Introduction

- Traditional definition of quality:
  - fitness of purpose,
    - a quality product does exactly what the users want it to do.

Fitness of purpose

- For software products,
  - fitness of purpose:
    - satisfaction of the requirements specified in SRS document.

Fitness of purpose

- A satisfactory definition of quality for many products:
  - a car, a table fan, a food mixer, microwave oven, etc.
- But, not satisfactory for software products.

Introduction

- Consider a software product:
  - functionally correct,
    - i.e. performs all functions as specified in the SRS document,
  - but has an almost unusable user interface.
    - cannot be considered as a quality product.

Introduction

- Another example:
  - a product which does everything that users want.
  - but has an almost incomprehensible and unmaintainable code.

Modern view of quality

- Associates several quality factors with a software product :
  - Correctness
  - Portability
  - Usability
  - Reusability
  - Maintainability

Correctness

- A software product is correct,
  - if different requirements as specified in the SRS document have been correctly implemented.
  - Accuracy of results.

Portability

- A software product is said to be portable,
  - if it can be easily made to work in different operating systems,
  - in different machines,
  - with other software products, etc.

Reusability

- A software product has good reusability,
  - if different modules of the product can easily be reused to develop new products.

Usability

- A software product has good usability,
  - if different categories of users (i.e. both expert and naive users) can easily invoke the functions of the product.

Maintainability

- A software product is maintainable,
  - if errors can be easily corrected as and when they show up,
  - new functions can be easily added to the product,
  - functionalities of the product can be easily modified, etc.

MCCall’s Model

-- **Product operation factors:**
Correctness, reliability, efficiency,
Integrity, Usability

-- Product revision factors:
maintainability, flexibility, testability

-- Product transition factors:
portability, reusability, interoperability

Software Quality Management System

- Quality management system (or quality system):
  - principal methodology used by organizations to ensure that the products have desired quality.

Quality system

- A quality system consists of the following:
  - Managerial Structure
  - Individual Responsibilities.
- Responsibility of the organization as a whole.

- The quality system activities encompass the following:

- Auditing of projects
- Review of the quality system
- Development of standard, procedures and guidelines
- Production of reports for the top management summarizing the effectiveness of quality system.

Quality system

- Every quality conscious organization has an independent quality department:
  - performs several quality system activities.
  - needs support of top management.
  - Without support at a high level in a company,
    - many employees may not take the quality system seriously.

Quality System Activities:

- Auditing of projects
- Development of:
  - standards, procedures, and guidelines, etc.
- Production of reports for the top management
  - summarizing the effectiveness of the quality system in the organization.
- Review of the quality system itself.

Quality system

- A good quality system must be well documented.
  - Without a properly documented quality system,
    - application of quality procedures become ad hoc,
    - results in large variations in the quality of the products delivered.

Quality system

- An undocumented quality system:
  - sends clear messages to the staff about the attitude of the organization towards quality assurance.
- International standards such as ISO 9000 provide:
  - guidance on how to organize a quality system.

