

Reporte final Calidad

Ivan Escalante

A01380976

Contenido

[Introducción 4](#_Toc26529973)

[Análisis de la Estimación de Tamaño 4](#_Toc26529974)

[1. ¿Cuál fue el tamaño (LDC AyM) de cada uno de tus programas? ¿Cuál fue tu promedio de LDC AyM desarrolladas por programa? 4](#_Toc26529975)

[2. ¿Cuál fue el programa con más LDC AyM y cuál el que tuvo menos? ¿Por qué? 4](#_Toc26529976)

[3. ¿Cuál fue el % de error en la estimación de tamaño (real/plan-1) en tus programas 2 al 7? En promedio, ¿Cuál fue el % de error en la estimación de tamaño de estos 6 programas juntos? 5](#_Toc26529977)

[4. ¿Cuál fue el programa con mayor error de estimación de tamaño y cuál el menor? ¿Por qué? 5](#_Toc26529978)

[5. En aquellos programas donde pudiste utilizar el método PROBE “A” o el “B”, ¿el tamaño real de LDC AyM estuvo dentro del intervalo de predicción estadístico del 70%? Si no estuvo, ¿Qué pasó? 5](#_Toc26529979)

[6. Basado en los datos históricos que has analizado sobre el error de estimación del tamaño, ¿Qué meta realista (alcanzable) te puedes poner para mejorar? 6](#_Toc26529980)

[7. ¿Qué cambios vas a hacerle a tu proceso personal para alcanzar esta meta? 6](#_Toc26529981)

[Análisis de la Estimación de Tiempo 6](#_Toc26529982)

[1. ¿Cuál fue el tiempo en horas dedicado a cada uno de tus programas? ¿Cuál fue tu promedio de horas dedicadas por programa? 6](#_Toc26529983)

[2. ¿Cuál fue el programa al que le dedicaste más tiempo y a cuál menos? ¿Por qué? 6](#_Toc26529984)

[3. ¿Cuál fue el % de error en la estimación de tiempo (real/plan-1) en tus programas 1 al 7? En promedio, ¿Cuál fue el % de error en la estimación de tiempo de estos 7 programas juntos? 7](#_Toc26529985)

[4. ¿Cuál fue el programa con mayor error de estimación de tiempo y cuál el menor? ¿Por qué? 7](#_Toc26529986)

[5. En aquellos programas donde pudiste utilizar el método PROBE “A” o el “B”, ¿El tiempo real estuvo dentro del intervalo de predicción estadístico del 70%? Si no estuvo, ¿Qué pasó? 7](#_Toc26529987)

[6. ¿Cuál fue la productividad de cada uno de tus programas? ¿Cuál fue tu productividad promedio? 8](#_Toc26529988)

[7. ¿Cuál es la tendencia de tu productividad? ¿Va a la baja, o al alta? ¿Es inestable? ¿Qué puedes hacer para mejorarla? 8](#_Toc26529989)

[8. ¿Qué tanto fueron afectados los errores de estimación de tiempo por tus errores de estimación del tamaño? 8](#_Toc26529990)

[9. Basado en los datos históricos que has analizado sobre el error de estimación del tiempo, ¿Qué meta realista (alcanzable) te puedes poner para mejorar? 9](#_Toc26529991)

[10. ¿Qué cambios vas a hacerle a tu proceso personal para alcanzar esta meta? 9](#_Toc26529992)

[Análisis de la Calidad (Defectos) 9](#_Toc26529993)

[1. ¿Qué tipo de defecto es al que más tiempo le dedicaste en la fase de compilación? ¿Y en la fase de pruebas? 9](#_Toc26529994)

[2. ¿Qué tipo de defecto es el que más inyectaste en la fase de Diseño? ¿Y en la fase de Codificación? 10](#_Toc26529995)

[3. ¿Cuál es el promedio de horas por defecto dedicadas al remover defectos en las fases de Revisión del Diseño, Revisión del Código, Compilación y Pruebas? 12](#_Toc26529996)

[4. ¿Cuál es la tendencia en los 7 programas del total de defectos inyectados por KLDC? 13](#_Toc26529997)

[5. ¿Cuál es la tendencia en los 7 programas de los defectos removidos por KLDC en las fases de Revisión del Diseño, Revisión del Código, Compilación y Pruebas? 14](#_Toc26529998)

[6. ¿Cómo se comparan las tasas de remoción de defectos (defectos removidos por hora) en las fases de Revisión del Diseño, Revisión del Código, Compilación y Pruebas en los 7 programas? 14](#_Toc26529999)

[7. ¿Cuál fue tu velocidad (LDC por hora) a la que revisaste el código de cada programa? 15](#_Toc26530000)

[8. ¿Cuál fue la palanca de remoción de defectos (DRL) de cada programa para las fases de Revisión del Diseño y Revisión del Código? ¿Es bueno o malo? 16](#_Toc26530001)

[9. ¿Existe alguna relación entre el Yield de cada programa y la velocidad (LDC por hora) a la que revisaste el código? 16](#_Toc26530002)

[10. ¿Existe alguna relación entre el Yield de cada programa y su A/FR? 17](#_Toc26530003)

[11. ¿Estás encontrando los defectos en las fases de Revisión del Diseño y del Código, o se te están escapando a las fases de Compilación y Pruebas? ¿Por qué? 17](#_Toc26530004)

[12. Basado en los datos históricos que has analizado sobre la calidad de tus programas, ¿Qué meta realista (alcanzable) te puedes poner para mejorar? 17](#_Toc26530005)

[13. ¿Qué cambios vas a hacerle a tu proceso personal para alcanzar esta meta? 17](#_Toc26530006)

[Conclusión: 18](#_Toc26530007)

[Párrafo de honestidad: 18](#_Toc26530008)

# Introducción

El documento que se esta entregando es un análisis de las métricas recopiladas a través de la metodología PSP del alumno IVAN EDUARDO ESCALANTE MUÑOZ en el semestre Ago-Diciembre 19.

