



SE-3002

SOFTWARE QUALITY ENGINEERING

RUBAB JAFFAR

RUBAB.JAFFAR@NU.EDU.PK

SQA Plan and Development Plan

Overview and Basics

Lecture # 7, 8, 9

TODAY'S OUTLINE

- SQA Activities
- SQA plan
- Project plan
- SQA metrics

QUALITY ENGINEERING

- Different customers and users have different quality expectations under different market environments. Therefore, we need to move beyond just performing QA activities toward quality engineering by managing these quality expectations as an engineering problem.
 - A goal might be to minimize the cost and project risk

SOFTWARE QUALITY ENGINEERING

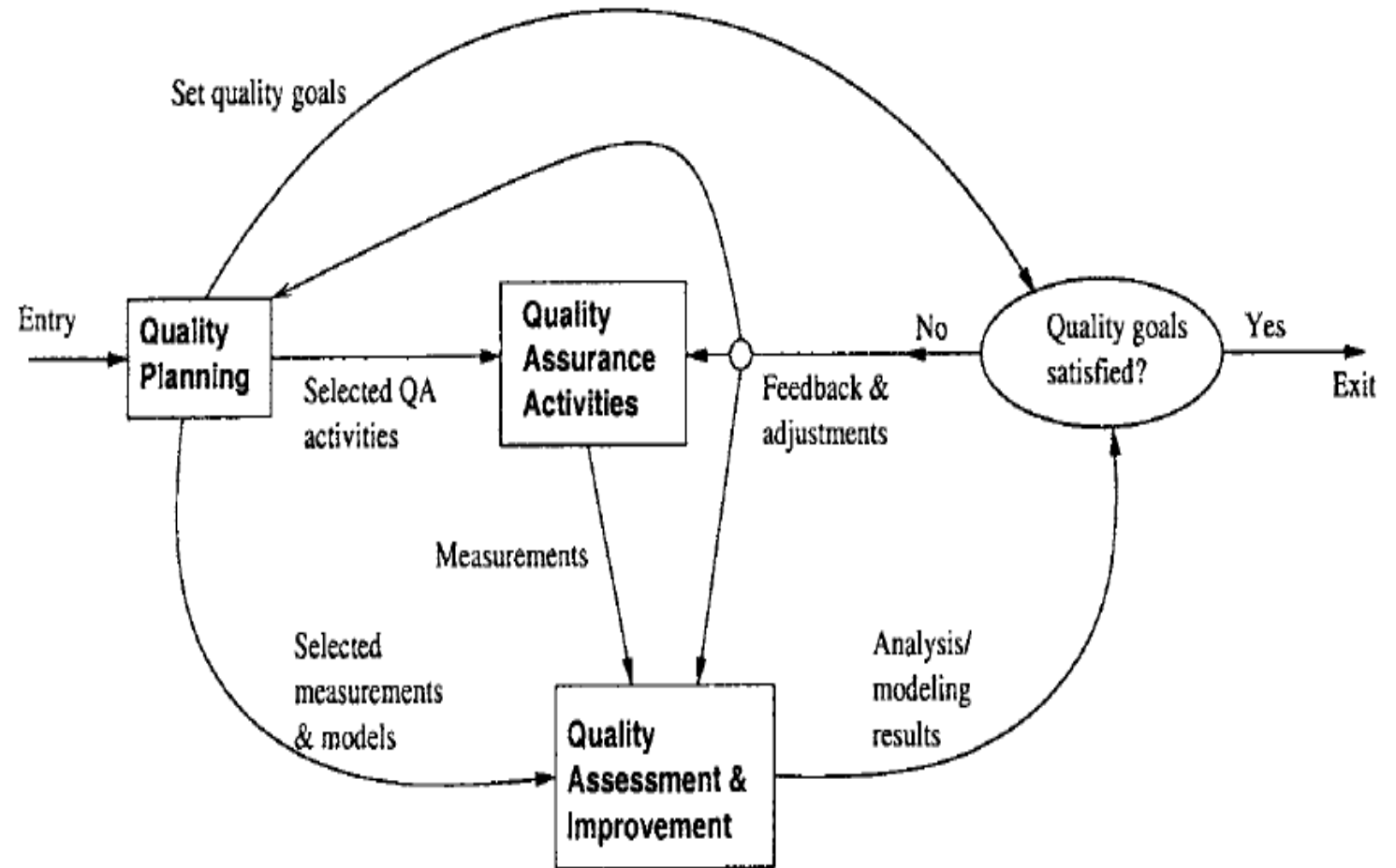
- 1) The application of a continuous, systematic, disciplined, quantifiable approach to the development and maintenance of quality throughout the whole life cycle of software products and systems; that is, the application of quality engineering to software,
- 2) The study of approaches as in (1).

