1,2,3 * number of Help items
多个输入	Multiple input methods	Media types 媒体类型	3,4,6 * number of controls
多个输入	Multiple output methods	Media types 媒体类型	3,4,6 * number of controls
多平台	Multiple platforms	No. of platforms to operate	8 * Number of platforms
数据库技术	Database technologies	Level & type of normalization of the physical schema 标准化等级和类型	1,3,4,5,7 * levels of normalization
系统配置	System configuration	中间件, 后台, 接口	$SP=(\text{middleware config.})+2*(\#\text{backend config.})+3*(\#\text{interface config.})$
批处理	Batch processing	Number of batches or transactions	2*(number of batches or transactions)
关键业务处理	System critical (real-time)	- Type of transactions - No. critical trans.	5,10,15 * number of critical transactions
基于软件的组件	Component based software	Type of components (In-house reuse or 3 rd party component) 内部复用或第三方组件	4,8 * number of unique components
设计复杂度	Design complexity	Interface complexity 接口复杂度	8,16,24 * # of COTS applications + 12,24,36 * #nonCOTS applications

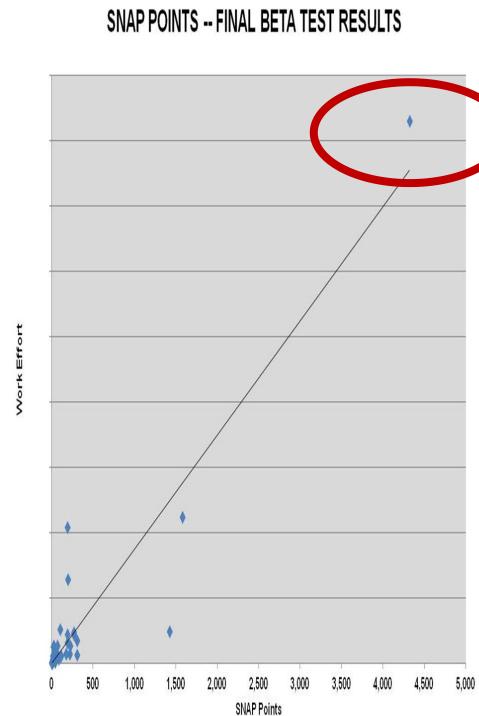


It fails primary school maths! 小学数学不及格



SNAP empirical study研究成員

N = 48, and R² = 0.89



来源Source: DOD
Crosstalk 2013

Figure 4:

SNAP POINTS -- FINAL BETA TEST RESULTS

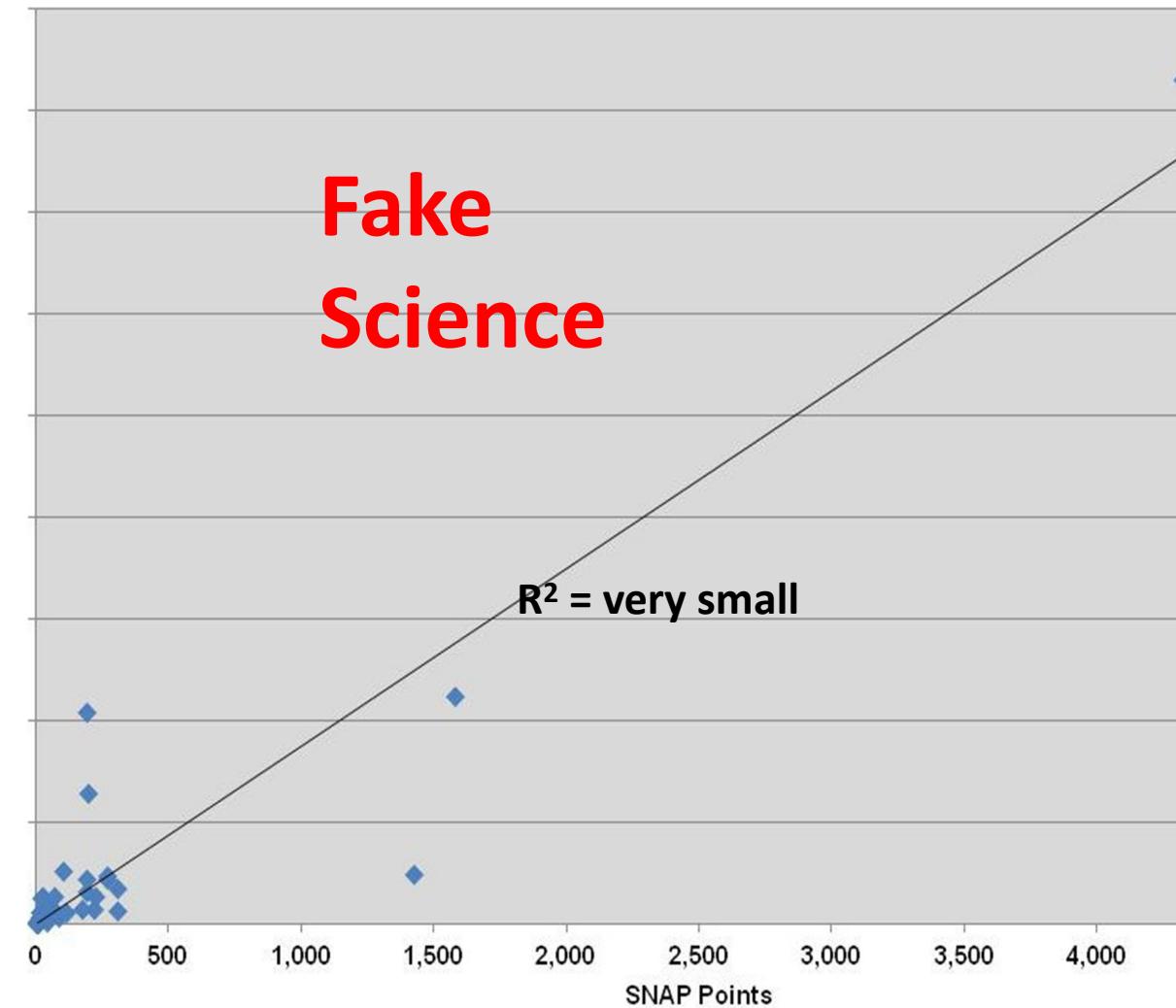


Figure 4:

SNAP Certification!

SNAP = Placebo 安慰剂



What do you think of being certified on a 'Placebo'?

你如何看待用安慰剂进行认证？

Software Metrics with too many factors & no sound measurement units

包含太多因子的软件度量以及不完备的度量元

The '*feel-good*'

感觉上很好



Quick &
Easy...又快又简单

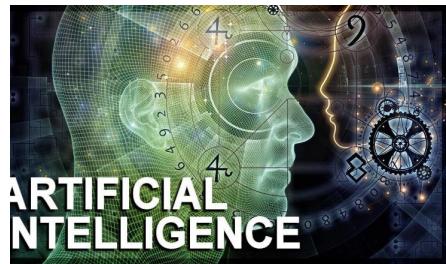


dead end! 死胡同



Lines of code-based estimation models

以代码行为基础的估算模型



Lines of code-based estimation models



Function Points 功能点

5 distinct ISO standards 5个不同的ISO标准

- ISO 20926 : IFPUG
- ISO 24570 : NESMA
- ISO 20968 : MRKII
- ISO 29881 : FISMA
- ISO 19761 : COSMIC

1st Generation 第一代

2nd Generation 第二代

*1st Generation of Function Points =
Complexity tables & Weights* 第一代功能点=复杂的表格和权重

FTR's	DATA ELEMENTS		
	1-4	5-15	> 15
0-1	Low	Low	Ave
2	Low	Ave	High
3 or more	Ave	High	High

Inputs - Matrix 输入
矩阵

FTR's	DATA ELEMENTS		
	1-5	6-19	> 19
0-1	Low	Low	Ave
2-3	Low	Ave	High
> 3	Ave	High	High

Output & Enquiries - Shared Matrix 输出查询
-共享矩阵

Rating	VALUES		
	EO	EQ	EI
Low	4	3	3
Average	5	4	4
High	7	6	6

Transactions: weights in FP (Function Points) 处理：功能点的权重

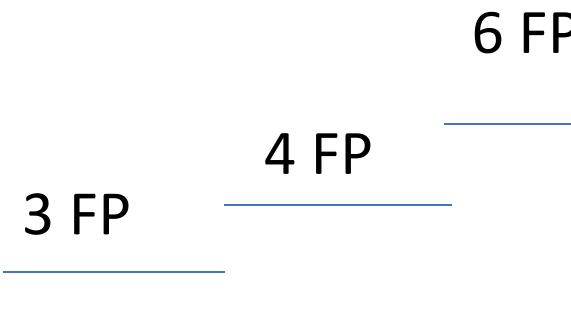
1st Generation key weakness 第一代方法的主要缺陷

Function Points weights =
Step functions 功能点权重 =
阶梯式

FTR's	DATA ELEMENTS		
	1-4	5-15	> 15
0-1	Low	Low	Ave
2	Low	Ave	High
3 or more	Ave	High	High

FTR's	DATA ELEMENTS		
	1-5	6-19	> 19
0-1	Low	Low	Ave
2-3	Low	Ave	High
> 3	Ave	High	High

Rating	VALUES		
	EO	EQ	EI
Low	4	3	3
Average	5	4	4
High	7	6	6

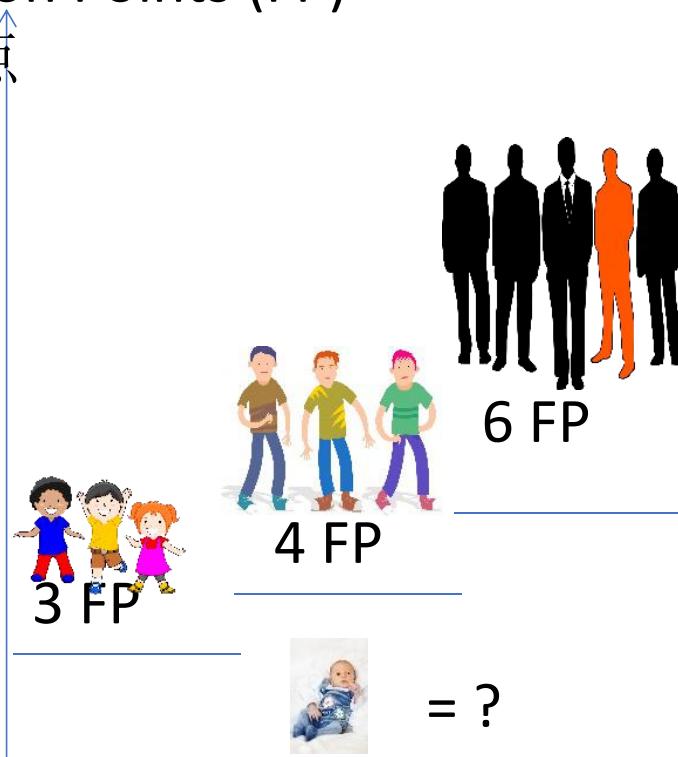


3-step size range - IFPUG External Input Transactions
3步规模范围- IFPUG 外部输入事务

1st Generation of Function Points 第一代功能点

Function Points (FP)

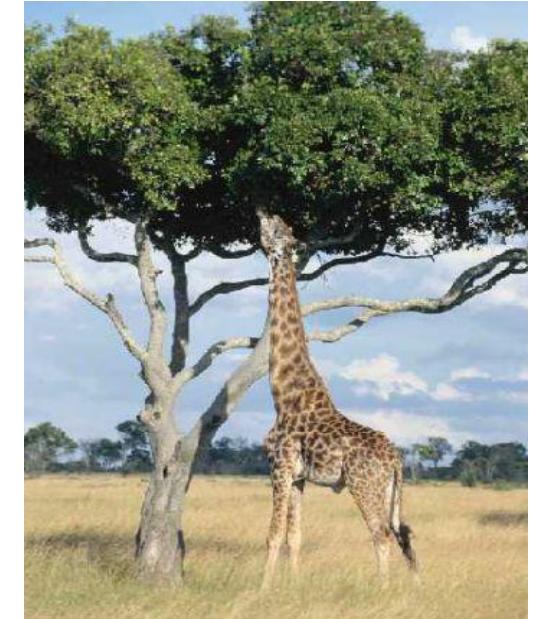
功能点



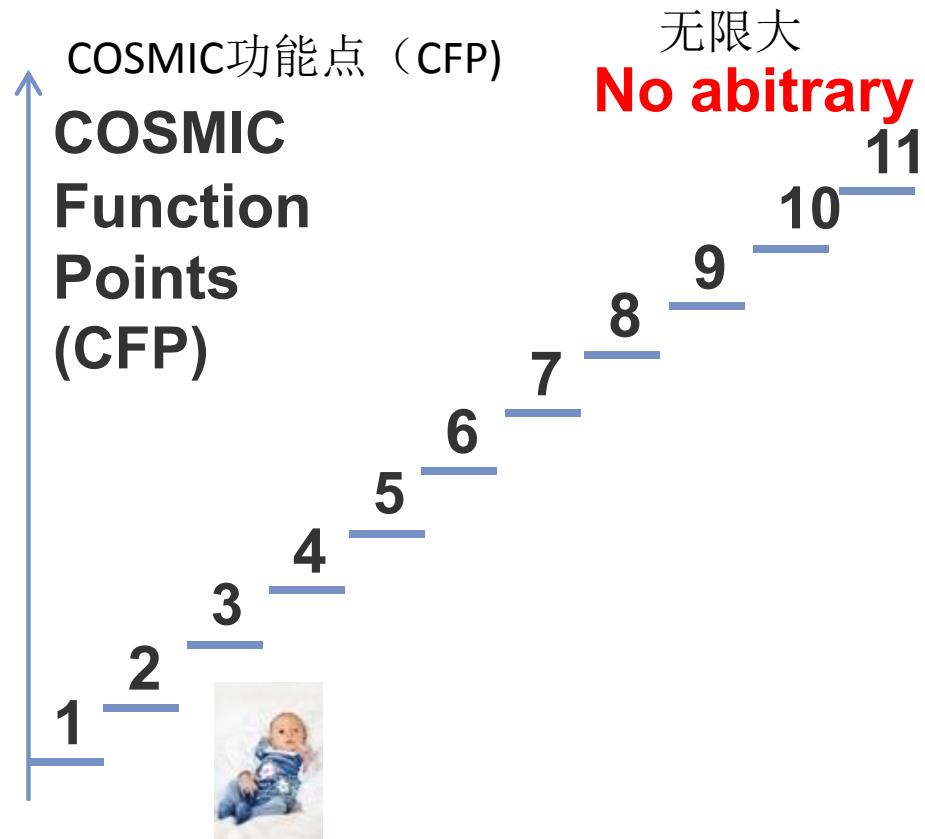
Shutterstock - 303008546

Key limitations: 主要缺陷

- **Only 3 values** 只有三个值
- **Limited ranges (min,max)** 范围有限（最小，最大）

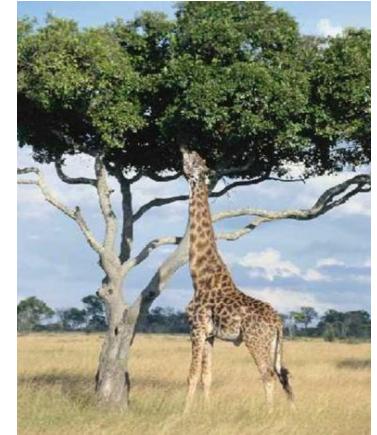


2nd Generation: COSMIC – ISO 19761 第二代



无限大
No arbitrary max

航空 In avionics > 100 CFP
银行业 In banking > 70 CFP



Black-box Estimation Tools 黑盒估算工具

Mix of: 包括

- Tables of weights (judgment approach)