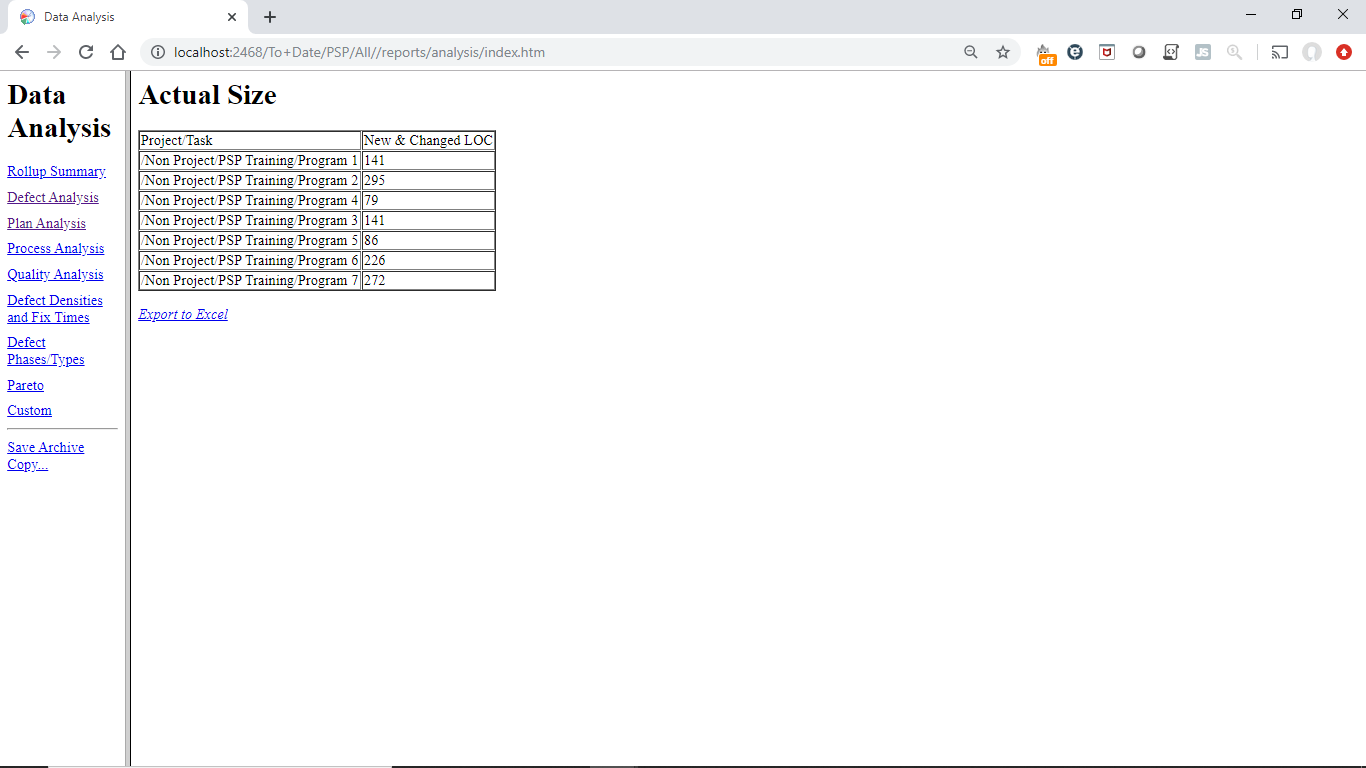
Las metricas analizadas recopiladas por la metodología Personal Software Process se dividen en 3 partes;

* Analisis de estimación de tamaño
* Analisis de estimación de tiempo
* Analisis de calidad

Se hacen diferentes preguntas para cada división, y al final se concluye un área de oportunidad de mejora para el proceso personal del mismo, como una autoretrospectiva para la mejora de nuestro proceso de desarrollo de software.