SQE- COOPERATION



QUALITY ENGINEERING: ACTIVITIES AND PROCESS

- There are three major groups of activities in the quality engineering process:
 - pre-QA activities, in-QA activities, and post-QA activities



QUALITY ENGINEERING: ACTIVITIES AND PROCESS

- ***Pre-QA activities:***

- **Quality planning**

- These are the activities that should be carried out before carrying out the regular QA activities. There are two major types of pre-QA activities in quality planning, including:
 - Set specific quality goals.
 - Form an overall QA strategy, which includes two sub-activities:
 - Select appropriate QA activities to perform.
 - Choose appropriate quality measurements and models to provide feedback, quality assessment and improvement.

PRE-QA PLANNING

■ Pre-QA planning:

- Quality goal
- Overall QA strategy:
- QA activities to perform?
- measurement/feedback planning

■ Setting quality goal(s):

- Identify quality views/attributes
- Select direct quality measurements
- Assess quality expectations vs. cost

■ Identify quality views/attributes

- customer/user expectations,
- market condition,
- product type, etc.

■ Select direct quality measurements

- direct: reliability
- defect-based measurement
- other measurements

■ Assess quality expectations vs. cost

- cost-of-quality/defect studies
- economic models: COCOMO etc

QUALITY ENGINEERING: ACTIVITIES AND PROCESS

- ***In-QA activities:***
 - **Executing planned QA activities and handling discovered defects**
 - In addition to performing selected QA activities, an important part of this normal execution is to deal with the discovered problems.

QUALITY ENGINEERING: ACTIVITIES AND PROCESS

- **Post-QA activities:**
 - **Quality measurement, assessment and improvement**
 - The primary purpose of these activities is to provide quality assessment and feedback so that various management decisions can be made and possible quality improvement initiatives can be carried out
- **Short-term feedback**
 - provides information for progress tracking, activity scheduling, and identification of areas that need special attentions
- **Long-term feedback**
 - Feedback to quality planning so that necessary adjustment can be made to quality goals and QA strategies
 - Feedback to the quality assessment and improvement activities

QUALITY ENGINEERING AND QUALITY IMPROVEMENT PARADIGM (QIP)

- The Quality Improvement Paradigm (QIP) suggests the following activities to achieve quality improvement
 - measurement, analysis, feedback, and organizational support
- QIP includes three interconnected steps: understanding, assessing, and packaging, which form a feedback and improvement loop, as briefly described below:
 - The first step is to understand the baseline so that improvement opportunities can be identified and clear measurable goals can be set. All future process changes are measured against this baseline.
 - The second step is to introduce process changes through experiments, pilot projects, assess their impact, and fine tune these process changes.
 - The last step is to package baseline data, experiment results, local experience, and updated process as the way to infuse the findings of the improvement program into the development organization.

