**ESTRATEGIA DE DESARROLLO**

1. **Desarrollar el producto en un proceso cíclico (2)**

Para este proyecto, aplicaremos TSP (Team Software Process). Usuremos la planeación basada en el tamaño de los proyectos históricos individuales.

El tamaño del programa suele tener un buen grado de correlación con el esfuerzo necesario para implantarlo y para efectos de la planeación, normalmente se estima primero el tamaño y, con base en él, el tiempo.

**Criterios de la estrategia:**

* El producto provee un subconjunto mínimo funcional del producto final
* El producto provee una base que puede ser fácilmente mejorada
* Todos los ciclos del producto son de alta calidad y pueden ser fácilmente probados.
* El diseño del producto tiene una estructura modular que permite a los miembros del equipo trabajar de manera independiente.

1. **MODELO CONCEPTUAL**

Insertar el modelo conceptual**ESTIMADO DE TAMAÑO PRELIMINAR**

Estimado de tamaño y tiempo preliminar a partir del diagrama conceptual del mundo del problema y de históricos de cada integrante del grupo.

**TSPi Strategy Form: Form STRAT**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name |  | | |  |  | Date | | 16/04/2014 | | |
| Team | EAIT | | | | | Instructor | | LUIS DANIEL BENAVIDES | | |
| Part/Level |  | | | | | Cycle | | 1 | | |
|  |  | | | | | |  |  | | |
|  |  | **Cycle LOC** | | | | | | **Cycle Hours** | | |
| **Ref** | **Functions** | 1 | 2 | | | 3 | | 1 | 2 | 3 |
| 1 | Diseñe el programa para funcionar inicialmente con Java | X | X | | |  | | X | X |  |
| 2 | Comparar un programa modificado con su versión anterior | 189 | X | | |  | | 10 | X |  |
| 3 | Identificar los LOC adicionadas y eliminadas en el programa | 133 | X | | |  | | 7,4 | X |  |
| 4 | Contar los LOC adicionadas y eliminadas en el programa. | X |  | | |  | | X |  |  |
| 5 | Contar el LOC total en el programa modificado | X |  | | |  | | X |  |  |
| 6 | Agregar una etiqueta a la línea, para cada línea modificada o eliminada para indicar el número del cambio | X |  | | |  | | X |  |  |
| **Totals** |  |  |  | | |  | |  |  |  |

1. **ALCANCE DEL PROYECTO**

Se definen los requerimientos a ser implementados en el cada ciclo.

* 1. Ciclo No. 1
* Diseñe el programa para funcionar inicialmente con Java
* Comparar un programa modificado con su versión anterior
* Identificar los LOC adicionadas y eliminadas en el programa
  1. Ciclo No. 2
* Contar los LOC adicionadas y eliminadas en el programa.
* Contar el LOC total en el programa modificado
* Agregar una etiqueta a la línea, para cada línea modificada o eliminada para indicar el número del cambio

1. Evaluar los riesgos
2. **Riesgos**

**PLAN DE CALIDAD**

El plan de calidad de este proyecto de desarrollo de software, tiene como fin orientar el proceso de desarrollo de cada requerimiento, aplicando buenas prácticas para asegurar y controlar la calidad en cada proceso del proyecto.

**Propósito**

Establecer un marco conceptual de trabajo que defina los lineamientos necesarios que debe seguir el equipo para mantener un trabajo organizado y orientado a la calidad de los procesos.

El plan de calidad propuesto pretende:

Mantener organizado todos los scripts TSP y registros de actividades desarrolladas en cada ciclo.

Tener información grupal que nos sirva para otros proyectos o ciclo 2.

**Definiciones, Acrónimos y Abreviaciones**

**Calidad**: Concepto basado en los entregables y procesos, tomando como guía los test y resultados del cliente. Es decir, entregar lo que realmente el cliente estaba esperando.

**TSP**: Team Software Process.

**Project Charter**: Acta de constitución del proyecto, donde se destaca el líder del proyecto.

**WBS**: Work Breakdown Structure, donde se genera los entregables del proyecto, esta es forma de controlar la calidad de los procesos, para dar como resultado la gestión de cada entregable.

