# Introducción.

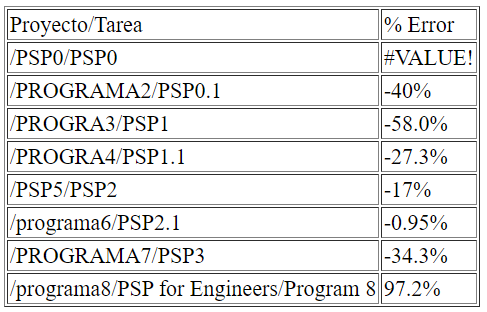
Durante el proceso personal del software se entiende que es un sistema para el desarrollo personal que permite al propio desarrollador o ingeniero a conocer su propia forma de hacer las cosas. Además, de introducirlo a la guía de una planeación y diseño. Ciertamente muchas veces, los desarrolladores en sus comienzos pueden dejar de lado la planeación y el diseño y entrar directamente a codificar. Dentro de la ingeniería de software los modelos de desarrollo no son solo una serie de pasos para para generar proyectos, sino también una forma de garantizar el software de calidad. Es así que a través de este curso se plantea poco a poco se introduzca los ingenieros y/o desarrolladores en el concepto *ciclo de vida del desarrollo de software* y a continuación examinaremos los resultados de este.

# Planeación

|  |  |  |
| --- | --- | --- |
| Fase # | Propósito | Para orientar el análisis y la redacción del Informe final del PSP. |
|  | Criterio para entrar | • Completa todos los programas de PSP  • Completar todas las asignaciones de informes anteriores |
|  | Planificación | * producir un informe sobre lo que ha aprendido en este curso * comprender su rendimiento actual de desarrollo de software y sus áreas de mayor prioridad para mejorar * comprender cómo y por qué cambió su desempeño entre el informe intermedio y el programa final * Adquirir experiencia en el establecimiento de objetivos medibles y la definición de las acciones de mejora correspondientes. * aprender a actualizar un proceso personal |
|  |  | * Durante la planeación, habrá que reunir todos los reportes de los programas 1 al 8 * Plan summay * Estimación de tamaño * Reporte de tiempo * Reporte de defectos * Reporte r3 y r4 * Reporte PROBE * Así como las gráficas de:  |  | | --- | | • Análisis de Defectos  • Análisis de Planeación  • Análisis de Proceso  • Análisis de Calidad |   Se darán respuestas a las siguientes preguntas  Análisis de precisión de estimación de tamaño Compare la línea de base del informe provisional para determinar la precisión de la estimación del tamaño con los programas posteriores al informe. ¿Cuánto cambió su precisión de estimación de tamaño? ¿Por qué?  ¿Con qué frecuencia el tamaño real de mi programa estuvo dentro de mi intervalo de predicción estadística del 70%?  ¿Tengo tendencia a agregar / perder objetos enteros?  ¿Tengo una tendencia a juzgar mal el tamaño relativo de los objetos?  ¿Necesito calcular los datos de rango de tamaño relativo utilizando mis datos de objetos históricos? ¿Yo puedo?  Según mis datos históricos de precisión de estimación de tamaño, ¿cuál es una meta de estimación de tamaño realista para mí?  ¿Cómo puedo cambiar mi proceso para alcanzar ese objetivo?  Análisis de la precisión de la estimación del tiempo Compare la línea de base del informe provisional para determinar la precisión de la estimación del tiempo con los programas posteriores al informe. ¿Cuánto cambió su tiempo estimando la precisión? ¿Por qué?  ¿Con qué frecuencia estuvo mi tiempo de desarrollo real dentro de mi intervalo de predicción estadística del 70%?  ¿Mi productividad es estable? ¿Por qué o por qué no?  ¿Cómo puedo estabilizar mi productividad?  ¿En qué medida se ven afectadas mis estimaciones de tiempo por la precisión de mis estimaciones de tamaño? (¿Me ayudaría la regresión múltiple?)  Según mis datos históricos de precisión de estimación de tiempo, ¿cuál es una meta realista de estimación de tiempo para mí?  ¿Cómo puedo cambiar mi proceso para alcanzar ese objetivo?  Continúa en la siguiente página  Preguntas de estudio, continuación  Análisis de defectos y rendimiento ¿Qué tipo de defectos inyecto durante el diseño y la codificación?  ¿Qué tendencias son evidentes en los defectos por unidad de tamaño (por ejemplo, KLOC) que se encuentran en las revisiones, compilaciones y pruebas?  ¿Qué tendencias son evidentes en el total de defectos por unidad de tamaño?  ¿Cómo se comparan mis tasas de eliminación de defectos (defectos eliminados / hora) para la revisión del diseño, la revisión del código, la compilación y la prueba?  ¿Cuáles son mis tasas de revisión (tamaño revisado / hora) para revisión de diseño y revisión de código?  ¿Cuáles son mis ventajas de eliminación de defectos para la revisión del diseño, la revisión del código y la compilación frente a la prueba unitaria?  ¿Existe alguna relación entre el rendimiento y la tasa de revisión (tamaño revisado / hora) para las revisiones de diseño y código?  ¿Existe una relación entre el rendimiento y el A / FR para los programas 5 a 8?  Análisis de calidad Compare la línea de base del informe provisional para la calidad en la prueba unitaria con los programas posteriores al informe. ¿Cuánto cambió la calidad de los programas que ingresaron a la prueba unitaria? ¿Por qué?  ¿Cómo puedo juzgar la calidad de mi producto final al principio de mi ciclo de desarrollo?  ¿Estoy encontrando defectos en las revisiones de diseño y código? ¿Por qué o por qué no?  ¿Cómo puedo hacer que mi proceso sea más efectivo y eficiente?  Según mis datos históricos, ¿cuáles son algunos de mis objetivos de calidad realistas?  ¿Cómo puedo cambiar Adjuntar en un directorio los siguientes archivos:   * • Análisis del Reporte 4 del programa cuatro. * • Informe del Reporte intermedio. * • Obtener el Generic Project Planning Summary.   • Generar la Size Estimating Template. mi proceso para alcanzar esos objetivos? |
|  | Post mortem | Redacción de conclusiones tomando en cuenta   * identificar dónde necesita mejorar * establecer objetivos de mejora desafiantes   comprender cómo cambiar su proceso personal para alcanzar esos objetivos |
|  | Criterio de salida | Adjuntar en un directorio los siguientes archivos:  • Informe del Reporte final.  • Obtener el Generic Project Planning Summary.  • Generar la Size Estimating Template. |

