



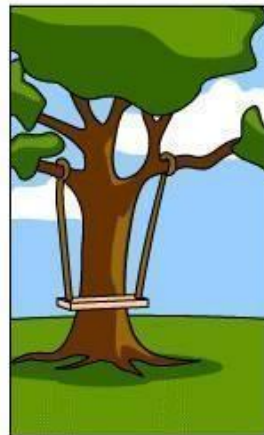
Metodologías de desarrollo

ING. FRANCISCO YUMAN

¿Conoces el análisis del columpio en el árbol?



Como el cliente lo explicó



Como el lider del proyecto lo entendió



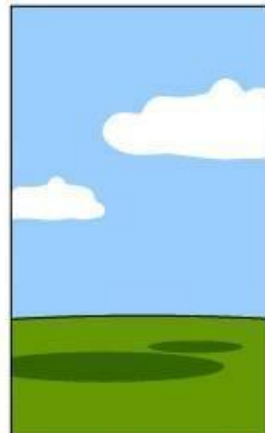
Como el analista lo diseñó



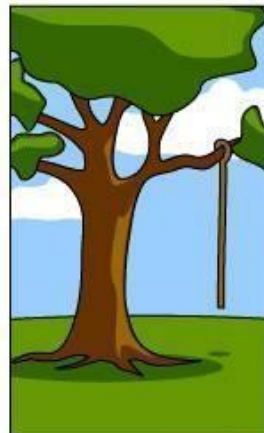
Como el programador lo escribió



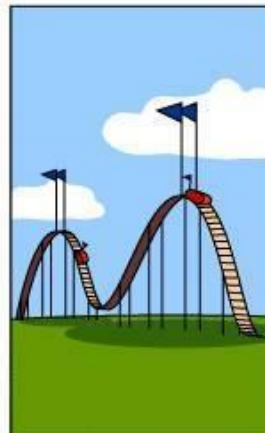
Como el vendedor lo describió