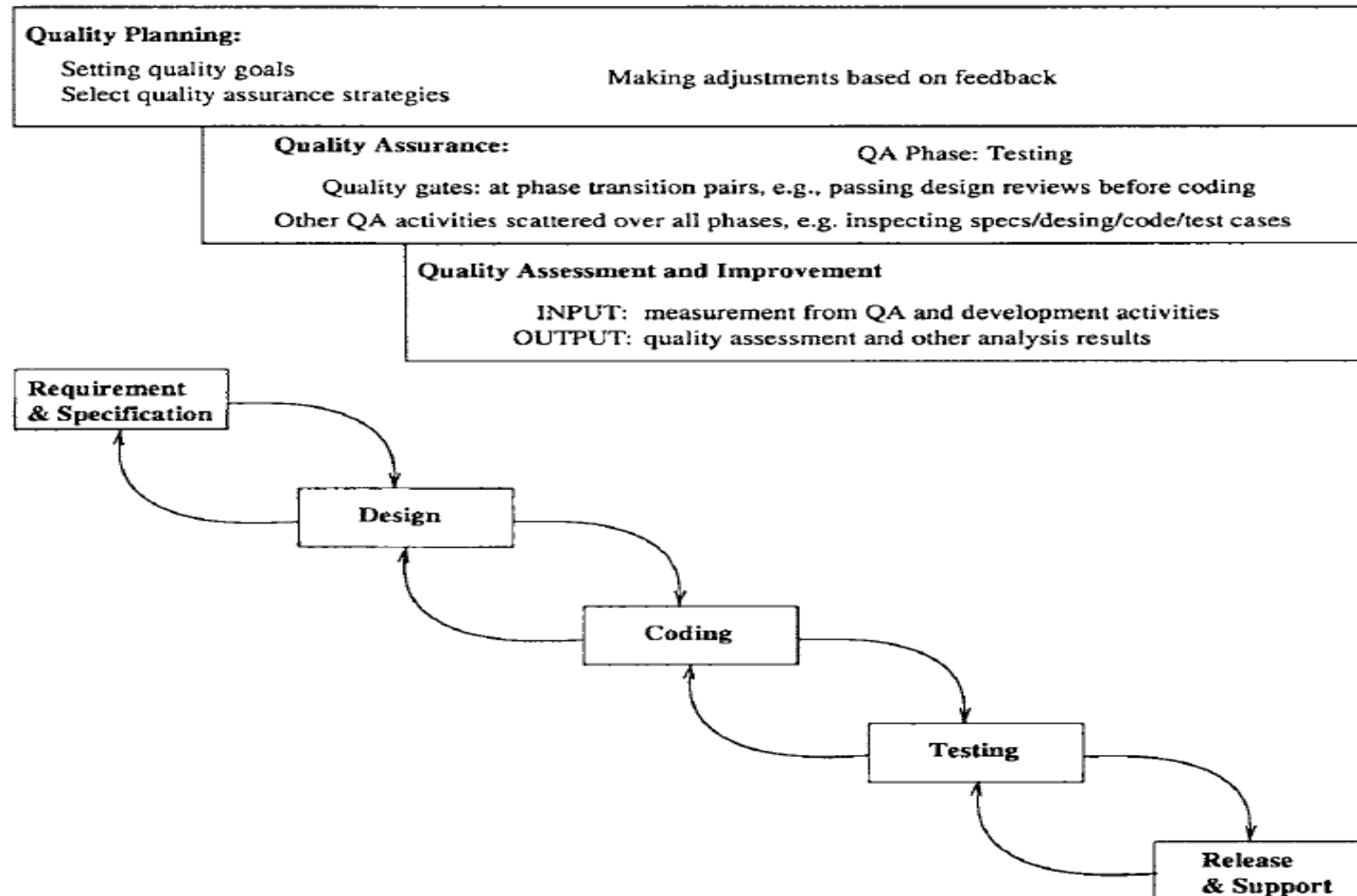
QUALITY ENGINEERING IN SOFTWARE PROCESS

- The quality engineering process forms an integral part of the overall software engineering process, where other concerns, such as cost and schedule, are also considered and managed.
- Individual QA activities can be carried out and integrated into the software process. When we broaden our scope to quality engineering, it also covers pre-QA quality planning as well as the post-QA measurement and analysis activities carried out parallel to and after QA activities to provide feedback and other useful information.
- All these activities and the quality engineering process can be integrated into the overall software process.

QUALITY ENGINEERING IN SOFTWARE PROCESS

- ❑ SQE activities \subset development activities:
 - ❑ quality planning \subset product planning
 - ❑ QA activities \subset development activities
 - ❑ analysis/feedback \subset project management
- ❑ Fitting SQE in software processes:
 - ❑ different start/end time
 - ❑ different sets of activities, sub-activities, and focuses
 - ❑ In waterfall process: more staged (planning, execution, analysis/feedback)
 - ❑ In other processes: more iterative or other variations