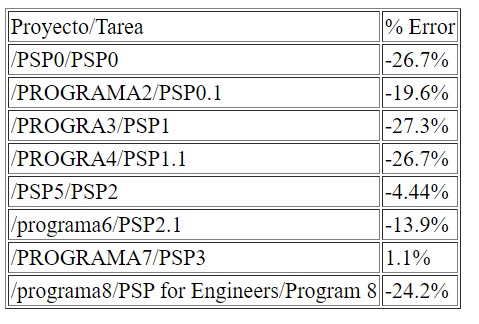
# Análisis.

• Analice su precisión de estimación de tamaño y determine el grado en que sus estimaciones estuvieron dentro de los intervalos de predicción estadística del 70%. También muestre cómo ha evolucionado su precisión de estimación de tamaño durante las asignaciones.



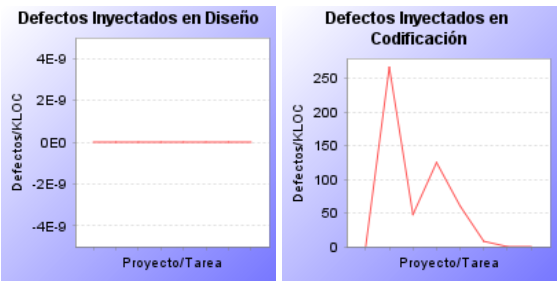
En las estimaciones de tamaño con relación a los proyectos han ido en decrecimiento en donde cada vez se va acercando el tamaño estimado al tamaño real, bajando el porcentaje de error en los calculos. A excepcion del programa final donde a pesar de la estimacion, se agregaron ma slineas durante el Desarrollo.

• Analice la precisión de la estimación del tiempo y determine el grado en el que sus estimaciones estuvieron dentro de los intervalos de predicción estadística del 70%. También muestre cómo ha evolucionado la precisión de su estimación del tiempo durante las asignaciones.