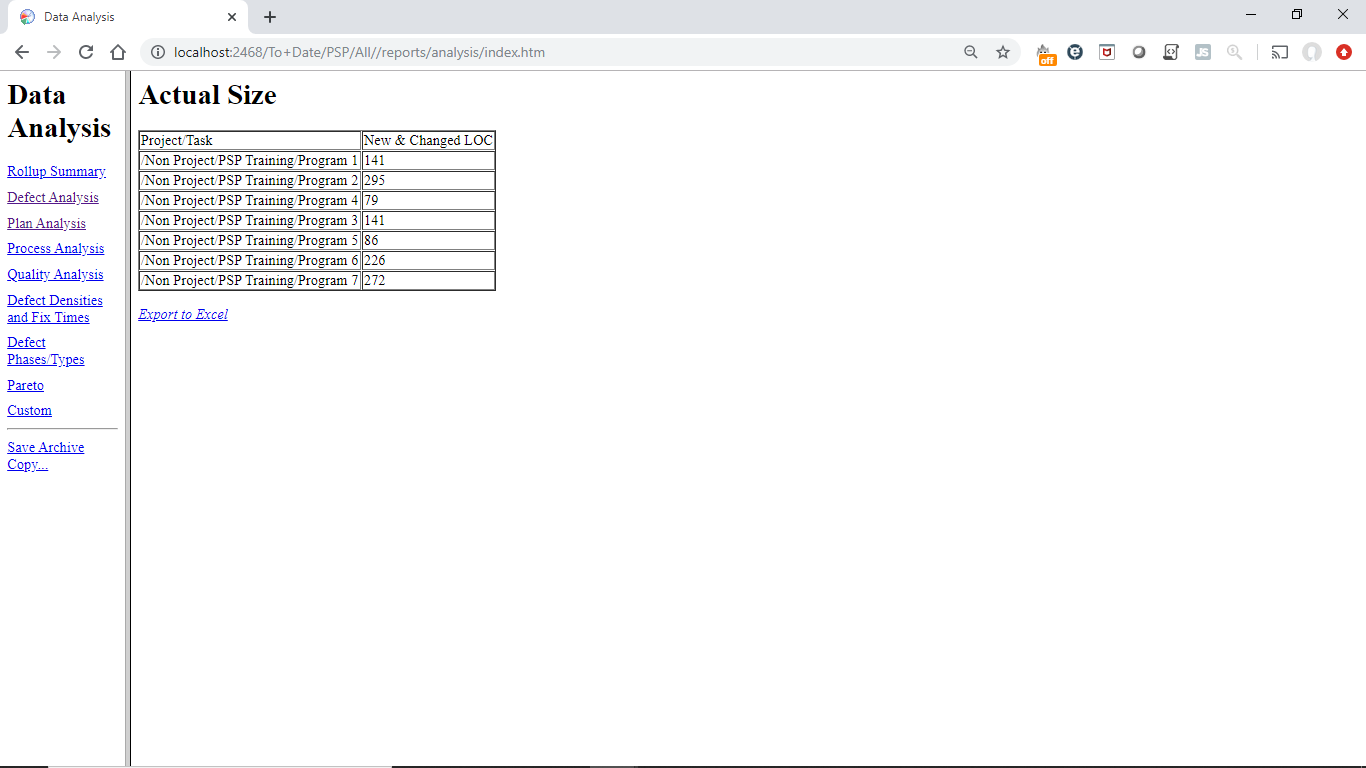
# Análisis de la Estimación de Tamaño

## ¿Cuál fue el tamaño (LDC AyM) de cada uno de tus programas? ¿Cuál fue tu promedio de LDC AyM desarrolladas por programa?



Promedio 177.1428571

## ¿Cuál fue el programa con más LDC AyM y cuál el que tuvo menos? ¿Por qué?



El programa con menos líneas de código Agregadas o modificadas fue e programa 4.

El programa 4 era bastante sencillo, después de hacer un diseño bien hecho, la traducción a código fue bastante sencilla y smooth.

El programa con MAS líneas de código agreadas o modificadas fue el 2.

Definitivamente el 2 fue el que mas líneas tenia puesto que era el programa mas completo y complicado, fue también el que mas tiempo me tomo en realizar.

## ¿Cuál fue el % de error en la estimación de tamaño (real/plan-1) en tus programas 2 al 7? En promedio, ¿Cuál fue el % de error en la estimación de tamaño de estos 6 programas juntos?

|  |  |
| --- | --- |
| Project/Task | % Error |
| /Non Project/PSP Training/Program 1 | #VALUE! |
| /Non Project/PSP Training/Program 2 | 36.10% |
| /Non Project/PSP Training/Program 4 | -53.30% |
| /Non Project/PSP Training/Program 3 | -46.40% |
| /Non Project/PSP Training/Program 5 | 86.90% |
| /Non Project/PSP Training/Program 6 | 35.70% |
| /Non Project/PSP Training/Program 7 | 70.20% |

El % de error fue 54.767% (promedio )

## ¿Cuál fue el programa con mayor error de estimación de tamaño y cuál el menor? ¿Por qué?

Mayor error de estimación fue El programa 5;

A la hora de estar leeyendo los requerimientos no pensé que fuera a ser tan complicado.

El de menor error de estimación fue el programa 4;

A la hora de ver los requerimientos pensé que era mucho mas trabajo que el que estuve realizando.

## En aquellos programas donde pudiste utilizar el método PROBE “A” o el “B”, ¿el tamaño real de LDC AyM estuvo dentro del intervalo de predicción estadístico del 70%? Si no estuvo, ¿Qué pasó?

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Programa | Métodos recomendados | LPI | UPI | Método utilizado |  | A & M | Dentro del limite? |
| 3 | C | - | - | C |  | - | - |
| 4 | C | - | - | C |  | - | - |
| 5 | C | - | - | C |  | - | - |
| 6 | C | - | - | C |  | - | - |
| 7 | C | - | - | C |  | 1240 | Si |

Me marcaba siempre que no aplicaban los LPI y los UPI posiblemente porque faltaban datos.

## Basado en los datos históricos que has analizado sobre el error de estimación del tamaño, ¿Qué meta realista (alcanzable) te puedes poner para mejorar?

A medida que vaya siendo mas consiente del tiempo que me tome realizar o implementar ciertos requerimientos, después de haber hecho una estimación, mis estimaciones se harán mas precisas. Es una cuestión de practica.

Me puedo poner como meta medible alcanzar un margen de error menor al 30% para los siguientes 10 programas que haga. A medida que vaya haciendo estimaciones, y me equivoque, podré analizar detenidamente el porque existen mis errores mentales de estimación.

Posteriormente buscar hacer un promedio de error menor a 20% en los siguientes 10 programas. En caso de fallar, repetir 20 para el margen menor a 20%. Y posteriormente en los siguientes 40, hacer que llegue a una varianza menor a 10%.

Tamaño es el que mas me fallo con un promedio de 54%.

## ¿Qué cambios vas a hacerle a tu proceso personal para alcanzar esta meta?

Dentro de mi proceso, la estimación la hacia justo después de leer los requerimientos, y me daba una idea medio basta y amplia del tiempo.

