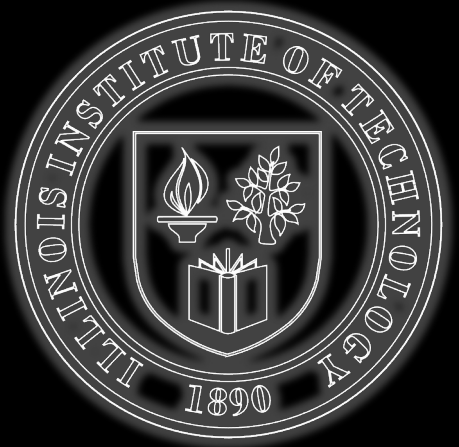
ILLINOIS TECH | College of Computing

ITMD 536 Software Testing & Maintenance

Chapter 1 and 2
Maintenance Is Everybody's Primary Business &
Software Maintenance Overview

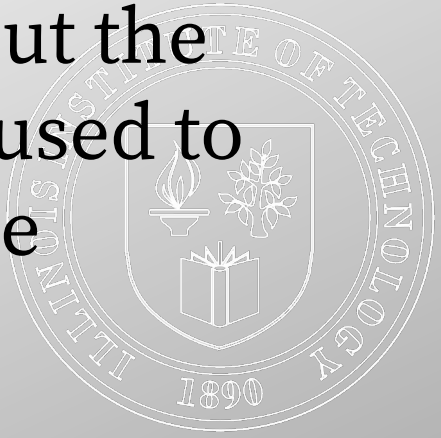


Objectives

- What is Software Maintenance?
- Who does software maintenance?
- How can you transition from development to maintenance mode?
- What is required to do emergency fixes?
- How do you structure effective regression testing

Software Maintenance

- **Maintenance Process:** The set of processes used to maintain and sustain software after it has been delivered. These processes involve more than just development. For example, they may address distribution control and adaptation to make the software work at different sites.
- **Maintenance Plan**: Refers to a document that sets out the activities', schedules, practices and resources to be used to maintain a software product and generate a software release.



1.1 Software Maintenance

US has 32% of software developers are in different industries.

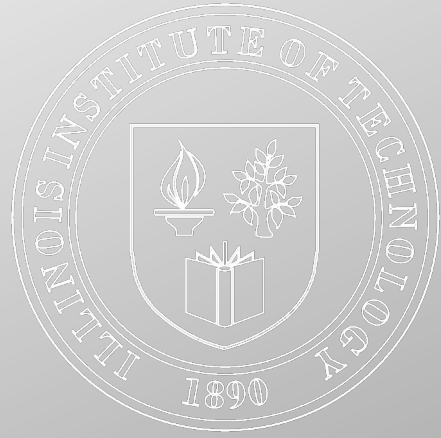
Remaining percentage are involved in software maintenance tasks.

There are large number of computer programmers are self employed.

Lower prevailing wages and higher educated workers are in India. Over 6 to 7 billion software projects are offshored

1.2 Goals and Scope

- Refers to the process of keeping a product current after delivery. This involves updating the product to address new functionality and repairs and sustaining the facilities and infrastructure required to accomplish this tasks.



1.3 Maintenance Viewpoints

1. What work is involved?

   While changes and fixes are performed, at least half the work involves other activities like sustaining engineering and field, user and product support.

2. What does the work?

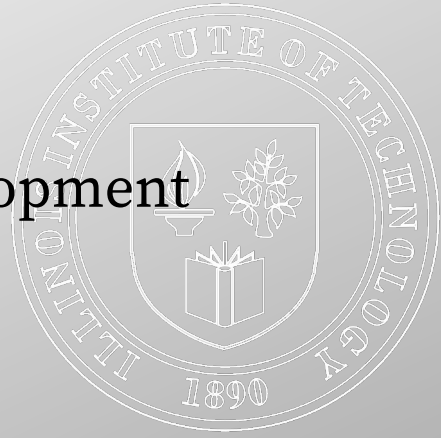
   Teams with the lease half the workforce being senior and experienced

. What is the work done?

   During both the development (for increments) and post -delivery

4. Where is the work performed?

   At a maintenance facility that provides more capability than the development site (real equipment, users in the loop, etc.); patches are made at the operational site



1.3 Maintenance Viewpoints

5. What activities are typically performed during maintenance?

   As much as 60 percent of the work involves testing; requirements analysis, reengineering, and other tasks consume the remainder

6. What are the primary drivers?

   Backlog of new requirements and bug fixes

7. What are the primary risks?

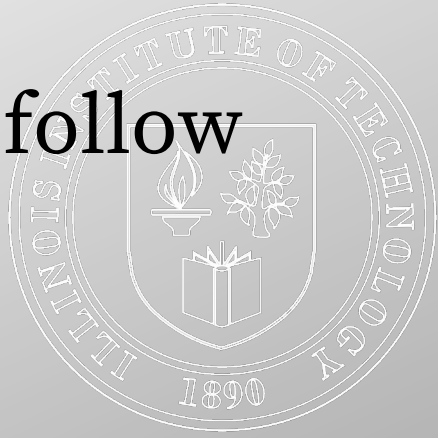
   Unrealistic expectations for efforts done on level-of effort (LOE) basis, inadequate budgets, facility shortfalls, and no visibility into progress

8. How do you measure success?

   Deliver a release or version that performs at least as well as that which it replaces after modifications have been made to it

1.3.1 Product-User needs Requirements

- Enhancements (new features and functionality) and repairs (bug fixes)
- Good architecture is emphasized
- Over 60 to 70 % of maintenance work revolves around software testing
- Regression testing is used to for maintenance to follow the local/national rules, regulations and laws

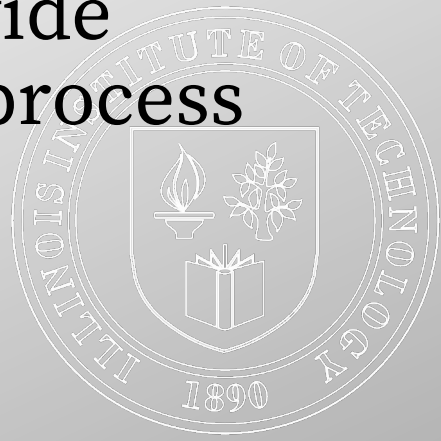


Four Views of Software Maintenance

Product Planned evolutionary paths and renewal cycles Traceability between requirements and product features Well-architected and engineered products Emphasis on testing, retesting, regression testing, and reengineering	**Process** Effective product update, versioning, and release processes Capable including field, user, and product support of sustaining engineering processes Responsive configuration and distribution management procedures Superior acquisition or supply chain management and licensing processes
People Highly skilled and motivated workforces Trained, efficient, and effective workers Interdisciplinary terms Ethics, accountability, and clear responsibility	**Project/Infrastructure** Enlightened leadership Proven project management processes Insightful measurement and control processes Responsible risk management processes

1.3.2 Process evolution & Change Management

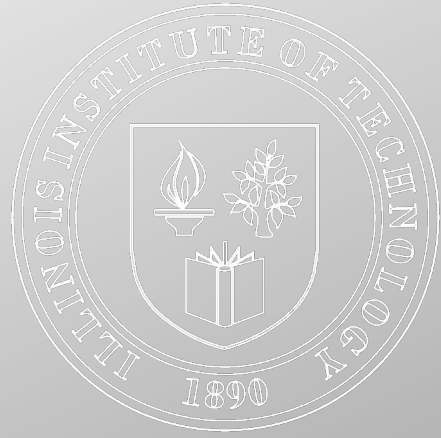
- Software life cycle starts with conceptual of a product and ends with its retirement. The product is developed based on requirements and is architected and designed with change in mind.
- Documentation is required for change management.
- Capability Maturity Model Integration (CMMI) provide adequate structure for implementing a responsive process framework and improvement program.



1.3.3 People - Workforce Needs

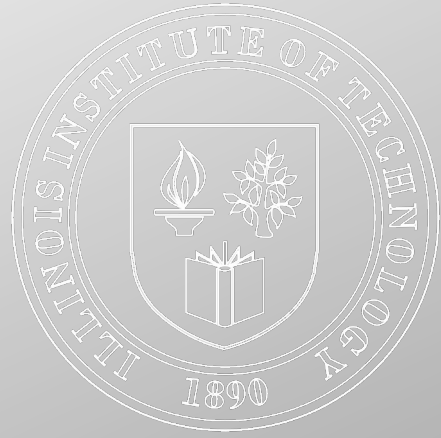
Most experienced person should be assigned for the maintenance tasks.

1. Formulating requirements
2. Development work is more challenging than maintenance
3. Pay, prestige and rewards
4. Easy to recruit and retain developers than maintainers



1.3.4 Project Deadlines and Management

1. Primary purpose – User req - defects
2. Basis – Tech req – Software Change req
3. Schedule – satisfy req – SCR batched
4. Effort – satisfy req – level of effort (LOE)
5. Primary activities – 40/30/30% – 25/20/55%
6. Work tasks – doc, CM, QM, PM –PM&UAT
7. Product – ISS documentation – schedule
8. Management techniques – WBS – Metrics



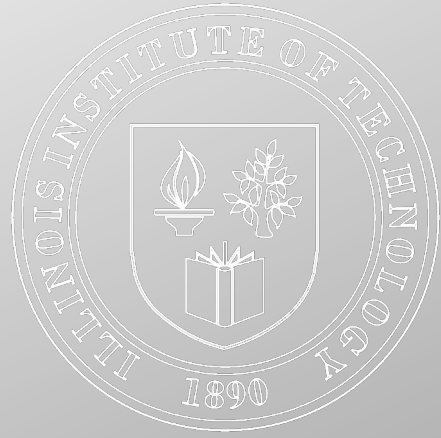
Software Maintenance

- Make sure that testability is designed into products during development because of its impact during software maintenance.
- Ensure that your process infrastructure is expanded to include unique maintenance practices.
- Software maintenance is labor intensive, with the majority of costs arising form programmers' salaries.
- Software under maintenance production system failures occur randomly, and user requests come in on an irregular basis. Without agreed-upon and mature queue-management mechanisms supported by detailed service-level agreement. (SLAs).