权重表（判断法）

- Maths from books 数学

- Guesstimate of Lines of Code: 估计代码行数

❖ Limited knowledge of mix of programming languages, etc. 有限的编程语言知识

- Too many variables & cost drivers 太多变量&成本因子

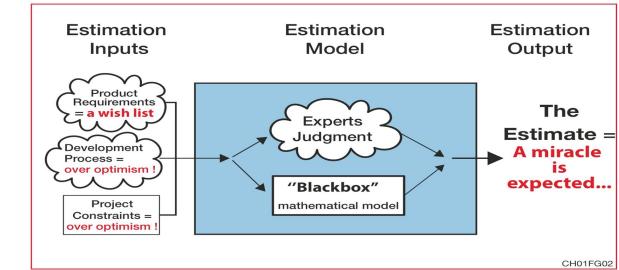
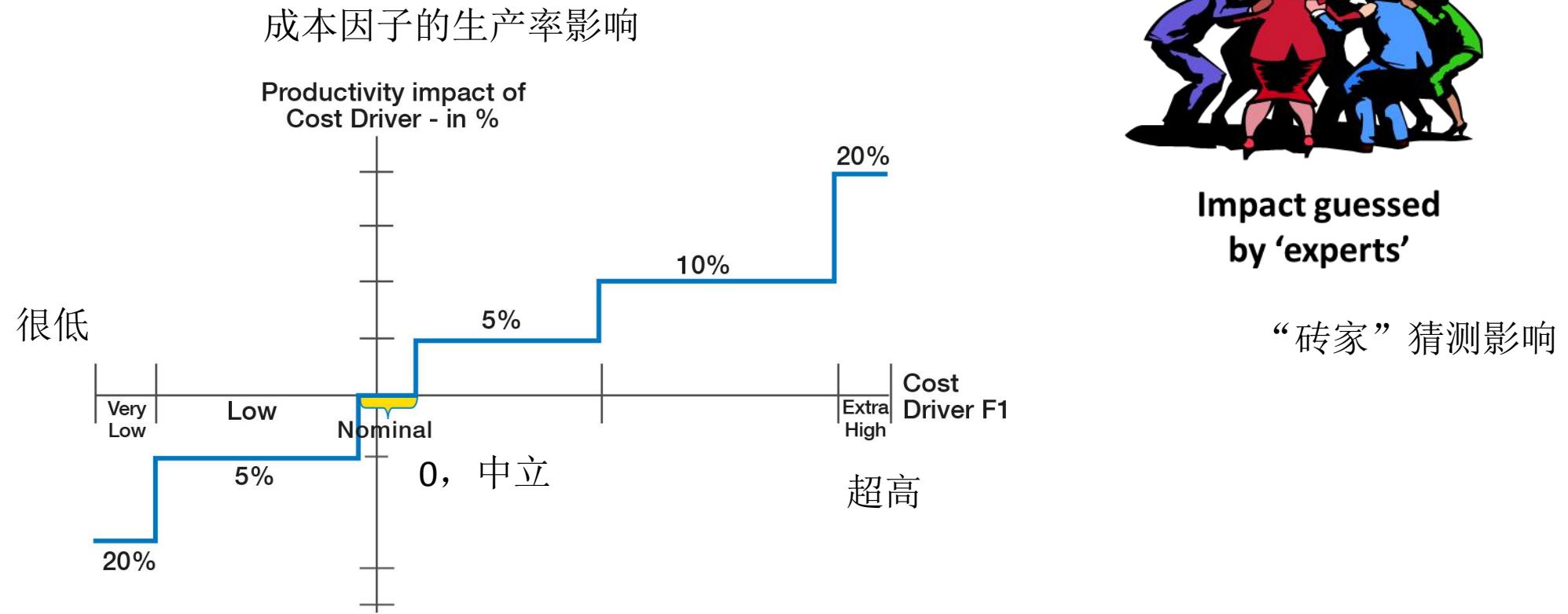


Figure 1.2 Some poor estimation practices observed in industry.

3

‘COCOMO-like’ Factors = Step Functions 阶梯函数



Tables of weights 权重表 = Step-function 阶梯函数 = Approximation 近似法

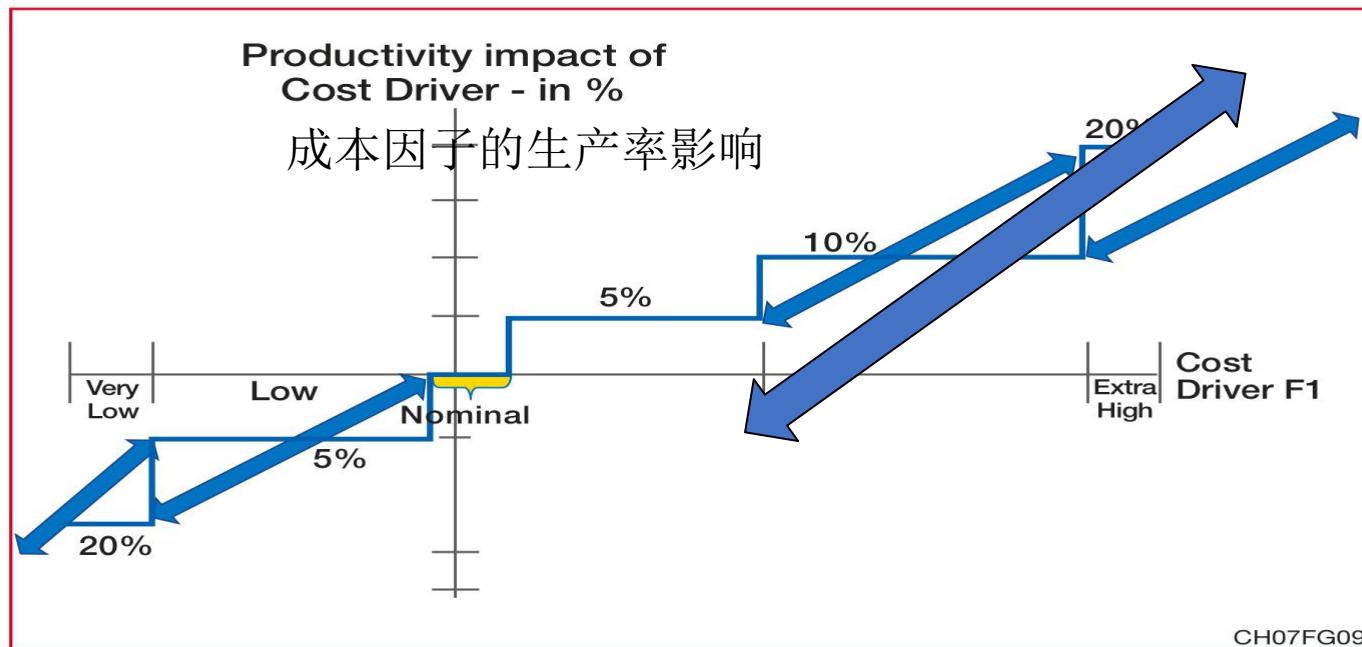


Figure 7.9 Approximation of step-funtion productivity models with irregular intervals.

表7.9 区间不规则的函数生产率模型的近似方法



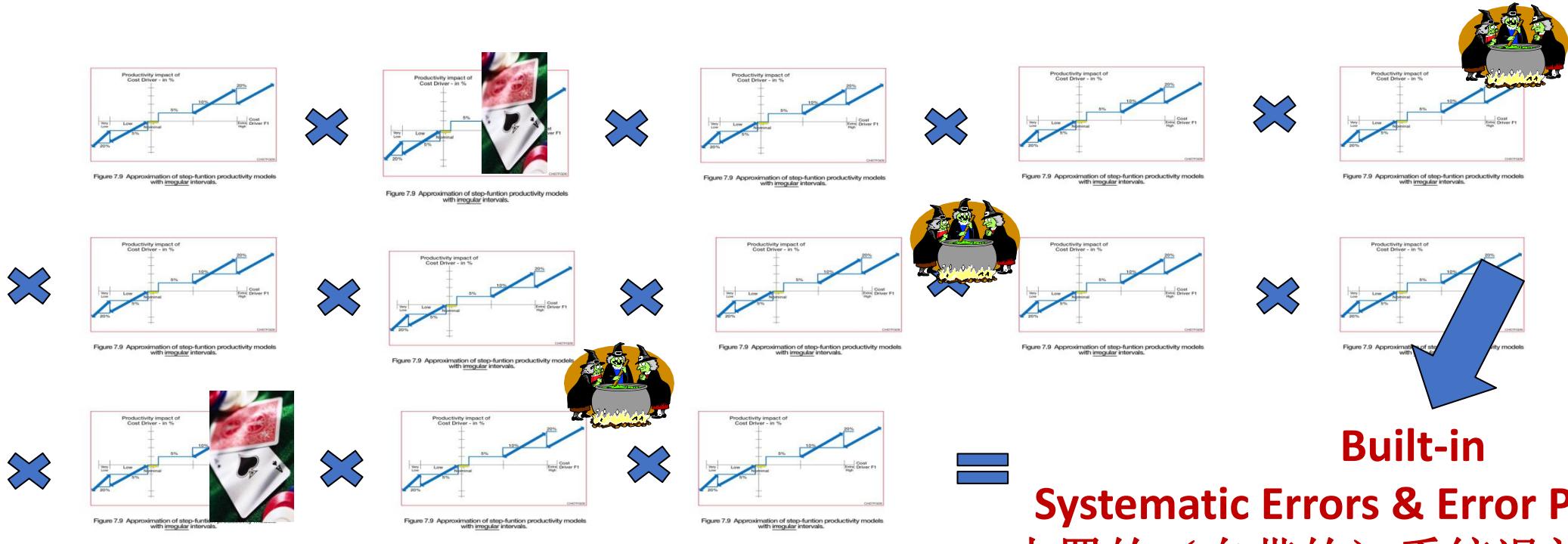
Each COCOMO cost driver
每个COCOMO成本因子 =

an estimation
sub-model
with unkown quality &
large errors 质量未知&误
差较大的估算子模型



COCOMO-like estimation models: 估算模型

Effort = function (Size & Step-functions of unknown quality combined into a single number!
工作量是规模阶梯函数的一个函数，其质量未知且结果是一个单点值



