DONALD COUTTS

CIT185753 Kitman Yiu

Testing Metrics

Application Testing & Metrics

目錄

[Metrics Life Cycle 3](#_Toc419144578)

[SOFTWARE METRICS 3](#_Toc419144579)

[SUBSET 4](#_Toc419144580)

[Test PROCESS Metrics 4](#_Toc419144581)

[Test Product Metrics 5](#_Toc419144582)

[Product Quality Measures 5](#_Toc419144583)

[TYPE OF METRICS 7](#_Toc419144584)

[Recommendations 8](#_Toc419144585)

[Importance of Metrics 8](#_Toc419144586)

[Examples 9](#_Toc419144587)

[angry bird 10](#_Toc419144588)

[Description 10](#_Toc419144589)

[RATM 11](#_Toc419144590)

[GTA 5 11](#_Toc419144591)

[Introduction 12](#_Toc419144592)

[REFERENCE 13](#_Toc419144593)

[Manual Testing Metrics 14](#_Toc419144594)

[Test Case Productivity (TCP) 14](#_Toc419144595)

[Test Execution Summary 14](#_Toc419144596)

[Defect Acceptance (DA) 14](#_Toc419144597)

[Defect Rejection (DR) 14](#_Toc419144598)

[Bad Fix Defect (B) 14](#_Toc419144599)

[Test Execution Productivity (TEP) 14](#_Toc419144600)

[Test Efficiency (TE) 14](#_Toc419144601)

[Defect Severity Index (DSI) 14](#_Toc419144602)

[Performance Testing Metrics 14](#_Toc419144603)

[Performance Scripting Productivity (PSP) 14](#_Toc419144604)

[Performance Execution Summary 14](#_Toc419144605)

[Performance Execution Data - Client Side 14](#_Toc419144606)

[Performance Execution Data - Server Side 14](#_Toc419144607)

[Performance Test Efficiency (PTE) 14](#_Toc419144608)

[Performance Severity Index (PSI) 14](#_Toc419144609)

[Common Metrics for all types of testing 14](#_Toc419144610)

[Effort Variance (EV) 14](#_Toc419144611)

[Schedule Variance (SV) 14](#_Toc419144612)

[Scope Change (SC) 14](#_Toc419144613)

# Introduction

There is variety of literature pertaining to programming and gaming. There is also a lot of information about software testing metrics principles, however this information has never been applied in game environments. This essay will examine “testing metrics principles” as they are applied in a games environment. Firstly, the benefits of applying testing metrics will be considered. Next, the elements in testing metrics we be analysed to determine the benefits of each of these. Lastly, examples of both good and bad metrics we be presented a discussion of how these can applied to small and large game companies will follow.

# Background

In the Programming discipline there are well established testing metrics: software testing metrics are the “Standards of measurement”.(ref). The goal of Testing Metrics is too enable Test Leaders to measure software quality, so it can deliver the best product to the customer. Within Software testing metrics there are three categories to be investigated: Test Process Metrics, Test Product Metrics and Test Project Metrics (ref). Test Process Metrics include the process of …. Test Product Metrics include the process of ….. and Test Project Metrics include the process of …..(ref).

Software metrics and Software testing metrics are different

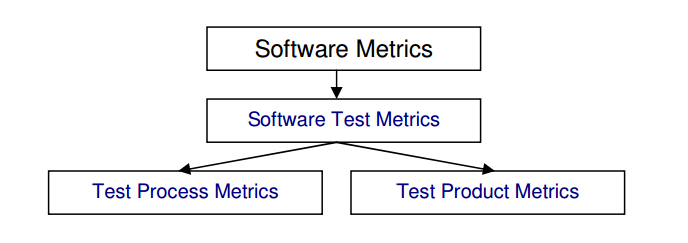
# Metrics Life Cycle

The main goal the testing Life Cycle, we able to gives us the following ideas

* Who will collect metrics data?
* When will the metrics data be collect?
* How to collect the metrics data.

