# Punto A

## Metodologías Tradicionales

### Rup

Su objetivo final es asegurar la producción de software de alta calidad, el cual cumpla con los requerimientos del cliente objetivo. Tomado de [1]

#### Fases

##### Fase Concepción

En esta se definen el rumbo del proyecto estableciendo los alcances de la mano del cliente, en base a lo anterior se asocian riesgos al mismo con el fin de elaborar un plan de acción que contemple los posibles casos secundarios y una arquitectura. Tomado de [1]

##### Fase Elaboración

En esta se bosqueja una posible solución la cual tiene que dar razón de casos de uso acordes a una arquitectura del sistema. Con el objetivo de evaluar la viabilidad de esta solución. Tomado de [1]

##### Fase Construcción

En esta se ejecuta la solución mas viable seleccionada en la anterior fase. Sin embargo, se pueden pulir detalles de funcionalidad o requerimientos del sistema de acuerdo con las reuniones con el cliente. Tomado de [1]

##### Fase Transición

En esta se comprueba el funcionamiento del software con pruebas y se le entrega al usuario final. Se pulen detalles de aceptación para posteriormente capacitar a los usuarios o dar soporte en caso de ser necesario. Tomado de [1]

### Microsoft Solution Framework (MSF)

Es una serie de modelos diseñados para proyectos de software. Tomado de [1]

#### Fases

##### Fase Visión y Alcances

Esta consiste en establecer un horizonte en común con el cliente para poder acordar un punto de meta con todos los involucrados en el proyecto. En base al punto de meta definir objetivos para alcanzarla evaluando los posibles riesgos del proyecto. Una vez establecido lo mencionado se asignan responsables/líderes de este. Tomado de [1]

##### Fase Planificación

Una vez la planificación está clara se preparan funcionalidades para dar una noción de solución estimando costos, tiempos de los hitos del plan. Tomado de [1]

##### Fase Desarrollo

Se materializan componen y arquitectura del sistema. Tomado de [1]

##### Fase Estabilización

Se realizan pruebas especializadas en la usabilidad y puesta en acción del producto en un entorno realista. En caso de existir errores se corregirán en la misma fase. Tomado de [1]

##### Fase Implantación

Se entrega la tecnología, realizando el proceso de instalación, Entrega de material soporte y operación. Con el objetivo de obtener la aprobación final del cliente. Tomado de [1]

## Metodologías Agiles

### Extreme Programming (XP)

Hace más énfasis en la adaptabilidad. Tomado de [1]

#### Fases

##### Fase Exploración

Se establece con el cliente de manera general lo deseado en paralelo el equipo de trabajo trabaja en familiarizarse con la tecnología a usar y posible arquitectura. Con el objetivo final de mostrar un prototipo que ayuda a definir de manera certera lo que se quiere. Tomado de [1]

##### Fase Planificación de Entrega

Se le asigna una prioridad a cada historia de usuario con el objetivo de tener una jerarquía de ejecución de estas. Con ello y las tecnologías a usar se estima el esfuerzo y el tiempo que demoraría cada una de ellas para darle fechas a las diferentes entregas del proyecto y presentárselo al cliente para conocer su punto de vista del plan y posibles sugerencias que pueden modificar lo establecido. Tomado de [1]

##### Fase Iteraciones

Esta consiste en lapsos de tiempo no mayores a tres semanas en los cuales se le dará respuesta a historias de usuario específicas que se verán reflejadas con tareas de programación a cargo de una persona, pero se desarrollarán en pareja. Tomado de [1]

##### Fase Producción

En esta fase se realizan pruebas a lo desarrollado en la primera versión y se corrigen errores. Puede que surjan nuevas características producto de sugerencias o propuestas las cuales deben quedar documentadas y programadas para la siguiente fase. Tomado de [1]

##### Fase Mantenimiento

Puesta en marcha de la primera versión la cual se debe hacer mantenimiento mientras se realizan las siguientes iteraciones. Tomado de [1]

##### Fase Muerte del Proyecto

Esta fase se define como un hito donde todas las historias de usuario se han completado y se encuentran en funcionamiento en la ultima versión. Posterior a ello se realiza entra de documento final del sistema. Tomado de [1]

### Metodología Scrum

#### Fases

##### Fase Planificación Sprint

En la reunión se establece el producto a entregar en esa iteración y como se realizará. Tomado de [1]

##### Fase Diario

Este se realiza cada día y en esta se pone al día todas actividades realizadas o por realizar para coordinar todo el equipo en un plan para ese día. Tomado de [1]

##### Fase Desarrollo Sprint

Hace referencia al trabajo que aporte al objetivo final establecido en la planeación, el cual no debe modificar la idea planteada originalmente. Tomado de [1]

##### Fase Revisión Sprint

Este hace referencia a una revisión final del trabajo del sprint para ser incrementado al producto. Tomado de [1]