Estimation Models with too many factors... 因子太多的估算模型

The '*feel-good*' 感觉上很好



Quick &
Easy...又快又简单



List of topics

1. How to recognize Estimating Placebos & Fake Science?

2. How to build a competitive advantage?

3. My recommendations for software sizing

Estimation: 估算
not a 1-step technical task
并非一步搞定

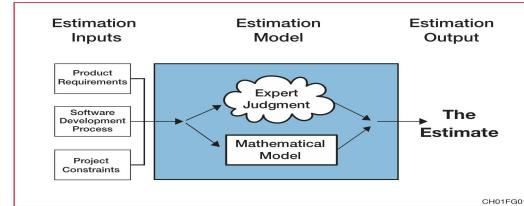
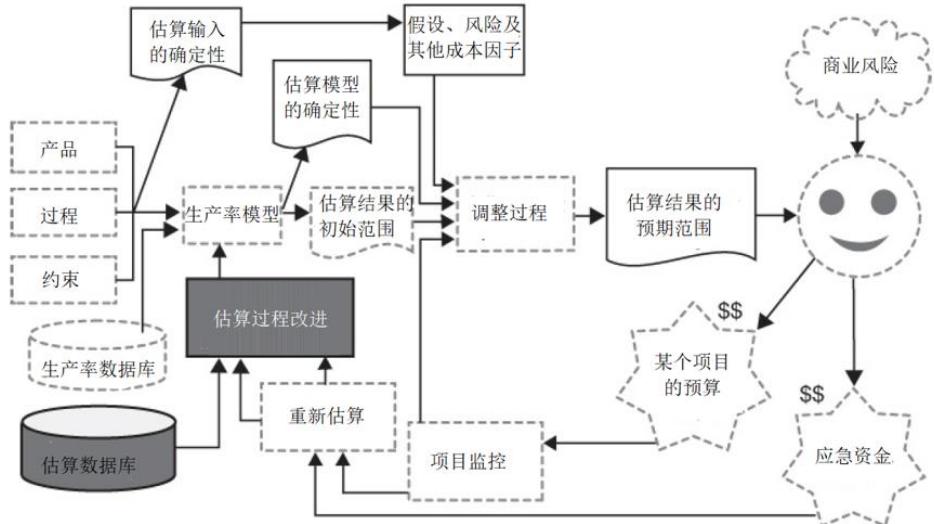
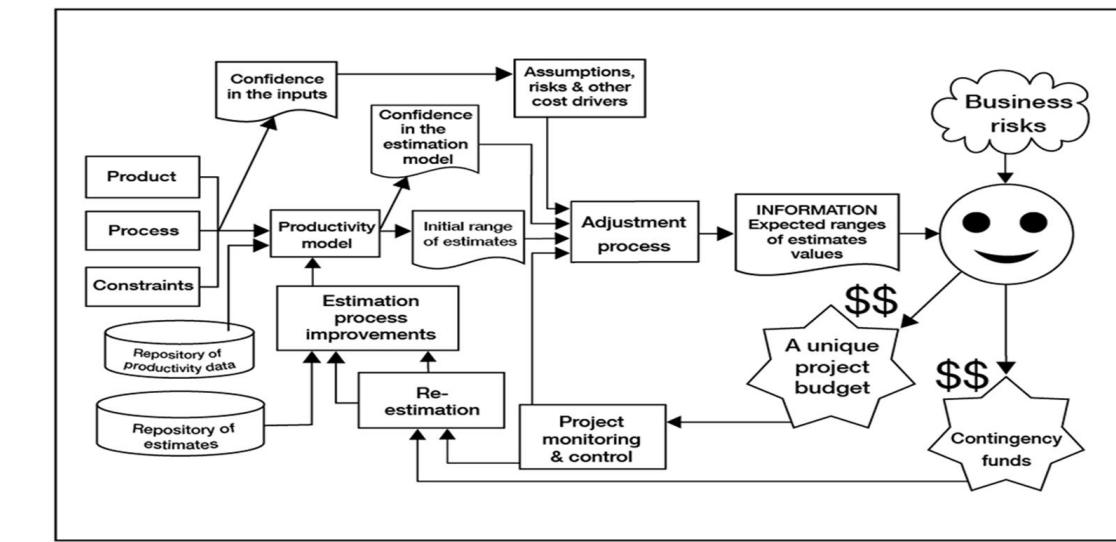
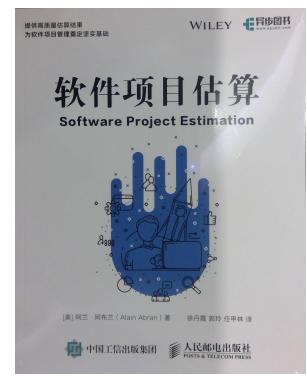
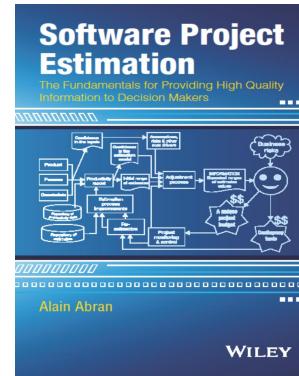


Figure 1.1 Common view of an estimation process.



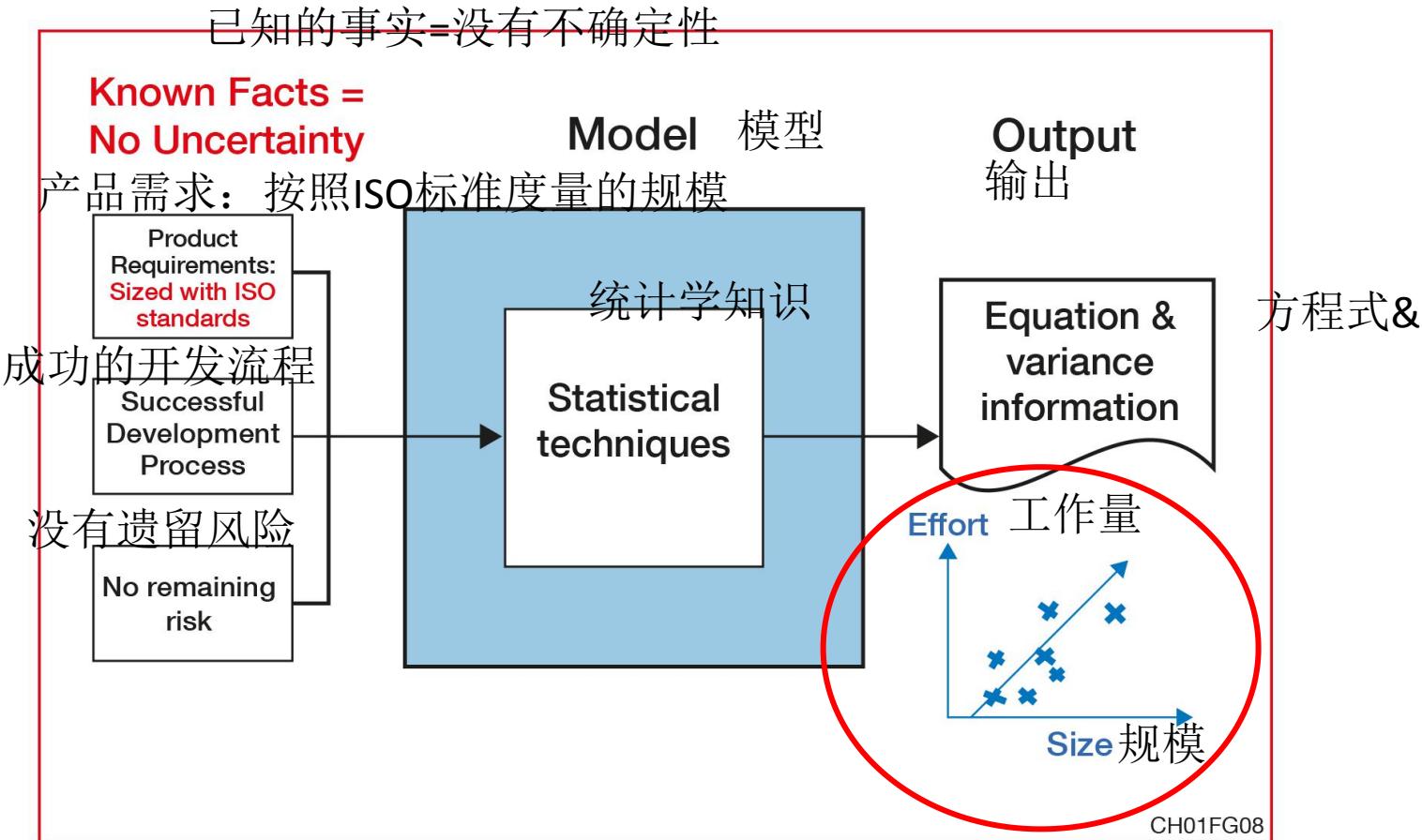
Estimation: 估算
A full cycle management process
全周期的管理流程



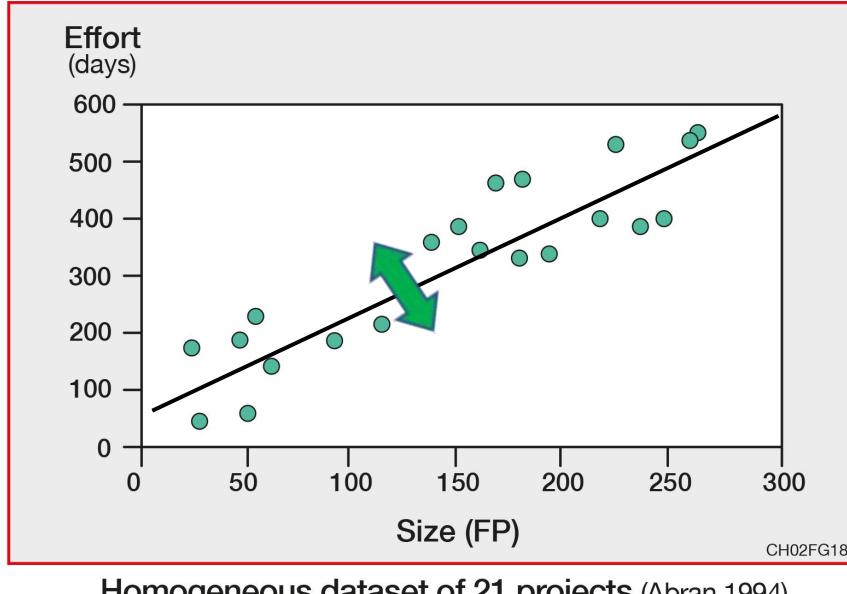
估算流程的第一步 Initial step of an Estimation Process

What is your starting point? 你从哪里开始?

What is your own productivity? 你们公司的生产率是什么?



Size as a dominant variable in a dataset 规模 作为数据集中的主要变量



有21个项目的同质数据集



In a development process under control: 在可控的开发流程中：

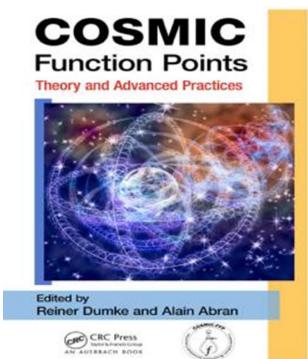
- Size explains **80 to 90%** of the Effort variation
规模可以解释80%-90%的工作量的变化。
- **all other factors combined impact 5% to 20%!**
其他的因素加起来的影响是5%-20%。

COSMIC method in Automotive
embedded software

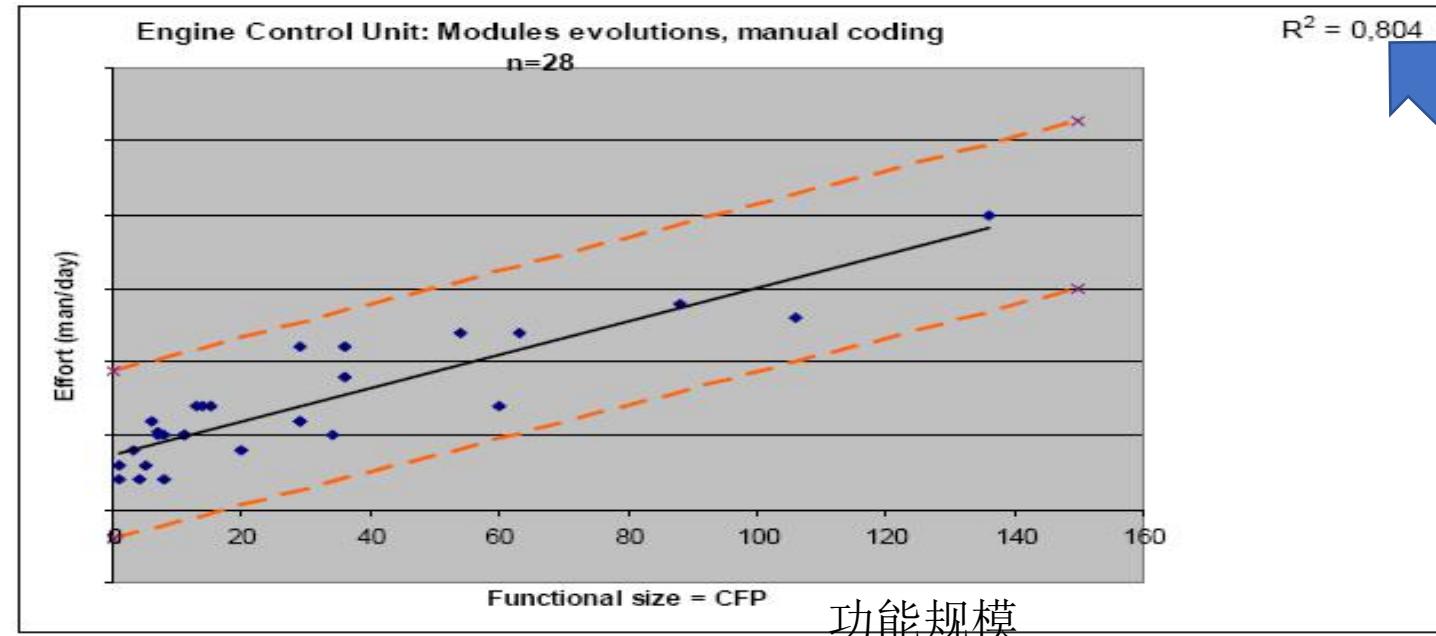
COSMIC在汽车行业嵌入式软件中的应用

By: Sophie Stern

Renault 雷诺汽车

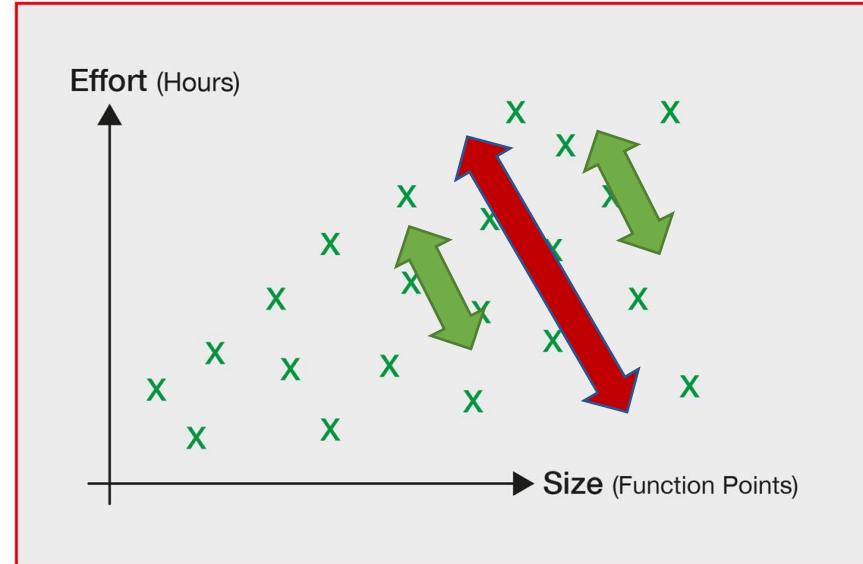
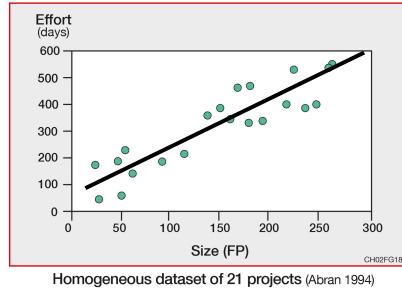