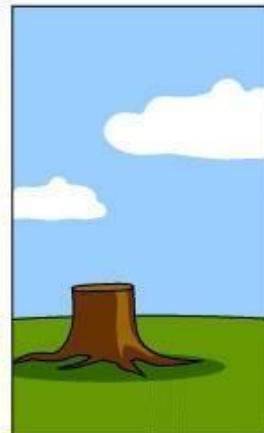
Como fue documentado el proyecto



Que aplicaciones se instalaron



Como le fue facturado al cliente

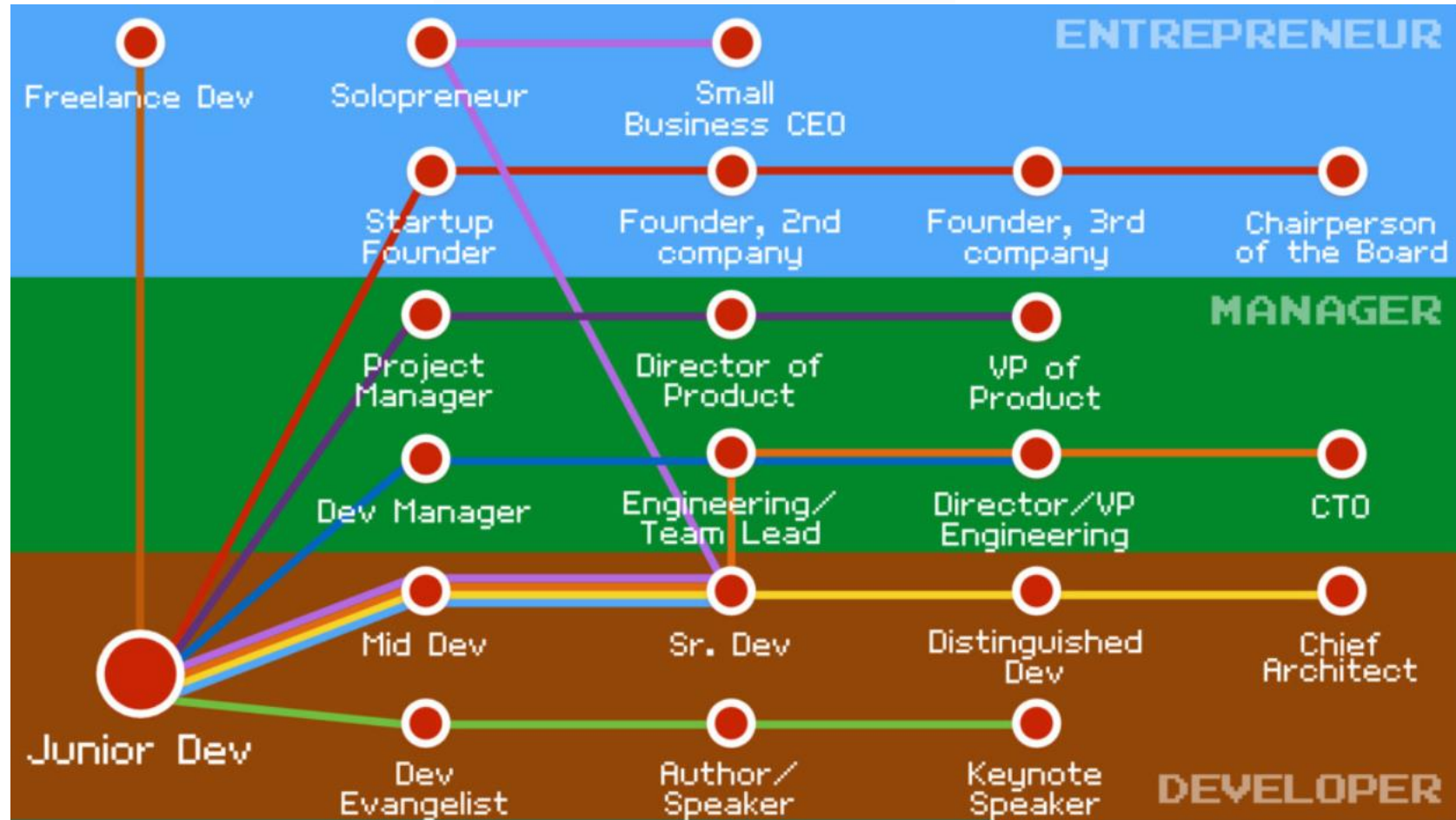


Como se le dio soporte



Lo que el cliente realmente necesitaba

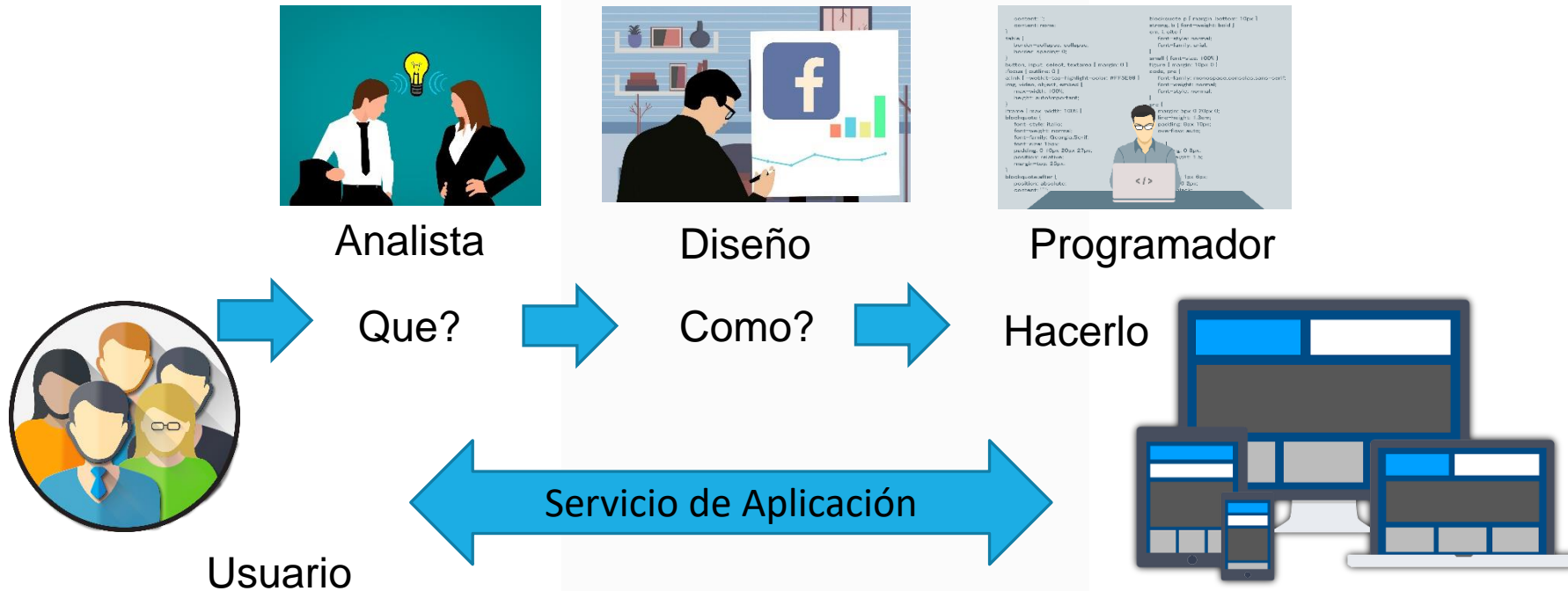
La vida de un desarrollador de Software



Fuente: <https://speakerdeck.com/tehvik/hacking-spacetime-for-a-successful-career?slide=77>

