Unit- V
Testing tool and measurement

Manual testing

- Process of manually testing s/w for defects
- Testers takes over the role of an end user and test the s/w to identify any unexpected behavior or bugs.
- Advantages :
- 1. it is covered in limited cost
- 2. easy to reduce and add test cases according to project movement
- 3. less time required to begin manual testing

- Limitation :
- 1. time consuming
- 2. limited support for regression
- 3. Not consistent or repeatable
- 4. error prone testing
- 5. impractical performance testing
- 6. limited scope

Automated testing

- Automating the manual testing process.
- Reduce manual testing work by using tools.
- Advantages:
- 1. speed
- 2. efficiency
- 3.accuracy and precision
- 4. resource reduction
- 5. simulation
- 6. relentlessness

Need for automated testing tools

- 1.**Effective testing:**
- --automation perform test repeatedly , so save human time.
- --Eliminate required think time.
- --Perform test at machine speed.
- **2.reducing testing costs**
- --test s/w faster with fewer errors than individual.
- --testing tools can replicate the activity of large number of users.
- --require only fraction of h/w to perform load/stress testing

- 3.**replicating testing across different platform**
- **4.greater application coverage**
- --test tool cover all modules
- **5.result reporting**
- --produce convenient report

Testing tool

- Two types:
- 1. static testing tool
- 2.dynamic testing tool
- **1. static testing tool**
- --code is not executed. Tool it self is executed and source code is input data to the tool
- --it is extension of compiler technology.
- **2. dynamic testing tool:**
- --code are in running state
- --to identify arithmetic errors

- 1. static testing tool include :
- **--flow analyzer :** ensure consistency in data flow from input to output.
- **-- path test :** find unused code and code with contradictions in the software.
- **--coverage analyzer:** ensure all logic path
- **--interface analyzer :** examine effect of passing variables between modules.
- check out the consequences of passing variables and data in the modules.

- 2.Dynamic test tool include:
- **--test driver :** input data into under tested modules.
- **--test bed :**
- it display source code along with program under execution at the same time
- --**Emulators** : emulate part of the system not yet developed.

- **--mutation analyzer :**
- They are used for testing fault tolerance of the system by knowingly providing the errors in the code of the software.
- mutate (change) certain statements in the source code and check if the **test** cases are able to find the errors.
- errors are purposely fed in order to test fault tolerance

- **Advantages of test tool**
- Reduction of repetitive work
- Consistency
- Ease of access of information about tests(chart & graph)
- **Disadvantages of test tool:**
- Unrealistic expectation from tool
- People make mistake by ignoring time , cost , effort for the initial introduction of a tool
- People doesn't maintain tests assets
- Depend on tool a lot

When to use manual testing

- **Subjective Validation**: For application functions that must be validated subjectively by humans such as usability or look-and-feel.
- **New/Changing Functionality**: For new application functions that are still being developed , changing frequently.
- **Strategic Development**: For strategic application functions that you want testers to pay specific attention . eg : hands-on manual testing may be a better alternative .
- **Complex Functionality:** functions that are extremely complex.

When to use automation testing

- **Regression Testing:** For re-testing pre existing application functions.
- **Static & Repetitive Tests:** For automating testing tasks that are repetitive and relatively unchanging from one test cycle to the next.
- **Data Driven Testing:** For testing application functions where the same functions needs to be validated with lots of different inputs & large data sets (i.e. login, search)
- **Load & Performance Testing**

Criteria for Selecting a testing tool

- **Meeting Requirements**
- -- lots of tools available in the market.
- Evaluating different tools for different requirements involve significant effort, money, and time.
- **Technology expectation:**
- --Test tools in general may not allow test developers to extends/modify the functionality. So extending the functionality requires going back to tool vendor which involves additional cost and effort.
- -- test tools are not 100% cross platform.
- **Training skills :**
- --While test tools require plenty of training, very few vendors provide the training .
- **Management aspects :**
- -- test tool require the h/w and s/w to be upgraded.
- --migrating from one test tool to another may be difficult.