引擎的控制单位：模块演变，手工编码



© Copyrights Renault 2012

Size as a dominant variable in a dataset 规模 作为数据集中的主要变量



Development Process out of control! 不受控的开发过程

- Management investments required to get the development process back to within control limits 需要进行管理投资，以使开发过程受控。

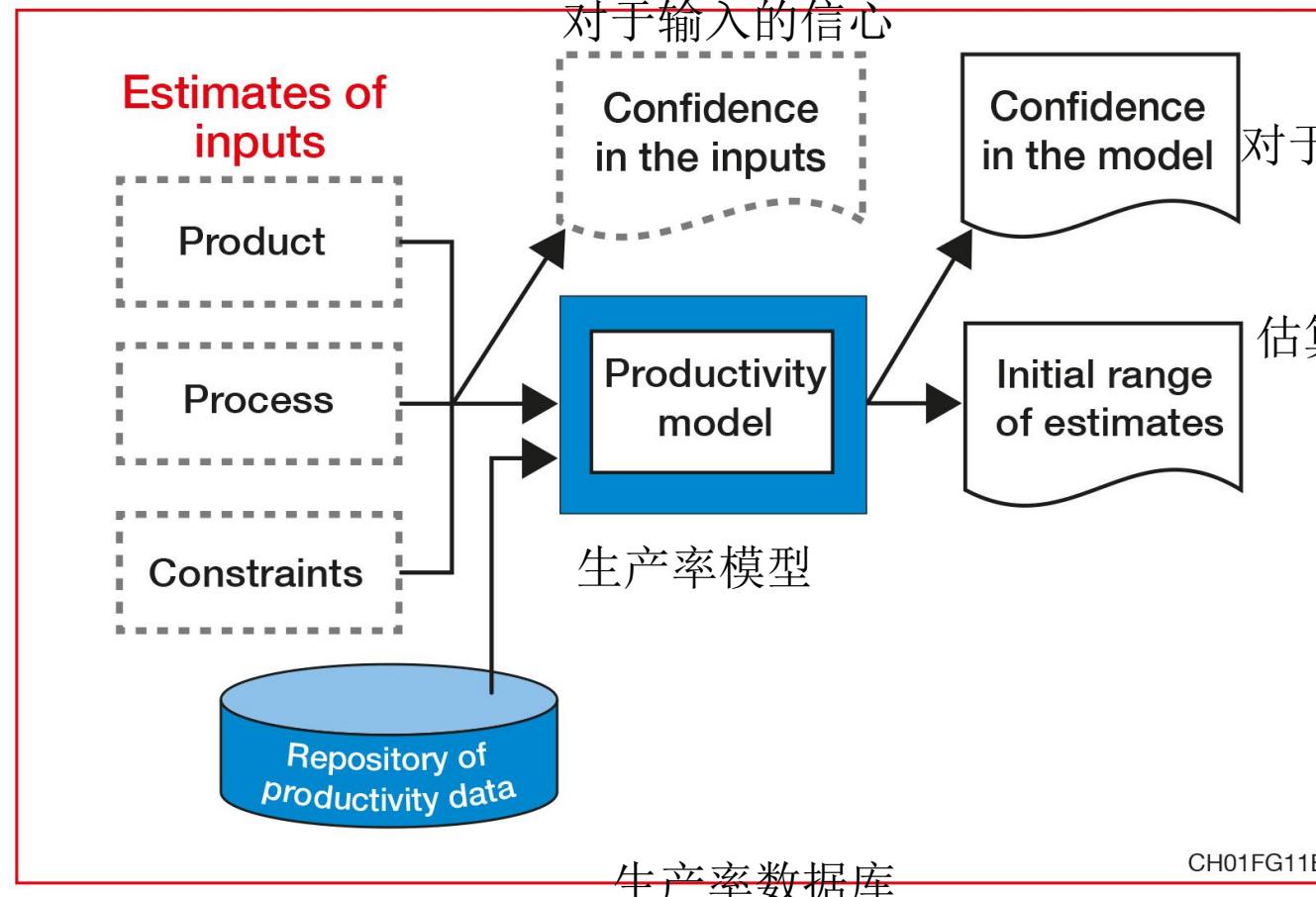
Execution of the productivity model 执行生产率模型

估算输入

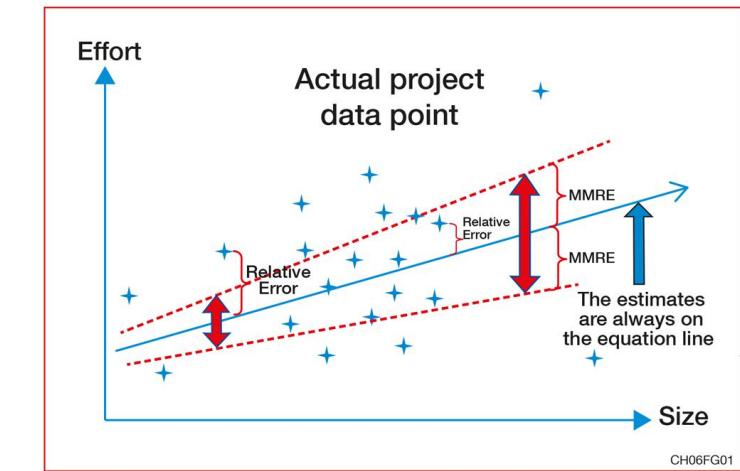
产品

流程

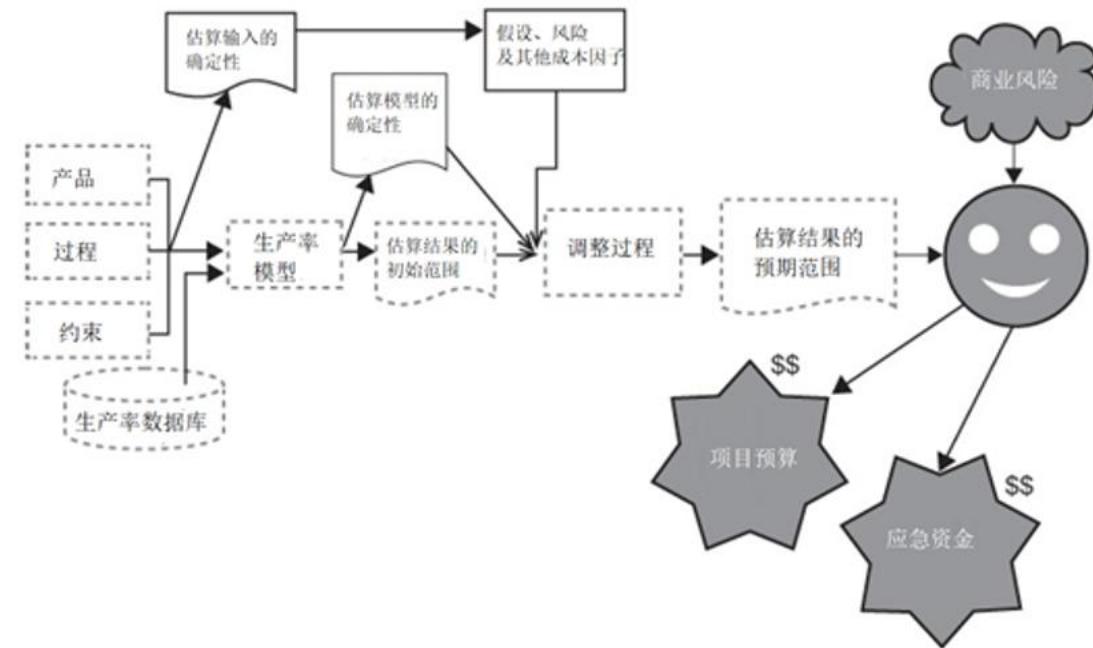
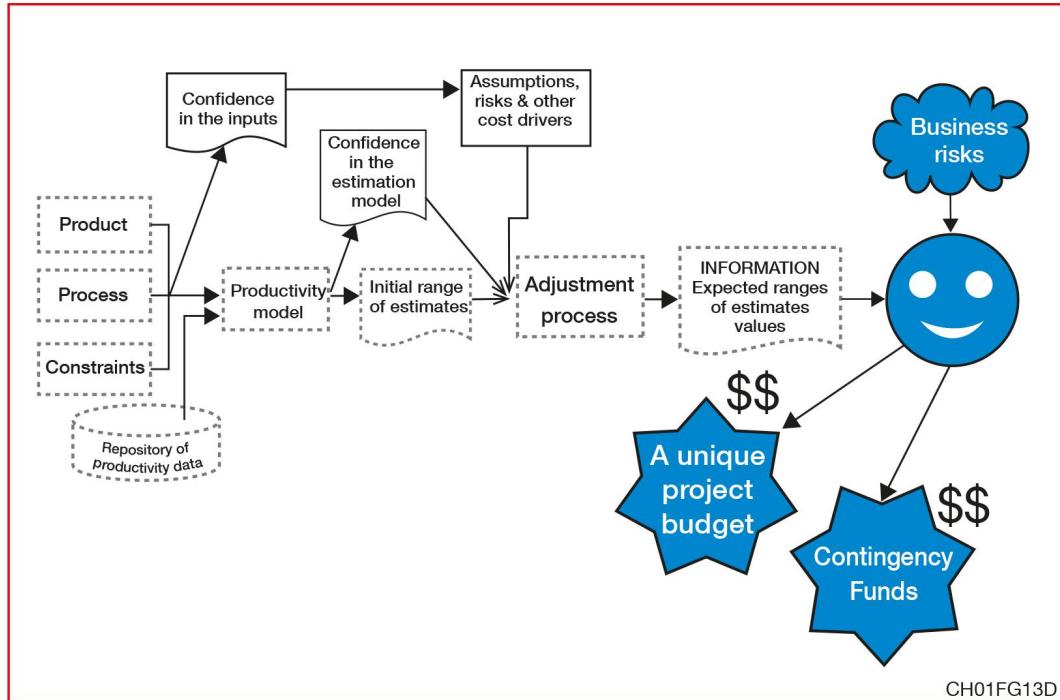
约束



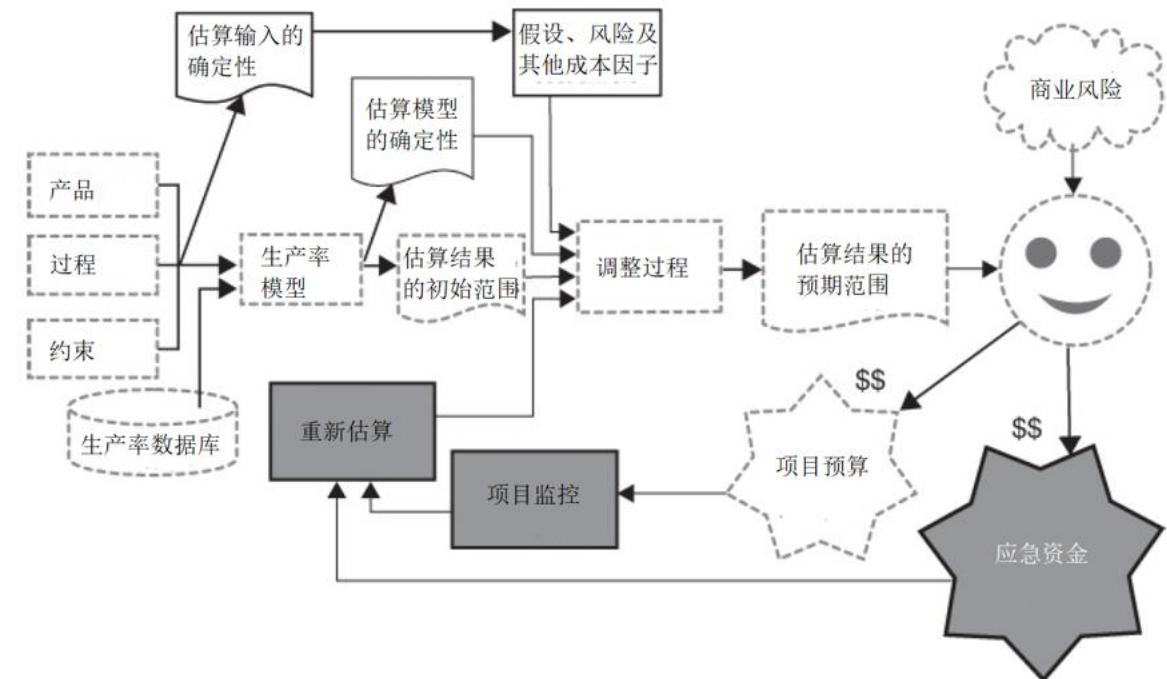
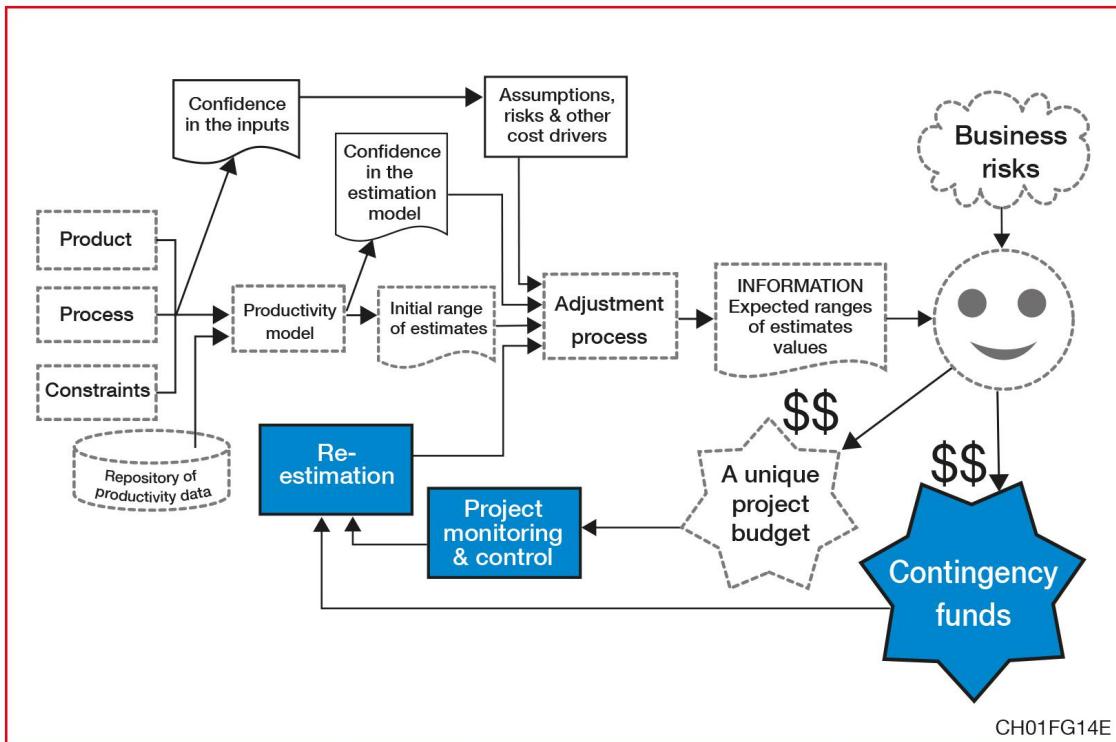
估算结果的初始区间