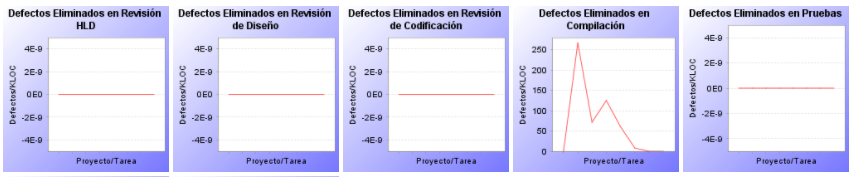
En las estimaciones de tiempo de los proyectos han ido variando, e su scenario donde se platea reutilizar codigo, hay algunos programas que pueden reducirse el tiempo, lo que impode una revisionde codigo para integrarlo y/o modificarlo de manera adecuada

• Analice los tipos de defectos que inyectó en el diseño y en la codificación. Incluya un análisis de Pareto de estos tipos de defectos.



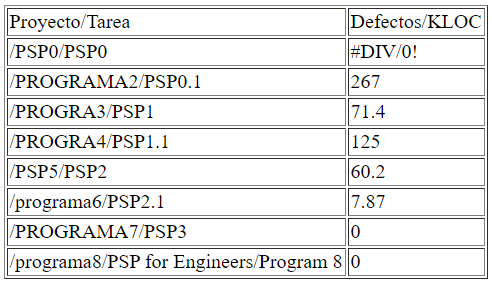
Gracias a la espericand¿cia labroral previa al curso, el diseño es un apartadon donde no se han observado errores se ha logrado trabjar muy bien esa parte. La mayor parte de errores se han generado durante la codificacion y han disminuido conforme se ha avanzado en el cuso.

• analizar sus tendencias en busca de defectos por unidad de tamaño (por ejemplo, KLOC) encontrados en la revisión del diseño, revisión del código, compilación y prueba



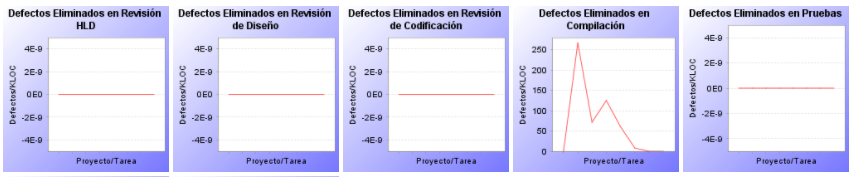
La mayor parte de los defectos fue encontrada durante la compilacion

• analice su tendencia de defectos totales por unidad de tamaño



Atraves de las reviciones y buscar los defectos en donde antes se presentaban, se propido revisar y reducirlos. Al punto de no tener errores en los programas finales

• analice sus tasas de eliminación de defectos para revisiones de diseño, revisiones de código, compilación y prueba, y su ventaja de eliminación de defectos para revisiones de diseño, revisiones de código y compilación versus prueba unitaria. En aquellos casos en los que no tuvo defectos de prueba, utilice la tasa promedio de eliminación de defectos de prueba unitaria para los programas desarrollados hasta la fecha.



La compilacion es donde se han encontrado muchos defectos y Tambien en donde se han corregido.

La eliminacion de defectos en diseño, podra ayudar a una mejor construccion durante la codificacion.

