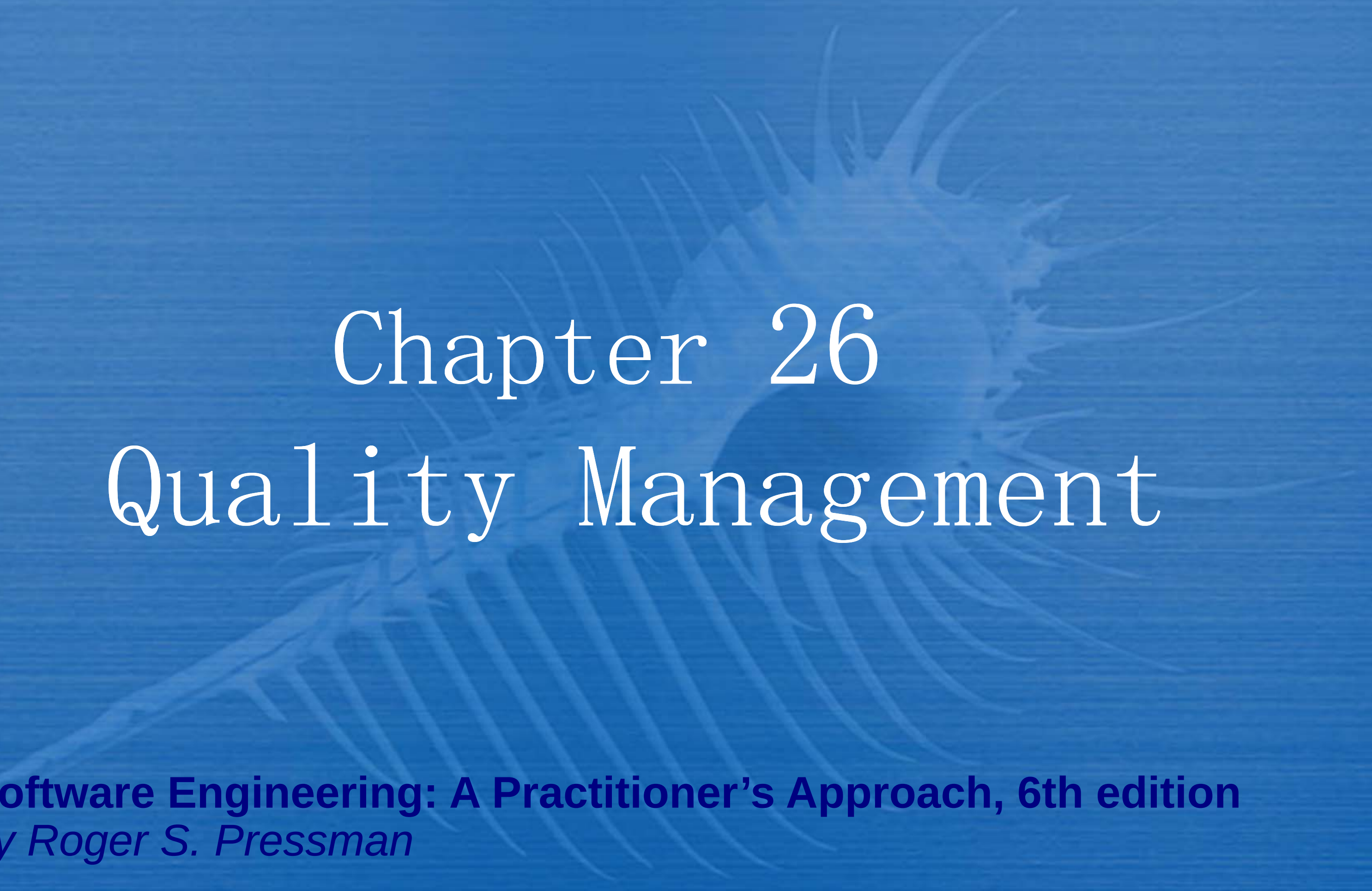


Chapter 26
Quality Management

Software Engineering: A Practitioner’s Approach, 6th edition
by Roger S. Pressman

本章要点

- 质量成本
- 质量保证

Quality

- The American Heritage Dictionary[美国传统大辞典] defines quality as
  - “a characteristic or attribute of something.”
- For software, two kinds of quality may be encountered:
  - Quality of design encompasses requirements, specifications, and the design of the system.
  - Quality of conformance is an issue focused primarily on implementation.
- user satisfaction = compliant product[合格产品] + good quality + delivery within budget and schedule

Software Quality

Conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software.

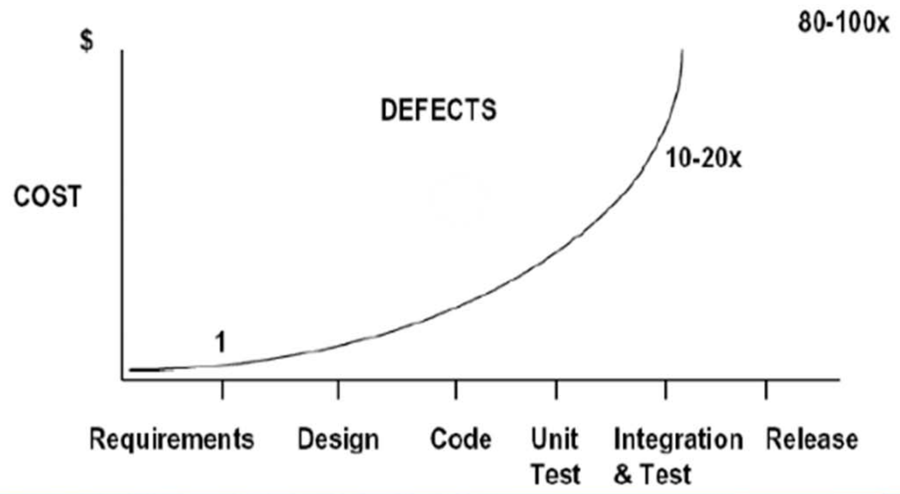
质量控制

- 质量成本
- 质量保证

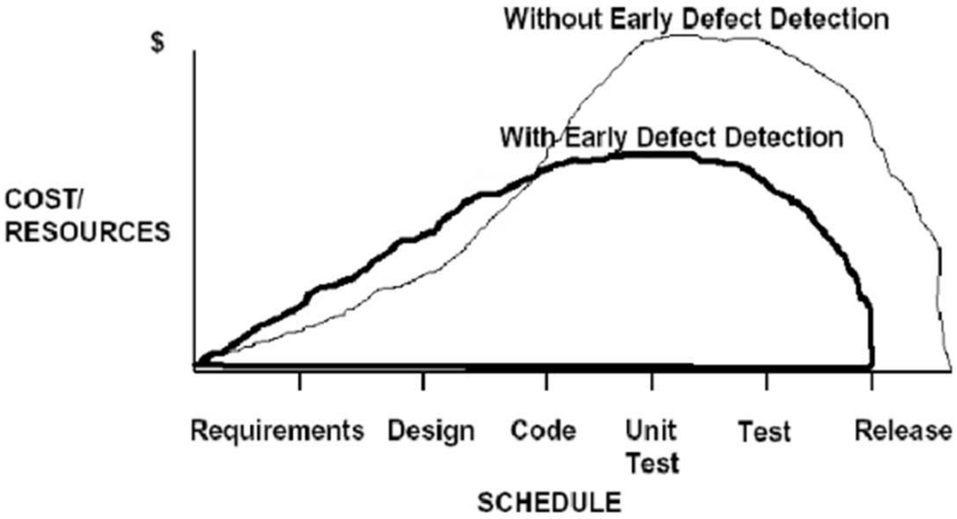
Cost of Quality ⇨

- Prevention costs include
  - quality planning
  - formal technical reviews
  - test equipment
  - Training
- Internal failure costs include
  - rework
  - repair
  - failure mode analysis
- External failure costs are
  - complaint resolution[抱怨]
  - product return and replacement
  - help line support [热线电话]
  - warranty work [保障工作]