The budget decision 预算决策



The re-estimation process 重新估算



Estimation Roles 参与估算的角色

Estimator 估算人员

- Build the productivity model 建立生产率模型
- Provide information on expected range of estimates 提供预测估算的区间结果



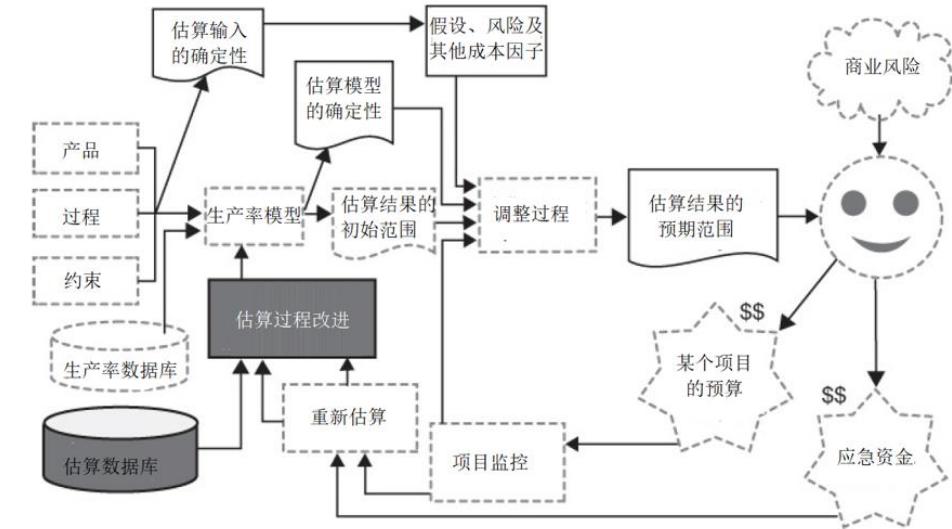
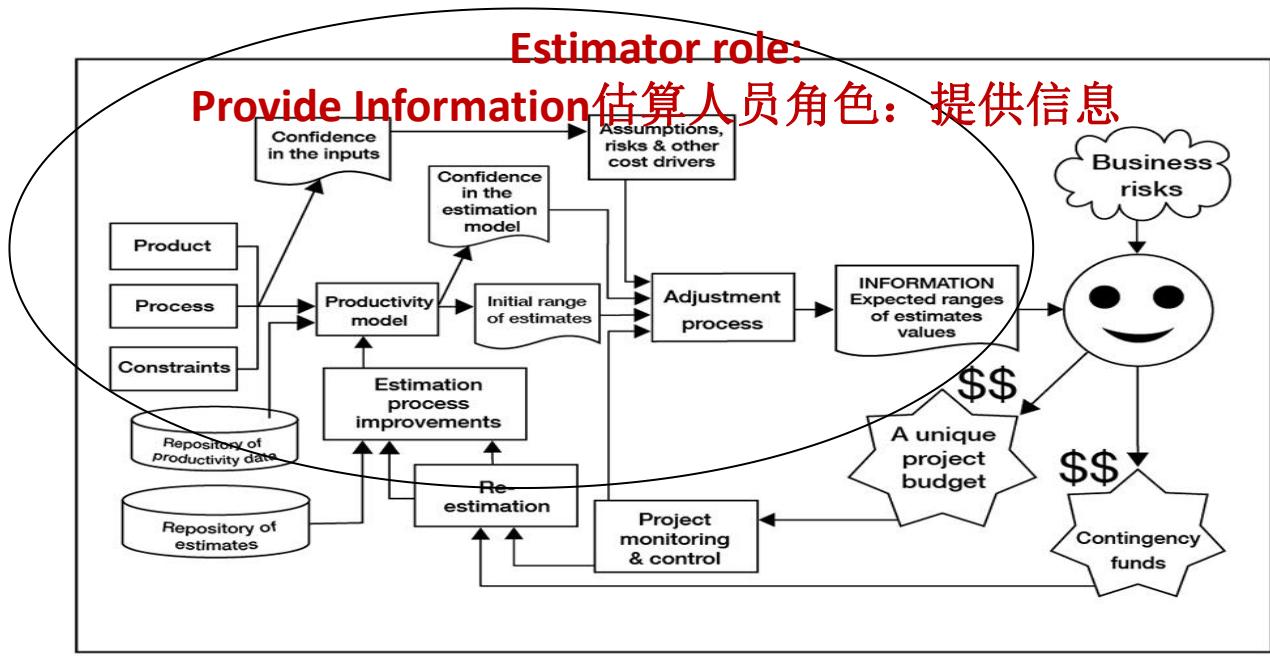
<http://www.ppiclaimcompany.co.uk/claim-estimator>

Manager 管理者

- Select the optimal budget 选择最优预算
- Allocate resources for implementing the estimation process 为实施估算过程分配资源
- Assign skilled & trained resources to the estimation process 为估算过程分配有技能且经过培训的资源

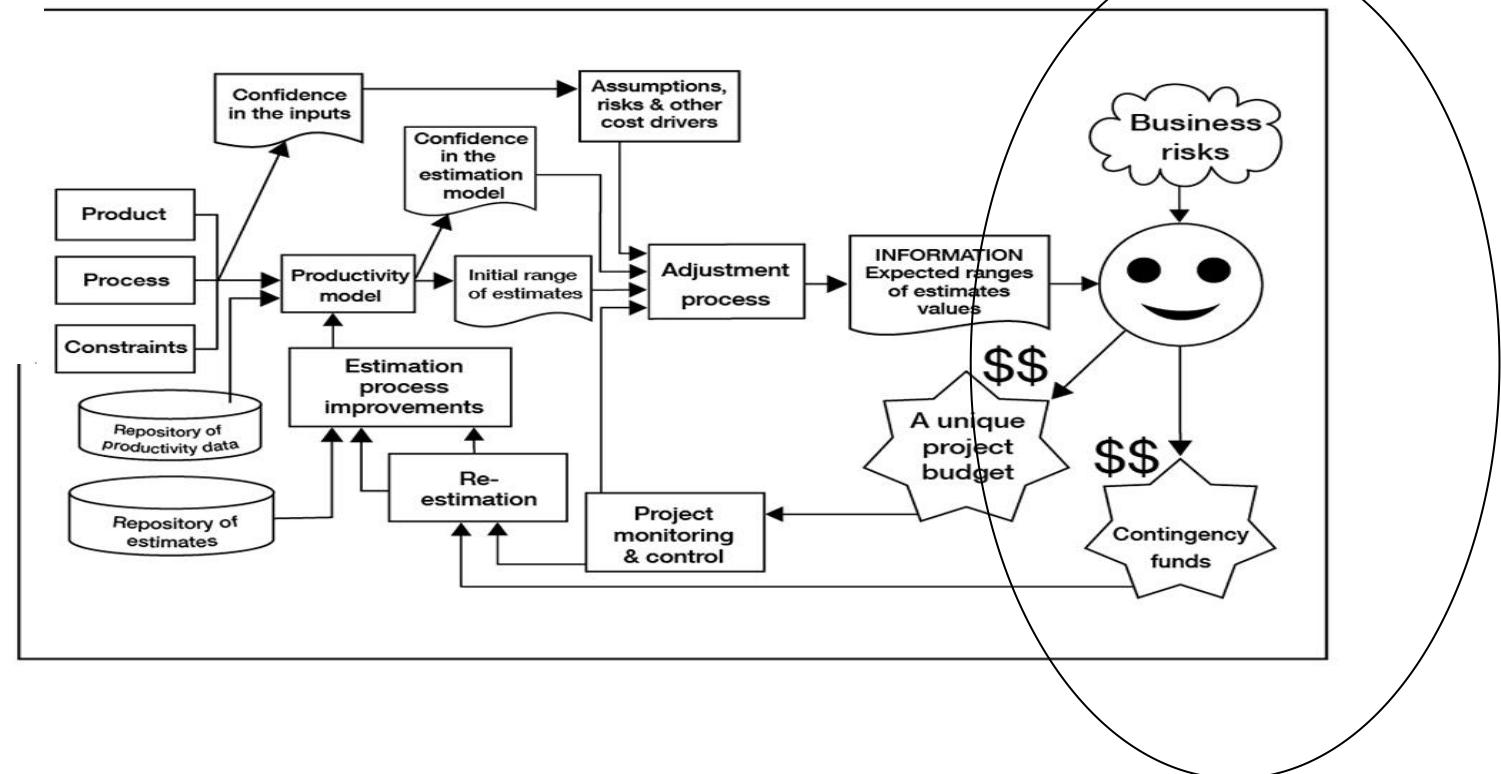
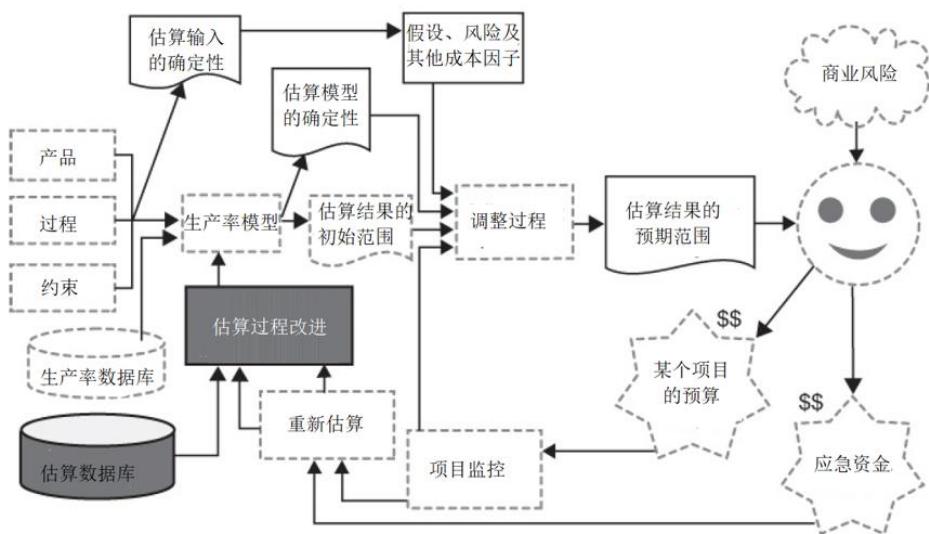


<http://www.successandyouth.com/youth-development/are-you-a-good-decision-maker.html>



Manager role:
Pick a number

& Manage Risks
经理的角色：选择一个值并管理风险



List of topics

1. How to recognize Estimating Placebos & Fake Science?

 2. How to build a competitive advantage?

 3. My recommendations 我的建议
-

Function Points 功能点

5 distinct ISO standards 5个不同的ISO标准

- ISO 20926 : IFPUG
- ISO 24570 : NESMA
- ISO 20968 : MRKII
- ISO 29881 : FISMA
- **ISO 19761 : COSMIC**

1st Generation 第一代

2nd Generation 第二代

COSMIC Function Points 功能点- ISO 19761

Every software is different, but 每个软件都不一样,

what is common across all software: 那每个软件有什么相同点?

