**FASE 1: RECOPILACIÓN DE REQUISITOS**

**Descripción Detallada**

En esta fase, el objetivo es comprender a fondo las necesidades y expectativas del cliente. Se recibe información preliminar del cliente, como el contexto del problema y sus expectativas de solución. Los participantes principales son el Analista o Gerente del proyecto y el Cliente. El Analista o Gerente del Proyecto, lidera la recopilación de información a través de entrevistas, encuestas y talleres con el cliente y los usuarios clave. También se asegura de que se definan metas claras para evitar ambigüedades en los requisitos.

Se recopilan tanto los requisitos funcionales (qué debe y necesita hacer el sistema) como los no funcionales (diseño, rendimiento, seguridad, usabilidad), que luego son documentados y validados por el equipo. El equipo realiza un análisis para asegurarse de que los requisitos sean claros, viables y alineados con los objetivos del proyecto.

En esta fase, es crucial identificar riesgos como:

* **Falta de claridad en los requisitos:** Se debe implementar una metodología clara de comunicación para que el cliente pueda expresar sus necesidades de manera efectiva.
* **Cambios frecuentes de requisitos:** Se puede utilizar técnicas de priorización para minimizar el impacto de los cambios.
* **El Gerente del Proyecto o el Analista deben llevar un registro de riesgos identificados y actualizarlo periódicamente.**

**Entregables:**

* **Documento de requisitos del Sistema:** En este documento se deberá contener el registro de la fecha en que se realizó, al igual que se plasmaran elementos como los requisitos funcionales y no funcionales, el propósito del proyecto, alcance, además de algunos aspectos en particular que tenga el programa.
* **Documento de Validación del Software:** En este entregable deberá contener la fecha de realización, se deberán describir los requisitos para que el cliente y el gerente de proyecto puedan aprobarlos y seguir con el desarrollo del proyecto además que se deberán plasmar los acuerdos que generaron en sus juntas

**Roles:**

* **Analista o Gerente del proyecto:** Lidera la recopilación de requisitos, supervisa el proceso y gestiona los riegos.
* **Cliente:** Proporciona el conocimiento del dominio y las expectativas.

**FASE 2: ESPECIFICACIÓN DEL SOFTWARE**

**Descripción Detallada**

En esta fase, los requisitos recopilados se transforman en una especificación técnica detallada del sistema. El Arquitecto de Software o en el caso de que el proyecto sea más simple un Desarrollador recibe el Documento de Requisitos y diseña la arquitectura del sistema, definiendo cómo se estructurará y cómo interactuarán sus componentes. Esto incluye tanto el diseño general del sistema (arquitectura de software) como los detalles de implementación (funcionalidades clave, manejo de datos, y componentes específicos).

Se desarrollan diagramas de flujo, diagramas de clases, y prototipos para visualizar cómo el sistema interactuará con el usuario y cómo manejará los datos. El equipo de desarrollo y el Documentalista contribuyen documentando las decisiones y asegurándose de que la arquitectura cumpla con las expectativas del cliente.

Se debe identificar riesgos como:

**Errores de diseño o mala comunicación:** Para mitigar este riesgo, el Gerente del Proyecto o Analista debe asegurar revisiones frecuentes entre los equipos de diseño y desarrollo, además de la validación continua del cliente para prevenir malentendidos en la especificación.

**Dependencias tecnológicas no comprobadas:** Se deben hacer análisis de viabilidad y pruebas de conceptos para evaluar tecnologías o componentes complejos.

**El Gerente del Proyecto o Analista debe monitorear estos riesgos, asignar recursos y ajustar el cronograma cuando sea necesario**.

**Entregables:**

* **Diagramas de Arquitectura:** Se deberá especificar el tipo de diagrama, el propósito de este, así como que herramientas se utilizó para su realización, el formato en el que se encuentra y la ruta de este, así como unas observaciones
* **Prototipos Funcionales**: Llevar un control del desarrollo de interfaces preliminares de sistema registrando elementos de sus características, el fin de prototipo, así como su ruta en el que se encuentra alojada
* **Modelo:** Se deberá especificar el tipo de modelo desarrollado (por ejemplo, modelo de datos, modelo de casos de uso, modelo de procesos, etc.), el propósito de este dentro del sistema o proyecto, las herramientas empleadas para su construcción (por ejemplo, ERwin, UML, etc.)