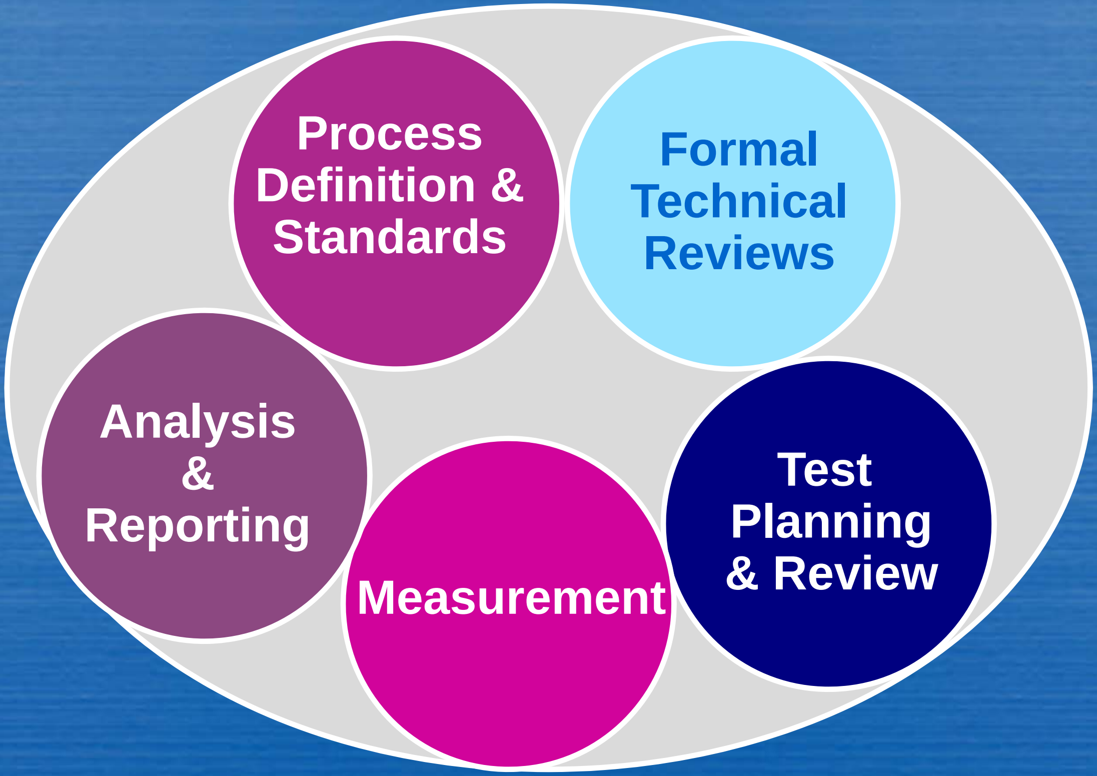
质量代价比率变化



“早”发现“早”收益



软件质量保证（SQA）


Process Definition & Standards
Formal Technical Reviews
Analysis & Reporting
Measurement
Test Planning & Review

Role of the SQA Group

- **Prepares an SQA plan for a project.**
  - The plan identifies
    - evaluations to be performed
    - audits and reviews to be performed
    - standards that are applicable to the project
    - procedures for error reporting and tracking
    - documents to be produced by the SQA group
    - amount of feedback provided to the software project team
- **Participates in the development of the project’s software process description.**
  - The SQA group reviews the process description for compliance with[符合] organizational policy, internal software standards, externally imposed standards (e.g., ISO-9001), and other parts of the software project plan.

Role of the SQA Group…

- **Reviews software engineering activities to verify compliance with the defined software process.**
  - identifies, documents, and tracks deviation[偏差] from the process and verifies that corrections have been made.
- **Audits designated software work products to verify compliance with those defined as part of the software process.**
  - reviews selected work products; identifies, documents, and tracks deviations; verifies that corrections have been made
  - periodically reports the results of its work to the project manager.
- **Ensures that deviations in software work and work products are documented and handled according to a documented procedure.**
- **Records any noncompliance[违规] and reports to senior management.**
  - Noncompliance items are tracked until they are resolved.

SQA各种方法

- 评审/正式技术评审[FTR]
- SQA的形式化方法
- 基于统计的软件质量保证
- 软件可靠性度和安全性评估
- ISO 9000质量标准
- ……

Reviews & Inspections

... there is no particular reason why your friend and colleague cannot also be your sternest critic [不苟言笑评论家].

Jerry Weinberg

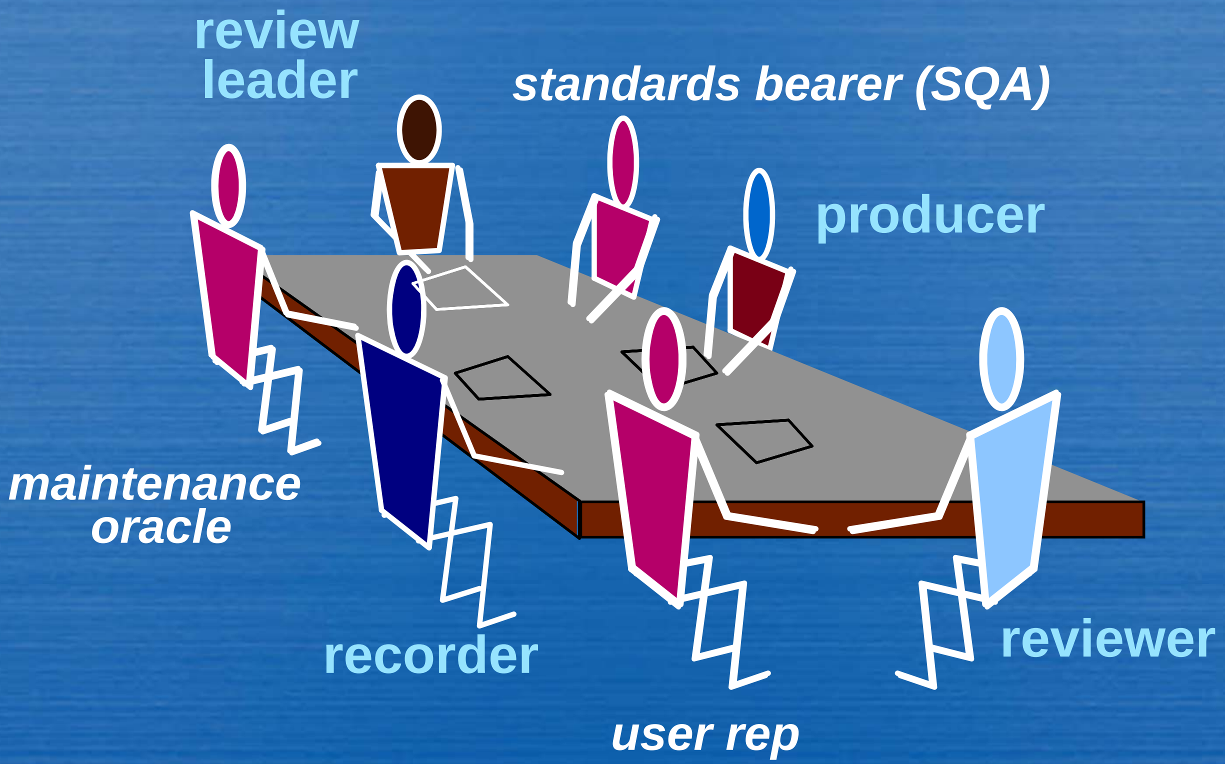
What Are Reviews?

