

Metodología para el desarrollo de proyectos

Selección del modelo de desarrollo de software

Antecedentes

Sin ser exhaustivo, sino simplemente genérico se revisaron los siguientes modelos de desarrollo de software (también citados como paradigmas): **Cascada**, **Modelo en V**, RAD (Rapid Application Development), Incremental, **Prototipado** (o por prototipos), **Espiral**, **XP** (**Extremme Programming**), **SCRUM**, ASD (Adaptive Software Development) y CM (Crystal Methods).

Los criterios identificados para seleccionar estos modelos son:

1. Disponibilidad de recursos
2. Complejidad del proyecto
3. Entendimientos de requerimientos
4. Conocimiento del dominio del problema
5. Manejo de la perspectiva de riesgos
6. Tiempos de desarrollo
7. Costo de los proyectos
8. Calidad del software
9. Documentación

Disponibilidad de recursos

Este criterio hace referencia los recursos equipo y materiales calificados en el momento indicado y durante el tiempo requerido. En este criterio, los valores posibles son:

- ▶ **Todos** (Que la organización ha dispuesto o dispondrá todos los recursos necesarios para la ejecución del proyecto)
- ▶ **Algunos** (Que la organización ha dispuesto o dispondrá solo algunos de los recursos necesarios para la ejecución del proyecto).

Complejidad del proyecto

Este criterio hace referencia al tamaño del sistema y la complejidad del mismo, donde según la complejidad del proyecto aplica o no aplica una metodología específica. Los cuantificadores se han obtenido en base a dos parámetros: (a) en base a un cálculo de complejidad como Puntos de Función o Puntos de Casos de Uso u otros; y (b) en base a la experiencia de las personas y de la madurez en los procesos empleados por la organización). En este criterio, los cuantificadores posibles son:

- ▶ **Alta** (Proyectos de Complejidad Alta)
- ▶ **Media** (Maneja proyectos de complejidad media)
- ▶ **Baja** (Es recomendable para proyectos de complejidad baja)

Entendimientos de requerimientos

Este criterio hace referencia a tener claro los requerimientos del sistema en la etapa inicial del proyecto por parte del analista o diseñador UML. Los cuantificadores se han obtenido en base a la conjugación de dos cosas: (a) la experiencia del observador; y, (b) el manejo que tenga de las herramientas de descripción. En este criterio, los cuantificadores posibles son:

- ▶ **Específico** (Es necesario que el analista o diseñador tengan muy claro todos los requerimientos de forma detallada y específica al inicio del proyecto)
- ▶ **Bajo** (No es necesario por parte del analista o diseñador el entendimiento de todos los requerimientos al inicio del proyecto).

Conocimiento del dominio del problema

El Conocimiento del dominio del problema se refiere al conocimiento del problema de negocio y su entorno que posee el analista o diseñador antes de revisar una situación o fenómeno. Los cuantificadores se han obtenido en base a la conjugación de dos cosas: (a) el grado de capacitación del observador en el tema; y, (b) la experiencia del observador en casos iguales o similares y/o en proyectos similares o parecidos). En este criterio, los cuantificadores posibles son:

- ▶ **Alto** (cuyo valor se consigue cuando el observador tiene un completo dominio de todos los factores y procesos que participan del fenómeno en estudio),
- ▶ **Regular** (cuyo valor se consigue cuando el observador tiene algún dominio de los factores y procesos que participan del fenómeno de estudio); y,
- ▶ **Pobre** (cuyo valor se consigue cuando el observador tiene poco dominio de todos los factores y procesos que participan del fenómeno en estudio).

Manejo de las perspectivas de riesgos

Este criterio hace referencia a tener en cuenta la definición de riesgos y perspectivas de los mismos, en alguna de las metodologías tienen en cuenta este criterio. Los cuantificadores se han obtenido en base a la consideración de dos aspectos: (a) en base a instrumentos cuantitativos de medición de riesgos, y (b) en base a la experiencia de las personas y la madurez de la organización en sus procesos de desarrollo. En este criterio, los cuantificadores posibles son:

- ▶ **SI** (cuyo valor se consigue cuando el paradigma contempla entre sus fases la definición de perspectivas de riesgo)
- ▶ **NO** (cuyo valor se consigue cuando el paradigma no contempla entre sus etapas la definición de manejo de perspectivas de riesgo)

Tiempos de desarrollo

Este criterio hace referencia al tiempo requerido para el desarrollo de proyectos de software utilizando un paradigma particular. Los cuantificadores se han obtenido en base a: (a) instrumentos cuantitativos de medición como Puntos de Función, COCOMO u otros, (b) en la experiencia de las personas y la madurez de la organización en sus procesos de desarrollo. En este criterio, los cuantificadores posibles son:

- ▶ **Alto** (cuyo valor se consigue cuando los tiempos de desarrollo para ejecutar el proyecto de software son mayores de un año)
- ▶ **Medio** (cuyo valor se consigue cuando los tiempos de desarrollo para ejecutar el proyecto de software son mayores de entre 6 meses y un año)
- ▶ **Bajo** (cuyo valor se consigue cuando los tiempos de desarrollo para ejecutar el proyecto de software son menos de 6 meses)

Costos de los proyectos

Este criterio hace referencia a los costos tangibles e intangibles para poder llevar a cabo proyectos de software. Los cuantificadores se han obtenido en base a: (a) a instrumentos cuantitativos de medición como Puntos de Función, COCOMO u otros, y (b) en la experiencia de las personas y la madurez de la organización en sus procesos de desarrollo. Adicionalmente, los costes deben evaluarse en función de cualitativos como el coste de no-calidad, de la rentabilidad a obtener y la inversión a incurrir en términos organizacionales (no limitados a los usados en un proyecto). En este criterio, los cuantificadores posibles son:

- ▶ **Alto** (cuyo valor se consigue cuando se requiere invertir grandes cantidades de dinero)
- ▶ **Medio** (cuyo valor se consigue cuando se requiere invertir un valor considerable de dinero)
- ▶ **Bajo** (cuyo valor se consigue cuando se requiere invertir pocas cantidades de dinero)

Calidad del software

Este criterio hace referencia que el paradigma asegura, de una manera objetiva, que los productos software y los procesos son conformes a los requerimientos especificados y se ajustan a los planes establecidos. Los cuantificadores se han obtenido en base a: (a) la consideración de instrumentos cuantitativos de medición de tiempo como Puntos de Función, COCOMO u otros que aportan luces de sobre tiempo y coste, y (b) en la experiencia de las personas y la madurez de la organización en sus procesos de desarrollo. Cabe añadir, que la calidad no es un bien o un activo organizacional transable, es una decisión de alto riesgo. Ningún proyecto debería comenzar con una calidad esperada que no sea ALTA. En este criterio, los cuantificadores posibles son:

- ▶ **Bajo** (cuyo valor se consigue cuando el paradigma no contempla entre sus fases el aseguramiento de la calidad del software)
- ▶ **Alto** (cuyo valor se consigue cuando el paradigma contempla entre sus fases el aseguramiento de la calidad del software)

Documentación

Este criterio hace referencia que el paradigma contempla el proceso para registrar la documentación producida por un proceso o actividad del ciclo de vida. El proceso contiene el conjunto de actividades para planificar, diseñar, desarrollar, producir, editar, distribuir y mantener aquellos documentos que necesitan todos los involucrados tales como gerentes, ingenieros y usuarios del sistema o producto software. Los cuantificadores se han obtenido en base a la conjugación de consideraciones como: (a) la cantidad de documentos a generar, (b) el nivel de “especificación” exigida, (c) el proceso de actualización, control y autorización de documentos, y (d) las copias a generar y distribuir. Cabe añadir que si bien no incluyen aspectos de documentación, NO hay que dejar de considerar el uso de instrumentos cuantitativos de medición de tiempo como Puntos de Función, COCOMO u otros, la experiencia de las personas y la madurez de la organización en sus procesos de desarrollo. En este criterio, los cuantificadores posibles son:

- **Bajo** (cuyo valor se consigue cuando el paradigma no contempla entre sus fases la documentación),
- **Medio** (cuyo valor se consigue cuando el paradigma contempla entre sus fases la documentación de forma no tan estricta); y,
- **Alto** (cuyo valor se consigue cuando el paradigma contempla entre sus fases la documentación).

Análisis por criterio de los paradigmas

A continuación se comparan los paradigmas en base a los criterios anteriormente introducidos. Se destaca que los valores (cuantificadores) puestos en cada paradigma por cada criterio, son referenciales y pueden variar según cada caso de proyecto, y experiencia de las personas y las organizaciones.