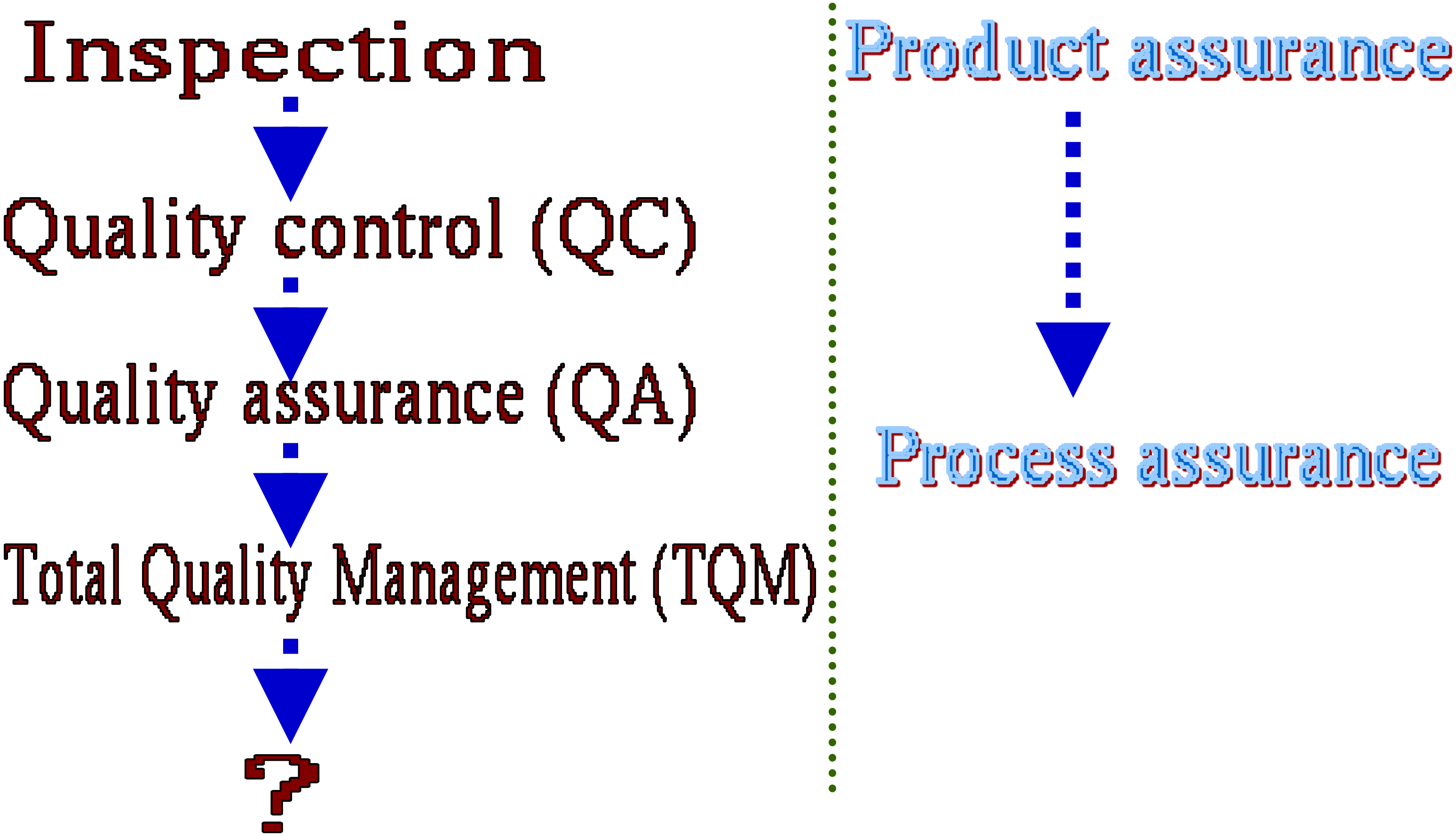
Evolution of Quality Systems

- Quality systems have evolved:
  - over the last five decades.
- Prior to World War II,
  - way to produce quality products:
    - inspect the finished products
    - eliminate defective products.

Evolution of Quality Systems

- Since that time,
  - quality systems of organizations have undergone
    - four stages of evolution.

Evolution of Quality Systems



Inspection

Quality control (QC)

Quality assurance (QA)

Total Quality Management (TQM)

?

Product assurance

Process assurance

Evolution of Quality Systems

- Initial product inspection method :
  - gave way to quality control (QC).
- Quality control:
  - not only detect the defective products and eliminate them
  - but also determine the causes behind the defects.

Quality control (QC)

- Quality control aims at correcting the causes of errors:
  - not just rejecting defective products.
- Statistical quality control
  - quality of the output of the process is inferred using statistical methods
  - in stead of inspection or testing of all products

Quality control (QC)

- The next breakthrough,
  - development of quality assurance principles

Quality assurance

- Basic premise of modern quality assurance:
  - if an organization's processes are good and are followed rigorously,
    - the products are bound to be of good quality.

Quality assurance

- All modern quality paradigms include:
  - guidance for recognizing, defining, analyzing, and improving the production process.

Total quality management (TQM)

- Advocates:
  - continuous process improvements through process measurements.

Business Process reengineering

- A term related to TQM.
- Process reengineering goes a step further than quality assurance:
  - aims at continuous process improvement.

Business Process reengineering

- Our focus is reengineering of the software process.
- Whereas BPR aims at reengineering the way business is carried out in any organization
  - not just software development organizations.

Total quality management (TQM)

- TQM goes beyond documenting processes
  - optimizes them through redesign.
- Over the years the quality paradigm has shifted:
  - from product assurance to process assurance.

ISO 9000

- ISO (International Organization for Standardization):
  - a consortium of 63 countries established to formulate and foster standardization.
- ISO published its 9000 series of standards in 1987.

What is ISO 9000 Certification?

- ISO 9000 certification:
  - serves as a reference for contract between independent parties.
- The ISO 9000 standard:
  - specifies guidelines for maintaining a quality system.

What is ISO 9000 Certification?

- ISO 9000 specifies:
  - guidelines for repeatable and high quality product development.
  - Also addresses organizational aspects
    - responsibilities, reporting, procedures, processes, and resources for implementing quality management.