Lo cambiare a estimarlo hasta después de hacer el diseño completo de cuanto me tardare, dentro del diseño tengo una visión mas absoluta de que eso que voy a necesitar programar, cuanto tiempo me tomará hacerlo y si tengo experiencia en el tema o tendré que buscar documentación en línea para aprender a hacer las cosas.

# Análisis de la Estimación de Tiempo

## ¿Cuál fue el tiempo en horas dedicado a cada uno de tus programas? ¿Cuál fue tu promedio de horas dedicadas por programa?

|  |  |
| --- | --- |
| Project/Task | Hours |
| /Non Project/PSP Training/Program 1 | 2.07 |
| /Non Project/PSP Training/Program 2 | 8.5 |
| /Non Project/PSP Training/Program 4 | 2.25 |
| /Non Project/PSP Training/Program 3 | 2.5 |
| /Non Project/PSP Training/Program 5 | 2.38 |
| /Non Project/PSP Training/Program 6 | 3.53 |
| /Non Project/PSP Training/Program 7 | 3.35 |

Promedio

3.511 horas

## ¿Cuál fue el programa al que le dedicaste más tiempo y a cuál menos? ¿Por qué?

Al que mas le dedique fue al programa 2.

El programa 2, como lo comente arriba también, era de los programas mas difíciles y completos, tenia retos y requerimientos con los que no había trabajado antes.

Al que menos le dedique fue al 1.

Era un programa bastante sencillo, no tenia tanto reto.

## ¿Cuál fue el % de error en la estimación de tiempo (real/plan-1) en tus programas 1 al 7? En promedio, ¿Cuál fue el % de error en la estimación de tiempo de estos 7 programas juntos?

|  |  |
| --- | --- |
| Project/Task | % Error |
| /Non Project/PSP Training/Program 1 | 3.33% |
| /Non Project/PSP Training/Program 2 | 64.1% |
| /Non Project/PSP Training/Program 4 | -47.3% |
| /Non Project/PSP Training/Program 3 | -46.7% |
| /Non Project/PSP Training/Program 5 | 85.0% |
| /Non Project/PSP Training/Program 6 | -23.2% |
| /Non Project/PSP Training/Program 7 | -7.36% |

Promedio = 39.57%

## ¿Cuál fue el programa con mayor error de estimación de tiempo y cuál el menor? ¿Por qué?

Mayor error de estimación de tiempo programa 5 :

Porque tenia un error que no encontraba, pensé que era funcinoal, pensé que era cuestión de la memoria o algún problema de como comunicaba la información entre clases… y no… era operacional, puse un + envez de un \*.

Menor error de estimación de tiempo, programa 7 ;

Porque fue el programa que mas formalmente hice el diseño y la planeación, aparte tenia un poco ya de practica observando cuanto tiempo asumía que me iba a tardar y cuanto realmente me tardaba.

## En aquellos programas donde pudiste utilizar el método PROBE “A” o el “B”, ¿El tiempo real estuvo dentro del intervalo de predicción estadístico del 70%? Si no estuvo, ¿Qué pasó?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Programa | Métodos recomendados | LPI | UPI | Método utilizado | Horas | Dentro del limite? |
| 3 | C3 , C1, C2 | - | - | C3 | - | - |
| 4 | C3, C1, C2 | - | - | C2 | - | - |
| 5 | C3, C1, C2 | - | - | C1 | - | - |
| 6 | C3, C1, C2 | - | - | C1 | - | - |
| 7 | C3, C1, C2 | - | - | C2 | - | - |

Me recomendó en todos los programas utilizar la metodología para TAMAÑO C, y como siempre me recomendó la C, solo fue natural que me recomendara en mteodologías probe para TIEMPO, C1,C2, C3; lo que hace que no apliquen tampoco los limites inferiores y superiores de PI.

## ¿Cuál fue la productividad de cada uno de tus programas? ¿Cuál fue tu productividad promedio?

|  |  |
| --- | --- |
| Project/Task | Productivity (LOC/Hr) |
| /Non Project/PSP Training/Program 1 | 68.2 |
| /Non Project/PSP Training/Program 2 | 34.7 |
| /Non Project/PSP Training/Program 4 | 35.1 |
| /Non Project/PSP Training/Program 3 | 56.4 |
| /Non Project/PSP Training/Program 5 | 36.1 |
| /Non Project/PSP Training/Program 6 | 64 |
| /Non Project/PSP Training/Program 7 | 81.2 |

Productividad promedio 53.67142857

## ¿Cuál es la tendencia de tu productividad? ¿Va a la baja, o al alta? ¿Es inestable? ¿Qué puedes hacer para mejorarla?

No tengo punto de referencia para saber si es alta o baja, pero asumo que 34 lineas de código por hora, con una buena calidad es promedio.

Siento que en el programa 7, fue mucho mas rápido, porque fue cuando realmente hice el disñeo y planeación formal de mi trabajo.

## ¿Qué tanto fueron afectados los errores de estimación de tiempo por tus errores de estimación del tamaño?

|  |  |
| --- | --- |
| Tiempo | Tamaño |
| |  |  | | --- | --- | | Project/Task | % Error | | /Non Project/PSP Training/Program 1 | 3.33% | | /Non Project/PSP Training/Program 2 | 64.1% | | /Non Project/PSP Training/Program 4 | -47.3% | | /Non Project/PSP Training/Program 3 | -46.7% | | /Non Project/PSP Training/Program 5 | 85.0% | | /Non Project/PSP Training/Program 6 | -23.2% | | /Non Project/PSP Training/Program 7 | -7.36% |   Promedio = 39.57% | |  |  | | --- | --- | | Project/Task | % Error | | /Non Project/PSP Training/Program 1 | #VALUE! | | /Non Project/PSP Training/Program 2 | 36.10% | | /Non Project/PSP Training/Program 4 | -53.30% | | /Non Project/PSP Training/Program 3 | -46.40% | | /Non Project/PSP Training/Program 5 | 86.90% | | /Non Project/PSP Training/Program 6 | 35.70% | | /Non Project/PSP Training/Program 7 | 70.20% |   El % de error fue 54.767% (promedio ) |

A medida que fue mas grande el programa de lo que yo había estimado, me tomo mas tiempo. Es directamente proporcional.

