STQA 3rd mod

Objectives - for assessing a T. process should be well defined.

GQM framework :- (Goal Ques Metric)
1. List major goals.
2. Drives from each goal. Q & A.
3. Decides what must be measured in order to ans the ques. adequately.

Attr & com. metrics.

Test Progress - 1) Scope of testing 2) T. progress. (Track) 3) Defect Backlog 4) Staff productivity (Defect) (temp) 5) Suspension Criteria. 6) Exit criteria (Staff) (Suspension) (Exit.)

Schedule, Budget & resources.

Cost - 1) Testing Cost. 2) Duration of testing (money) 3) Resource req. 4) Cost eff. of automated tools. (Time) 5) Training needs of testing groups & tool req (resource)

Quality - 1) Eff of T. Cases. 2) Eff of smoke tests. 3) Quality of test plan. 4) Test completeness

Size - 1) Est. of T. case 2) No. of regression test 3) Test to automate.

scope - est overall work req. by doc.
Track - Decide, document attempted, doc complet

2) Models for est. T. efforts.

1) Halstead Metrics.

$$PL = \frac{1}{n_1/2} \times \frac{N_2}{n_2}$$

Effort - $e = V / PL$

% of overall testing effort to mod k can be estimated.

% of testing effort $(k) = e(k) / \sum e(i)$

eg → Small n for unique.

$n = n_1 + n_2 = 17 + 12 = 29$

$N = N_1 + N_2 = 45 + 32 = 77$ (Total)

$V = N \log_2 n = 77 \log_2 92 = 502.31$

$$P2 = \frac{1}{17/2} \times \frac{32}{12} = \frac{3}{68} = 0.044.$$

$$e = \frac{V}{PL} = \underline{11416.30}$$

Spoilage = $\frac{\text{Sum (No of Def} \times \text{Def age)}}{\text{Total Def.}}$

Architectural Design metrics

$S(m) = f^2_{out}(m)$.

↑ structural complexity → fan out of module m.

gives no. of stubs req for u.t.

Data complexity

$D(m) = v(m) / [f_{out}(m) + 1]$

$v(m)$ is no. of IP & OP.

Gives probab of error in mod m.

Compare x & y.

System Complexity

$SC(m) = S(m) + D(m)$.

Info. flow metrix used for testing

① Local Direct flow ← ① Imod invokes other mod & passes info.
② invoked mod returns result to caller.

② Local indirect flow → invoked mod returns info i.e subsequently passed to 2nd invoked mod.

③ Global flow: flow via global DS.

2 altr – (1) Fan in (2) Fan out.

Henry Kafura Design.

$IFC = len(m) \times (fan\text{-}in(m) \times fanout(m))^2$

↑IF ↑ effort in integration ↑ ∴ probab of ↑ err.

Cyclomatic complexity measures:

C. Nos measure the no. of linearly independent paths through flow graph.
it can be used as TC.
if TC < CN. find missing TC.
∴ CN provides min TC