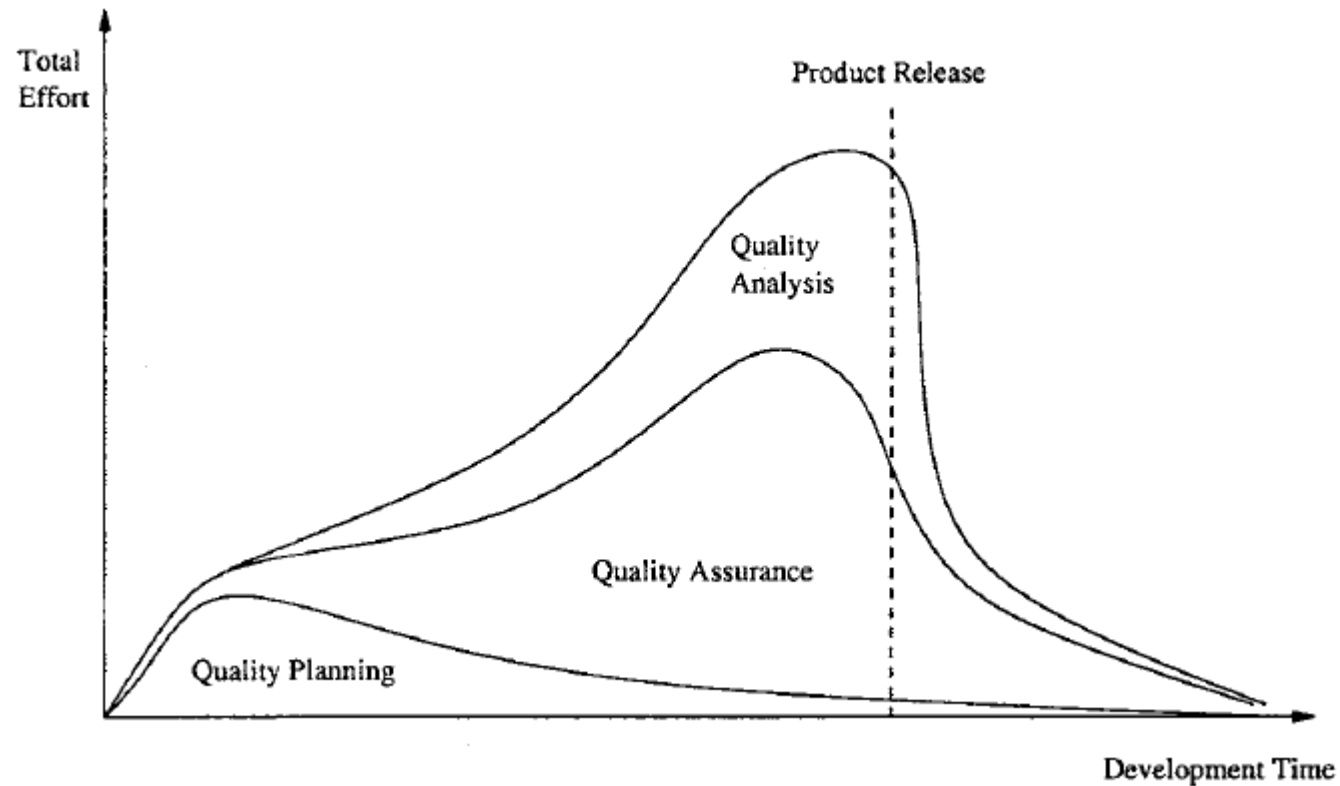
QUALITY ENGINEERING IN THE WATERFALL PROCESS



EFFORT PROFILE

- Among the three major types of activities in the quality engineering process, the execution of specific QA activities is central to dealing with defects and assuring quality for the software products. Therefore, they should and normally do consume the most resources in terms of human effort as well as utilization of computing and other related resources.
- However, the effort distribution among the three is not constant over time because of the process characteristics described above and the shifting focus over time.

EFFORT PROFILE



The exact profile based on real data would not be as smooth and would naturally show large amount of variability, with many small peaks and valleys. But the general shape and pattern should preserve.

EFFORT PROFILE

- Waterfall process would see more dominance of quality planning in the beginning, and dominance of testing near product release, and measurement and quality assessment activities peak right before product release.
- Other development processes, such as incremental, iterative, spiral, and extreme programming processes, would be associated with curves that vary less between the peaks and valleys. QA is spread out more evenly in these processes than in the waterfall process, although it is still expected to peak a little bit before product release.