Software Maintenance Issues

There are three different maintenance problems:

1. Problems of alignment with the organization's objectives,
2. Process problems, and
3. Technical problems.



Software Maintenance Issues - Process

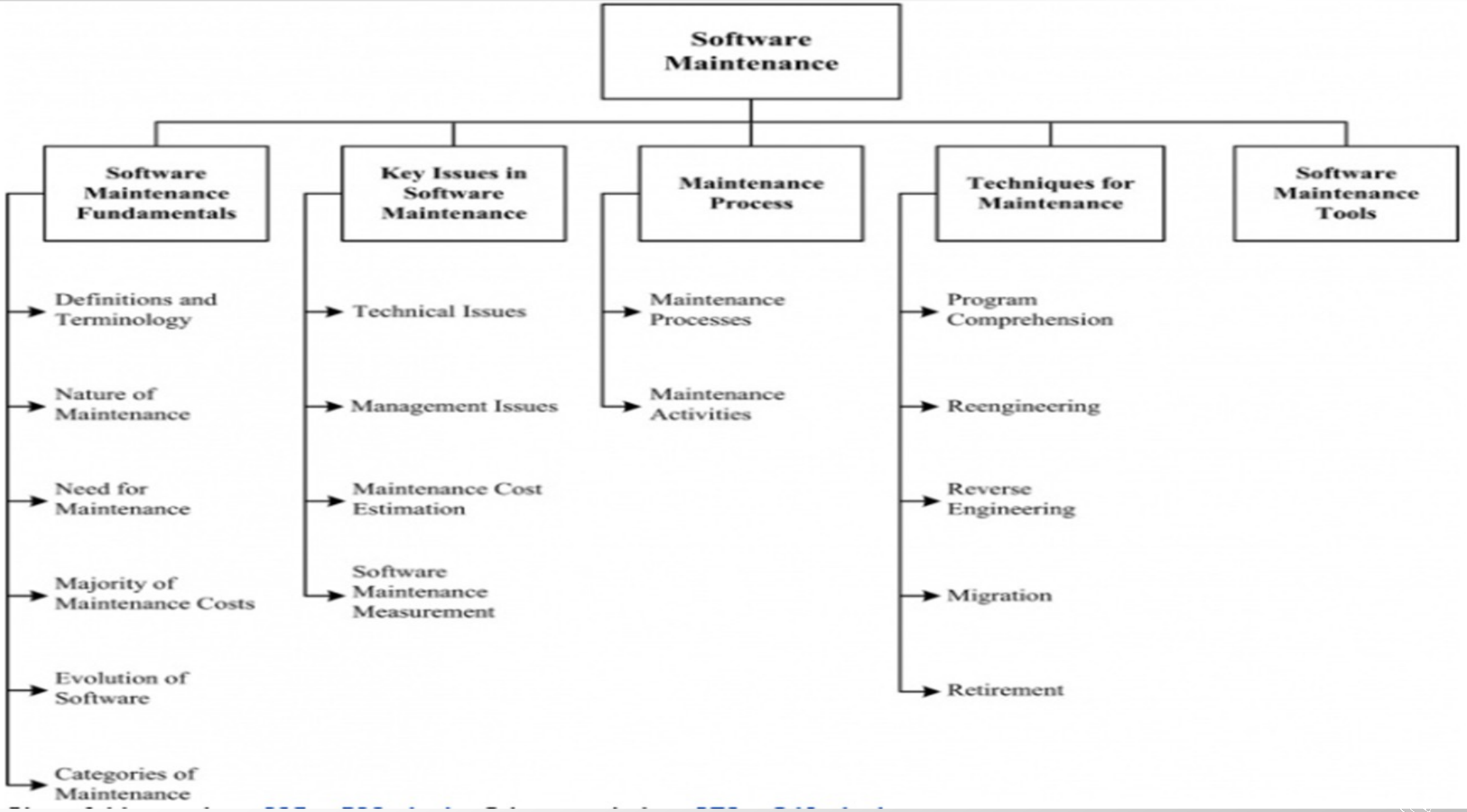
Other maintenance problems are:

1. Poor traceability to the processes and products that created the software
2. Changes rarely documented
3. Difficulty of change management and monitoring
4. Ripple effects of software changes

Software Maintenance Issues - Technical

Technical Maintenance problems:

1. Inadequate testing techniques
2. Little methodology, few standards, procedures and tools specific to maintenance
3. Source code in existing software complex and unstructured
4. Integration, overlap and incompatibility of existing system
5. Maintenance software runs on obsolete systems and technologies
6. There is a lack of automated testing and diagnostic tools for successful change implementation.


Software Maintenance
Software Maintenance Fundamentals
Key Issues in Software Maintenance
Maintenance Process
Techniques for Maintenance
Software Maintenance Tools
Definitions and Terminology
Nature of Maintenance
Need for Maintenance
Majority of Maintenance Costs
Evolution of Software
Categories of Maintenance
Technical Issues
Management Issues
Maintenance Cost Estimation
Software Maintenance Measurement
Maintenance Processes
Maintenance Activities
Program Comprehension
Reengineering
Reverse Engineering
Migration
Retirement

Fundamentals	Key Issues	Maintenance Process	Techniques	Tools
Definitions & Terminology	Technical Issues	Maintenance Processes	Program Comprehension	
Nature of Maintenance	Management Issues	Maintenance Activities	Re-engineering	
Need for Maintenance	Maintenance Cost Estimation		Reverse Engineering	
Majority of Maintenance Costs	Software Maintenance Measurement			
Evolution of Software				
Categories of Maintenance				

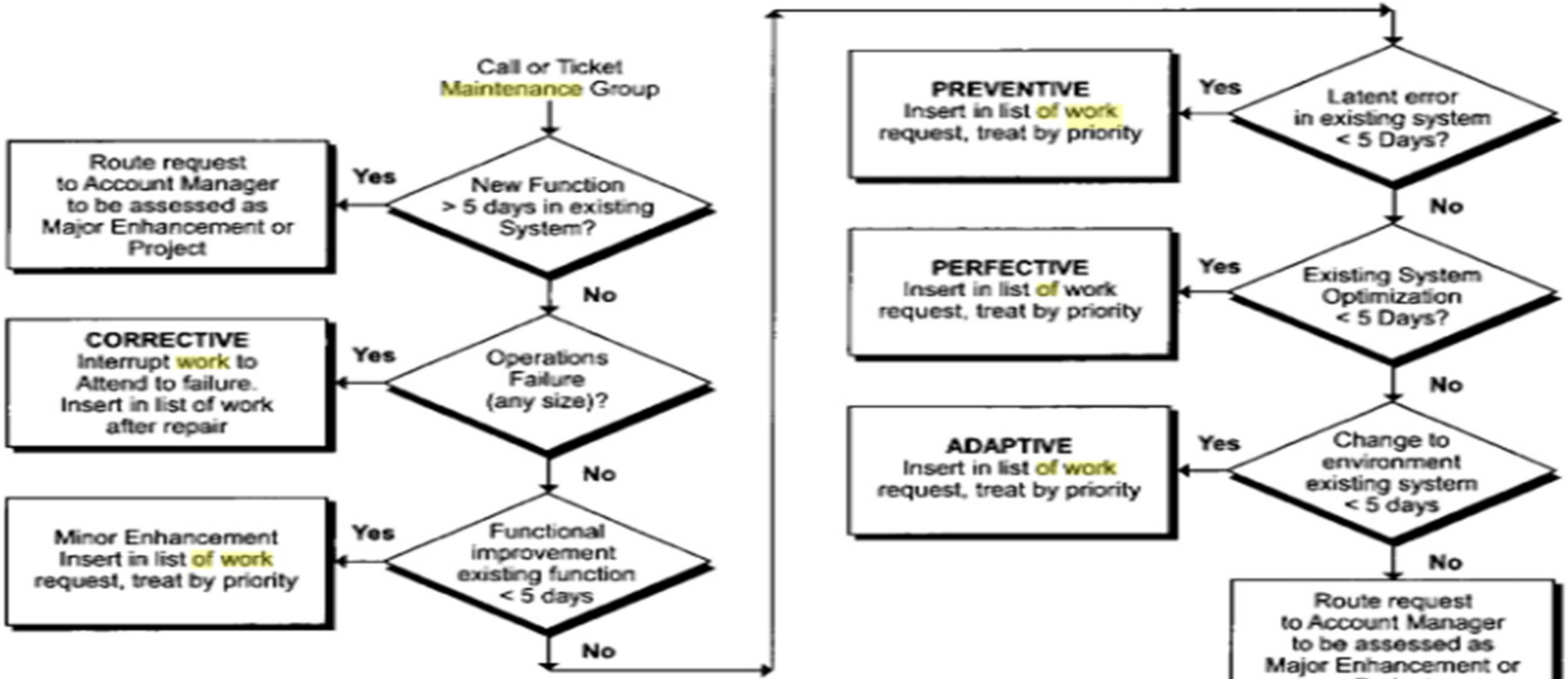
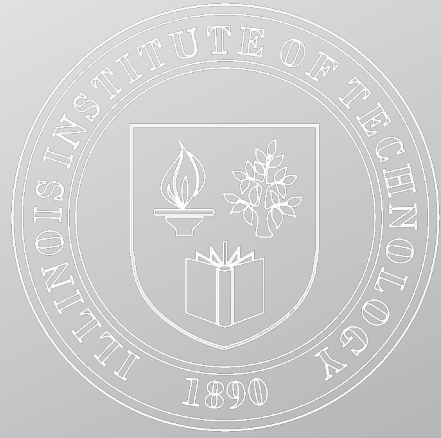


Figure 1.4 Example of the acceptance/refusal process of maintenance work [April 2001].

Software Maintenance Standards

ISO 12207 software maintenance process includes following six sub processes

1. Process implementation
2. Problem & modification analysis
3. Modification implementation
4. Maintenance review/acceptance
5. Migration
6. Software retirement