Notice that: All these we be written in a test plan [documentation](http://en.wikipedia.org/wiki/Documentation)

# SOFTWARE METRICS



A metric is a quantifiable measurement of software product, process, or project that is directly observed, calculated, or predicted. Metrics (or indicators) are computed from measures. Metrics facilitate the quantification of some particular characteristic. Metrics are measurements of different aspects of an endeavor that help us determine whether or not we are progressing toward the goal of that endeavor. Metrics are usually specialized by the subject area, in which case they are valid only within a certain domain and cannot be directly benchmarked or interpreted outside it. "Essentially, software metrics deals with the measurement of the software product and the process by which it is developed. They are quantifiable indices used to compare software products, processes, or projects or to predict their outcomes. Software Metrics should be well defined before they are utilized; Table 1 defines the elements that must be defined properly. With Software testing metrics, we can:

* Monitor requirements
* Predict development resources
* Track development progress
* Understand maintenance costs

# SUBSET

## Test PROCESS Metrics

The goal of Process metrics is to find out how much time and money did it take during developing the game. To following examples are somewhat we can test

* Test Coverage = Number of units (KLOC/FP) tested / total size 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.
* 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

## Test Product Metrics

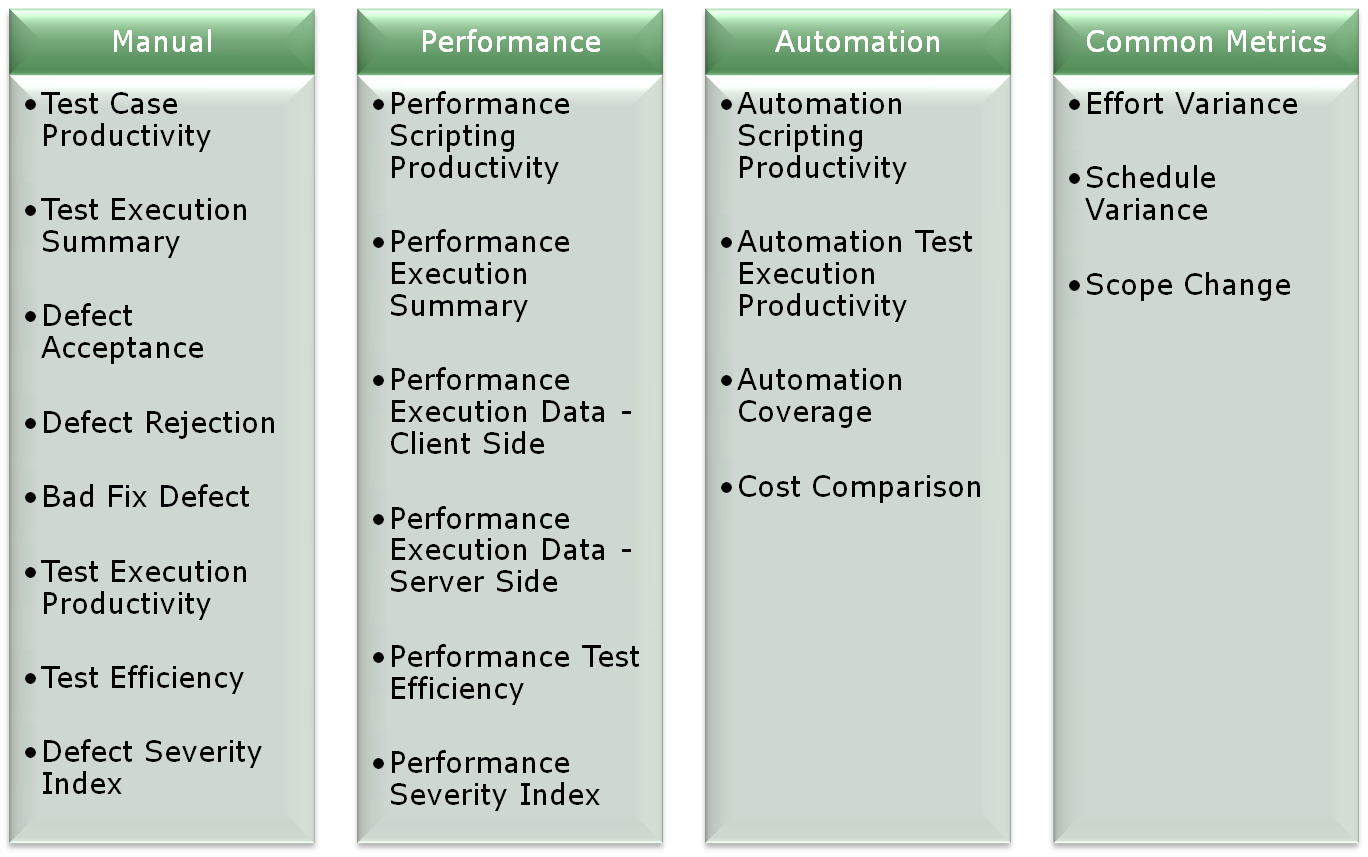
The goal of Process metrics is to find out how much time and money did it take during developing the game. To following examples are somewhat we can test

