ISO certified ⇒ Quality (9000 : 9126)

Sw Quality Models

WK # 05

1) ISO 9126 — integration with 3rd party software

Quality Characteristics	Sub-characteristics
1) Functionality	Suitability, Accuracy, Interoperability, security.
2) Reliability	Maturity, Fault-tolerance, understandibility / reuseability, reusability.
3) Usability	Understandibility, Learnability, operability, Attractiveness.
4) Efficiency	Time behaviour, resources (H/w configuration), utilization.
5) Maintainability	Analyzability, changeability, stability, testability.
6) Portability	Adaptability, installability, co-existence, Replaceability

Quality Factor Trade Off

Usability vs Functionality
Usability vs Portability
Performance vs Security

2) Mc'call's Method..

* Prod. Operation factors :-

Correctness, reliability, efficiency, integrity &

Q: Why we need SW Quality models? When we have SW dev. models.

usability.

* Prod. Revision factors:-
  Maintainability, Flexibility, Testability.

* Prod. Transmission Factors:-
  Portability, reusability, Interepesability.

Q: Map the requirements for "Superlab" SW System to Quality factor using Mc'Calls model.

1. The problem that "Superlab" SW will be found in a state of failure during 9-4pm to regd. to below 0.5%. (reliability)

2. The superlab consisting a module that prepare detailed report of Patient's Lab Test results. (Correctness)

3. The SW will enables direct transfer of lab results to "M-D files SW pkg" (interoperability)

4. The "superlab" will also be able to accept CT-Scan files in feb months. (Flexibility)

5. It will record detailed user's log. The system will report unauthorized attempts (integrity)

6. The training of lab technician reqd. no more than 3-days. (usability)

7. The "super-lab" system that deals with billing can also be used as a subsystem of "physiotherapy sw pkg" (reusability)

8. The sw should be able to run on 12 workstations & 8 automatic testing m/c with a single model server AS20 (portability) (efficiency)

9. Sw developed for Linux OS should be compatible with Windows OS (portability)

Usability vs Functionality is a Tradeoff in the sense one lacks understanding of some system provided that high functions can costly reduce the usability for the end user (vice-versa not true)

Usability vs Portability is a trade off when usage of using the software changes with respect to in the different systems.

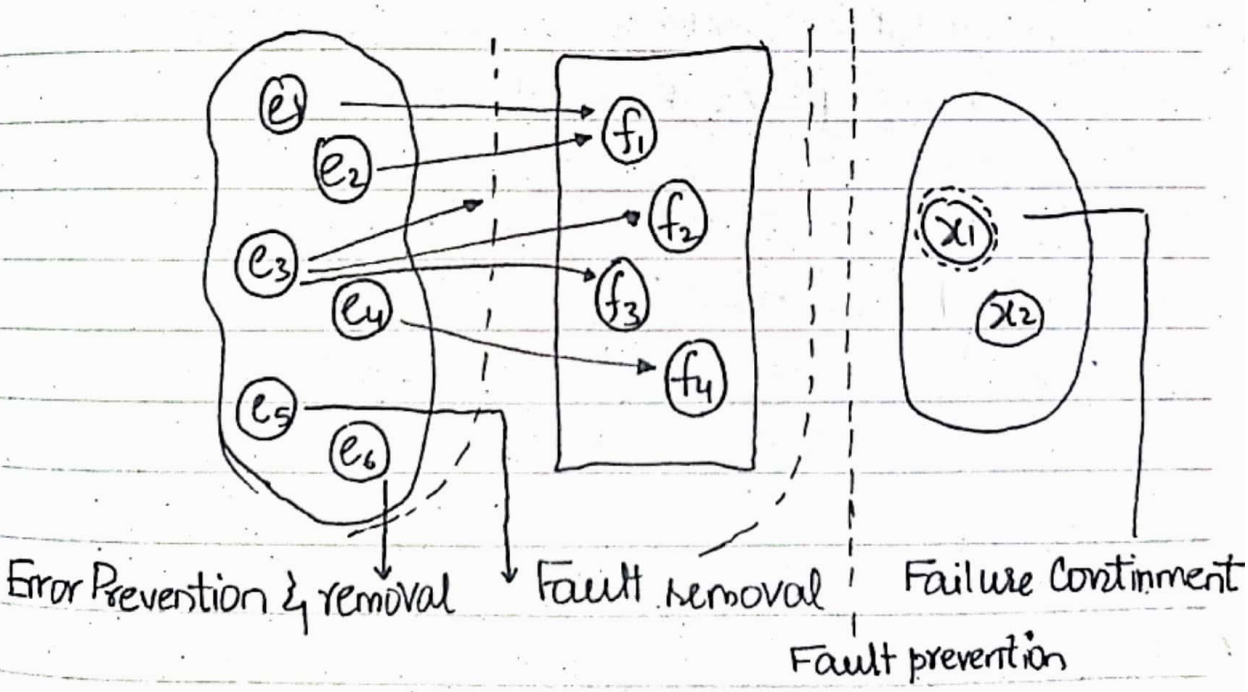
Performance vs Security:

One cannot achieve high performance & security in a single system because often high performance leads to a certain big loopholes and if one tries to create better security, performance will be compromised eventually.

(vice-versa not true)

QA As dealing with defects | QA Alternatives

WK-07



1) Defect Prevention: Through error blocking / error source removal
2) Defect Reduction: Through Fault Detection & removal
3) Defect containment: Through failure ~~to~~ prevention and containment.

1) Defect Prevention Alternatives.

Defect causes :-

→ Human misconception.
→ Imprecise design and implementation (doing by own method.)
→ Non-conformance to selected processes
→ Tools/Technologies

Alternatives:

Education & Training:
→ Prod & domain specific knowledge.
→ Slw develop. know & expertise.
→ Knowledge & Technology.
→ Develop. process knowledge

2) Defect Reduction:

Alternatives:

→ Inspections (1 to 2 weeks duration)
→ Slw Testing.

Q: At what lvl, when & defect we need to stop testing?

Ans: After we get coverage info. (like: Reliability → 99.9%) is achieved then we stop.

Various causes Of slw errors :

1. Faulty definition of requirements. (SA, C)
2. Client developer communication common failure. (P, T, SA)
3. Deliberate deviation from slw Requirm. (P, SA, P)
4. logical design error (SA, TT
5. Coding error. (P, TT

techni.

6. non-compliance with ~~education~~ documentation & coding instruction. (SA, TT, P
7. Short coming of testing process. (TT
8. Documentation error. (SA , TT)

Q: Classify the ~~source~~ causes of error(s) a/c to group responsible for this error
Client ; System Analyst ; Programmer ; Testing team.

Ans: Done on above Example.

Cost of Slw Quality :-
"Fixing bugs @ later stages are more expensive than @ early stages of development cycle." (: Agree)

Prevention Cost

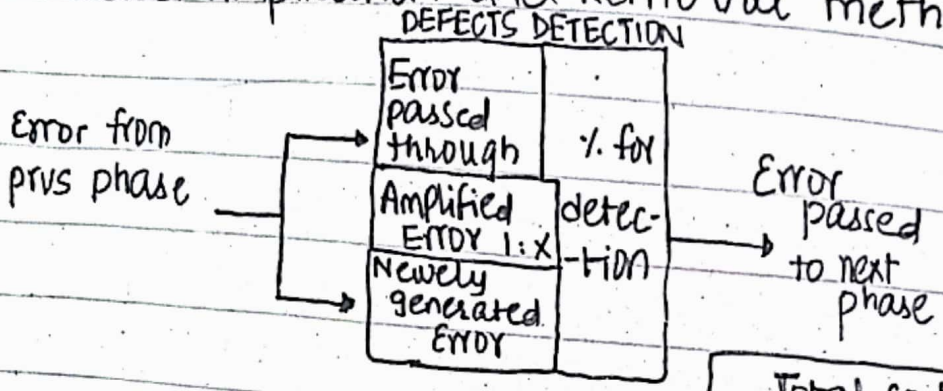
Total Cost = PC + AC + IFC + EFC of Slw Quality.

Prevention cost
Appraisal
~~Approval~~ cost
Internal Failure cost
External " " (After product deliver.)

Importance of Software Quality models
Software quality drives predictability. Do it once and do it right & there will be less re-work, less variation in productivity & better performance overall. Products get delivered on time, & they get built more productively. Poor Quality is much more difficult to manage.

latent errors = Errors with the deliverables at delivery to the client.