ISO 9000

- A set of guidelines for the production process.
  - not directly concerned about the product it self.
  - a series of three standards:
    - ISO 9001, ISO 9002, and ISO 9003.

ISO 9000

- Based on the premise:
  - if a proper process is followed for production:
    - good quality products are bound to follow.

ISO 9001:

- Applies to:
  - organizations engaged in design, development, production, and servicing of goods.
  - applicable to most software development organizations.

ISO 9002:

- ISO 9002 applies to:
  - organizations who do not design products:
    - but are only involved in production.
- Examples of this category of industries:
  - steel or car manufacturing industries
  - buy the product and plant designs from external sources:
    - only manufacture products.
  - not applicable to software development organizations.

ISO 9003

- ISO 9003 applies to:
  - organizations involved only in installation and testing of the products.

ISO 9000 for Software Industry

- ISO 9000 is a generic standard:
  - applicable to many industries,
    - starting from a steel manufacturing industry to a service rendering company.
- Many clauses of ISO 9000 documents:
  - use generic terminologies
  - very difficult to interpret them in the context of software organizations.

Software vs. other industries

- Very difficult to interpret many clauses for software industry:
  - software development is radically different from development of other products.

Software vs. other industries

- Software is intangible
  - therefore difficult to control.
    - It is difficult to control anything that we cannot see and feel.
  - In contrast, in a car manufacturing unit:
    - we can see a product being developed through stages such as fitting engine, fitting doors, etc.
    - one can accurately tell about the status of the product at any time.
  - Software project management is an altogether different ball game.

Software vs. other industries

- During software development:
  - the only raw material consumed is data.
- For any other product development:
  - Lot of raw materials consumed
    - e.g. Steel industry consumes large volumes of iron ore, coal, limestone, etc.
- ISO 9000 standards have many clauses corresponding to raw material control .
    - not relevant to software organizations.

Software vs. other industries

- Radical differences exist between software and other product development,
  - difficult to interpret various clauses of the original ISO standard in the context of software industry.

ISO 9000 Part-3

- ISO released a separate document called ISO 9000 part-3 in 1991
  - to help interpret the ISO standard for software industry.
- At present,
  - official guidance is inadequate

Why Get ISO 9000 Certification?

- Several benefits:
  - Confidence of customers in an organization increases
    - if organization qualified for ISO 9001 certification.
    - This is especially true in the international market.

Why Get ISO 9000 Certification?

- Many international software development contracts insist:
  - development organization to have ISO 9000 certification.

Why Get ISO 9000 Certification?

- Requires:
  - a well-documented software production process to be in place.
  - contributes to repeatable and higher quality software.
- Makes development process:
  - focussed, efficient, and cost-effective

Why Get ISO 9000 Certification?

- Points out the weakness of an organizations:
  - recommends remedial action.
- Sets the basic framework:
  - for development of an optimal process and TQM.

How to Get ISO 9000 Certification?

- An organization intending to obtain ISO 9000 certification:
  - applies to a ISO 9000 registrar for registration.
- ISO 9000 registration process consists of several stages.

How to Get ISO 9000 Certification?

- Application stage:
  - Applies to a registrar for registration.
- Pre-assessment:
  - the registrar makes a rough assessment of the organization.

How to Get ISO 9000 Certification?

- Document review and adequacy audit:
  - process and quality-related documents.
  - the registrar reviews the documents
  - makes suggestions for improvements.

How to Get ISO 9000 Certification?

- Compliance audit: the registrar checks
  - whether the suggestions made by it during review have been complied.

How to Get ISO 9000 Certification?

- Registration:
  - The registrar awards ISO 9000 certificate after successful completions of all previous phases.
- Continued surveillance:
  - The registrar continues monitoring the organization periodically.

ISO 9000 Certification

- An ISO certified organization
  - can use the certificate for corporate advertizements
  - cannot use the certificate to advertize products.
    - ISO 9000 certifies organization's process
    - not any product of the organization.
  - An organization using ISO certificate for product advertizements:
    - risks withdrawal of the certificate.

Summary of ISO 9001 Requirements

- **Management responsibility(4.1):**
  - Management must have an effective quality policy.
  - The responsibility and authority of all those whose work affects quality:
    - must be defined and documented.

Management responsibility(4.1)

- Responsibility of the quality system.
  - independent of the development process,
  - can work in an unbiased manner.
- The effectiveness of the quality system:
  - must be periodically by audited.