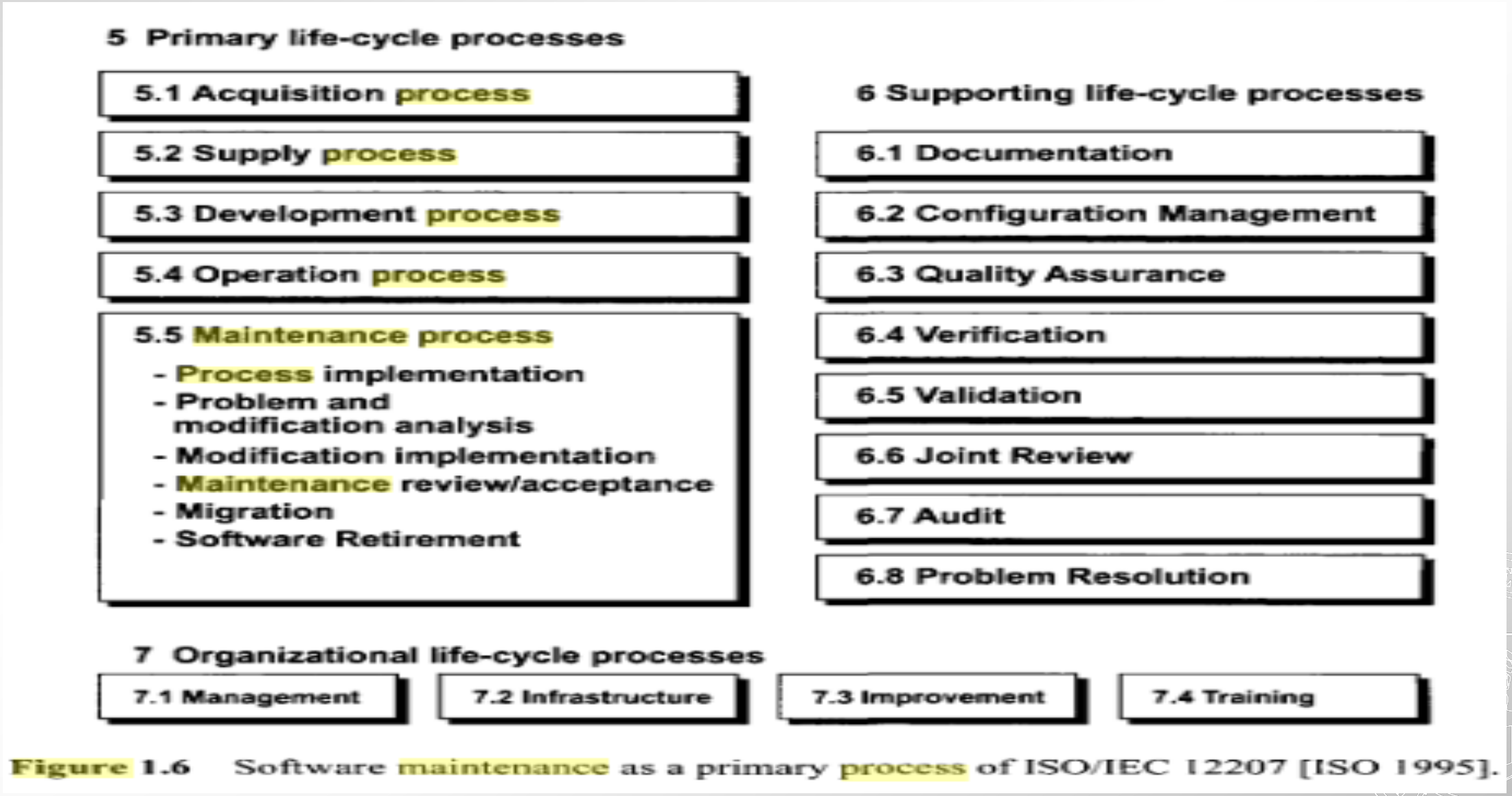


Figure 1.6 Software maintenance as a primary process of ISO/IEC 12207 [ISO 1995].

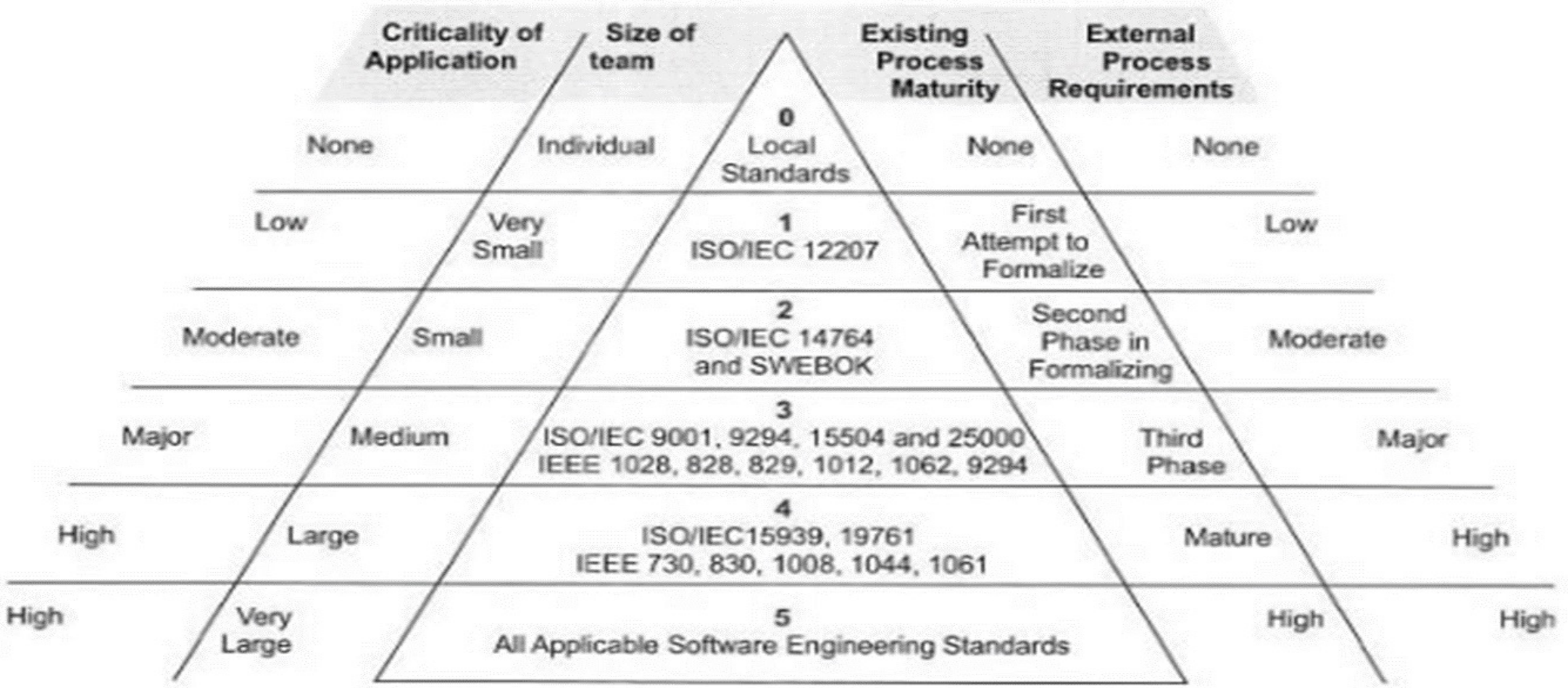


Figure 1.8 Applicable standards pyramid—a simple maturity model.

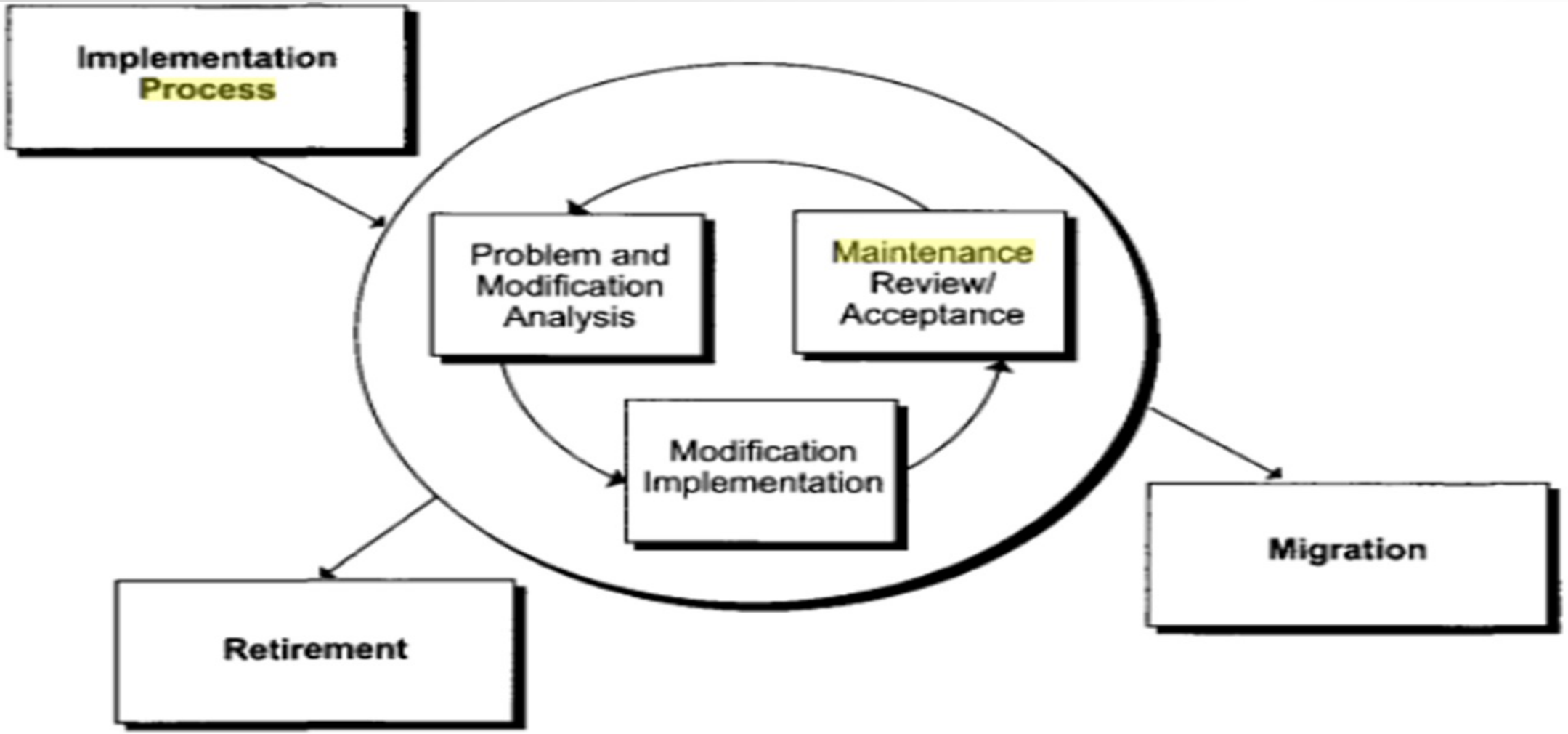
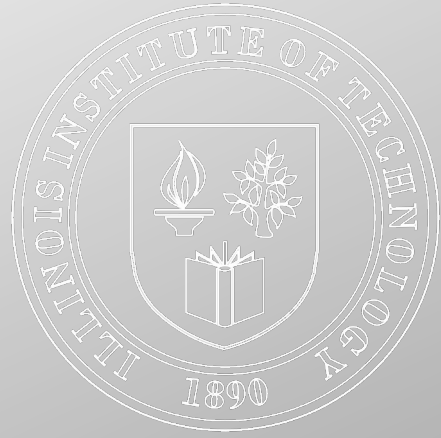


Figure 1.9 Software maintenance key processes [ISO 2006].