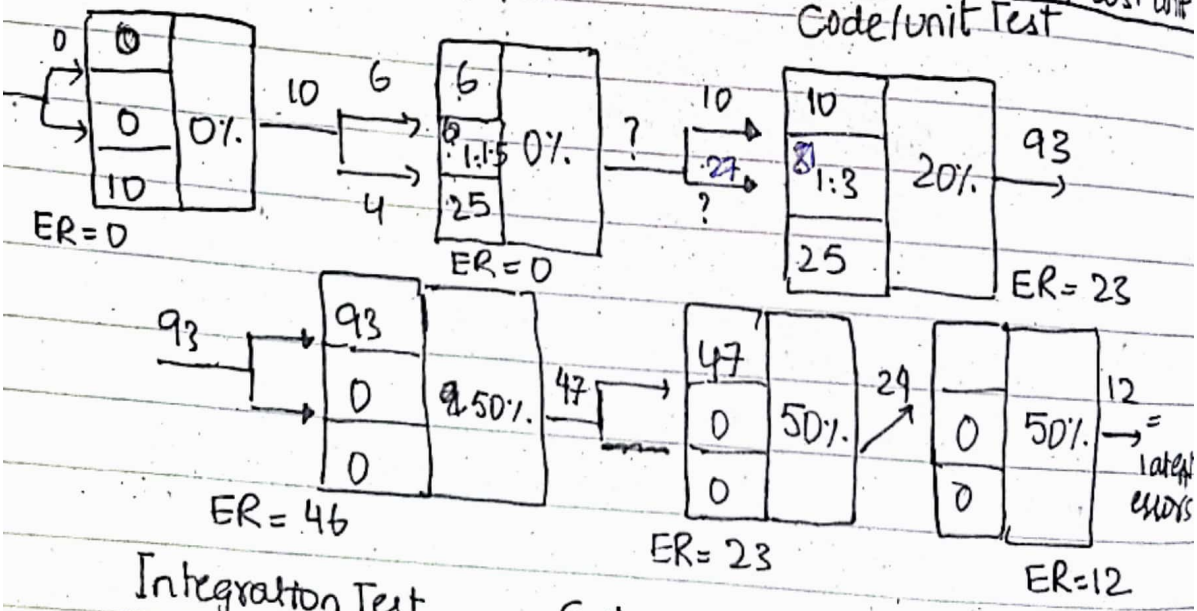
Defect Amplification and Removal method.



Eg: No reviews: (without involving static testing)

Total cost :
Before Integration Test = 6.5 co
During " - UAT = 15 cost
AIF Release = 67 cost unit

Initial phase: Design phase Code/unit Test

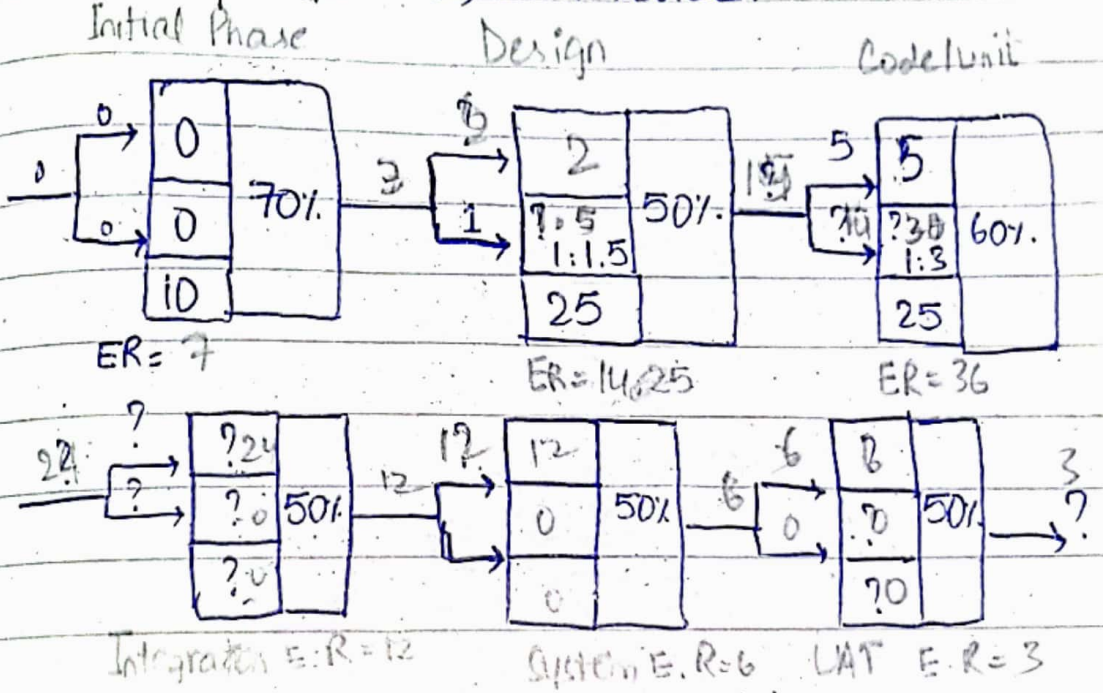


Integration Test System Testing UAT

$0 + 0 + 23 \times 6.5 + 46 \times 15 + 23 \times 15 + 12 \times 15 + 12 \times 67 =$
$= 2168.5$ cost unit

∴ ER = Error removed.