- **a meeting** conducted by technical people for technical people
- **a technical assessment** of a work product created during the software engineering process
- a software quality assurance **mechanism**
- **a training ground**

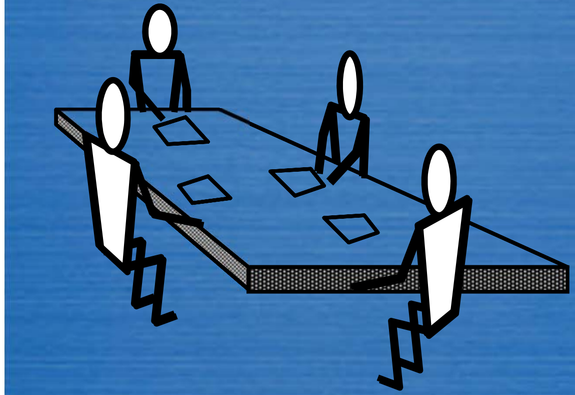
What Reviews Are Not

- A project summary or progress assessment
- A meeting intended solely to impart[透露] information
- A mechanism for political or personal reprisal[报复]!

The Players


review leader
standards bearer (SQA)
producer
maintenance oracle
recorder
reviewer
user rep

Conducting the Review



1. be prepared—evaluate product before the review
2. review the product, not the producer
3. keep your tone mild[适度的], ask questions instead of making accusations[谴责]
4. stick to the review agenda
5. raise issues, don't resolve them
6. avoid discussions of style—stick to technical correctness
7. schedule reviews as project tasks
8. record and report all review results

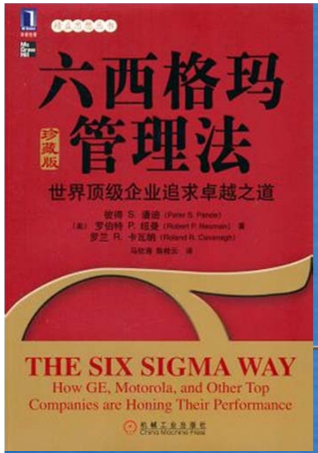
Review Options Matrix

	IPR*	WT	IN	RRR
trained leader	no	yes	yes	yes
agenda established	maybe	yes	yes	yes
reviewers prepare in advance	maybe	yes	yes	yes
producer presents product	maybe	yes	no	no
“reader” presents product	no	no	yes	no
recorder takes notes	maybe	yes	yes	yes
checklists used to find errors	no	no	yes	no
errors categorized as found	no	no	yes	no
issues list created	no	yes	yes	yes
team must sign-off on result	no	yes	yes	maybe

*IPR—informal peer review WT—Walkthrough
IN—Inspection RRR—round robin review

Metrics Derived from Reviews

- inspection time per page of documentation
- inspection time per KLOC or FP
- inspection effort per KLOC or FP
- errors uncovered per reviewer hour
- errors uncovered per preparation hour
- errors uncovered per SE task (e.g., design)
- number of minor errors (e.g., typos)
- number of major errors
  (e.g., nonconformance to req.)
- number of errors found during preparation


六西格玛管理法
珍藏版
世界顶级企业追求卓越之道
彼得 S. 潘迪 (Peter S. Pande)
(美) 罗伯特 P. 纽曼 (Robert P. Neuman) 著
罗兰 R. 卡瓦纳 (Roland R. Cavanagh)
THE SIX SIGMA WAY
How GE, Motorola, and Other Top Companies are Honing Their Performance
机械工业出版社
China Machine Press

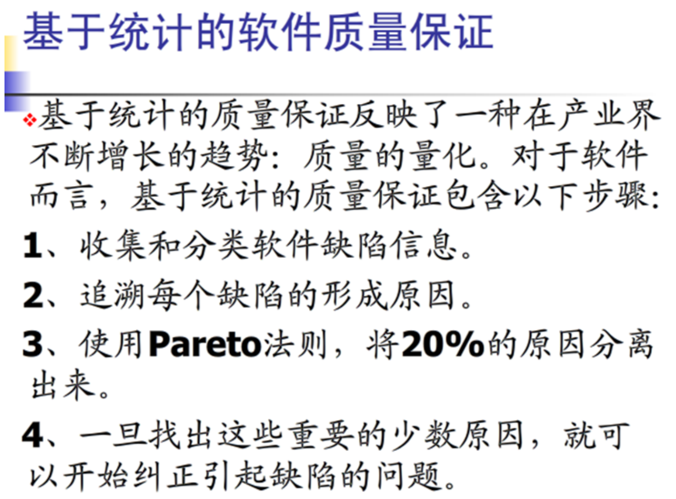
Statistical SQA

Product & Process

Collect information on all defects
Find the causes of the defects
Move to provide fixes for the process

measurement

... an understanding of how to improve quality ...



基于统计的软件质量保证

- 基于统计的质量保证反映了一种在产业界不断增长的趋势：质量的量化。对于软件而言，基于统计的质量保证包含以下步骤：

1、收集和分类软件缺陷信息。

2、追溯每个缺陷的形成原因。

3、使用**Pareto**法则，将**20%**的原因分离出来。

4、一旦找出这些重要的少数原因，就可以开始纠正引起缺陷的问题。

Software Reliability

- A simple measure of reliability is *mean-time-between-failure* (MTBF), where

$$\text{MTBF} = \text{MTTF} + \text{MTTR}$$

- The acronyms MTTF and MTTR are *mean-time-to-failure* and *mean-time-to-repair*, respectively.
- *Software availability* is the probability that a program is operating according to requirements at a given point in time and is defined as

$$\text{Availability} = [\text{MTTF}/(\text{MTTF} + \text{MTTR})] \times 100\%$$

Software Safety

- Software safety is a software quality assurance activity that focuses on the identification and assessment of potential hazards[潜在危险] that may affect software negatively and cause an entire system to fail.
- If hazards can be identified early in the software process, software design features can be specified that will either eliminate or control potential hazards.

ISO 9001:2000 Standard

- ISO 9001:2000 is the **quality assurance standard** that applies to software engineering.
- The standard contains 20 requirements that must be present for an effective quality assurance system.
- The requirements delineated by ISO 9001:2000 address topics such as
  - management responsibility, quality system, contract review, design control, document and data control, product identification and traceability, process control, inspection and testing, corrective and preventive action, control of quality records, internal quality audits, training, servicing, and statistical techniques.