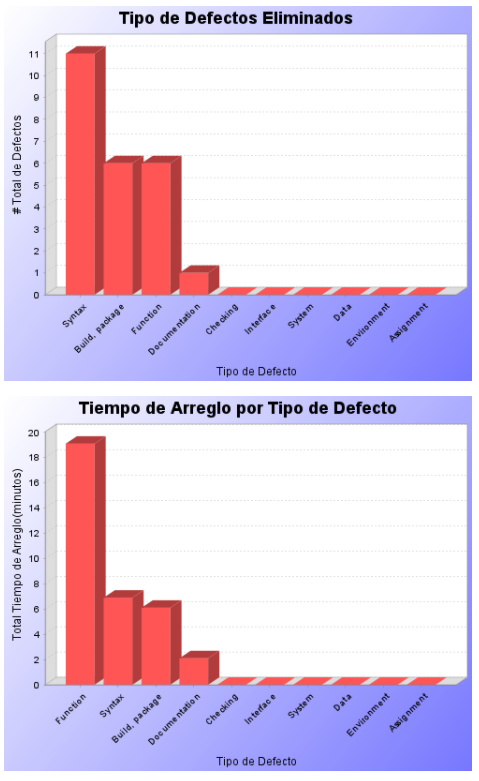
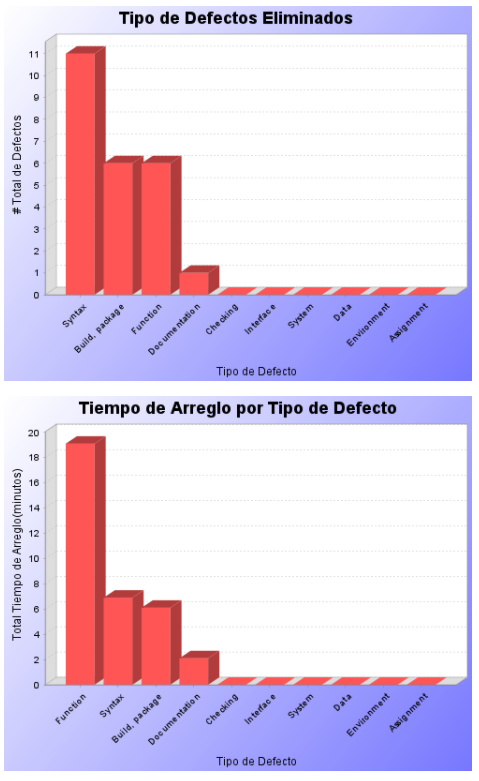
La eliminacion de defectos en codifocaion, podra ayudar a pasar no regresar a hacer modificaciones durante la compilacion y ayudaran a no retrasar la fase de pruebas.

• Analice el rendimiento frente al tamaño revisado por hora en sus revisiones de Código



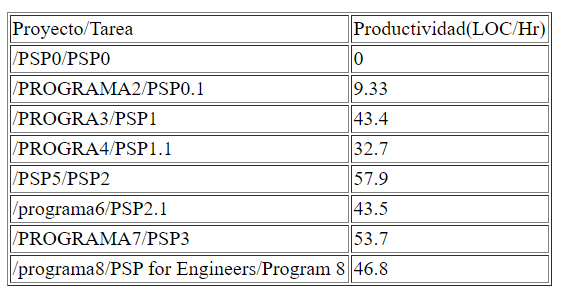
Se ha destinado mas tiempo para los errores de tipo function

• analizar el rendimiento de revisión de diseño versus revisiones de tamaño por hora



Se ha destinado mas tiempo para los errores de tipo function

• analizar el rendimiento frente al A / FR para los programas 5 a 8



Del 5 al 8, es una oscilacion entre subir y bajar el porcentaje de productividad

• describa sus áreas de mayor prioridad para la mejora de procesos personales y por qué. Asegúrese de resumir su desempeño actual, el desempeño futuro deseado y los objetivos de mejora. Describa cómo piensa alcanzar estos objetivos.

Un análisis de los requerimientos, y mejore planeación.

# PREGUNTAS

## Análisis de precisión de estimación de tamaño

* Compare la línea de base del informe provisional para determinar la precisión de la estimación del tamaño con los programas posteriores al informe. ¿Cuánto cambió su precisión de estimación de tamaño? ¿Por qué?

Se ha ido perfeccionando conforme se avanzo en el proyecto debido a la reutilización de

* ¿Con qué frecuencia el tamaño real de mi programa estuvo dentro de mi intervalo de predicción estadística del 70%?

Esto fue mas frecuente el los programas del 4 al 7

* ¿Tengo tendencia a agregar / perder objetos enteros?

No, se agregan solo las funciones

* ¿Tengo una tendencia a juzgar mal el tamaño relativo de los objetos?

Solo si se trata de algo que no he hecho con anterioridad

* ¿Necesito calcular los datos de rango de tamaño relativo utilizando mis datos de objetos históricos? ¿Yo puedo?