**Roles:**

* **Arquitecto de Software o desarrollador:** Diseña la estructura técnica del sistema.
* **Desarrolladores:** Contribuyen con feedback sobre la viabilidad técnica.
* **Gerente del Proyecto o Analista:** Supervisa la coherencia entre el diseño y los objetivos
* **Documentalista:** El documentalista es responsable de capturar, organizar y mantener toda la documentación relacionada con la especificación técnica del sistema.

**FASE 3: DESARROLLO**

**Descripción Detallada**

El desarrollo es la fase en la que se convierte el diseño técnico en un código funcional. El Equipo de desarrolladores reciben el Documento de Especificación Técnica y comienza a codificar siguiendo las especificaciones y estándares de calidad. La implementación se organiza en ciclos de trabajo, donde se desarrollan características individuales. Cada ciclo culmina con la integración y revisión del código por parte del equipo de desarrolladores o Arquitecto de software, asegurando que las funcionalidades individuales encajen correctamente con el sistema en su totalidad.

Además, se crean pruebas unitarias que validan cada componente de software, asegurando que funcionen conforme a lo especificado en el diseño. Es fundamental mantener una integración continua para detectar y corregir errores a tiempo.

Analizar posibles riesgos como:

* **Falta de alineación entre el código y el diseño:** Implementar revisiones de código y revisiones arquitectónicas periódicas para asegurar la coherencia entre lo que se está construyendo y lo que se diseñó.
* **Errores de integración o bugs no detectados:** Se recomienda utilizar un enfoque de integración continua (CI/CD) para facilitar la detección temprana de problemas.
* **El Gerente del Proyecto o Analista debe monitorizar el progreso y los riesgos que puedan afectar la calidad del código.**

**Entregables:**

* **Código Fuente Funcional**: En este entregable deberá contener datos como la versión, descripción de los componentes, funcionalidad, dependencias, lenguaje en el que se programó y el lugar donde este alojado
* **Pruebas Unitarias:** En este apartado almacenara la información de cada prueba, datos como el elemento que se probó con su descripción, herramientas utilizadas, así como los resultados obtenidos de cada prueba y los resultados que esperaba
* **Documentación Técnica**: proporciona una descripción detallada de cómo funciona un sistema, cubriendo aspectos de su arquitectura, diseño, y procesos.

**Roles:**

* **Desarrolladores:** Implementan las funcionalidades.
* **Gerente del Proyecto o Analista:** Supervisa el progreso y los riesgos.
* **Arquitecto de Software o desarrollador:** Asegura que el código siga la arquitectura especificada.

**FASE 4: VERIFICACIÓN Y VALIDACIÓN DEL SOFTWARE**

**Descripción Detallada**

Esta fase asegura que el software desarrollado cumpla con los requisitos y que funcione correctamente en su entorno operativo. El equipo de Pruebas que se conforma por el ingeniero de pruebas/ Tester y Analista de Calidad, recibe el Código Fuente y lo somete a pruebas exhaustivas que incluyen pruebas unitarias, funcionales, de integración y de aceptación. Estas pruebas se basan en los casos de uso y requisitos definidos en la fase inicial.

Además de las pruebas automáticas, el Cliente o los Usuarios Finales validan el software durante las pruebas de aceptación, asegurando que cumpla con sus expectativas. Esta fase es crítica para asegurar que el sistema esté libre de errores antes de su despliegue.

Identificar riesgos como:

* **Riesgos de fallas críticas:** La implementación de pruebas exhaustivas y de un proceso de pruebas automatizado ayudará a detectar errores de manera temprana.
* **Requisitos no cumplidos:** Se pueden realizar revisiones periódicas con el cliente para asegurar que todo lo solicitado esté cubierto.
* **El Gerente del Proyecto o Analista debe gestionar las correcciones de errores con prioridad y planificar mejoras según la criticidad.**

**Entregables:**

* **Reportes de Pruebas:** lo que se busca que existe un control con la aparición de errores que vayan surgiendo alrededor de esta prueba y lo que se busca que se documente con lo datos más relevante que surjan durante la prueba
* **Documento de Validación de Cumplimiento**: Confirmación de que el software cumple con los requisitos establecidos registrando características de estos como su clasificación, alguna descripción, su prioridad y resultado relacionados con las pruebas relacionales además de incluir una validación tanto del gerente y el cliente

**Roles:**

* **Testers/Ingenieros de Pruebas:** Ejecutan las pruebas.
* **Desarrolladores:** Corrigen los errores.
* **Analista de calidad:** Responsable de asegurar que el software cumpla con los estándares de calidad definidos antes de su lanzamiento
* **Gerente del proyecto o Analista:** Gestiona el proceso de validación y pruebas.