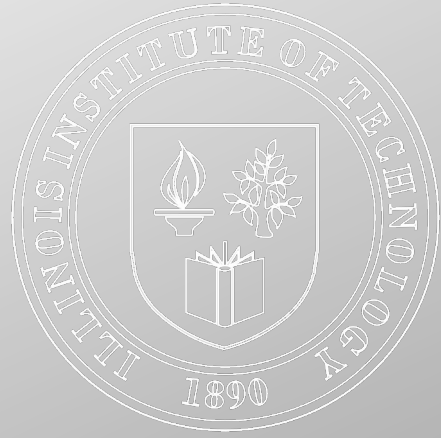
Software Maintenance Process

1. Application (***Before***)
    1. Functional size
    2. Lines of code
    3. Environment characteristics
2. Maintenance Request
    1. Number
    2. Category
3. Effort
    1. Person-days
    2. Planned
4. Other Inputs
    1. Tools
    2. Administrative Services



Software Maintenance Process

1. Application(After)
    1. Functional Size
    2. Lines of code
    3. Environment characteristics
2. Completed Requests
    1. Number, Category
    2. Functional size, Lines of code
    3. Person-days



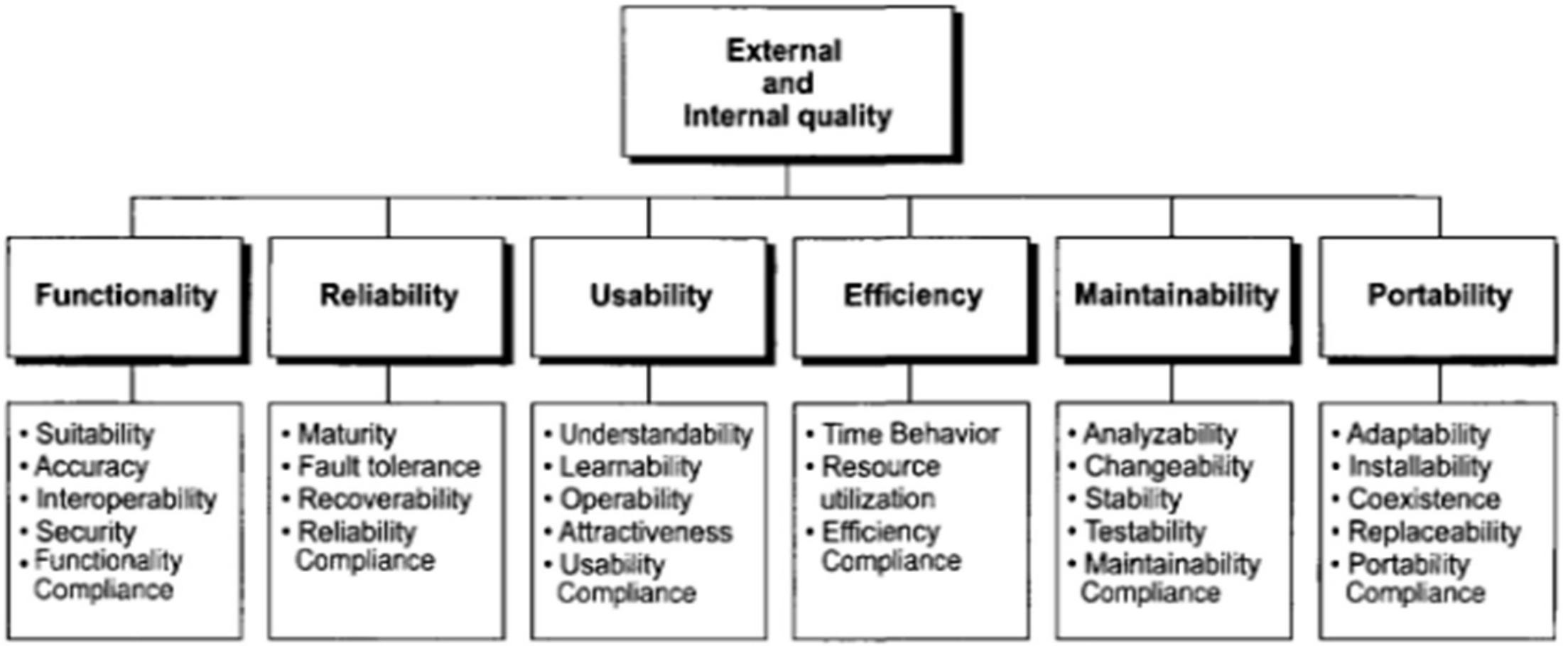
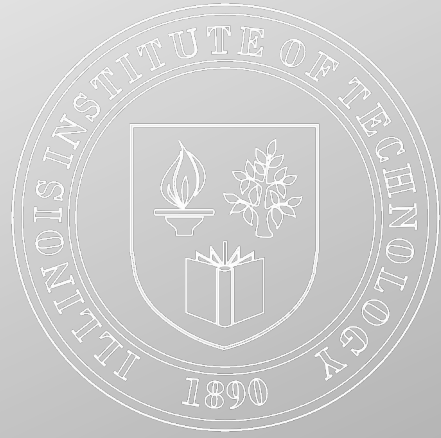


Figure 1.14 ISO 9126—Software product quality model [ISO 2001].

Service Measurement

There are three service level agreements in software maintenance:

1. Internal service-level agreement (SLA)
2. Maintenance service contract
3. Outsourcing contract



Internal Service – Level Agreement

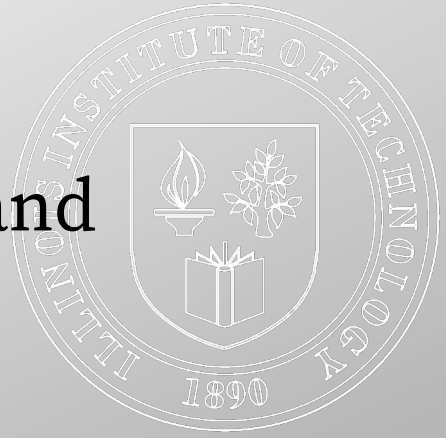
1. Responsibilities of the maintenance customer
2. Responsibilities of the maintenance organization
3. Description of maintenance services:

Internal Service – Level Agreement

3. Description of maintenance services:
   - Maintenance program management
   - Management of requests, priorities, and
   - The request management software
   - Corrective, preventive, adaptive, and perfective maintenance
   - Planning and management of software versions (releases)
   - Configuration management
   - License management, escrow delivery, and contracts with third parties
   - Recovery after disasters
   - Customer support

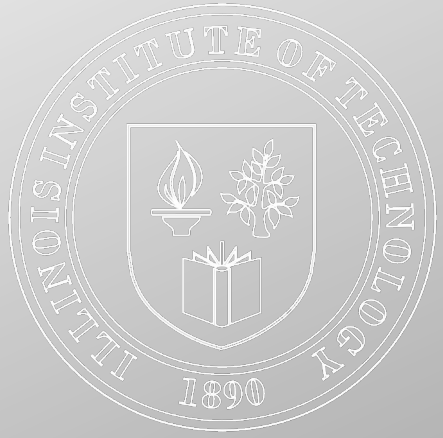
Internal Service – Level Agreement

- Exclusions
- Detailed list of supported software, by priority
- Service fees
- Service hours
- Escalation procedures in case of problems
- Measures of performance
- Review and mechanisms for solving disagreements and conflicts



Maintenance Service Contracts-External Service Agreement

- Definition
- Supplier obligations
- Maintenance services and optional services
- Scope of maintenance
- General terms of the agreement
- Customer and customer personal obligations



Maintenance Service Contracts-External Service Agreement

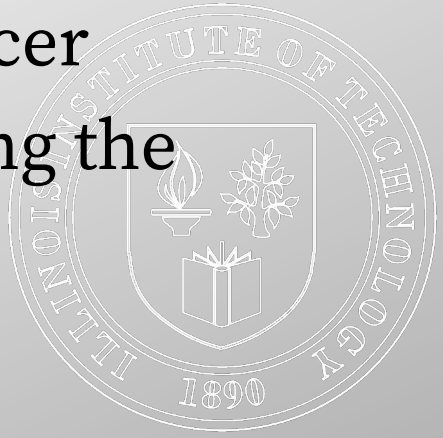
- Confidentiality
- Reproduction of documentation and source code
- Limits of responsibility, acts of God, order payments, survival, laws
- Support procedures

Maintenance Service Contracts-External Service Agreement

- Suppliers proposes temporary solutions
- Suppliers negotiate renewals of the license and maintenance
- Additional charges are not clearly defined
- Escrow services are rarely discussed or included in the costs

Outsourcing Agreements

- Promising of decreasing costs
- Access to the expertise of the outsourcer's personnel
- Move from a fixed-cost structure to a variable-cost structure
- Collect revenue from the sale of an asset
- IS/IT not being one of the company's strategic activities
- Transfer of technical details and problems to the outsourcer
- The outsourcer offers typically a 90-day warranty following the correction of a specific problem



Outsourcing Agreements

- The outsourcer asks the client to determine the priority of maintenance work items
- The outsourcer keeps a list and a history of problems submitted using help desk software
- ISO 14764 software maintenance work categories are not often used
- Outsourcing agreement is for 5 -10 years

2.1 What is Maintenance & why is it important?

- "Software maintenance – software maintenance is the process of modifying a software system or its components after delivery to correct faults, improve performances or other attributes, or adapt to a changed environment."
- Maintenance is flexibility agent.
- Software must adapt as the environment in which it operates changes.
- There are five reasons for software changes.

2.1 What is Maintenance & why is it important?

1. Enhancements: Add capabilities to an existing capability in a manner that preserves the previous functionality and performance.

- Changes may also be needed to remove or alter functionality, in particular when it becomes irrelevant.

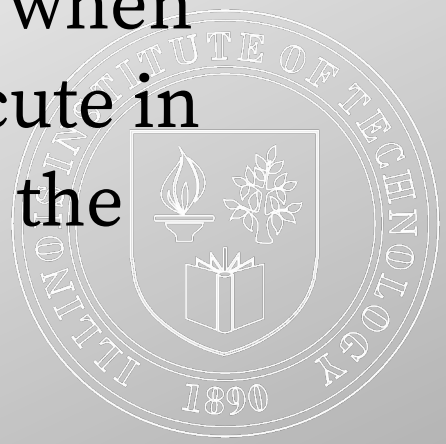
2. Repairs: Actions taken on nonconforming product to find and fix problems so that it will perform as intended.

- Defects that must be fixed in the software application.

2.1 What is Maintenance & why is it important?

3. Performance Improvement: The degree to which the software accomplishes its designated functions within given constraints, such as speed, accuracy, or memory utilization.

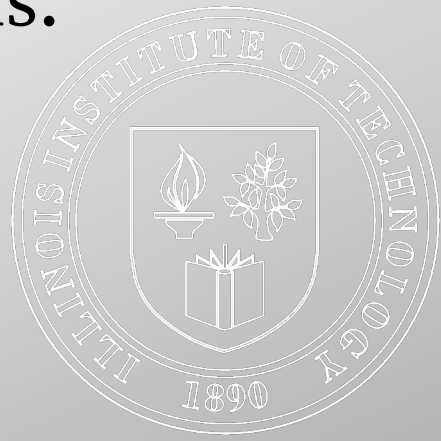
- Improve speed, performance, and access, especially when the system becomes mired down with tasks that execute in the background and with temporary files that clutter the storage devices.



2.1 What is Maintenance & why is it important?

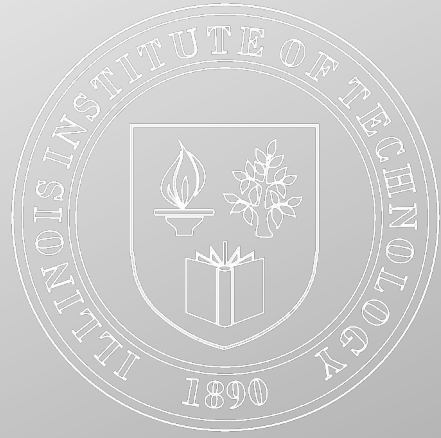
4. Environment upgrades: The software does not operate in a static world. The platforms that it runs on and environments in which it operates are periodically updated, upgraded, and replaced.