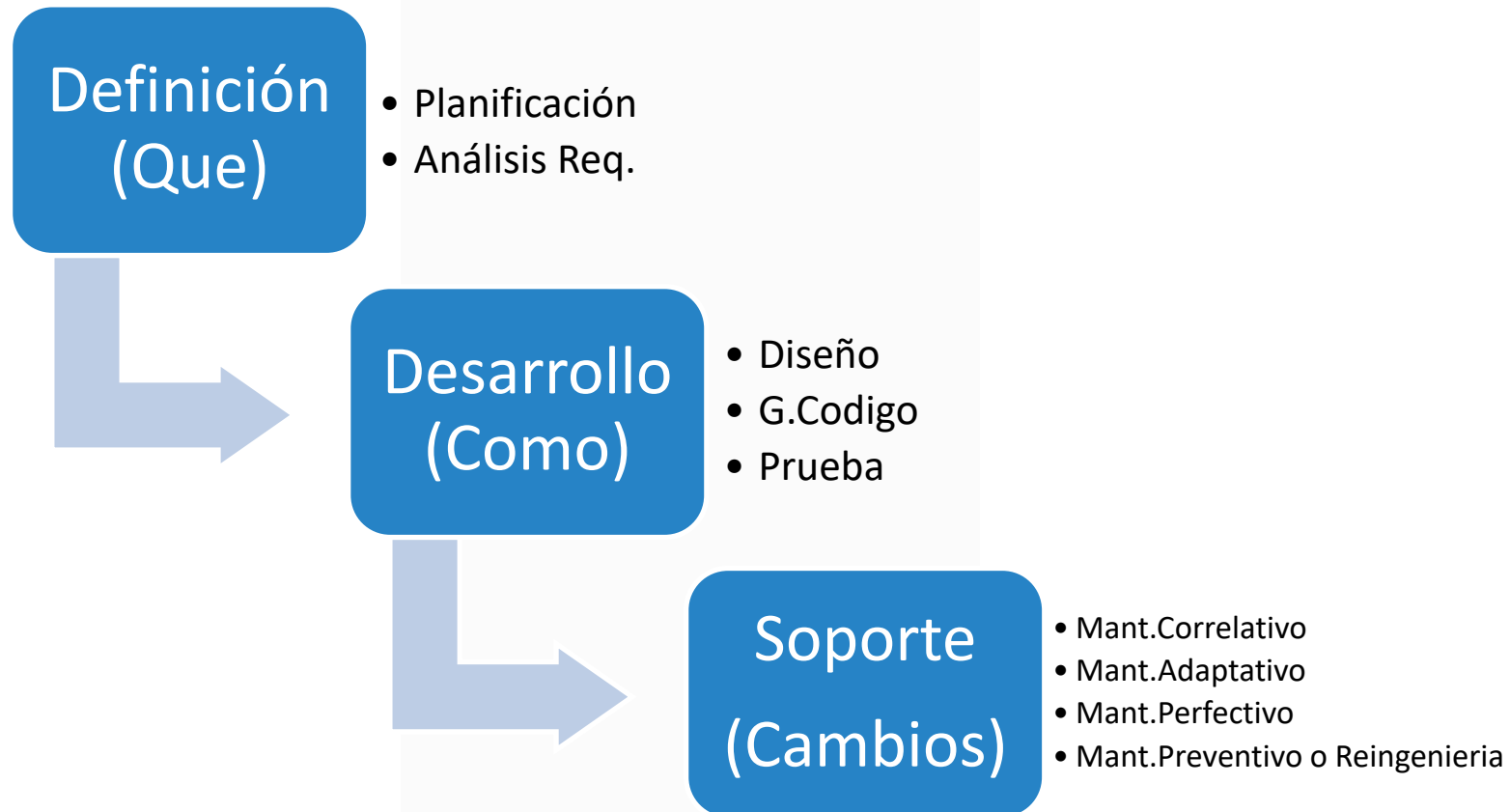
Conceptos Básicos

- Que es el desarrollo de Software



Proceso de Desarrollo

- El Proceso: Marco de trabajo a realizar para desarrollar Software de alta calidad.





Metodologías tradicionales en el desarrollo de software

- Qué es una Metodología?

En desarrollo de Software existen gran variedad de metodologías enfocadas a brindar a el equipo de trabajo (Análisis, Diseño, Desarrollo, Pruebas, Etc) Lineamientos para la construcción de un sistema de calidad



Metodologías Tradicionales

Se enfocan principalmente en procesos asociados a la planeación y control del proyecto con un alto detalle en la especificación de requisitos, el diseño del sistema, imponiendo la metodología sobre el proceso de desarrollo.



Metodologías Tradicionales

En las metodologías tradicionales las fases del ciclo de vida del software van encadenadas.

Al terminar cada fase, las salidas resultantes definen las entradas para la siguientes fases...