## Basado en los datos históricos que has analizado sobre el error de estimación del tiempo, ¿Qué meta realista (alcanzable) te puedes poner para mejorar?

Considero que este tipo de estimación de “oráculo” es una habilidad que se va perfeccionando o se va haciendo mas precisa a medida que la voy practicando mas. Mi meta seria seguir practicando hacer mi estimaciones en diferentes aspectos de mi trabajo para así tener una base de datos mental histórica en cuanto a cuanto tiempo em toma ciertas actividades para hace runa mejor planeación de mi trabajo.

Al igual que con el tamaño, buscare hacer que mis estimaciones tengan un margen de error menor del 20% como varianza. También comparten el área de oportunidad de crecimiento de ver y analizar detenidamente el porque de mis errores de cálculos y asumptions mentales a la hora de pretender querer saber cuanto me tardare.

Me propongo tener una variación menor al 20% en los siguientes 15 programas, porque aunque el tiempo no me fallo tanto como el tamaño, una se alimenta de la otra.

Posteriormente una variación menor a 15% en los siguientes 20. Y como meta 10% en los siguientes 50.

## ¿Qué cambios vas a hacerle a tu proceso personal para alcanzar esta meta?

Antes hacia la estimación de tiempo después de leer la lista de requerimientos de el programa por hacer.

Hacer la estimación de tiempo después de hacer los Formatos de especificación Logica, funcional y operacional me permitirá tener una visión mucho mas grande y amplia de lo que estoy por hacer, estoy seguro que me ayudará a bajar ese % de error considerablemente.

# Análisis de la Calidad (Defectos)

## ¿Qué tipo de defecto es al que más tiempo le dedicaste en la fase de compilación? ¿Y en la fase de pruebas?

|  |  |  |  |
| --- | --- | --- | --- |
| **Defect Type** | **Number of defects at Compile Entry** | **Number of defects found in Compile** | **Percentage of Type found by the Compiler** |
| Assignment | - | - | - |
| Build, package | 1 | 1 | 100% |
| Checking | - | - | - |
| Data | 1 | 1 | 100% |
| Documentation | - | - | - |
| Environment | - | - | - |
| Function | 7 | 5 | 71% |
| Interface | - | - | - |
| Syntax | 16 | 16 | 100% |
| System | - | - | - |
| Total | 25 | 23 | 92% |

Definitivamente de Funcional, aunque fueron menos, los tiempos de corrección funcioales fueron los siguientes;

Y los de sintaxis

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 25.9 | 5.4 | 15.9 | 2.1 | 0.3 | 61.5 |

A pesar de que fueron 16, el promedio fue menor a 1; sumados fueron menos de 20 minutos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Defect Fix Times** | | | | |
|  | | Defects Found in Compile | Defects Found in Test | Total Defects Found |
| Defects Injected in Design | Tot. fix time | - | - | 34.7 |
| Tot. defects | - | - | [1](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?qf=compProj.rpt&inj=Design) |
| Avg. fix time | - | - | 34.7 |
| Defects Injected in Code | Tot. fix time | 108.4 | 15.9 | 134.3 |
| Tot. defects | [23](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?qf=compProj.rpt&inj=Code&rem=Compile) | [1](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?qf=compProj.rpt&inj=Code&rem=Test) | [27](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?qf=compProj.rpt&inj=Code) |
| Avg. fix time | 4.71 | 15.9 | 4.97 |
| Total Defects Injected | Tot. fix time | 108.4 | 15.9 | 169 |
| Tot. defects | [23](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?qf=compProj.rpt&rem=Compile) | [1](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?qf=compProj.rpt&rem=Test) | [28](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?qf=compProj.rpt&) |
| Avg. fix time | 4.71 | 15.9 | 6.04 |