Chapter 27
Change Management

Software Engineering: A Practitioner’s Approach, 6th edition
by Roger S. Pressman

本章要点

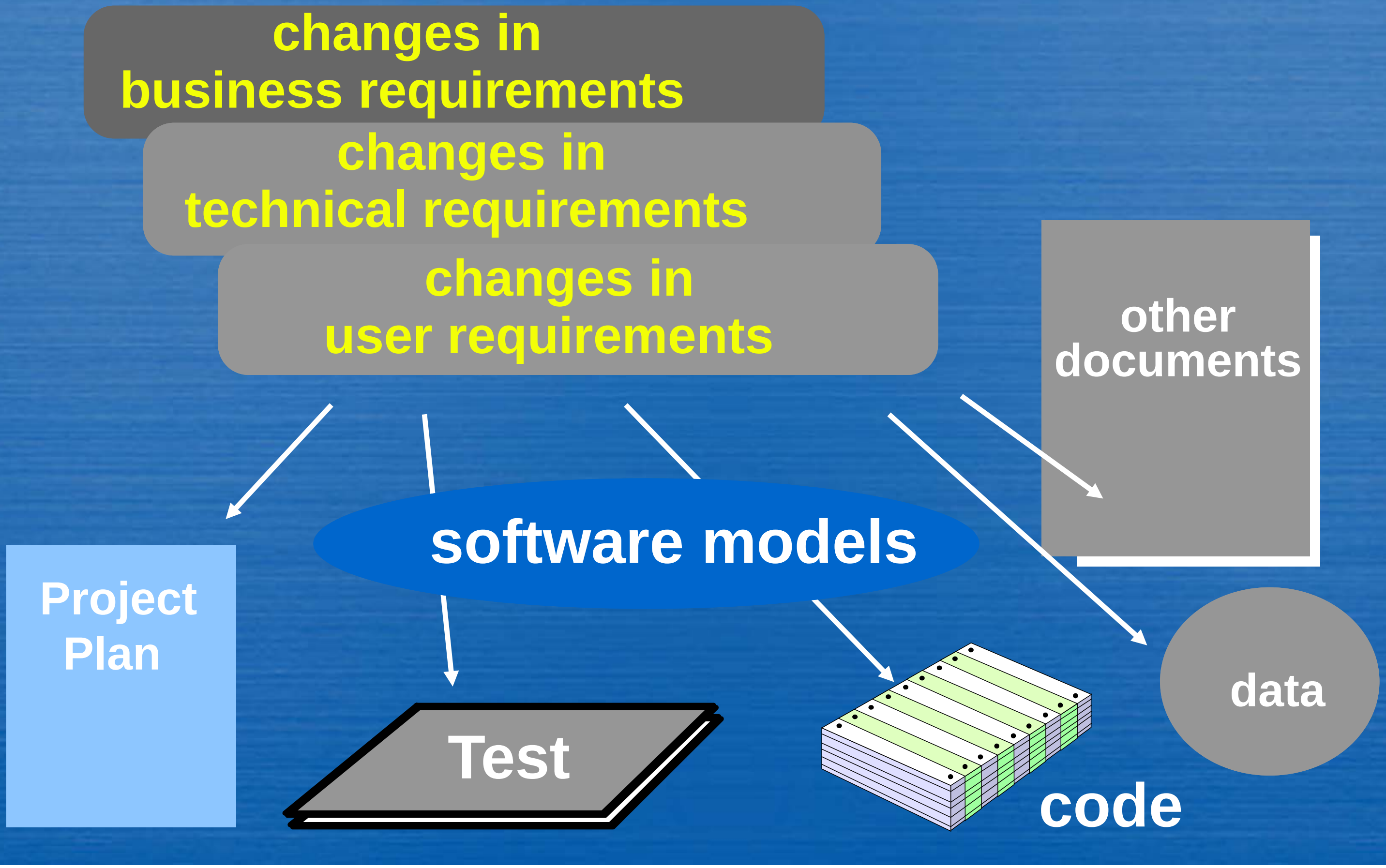
- 软件配置管理
- 版本控制
- 变更控制

The "First Law"

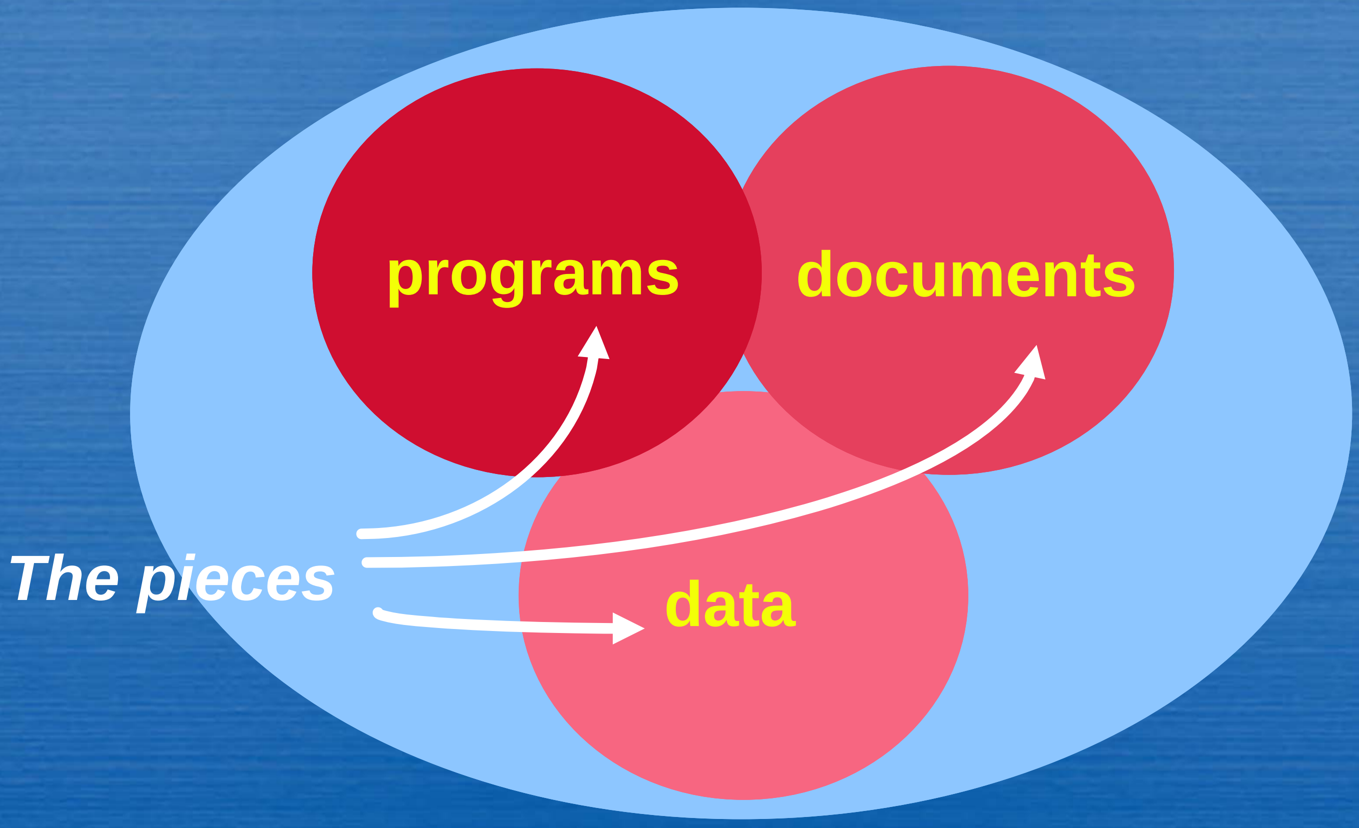
No matter where you are in the system life cycle, the system will change, and the desire to change it will persist throughout the life cycle.