CRITERIOS	PARADIGMAS									
	CASCADA	MODELO EN V	RAD	INCREMENTAL	PROTOTIPOS	ESPIRAL	XP	SCRUM	ADS	CM
Disponibilidad de recursos	Todos	Todos	Todos	Algunos	Algunos	Algunos	Algunos	Algunos	Algunos	Algunos
Complejidad del proyecto	Baja	Alta	Media	Media	Media	Alta	Media	Alta	Media	Media
Entendimiento de requerimientos	Específico	Específico	Específico	Bajo	Bajo	Bajo	Bajo	Bajo	Bajo	Alto
Conocimiento del dominio del problema	Alto	Alto	Alto	Regular	Pobre	Pobre	Regular	Regular	Regular	Alto
Manejo de las perspectivas de riesgos	No	No	No	Si	No	Si	No	Si	Si	Si
Tiempos de desarrollo	Medio	Alto	Bajo	Alto	Alto	Bajo	Bajo	Bajo	Bajo	Bajo
Costos de los proyectos	Medio	Alto	Medio	Medio	Medio	Alto	Medio	Medio	Medio	Medio
Calidad del software	Medio	Alto	Medio	Medio	Medio	Alto	Alto	Alto	Alto	Alto
Documentación	Medio	Alto	Bajo	Bajo	Bajo	Alto	Bajo	Bajo	Medio	Bajo

CRITERIOS	PARADIGMAS									
	CASCADA 2	MODELO EN V 2	RAD 4	INCREMENTAL 6	PROTOTIPOS 4	ESPIRAL 4	XP 7	SCRUM 7	ADS 7	CM 6
Disponibilidad de recursos	Todos	Todos	Todos	Algunos	Algunos	Algunos	Algunos	Algunos	Algunos	Algunos
Complejidad del proyecto	Baja	Alta	Media	Media	Media	Alta	Media	Alta	Media	Media
Entendimiento de requerimientos	Específico	Específico	Específico	Bajo	Bajo	Bajo	Bajo	Bajo	Bajo	Alto
Conocimiento del dominio del problema	Alto	Alto	Alto	Regular	Pobre	Pobre	Regular	Regular	Regular	Alto
Manejo de las perspectivas de riesgos	No	No	No	Si	No	Si	No	Si	Si	Si
Tiempos de desarrollo	Medio	Alto	Bajo	Alto	Alto	Bajo	Bajo	Bajo	Bajo	Bajo
Costos de los proyectos	Medio	Alto	Medio	Medio	Medio	Alto	Medio	Medio	Medio	Medio
Calidad del software	Medio	Alto	Medio	Medio	Medio	Alto	Alto	Alto	Alto	Alto
Documentación	Medio	Alto	Bajo	Bajo	Bajo	Alto	Bajo	Bajo	Medio	Bajo

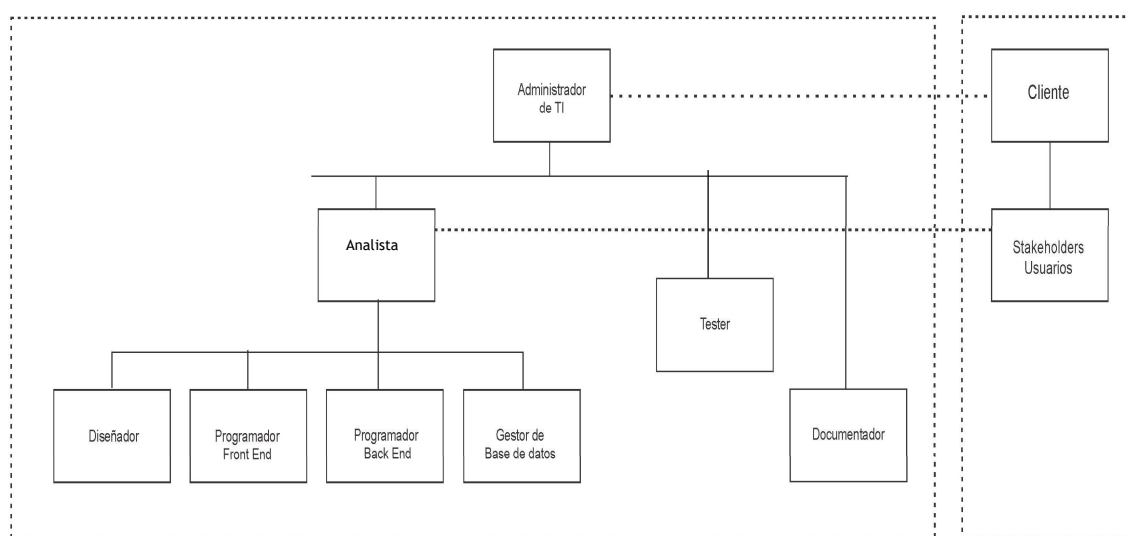
Implementación de organigrama de trabajo