Metodologías Tradicionales

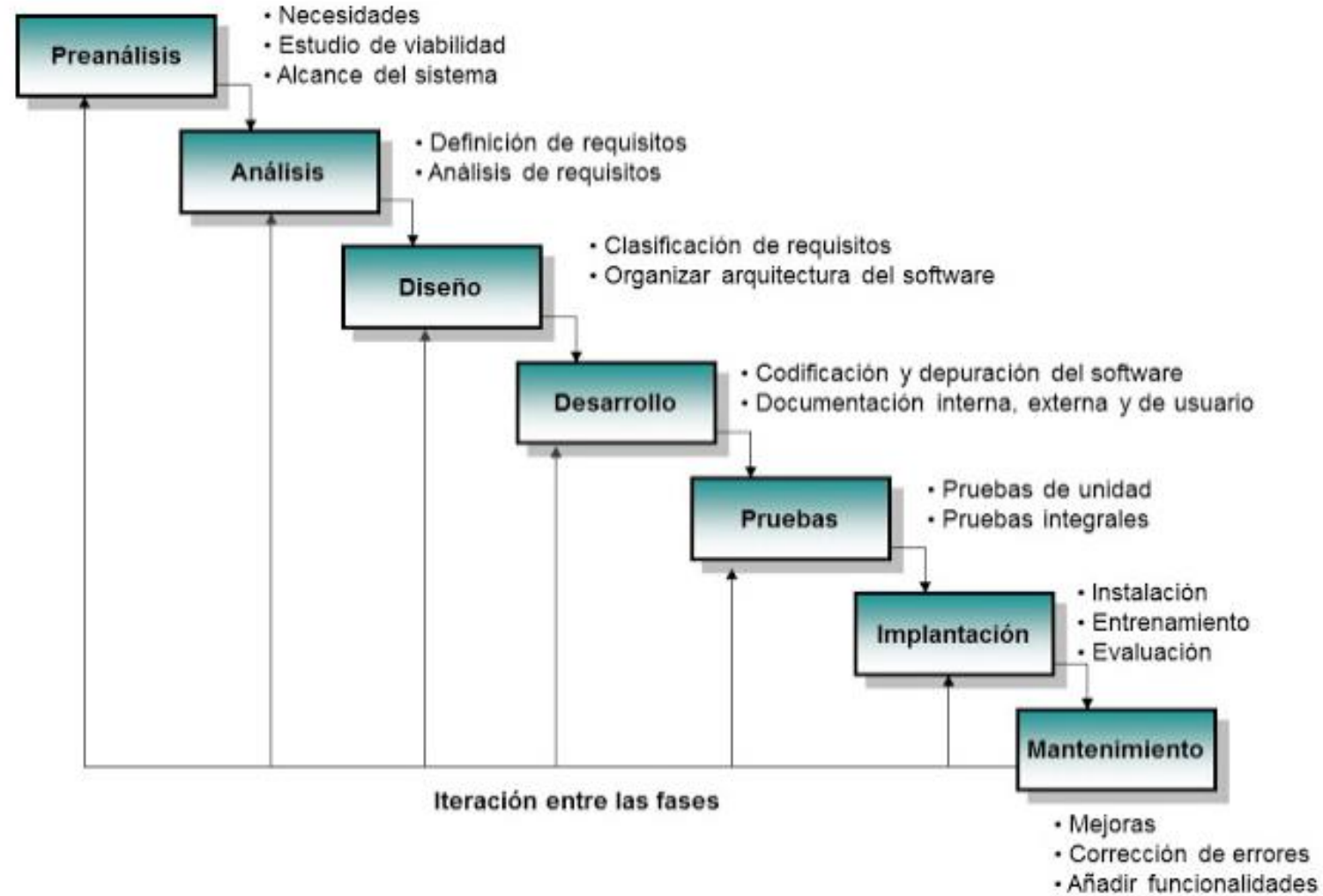
Estas metodologías se caracterizan por un riguroso trabajo a nivel de definición de roles, requisitos, diseño y modelado y un alto grado de documentación.



Ciclo de Vida del Software

Estas metodologías se caracterizan por un riguroso trabajo a nivel de definición de roles, requisitos, diseño y modelado y un alto grado de documentación.

Ciclo de Vida del Software





Metodologías Tradicionales

Como alternativa para abordar el ciclo de vida del software se crearon los modelos del ciclo de vida que buscan brindar lineamientos de trabajo, tales como el modelo en cascada, en espiral, lineal, en V etc, usados dependiendo del contexto y tamaño del proyecto...



Metodologías Tradicionales

Rup

(Rational Unified Process)