SQA TASKS, GOALS, ATTRIBUTES, AND METRICS

- Software quality assurance (SQA) is a process which assures that all software engineering processes, methods, activities and work items are monitored and comply against the defined standards. These defined standards could be one or a combination of any like ISO 9000, CMMI model, ISO 15504, etc.
- SQA incorporates all software development processes starting from defining requirements to coding until release. Its prime goal is to ensure quality.

SQA GOALS

- Requirement quality : The correctness, completeness, and consistency of the requirements model will have a strong influence on the quality of all work products that follow. SQA must ensure that the software team has properly reviewed the requirements model to achieve a high level of quality.
- Design quality : Every element of the design model should be assessed by the software team to ensure that it exhibits high quality and that the design itself conforms to requirements.

SQA GOALS

- Code quality : Source code and related work products must conform to local coding standards and exhibit characteristics that will facilitate maintainability.
- Quality control effectiveness : A software team should apply limited resources in a way that has the highest likelihood of achieving a high-quality result. SQA analyzes the allocation of resources for reviews and testing to assess whether they are being allocated in the most effective manner.

SOFTWARE QUALITY GOALS, ATTRIBUTES, AND METRICS

Goal	Attribute	Metric
Requirement quality	Ambiguity	Number of ambiguous modifiers
	Completeness	Number of TBA, TBD
	Understandability	Number of sections/subsections
	Volatility	Number of changes per requirement Time (by activity) when change is requested
	Traceability	Number of requirements not traceable to design/code
	Model clarity	Number of UML models Number of descriptive pages per model Number of UML errors

SOFTWARE QUALITY GOALS, ATTRIBUTES, AND METRICS

Design quality	Architectural integrity	Existence of architectural model
	Component completeness	Number of components that trace to architectural model, Complexity of procedural design
	Interface complexity	Average number of pick to get to a typical function or content Layout appropriateness
	Patterns	Number of patterns used

SOFTWARE QUALITY GOALS, ATTRIBUTES, AND METRICS

Code quality	Complexity Maintainability Understandability Reusability Documentation	Cyclomatic complexity Design factors Percent internal comments Variable naming conventions Percent reused components Readability index
--------------	--	---

SOFTWARE QUALITY GOALS, ATTRIBUTES, AND METRICS

QC effectiveness	Resource allocation Completion rate	Staff hour percentage per activity Actual vs. budgeted completion time
	Testing effectiveness	Number of errors found and criticality Effort required to correct an error Origin of error

SOFTWARE QUALITY ASSURANCE (SQA)

- Software Quality Assurance (SQA) consists of a means of monitoring the software engineering processes and methods used to ensure quality.
- The methods by which this is accomplished are many and varied, and may include ensuring conformance to one or more standards, such as ISO 9000 or a model such as CMMI.
- SQA encompasses the entire software development process
 - such as requirements definition, software design, coding, source code control, code reviews, software configuration management, testing, release management, and product integration.
- SQA is organized into goals, commitments, abilities, activities, measurements, and verifications.

SQA–THE IEEE DEFINITION

- Software quality assurance is:
 1. A **planned** and **systematic pattern** of all **actions** necessary to provide adequate **confidence** that an item or product conforms to established technical requirements.
 2. A **set of activities** designed to evaluate the **process** by which the products are developed or manufactured. Contrasts with quality control.

SOFTWARE QUALITY ASSURANCE PLAN

- Abbreviated as SQAP, the software quality assurance plan comprises of the procedures, techniques, and tools that are employed to make sure that a product or service aligns with the requirements defined in the SRS (software requirement specification).
- The plan identifies the SQA responsibilities of a team, lists the areas that need to be reviewed and audited. It also identifies the SQA work products.

SQA PLAN SECTION

- Purpose section
- Reference section
- Software configuration management section
- Problem reporting and corrective action section
- Tools, technologies and methodologies section
- Code control section
- Records: Collection, maintenance and retention section
- Testing methodology

ELEMENTS OF SQA

- Standards
- Reviews and audits
- Testing
- Error/defect collection and analysis
- Change management
- Educational programs
- Vendor management
- Security management
- Safety
- Risk management

SQA ACTIVITIES

- The charter of the SQA group is to assist the software team in achieving a high quality end product. The Software Engineering Institute recommends a set of SQA activities that address quality assurance planning, oversight, record keeping, analysis and reporting.
- It is these activities that are performed (or facilitated) by an independent SQA group that
 - Prepares an SQA plan for a project
 - Participates in the development of the project's software process description
 - Reviews software engineering activities to verify compliance with the defined software process
 - Audits designated software work products to verify compliance with those defined as part of the software process
 - Ensures that deviations in software work and work products are documented and handled according to a documented procedure
 - Records any noncompliance and reports to senior management.