CN should be less than 10.
if >10 : 1) Error ↑
2) Time req ↑

function pt. for testing
size of SW System

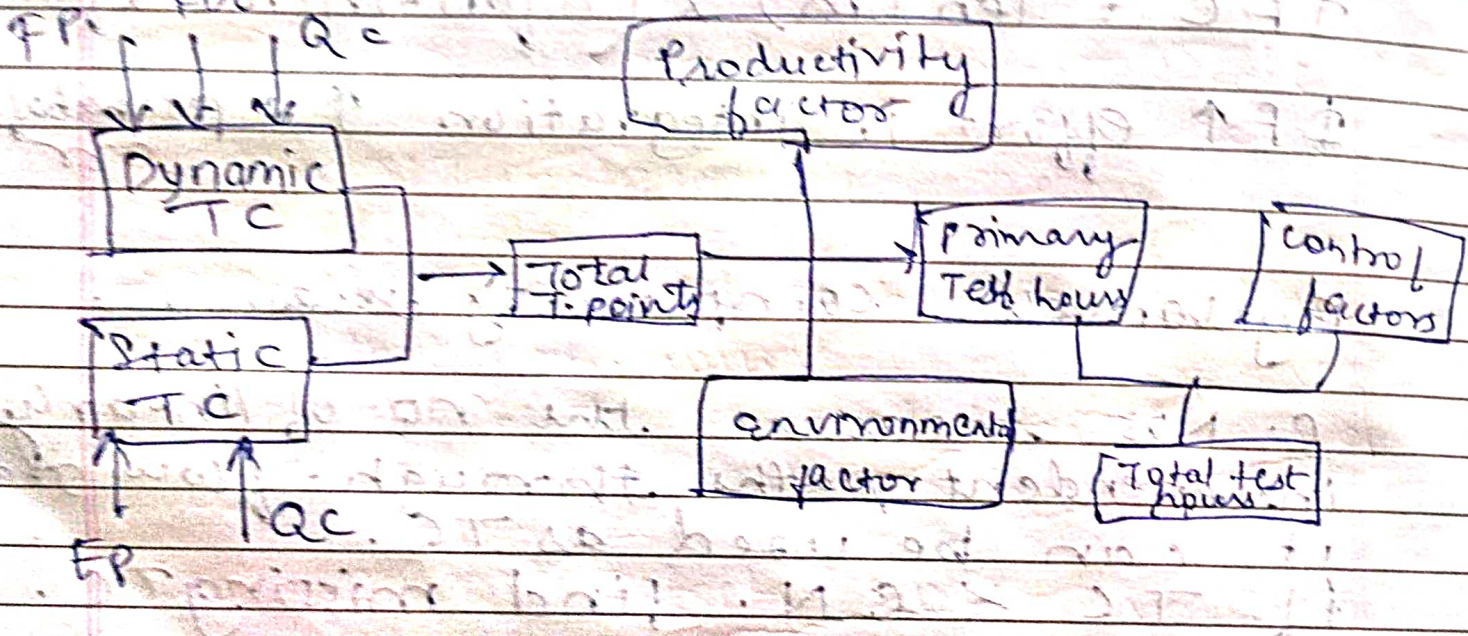
Focus measured -
1) No. of hours req. for testing per FP
2) Nos of FP's tested per person-month
3) Total cost of testing per FP.
4) Defect Density - no. of Defects / Total no. of FP

No. of TC = $(FP)^{1.2}$

(Accept) No. of TC = FP × 1.2

resume - TC coverage = $\frac{\text{No. of TC}}{\text{Total FP}}$
to forecast req.

TPA



1) Calc. dynamic T.P.

DTP = FP × FDC × QC

2) Cal Static tp

STP = FP × ΣQC / 500

Total Testpoints = DTP + STP.

3) Calc primary Test hours.

PTH = TTP x prod. x env. factor
factor

[0.7 - 2.0]

Envfactor = weight of (test tool + Dev testing + test basis + develop. environment + testing env + testware) / 21

Some testing metrix.

T.P exec Status (%) = No. of executed / Total TC.
Gives no. of % of TC remaining to be executed.

Defect aging - turnaround time for D to be corrected.
DA = Closing date of bug - Start date when opened.

D. fix time to retest

= Date of fixing & releasing - Date of retesting.

D. Trend Analysis.

⇒ No. of Defects of each type detected in unit test per hour.
⇒ No. of Defects of each type detected in integration test per hour.

Also no. of Defects over severity levels.

if trend shows err increasing -
1) Previous bugs have not closed / fixed.
2) Testing technique for + coverage is not adequate for earlier builds.
3) Inability to exec some test until

Recurrence Ratio

if bugfixing introduces new bugs in already working SW / same bug re-occurs the quality is not good.

this metrix indicates the test team the degree to which previously working functionality is being adversely by bugfixes. If ↑ dev. is informed.

No. of bugs remaining per fix

Defect Density

$$DD = \frac{\text{Total no. of D found in req}}{\text{No. of TC executed for req}}$$

if DD ↑ :

1) Complex functionality
2) Inadequate resources.
3) Prob. with Design & implementation

Preship DD / Postship DD.

Coverage measures. (349)

White box.

Actual degree of C / planned degree of C

Testing productivity measures:-

1) Time spent in T planning
2) ts in T case design
3) ts in t exexe
4) ts in T reporting
5) No. of test case developed
6) No. of tc executed.