**MANEJO DE ARCHIVOS**: Proceso para organizar los archivos con el objetivo de compararlos.

**CARGA DE ARCHIVOS**: Proceso para obtener los archivos a comparar.

**COMPARADOR**: Proceso que realiza la lógica de comparación.

**LOC**: Line of Code

* + 1. Referencias

PMBOK capítulo 8

* + 1. Gestión

Teniendo en cuenta que este proceso de desarrollo está fundamentado con bases en PSP (Personal Software Process) y ahora implementar la metodología TSP(Team Software Process), la estructura para la administración y organización del proyecto de software está definida de acuerdo a los roles ya establecidos, actividades, artefactos de inspección y el plan de calidad propuestos por TSP.

El equipo conformado para la ejecución del proyecto estará compuesto por 6 estudiantes de la Especialización en Construcción de Software de la Universidad de los Andes, entre los cuales se hará la distribución de los respectivos roles.

Este proceso se llevó a cabo la primera semana con el Lanzamiento del Proyecto.

* + 1. Estándares, Prácticas, Convenciones y Métricas

Estándares y Convenciones

Documentación: Como entregables utilizaremos los formatos TSP tomados de SEI (Software Engineering Institute) y elementos conformados como mínimo los siguientes elementos:

Nombre del entregable

Nombre del proyecto, incluido el ciclo

Portada del entregable

Gráficas en caso que haya, esta con su respectiva identificación

Nota: Intentar entregar documentos no muy extensos

Codificación: <http://www.oracle.com/technetwork/java/codeconv-150233.zip> mas adaptaciones para poder hacer cierto procesos de conteo de líneas.

1.1.5.2. Métricas

Las métricas de calidad utilizadas en el proyecto están basadas en el Plan Summary propuesto por TSPi, estas métricas son:

* Defectos por Página: Muestra el número de defectos removidos de cada página de documento elaborado en el proyecto.
* Defectos por KLOC: Defectos encontrados por cada 1000 líneas de código desarrolladas.
* A/FR: Es una proporción que se obtiene al dividir la cantidad de tiempo invertido en las revisiones e inspecciones entre el tiempo invertido en las pruebas del producto. Lo ideal sería que el tiempo de revisiones e inspecciones fuera el doble del tiempo de las pruebas.
* Proporción de Defectos: Es una proporción que se obtiene al dividir la cantidad de defectos encontrados en las revisiones de diseño del producto entre los defectos encontrados en las pruebas unitarias.
* Porcentaje de Revisiones: Indica el porcentaje de páginas, para el caso de documentos, o LOC para el caso de código, revisadas por hora.
* Porcentaje de Inspecciones: Indica el porcentaje de páginas, para el caso de documentos, o LOC para el caso de código, inspeccionadas por hora.
* Porcentaje de defectos inyectados: Indica el porcentaje de defectos inyectados por hora en los diferentes artefactos elaborados.
* Porcentaje de defectos removidos: Indica el porcentaje de defectos removidos por hora de los diferentes artefactos revisados e inspeccionados.
* Tiempo por fases: Compara el tiempo planeado contra el tiempo real en la elaboración de los artefactos definidos para las fases del proyecto.

**Inspecciones**

El propósito de esta sección es definir como parte de las estrategias trazadas por el equipo de trabajo un modelo practico para soportar el proceso de inspección de artefactos. A continuación se listan los lineamientos o reglas básicas de trabajo para atender estas actividades de la mejor forma:

Definir un formato estándar para el reporte de los defectos encontrados.

* Definir la lista de chequeo con los ítems a evaluar para cada artefacto construido.
* Cada integrante del equipo responsable de elaborar un artefacto, debe hacer la revisión del mismo y elaborar el reporte de defectos respectivo (lista de chequeo y análisis de defectos).
* Las inspecciones se distribuirán previa evaluación del equipo y bajo la coordinación del líder de grupo. Cada inspector será responsable de hacer el reporte de inspección (lista de chequeo y análisis de defectos) y estará en la capacidad de hacer las correcciones necesarias sobre el artefacto cuando estas son de forma; en caso de presentarse defectos de contenido, éstos deberán ser discutidos con el autor del artefacto y si es necesario hacer correcciones, el creador estará en la obligación de hacerlos lo más pronto posible sin afectar el plan de la semana, y notificar el resultado al moderador para llevar a cabo la actualización de los reportes de inspección.
* Todo el proceso debe estar supervisado por el moderador, persona responsable de elaborar semanalmente el reporte consolidado del proceso y discutir con el grupo durante las reuniones de seguimiento los factores positivos y negativos con el objetivo de madurar y fortalecer la mecánica del proceso.
* Cada artefacto debe contar con una sección de control de versiones, donde se registre la fecha, responsable y una breve descripción del cambio. Cuando los cambios se hagan como resultado de una revisión o inspección se debe colocar el identificador del reporte.