Quality system (4.2) and contract reviews (4.3):

- **A quality system must be maintained and documented**.
- **Contract reviews** (4.3):
  - Before entering into a contract, an organization must review the contract
    - ensure that it is understood,
    - organization has the capability for

Design control (4.4):

- The design process must be properly controlled,
  - this includes controlling coding also.
- A good configuration control system must be in place.

Design control (4.4):

- Design inputs must be verified as adequate.
- Design must be verified.
- Design output must be of required quality.
- Design changes must be controlled.

Document control (4.5):

- Proper procedures for
  - document approval, issue and removal.
- Document changes must be controlled.
  - use of some configuration management tools is necessary.

Purchasing (4.6):

- Purchased material, including bought-in software:
  - must be checked for conforming to requirements.

Purchaser Supplied Products (4.7):

- Material supplied by a purchaser,
  - for example,
    - client-provided software must be properly managed and checked.

Product Identification (4.8):

- The product must be identifiable at all stages of the process.
  - In software development context this means configuration management.

Process Control (4.9) :

- The development must be properly managed.
- Quality requirements must be identified in a quality plan.

Inspection and Testing (4.10) :

- In software terms this requires effective testing i.e.,
  - unit testing, integration testing and system testing.
- Test records must be maintained.

Inspection, measuring and test equipment(4.11):

- If integration, measuring, and test equipments are used,
  - must be properly maintained and calibrated.

Control of nonconforming product (4.13) :

- In software terms,
  - keeping untested or faulty software out of released product,
    - or other places whether it might cause damage.

Corrective Action (4.14) :

- This is both about correcting errors when found,
  - investigating why they occurred
  - improving the process to prevent further occurrences.
- If an error reoccurs despite the quality system,
  - the system needs improvement.

Handling (4.15) and Quality audits (4.17):

- Handling (4.15) Deals with:
  - storage, packing, and delivery of the software product.
- Quality Audits (4.17) :
  - quality system audit must be carried out to ensure its effectiveness.

Training (4.18) :

- Training needs must be identified and met.
- Most items of ISO standard
  - are largely common sense.

Salient features of ISO 9001 requirements:

- All documents concerned with the development of a software product
  - should be properly managed, authorized, and controlled.
- Proper plans should be prepared
  - progress against these plans should be monitored.

Salient features of ISO 9001 requirements:

- Important documents independently checked and reviewed:
  - for effectiveness and correctness.
- The product should be tested :
  - against specification.
- Several organizational aspects:
  - e.g., management reporting of the quality team.

Shortcomings of ISO 9001 Certification (1)

- ISO 9000 requires a production process to be adhered to:
  - but does not guarantee the process to be of high quality.
  - Does not give any guideline for defining an appropriate process.

Shortcomings of ISO 9001 Certification (6)

- ISO 9001 addresses mostly management aspects.
- Techniques specific to software development have been ignored
  - Configuration management
  - Reviews
  - Release builds
  - Problem Notification system

SEI Capability Maturity Model

- Developed by Software Engineering Institute (SEI) of the Carnegie Mellon University, USA:
  - to assist the U.S. Department of Defense (DoD) in software acquisition.
  - The rationale was to include:
    - likely contractor performance as a factor in contract awards.

SEI Capability Maturity Model

- Major DoD contractors began CMM-based process improvement initiatives:
  - as they vied for DoD contracts.
- SEI CMM helped organizations:
  - Improve quality of software they developed
  - Realize adoption of SEI CMM model had significant business benefits.
- Other organizations adopted CMM.

SEI Capability Maturity Model

- In simple words,
  - CMM is a model for apprising the software process maturity of a contractor into different levels.
  - Can be used to predict the most likely outcome to be expected from the next project that the organization undertakes.

SEI Capability Maturity Model

- Can be used in two ways:
  - Capability evaluation
  - Software process assessment.

Capability Evaluation

- Provides a way to **assess** the software process capability of an organization
  - Helps in selecting a contractor
  - Indicates the likely contractor performance