- Repair and replace packages and continue to use as-is.



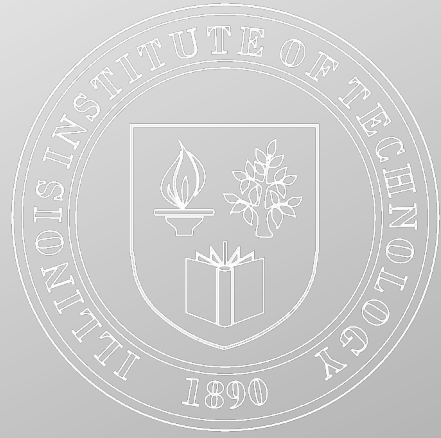
2.1 What is Maintenance & why is it important?

- **5. Configuration changes:** In configuration management, the functional and physical characteristics of hardware or software as set forth in technical documentation or achieved in a product.



2.1 What is Maintenance & why is it important?

- "*Maintainability*- refers to the ease, in which the software can be maintained, enhanced, adapted, or corrected to satisfy specified requirements."
  1. Maintainer
  2. Maintenance environment
  3. Maintenance process
  4. Sustaining engineering



Who does it (Maintenance)Why, When, Where, & How

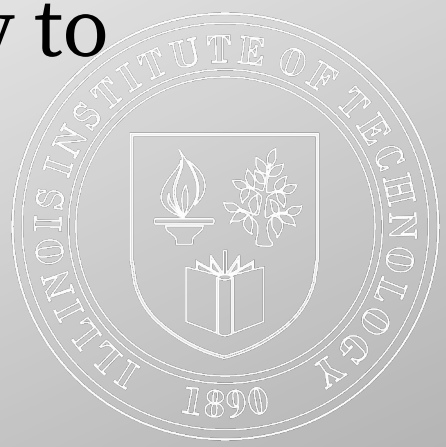
- **Who:**
  - *Reality*: Separate maintenance teams whose workforce is senior and more skilled and experienced than the development team:
    - Commercial-mainly outsourced.
    - Government-primarily introduced.
- **Why:**
  - *Reality:* To reduce the backlog of software change requests and software trouble reports addressed in priority order based on an assessment of priority, change impact, and cost

Who does it (Maintenance)Why, When, Where, & How

- **Where:**
  - *Reality:* At both a central maintenance facility and operational sites in the field; testing was done at some integration facility
- **When:**
  - *Reality:* After delivery of product, during the first year of operations and, thereafter, on a periodic block release basis
- **How:**
  - *Reality:* Often done on a catch as catch can basis; however, mature maintenance shops have well-disciplined set of processes that are broader than those used during development

2.2.1 As Part of Development

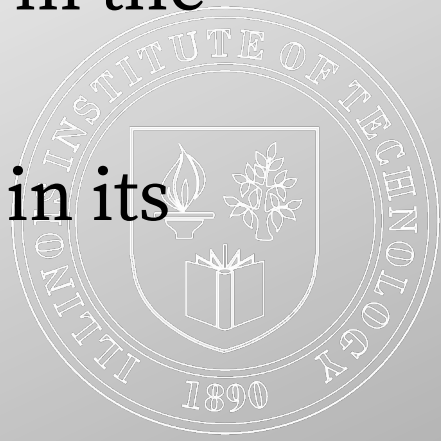
- Planning for transition is needed so that the maintenance can take over after the product development and deployment.
- Planning also enable the maintenance shop to acquire staff resources with skills, knowledge and ability to update and fix software.



2.2.2 During Transition

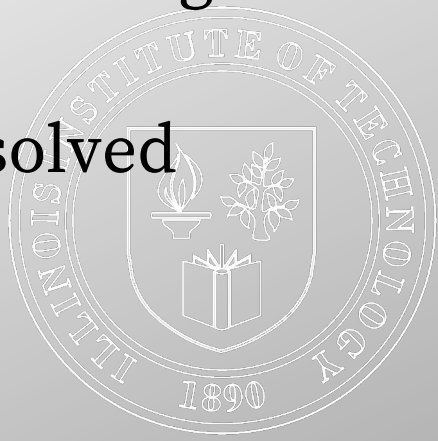
Readiness checklist:

- Have all functional and performance requirements been satisfied?
- Does the product work as indented in its environment?
- Has a set of baselined regression tests been included in the deliver?
- Has the documentation agreed upon been furnished in its desired form?



2.2.2 During Transition

- Have all high-priority defects been fixed, and has an open problem list been furnished with the deliverable?
- Have functional, physical and requirements baselines been created for the configurations?
- Has a configuration index containing a listing of all components in the deliverable (e.g., a bill of materials) been furnished along with “make” and “configure” instructions?
- Has a list of open problems and issues that need to be resolved been provided?

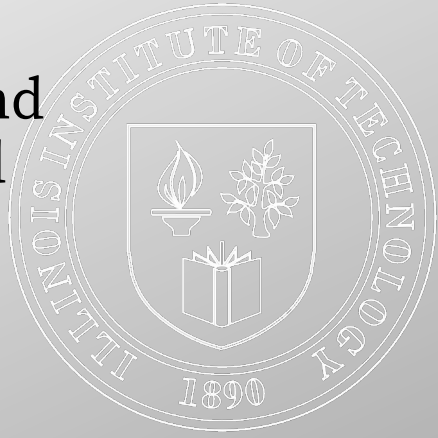


2.2.3 After Turnover

- The point in the life cycle when the software and responsibility for is maintenance and support are transferred from the developer to the maintainer. For this transfer to occur, all terms and conditions spelled out in the transfer agreement must be satisfied (or waived).

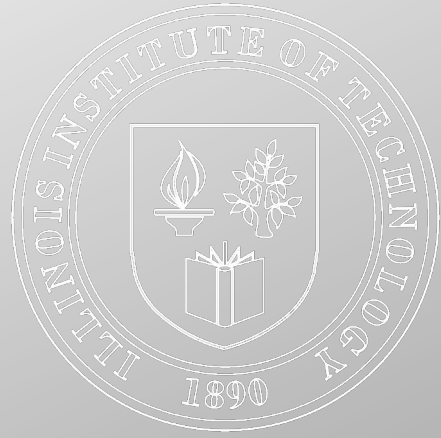
2.3 Operation Concepts and Constraints

- **Operational Concept:** For maintenance, refers to the maintenance strategy that will be followed for the release (in-house/contracted, in-sourced/out-sourced, etc.).
  - The operation and maintenance phase begins immediately after turnover is complete.
  - Before the software is placed into operations, the following preparatory steps must be completed.
  - First, the source code is subjected to a quality assurance audit.
  - The code must be self-documenting, and the user, maintenance, and reference manuals that are delivered which must be up-to-date and consistent.



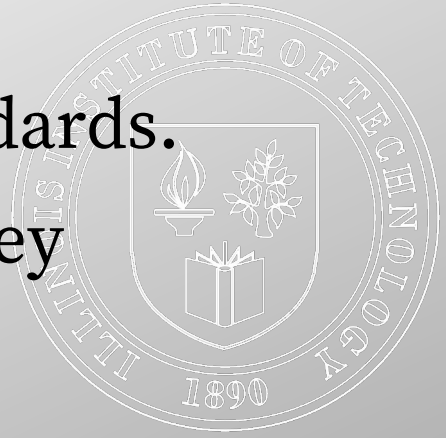
2.3 Operation Concepts and Constraints

- User Guide, Installation Manual are usually written by programmers/Technical Writers
- Finally, the software must be configured and tailored if it is intended to be used on several platforms.
- It must be tested to ensure that it functions properly and performs as intended on each of these platforms.



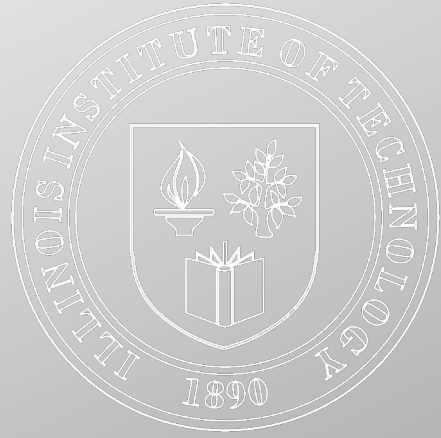
2.4 Characteristics of World-Class Organization

- They train their project managers in this infrastructure and standards and hold their teams accountable for results.
- They plan for transition and turnover and hold a readiness review prior to delivery.
- They anticipate issues that can occur during transition and turnover.
- They have strong product and process management standards.
- They embrace configuration management concepts as they ensure the integrity of the product and test the baselines.



2.4 Characteristics of World-Class Organization

- The embrace quality assurance techniques and use them to ensure that the product delivered will make them proud.
- They have a capable workforce trained in their process infrastructure and product standards.
- Deliver quality product on time and within budget.
- Keep track of risks and work proactively.
- Achieve transition goals in a seamless manner.
- Use regression testing to validate changes.



2.5 Issues and Answers

- Top ten Issues and Answers during Transition and Turnover: (*Issue/Mitigation*)

1. Lack of agreement over what constitutes acceptable delivery from development to maintenance
    - Set up a working group to define expectations and measures for use in accepting delivery to maintenance.
2. Development group says here is the software and then leaves you alone to deal with it; they then reassign the people and forget about you
    - Conduct an end-product acceptance review to ensure that delivery expectations are met; have the courage to say “no, this is not acceptable”

2.5 Issues and Answers

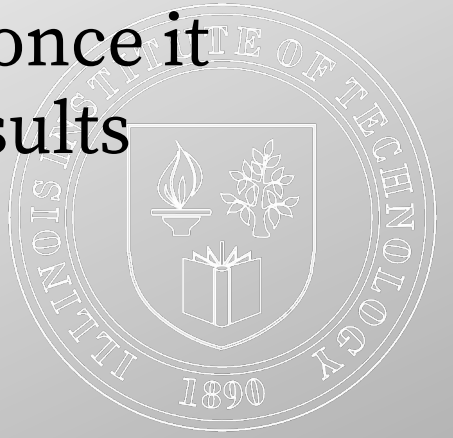
3. You take delivery under the impression that the quality is good; however, the number of defects occurring during transition is higher than expected and the system crashes unpredictably

- If possible, have your team present when the developers qualify their software; make sure that test records confirm that software is ready to be transitioned; use defect measures to quantify quality

2.5 Issues and Answers

4. Failure to establish, baseline and turnover a set of tests (scripts, test cases, and test results) for use in revalidating software when it undergoes change (i.e., the regression test baseline)

- Work with the development group to create a regression test baseline that can be used to revalidate software once it is changed; ensure that the test scripts, cases and results used supply adequate test coverage