Bersoff, et al, 1980

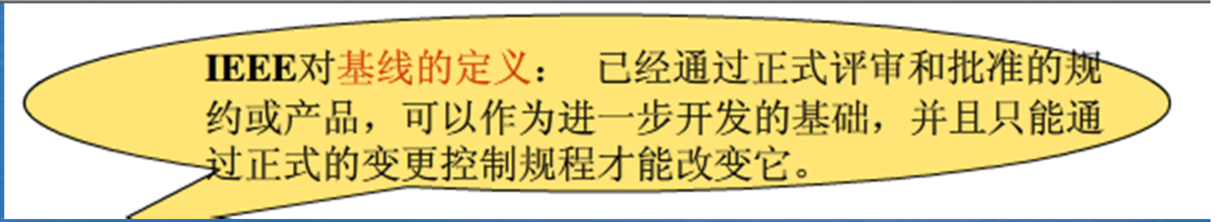
What Are These Changes?


changes in business requirements
changes in technical requirements
changes in user requirements
other documents
software models
Project Plan
Test
data
code

The Software Configuration



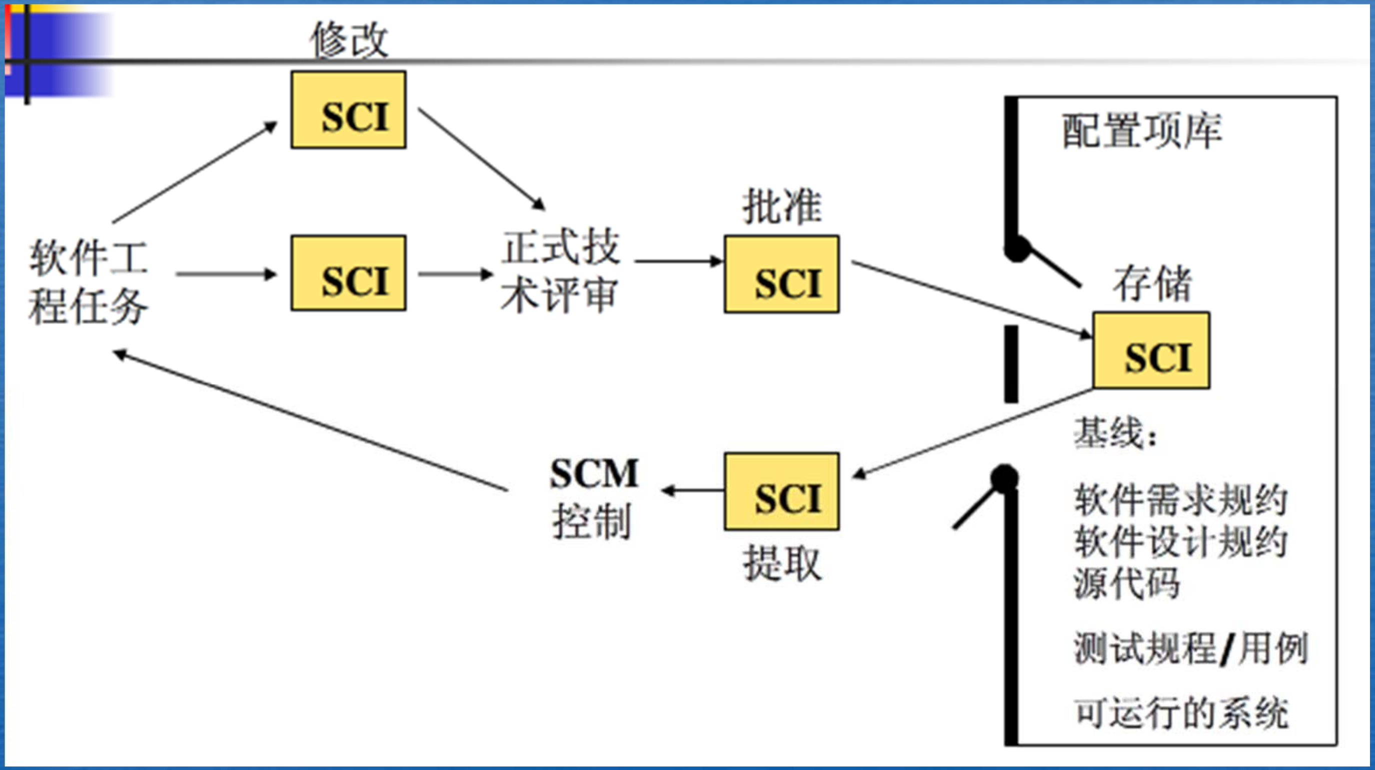
Baselines



IEEE对基线的定义： 已经通过正式评审和批准的规约或产品，可以作为进一步开发的基础，并且只能通过正式的变更控制规程才能改变它。

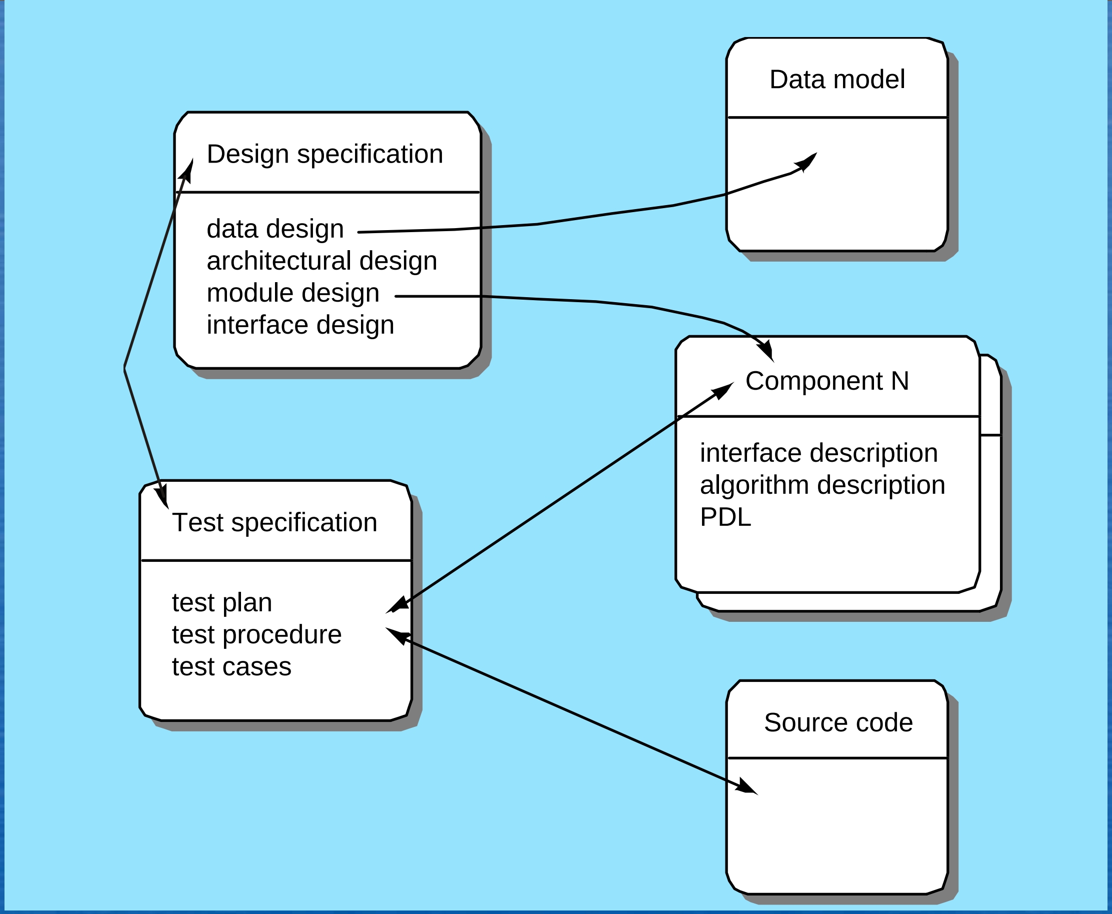
- The IEEE (IEEE Std. No. 610.12-1990) defines a **baseline** as:
  - **A specification or product** that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.
- a baseline is a milestone in the development of software that is marked by the delivery of one or more software configuration items (SCIs) and the approval of these SCIs that is obtained through a formal technical review