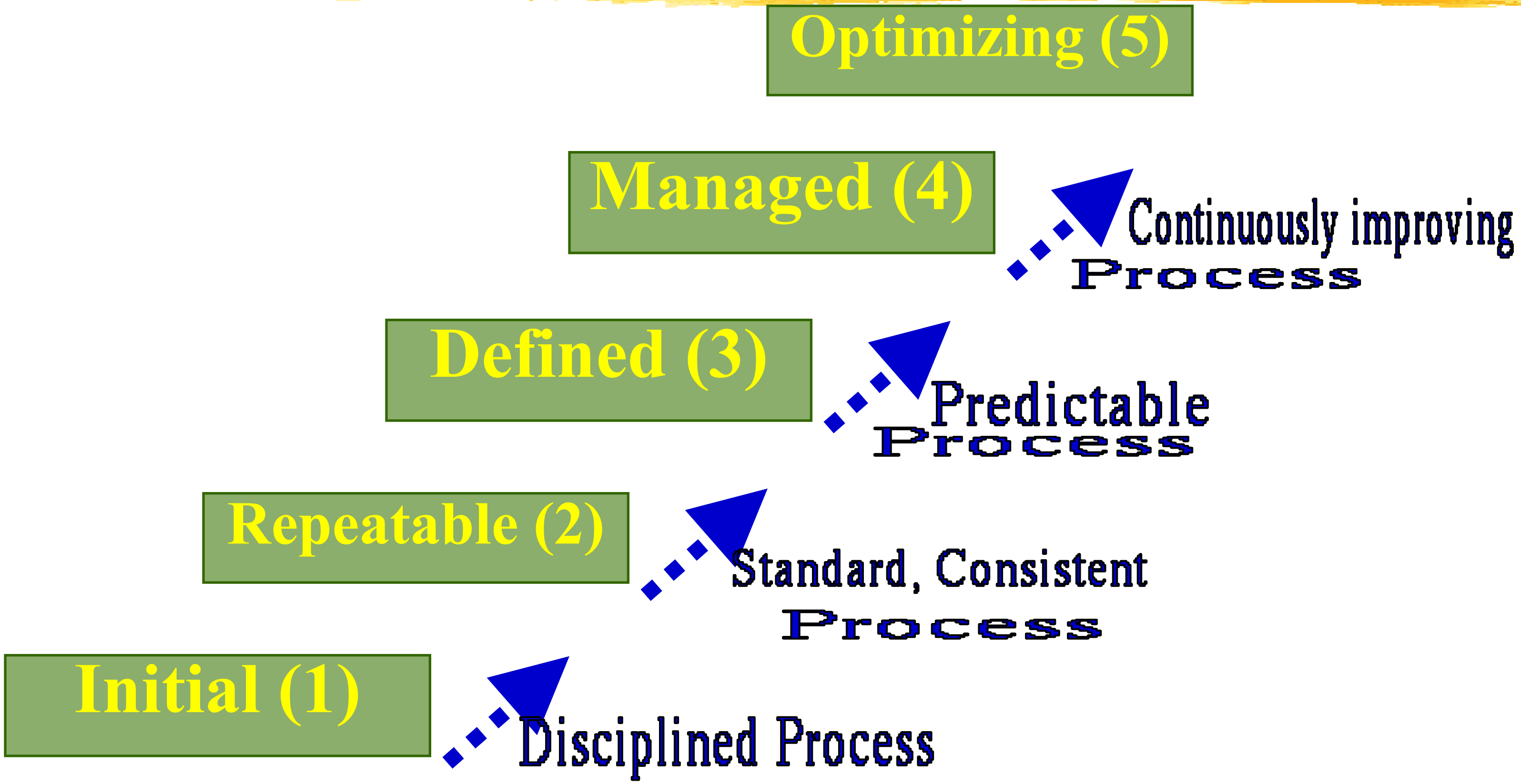
Software Process Assessment

- Used by an organization to assess its current process:
  - Suggests ways to improve the process capability.
  - This type of assessment is for purely internal use.

SEI Capability Maturity Model

- The SEI CMM classifies software development industries into:
  - Five maturity levels.
  - Stages are ordered so that improvements at one stage provide foundations for the next
  - Based on the pioneering work of Philip Crosby

SEI Capability Maturity Model



Optimizing (5)

Managed (4)

Continuously improving Process

Defined (3)

Predictable Process

Repeatable (2)

Standard, Consistent Process

Initial (1)

Disciplined Process

Level 1: (Initial)

- Organization operates
  - without any formalized process or project plans
- An organization at this level is characterized by
  - ad hoc and often chaotic activities.

Level 1: (Initial)

- Software production processes are not defined,
  - different engineers follow their own process
  - development efforts become chaotic.
  - The success of projects depend on individual efforts and heroics.

Level 2: (Repeatable)

- Basic project management practices
  - tracking cost, schedule, and functionality are followed.
- Size and cost estimation techniques
  - function point analysis, COCOMO, etc. used.
- Production process is ad hoc
  - not formally defined
  - also not documented.

Level 3: (Defined)

- Management and development activities:
  - defined and documented.
  - Common organization-wide understanding of activities, roles, and responsibilities.

Level 3: (Defined)

- The process though defined,
  - process and product qualities are not measured.
- ISO 9001 aims at achieving this level.