**FASE 5: DESPLIEGUE**

**Descripción Detallada**

Una vez que el software ha sido validado, es hora de ponerlo en producción. El equipo de Ingenieros de DevOps instala y configura el sistema en el entorno del cliente, asegurando que esté correctamente integrado con otros sistemas. También se realiza una migración de datos, si es necesario, y el Equipo de Soporte Técnico prepara la infraestructura para el uso final del software. Se capacita a los usuarios finales para que puedan utilizar el sistema de manera efectiva.

El Gerente del Proyecto o Analista coordina el plan de despliegue y asegura que todos los pasos críticos sean seguidos para evitar interrupciones.

Administración del Riesgo:

* **Fallas en el entorno de producción:** Realizar pruebas piloto para garantizar que el software funcione adecuadamente antes de su implementación final.
* **Problemas de adopción por parte de los usuarios:** Implementar un plan de capacitación estructurado para los usuarios finales y ofrecer soporte post-despliegue.

**Entregables:**

* **Software en Producción:** Listo para ser utilizado por los usuarios finales y se entrega junto con La guía de Capacitación de Usuarios.
* **Capacitación de Usuarios:** Documento donde se especifica con detalle como usar correctamente el software entregado. Contiene una portada donde aparece el nombre del proyecto, posteriormente, la siguiente página, contiene una tabla de contenidos (índice), cada una (página) con número de página.

**Roles:**

* **Especialista en Seguridad:** Asegura que, durante el despliegue, el software cumple con los estándares de seguridad, mitigando riesgos como ataques o vulnerabilidades en el entorno de producción.
* **Ingenieros de DevOps (externos):** Despliegan el sistema.
* **Soporte Técnico (externos):** Ofrecen asistencia durante el despliegue.
* **Gerente del Proyecto o Analista:** Coordina las actividades de despliegue.

**FASE 6: EVOLUCIÓN DEL SOFTWARE**

**Descripción Detallada**

El ciclo de vida del software no termina con el despliegue. El software debe seguir evolucionando para adaptarse a nuevas necesidades del cliente, cambios en el entorno operativo o mejoras tecnológicas. El equipo de Desarrolladores recibe reportes de errores y solicitudes de mejora del Cliente y del Equipo de Soporte. Estas solicitudes son evaluadas y priorizadas para su implementación.

La evolución incluye la adición de nuevas funcionalidades, corrección de errores reportados en el uso real del sistema y la optimización de su rendimiento. Además, el Equipo de Soporte juega un papel crucial, ya que proporciona asistencia continua al cliente y realiza el mantenimiento preventivo y correctivo del sistema. Se pueden emitir nuevas versiones del software con actualizaciones que mejoran la seguridad o la usabilidad.

Administración del Riesgo:

* **Inestabilidad del software después de actualizaciones:** Antes de implementar actualizaciones, es crucial realizar pruebas en un entorno controlado. Esto incluye pruebas de regresión para asegurarse de que las nuevas funciones no afecten las existentes.
* **Falta de alineación con las necesidades cambiantes del cliente:** Se debe mantener una comunicación fluida con el cliente para entender nuevas expectativas y ajustar las prioridades de evolución.
* **El Gerente del Proyecto o Analista debe planificar las iteraciones de evolución y coordinar con el cliente para asegurar que el software siga siendo relevante.**

**Entregables:**

* **Actualizaciones del Software:** Nuevas versiones del sistema que corrigen errores o añaden funcionalidades.
* **Parches de Seguridad:** Se entregará un archivo en cual se describirá elementos del parche tales como una descripción corta, en que elemento se modificó y información del parche como la que versión del programa y notas del parche
* **Documentación Actualizada:** Refleja las nuevas funcionalidades y mejoras implementadas. Se entrega en un documento como el de Capacitación de Usuarios, esto quiere decir que contendrá la misma información, a diferencia de que, al final de la Tabla de Contenidos se le agregará un anexo y el número correspondiente, el cual se nombrará “Nuevas funcionalidades y Modificaciones [Nombre del proyecto] [Versión del proyecto]. También se le agregará la información correspondiente respecto a las nuevas actualizaciones al final del documento, siguiendo así, el orden de la Tabla de Contenidos.

**Roles:**

* **Desarrolladores:** Implementan nuevas funcionalidades y actualizaciones.
* **Equipo de Soporte(externos):** Mantiene y optimiza el sistema, y gestiona incidentes.
* **Gerente del Proyecto o Analista:** Coordina las actividades de evolución y gestiona los riesgos asociados.