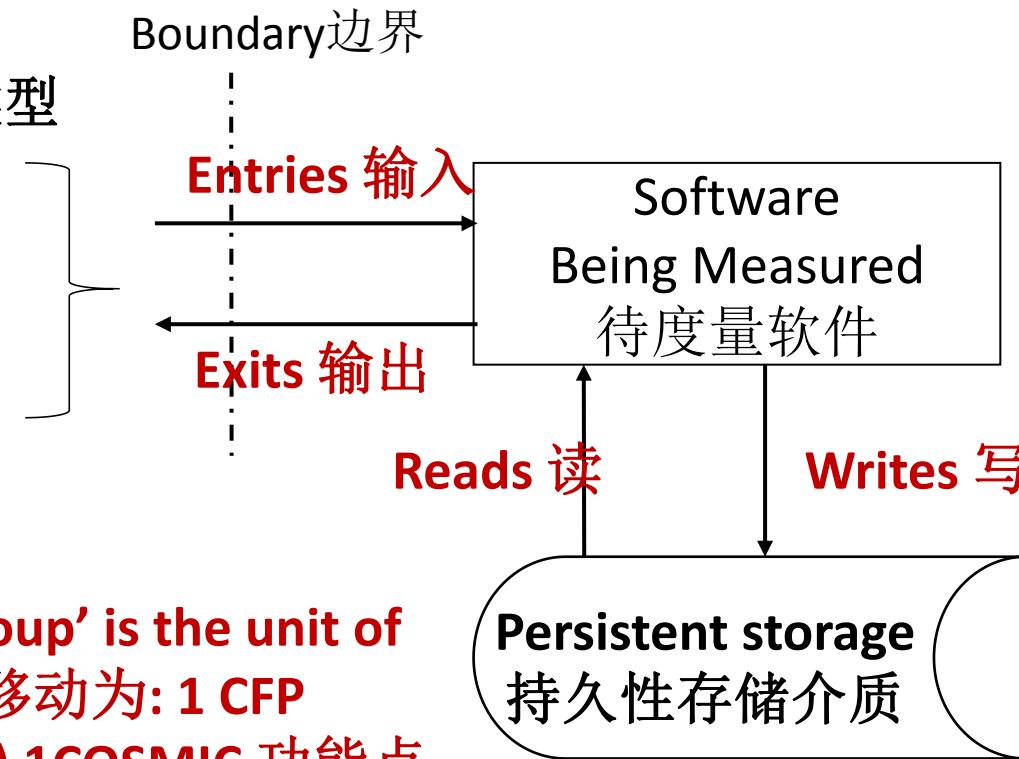
- In different types of software? 不同类型的软件
- In very small software? 非常小的软件
- In very large software? 非常大的软件
- In distinct software domains? 不同的软件领域
- In various countries? 不同国家

COSMIC view of software COSMIC视角的软件

All software does this:所有软件依据如下规则划分:

Functional Users types:功能用户类型

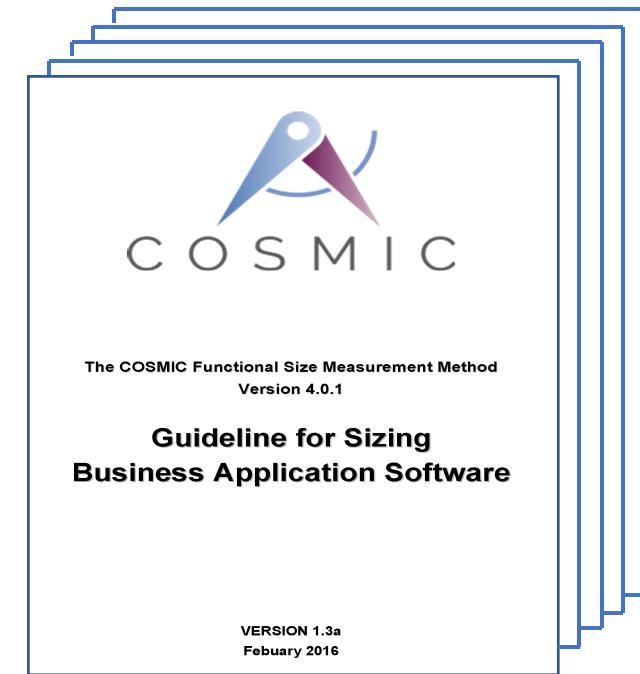
1. Humans 人类
2. Hardware devices 硬件设备
3. Other software 其他软件



The 'Data Movement of 1 data group' is the unit of measurement
1个数据组的数据移动为: 1 CFP
(1 CFP = 1 COSMIC Function Point) 1COSMIC 功能点

COSMIC Guidelines指南

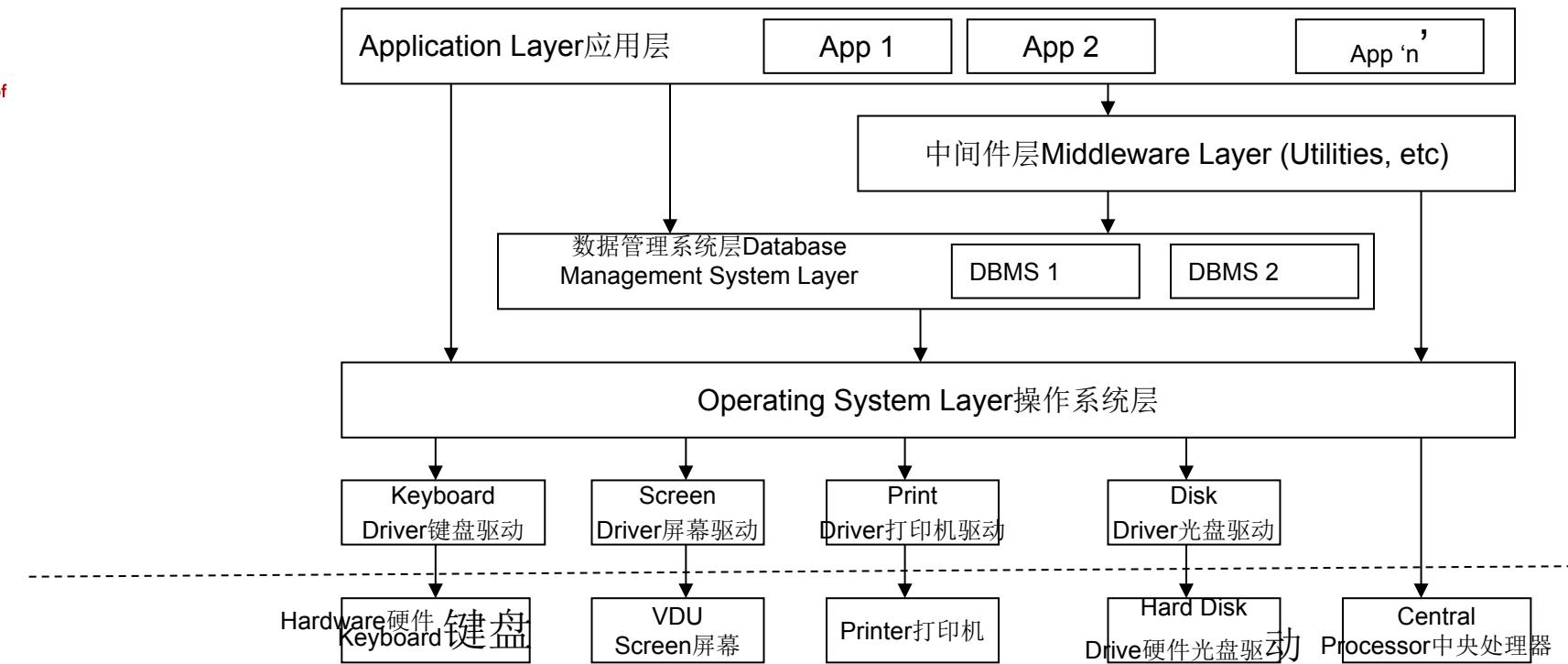
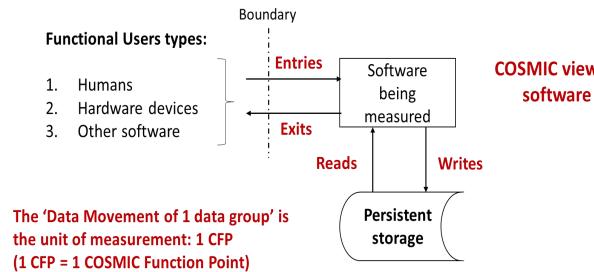
1. Business applications 业务应用类
2. Real-time software 实时类软件
3. Data Warehouse software 数据仓库软件
4. SOA software (SOA: Service Oriented Architecture) 面向服务的结构软件
5. Mobile apps 手机应用
6. Agile Development 敏捷开发



All COSMIC documents free on the web at: 所有COSMIC的文档都是免费在网下载的:

www.cosmic-sizing.org

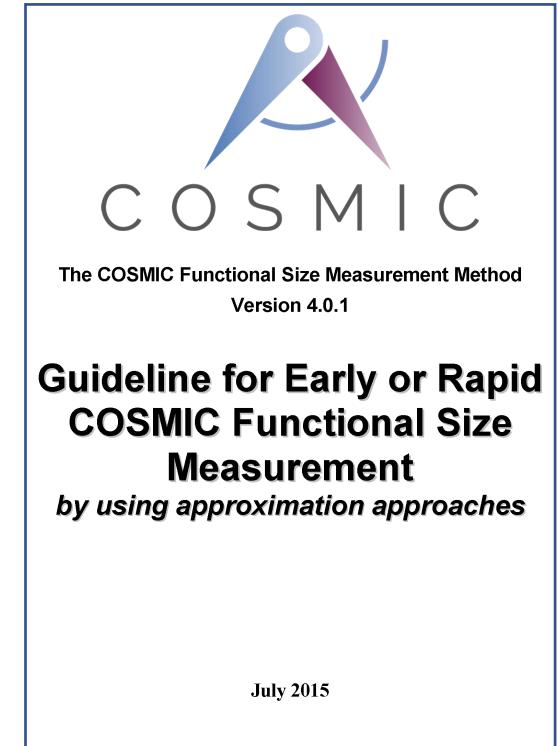
COSMIC – applicable throughout a software architecture 在软件整个架构中都可以使用COSMIC



COSMIC at Estimation Time 估算时如何使用COSMIC

Approximation techniques 近似估算技术 (with reported use, strengths & weaknesses 报告使用情况, 优点, 缺点):

1. Average functional process 平均功能处理近似估算
2. Fixed size classification 固定规模分类近似估算
3. Equal size bands 相等规模带近似估算
4. Average use case 平均用例近似估算
5. Software Iceberg analogy 软件冰山类比法
6. Easy function points 简单功能点近似估算
7. Approx. from informally written texts 非正式文档近似估算
8. Etc. 等等



早期&快速功能规模
度量指南

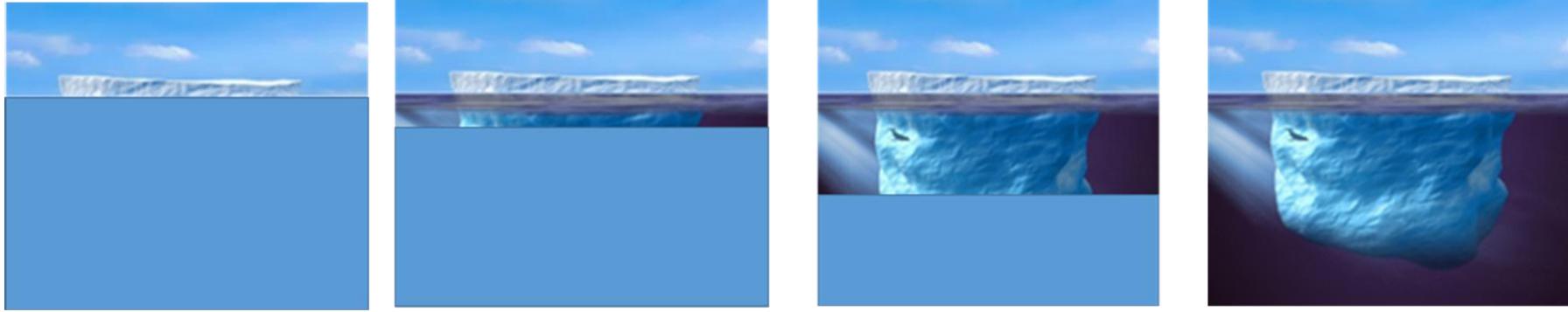
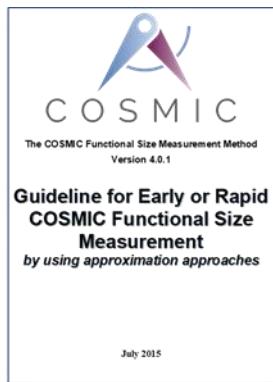
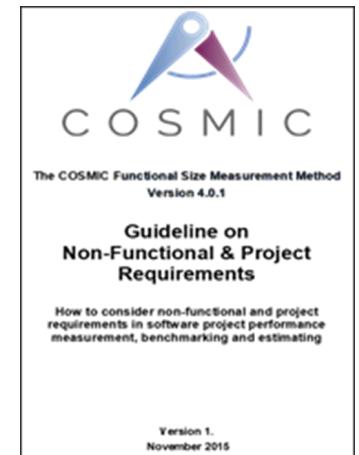
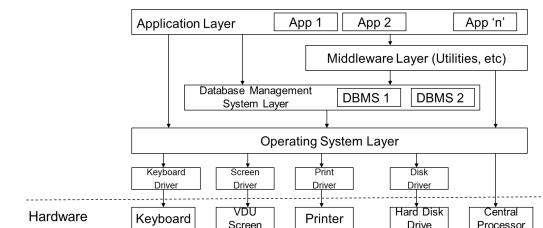


Figure 6: The Iceberg analogy : Initially visible functions (left) to full view (right)
 图6：冰山理论：最初可见的功能（左边），全视角（右边）

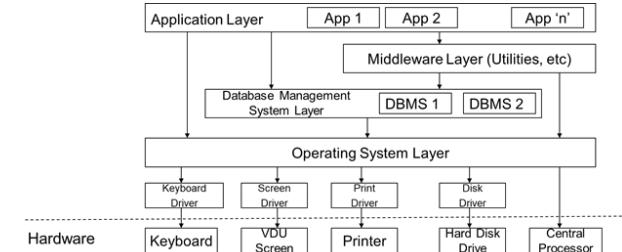
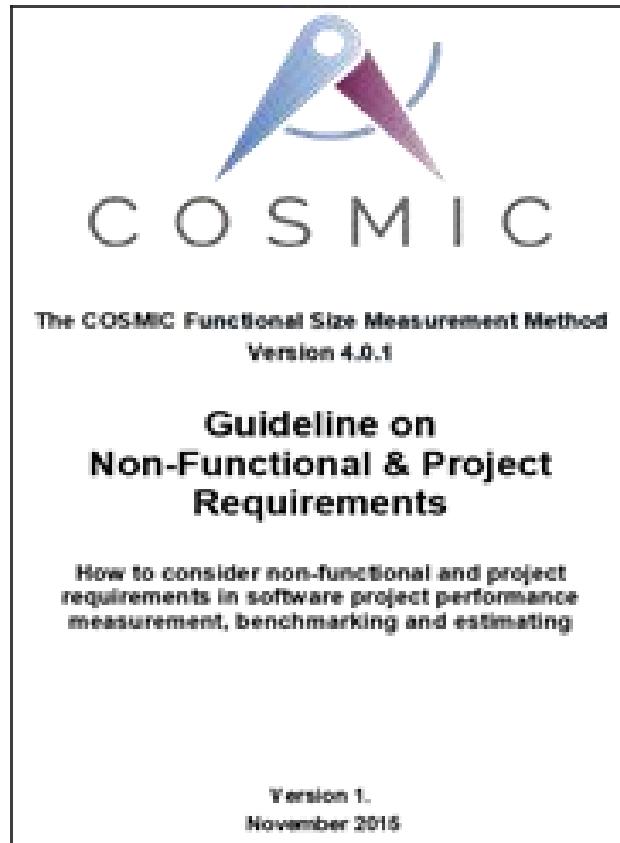