SQA ACTIVITIES

- Prepares an **SQA plan** for a project
 - The plan is developed during project planning and is reviewed by all interested parties. Quality assurance activities performed by the software engineering team and the SQA group are governed by the plan.
 - The plan identifies many important aspects of QA summarized below:
 - 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 software project team

SQA ACTIVITIES

- Participates in the **development** of the project's software process description
 - The software team selects a process for the work to be performed.
 - The SQA group reviews the process description for compliance with organizational policy, internal software standards, externally imposed standards (e.g. ISO-9001) and other parts of the software project plan.
- Reviews **software engineering activities** to verify compliance with the defined software process
 - The SQA group reviews selected work products, identifies, and tracks deviations.
- Audits designated **software work products** to verify compliance with those defined as part of the software process
 - verifies that corrections have been made and periodically reports the results of its work to the project manager

SQA ACTIVITIES

- Ensures that **deviations** in software work and work products are documented and handled according to a documented procedure
 - Deviations may be encountered in the project plan, process description, applicable standards or technical work products
- **Records any noncompliance and reports** to senior management
 - Noncompliance items are tracked until they are resolved.

PLANS-INTRODUCTION

- Project managers prepare
 - **development** and
 - **quality plans**
- Arduous task,
 - Senior level **management** on one end and
 - **Developers** on the other
- These plans are **vitaly important** to meet contractual commitments.
- Thus, we need to look at both development and quality plans.

INTRODUCTION...

- We have proposal plans and internal documents. **Enough?** No!
- These plans typically included time tables, estimates, staffing requirements, scheduled reviews, risks and more.
 - Seems like enough....
- But the **time invested** in developing a development plan and a quality plan will play dividends.

INTRODUCTION...

- The project needs development and quality plans that:
 - Are based on proposal materials that have been re-examined and thoroughly updated.
 - Are more comprehensive than the approved proposal, especially with respect to schedules, resource estimates, and development risk evaluations.
 - Were prepared at the beginning of the project to sound alerts regarding scheduling difficulties, potential staff shortages, scarcity of development facilities, problems with meeting contractual milestones, modified development risks, and so on.

INTRODUCTION...

- Development and quality **standards** (ISO 9000.3 and CMM) require viable plans.
- We need to look at development plans and quality plans – their **objectives** and **elements**.
- They are **related** but NOT the same.

DEVELOPMENT AND QUALITY PLANS - OBJECTIVES

- Planning is meant to prepare adequate foundations for successful and timely completion of the project. The planning process includes:
 - Scheduling development activities and estimating the required manpower resources and budget
 - Recruiting team members and allocating development resources
 - Resolving development risks
 - Implementing required SQA activities
 - Providing management with data needed for project control

ELEMENTS OF THE DEVELOPMENT PLAN

1. Project products, specifying “deliverables”
2. Project interfaces
3. Project methodology and development tools
4. Software development standards and procedures
5. Map of the development process
6. Project milestones
7. Project staff organization
8. Required development facilities
9. Development risks and risk management actions
10. Control methods
11. Project cost estimates

ELEMENTS OF THE DEVELOPMENT PLAN

- **Project Products, specifying “Deliverables”**
 - Must specify items to be **delivered to customer**
 - documents, user manuals
 - Must specify specific software products (along with completion and installation dates)
 - Must specify or discuss training.
 - Who and how does training take place?
 - Must specify **customer support!**

EXAMPLES OF DELIVERABLES

No.	Deliverables/Release	Media	Qty	Due date
1	Proposal	Hard copy	4	12-12-2011
2	SRS	Hard copy	1	20-12-2011
3	Acceptance test report	Hard copy	1	30-01-2012
4	Software	DVD	10	1-02-2012

ELEMENTS OF THE DEVELOPMENT PLAN

■ Project Interfaces

- Interfaces with existing software packages (software interface)
 - (A course enrollment system might interface with an existing Billing System or Course Scheduling System...)
- Interfaces with other software / hardware development and maintenance teams working on similar system or extension of the system (cooperation and coordination links)
- Interface with existing or new hardware (hardware interface).