Si, para tener en cuenta los reutilizados y las base

* Según mis datos históricos de precisión de estimación de tamaño, ¿cuál es una meta de estimación de tamaño realista para mí?

Se acerca al real mientras mas se acerca trabajo en relación a lo previos

* ¿Cómo puedo cambiar mi proceso para alcanzar ese objetivo?

Tener un registro histórico de proyectos para poder usarlos de referencia

## Análisis de la precisión de la estimación del tiempo

* Compare la línea de base del informe provisional para determinar la precisión de la estimación del tiempo con los programas posteriores al informe. ¿Cuánto cambió su tiempo estimando la precisión? ¿Por qué?

Se he procurado no salir de los estándares esperados y la planeación, el porcentaje de error no excede el 26%

* ¿Con qué frecuencia estuvo mi tiempo de desarrollo real dentro de mi intervalo de predicción estadística del 70%?

Estuvo mas cerca cuando se ha trabajado en un programa similar, aun así, la estimación no baja del 26%de error

* ¿Mi productividad es estable? ¿Por qué o por qué no?

No, depende del tamaño del proyecto y las revisiones

* ¿Cómo puedo estabilizar mi productividad?

Con estimaciones de tiempo más aproximadas

* ¿En qué medida se ven afectadas mis estimaciones de tiempo por la precisión de mis estimaciones de tamaño? (¿Me ayudaría la regresión múltiple?)

Hay una relación entre el tiempo y el tamaño del programa tener un historial es algo a tener en cuenta para usar la regresión y estimar los próximos trabajos

* Según mis datos históricos de precisión de estimación de tiempo, ¿cuál es una meta realista de estimación de tiempo para mí?

Para una estimación habrá que tomar en cuenta la operación a realizar y la forma en como subdividirla en diferentes funciones mas pequeñas.

* ¿Cómo puedo cambiar mi proceso para alcanzar ese objetivo?

Creando un historial, para estimaciones futuras

## Análisis de defectos y rendimiento

* ¿Qué tipo de defectos inyecto durante el diseño y la codificación?

Mayor parte son de sintaxis

* ¿Qué tendencias son evidentes en los defectos por unidad de tamaño (por ejemplo, KLOC) que se encuentran en las revisiones, compilaciones y pruebas?

Al trabajar con una librería nueva donde se desconocen todas las funfiones y como de seclaran es el mayor problema, además de preveer revisiones de las incidencias.

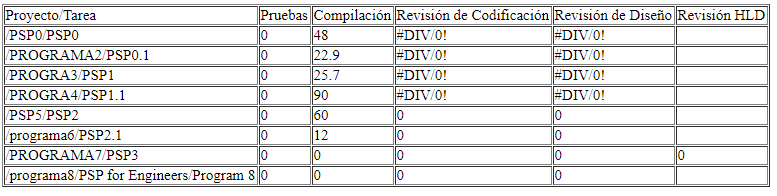
* ¿Qué tendencias son evidentes en el total de defectos por unidad de tamaño?

Se tiene mas a errores de sintaxis

* ¿Cómo se comparan mis tasas de eliminación de defectos (defectos eliminados / hora) para la revisión del diseño, la revisión del código, la compilación y la prueba?

Todos los errores se han detectado en el tiempo de compilación, el tiempo de eliminación ha sido bastante rápido

* ¿Cuáles son mis tasas de revisión (tamaño revisado / hora) para revisión de diseño y revisión de código?



La tasa de revisión han ido la baja mientras las se ha avanzado en el proyecto

* ¿Cuáles son mis ventajas de eliminación de defectos para la revisión del diseño, la revisión del código y la compilación frente a la prueba unitaria?

El desempeño de la fase de programación, ayuda a dividir las funciones en las clases bien definidas, además de que no retrasa la fase de compilación y pruebas

* ¿Existe alguna relación entre el rendimiento y la tasa de revisión (tamaño revisado / hora) para las revisiones de diseño y código?

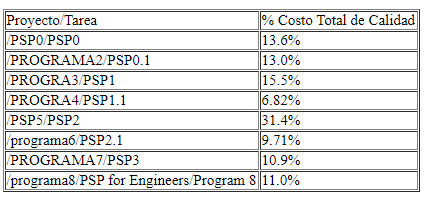
Mientras mas se destine a la revisión, mejor es el rendimiento en otras áreas.