Baselines…



基线化的SCI和项目数据库

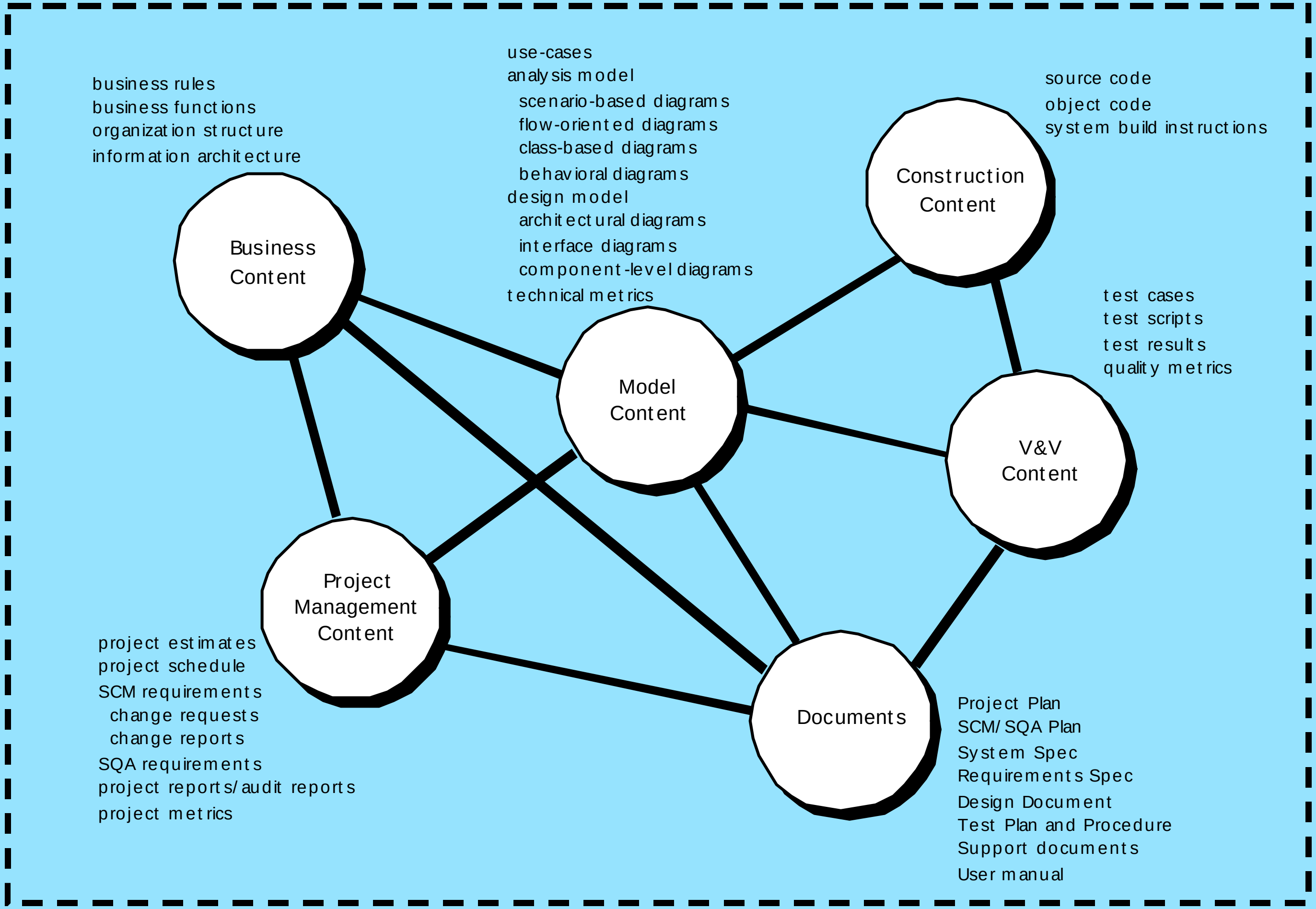
Software Configuration Objects



SCM Repository

- The SCM repository [知识库] is the set of **mechanisms** and **data structures** that allow a software team to manage change in an effective manner
- The repository performs or precipitates[沉淀] the following functions [FOR89]:
  - Data integrity
  - Information sharing
  - Tool integration
  - Data integration
  - Methodology enforcement
  - Document standardization

Repository Content



Repository Features

- **Versioning**: saves all of these versions to enable effective management of product releases and to permit developers to go back to previous versions
- **Dependency tracking and change management**: The repository manages a wide variety of relationships among the data elements stored in it.
- **Requirements tracing**: Provides the ability to track all the design and construction components and deliverables that result from a specific requirement specification
- **Configuration management**: Keeps track of a series of configurations representing specific project milestones or production releases. Version management provides the needed versions, and link management keeps track of interdependencies.
- **Audit trails**[查账索引]: establishes additional information about when, why, and by whom changes are made.

SCM Elements

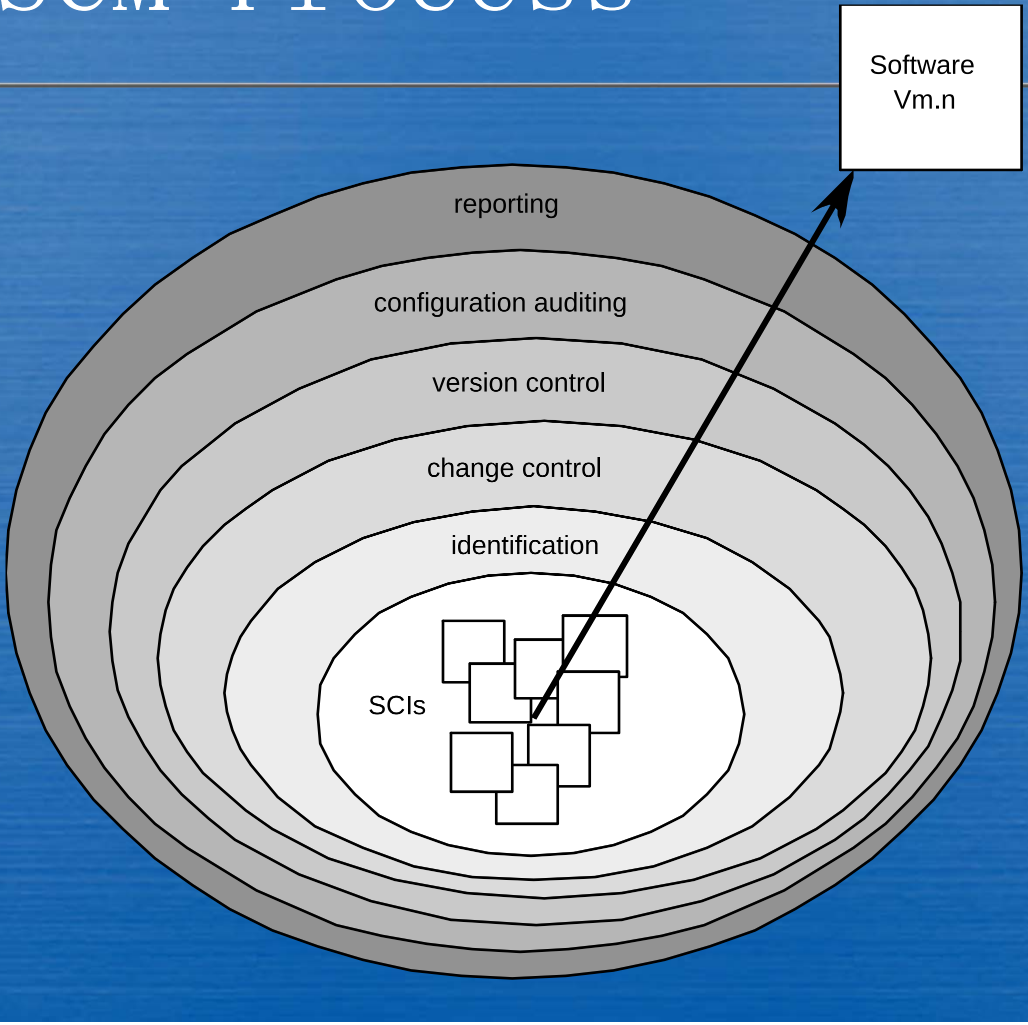
- **Component elements**—a set of tools coupled within a file management system (e.g., a database) that enables access to and management of each software configuration item.
- **Process elements**—a collection of procedures and tasks that define an effective approach to change management (and related activities) for all constituencies involved in the management, engineering and use of computer software.
- **Construction elements**—a set of tools that automate the construction of software by ensuring that the proper set of validated components (i.e., the correct version) have been assembled.
- **Human elements**—to implement effective SCM, the software team uses a set of tools and process features (encompassing other CM elements)