ELEMENTS OF THE DEVELOPMENT PLAN

■ Project **methodology** and **development tools**

- Process used, tools / environment needed
- Requirements capture and technologies used
- Design approaches
 - architectural; interface; communications; databases
 - Programming methodology
- Testing Approaches etc.
 - What are the testing responsibilities and who does what? Individual testing? Separate testing shop?
- Deployment
 - One shot; parallel; incremental...

■ Software Development **Standards** and **Procedures**

- A list should be prepared of the software development standards and procedures to be applied in the project

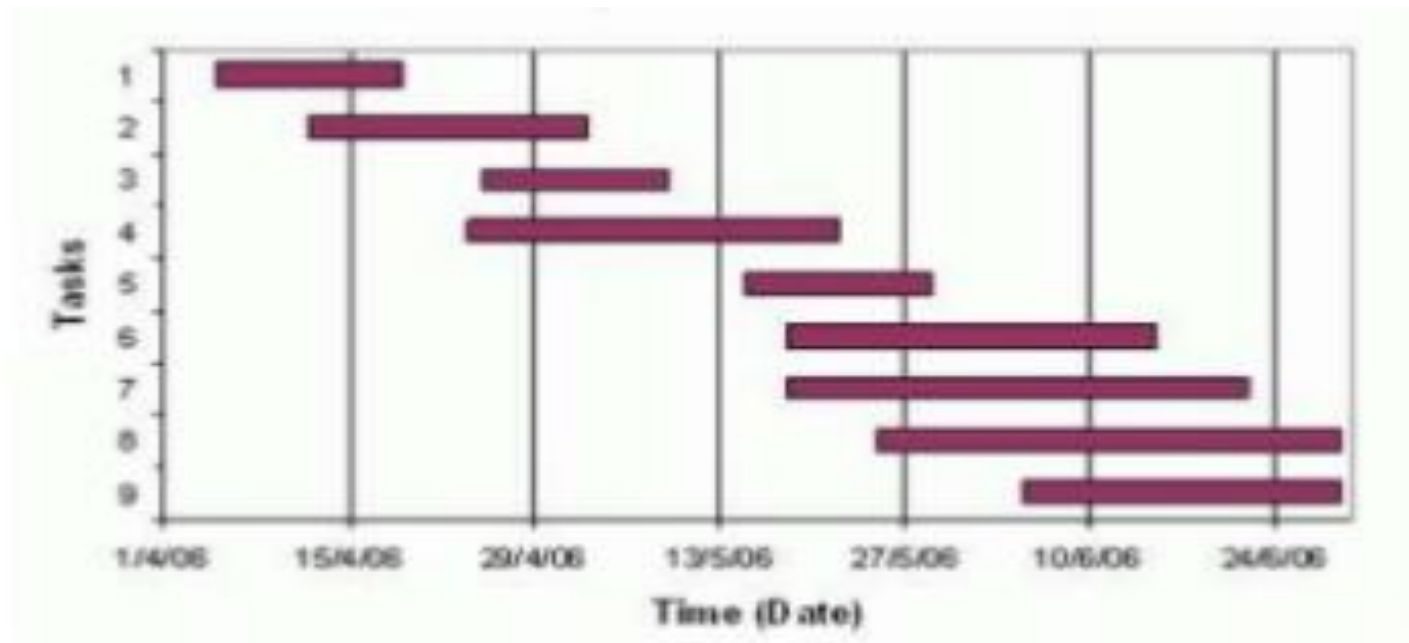
ELEMENTS OF THE DEVELOPMENT PLAN

- **Map of the Development Process**
 - Detailed definition of project's phases:
 - inputs, activities, outputs (artifacts), specific activities
 - Estimates of each activity's duration:
 - design reviews (managerial and technical), tests, design and code correction activities
 - Sequencing and dependency of activities.
 - List of professional resources needed overall and for each activity.
 - Can show these in GANTT charts, which show various activities by horizontal bars and whose lengths are proportional to the activity's duration.