Example (part b) with reviews:



Calculate latent errors and total cost.

Data:-

Cost unit Reqd. to remove error in UnitTest = 6.5 CU

" " " " " till design = 1.5 CU

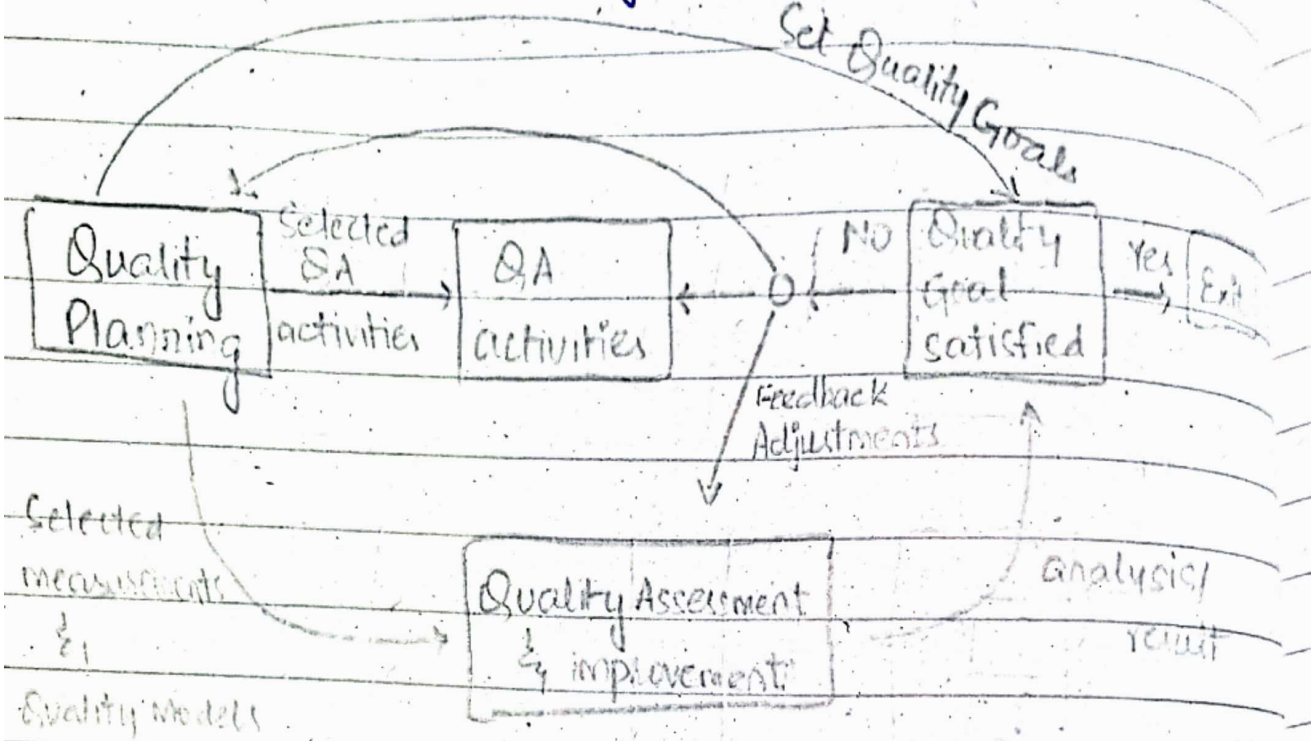
During integration to UAT = 15 CU

fix error of release = 67 C.U.

$7 \times 1.5 + 14.25 \times 1.5 + 36 \times 6.5 + 12 \times 15 + 6 \times 15 + 3 \times 15$

$+ 3 \times 67 = 781$ CU.

Software Quality Engg. Activities:-



SQE activities:

1) Pre-QA {Quality Planning}
   a) set specific Quality goals.
   b) Frame form an overall QA strategy.

2) In-QA { Execute a/c to planning}

3) Post-QA { Quality Measurement, assessment & improvement}
   (Just start after its In-QA initiates)

On the basis of 3 software activities, we achieve goals.