Budget & Resource monitoring measures:

Earned value tracking:
used to monitor the use of resources in testing

for EVT, we need foll. measurement data:-
① Actual time / cost
② Est time or cost for each T. activity
③ Total est. time / cost for overall testing eff.

foll. metric is used to track cost & time

Est cost / time for testing activity / Actual cost / time of testing activity

TC effectiveness = $\frac{\text{no. of Defects found}}{\text{Total no. of defects}} \times 100$

Baseline TCE is assigned
if TCE is at / above baseline

Need for Automation

1) Reduction of testing effort.
2) Reduces the testers involvement in exec test
3) Facilities regression testing
4) Avoids human mistakes
5) Reduce overall cost of Sw
6) Simulated testing 7) Internal testing
8) Test enablers 9) TC Design.

Categorization of testing tools

Static testing tools - parse prog text, recognize the various sentences. Detect the foll:-
1) Statements are well formed
2) Inferences abt control flow for prog data
3) compute the set all possible values for prog. data.

① control flow Analysis ⑪ Interface Analysis
② Data use Analysis ⑤ Path.
③ Path Analysis

Dynamic Testing tools.

- tools are programm monitors -

functions :-

① List no. of times a component is called
LOC is executed.

② Report on whether a decision point has
branch coverage.

③ Report summary.

info is imp when test obj are related
in terms of coverage.

Testing Activity tools.

- Reviews & inspections.
- Test planning.
- Test design & development
- Test exec and evaluation.

- Tools for review & inspection
  ① Complexity Analysis tools ② Code comprehension

- Tools for T. planning.
  1. Templates for Tplan doc
  2. T. Schedule & staffing estimates
  3. Complexity Analysis.

- Tools for T. Design & Development.

- Test Data generator

- T.C generator

→ Test execution & evaluation tools :-

capture / playback tools.

Coverage analysis tools.

memory testing tools.

Test management tools.

Network testing tools.

- ① Cover the performance of server & NW
- ② overall system performance
- ③ functionality across server, client, network

Selection of Testing Tools.

① Match the tool to its appropriate use

② Select tool to its app. SDLC phase.

③ Select tool to skill of tester.

④ Select a tool which is affordable

⑤ Determine how many tools are required for testing system.

⑥ Select tool after examining the schedule of testing.

Testing of web based sys.

1) Interface testing.

2 main I - ① Web server & app server I.
② App server & DB server I.

Need for automation

① Redⁿ of T. effort

② Redⁿ in the testers involvement in executing tests

③ facilitates regression testing

④ Avoid human mistakes.

⑤ Reduce cost.

⑥ Simulate T ⑦ Internal testing

⑧ Test enablers. ⑨ TC Design.

Categorization

Static & Dynamic Test

Static T. tools - Detect poss fault & anomalies

① Control flow Analysis ② Data use A.
③ Interface A. ④ Path A.

Dynamic tools.

① Difficult to predict conditions & generate representative Tc.

T. tools preserve a ss of conditions known as program monitors.

② List no. of times component is called or

Need for automation

1. Red^n of T. effort
2. Red^n of the testers involvement in executing tests
3. facilitates regression testing
4. Avoid human mistakes.
5. Reduce cost.
6. Simulate T
7. Internal testing
8. Test enablers.
9. TC Design.

categorization

Static & Dynamic Test.

Static T. tools - Detect poss fault & anomalies

1. Control flow Analysis
2. Data use A.
3. Interface A.
4. Path A.

Dynamic tools.

1. Difficult to predict conditions & generate representative TC.

T. tools preserve a ss of conditions known as program monitors.

2. List no. of times component is called or

IDC is exec. gives info about path coverage. of their t. cases is used by testers.

③ Report on Decision pt has branched in all dir", thereby providing info about branch coverage.

④ Report summary stats providing HLD view of % of statements, paths & branches.

Info imp when T obj are created in terms of coverage.

Tools req - ① Temp. for T. plan do
② T. schedule and staffing est.
③ Complexity analyser

Activities done in these phases-
① T. Design & Develop.
a) T. data gen
b) T. C generator

② T. exec and evaluation tools.
a) Capture/playback tools
b) Coverage analysis tools
c) Memory testing tools
d) N/w testing.
e) Performance T.