S/w test measurement

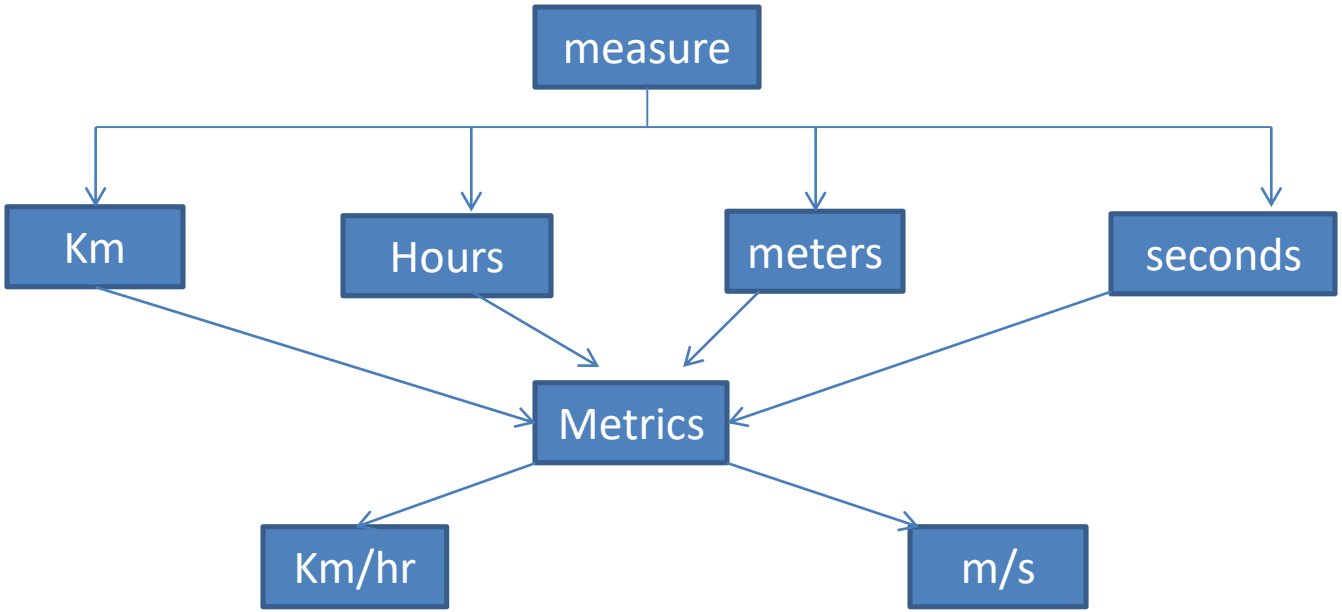
- Measurement:
- --is the quantitative indication of extent , amount , dimension , capacity of some attribute of product
- **Test measurement example:** Total number of defects.

S/w Test Metrics

- *A Metric is a quantitative measure of the degree to which a system possesses a given attribute.*
- test Metrics are used to measure the quality of the project.
- Metric is a scale for measurement.
- Eg:in software, “How many defects are found in thousand lines of code?”,
- h*ere No. of defect is one measurement & No. of lines of code is another measurement. Metric is defined from these two measurements*.

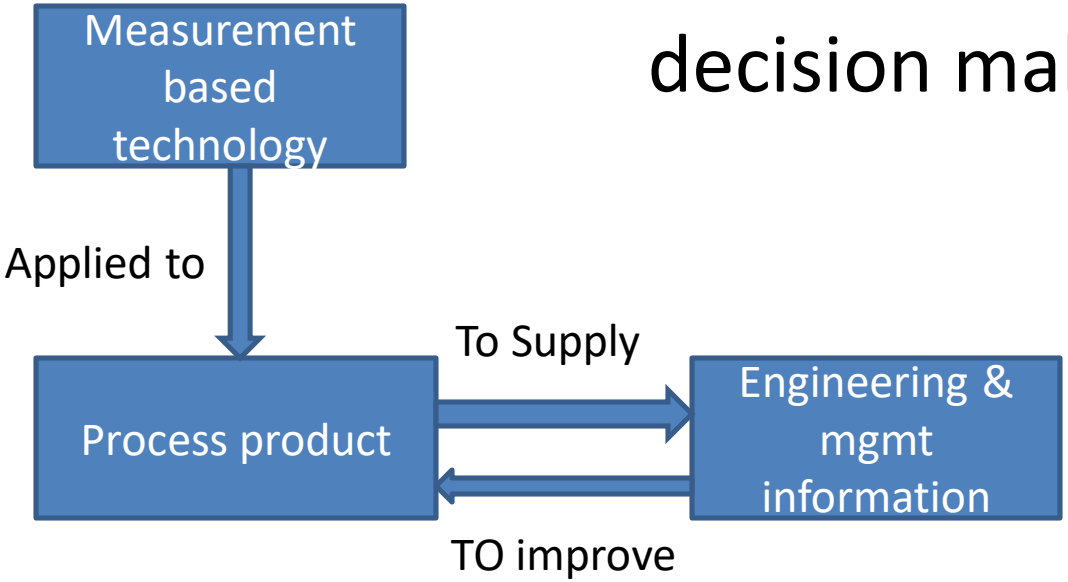
- Eg of Metrics:
- -- %ge of work completed
  --%ge of work yet to be completed
- --how many test cases are executed per person.
- --What is the test coverage %