MSF

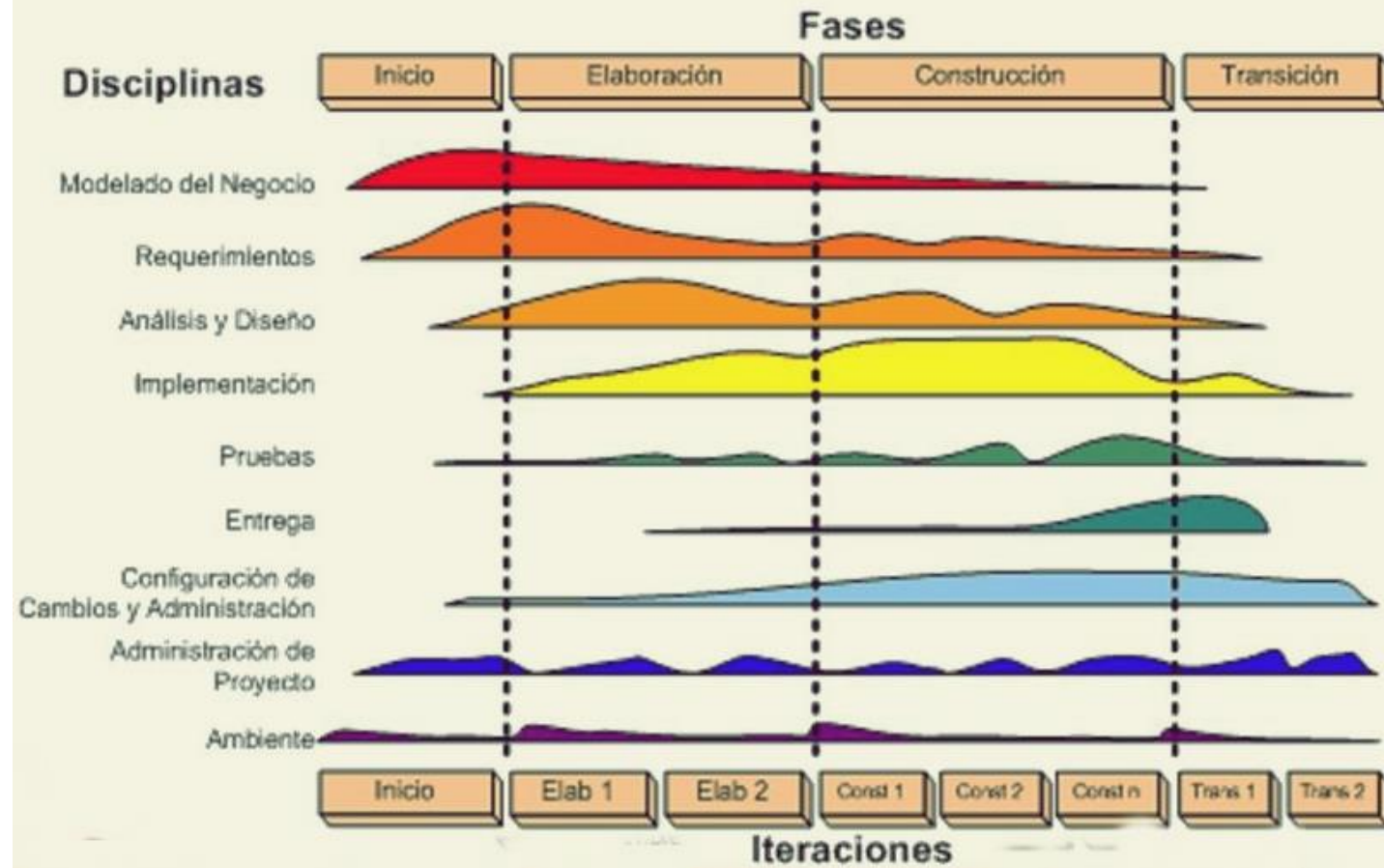
(Microsoft Solutions Framework)

MSF

Ciclo de Vida del Proyecto



RUP





Desventajas

Se debe esperar la finalización de una etapa para poder iniciar la siguiente.

Si algo está mal en una etapa y no se identifica, afecta las etapas posteriores.

El control de cambios es un proceso normal dentro del desarrollo de software, en las metodologías tradicionales abordar los cambios por lo regular se torna complejo.



Desventajas

El cliente no se contempla como un miembro del equipo.

Por lo regular el cliente debe esperar hasta el final para conocer el producto.



¿Qué son las metodologías Ágiles en el desarrollo de Software?

Se enfocan en procesos incrementales con entregas funcionales, buscan aumentar la confianza de los clientes al vincularlos en el proceso mediante cooperación entre estos y el equipo de trabajo...



¿Qué es una Metodología Ágil?

Se caracterizan por reducir la gran cantidad de documentación que es común en las tradicionales.

Buscan ser Adaptables

Se basadas en el modelos de desarrollo Incrementales.

Manifiesto Ágil

Define las directrices que debe cumplir una metodología para considerarse Ágil

- Individuos e interacciones
- Software funcionando
- Colaboración con el cliente
- Respuesta ante el cambio

- Procesos y herramientas
- Documentación exhaustiva
- Negociación contractual
- Seguir un plan





Metodologías Ágiles

- eXtreme Programming XP
- Iconix
- Test Drive Development (TDD)
- SCRUM
- Kanban



Manifiesto Ágil

Estas Metodologías se pueden adaptar al problema en cuestión sin importar el tamaño, trabajando por fases definidas y descomponiendo el proyecto para facilitar su gestión.