## ¿Qué tipo de defecto es el que más inyectaste en la fase de Diseño? ¿Y en la fase de Codificación?

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Number Injected** | | **Percentage Injected** | | **Number Removed** | | **Percentage Removed** | |
| Type | Design | Code | Design | Code | Compile | Test | Compile | Test |
| [Assignment](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?strict&qf=compProj.rpt&type=Assignment) | - | - | - | - | - | - | - | - |
| [Build, package](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?strict&qf=compProj.rpt&type=Build%2C+package) | - | [1](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?inj=Code&strict&qf=compProj.rpt&type=Build%2C+package) | - | 4% | [1](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?rem=Compile&strict&qf=compProj.rpt&type=Build%2C+package) | - | 4% | - |
| [Checking](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?strict&qf=compProj.rpt&type=Checking) | - | - | - | - | - | - | - | - |
| [Data](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?strict&qf=compProj.rpt&type=Data) | - | [1](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?inj=Code&strict&qf=compProj.rpt&type=Data) | - | 4% | [1](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?rem=Compile&strict&qf=compProj.rpt&type=Data) | - | 4% | - |
| [Documentation](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?strict&qf=compProj.rpt&type=Documentation) | - | - | - | - | - | - | - | - |
| [Environment](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?strict&qf=compProj.rpt&type=Environment) | - | - | - | - | - | - | - | - |
| [Function](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?strict&qf=compProj.rpt&type=Function) | [1](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?inj=Design&strict&qf=compProj.rpt&type=Function) | [7](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?inj=Code&strict&qf=compProj.rpt&type=Function) | 100% | 26% | [5](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?rem=Compile&strict&qf=compProj.rpt&type=Function) | [1](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?rem=Test&strict&qf=compProj.rpt&type=Function) | 22% | 100% |
| [Interface](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?strict&qf=compProj.rpt&type=Interface) | - | - | - | - | - | - | - | - |
| [Syntax](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?strict&qf=compProj.rpt&type=Syntax) | - | [18](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?inj=Code&strict&qf=compProj.rpt&type=Syntax) | - | 67% | [16](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?rem=Compile&strict&qf=compProj.rpt&type=Syntax) | - | 70% | - |
| [System](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?strict&qf=compProj.rpt&type=System) | - | - | - | - | - | - | - | - |
| [Total](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?strict&qf=compProj.rpt) | [1](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?inj=Design&strict&qf=compProj.rpt) | [27](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?inj=Code&strict&qf=compProj.rpt) | 100% | 100% | [23](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?rem=Compile&strict&qf=compProj.rpt) | [1](http://localhost:2468/To+Date/PSP/All/reports/defectlog.class?rem=Test&strict&qf=compProj.rpt) | 100% | 100% |

Diseño – FUNCION

Codificacion – SYNTAXIS

## ¿Cuál es el promedio de horas por defecto dedicadas al remover defectos en las fases de Revisión del Diseño, Revisión del Código, Compilación y Pruebas?

A partir de que encontré todos los errores en al fase de compilación, me dedique a arreglar los problemas ahí mismo.

Considero que tuve un pequeño error de registro a la hora de estar metiendo la información de cuando arreglé los errores como;

Compilo exitosamente pero marco un resultado diferente al que tenia registrado en el documento de pruebas; pero como aun no llegaba a esa prueba me tome el tiempo de comenzar a buscarlo, dentro de la fase de compilación, aunque realmente era de la parte de pruebas.

|  |  |
| --- | --- |
| Fase | Tiempo dedicado en min |
| Revisión del Diseño | 0 |
| Revisión del Código | 34.7 |
| Compilación | 108.4 |
| Pruebas | 15.9 |

Me tome un total de 159 minutos entre 4 tipos de fases, en teoría da un promedio 39.75 minutos entre las 4 fases, aunque es información con ruido de investigación.

Entre 3 es 53 minutos = .8833 horas

Entre 4 es 39.75 = .6625 horas

|  |  |  |  |
| --- | --- | --- | --- |
| Design Review/Estimated Defects Removed per Hour | Test/Estimated Defects Removed per Hour | Code Review/Estimated Defects Removed per Hour | Compile/Estimated Defects Removed per Hour |
| 0 | 0.5 | 0 | 4.61 |

|  |
| --- |
| Compile/Estimated Defects Removed per Hour |
| 4.61 |

En promedio de los 4 son 1.2775 defectos removidos por hora.

## ¿Cuál es la tendencia en los 7 programas del total de defectos inyectados por KLDC?

|  |
| --- |
| Defects/KLOC |
| 21.3 |
| 20.3 |
| 38 |
| 28.4 |
| 46.5 |
| 8.85 |
| 18.4 |
| 21.8 |
|  |

Promedio 29.07857143

Mi tendencia fue que introduzco 29 defectos por cada 1,000 lineas de código.

Tuve un peak en el 5, pero es congruente con mi mala estimacion de tamaño/tiempo.

## ¿Cuál es la tendencia en los 7 programas de los defectos removidos por KLDC en las fases de Revisión del Diseño, Revisión del Código, Compilación y Pruebas?

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase Defects** | | | |
| Defects Found in Compile | Compile Defects/KLOC | Defects Found in Test | Test Defects/KLOC |
| 0 | 0 | 0 | 0 |
| 6 | 20.3 | 0 | 0 |
| 2 | 25.3 | 1 | 12.7 |
| 4 | 28.4 | 0 | 0 |
| 4 | 46.5 | 0 | 0 |
| 2 | 8.85 | 0 | 0 |
| 5 | 18.4 | 0 | 0 |
| 23 | 18.5 | 1 | 0.81 |
| 6.571428571 | 23.75 | 0.285714286 | 1.93 |

Mis tendencias en promedio por KLDC’s son en promedio;

* 6.6 defectos encontrados en compilacio
* Introduzco 23 por cada mil líneas de código.