* ¿Existe una relación entre el rendimiento y el A / FR para los programas 5 a 8?

El tiempo de revisión destinado a estos programas ha favorecido a una revisión mucho más exhaustiva no solamente en tiempo de codificación o diseño sino también como un tiempo extra en donde ha permitido revisar aquellos errores en los que con anterioridad se han incidido para no volver a reincidir teniendo una mayor calidad en el desarrollo del software.

## Análisis de calidad

* Compare la línea de base del informe provisional para la calidad en la prueba unitaria con los programas posteriores al informe. ¿Cuánto cambió la calidad de los programas que ingresaron a la prueba unitaria? ¿Por qué?



La calidad del software presenta un pico más alto en el programa 5 contando también con su pico más bajo en el programa 4, esto claro siempre puede deberse al hecho de que se reutilizan códigos en ciertas partes del proyecto.

* ¿Cómo puedo juzgar la calidad de mi producto final al principio de mi ciclo de desarrollo?

De manera personal se puede decir que dependiendo del entorno de trabajo los estándares de calidad pueden variar de empresa a empresa de manera personal puedo decir que mientras se puedan seguir los patrones de diseño que son soluciones óptimas ya documentadas puede ayudar mucho a incrementar la calidad.

* ¿Estoy encontrando defectos en las revisiones de diseño y código? ¿Por qué o por qué no?

De manera personal durante el tiempo de desarrollo y diseño no solamente se codificó o se diseñó vaya la redundancia sino que también se hicieron revisiones el tiempo de revisión de diseño y código es bastante corto comparado con el tiempo que se le ha dedicado a la realización de estas fases por lo que sea buscado también usar ese tiempo para revisiones después de construir cada sección

* ¿Cómo puedo hacer que mi proceso sea más efectivo y eficiente?

Un programa con requerimientos que no han sido debidamente definidos provocará un caos durante la codificación en donde veremos cambios y cambios y cambios en diferentes áreas algunas empresas optan por trabajar con algo que se denomina prototipo en donde se gestionan los requerimientos y la funcionalidad del proyecto una vez definidos estos se pueden comenzar a codificar lo que evita hacer cambios durante la programación.

* Según mis datos históricos, ¿cuáles son algunos de mis objetivos de calidad realistas?

Conforme se ha ido progresando y se ha reutilizado el código el número de fallas ha decretado Asimismo los costos de calidad de fallas se han ido de Clement ando teniendo el pico más alto en el programa 3 y su pico más bajo en el programa 7. Cuando se trabaja en un área nueva o se pretende realizar algún tipo de código nuevo aún si se reutiliza código esto pretende tener un fallo en la calidad ya que se pueden cometer nuevos errores.

* ¿Cómo puedo cambiar mi proceso para alcanzar esos objetivos?

Para cualquier proceso se debe documentar bien el tipo de librería usar y poder definir si hay funciones nuevas las cuales ayudarán a organizar mejor el proyecto tratando de utilizar funciones debidamente documentadas y evitar la aparición de errores.

# Conclusión.

El proceso personal de software es una herramienta que ayuda a los desarrolladores a poder entenderse a sí mismos y su proceso de programación personal. Es una herramienta, que del mismo modo nos ayuda a mejorar nuestra planeación, tratando de ser mucho más exactos a la hora de estimar el tiempo y el tamaño de un proyecto, esto basado en el historial que el desarrollador tiene. El proceso de desarrollo de software es una disciplina personal, en donde cada desarrollador puede codificar de una forma diferente siguiendo ciertos patrones o ciertas soluciones que le parecen más óptimas. Es por esto que siguiendo con esa disciplina para el desarrollo de software es indispensable minimizar la cantidad de errores cometidos y dedicar tiempo tanto a la planeación como el diseño, esto para poder gestionar mejor los diferentes métodos que existen en cada clase de la mejor manera. Esto ayudará a claro a no retrasar procesos y poder seguir adelante con las siguientes estados del ciclo de vida del software. No solo esto si no que ayudan al desarrollador a mejorar su rendimiento y planificación.