Functional Requirements 功能需求

Non Functional & Quality Requirements 非功能*质量需求



Non Functional Requirements with COSMIC 非功能需求



- A Standards-based Reference Framework for System **Portability Requirements**, 系统可移植性的基于标准的参考框架
 - Computer Standards and Interfaces Journal, Elsevier, Vol 35, 2013 (Abran, Al-Sarayreh...) <http://dx.doi.org/10.1016/j.csi.2012.11.003>
 - 计算机标准和接口期刊
- 'A Standards-Based Model of System **Maintainability Requirements**',
 - Journal of Software: Evolution and Process, 2013, Vol. 25, no. 5. (Al Sarayreh, Abran...)
 - 系统维护性需求的基于标准的模型
-

Portability NFR 可移植性非功能需求

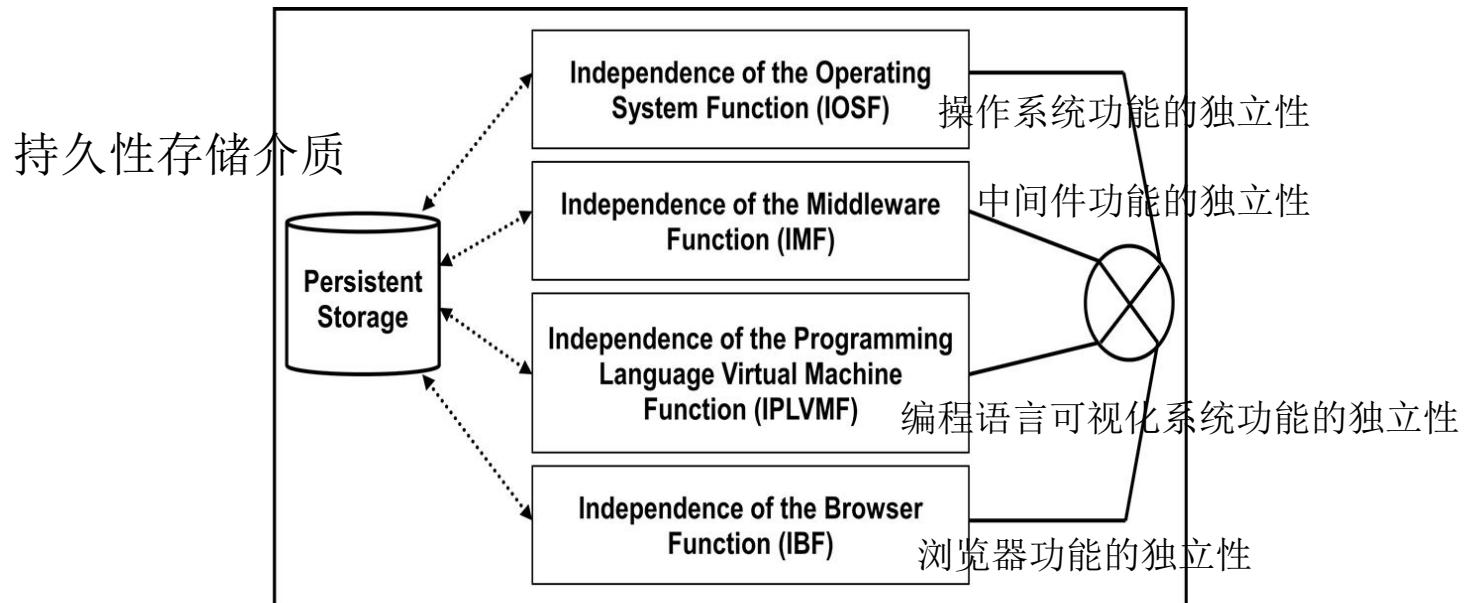


Fig. 3 Software Component Portability functions: system modeling view

图3：软件-组件，可移植性功能：系统模型视图

Alain Abran, Khalid T. Al-Sarayreh, and Juan J. Cuadrado-Gallego, 'A Standards-based Reference Framework for System Portability Requirements', Computer Standards and Interfaces Journal, Elsevier, Vol 35, 2013, pp. 380-395.
<http://dx.doi.org/10.1016/j.csi.2012.11.003>

Portability Example with Valve Control 可移植性例子： 阀门控制

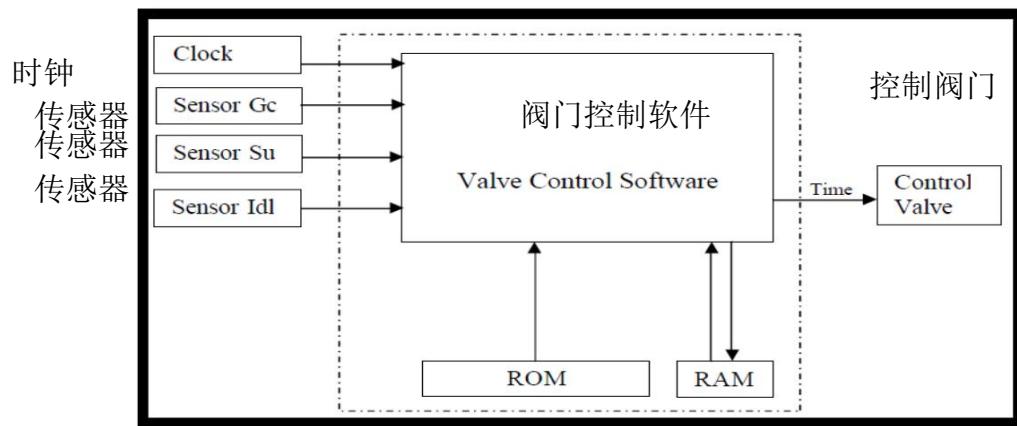


Fig. 12 Block diagram of the hardware and software components of the Valve Control case study in [52]

阀门控制硬件和软件的研究学习

操作系统服务的独立性

存储服务的独立性

网络服务的独立性

网络服务的独立性

网络服务的独立性

软件系统调用服务的隔离性

R1与R2的中间服务

R2与R3的中间服务

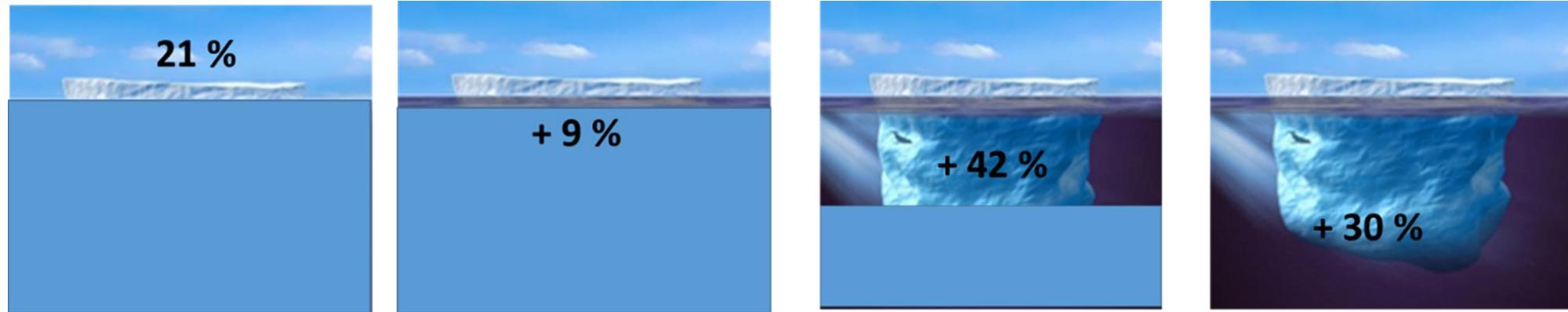
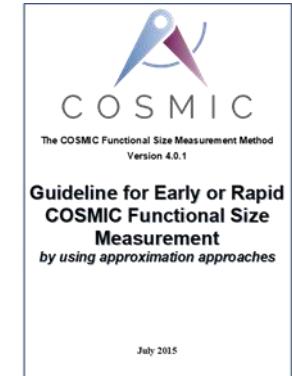
Table 18

The measurement details for the system portability requirements allocated to software functions

按照标准的软件FUR可移植性功能

ID	Standards-based Software-FUR for portability functions	R	Data Movements identified				识别的数据移动 Size CFP
			E	X	R	W	
1	Independence of the operating system service	R1	-	-	1	1	2
2	Independence of the storage services	R2	-	-	1	1	2
3	Independence of the network services	R3	-	-	1	1	2
4	Independence of the network services	R3	-	-	1	1	2
5	Independence of the network services	R3	-	-	1	1	2
6	Isolation of the software system call services	R4	-	-	1	1	2
7	One Intermediary Service between R1(in function type 1) and R4 (in function type 4)-See Fig. 14.	R1 & R4	4	4	-	-	8
8	One Intermediary Service between R2 and R3(in function type 3) and R4 (in function type 4)-See Fig. 14.	R2,R3 & R4	4	4	-	-	8
Functional Size			8	8	6	6	28 CFP

COSMIC Course Registration case study 案例研究



Systems Functions
系统功能

Details of Business Functions
业务功能的详细程度

Operational Functions
操作功能

Quality Functions
质量功能

Figure 10: CRS case study - Scaling factors based of the size of functional processes.

Summary 总结

Which software estimation techniques can you trust and which ones can provide you with a long-term competitive advantage? 可以信任哪些软件估算技术？哪些技术可以为你提供长期的竞争优势？

Talk approach: 演讲内容

Look at some of the widely used estimation techniques to provide insights to figure out:

- ❖ **which ones are merely ‘placebos’ based on ‘fake science’,**
- ✓ **which ones are based on best professional practices in this field.**

介绍一些广泛使用的估算技术，并为你提供一些鉴别方法，
以识别哪些技术只是基于“伪科学”的“安慰剂”，而哪些技术才是基于领域内的
最佳专业实践。

Summary 总结

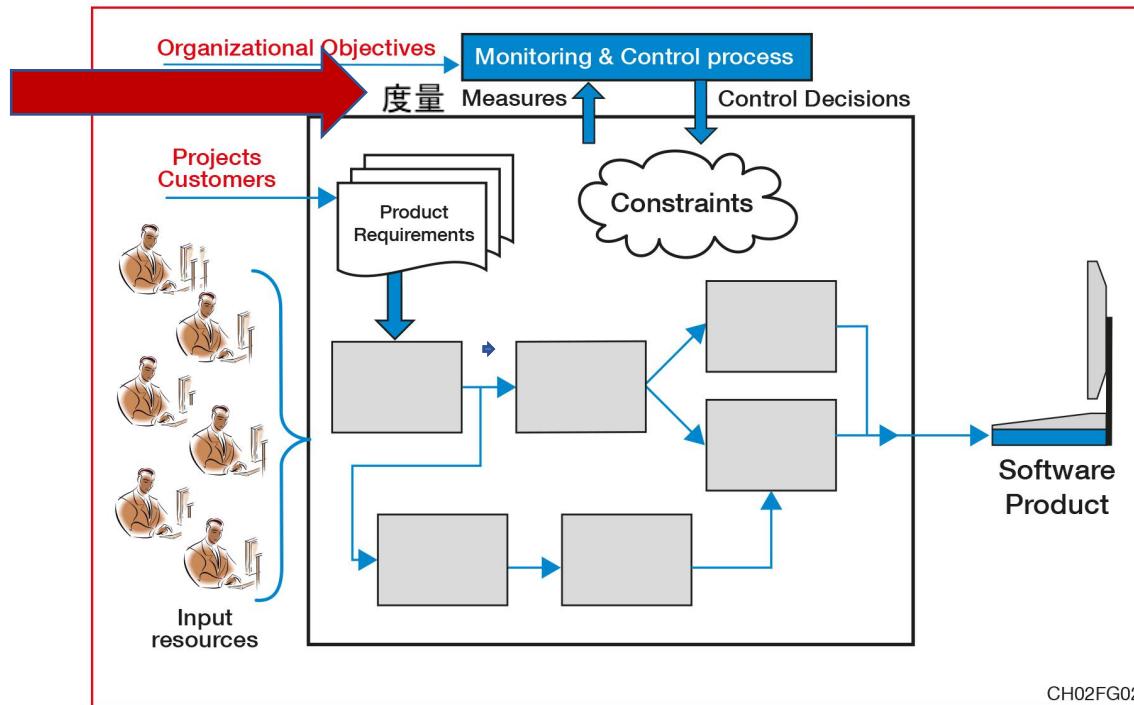
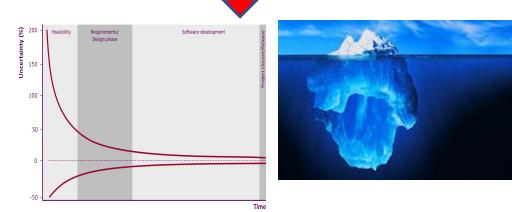
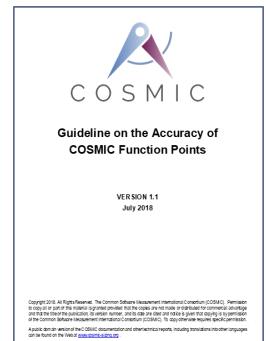
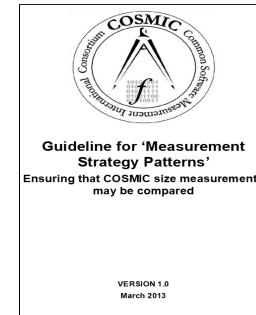
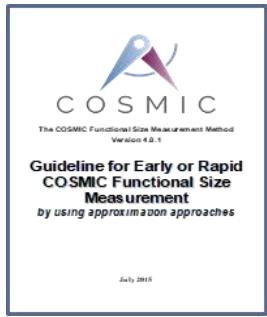
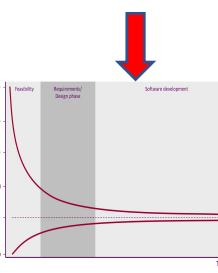
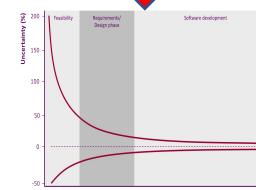
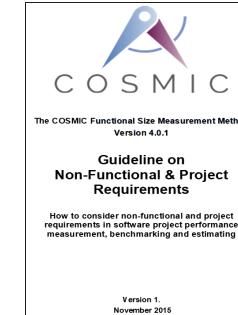
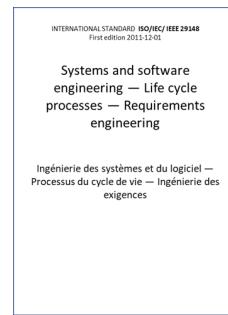
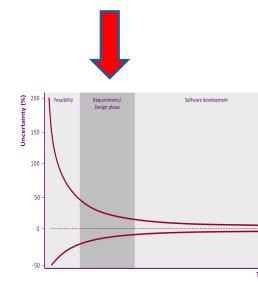
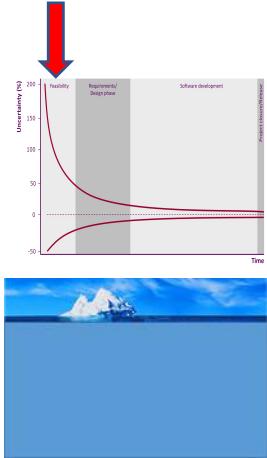