**TSPi QUALITY PLAN: FORM SUMQ**

Ver archivo en el repositorio GitHub (SUMQ) en el archivo cliclo1: doc/TSPi\_workbook.20041202.xls:

**ANÁLISIS DE DEFECTOS**

Los defectos encontrados tanto en las revisiones como en las inspecciones deben ser registrados por el responsable de la actividad en los formatos establecidos por TSP para tal fin. Dentro de los formatos de inspección se encuentran:

* SUMS: TSPi Size Summary
* SUMQ: Quality Plan Summary
* SUMP: TSPi Plan Summary Instructions:
* SUMT: TSPi Development Time Summary
* SUMDR: TSPi Defects Removed Summary
* SUMDI: TSPi Defects Injected Summary
* LOGD: TSPi Defect Recording Log
* LOGT: TSPi Time Recording Log
* INS: TSPi Inspection Report

**1.1.9. HERRAMIENTAS, TÉCNICAS Y METODOLOGÍAS**

TSP: Team Software Process es un proceso iterativo para el desarrollo de software que define roles, actividades, fases y ciclos. Presenta conceptos para la conformación eficiente de equipos de trabajo. Las fases definidas para el desarrollo del software son:

* Lanzamiento
* Estrategia
* Planeación
* Requerimientos
* Diseño
* Implementación
* Pruebas
* Postmortem

Una de las estrategias de TSP para conformar equipos de trabajo, es dividir las responsabilidades entre todos los integrantes del grupo. Por eso define los siguientes roles:

* Líder del Grupo
* Líder de Calidad
* Líder de Soporte
* Líder de Desarrollo
* Líder de Planeación

Se utilizarán la siguiente estandarización de herramientas para llevar a cabo el proyecto, tales como:

Se manejará un repositorio de herramientas para evitar la confusión de versiones al momento de descargar individualmente.

Sistema Operativo: Windows 7

Eclipse, versión del Skydrive

Java: Versión 7

Versionador: GitHub

Framework: Eclipse

Cada rol es responsable de tareas específicas del proyecto pero todos son responsables de la implementación del producto.

TSPiPlanSummary: Formato para llevar el seguimiento del proyecto con respecto a métricas de calidad.

Metrics: Herramienta para obtener métricas a nivel de código. Será utilizada para apoyar nuestros indicadores de gestión.

En cuanto a las herramientas de comunicación, a continuación se mencionan, describen y se asigna la responsabilidad de cada una.

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | **Responsabilidades** |
| Skype | Mensajería instantánea | Se usa para chat y llamadas grupales del equipo, las reuniones semanales se realiza por este medio cuando no se pueden realizar de manera presencial. La mayoría de los miembros del equipo usan este medio en la noche. |
| Google Hangouts | Mensajería instantánea | Alternativo a Skype. Se usa para chat y llamadas grupales del equipo, las reuniones semanales se realiza por este medio cuando no se pueden realizar de manera presencial. La mayoría de los miembros del equipo usan este medio en la noche. |
| Join.me | Compartir pantalla | Se usa para compartir la pantalla a los demás miembros del equipo durante las llamadas con el fin de que todos vean lo que se está haciendo. |
| Correo uniandes | Correo electrónico | Sirve para hacer difusión y comunicar al equipo novedades, inquietudes, inconvenientes, programar reuniones o consultas al equipo que no sean urgentes. No se debe usar para enviar los documentos de TSP ya que deben ser subidos a GIT. |
| Whatsapp | Mensajería instantenea | Se usa para mantener comunicado al equipo a cualquier hora del día e informar sobre situaciones urgentes o de atención inmediata. |
| Git | Sistemas de control de versiones | Repositorio de toda la documentación y archivos fuente de TSPi |