##### Fase Retrospectiva Sprint

En esta se realiza una auto evaluación para mejorar para el siguiente sprint. Tomado de [1]

## Metodologías Seguras

### Correctness by Construction (CbyC)

#### Fases

##### Fase Requerimientos

En esta fase se especifica el propósito del producto en conjunto con los requerimientos los cuales tienen que ir soportados por diagramas de clase y definición de operativas. Además, deben ir asegurados con una trazabilidad y requerimientos secundarios de seguridad a las amenazas de pendiendo del tipo de requerimiento principal. Tomado de [2]

##### Fase Diseño de Alto Nivel

Se diseña la estructura del sistema y la conectividad de los diferentes componentes que la conforman. Además de reforzar con requerimientos no funcionales que brinden protección y seguridad a este. Tomado de [2]

##### Fase Especificación del Software

Se documenta las características de la interfaz de usuario para posteriormente desarrollar un prototipo y comprobar su validez. Tomado de [2]

##### Fase Diseño Detalles

En base al prototipo se definen módulos, procesos y funcionalidades a desarrollar. Tomado de [2]

##### Fase Especificación de los módulos

Se estudia el flujo de información por cada modulo y como este interactúa en el sistema completo. Tomado de [2]

##### Fase Codificación

Se programa con lenguaje especializado para dent5ro de la misma fase realizar pruebas de análisis estático para eliminar errores. Tomado de [2]

##### Fase Especificaciones de las pruebas

Se realizan pruebas rigurosas de valores límites, pruebas de comportamiento y de requerimientos no funcionales. Tomado de [2]

##### Fase Construcción de software

Se realiza entrega de una primera versión con el esqueleto completo del sistema que incluye interfaces, comunicación y funcionalidad limitada mientras se sigue trabajando en la siguiente iteración. Tomado de [2]

### Security Development Lifecycle (SDL)

#### Fases

##### Fase planes

Se realiza un estudio de la seguridad básica de los diferentes componentes a utilizar en el proyecto y tipo de software a desarrollar. Tomado de [2]

##### Fase Requerimiento

Se establece requenetos de seguridad asociados a los requerimientos del sistema. Posteriormente se establece un límite de calidad y errores. En base al tipo de riesgo de seguridad y privacidad que apliquen a los requerimientos Tomado de [2]

##### Fase Diseño

Se establece requerimientos de diseño a los cuales se le asocian ataques posibles por medio del modelado de amenazas que conlleve los requerimientos. Tomado de [2]

##### Fase Implementación

Se desarrolla en lenguaje específico para posteriormente hacer pruebas de análisis estático. Hay que mencionar que esta prohibido un grupo de funciones no seguras. Tomado de [2]

##### Fase Verificación

Se implementa análisis dinámico al producto(prototipo) en conjunto con fuzz testing que complementa la revisión de ataques. Tomado de [2]

##### Fase Lanzamiento

Una vez realizado la verificación se le realiza una revisión final con la cual se aprueba para ser archivada con los planes de respuesta de posibles incidentes. Tomado de [2]

##### Fase Respuesta

En caso de presentarse algún incidente se pondrá en marcha el protocolo consignado en el plan de respuesta a incidentes. Tomado de [2]

## Metodologías Web

### WSDM (WEB DESIGN METHOD)

#### Fases

##### Fase Modelado de Usuario

Se identifica los usuarios y la información que manejan cada uno en el sitio web. Tomado de [3]

##### Fase Diseño Conceptual

Se organiza en un modelo conceptual la información clasificada por usuario. Con el fin de modelar el objeto, crear un diagrama e-r con y la navegación. Tomado de [3]

##### Fase Diseño Implementación

Se crea un prototipo en base a los requerimientos del usuario y los diagramas anterior4es. Con el objetivo de que sea agradable, eficiente y seguro. Tomado de [3]

##### Fase Implementación

Se selecciona un entorno de desarrollo y se construye la arquitectura para continuar con la codificación. Con el objetivo de entregar una primera versión que permita retroalimentar el proceso. Tomado de [3]

### SOHDM (SCENARIO-BASED OBJECT-ORIENTED HYPERMEDIA DESIGN METHODOLOGY)

#### Fases

##### Fase Análisis del dominio

Se establece los límites de la aplicación y se diagramara por medio de un Diagrama de Flujo. Tomado de [3]

##### Fase Modelo de Objetos

Se desarrolla el modelo e objetos relacionado con los usuarios, los cuales son descritos en el documento de desarrollo con sus correspondientes atributos. Tomado de [3]

##### Fase Diseño Vistas

Se diseña la navegación por medio de unidades de navegación que describen los elementos de las vistas y la información que manejan. Tomado de [3]