Manifiesto Ágil

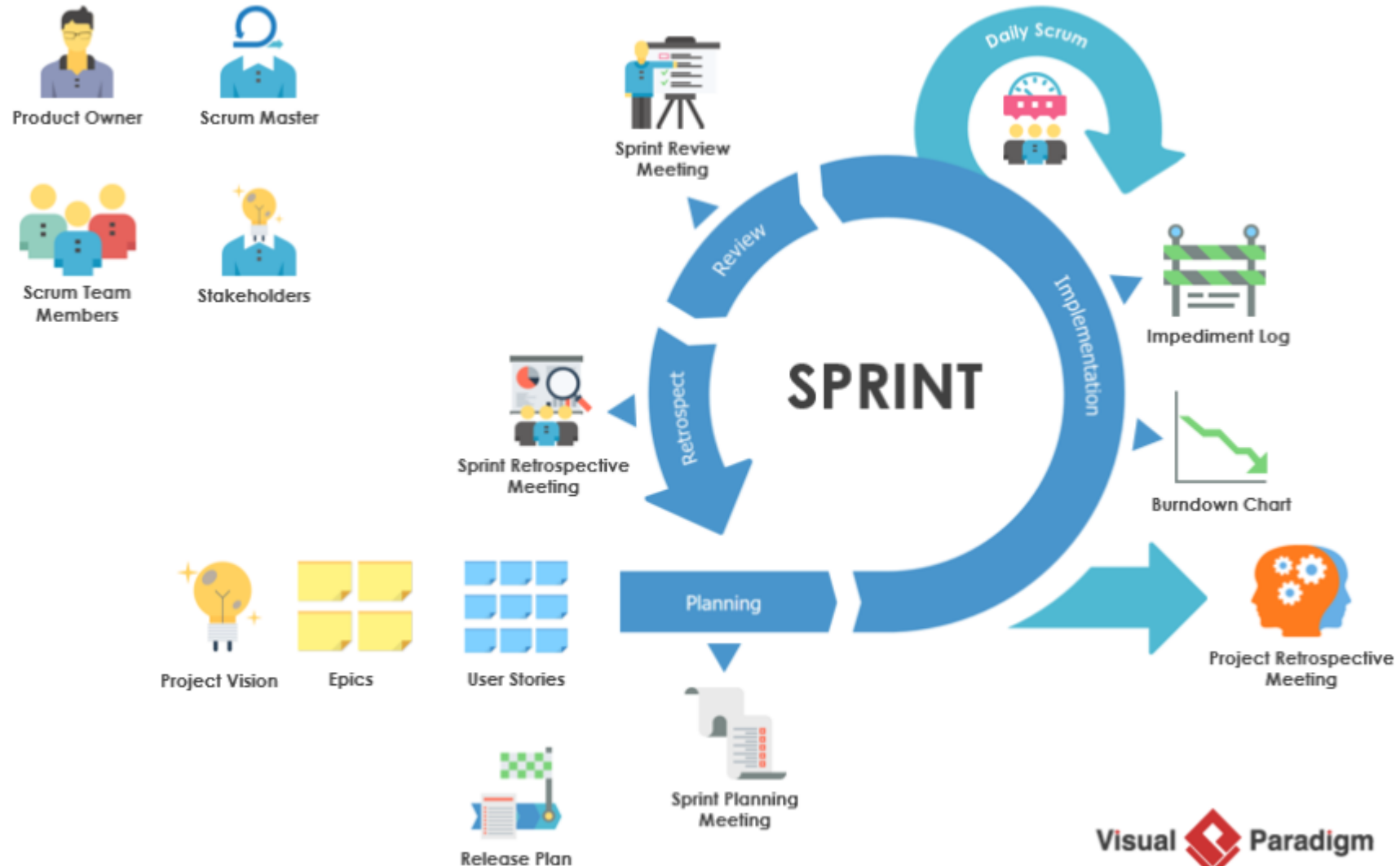
En SCRUM por ejemplo al trabajar por etapas o módulos (Sprint) se ataca una parte del proyecto facilitando su gestión de forma independiente...

¿Qué es SCRUM?

Scrum es un marco de trabajo para el desarrollo y mantenimiento de productos complejos o sencillos.

Es una de las metodológicas ágiles más populares y usadas en proyectos de software, aunque una de sus ventajas es la adaptabilidad lo que hace ideal para trabajar en diferentes contextos.