- ▶ La estructura organizacional define al organigrama de autoridad y control dentro de la empresa. Así se determina cómo funciona la relación jerárquica en la organización. El modo en que esté constituida será crucial para el desarrollo de tus proyectos.
- ▶ El modo de dirigir los proyectos estará condicionado por la forma en que la empresa esté estructurada. En términos generales afecta a la disponibilidad de recursos, a la relación entre las diferentes áreas de la compañía o a la priorización de objetivos.
- ▶ Hay algo que muchas veces olvidamos, quizá porque se puede considerar evidente. Sin embargo, es de gran importancia. Debe haber una alineación entre el equipo, la gestión del proyecto y la estructura de la organización. Éste es uno de los primeros retos en la gestión de proyectos a los que te enfrentarás. Se trata de una idea previa a la gestión de proyectos, algo que existe antes de que se conciba el proyecto. Y, sin embargo, condiciona toda su ejecución.

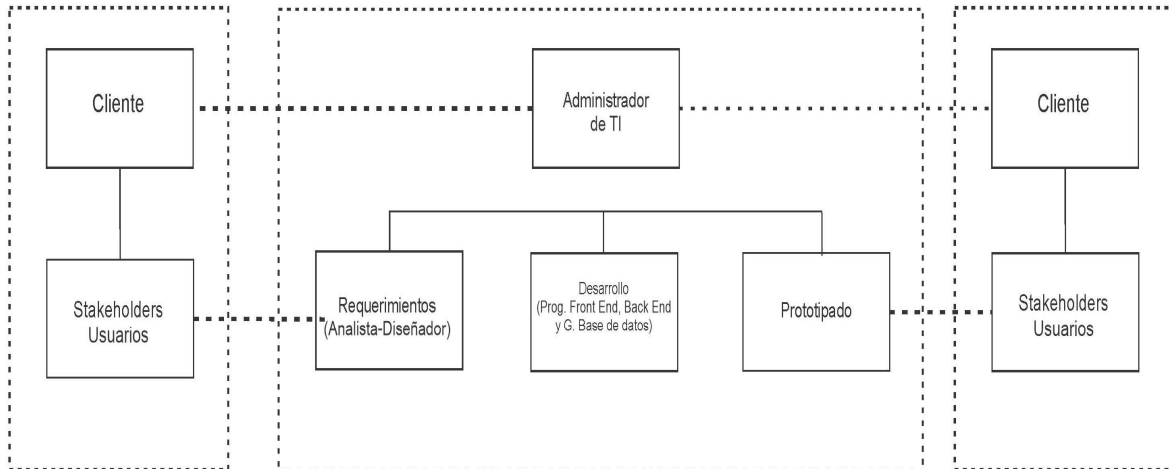
Ventajas de la estructura a través de organigramas orientada a proyectos

- ▶ La estructura jerárquica de la organización facilita la existencia de una línea de autoridad evidente. Lo que facilita una toma de decisiones rápida y eficaz.
- ▶ La comunicación dentro del equipo de un proyecto es muy fluida.
- ▶ Los miembros de los equipos ganan en experiencia al participar en diferentes modalidades de proyectos.
- ▶ La eficiencia en la ejecución de los proyectos es muy elevada.
- ▶ El nivel de compromiso y motivación se eleva sustancialmente. Cada miembro de un equipo se vuelca en el proyecto que esté desarrollando.
- ▶ Se trata de una estructura flexible. Dependiendo de las necesidades y de los proyectos en ejecución los equipos tienen la posibilidad de adaptarse. Equipos grandes pueden desglosarse en varios pequeños. Los pequeños pueden agruparse en uno grande.

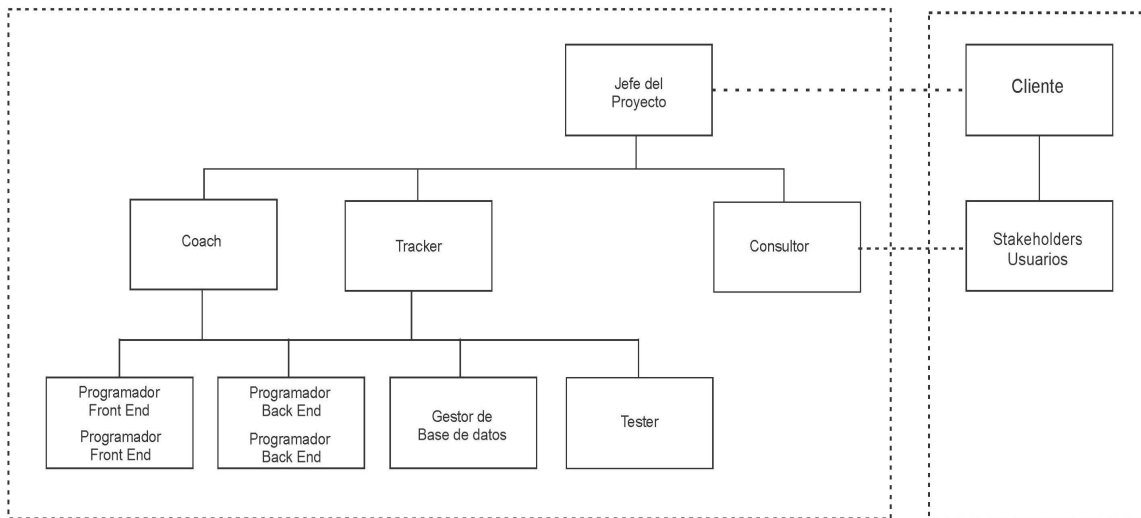
Organigrama por roles del modelo en Cascada y modelo en V