Guidelines for TA.

Consider building instead buying
T. the tool or app. prototype
Not all t should be automated
Select Tool acc. to org needs
Use proven t. script dev. technique
Automate reg. test where feasible.

Software Quality

SW quality may be defined in ter
of delivered defect density or defect
rate etc.
i.e no of defects per unit size.
(LOC, FP or unit)

It is multi dimensional.
Every mem = responsible

Quality along w cost & schedule is
an imp activity in sw proj.

∴ few defects as poss. & high Quality Softw

Quality factors.

① Functionality ② Correctness. ③ Complet
④ Efficiency ⑤ Portability ⑥ Testabili
⑦ Usability ⑧ Reliability ⑨ Reusabili
⑩ Integrity ⑪ Maintainability

SW Quality Metrics.

3 categories - ① prod Q. M ② Inprocess Me
③ metrics for SW maintenanc

① Prod QM
① MTTF.
Defect Density metrics = $\frac{\text{no of defects}}{\text{size of prod}}$

Customer prob metrics
Prob per user month | PUM = $\frac{\text{Total no. of prob reported by customer}}{\text{Total no. of licensed month of SW during P}}$

PVM-

a) Improving dev. process & reducing prod defects.

b) Reducing nondefect oriented probs by improving all aspects of prod.

c) Increasing no. of installed licenses of prod

Customer satis metrics.

1. V satisfied 2. satisfied 3. Neutral
4. Dis dis 5. V. dissatisfied

② In proc. S.M.

a. Defect Density during T.

b. Defect arrival pattern during T.

Defect removal eff = $\frac{\text{Defects removed}}{\text{No. of prob arrival}} \times 100$.

Metrics for SW maintenance

Fix backlog & backlog management index.

BM index = $\frac{\text{No. of problem closed}}{\text{arrivals}} \times 100$

If BMI is larger than 100, i.e backlog is reduced.

If BMI less than 100, the backlog is increased.

Goal = BMI larger than 100.

Fix response & fix responsiveness.

% delinquent fixes = no. of fixes that exceeded the response Time criteria by severity level / No. of fixes delivery in specified time X 100

Fix Quality.

metric to measure the no. of fixes that turn out to be defective.

A fix is defective if it didn't fix the reported probs or injected a new defect as a consequence of fixing.

is measured as % defective fixes.

SQA models.

ISO9126 defines.
1. Quality req. defn
2. Evaluation prep
3. Evaluation procedure

Test Maturity model (TMM)

5 maturity level.

Assesment model containing assesment procedure, instrument, team training, selection criteria.

phases - Initial
Phase Defn.
Integration
management & measurement
optimization & defect prevention.

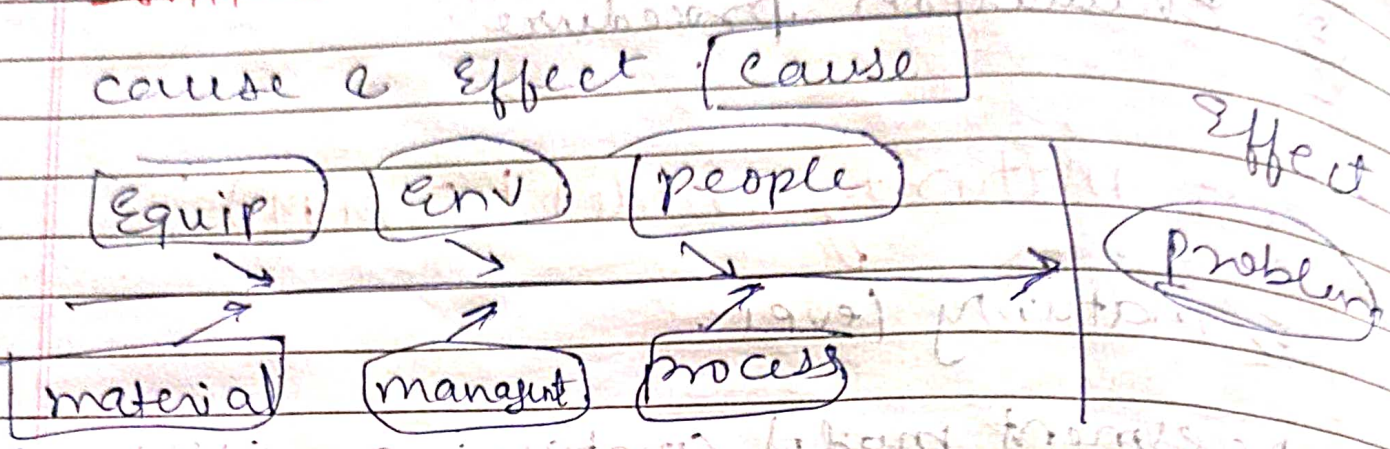
m goals
subgoals.
ATR.

