Software Engineering

SOFTWARE PROJECT PLANNING - V

Software Quality Assurance Plans
Project Monitoring Plans

Gagan Deep
Founder & Director
Rozy Computech Services
Kurukshetra, rozygag@yahoo.com
www.rozyph.com

SOFTWARE QUALITY ASSURANCE PLANS

What is Quality?

- The American Heritage Dictionary defines *Quality as a characteristic or attribute of something.*
- As an attribute of an item, quality refers to measurable characteristics - things we are able to compare to known standards such as length, color, electrical properties, and malleability.
- ***Quality of design*** refers to the characteristics that designers specify for an item. The grade of materials, tolerances, and performance specifications all contribute to the quality of design.

- ***Quality control*** involves the series of inspections, reviews, and tests used throughout the software process to ensure each work product meets the requirements placed upon it. Quality control includes a feedback loop to the process that created the work product.
- **Quality assurance** is the preventive set of the activities that provide greater confidence that the project will be completed successfully .
- **Quality assurance** focuses on, *how the engineering and management activity will be done?*
- As any one is interested in the quality of final product, it should be assured that we are building the right product.
- It can be assured only when we do inspection & review of intermediate products, if there are any **bugs**(error) then it is **debugged**(removed). Thus quality can be enhanced.

Importance of Quality

- We would expect quality to be a concern of all producers of goods and services. However, the special characteristics of software and in particular its intangibility and complexity, make special demands.
- ***Increasing criticality of software*** : The final customer or user is naturally concerned about the general quality of software, especially its reliability. This is increasing in the case as organizations become more dependent on their computer systems and software is used more and more in areas that are safety-critical. For example to control aircraft.

- ***The intangibility of software :*** This makes it difficult to know that a particular task in a project has been completed satisfactorily. The results of these tasks can be made tangible by demanding that the developer produce 'deliverables' that can be examined for quality.
- ***Accumulating errors during software development:*** As computer system development is made up of a number of steps where the output from one step is the input to the next, the errors in the earlier 'deliverables' will be added to those in the later steps leading to an accumulating detrimental effect. In general the later in a project that an error is found the more expensive it will be to fix. In addition because the number of errors in the system are unknown, the debugging phase of a project are particularly difficult to control.

Factors Affecting the Quality

- James A. McCall, for instance, grouped software qualities into three sets:
  - Product Operation Qualities;
  - Product Revision Qualities;
  - Production Transition Qualities

Product Operation Quality Factors

- *Correctness*
- *Reliability*
- *Integrity*
- *Usability*

Product Revision Quality Factors

- *Maintainability*
- *Testability*
- *Flexibility*

Product Transition Quality Factors

- *Portability*
- *Reusability*
- *Interoperability*

SQA encompasses

- A quality management approach,
- Effective software engineering technology (methods and tools),
- Formal technical reviews that are applied throughout the software process,
- A multi-tiered testing strategy,
- Control of software documentation and the changes made to it,
- A procedure to ensure compliance with software development standards (when applicable), and
- Measurement and reporting mechanisms.

Methods of Assurance of Quality

- There are three methods of assurance of quality:
  - Verification and Validation (V&V)
  - Inspections and Reviews
  - Static Analysis

PROJECT MONITORING PLANS

Once work schedules have been published and the project is under way, attention must be focussed on ensuring progress. This requires monitoring of

- What is happening?
- Comparison of actual achievement against the schedule and,
- Where necessary revision of plans and schedules to bring the project as far as possible back on target?

We will discuss

- How information about project progress is gathered ? and
- What actions must be taken to ensure a project meets its targets?

Tools and Technique of Monitoring

- Project Steering Committee
- Collecting Information using Charts
  - Time Sheets
  - Ball Charts
  - Cost monitoring
  - Earned Value
  - Unit Development Folder

SOFTWARE DESIGN FUNDAMENTALS

- Design is the highly creative phase in the software development where the designer plans, **"how"** *a software system should be produced in order to make it functional, reliable and reasonably easy to understand, modify and maintain.*
- A software requirements specifications (SRS) document tells us, **"what"** *a system does, and becomes input to the design process, which tells us* **"how"** *a software system works.*
- Designing software systems means determining **how** requirements are realized and result is a software design document (SDD).
- Thus, the purpose of design phase is to produce a solution to a problem given in SRS document.

OBJECTIVE/GOALS OF SOFTWARE DESIGN

- Correctness
- Efficiency
- Simplicity
- Maintainability
- Cost
- the goal is *to find the best possible design within the limitations imposed by the requirements.*
- However, the two major criteria to evaluate a design are
- *Efficiency*
- *Simplicity*.

DESING PRINCIPLE

- The design should "minimize the intellectual distance" between the software and the problem as it exists in the real world.
- The design should not reinvent the wheel.
- The design process should not suffer from "tunnel vision".
- The design should be traceable to the analysis model.
- The design should exhibit uniformity and integration.
- The design should be structured to degrade gently, even when aberrant data, events, or operating conditions are encountered.

- Design is not coding, coding is not design.
- The design should be assessed for quality as it is being created, not after the fact.
- The design should be reviewed to minimize conceptual (semantic) errors.
- The design should be structured to accommodate change.

DESIGN STRATEGIES

We have four types of design strategies are :

- Bottom-Up Design
- Top-Down Design
- Abstraction
- Refinement

Thanks!

Gagan Deep
rozygag@yahoo.com
www.rozyph.com