### Product Quality Measures

|  |  |
| --- | --- |
| **Test Metric** | **Description** |
|  | |
| **Customer satisfaction index**  This index is surveyed before product delivery and after product delivery  (and on-going on a periodic basis, using standard questionnaires).The following are analyzed: | |
| Number of system enhancement requests per year | For a game this can be just one year |
| Number of maintenance fix requests per year | For a game this can be just one year |
| User friendliness: call volume to customer service hotline , forums , online chat | In game environments this not only include hotlines , it can also include forums or online chat |
| User friendliness: training time per new user | This can change to how much player can be able to play the game without asking help |
| Number of product recalls or fix releases (software vendors) |  |
| **Delivered defect quantities** | |
| They are normalized per function point (or per LOC) at product delivery (first 3 months or first year of  operation) or Ongoing (per year of operation) by level of severity, by category or cause, e.g.:  requirements defect, design defect, code defect, documentation/on-line help defect, defect introduced by  fixes, etc. | |
| **Responsiveness (turnaround time) to users** | |
| Turnaround time for defect fixes, by level of severity | The level can of severity can be define by the company |
| Time for minor vs. major enhancements; actual vs. planned elapsed time | This metrics will be used when trying to improve the planning process of the game |
| **Product volatility** | |
| Ratio of maintenance fixes (to repair the system & bring it into compliance with specifications), vs. enhancement requests (requests by users to enhance or change functionality) | This maintenance rate should be higher than enhancement requests. |
| **Defect ratios** | |
| Defects found after product delivery per function point. | How many bugs have been found in game after released |
| Defects found after product delivery per LOC |  |
| **Defect removal efficiency** | |
| Number of post-release defects categorized by level of severity | How horrible is the bug , categorized by level of severity, this can found by clients in field operation |
| Ratio of defects found internally prior to release (via inspections and testing), as a percentage of all defects | How much bugs have been found before release. |
| All defects include defects found internally plus externally (by customers) in the first year after product delivery | Usually a big game will stop releasing update after one year |
| **Test coverage** | |
| The ratio of the number of detected faults to the number of predicted faults. |  |
| **Cost of defects** | |
| Business losses per defect that occurs during operation | This is related to business goal |
| Business losses per defect that occurs during operation | This is related to business goal |
| Business interruption costs; costs of work-arounds | This is related to business goal |
| Lost sales and lost goodwill | This is related to business goal |
| Litigation costs resulting from defects | This is related to business goal |
| Annual maintenance cost (per function point) | This include a wide range of cost: For example computer in the company that need to repair |
| Annual operating cost (per function point) | This can include bills |
| Measurable damage to your boss's career | This is related to business goal |
| **Costs of quality activities** | |
| Costs of reviews, inspections and preventive measures | This is related to business goal |
| Costs of test planning and preparation | This is related to business goal |
| Costs of test execution, defect tracking, version and change control |  |
| Costs of diagnostics, debugging and fixing |  |
| Costs of tools and tool support | The tools that you make or the game engine that use to developing the game such as Maya , Visual Studio 2012 |
| Costs of testing & QA education associated with the product | Allow the QA people get updated |
| Costs of monitoring and oversight by the QA organization | This need to be include if separate from the development and test organizations |
| **Re-work** | |
| Re-work effort | hours, as a percentage of the original coding hours |
| Re-work effort | hours, as a percentage of the original coding hours |
| Re-worked LOC | source lines of code, as a percentage of the total delivered LOC |
| Re-worked software components | this is a percentage of the total delivered components |
| **Reliability** | |
| Availability | percentage of time a system is available, versus the time the system is needed to be available |
| · Mean time between failure (MTBF). | How often does it take the game to failure , is will display as a percentage |
| Man time to repair (MTTR) | How often does it take the developer to repair the game bug , is will display as a percentage |
| Reliability ratio (MTBF / MTTR) | The ratio of MTBF and MTTR. The MTBF should be higher |
| Number of production re-runs as a ratio of production runs |  |