SCRUM

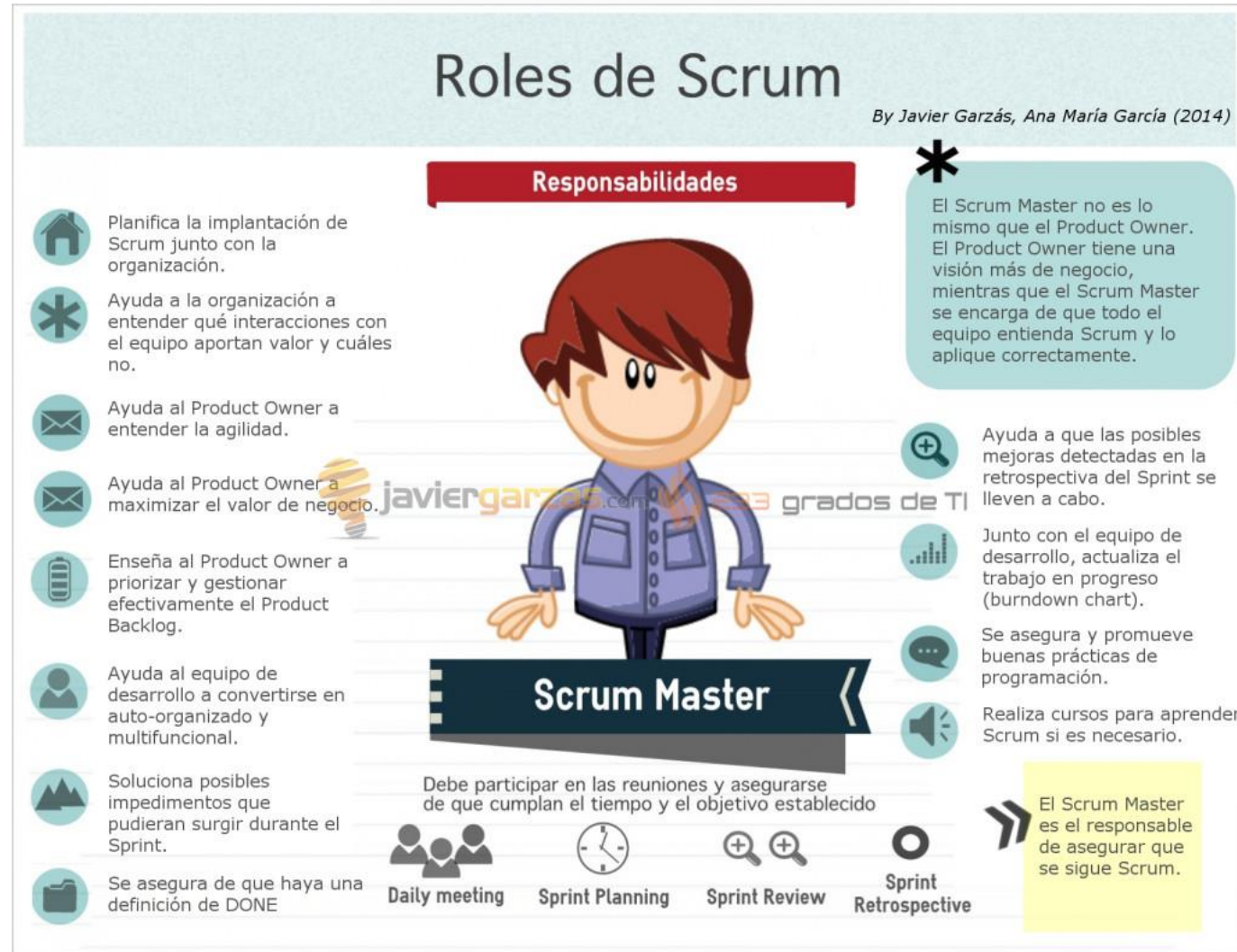


Product Owner (Dueño del producto)



Es la representación del cliente dentro del equipo de trabajo, su principal responsabilidad es expresar claramente la necesidad del cliente del Product Backlog.