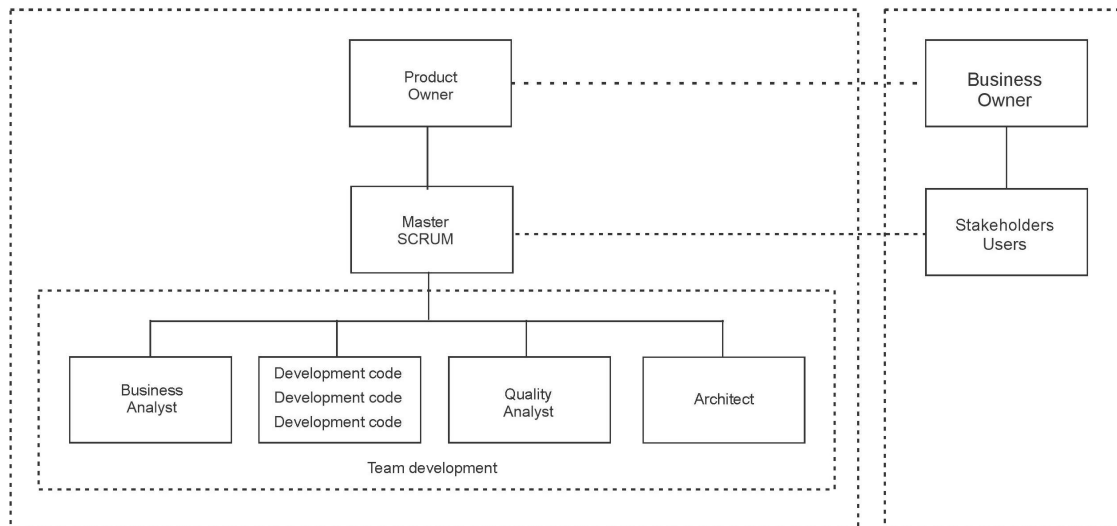
Organigrama por roles del modelo en Prototipos



Organigrama por roles del modelo XP



Organigrama por roles de la metodología SCRUM



Referencias

- ▶ Saurith, Arleth; y, Estay-Niculcar, Christian. (2010). *Análisis y Diseño Integral de Sistemas y Requerimientos*. Fundación Universitaria Iberoamericana. ISBN. 978-84-96542-77-8 . Mayo. 167 pp. Barcelona, España. [Ver](#).
- ▶ Algo más del tema y puesto en la misma referencia se puede consultar en el post “[Information Systems Philosophy: ¿Cómo modelar sistemas para insertarlos en organizaciones? una visión sistémica orgánico-estructural para una deconstrucción pensada en usuarios y operadores \(2010\)](#)” -Febrero 28, 2010-.
- ▶ Más ideas pueden enlazarse con este el post “[Information Systems Philosophy: los 5 sentidos como base de mejora de competencias en Analistas de Sistemas ... reporte de experiencia](#)” -Febrero 23, 2011- o con el post “[Information Systems Philosophy: el rol de los Sistemas de Información y la filmografía: 3 roles organizacionales ... 3 películas \(TRON, You’ve got email, Firewall\)](#)” -Abril 25, 2011-.
- ▶ Estay, C. (2020) Information Systems Philosophy: Criterios para tomar la decisión sobre cual modelo de desarrollo de software emplear en un proyecto o ante un determinado problema [Blog] recuperado de <https://cestay.wordpress.com/2011/03/05/sistemas-socio-tecnicos-criterios-para-tomar-la-decision-sobre-cual-modelo-de-desarrollo-de-software-emplear-en-un-proyecto-o-ante-un-determinado-problema/>