# TYPE OF METRICS

Following figure shows different software testing metrics.



# Recommendations

## Importance of Metrics

Metrics is used to improve the quality and productivity of products and services thus achieving Customer Satisfaction. Thus, Metrics help organization to obtain the information it needs to continue to improve its processes, products and services and achieve the desired Goal as:

* "You cannot control what you cannot measure" (Tom DeMarco)
* Easy for management to digest one number and drill down, if required.
* Different Metric(s) trend act as monitor when the process is going out-of-control.
* Metrics provides improvement for current process.
* Metrics for which one can collect accurate and complete data must be used.
* Metrics must be easy to explain and evaluate.

|  |  |
| --- | --- |
| **Test Metric** | **Description** |
|  | |
| **Manual Testing Metrics** | |
| Test Case Productivity | Provides the information for the number of step(s) written per hour. |
| Test Execution Summary | Provides statical view of execution for the release along with status and reason. |
| Defect Acceptance | Indicates the stability and reliability of the application. |
| Defect Rejection | Provides the percentage of invalid defects. |
| Bad Fix Defect | Indicates the effectiveness of the defect-resolution process |
| Test Execution Productivity | Provides detail of the test case executed per day. |
| Test Efficiency | Indicates the testing capability of the tester in identifying the defect. |
| Defect Severity Index | Provides indications about the quality of the product under test and at the time of release. |
| **Performance Testing Metrics** | |
| Performance Scripting Productivity | Provides scripting productivity for performance test flow. |
| Performance Execution Summary | Provides classification with respect to number of test conducted along with status (Pass/Fail), for various types of performance testing. |
| Performance Execution Data - Client Side | Gives the detail information of Client side data for execution |
| Performance Execution Data - Server Side | Gives the detail information of Server side data for execution |
| Performance Test Efficiency | Indicates the quality of the Performance team in meeting the performance requirement(s). |
| Performance Severity Index | Indicates quality of product under test with respect to performance criteria. |
| **Common metrics for all types of testing** | |
| Effort Variance | Indicates effort stability |
| Schedule Variance | Indicates schedule stability |
| Scope Change | Indicates requirement stability |

# Examples

As we mention above to able to identity what metrics we first need the following information

* Team size
* What are you trying?
* Time
* Money

## angry bird

Apply to small games, example angry bird.