Level 4: (Managed)

- Quantitative quality goals for products are set.
- Software process and product quality metrics are collected:

Level 4: (Managed)

- Organization sets quantitative quality goals
- World-wide about 100 organizations assessed at this level.

Level 5: (Optimizing)

- ⌘Statistics collected from process and product measurements are analyzed:
  - continuous process improvement based on the measurements.
    - ☒Known types of defects are prevented from recurring by tuning the process
    - ☒lessons learned from specific projects incorporated into the process

Level 5: (Optimizing)

- Identify best software engineering practices and innovations:
  - tools, methods, or process are identified
  - transferred throughout the organization
- World-wide about 50 organizations have been assessed at this level.

Key Process Areas

- Each level is associated with a key process area (KPA) identifies
  - where an organization at the previous level must focus to reach this level

Level 2 KPAs

- Software project planning
  - Size, cost, schedule.
  - project monitoring
- Configuration management
- Subcontract management

Level 3 KPAs

- Process definition and documentation
- Reviews
- Training program

Level 4 KPAs

- Quantitative measurements
- Process management

Level 5 KPAs

- Defect prevention
- Technology change management
- Process change management

Shortcomings of CMM

1. Focuses on what needs to improve but does not guide about how to improve.
2. Thicker documentations and longer meetings are required
3. Getting an accurate measure of an organization’s current maturity level is also an issue

Comparison between ISO 9001 and SEI CMM

- ISO 9001 awarded by an international standards body
  - can be quoted in official documents and communications
- SEI CMM assessment is purely for internal use.

Comparison between ISO 9001 and SEI CMM

- SEI CMM was developed specifically for software industry:
  - addresses many issues specific to software industry.
- SEI goes beyond quality assurance
  - aims for TQM
  - ISO 9001 correspond to SEI level 3.

Comparison between ISO 9001 and SEI CMM

- SEI CMM provides a list of key areas
  - on which to focus to take an organization from one level to the other
- Provides a way for gradual quality improvements over several stages.
  - e.g trying to implement a defined process before a repeatable process:
    - counterproductive as managers are overwhelmed by schedule and budget pressure.

CMMI

- It is successor of CMM.
- CMM developed from 1987-1997
- In 2002, CMMI version 1.1 was released
- CMMI aimed to improve the maturity models by integrating many different models into one framework
- CMM versions are: SE-CMM, T-CMM, SSE-CMM, P-CMM, IPD-CMM etc.
- CMMI integrated like: CMMI-SE/SW, CMMI-SE/SW/IPD etc.

Remarks on Quality Model Usage

- Highly systematic and measured approach to software development process suits certain circumstances
  - negotiated software, safety-critical software, etc
- What about small organizations?
  - Typically handle applications such as internet, e-comm.
  - without an established product range,
  - without revenue base, experience on past projects, etc.
  - CMM may be incompatible

Small Organizations

- Small organizations tend to believe:
  - We are all competent people hired to do a job, we can’t afford training
  - We all communicate with one another
    - Osmosis works because we are so close
  - We are all heroes
    - We do what needs to be done
    - Therefore rules do not apply to us

Small Organizations

- Often have problems:
  - Undocumented requirements
  - Inexperienced managers
  - Documenting the product
  - Resource allocation
  - Training
  - Peer reviews

Small Organizations

- A two week CMM-based appraisal is probably excessive:
- Small organizations need to operate more efficiently at lower levels of maturity
  - Must first fluorish if eventually they are to mature

Personal Software Process (PSP)

- Based on the work of Humphrey
- PSP is a scaled down version of industrial software process
  - suitable for individual use
- Even CMM assumes that engineers use effective personal practices

Personal Software Process (PSP)

- A process is the set of steps for doing a job
- The quality and productivity of an engineer
  - largely determined by his process
- PSP is framework that
  - helps software engineers to measure and improve the way they work.