- s/w test Measurement and Metrics

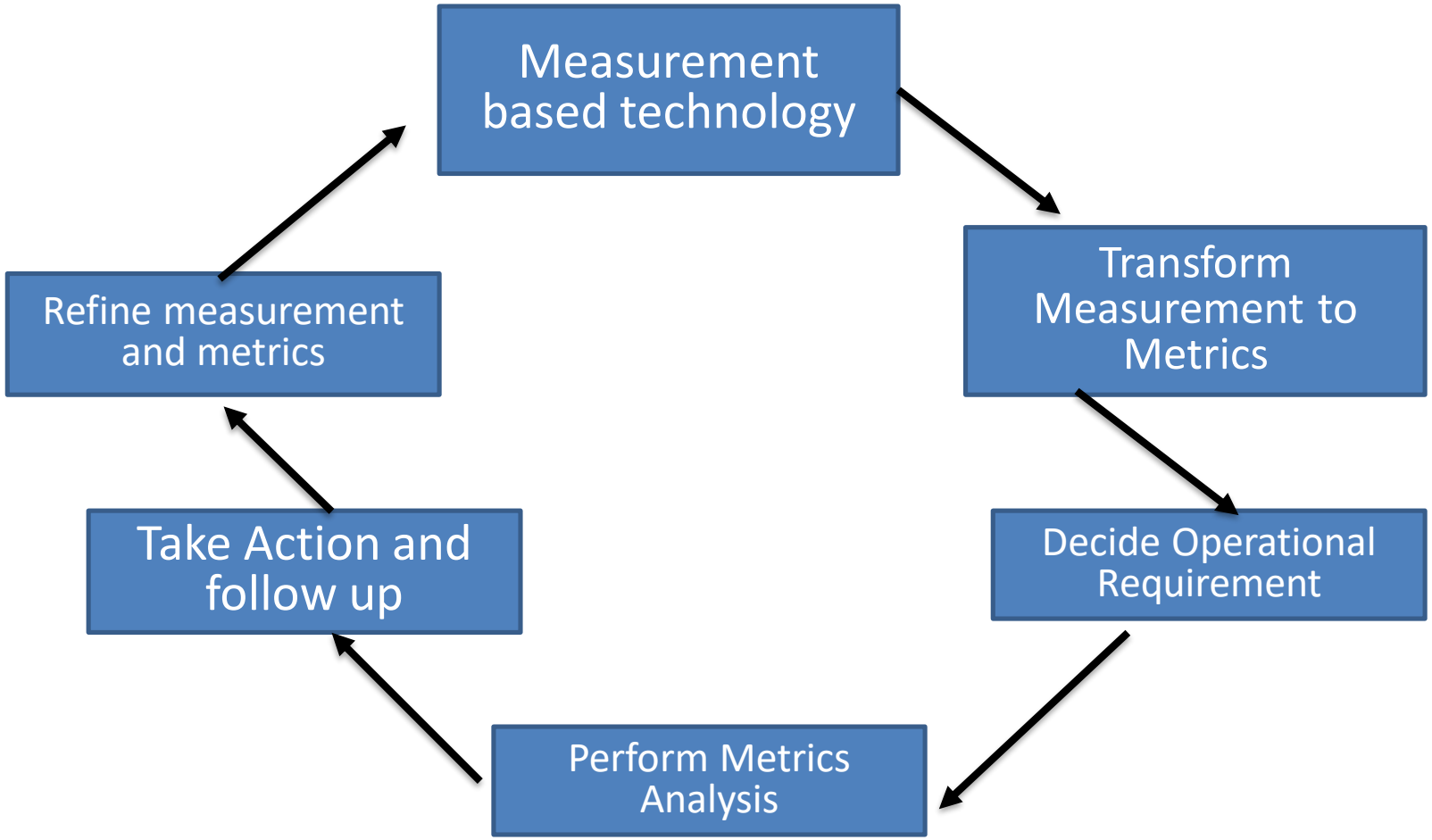


Metrics and measurement of s/w testing

- Metrics : quantifies results
- Measuring progress of testing
- Measure how much testing completed and how much more time is needed.
- Drive information from raw data to help in decision making