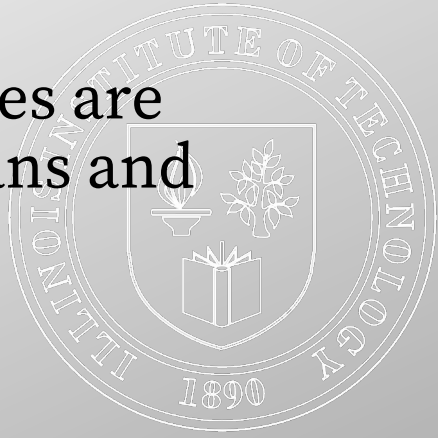
2.5 Issues and Answers

5. Personnel reward system does not provide incentives to do a good job of transition and turnover

- Put specific transition and turnover goals and rewards into appraisals for those personnel working these tasks

6. Facility readiness is an issue because resources were not available when needed to buy equipment (including software licenses) and upgrade facilities

- Make sure that long-lead items such as equipment and facilities are adequately addressed in approved transition and turnover plans and agreements



2.5 Issues and Answers

- 7. Commercial off-the-shelf (COTS) equipment and software are not treated as a major risk on the program; COTS vendors can be acquired or go out of business; they often evolve the package and update it in a manner that meets the market rather than your specific needs
  - Put COTS on the “Top10” risks; institute a market watch to make sure that replacements are available, if needed, that provide equivalent functionality at a reasonable price; consider source code escrow just in case the COTS vendor goes out of business

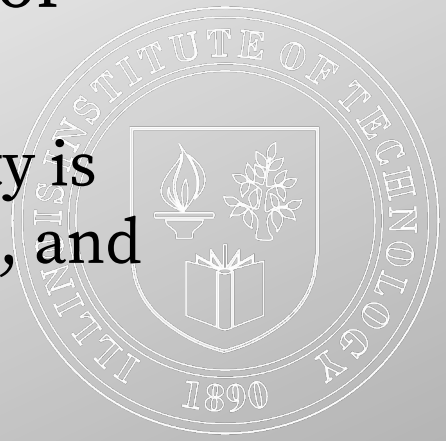
2.5 Issues and Answers

8. Process framework fails to provide the practices required for maintenance tasks during transition and afterwards

- Have process group develop practices that address needed software maintenance tasks and train the staff in their use

9. Product standards focus almost entirely on software development concerns; they do not look at what is needed for software maintenance

- Broaden your product standards to ensure that maintainability is addressed as the software is architected, designed, developed, and delivered.



2.5 Issues and Answers

10. Process framework fails to provide the practice required for maintenance tasks during transition and afterward

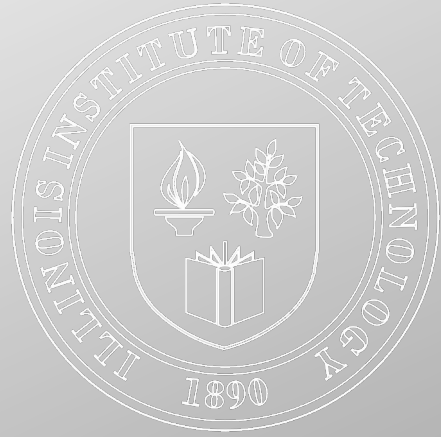
- Have process group develop practices that address needed software maintenance tasks and train the staff in their use

Overview of Basic Concepts (Process & Maturity)

- Process “A process can be defined as a set of activities that transform inputs into outputs and thereby fulfill a purpose. Typically the outputs from one process become the input(s) to one or more further processes. The activities operating within a process should be coherent and complete”

  Crosby’s quality management evolution grid is the source of the SEI’s design of maturity steps or levels.

  1. Uncertainty
  2 Awakening
  3. Enlightenment
  4. Wisdom
  5. Certainty



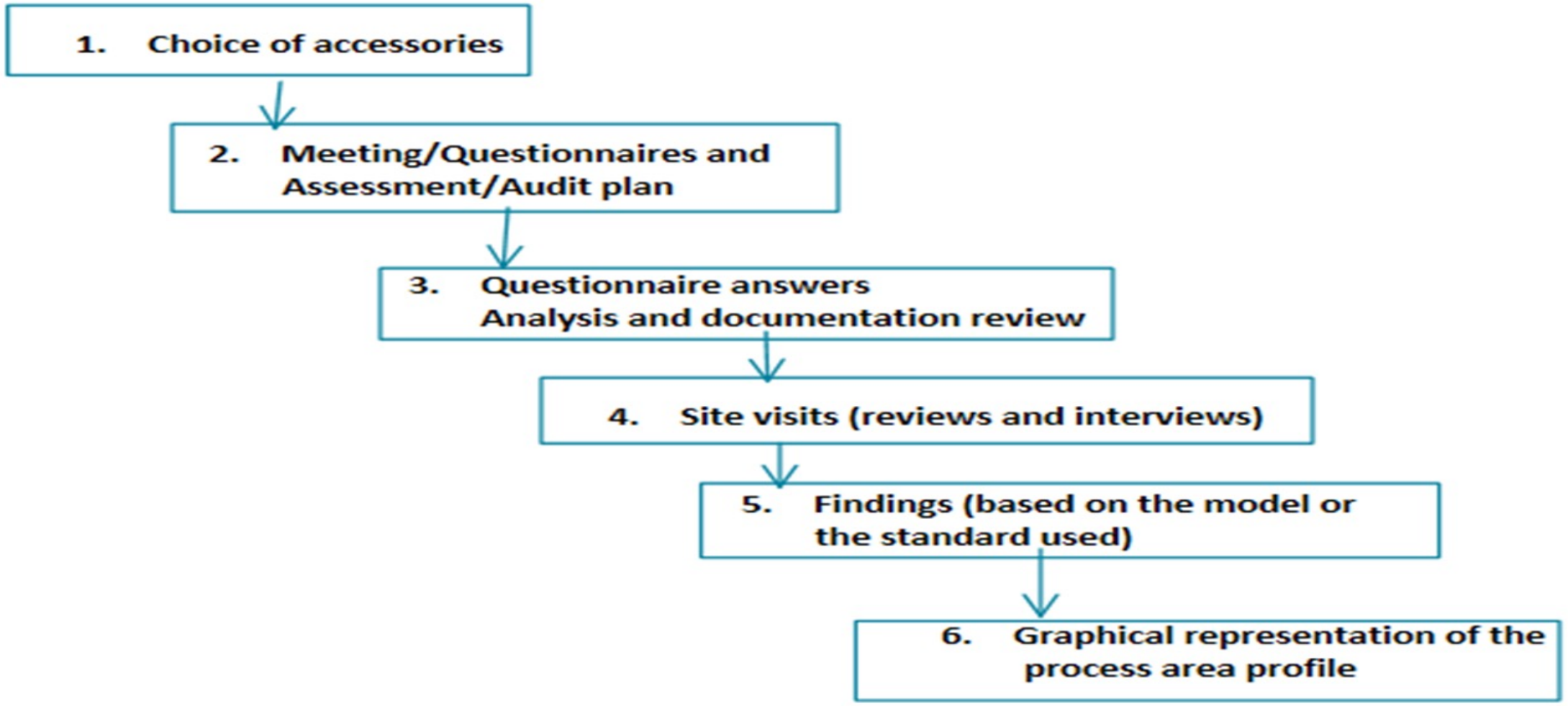
Capability Maturity Model (CMM)

- **Capability Maturity Model:** A description of the stages through which organizations evolve as they define, implement, measure, control, and improve their processes. The model provides a guide for selecting process capabilities and the identification of the issues most critical to quality and process improvement

Capability Maturity Model

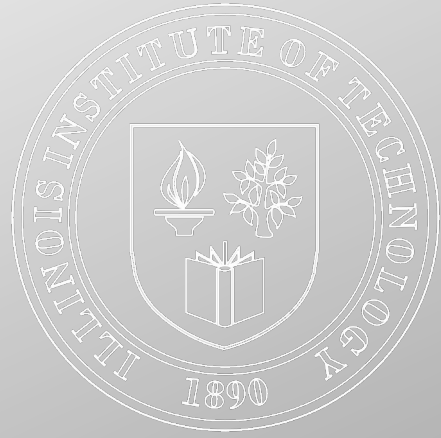
Choosing between ISO 9001 & CMMi

- ISO 9001 is extensively documented:
- Improvement of internal documentation
- A better quality product
- Competitive advantage
- Fewer customer audits
- Level 3 of the CMM is reachable for an organization that already has ISO 9001 certification


1. Choice of accessories
2. Meeting/Questionnaires and Assessment/Audit plan
3. Questionnaire answers Analysis and documentation review
4. Site visits (reviews and interviews)
5. Findings (based on the model or the standard used)
6. Graphical representation of the process area profile

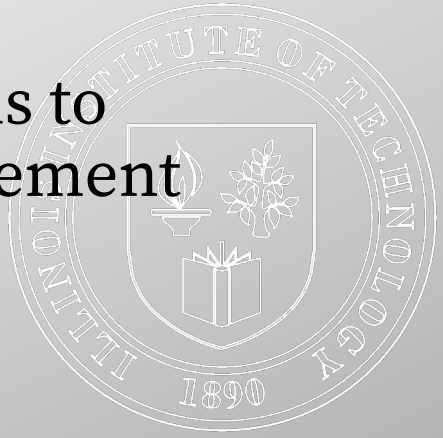
Evaluation Methods – ISO 9001 & CMMi

1. Selection of individual for evaluation
2. Validate scope/process with the organization and create an - agreement
3. Analysis of documentation reviews
4. On-site visit conduct reviews and interviews (evaluate/audit)
5. Reporting (nonconformities/opportunities for improvement)