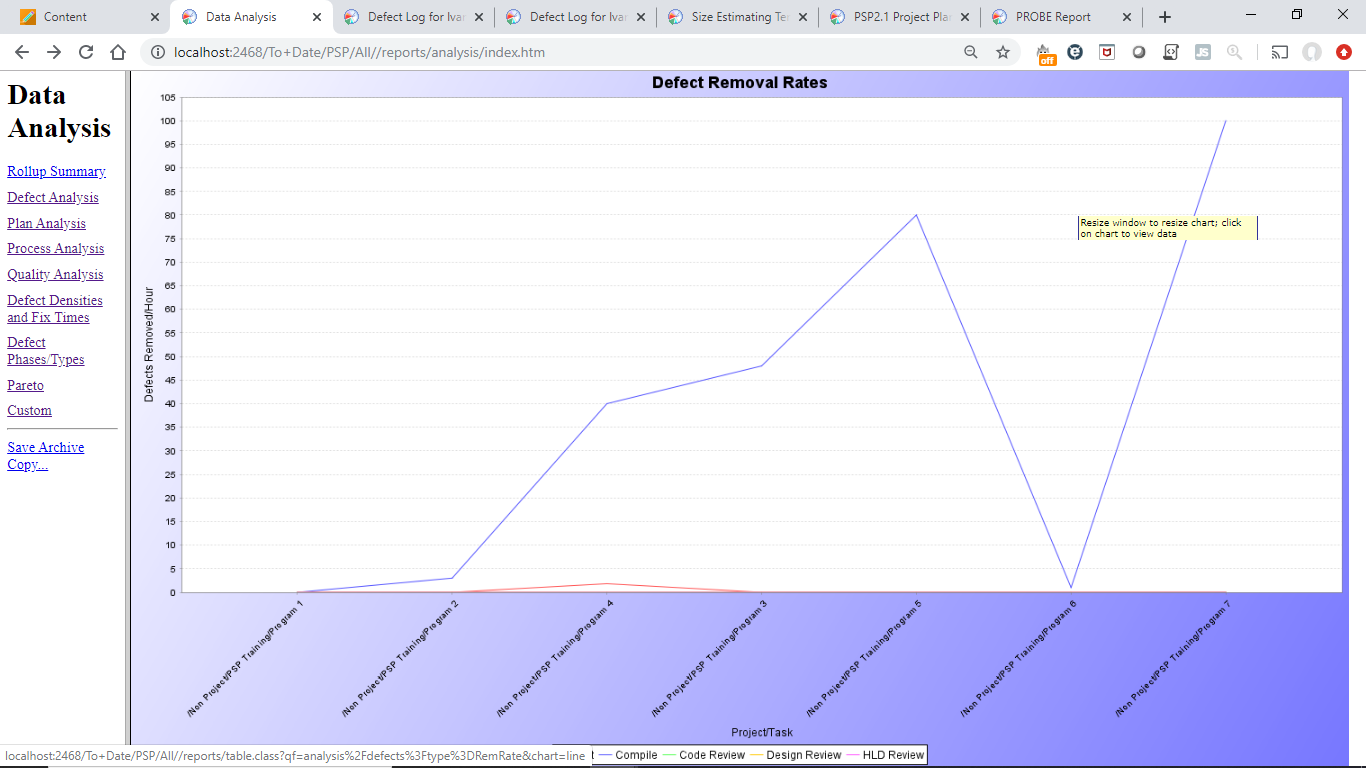
En la fase de testing tengo un promedio bastante bajo, de .28, introduzco 1.93 errroes por cada 1000 lineas de código.

## ¿Cómo se comparan las tasas de remoción de defectos (defectos removidos por hora) en las fases de Revisión del Diseño, Revisión del Código, Compilación y Pruebas en los 7 programas?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Defect Fix Times** | | | | |
|  | | Defects Found in Compile | Defects Found in Test | Total Defects Found |
| Defects Injected in Design | Tot. fix time | - | - | 34.7 |
| Tot. defects | - | - | 1 |
| Avg. fix time | - | - | 34.7 |
| Defects Injected in Code | Tot. fix time | 108.4 | 15.9 | 134.3 |
| Tot. defects | 23 | 1 | 27 |
| Avg. fix time | 4.71 | 15.9 | 4.97 |
| Total Defects Injected | Tot. fix time | 108.4 | 15.9 | 169 |
| Tot. defects | 23 | 1 | 28 |
| Avg. fix time | 4.71 | 15.9 | 6.04 |

(Defects Removed/Hour)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project/Task | Test | Compile | Code Review | Design Review | HLD Review |
| /Non Project/PSP Training/Program 1 | 0 | #DIV/0! | #DIV/0! | #DIV/0! |  |
| /Non Project/PSP Training/Program 2 | 0 | 3 | #DIV/0! | #DIV/0! |  |
| /Non Project/PSP Training/Program 4 | 1.82 | 40 | 0 | 0 |  |
| /Non Project/PSP Training/Program 3 | 0 | 48 | 0 | 0 |  |
| /Non Project/PSP Training/Program 5 | 0 | 80 | 0 | 0 |  |
| /Non Project/PSP Training/Program 6 | 0 | 0.99 | 0 | 0 |  |
| /Non Project/PSP Training/Program 7 | 0 | 100 | 0 | 0 |  |

Se puede observar que constantemente fueron creciendo los defectos removidos en la parte de compilación y uno que otro en la parte de pruebas.

Tengo la teoría que a medida que comience a mejorar la revisión de mi código en los errores que creo con mas frecuencia, que en este caso son los de syntaxis, podré disminuir considerablemente el tiempo de defectos removidos /hora.

Tuve la oportunidad de ver como se comportaban los defectos de otros compañeros y mi hipótesis parece ser cierta, pero aun falta ponerlo en practica y revisar los resultados de mi investigación.

## ¿Cuál fue tu velocidad (LDC por hora) a la que revisaste el código de cada programa?

|  |  |  |  |
| --- | --- | --- | --- |
| Project/Task | Total | Design Review | Code Review |
| /Non Project/PSP Training/Program 1 | #DIV/0! | #DIV/0! | #DIV/0! |
| /Non Project/PSP Training/Program 2 | #DIV/0! | #DIV/0! | #DIV/0! |
| /Non Project/PSP Training/Program 4 | 1185 | 2370 | 2370 |
| /Non Project/PSP Training/Program 3 | 2115 | 2820 | 8460 |
| /Non Project/PSP Training/Program 5 | 62.2 | 64.5 | 1720 |
| /Non Project/PSP Training/Program 6 | 6780 | 13560 | 13560 |
| /Non Project/PSP Training/Program 7 | 2720 | 4080 | 8160 |

## ¿Cuál fue la palanca de remoción de defectos (DRL) de cada programa para las fases de Revisión del Diseño y Revisión del Código? ¿Es bueno o malo?