Figure 2.2 A production Process -
the engineering & management perspective.



COSMIC Guidelines for Early Sizing

COSMIC 早期规模度量指南



Summary 总结

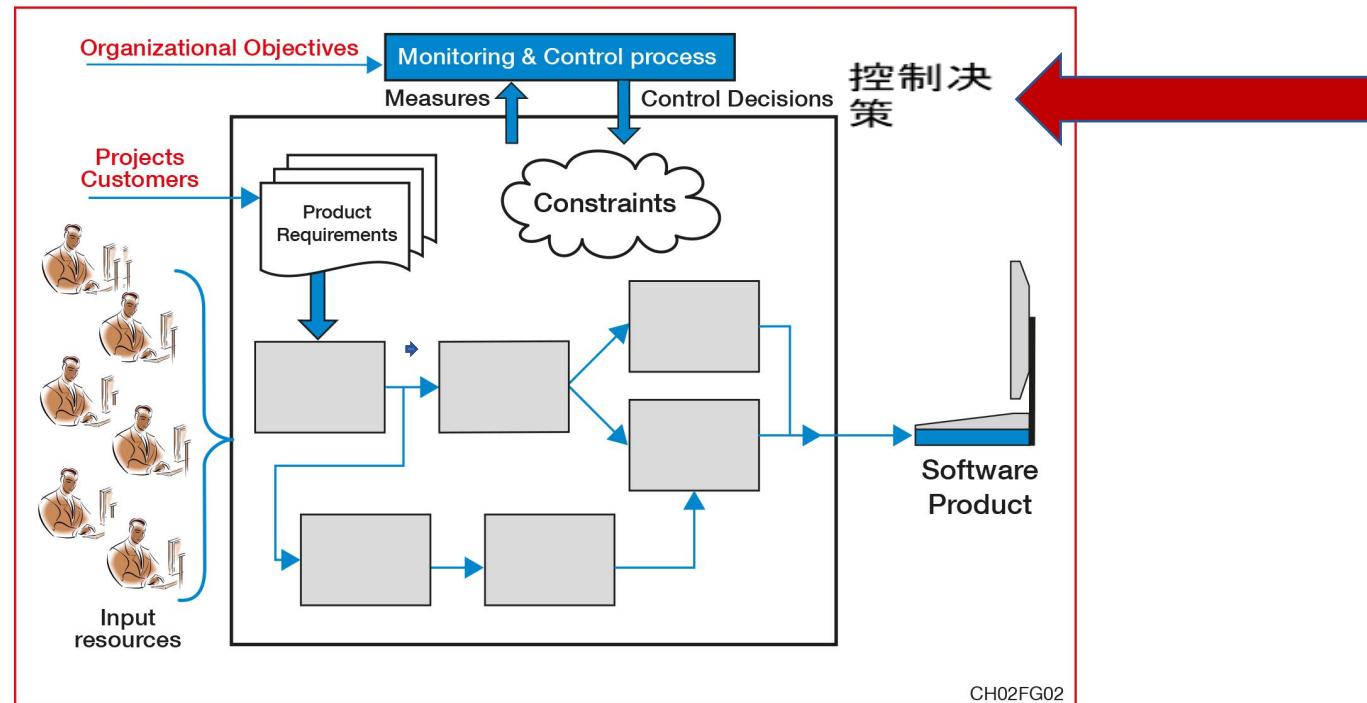


Figure 2.2 A production Process -
the engineering & management perspective.



Estimation: 估算: 不能一步搞定
not a 1-step technical task

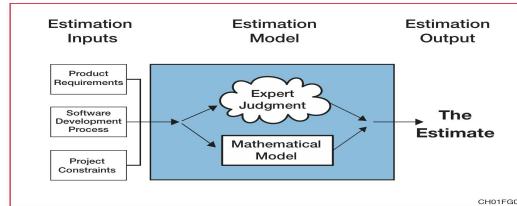
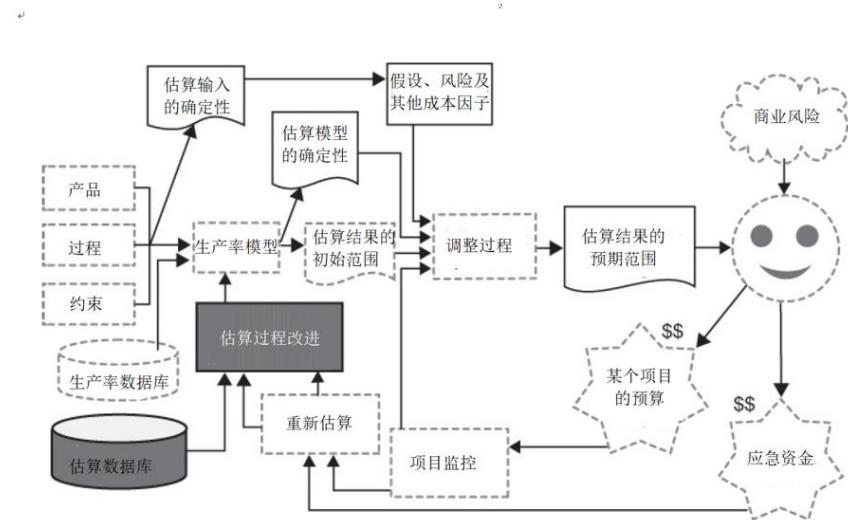
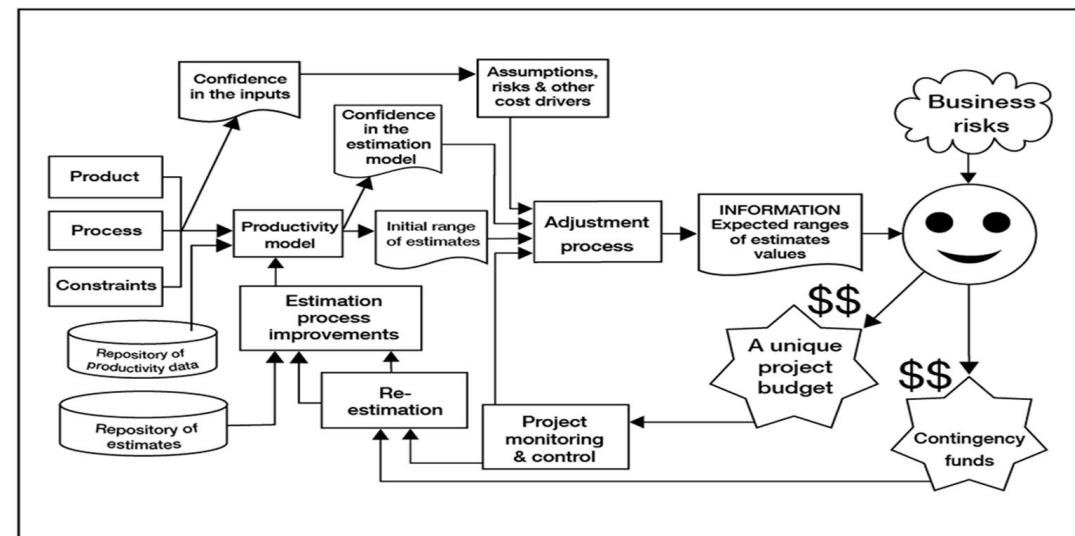


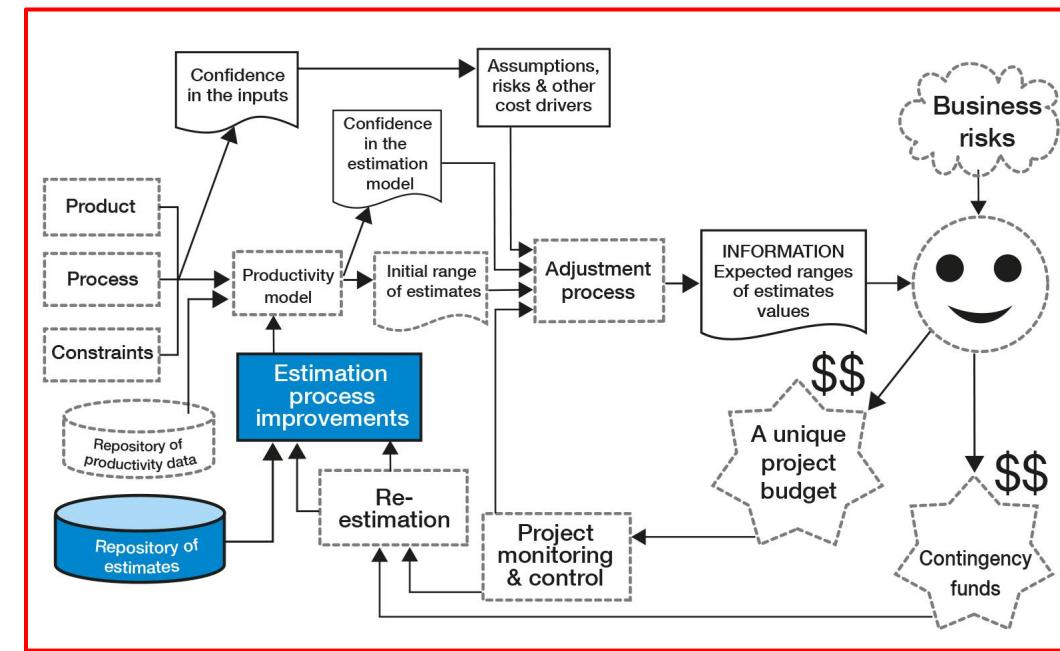
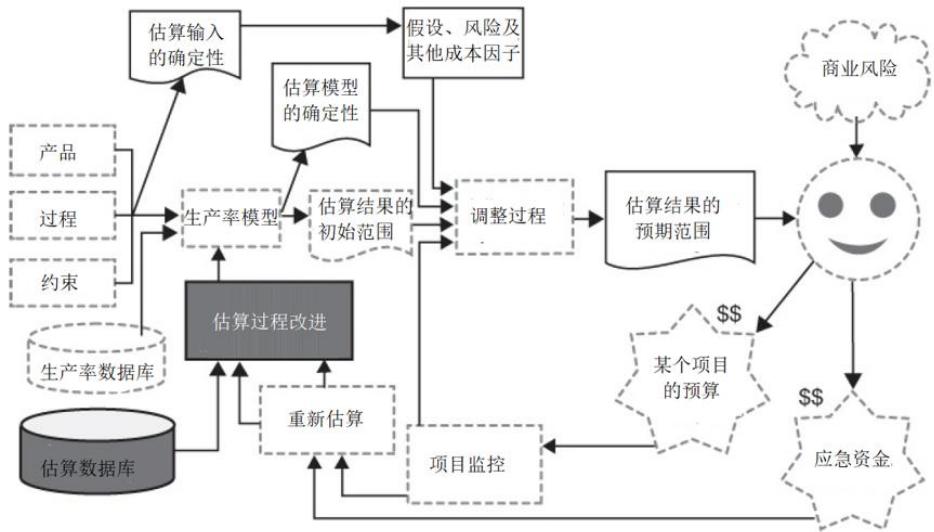
Figure 1.1 Common view of an estimation process.



Estimation: 估算: 全周期的管理流程
A full cycle management process



Estimation Process Improvement 估算流程改进



Summary 总结

Upcoming certifications: 即将开展的认证:

1. ICEAA: Software Cost Estimation – A broad view

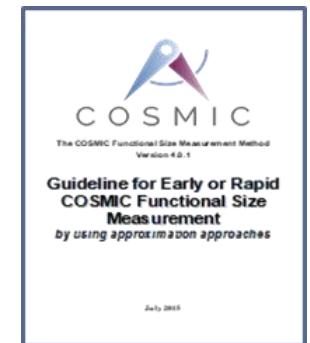
(International Costs Estimation & Analysis Association)

软件成本估算-更广泛



2. COSMIC – Size Estimation – A specialized view

COSMIC规模估算-更专业



**Thank you
&
Questions?**