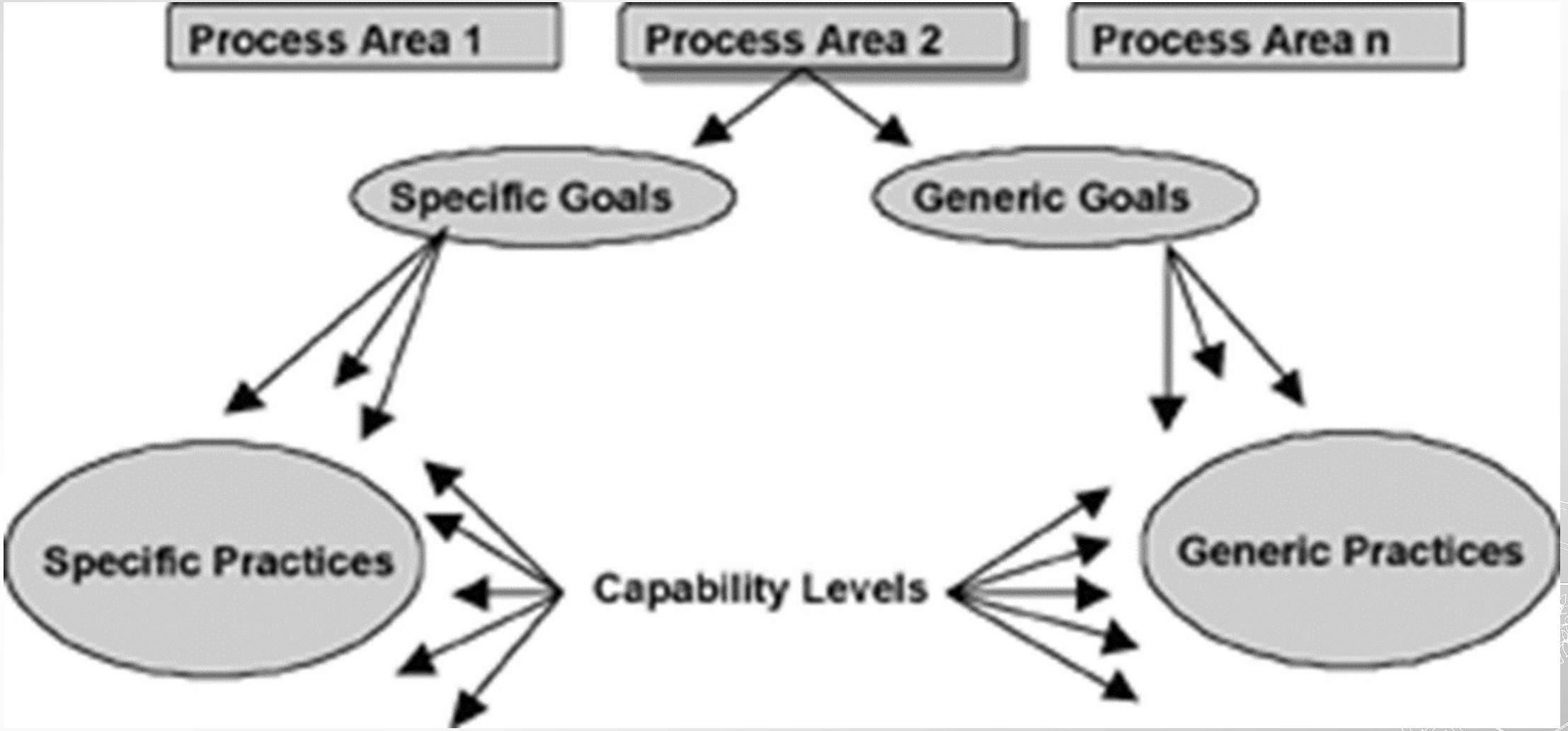
Evaluation Types

- **Audit:** The goal is to investigate a supplier's or service subcontractor's development process in on order to make business decision
- **External Audit:** Internal evaluation performed with the help of external auditors (ISO 9001 Certification or Re-Certification)
- **Joint Audit:** The goal of which is to evaluate the development process with a team of people from both organization (interfaces with subcontractors)
- **Self-Audit (Internal):** The goal of which is for an organizations to evaluate its own development processes. (Continuous improvement process)



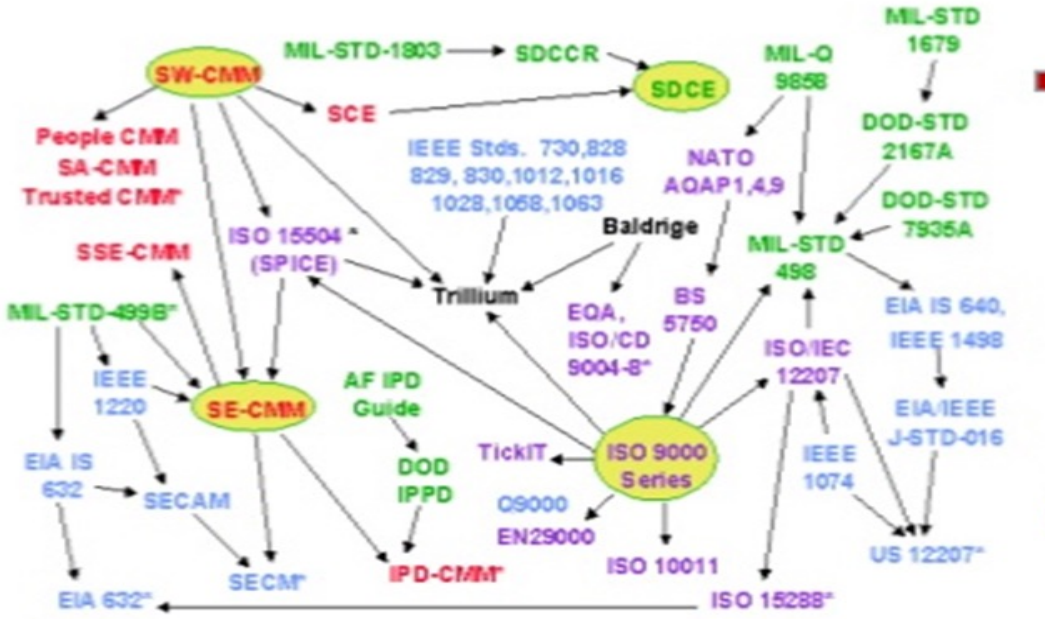
Maturity Model

- Maturity model is designed through:
- A consensus of experts
- Studying the architecture of current models
- Multiple mappings of standards and other approved source documents
- The addition of specialized practices peculiar to a specific domain
- A rationale for the classification of each practice into a specific maturity level
- Experimentation for model adjustment and improvement


Process Area 1
Process Area 2
Process Area n
Specific Goals
Generic Goals
Specific Practices
Capability Levels
Generic Practices

Process Evolution

Chief Engineer *increase awareness and consistency across the Agency and advance the practice of engineering …The engineering of NASA systems requires a systematic and disciplined set of processes … for the design development, operation, maintenance, and closeout of systems throughout the lifecycle of the programs and projects.*



Software Productivity Consortium "The Quagmire" 2000
www.software.org

ISO 9001
CMMI
NPR7150.2
ITIL
IEEE 12207
Six Sigma …

You are here

Best Practices
Process Improvement
Institutionalization

COMPLIANCE IS MANDATORY
Resistance is Futile

TIETRONIX

Root Cause Analysis

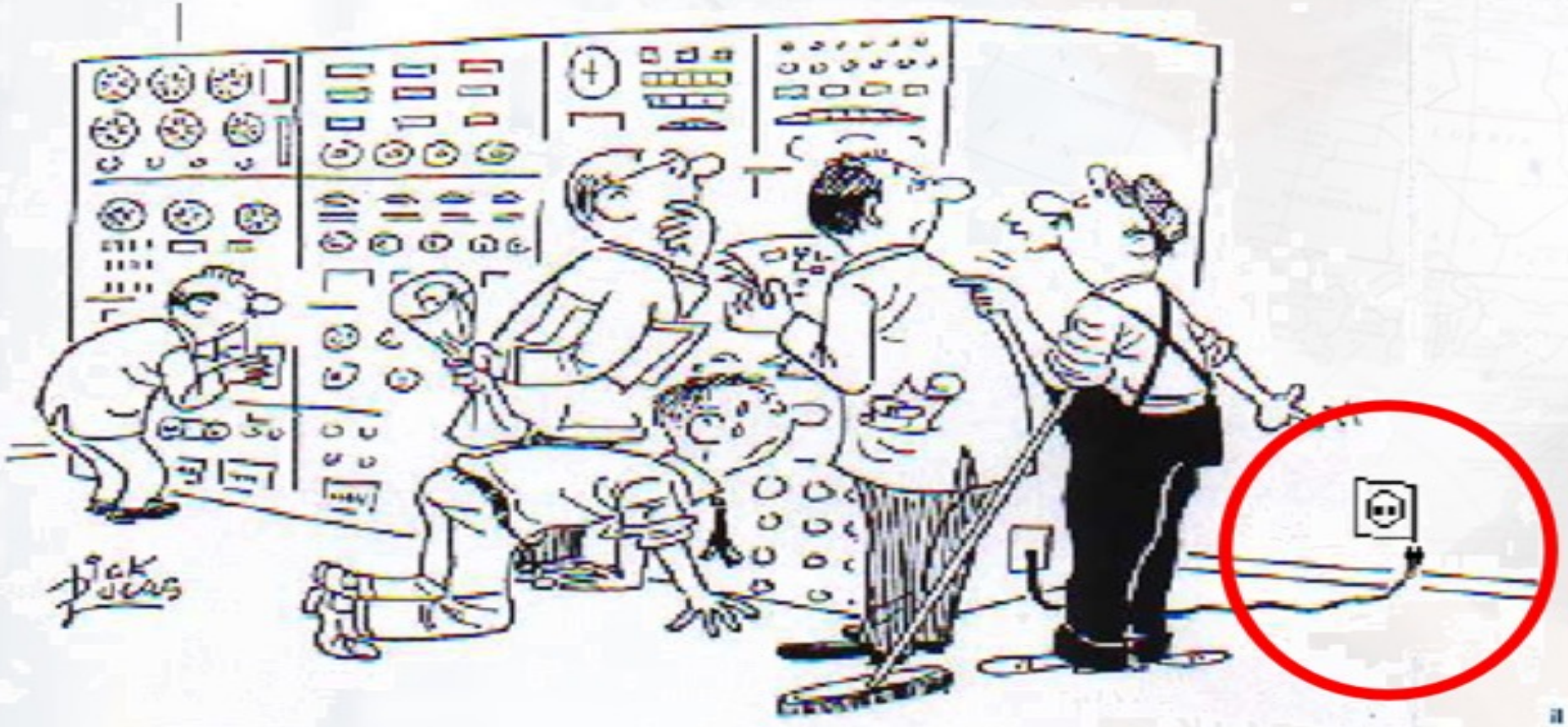
Root cause analysis:

The root cause analysis shall determine why the problem could occur (technical and system-related) and why it was not been detected.

The root cause has been found, if the problem can be resolved and can be switched on and off.

3 x 5 Why Technique

How to find the root cause?



3 x 5 Why Technique

➔ **5 Whys Technique:**

1. *Why the computer doesn't work?* **Because it is not connected**
2. *Why the computer is not connected?* **Because the plug cable was pulled**
3. *Why the plug cable was pulled?* **Because something tangled in the cable and pulled out**
4. *Why things tangling in the cable?* **Because it gets loose on the floor and in the way**
5. *Why it gets loose on the floor and in the way?* **Because it is too long**
6. *Why the cable is too long?* **I don't know**

~~**Solution:** To connect!~~

3 x 5 Why Technique

Question: Why the computer doesn't work ?

Answer: Because it is not connected!

Solution: **To connect !!!**

It is considered an easy and quick solution!