# Defect Removal Leverage

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project/Task | Test | Compile | Code Review | Design Review | HLD Review |
| /Non Project/PSP Training/Program 1 | 1 | #DIV/0! | #DIV/0! | #DIV/0! |  |
| /Non Project/PSP Training/Program 2 | 1 | #DIV/0! | #DIV/0! | #DIV/0! |  |
| /Non Project/PSP Training/Program 4 | 1 | 22 | 0 | 0 |  |
| /Non Project/PSP Training/Program 3 | 1 | #DIV/0! | #DIV/0! | #DIV/0! |  |
| /Non Project/PSP Training/Program 5 | 1 | 78.7 | 0 | 0 |  |
| /Non Project/PSP Training/Program 6 | 1 | 1.24 | 0 | 0 |  |
| /Non Project/PSP Training/Program 7 | 1 | 147 | 0 | 0 |  |

Que no tenga valores en las fases de revisión de código y la revisión de diseño es malo, puesto que significa que todo el trabajo de revisión lo estoy haciendo en la fase de compilación. Esto es una mala practica que va en contra de lo que estamos viendo ahorita en esta materia.

## ¿Existe alguna relación entre el Yield de cada programa y la velocidad (LDC por hora) a la que revisaste el código?

|  |  |  |
| --- | --- | --- |
| Project/Task | Design Review LOC/Hr | Design Review Yield % |
| /Non Project/PSP Training/Program 1 | #DIV/0! | 0% |
| /Non Project/PSP Training/Program 2 | #DIV/0! | #DIV/0! |
| /Non Project/PSP Training/Program 4 | 2370 | #DIV/0! |
| /Non Project/PSP Training/Program 3 | 2820 | #DIV/0! |
| /Non Project/PSP Training/Program 5 | 64.5 | #DIV/0! |
| /Non Project/PSP Training/Program 6 | 13560 | #DIV/0! |
| /Non Project/PSP Training/Program 7 | 4080 | #DIV/0! |

Desgraciadamente no tengo datos para calcular este tipo de datos.

## ¿Existe alguna relación entre el Yield de cada programa y su A/FR?

|  |  |
| --- | --- |
| Project/Task | A/FR |
| /Non Project/PSP Training/Program 1 | 0 |
| /Non Project/PSP Training/Program 2 | 0 |
| /Non Project/PSP Training/Program 4 | 0.11 |
| /Non Project/PSP Training/Program 3 | 0.25 |
| /Non Project/PSP Training/Program 5 | 7.55 |
| /Non Project/PSP Training/Program 6 | 0.01 |
| /Non Project/PSP Training/Program 7 | 0.38 |

|  |  |
| --- | --- |
| Project/Task | Yield % |
| /Non Project/PSP Training/Program 1 | 100% |
| /Non Project/PSP Training/Program 2 | 0% |
| /Non Project/PSP Training/Program 4 | 0% |
| /Non Project/PSP Training/Program 3 | 0% |
| /Non Project/PSP Training/Program 5 | 0% |
| /Non Project/PSP Training/Program 6 | 0% |
| /Non Project/PSP Training/Program 7 | 0% |

No existe una relación, porque realmente todos los defectos los saque hasta la parte de compilación, no me di cuenta.

## ¿Estás encontrando los defectos en las fases de Revisión del Diseño y del Código, o se te están escapando a las fases de Compilación y Pruebas? ¿Por qué?

Se están escapando a la parte de compilación y de pruebas.

Definitivamente porque no le estoy dando el tiempo de revisión que requieren el diseño y el código antes de la compilación. Pero personalmente me es mas fácil que me diga la computadora donde están que buscarlos manual/visualmente.

## Basado en los datos históricos que has analizado sobre la calidad de tus programas, ¿Qué meta realista (alcanzable) te puedes poner para mejorar?

La calidad de mis programas aumenta cuando hago una buena planeación y diseño de lo que estoy por hacer, me propongo como meta realista alcanzar a bajar mis defectos introducidos por mil líneas de código

Actualmente es 29.07, pero constantemente me encuentro corrigiendo errores de syntaxis por no hacer una buena revisión de código.

Propongo como meta alcanzable bajar de 29 a 25 errores por mil líneas de código en los siguientes 10 programas.

Como meta grande me propongo a bajarlo a menos de 20 en los siguientes 50. Desglosado seria

Menos de 25 en los siguientes 30

Menos de 18 en los siguientes 20(después de los 30).

## ¿Qué cambios vas a hacerle a tu proceso personal para alcanzar esta meta?

Hacer la buena planeación y diseño de las clases que hare a partir de los requerimientos que se me entregan.

Dentro de mi proceso personal de codificación, le dedicaré un verdadero tiempo detenido a revisar y buscar errores de syntaxis entre lo que he escrito.

# Conclusión:

Después de revisar los datos y hacer un análisis un algo profundo de los datos y mi comportamiento como programador encontré que existen algunas relaciones entre mis tiempos, defectos y estimaciones. En repetidas ocasiones hice hincapié que una forma de mejorar mi proceso y calidad de programación era hacer una mejor planeación y diseño de las cosas que iba a programar.

Sin querer prolongar la misma información que he repetido a lo largo de éste documento, puedo comprehender que el diseño realmente si representa una parte importante de el aumento de calidad.

# Párrafo de honestidad:

Por medio de este párrafo yo \_\_\_\_\_\_Ivan Escalante\_\_ certifico que soy el autor intelectual del material que arriba entrego y que no realicé una copia de ninguna otra persona. En aquellos casos en los que tuve que incluir material de otra persona, reporté su apropiada referencia.