Checklist - 1. Req Analysis 2. Test planning 3. Code 4. U.T 5. I.T 6. Sys T 7. perf T. 8. security T. 9. User Accep T 10. Release & Deployment. 11. Defect Tracking 12. Doc.

Pareto. 80/20
help prioritize most significant factor of problem.

Ishikawa

cause & effect

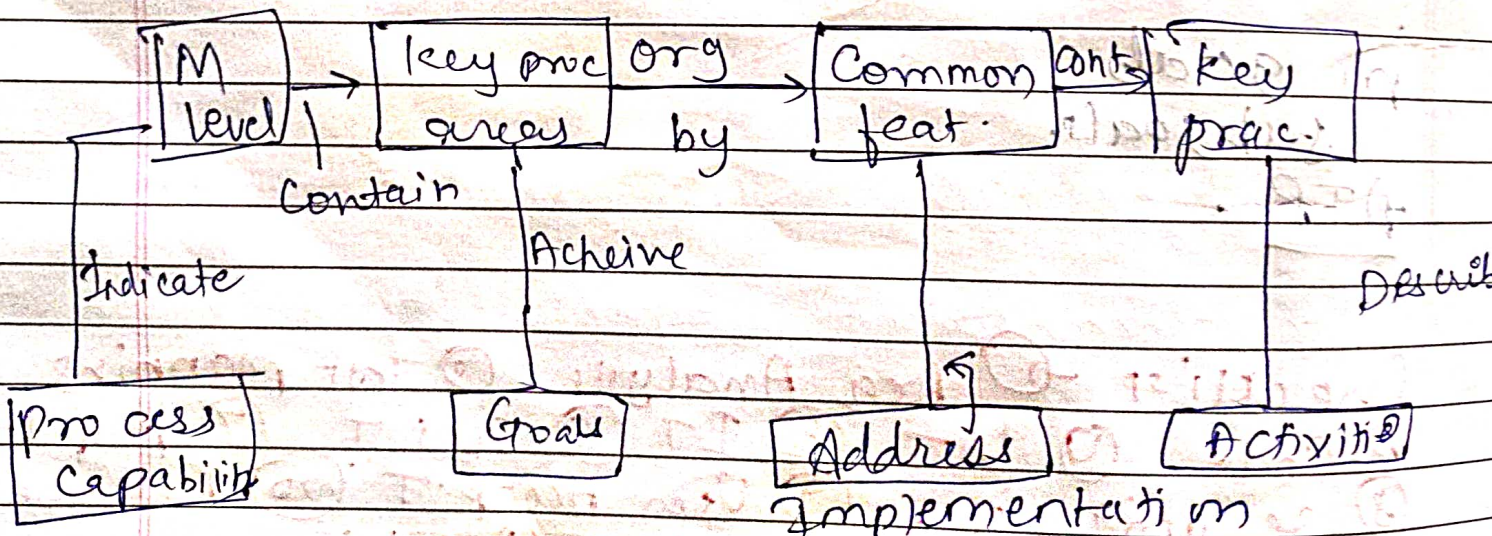


SQA models.

① ISO 9126.

FCM U.P.R.

② CMM.



Initial → Repeatable → Defined → Managed → Optimizing

Six Sigma

low cost, low defects

Define opp.
measure perf.
Analyse opp.
Improve performance
control performance

DMAIC.