The SCM Process

Addresses the following questions …

- How does a software team identify the discrete elements of a software configuration?
- How does an organization manage the many existing versions of a program (and its documentation) in a manner that will enable change to be accommodated efficiently[有效容纳]?
- How does an organization control changes before and after software is released to a customer?
- Who has responsibility for approving and ranking changes?
- How can we ensure that changes have been made properly?
- What mechanism is used to appraise[评价] others of changes that are made?

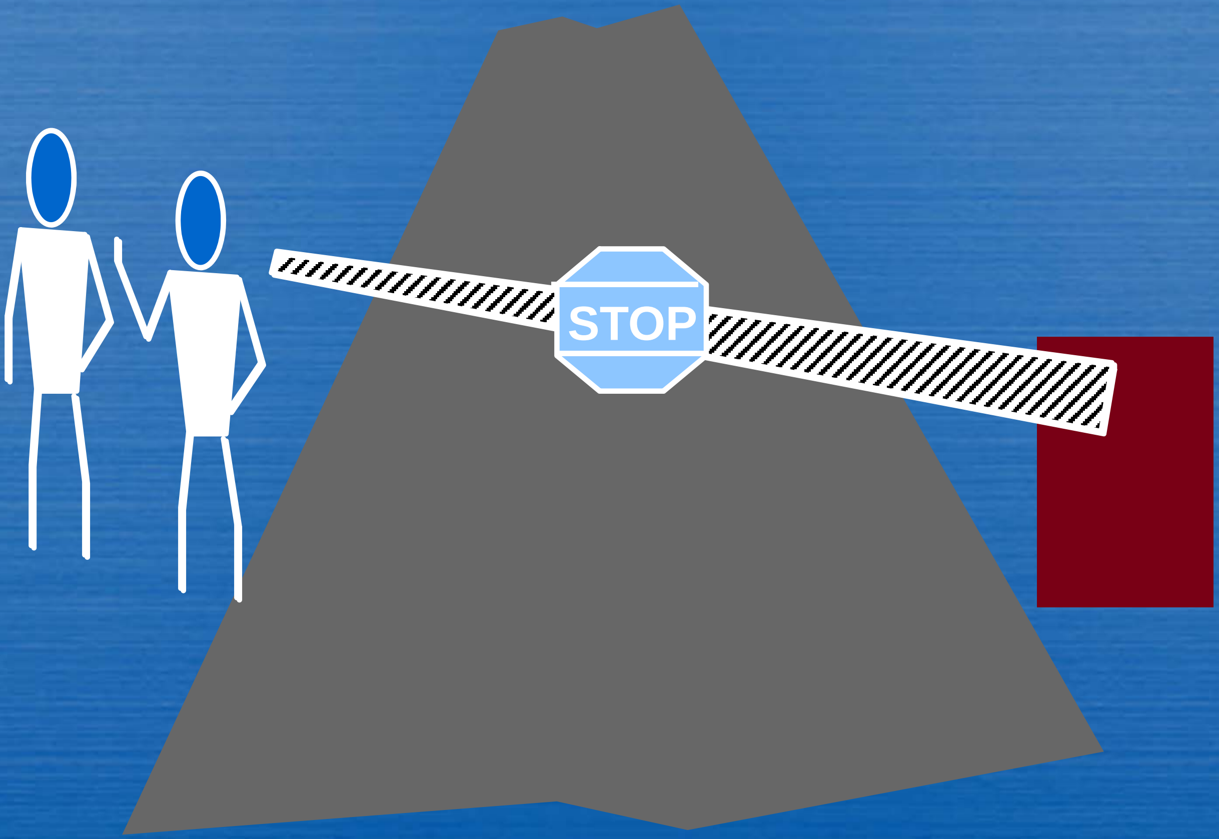
The SCM Process



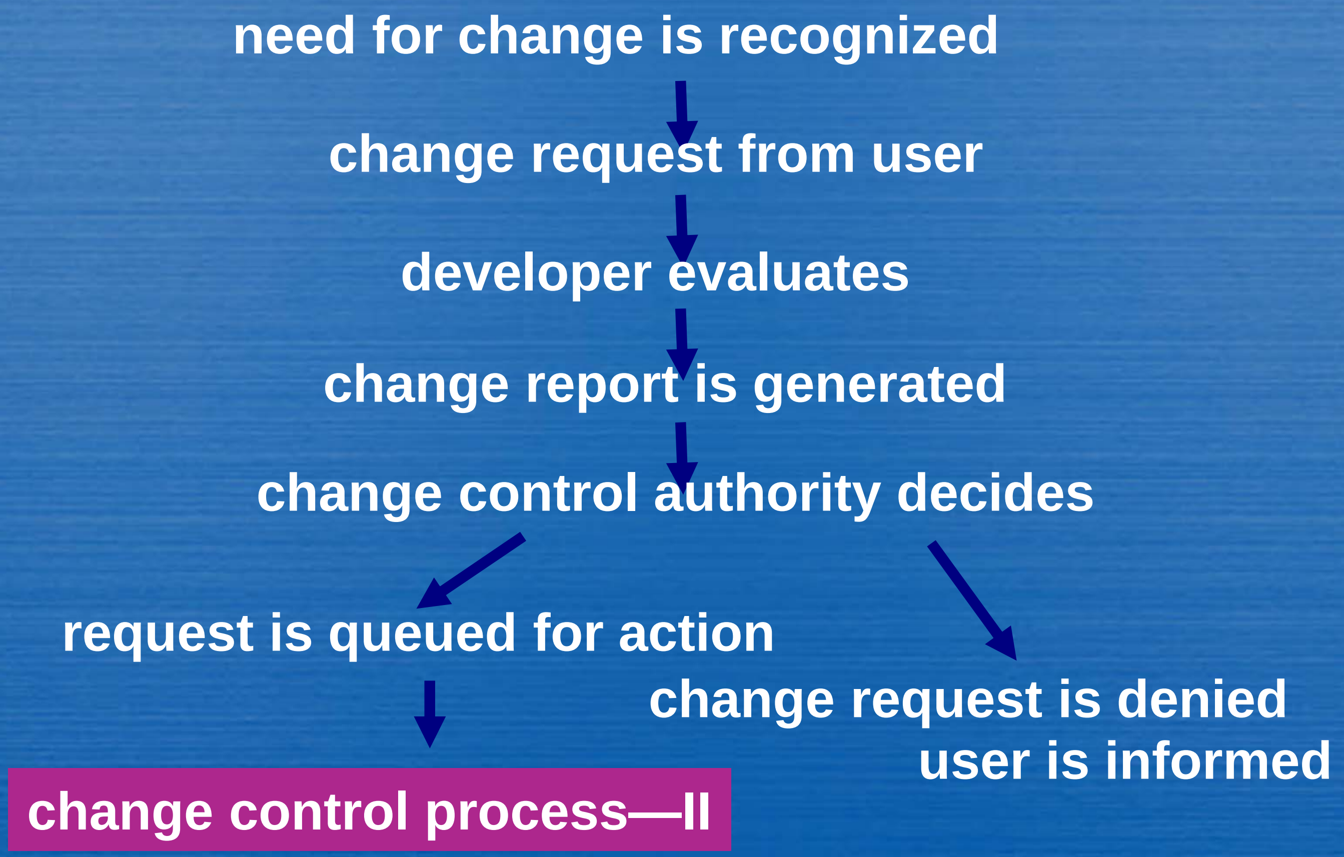
Version Control

- Version control combines procedures and tools to manage different versions of configuration objects that are created during the software process
- A version control system implements or is directly integrated with four major capabilities:
  - **a project database (repository)** that stores all relevant configuration objects
  - **a version management capability** that stores all versions of a configuration object (or enables any version to be constructed using differences from past versions);