Steps of Measurement and Metrics



- 1. **Measurement Technology**
- 2**. Transform Measurement to Metrics**
- 3. **Operational Requirement :**assign operational responsibility for collecting , recording and reporting the measurement and metrics information.
- 4. **Perform Metrics Analysis :** analysis metrics to identify both positive areas and improvement areas on product quality.
- 5. **Take Action and follow up**
- 6. **Refine measurement and metrics**

Metrics category

- 1. **base metrics** : derived from test cases development & execution
- --keep track of : --total number of test cases developed -- executed -- pass fail

Sr.no	Testing metric	Data retrieved during test case development and execution
1	Total no.of test cases executed	100
2	No.of test cases pass	65
3	No.of test cases failed	30
4.	Total no.of test cases unexecuted	35
5.	Total no.of defects	30

- 2. **calculated metrics** :
- --derived from data gathered in base metrics
- -- track by leader / mgr for test reporting
- **Formulas for calculating metrics:**
- %age test cases executed :

(no.of test cases executed)/total test cases *100

- **%age test case not executed:**

no.of test cases unexecuted/total test cases *100

- **Project Metrics:**
- -- set of metrics indicate how the project is planned and executed
- --eg:errors found per engineer –month
- **Progress Metrics :**
- --set of metrics which track progress.
- --eg: % of test cases executed successfully
- **Productivity Metrics:**
- Set of metrics which are important with respect to product development.
- --eg : quality of the developed product

Project metrics

- Assess status of project
- Track risk
- Evaluate project team ability
- Errors uncovered during each phase are measure
- Minimize the development schedule by making adjustments.

Project metrics can be categorised :

- **1. effort variance**
- -difference between the base lined(actual person hours) and revised effort(estimated person per hours).
- - provide quantitative measure of difference between actual and revised.
- **2. schedule variance**
- - deviation of actual schedule from estimate schedule.
- -calculate “ remaining days yet to be spent” and compare it along with the “ actual schedule spent”
- **3. effort distribution across phase:**
- -Product quality can be obtained if the effort distribution across the various phases are captured and analysed.
- -distribution % across different phase can be estimated at the time of planning and compare with actual at the time of release.
- Mature organization spend

  10-15% ……..requirement

  10-15%.....design

  20-50%.....testing

Progress metrics

- How different activities are progressing
- Track planned vs actual over time
- Man hours/test case executed
- Test cases executed/planned
- Test cases executed / defect found

- **Tests defect metrics :**
- Test team analysis test activities.
- Understand How the defects that are found can be used to improve testing and product quality.
- It includes :
- 1.Defect find rate
- 2.Defect fix rate
- 3.Outstanding defect rate
- 4.Priority outstanding rate
- 5. **Defect trends :** like defect fix rate is not in line with outstanding defect rate.
- Defect fix rate is not at the same degree of defect find rate.

- 6. **Defect classification trend** : defect can be classify as major , minor.

- 7. **Weighted defect trends** : critical defect are given higher weight age than less serious defect.

- 8.**Defect cause distribution:** finding root cause of defects helps in preventing defects.
- Eg: if code phase causes the maximum defect then put more effort on white box testing and code review.

- **Development defect metrics:**
- Development team analysis development activities.
- 1. **component wise defect metrics:**
- Map defect to different components so that they can be assigned to appropriate developer.

- 2. **Defect density :**
- = no . of defect / size

- 3. **Age analysis of out standing defects:**
- identify high priority defects wating for long time to be fixed

- 4. **introduced and reponed defect :**
- Introduced defect : occurrence of defect bcoz of defect fixing
- Reopened defect : occurrence of previous defect bcoz of changes.

Productivity metrics

- Finding out how well team is progressing understanding both (+ve , -ve) variations in result.
- Estimating number of defects that can be found.
- Estimating release date and quality
- Estimating cost involved in the release.
- Example :
- 1. defect per 100 hours of testing
- =(total defect found / total hours spent) *100
- 2. test case executed per 100 hours of testing
- 3. test cases developed per 100 hours of testing.
- 4. defect per 100 test cases

Process Productivity

- s/w test metrics used in the process of test preparation and test execution phase.
- **1. test case preparation productivity :**
- calculate no.of test cases prepared and effort spent for the preparation of test cases
- =total no.of test cases / effort spent for preparation
- **2. test design coverage :**
- Measure % of test case coverage against the number of requirements.
- **3. test execution productivity:**
- **Calculate no. of test cases that can be executed per hrs.**

- **4. test execution coverage :**
- No.of test case executes against planned.
- **5. test case passed :**
- Measure % of No.of test cases passed
- **6. test cases failed :**