### Description

Angry Birds is a [puzzle](http://en.wikipedia.org/wiki/Puzzle_video_game) [video game](http://en.wikipedia.org/wiki/Video_game) developed by [Finnish](http://en.wikipedia.org/wiki/Finns) [computer game developer](http://en.wikipedia.org/wiki/Computer_game_developer) [Rovio Entertainment](http://en.wikipedia.org/wiki/Rovio_Entertainment" \o "Rovio Entertainment) that started the [Angry Birds](http://en.wikipedia.org/wiki/Angry_Birds) franchise. Inspired primarily by a sketch of stylized wingless birds, the game was first released for [iOS](http://en.wikipedia.org/wiki/IOS" \o "IOS) in December 2009. Since that time, over 12 million copies of the game have been purchased from the iOS [App Store](http://en.wikipedia.org/wiki/App_Store_(iOS)),  which has prompted the company to design versions for other touchscreen-based [smartphones](http://en.wikipedia.org/wiki/Smartphone), most notably those using the [Android](http://en.wikipedia.org/wiki/Android_(operating_system)), [Symbian](http://en.wikipedia.org/wiki/Symbian), [Windows Phone](http://en.wikipedia.org/wiki/Windows_Phone) and [BlackBerry 10](http://en.wikipedia.org/wiki/BlackBerry_10) operating systems. The series has since expanded to include titles for dedicated [video game consoles](http://en.wikipedia.org/wiki/Video_game_console) and for [PCs](http://en.wikipedia.org/wiki/Personal_computer).

In the game, players use a [slingshot](http://en.wikipedia.org/wiki/Slingshot) to launch birds at pigs stationed in or around various structures with the goal of destroying all the pigs on the playing field. As players advance through the game new types of birds become available, some with special abilities that can be activated by the player. Rovio Mobile has supported Angry Birds with numerous free updates that add additional game content, and the company has also released stand-alone holiday and promotional versions of the game.

### Applying Testing Metrics

So assume to the following data

* Team size: 1
* Goal: Find out how much fun do they have during playing the game
* Time : No Limited
* Budget :No Limited

The following metircs can be applied

## RATM

Apply to medium game, RATM.

### Applying Testing Metrics

So assume to the following data

* Team size: 4
* Goal: Find out how much how effeicat are the programmer and artist are our testing was bad
* Time : No Limited
* Budget :No Limited

First of all we should use the life cycles to

* Who will collect?
* When will collect?
* How to collect?

Write down the list of defend

If next time the defend show that to our tester so they know you don’t need to test it.

The following metrics can be applied

## GTA 5

Apply to large game, GTA



### Introduction

Grand Theft Auto V is an [open world](http://en.wikipedia.org/wiki/Open_world), [action-adventure](http://en.wikipedia.org/wiki/Action-adventure_game) video game developed by [Rockstar North](http://en.wikipedia.org/wiki/Rockstar_North" \o "Rockstar North) and published by [Rockstar Games](http://en.wikipedia.org/wiki/Rockstar_Games" \o "Rockstar Games). It was released on 17 September 2013 for the [PlayStation 3](http://en.wikipedia.org/wiki/PlayStation_3) and [Xbox 360](http://en.wikipedia.org/wiki/Xbox_360). An [enhanced version of the game](http://en.wikipedia.org/wiki/Grand_Theft_Auto_V_(re-release)) was released on 18 November 2014 for the [PlayStation 4](http://en.wikipedia.org/wiki/PlayStation_4) and [Xbox One](http://en.wikipedia.org/wiki/Xbox_One), and 14 April 2015 for [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows). The game is the first main entry in the [Grand Theft Auto series](http://en.wikipedia.org/wiki/Grand_Theft_Auto_(series)) since 2008's [Grand Theft Auto IV](http://en.wikipedia.org/wiki/Grand_Theft_Auto_IV). Set within the fictional state of San Andreas (based on [Southern California](http://en.wikipedia.org/wiki/Southern_California)), the [single-player](http://en.wikipedia.org/wiki/Single-player_video_game) story follows three criminals and their efforts to commit heists while under pressure from a government agency. The open world design lets players freely roam San Andreas, which includes open countryside and the fictional city of Los Santos (based on [Los Angeles](http://en.wikipedia.org/wiki/Los_Angeles)).