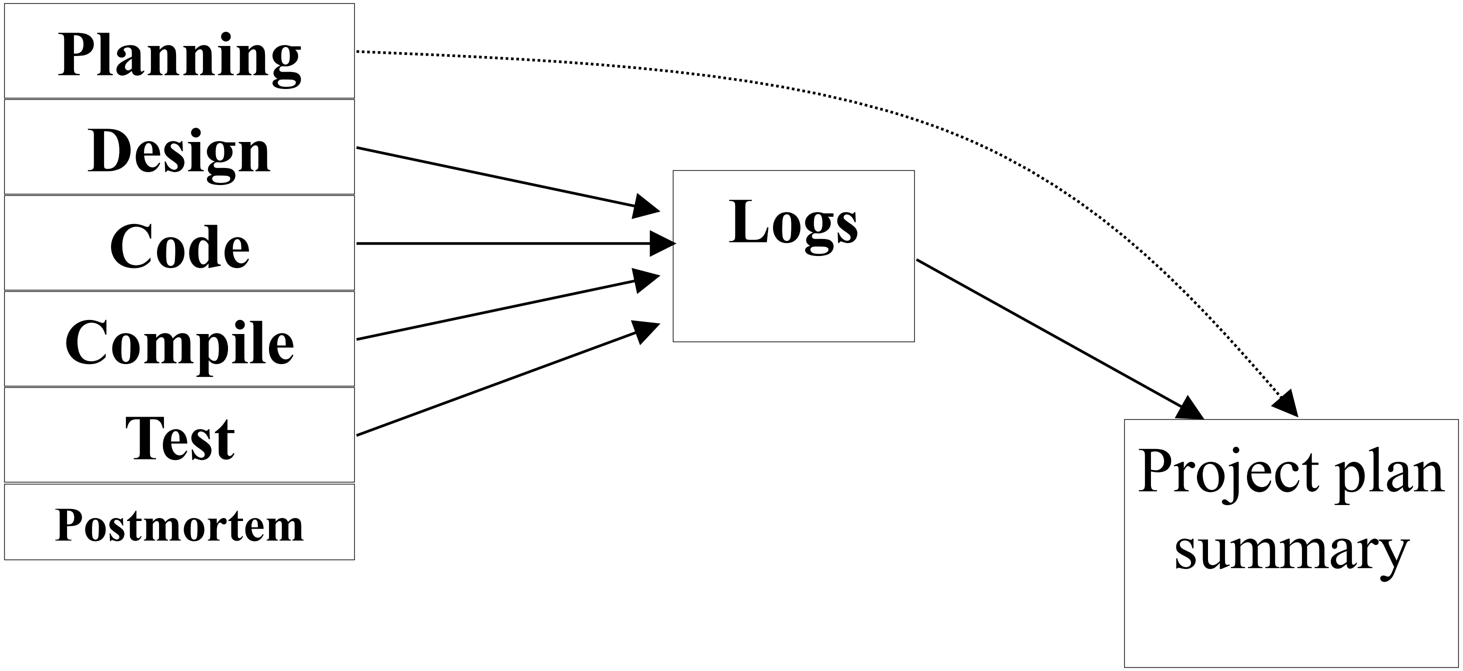
Personal Software Process (PSP)

- Helps developing personal skills and methods
  - Estimating and planning method
  - Shows how to track performance against plans
  - Provides a defined process
    - can be fine tuned by individuals
    - Recognizes that a process for individual use is different from that necessary for a team project.

Time Management

- Track the way you spend time
  - Boring activities seem longer then actual
  - Interesting activities seem short
- Record time for
  - Designing
  - Writing code
  - Compiling
  - Testing

Personal Software Process (PSP)



PSP-Planning

- Problem definition
- Estimate max, min, and total LOC
- Determine minutes/LOC
- Calculate max,min, and total development times
- Enter the plan data in project plan summary form
- record the planned time in Log

PSP-Design

- Design the program
- Record the design in specified format
- Record the Design time in time recording log

PSP-Code

- Implement the design
- Use a standard format for code text
- Record the coding time in time recording log