ELEMENTS OF THE DEVELOPMENT PLAN

- Can use PERT and CPM and other activities to communicate activities, durations, deliverable dates, ...
 - Both the PERT and CPM consider sequence dependencies into account in addition to duration of activities.
 - They enable calculation of the earliest and latest acceptable start times for each activity.
 - The difference between start times determines the activity's scheduling flexibility. Special attention is awarded to those activities lacking scheduling flexibility (which explains their being called “critical path” activities), and whose delayed completion may cause delay in the conclusion of the entire project.
- Some like Microsoft Project.
- Can use more modern tools too, like IBM's Rational Team Concert (RTC).

GANTT CHART



ELEMENTS OF THE DEVELOPMENT PLAN...

- **Project Milestones**

- For each milestone, its completion time and project products (documents and code) are to be defined.

ELEMENTS OF THE DEVELOPMENT PLAN

- **Project staff organization and coordination with external participants**
 - Organizational structure – defines teams and tasks;
 - Defines expertise needed (certifications, experience, specialties), programming languages, development tools, levels of expertise, numbers of individuals needed and for specific periods of time; names of team leaders and team members (sometimes).
 - Long term leadership; team losses due to many factors;
 - Estimates of staff availability is crucial and can cause the flag to be raised when certain levels are not met.

ELEMENTS OF THE DEVELOPMENT PLAN

■ Required development facilities

- Required hardware, software, tools, space, infrastructure, ...

■ Development Risks and Risk Management Actions

- Development Risk: *“a state or property of a development task or environment, which, if ignored, will increase the likelihood of project failure”*.

■ Risk Areas

- Technology Risks – lack of expertise; not correct / needed tools
- Personnel / Staff Shortages – loss of people; inability to recruit
- Environmental Risk:
 - Where / how is application to be deployed?
 - Where / how is app to be developed?
 - Financial Risks

- Interdependence of other organizational elements who supply resources (subcontractors, specialized hardware, etc.)

ELEMENTS OF THE DEVELOPMENT PLAN

■ Control methods

- Progress reports and coordination meetings are **planned** to control project implementation

■ Project Cost Estimates

- These are based on **proposal cost estimates** followed by thorough review and continuous updating.
- Changes can/will occur and these can be **major budget impacts**, such as subcontractors don't fulfill their obligations or other unplanned expenditures arise.
- Some projects are '**successes**' but way over budget.
- Ultimately, the **approval** of the development plan will take place within the organization(s).

CLASS ACTIVITY I

- George Wise is an exceptional programmer. Testing his software modules reveals very few errors, far fewer than the team's average. He keeps his schedule promptly, and only rarely is he late in completing his task. He always finds original ways to solve programming difficulties, and uses an original, individual version of the coding style. He dislikes preparing the required documentation, and rarely does it according to the team's templates.
- A day after completing a challenging task, on time, he was called to the office of the department's chief software engineer. Instead of being praised for his accomplishments (as he expected), he was warned by the company's chief software engineer that he would be fired unless he began to fully comply with the team's coding and documentation instructions.
 1. Do you agree with the position taken by the department's chief software engineer?
 2. If yes, could you suggest why his or her position was so decisive?

CASE STUDY: IS DEVELOPMENT DEPARTMENT MANAGER RIGHT?

- Imagine that you have been appointed as a head of a big project. As it often is the case in the software industry, you come under serious time pressure from the very first day. Because you were MEMBER of the proposal team and participated in most of the meetings held with customer, you are confident that you know all that is necessary to do the job. You INTEND to use the proposal plans and internal documents that the team had prepared as your DEVELOPMENT and QUALITY plans. You are prepared to RELY on these materials because you know that the proposal and its estimates, including the timetable, staff requirements, list of project documents, scheduled design reviews, and list of development risk.
- Development Department Manager asked you to develop updated development plan and quality plan.
- You, therefore, were a bit disappointed that at this CRUCIAL point of the project, the Development Department Manager demands that you immediately prepare new and separate project development plans (“Development Plan”) and project quality plan (“Quality Plan”). When you claim that the completed proposal could serve as the requested plans. BUT the manager still insists that they must be updated with new and more comprehensive topics should be added to plan. *“By the way , don’t forget that a period of 7 months has gone between the proposal preparation and the final signing of the contract”* mentioned the manager.

DEVELOPMENT DEPARTMENT MANAGER IS RIGHT?

- Because, it may be discovered that some team members will not be available at the scheduled dates due to delays in completion of their current assignment or
- The consulting company that had agreed to provide professional support in a specialized and crucial area has gone bankrupt.



That is all