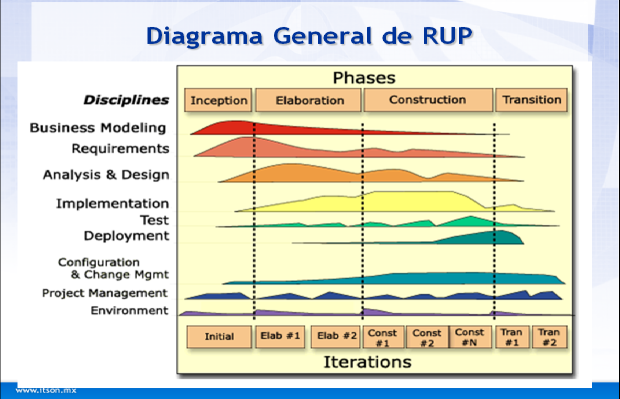
**METODOLOGÍA SCRUM, RUP**

|  |  |  |
| --- | --- | --- |
| METODOLOGÍA | SCRUM | RUP |
| CONCEPTO | Conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Es un ciclo completo | Es un proceso de ingeniera de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización desarrollo |
| VENTAJAS | * Conocimiento necesario para lograr un objetivo. * Involucra desde un principio y se da un rol a todos. * Entregables en tiempo y forma. | * Es modelado guiado por caso de uso. * Configuración y control de cambios. * Mayor documentación. * Es cerrado en arquitectura guido por riesgos. * Verifica la calidad de software. |
| DESVENTAJAS | * Los miembros del equipo se saltan pasos importantes en el camino para llegar al Sprint final. * Demasiadas reuniones para poco avance. | * Los cambios son en una fase * Proyectos grandes |
| ROLES | * Trabajo realizado por el equipo de especialistas. * Visible, transparente por el equipo de especialista. * Responsabilidad. | * Analistas * Desarrolladores * Gestores * Apoyo y especialistas * Coordinación de revisiones |
| CARACTERÍSTICAS | * Fundamentada en principios. * Reduce el costo del cambio en todas las etapas. * Equipo con formación elevada. | * Se abarca prácticas de gestión sin entrar en prácticas de desarrollo * Delega completamente en el equipo la responsabilidad. |

**METODOLOGÍA RUP**

El ciclo de vida de RUP está desarrollada de manera secuencial (en espiral), y al concluir cada fase se verifica con una evaluación si cumple con los objetivos de la misma. Estas fases se dividen en 4, estas a su vez se efectúan en iteraciones según el proyecto que se encuentre desarrollando. Estas fases son:

* Inicio: Consiste en especificar y delimitar los objetivos del proyecto y su alcance con las partes interesadas, describir los riesgos relacionados al mismo y asegurar que el proyecto sea viable, dando un enfoque general de la arquitectura de software.
* Elaboración: Se establece la arquitectura base del sistema para brindar una plataforma segura, se definen los casos de uso escogidos para ello, teniendo en consideración los aspectos de mayor relevancia y se realiza una evaluación de riesgo.
* Construcción: La finalidad de esta fase es culminar con la funcionalidad del sistema, esclareciendo las dudas que puedan existir, verificando que se cumplan los requerimientos pendientes, todo en función de la arquitectura base definida previamente.
* Transición o cierre: El propósito de esta fase es garantizar la disponibilidad del software para los usuarios finales, hacer cambios menos solicitados por el usuario, depurar el producto en relación a los errores encontrados en las pruebas, brindar la capacitación concerniente a los usuarios y verificar que el producto final cumpla con los requerimientos entregados por las partes interesadas.



RUP también permite estimar tareas y horarios del plan de trabajo, midiendo la velocidad de las iteraciones que se enfocan a las estimaciones, estas iteraciones se enfocan estrictamente a la arquitectura del software, por ello, esta metodología es actualmente una de las más utilizadas para el diseño, análisis e implementación de aplicaciones y sistemas, sus 9 disciplinas son:

* Modelado de negocio: esta permite medir el impacto del cambio organizacional, de forma que se pueda entender los problemas que presenta la empresa, y que desea solucionar identificando las mejoras potenciales.
* Requerimientos: Prevé una base para el planeamiento de los contenidos técnicos de las iteraciones.
* Análisis y diseño: La tarea de este requerimiento permitirá adoptar un diseño que corresponderá al ambiente de implementación y de esta forma desarrollar una arquitectura robusta para un desempeño optimo del sistema.
* Implementación: se probarán los componentes creados como unidades (archivos fuente, ejecutables, etc.).
* Pruebas: en esta fase se realizarán las validaciones respectivas del software para comprobar si trabaja como fue diseñado.
* Transición: Esta disciplina detalla las actividades que se asocian al aseguramiento de la disponibilidad del producto desarrollado hacia el usuario final.
* Administración y configuración: Esta consiste en controlar las versiones desarrolladas manteniendo la integridad de los productos que se incluyen en el proyecto.
* Administración de proyectos: En esta disciplina se proveen o diseñan guías prácticas para la planeación, ejecución y monitores de los proyectos a desarrollar.
* Ambiente: se provee a la organización de desarrollo del software el ambiente adecuado para el soporte que se brinde al equipo de desarrollo. (p. 26)