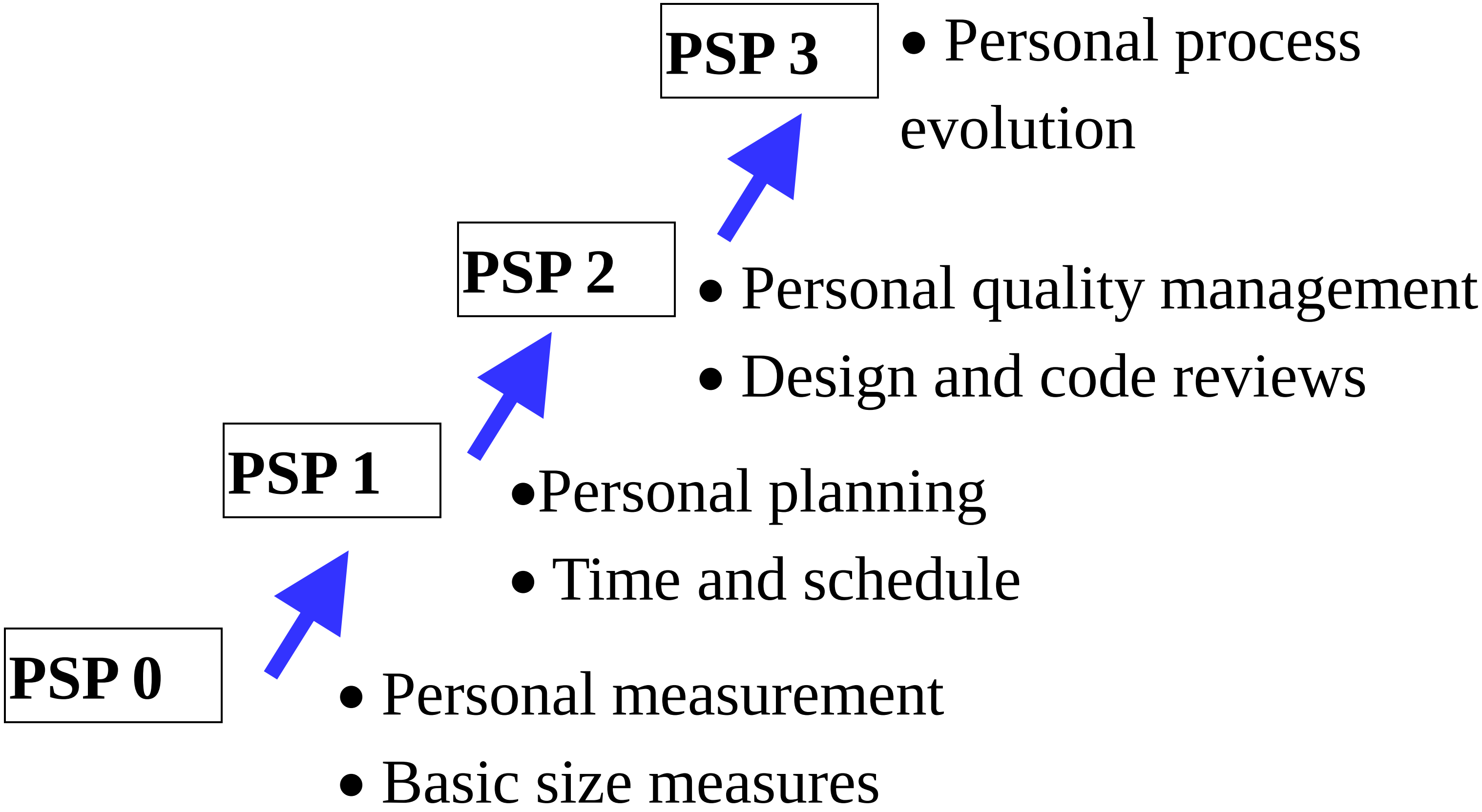
PSP-Compile

- Compile the program
- Fix all the defects
- Record compile time in time recording log

PSP-Test/Postmortem

- Test
  - Test the program
  - Fix all the defects found
  - Record testing time in time recording log
- Postmortem
  - Compare the log data with project plan to achieve better planning in future projects to improve his process.

Personal Software Process (PSP)



PSP 3

- Personal process evolution

PSP 2

- Personal quality management
- Design and code reviews

PSP 1

- Personal planning
- Time and schedule

PSP 0

- Personal measurement
- Basic size measures

Six Sigma

- Six sigma is a quantitative approach to eliminate defects
  - Applicable to all types of industry - from manufacturing, product development, to service
- The statistical representation of Six Sigma quantitatively describes
  - how a process is performing

Six Sigma

- To achieve six sigma
  - a process must not produce more than 3.4 defects per million opportunities.
  - 5 Sigma -> 230 defects per million
  - 4 Sigma -> 6210 defects per million
- Six sigma methodologies
  - DMAIC (Define, Measure, Analyze, Improve, Control)
  - DMADV: (Define, Measure, Analyze, Design, Verify)

Six Sigma Methodologies

- The methodologies are implemented by Green belt and Black belt workers
  - Supervised by Master black belt worker

Summary

- Evolution of quality system:
  - product inspection
  - quality control
  - quality assurance
  - total quality management (TQM)
- Quality paradigm change:
  - from product to process

Summary

- ISO 9000:
  - basic premise:
    - if a good process is followed
    - good products are bound to follow
  - provides guidelines for establishing a quality system.

Summary

- ISO 9000
  - series of three standards
    - 9001, 9002, and 9003
  - 9001 is applicable to software industry

Summary

- SEI CMM
  - developed specially for software industry
  - classifies software organizations into five categories.
    - According to the maturity of their development process.

Current Trends

- Many organizations have already tuned their process for
  - Budget,
  - Schedule, and
  - Quality product.
- Competition is challenging them to:
  - Reduce time for delivery
  - Adopt Six-Sigma methodology