(So easy that we don't matter to repeat several times if happen again)

3 x 5 Why Technique

➔ **Corrective actions - alternatives:**

Action a): reduce the cable length

Action b): pass the cable on a conduit

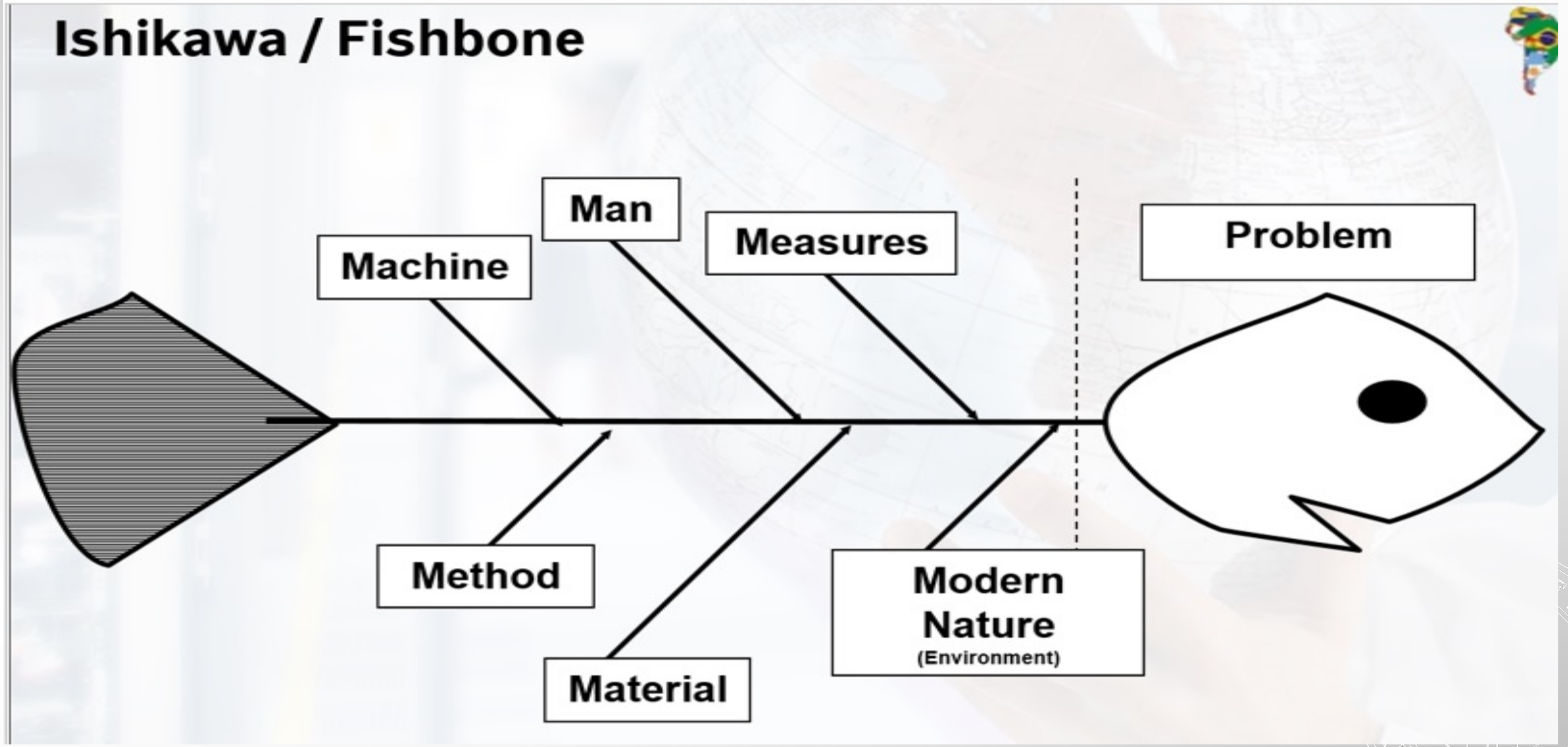
Action c): bring the machine closer

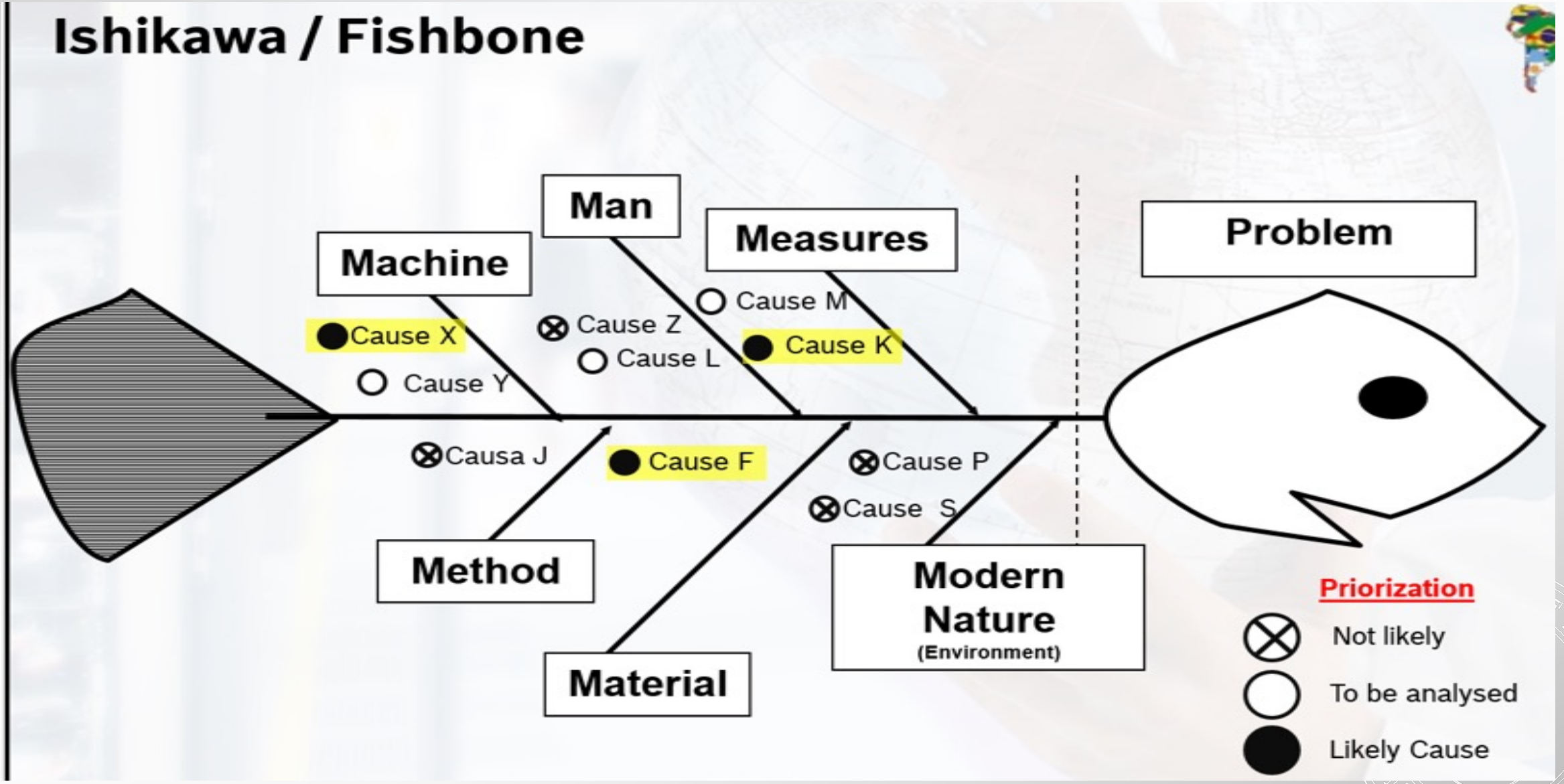
Is – Is not

7 Questions	Is	Is not

Is – Is not

7 Questions	Is	Is not
Who?	Who is involved in the problem? "Customers, sectors, documents, people, etc.	Who was not involved in the problem? "Customers, sectors, documents, people, etc.
What?	What product / object has the problem?	What product / object does not have the problem?
When?	When the problem was identified? "month, day, hour, minute, working shift"	When the problem was not identified? "month, day, hour, minute, working shift"
Where?	Where the problem was found? "Country, location, customer, industry, machine, supplier, etc. ..."	Where the problem was not found? "Country, location, customer, industry, machine, supplier, etc. ..."
Why?	Why is this problem? " effect arising"	Why is this not a problem? " effect arising"
How?	How does the problem appear?	How does the problem appear not appear, but could appear?
How often?	loads, quantitive noticeable problems/rhythms, intervals	


Ishikawa / Fishbone
Man
Machine
Measures
Problem
Method
Material
Modern
Nature
(Environment)


Ishikawa / Fishbone
Man
Machine
Measures
Problem
Cause M
Cause Z
Cause X
Cause K
Cause L
Cause Y
Causa J
Cause F
Cause P
Cause S
Method
Material
Modern Nature
(Environment)
Priorization
Not likely
To be analysed
Likely Cause

Problem Statement:

3x5 Why Leader:

Participants:

	1. Why?	2. Why?	3. Why?	4. Why?	5. Why?	Action	Resp. Person / due date
1) Why did it occur? Write the "Direct Cause" here:							
2) Why was it designed or processed like this?							
3) Why was it not detected?							

Problem Statement: Customer complains about the coffee temperature (the coffee was served cold).

3x5 Why Leader: John (Coffee Shop Manager)

Participants: Maria, Jose, Pedro (Waiters); Larissa e Mario (Cooks); Leandro (Supervisor)

	1. Why?	2. Why?	3. Why?	4. Why?	5. Why?	Action	Resp. Person / due date
1) Why did it occur? Write the "Direct Cause" here:	**Why the coffee was cold?**	**Why it took a long time to be served?**	**Why the waiter took a long time to deliver the order?**	**Why he had to answer the phone?**	**Why there wasn't anybody else available in that moment?**	1- Replan the employees functions. 2-Give a specific person responsibility for answering the calls.	1- John – 07/06/2013 2- John – 07/06/2013
	Because it took a long time to be served.	Because the waiter took a long time to deliver the order to the customer.	Because he had to answer the phone	Because there wasn't anybody else available to answer the phone in that moment.	Because the employees functions were not correctly specified.		
2) Why was it designed or processed like this?	**Why the waiter takes a long time to deliver the customer order?**	**Why there is a delay to start preparing the coffee?**	**Why much time is wasted in the deliver of previous orders?**	**Why there are many obstacles in the way?**		1- Redesign the layout. 2- Create a procedure to specify the maximum time between order and delivery.	1- John – 30/06/2013 2- Leandro – 15/07/2013
	Because there is a delay to start preparing the coffee.	Because much time is wasted in the deliver of previous orders.	Because there are many obstacles in the way from the kitchen to the customer's table.	Because the Coffee Shop layout was not well designed.			
3) Why was it not detected?	**Why was not detected that the coffee was cold?**	**Why the waiter did not check the coffee temperature?**	**Why there is no control or instructions on how to check coffees temperature?**	**Why there is no training related to customer service?**	**Why the organization does not provide this kind of training?**	1- Create a staff Training program. 2- Create instructions on how to control the coffees Temperature.	1- John – 30/06/2013 2- Leandro – 15/06/2013
	Because the waiter did not check the coffee temperature.	Because there is no control or instructions on how to check coffees temperature.	Because there is no training related customer service.	Because the organization does not provide this kind of training.	Because the organization does not have a staff Training Program.		

Using Quality Tools