* Introduction

In this article I will discuss “Testing Metrics” and the “relevant testing metrics principles” that may apply in a games environment. The goal of Testing Metrics is to able measure software quality, so it can deliver the best product to the customer.

* What is testing Metrics?

Testing metrics is “Standard of measurement”

**Test Metrics are used to,**

1. Take the decision for next phase of activities such as, estimate the cost & schedule of future projects.
2. Understand the kind of improvement required to success the project
3. Take decision on process or technology to be modified etc

What is principles?

Since there are no principles in testing metrics, the following definition for principle are define by our teacher as

“"Principles" is meant to be an antonym for "ideas" or "opinions" on testing metrics that were included in the notes for those sessions. The main thrust is to determine to what extent testing metrics assist in making assessments as to whether progress, productivity and quality goals are being met.”

* What is the benefits of applying testing metrics to games environment?
* Easily to lead to successfully
* Decrease cost
* Decrease time
* Successfully
* Able to tell whether the process implemented is improved a not
* Improvement of required
* Helps taking decisions on process
* Set goals and predict future trends.
* Identification of testing strengths and weaknesses.
* Providing insights into the current state of the testing process.
* Evaluating testing risks.
* Benchmarking.
* Improving planning.
* Improving testing effectiveness.
* Evaluating and improving product quality.
* Measuring productivity.
* Determining level of customer involvement and satisfaction.
* Supporting controlling and monitoring of the testing process.
* Comparing processes and products with those both inside and outside the organization.
* Able to use a Equations, tables, Pie charts, graphs.
* Form a basis for decision making.
* Set realistic expectations.
* Focus people’s activities.
* Supports process improvement.
* Provide status visibility.
* Improve morale.
* *Makes thing control things by putting in measurable data*

What is drawbacks without testing metrics in game environment?

* + User view

Without testing metrics one of the worst case it can be will be like simcity5 or CityXXL.

* + Company view

Without apply testing metrics in a game there will be a possiblitlbty to lead to waste cost and increasing the development specially large games (eg: GTA , )

* + Impossible to tell whether the process implemented is improved a not
  + For a small companies it will reduce the cost

APPLYING TESTING METRICS IS DEPENDINGS ON THE PROJECT SIZE, PROJECT GOAL,…..etc

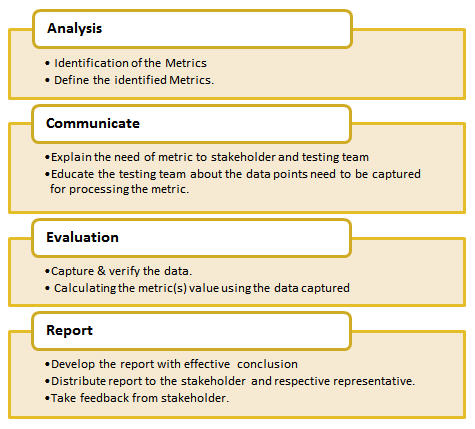
SHOULD HAVE A RANGE FROM SMALL TO LARGE

An example for a small game called SGame, and a sample for a large game called LGame.

* In conclude the following list show that what type of problem that maybe developer will run into without applying testing metrics.
  + Element 1 – Metrics Life Cycle
    - **What does it effect? Why does it effect?**

**The Life Cycle solve the point 1,2,3 problem by giving a ??? ,** The Life Cycle is similar like an instruction simple guide that teach you how you can build a computer.

**Image can be respsent below**



* + - **How does it effect the company or players?**

For a company which are making large game this can possible decrease the development time and cost, which to player it means it can play the game early.

* + - **How can we apply?**

**To be able to solve the 1 problem** .The Life Cycle 1 step is to find out what things would need to be test and apply names for the metrics. For example if we are making GTA5 there are a lot things we need to be tested, but since there are too much things to test we can’t or don’t have that much to test all of them because that will increase the cost of development, so we choice the most important ones that must be tested and write it down.

**To be able to solve the 2 problem**. The Life Cycle 2 step is to explain to other people why does it need to test or how can it increase the sale. And check is there any miss out points that was not included. After that has been conformed we also need to tell them how are we going to use the metrics and when does it start, how much time do we have and such.

**To able to solve the 3 problem**. The Life Cycle 3 step is to calculate all the formulas from the result that have been giving, for example, how much bugs have been fix? How much bugs do we still need to fix? How much bugs will there be in the game?

**The able to solve the 4 problem.** The Last step is to write report to stakeholder and take feedback from the stakeholders. This step is not recommend for small companies because this will be using too much time which also means this will increase the cost. And other reason is not needed is because in small companies the stakeholder usually we be attended in every meeting. But for large companies since the stakeholders mostly have something to do. So writing report to them is necessary.