##### Fase Diseño Navegación

Se desarrolla la navegación gráfica y la forma en que interactúa las vistas con sus diferentes elementos. Tomado de [3]

##### Fase Implementación

Se desarrolla una interfaz, incorporando la lógica de la información que se mueve dentro de ella correspondiente al esquema de datos. Tomado de [3]

##### Fase Construcción

Se desarrollo de la aplicación final para su entrega al cliente con su correspondiente documentación, no hay retroalimentación. Tomado de [3]

# Punto B

|  |  |  |
| --- | --- | --- |
| Metodología | Ventajas | Desventajas |
| Rup--- | Forma diciplinada asignar tareas/responsabilidades  Mejora productividad del equipo  Estime tareas y horarios del plan  Levantamiento Exhaustivo de requerimientos  Tomado de [4] | Un grupo grande de programadores  Mas apropiado Proyectos grandes  Los miembros deben ser expertos en el campo  Proceso de desarrollo complejo  Evaluación de riesgo es compleja  Tomado de [4] |
| Microsoft Solution Framework-- | Alta adaptabilidad a diferentes proyectos  Tiempos aproximados y estudio de riesgos  Entregas parciales en cronograma  Pruebas enfatizan en uso y operación en entorno real  Varias iteraciones para entrega  Tomado de [1] | Estudios bajos en seguridad  Alcances y limitaciones fijas.  Flexibilidad a errores por fuera del del análisis baja.  Participación del cliente en el desarrollo baja, solo participa en la entrega parcial de producto.  Bajo nivel de recodificación.  Tomado de [1] |
| Extreme Programming-- | Planificación más transparente(fechas entrega)  Menor taza de errores.  Retroalimentación de usuario  La presión se distribuye a lo largo del proyecto  Preparado para el cambio implica menos costos.  Tomado de [4] | Solo proyectos a corto plazo  Desarrollo riesgoso y difícil de controlar.  Se rediseña todo el tiempo  Limitaciones de la mano con el cliente  No deja experiencia para futuros proyectos debido a su débil documentación  No requiere de un nivel alto de programadores  Tomado de [4] |
| Scrum-- | Tomar proyectos con requerimientos incompletos  Permite desarrollo, testeo y corrección rápida  Se identifican avances y problemas  Mucho feedback del cliente  Felicita las entregas de calidad a tiempo  Tomado de [4] | En ausencia de fecha final se estropea el ciclo  Tareas no limitas causan costos y tiempos  Solo funciona bien con equipos pequeños/medianos  Se requiero parte del equipo experimentado  Un abandono por parte del equipo acarrea problemas  Tomado de [4] |
| Correctness by Construction | Proceso extensivo de requerimientos  Proceso extenso de análisis de requerimientos  Proceso detallado de aprobación por el cliente  Pruebas especializadas en seguridad.  Si retroalimentación y posibilidad de recodificación  Tomado de [2] | Proyectos largos  Lenguaje Especifico  Equipos grandes  Requiere equipo especializado en el tema  Tomado de [2] |
| Security Development Lifecycle-- | Proceso extensivo de requerimientos  Proceso extenso de análisis de requerimientos  Proceso detallado de aprobación por el cliente  Pruebas especializadas en seguridad.  Tomado de [2] | * No retroalimentación * Proyectos largos * Poca flexibilidad para cambios * Lenguaje especifico * Equipos grandes * Requiere equipo especializado en el tema   Tomado de [2] |
| WSDM | Enfoca en los usuarios del sistema  Parámetros de diseño en base a usuarios  La retroalimentación es posible al igual que la recodificación.  Tomado de [3] | Lapso corto tiempo  Aplicable a apelaciones web  Incompleta gestión de proyectos  Se enfoca en sitios web estáticos (brindan información)  No se centra en los datos  Tomado de [3] |
| SOHDM | Enfoca las necesidades del sistema  Se orienta a objetos e escenarios donde interactúa el usuario y el sistema  Contempla las entidades externas para analizar su comunicación con el sistema  Se estudia el flujo de actividad  Tomado de [3] | No permite recodificación.  Lapsos cortos/medianos  Bajo nivel de pruebas al producto final  Tomado de [3] |

# Bibliografía

[1]Metodologías Tradicionales Vs. Metodologías Ágiles Publicado en febrero 2007. Roberth Figueroa Díaz

[2]Metodologías para desarrollar software seguro Carlos Joaquín Brito Abundis Universidad Autónoma de Zacatecas

[3] COMPARACIÓN DE METODOLOGÍAS EN APLICACIONES WEB/COMPARISON OF METHODOLOGIES IN WEB APPLICATIONS/Jimmy Rolando Molina Ríos, Mariuxi Paola Zea Ordóñez, María José Contento Segarra, Fabricio Gustavo García Zerda

[4]Estudio Comparativo de Metodologías de Desarrolló de Software- José Gaspar Artego Camacho