  - **a make facility** that enables the software engineer to collect all relevant configuration objects and construct a specific version of the software.
  - **an issues tracking** (also called bug tracking) capability that enables the team to record and track the status of all outstanding issues associated with each configuration object.

Change Control



Change Control Process—I



need for change is recognized

change request from user

developer evaluates

change report is generated

change control authority decides

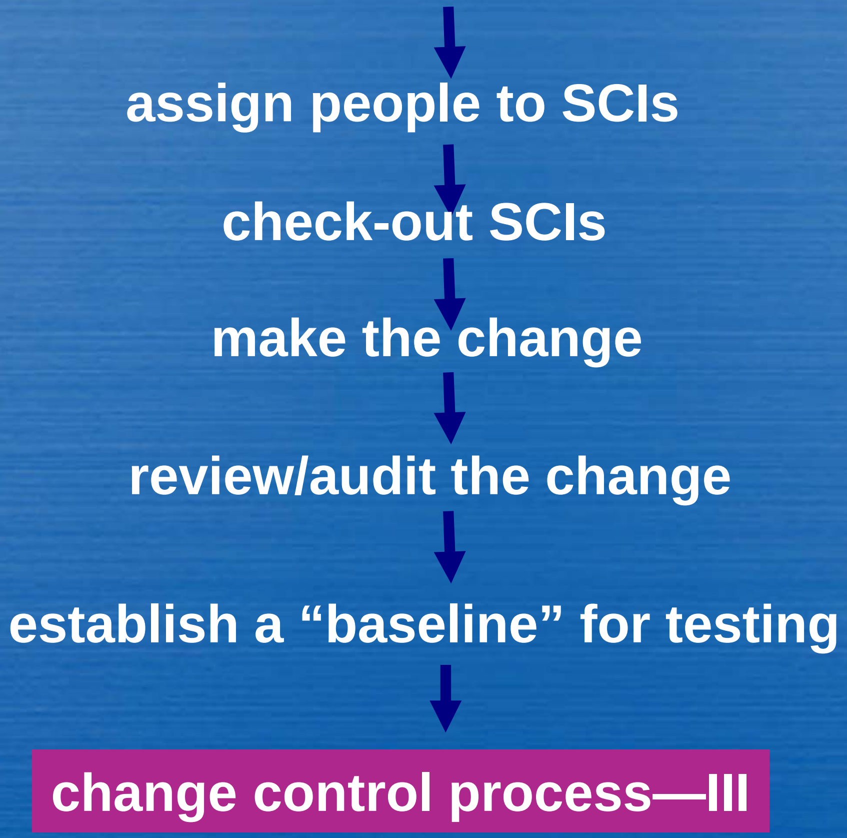
request is queued for action

change request is denied

user is informed

change control process—II

Change Control Process-II



assign people to SCIs

check-out SCIs

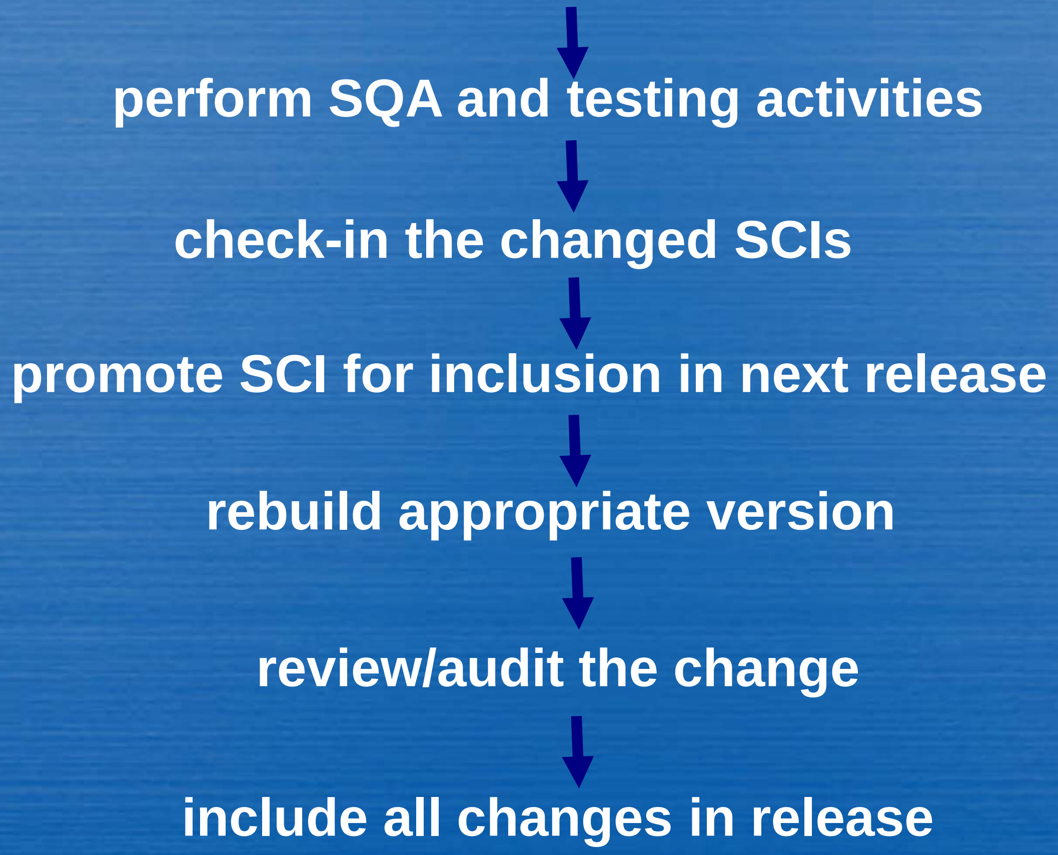
make the change

review/audit the change

establish a “baseline” for testing

change control process—III

Change Control Process-III



perform SQA and testing activities

check-in the changed SCIs

promote SCI for inclusion in next release

rebuild appropriate version

review/audit the change

include all changes in release