The game is played from either a [third-person](http://en.wikipedia.org/wiki/Third-person_view) or [first-person view](http://en.wikipedia.org/wiki/First-person_view)[[c]](http://en.wikipedia.org/wiki/Grand_Theft_Auto_V#cite_note-first-person-11) and its world is navigated on foot or by vehicle. Players control the three lead protagonists throughout the single-player mode, switching between them both during and outside of missions. The story is centered on the heist sequences, and many of the missions involve shooting and driving gameplay. Players who commit crimes may incite a response from law enforcement agencies, measured by a "wanted" system that governs the aggression of their response. [Grand Theft Auto Online](http://en.wikipedia.org/wiki/Grand_Theft_Auto_Online), the [online multiplayer](http://en.wikipedia.org/wiki/Online_multiplayer) mode, lets up to 30 players explore the open world and engage in [cooperative](http://en.wikipedia.org/wiki/Cooperative_gameplay) or competitive game matches.

### Applying Testing Metrics

So assume to following data

* Team size: 40
* Goal: publish a success business goal (this include finish the game in time and also earn a lot money) and do revision of the game
* Time : No Limited
* Budget :No Limited

First of all we should life cycles to

* Who will collect?
* When will collect?
* How to collect?

Depends on company???? how often to do these test we can apply process metrics the test people , and by applying different types of testing we can see is the product on track

And after the product has been released we can apply the following metrics

To be able to find out when we should testing, we can use the following metrics:

# Conclusion

Apply testing metrics is possible, and before we applying the following four points need to be calculated first

* Team size
* What are you trying?
* Time
* Money

Testing metrics is applies is actually apply game enviroments but not in formal way, because even software testing

Lack of reference

Terms are confusing

More research have been done

# REFERENCE

Amazon.com: Testing Metrics: Measuring Product, Process and Project Quality eBook: Rex Black: Kindle Store. 2015. Amazon.com: Testing Metrics: Measuring Product, Process and Project Quality eBook: Rex Black: Kindle Store. [ONLINE] Available at: <http://www.amazon.com/Testing-Metrics-Measuring-Product-Process-ebook/dp/B008S2A25A/ref=sr_1_1?s=digital-text&ie=UTF8&qid=1430998135&sr=1-1&keywords=Testing+Metrics>. [Accessed 11 May 2015].

Software Test Metrics - Official Website of Edista Testing Institute. 2015.Software Test Metrics - Official Website of Edista Testing Institute. [ONLINE] Available at: <http://www.edistatesting.com/training/test-project-management/software-test-metrics/>. [Accessed 11 May 2015].

A Brief Overview Of Software Testing Metrics - ResearchGate. 2015. A Brief Overview Of Software Testing Metrics - ResearchGate. [ONLINE] Available at:<http://www.researchgate.net/publication/50247283_A_Brief_Overview_Of_Software_Testing_Metrics>. [Accessed 11 May 2015].

Grand Theft Auto V - Wikipedia, the free encyclopedia. 2015. Grand Theft Auto V - Wikipedia, the free encyclopedia. [ONLINE] Available at:<http://en.wikipedia.org/wiki/Grand_Theft_Auto_V>. [Accessed 11 May 2015].

Angry Birds (video game) - Wikipedia, the free encyclopedia. 2015. Angry Birds (video game) - Wikipedia, the free encyclopedia. [ONLINE] Available at:<http://en.wikipedia.org/wiki/Angry_Birds_%28video_game%29>. [Accessed 11 May 2015].

Important Software Test Metrics and Measurements – Explained with Examples and Graphs — Software Testing Help. 2015. Important Software Test Metrics and Measurements – Explained with Examples and Graphs — Software Testing Help. [ONLINE] Available at:<http://www.softwaretestinghelp.com/software-test-metrics-and-measurements/>. [Accessed 11 May 2015].

<http://www.stpmag.com/downloads/stp-0507_testmetrics.htm>