**A Good Life Cycles:**

**A Bad Life Cycles:**

**Example 1:**

**Example 2:**

* + Element 2 – Measures
    - **What does it effect? Why does it effect?**

The Measures effects the point of 4, 5, and 6. By giving providing guide or rules that what are the things should we measure?

* + - **How does it effect the company or players**

Almost same as above, the difference between Measures and Life Cycles is a Measures need number and a more detail guide line that what you should be doing.

* + - **How can we apply?**

To be able to solve the 4, 5, 6 that maybe appear in game environment. I will recommend to apply the following testes that comes from.

1. Customer satisfaction index

2. Delivered defect quantities

3. Responsiveness (turnaround time) to users

4. Product volatility

5. Defect ratios

6. Defect removal

7. Complexity of delivered product

8. Test coverage Breadth of functional coverage

9. Cost of defects

10. Costs of quality activities

11. Re-work

12. Reliability

Notice when using the measurement the following points need to be consider

* Not all the components needed to be tested
* The questions that you ask or tested will be based on what your goal is.

Since not all the components needed to testing and also we are allow the setup the question we need, this leads us to solve the 1, 2.

The following list will provide more details example for the components above.

**A Good Measure Metrics:**

**A Bad Measure Metrics:**

**Example 1:**

**Example 2:**

* + Element 3 – Formulas
    - **What does it effect? Why does it effect?**

By applying the formulas is can help us to solve 1,2,3 point by applying equation to

* + - **How does it effect the company or players**
    - **How can we apply?**

The formula 1 help us to solve the 1 problem by applying this equation

**Test Coverage** = Number of units (KLOC/FP) tested / totalsize of the system. (LOC represents Lines of Code)

**Number of tests per unit size** = Number of test cases per KLOC/FP (LOC represents Lines of Code).

**Acceptance criteria tested** = Acceptance criteria tested / total acceptance criteria

**Defects per size** = Defects detected / system size

**Test cost (in %)** = Cost of testing / total cost \*100

**Cost to locate defect** = Cost of testing / the number of defects located **Achieving Budget** = Actual cost of testing / Budgeted cost of testing Defects **detected in testing** = Defects detected in testing / total system defects **Defects detected in production** = Defects detected in production/system size **Quality of Testing** = No of defects found during Testing/(No of defects found during testing + No of acceptance defects found after delivery) \*100 **Effectiveness of testing to business** = Loss due to problems / total resources processed by the system.

**Not all formulas needed to**

**System complaints** = Number of third party complaints / number of transactions processed

**Scale of Ten** = Assessment of testing by giving rating in scale of 1 to 10 **Source Code Analysis** = Number of source code statements changed / total number of tests.

**Effort Productivity** = Test Planning Productivity = No of Test cases designed / Actual Effort for Design and Documentation

**Test Execution Productivity** = No of Test cycles executed / Actual Effort for testing

**For small companies seems people as soon as they don’t get any lost it’s fine.** Because instead using time to test it’s better to just make a game

, but this will change if a game need to release more than 8 updates for a small game , they won’t need to know how much they spend that they should regozie they should do something to about it. And get a rougze enimate of how much it takes

**For Big companies,** its higher recommend because by knowing where the money is going you can control the money and also control the time.

* + Element 4 - Cost
    - **What does it effect? Why does it effect?**
    - **How does it effect the company or players**
    - **How to fix it?**
  + Element 5 - Time
    - **What does it effect? Why does it effect?**
    - **How does it effect the company or players**
    - **How to fix it?**

Two type of metrics

Base metrics (Direct Measure)

* Constitute the raw data gathered by the test engineers throughout the testing effort
* Provide status reports to the Test lead and to the project manager
* Provide the input data to feed into the formulas used to derive Calculated metrics

Calculated Mertrics()

Covert the Base metrics data into more useful information

The calculated metrics are generally prepared by the test lead and is used to track the progress of the project at different levels like at module level, at tester level and for the project as a whole.

The calculated metrics provide valuable information that when used and implemented often timer leads to significant improvements in Overall SDLC

#### Benefits of Metrics



####  Using Metrics

 Display in form of

 Develop or refine development processes.

43 Use for Pareto analysis (A technique used for decision making) - Determine where most of the project’s time should be spent.

 Discussing metrics with staff can be more valuable than the metrics themselves

####  Risks of Metrics

 Misuse for employee evaluations (your productivity was low - you’re fired!)

 Misuse for employee incentives (I’ll pay $10 for each bug found)

 Dependence on metrics outside of   
their scope

 Use of unreliable data

####  Limitations of Metrics

 Over-reliance on statistics

 Data inaccuracy

If ???% bugs where fix people can able to appect the software