SCRUM Master



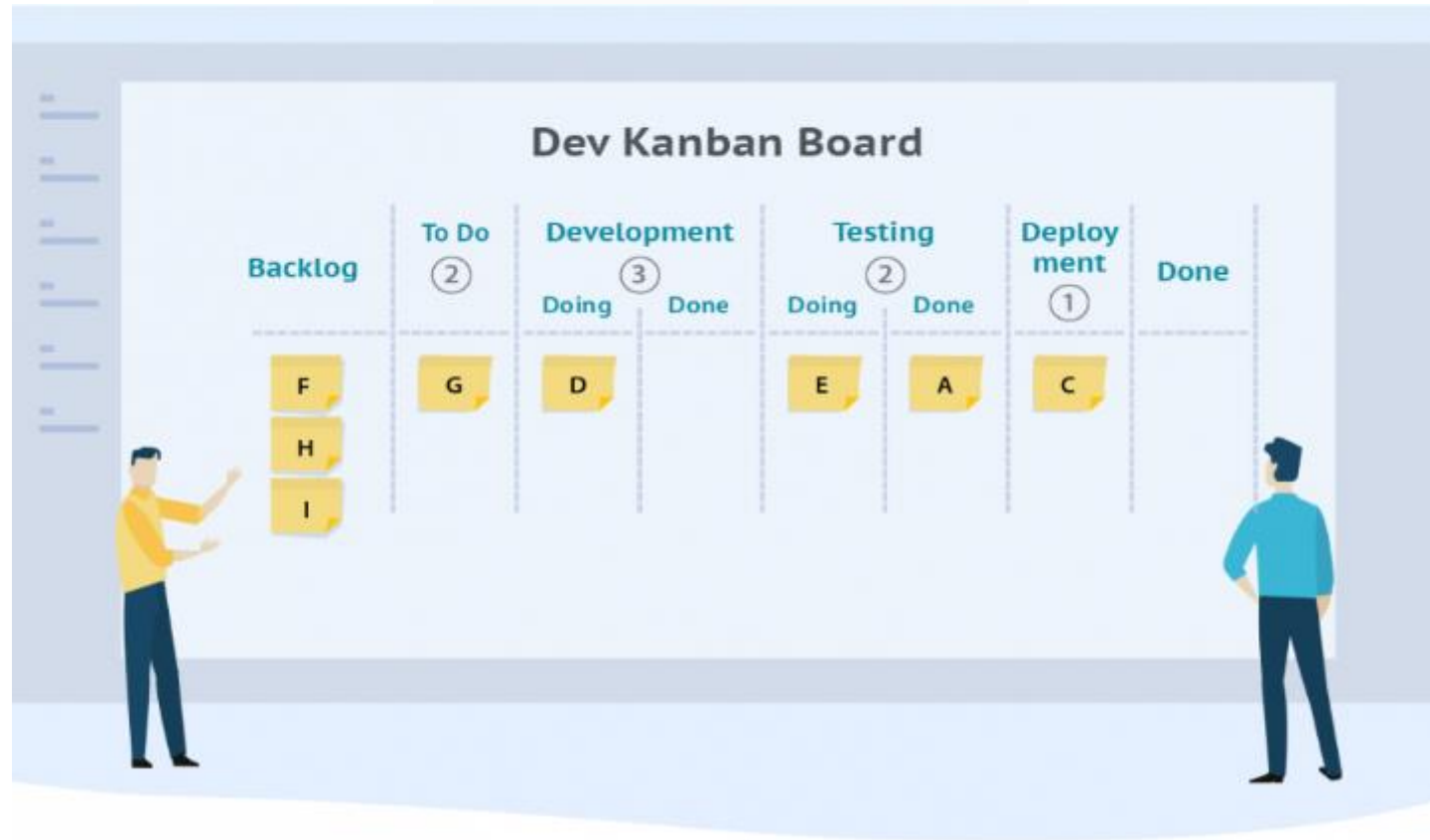
Fuente: <https://www.javiergarzas.com/2014/03/el-rol-del-scrum-master.html>

Development Team



El Equipo de desarrollo se compone de las personas responsables de dar cumplimiento a los SPRINT, son un equipo autogestionado y organizado.

Introducción a Kanban



Planeación

- Requerimientos

DERCAS'

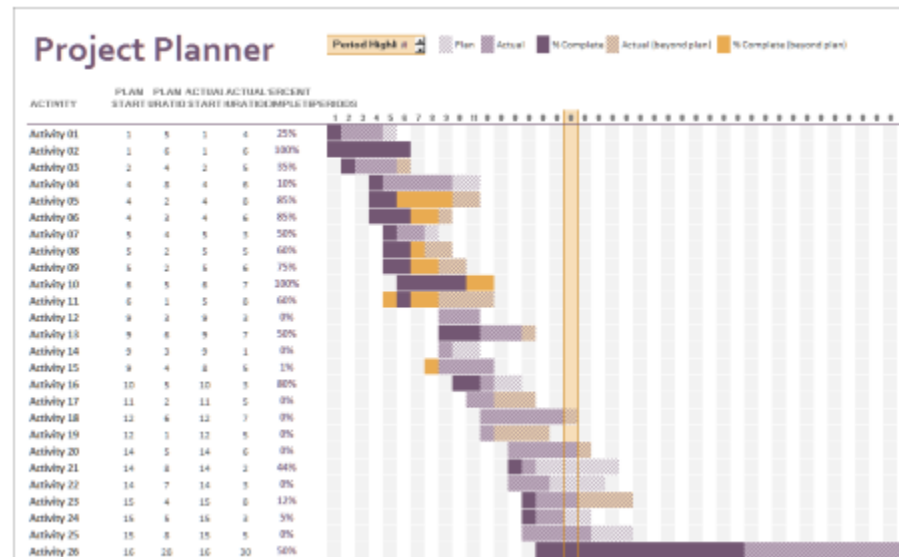
"Documento de Especificaciones, Requerimientos y Criterios de Aceptación de Software". El DERCAS es una metodología de la Ingeniería de software que permite definir los pasos esenciales para el análisis y desarrollo de un proyecto de software... Generalmente se realiza al principio del proyecto, e involucra tanto al arquitecto de software, al administrador del proyecto y a otros miembros del staff, que harán diversas visitas a los clientes para levantado de requerimientos y así enmarcar en un contexto la solución a desarrollar.

Plan de Acción

Se realiza un plan de actividades con tiempos y responsables para el logro de los objetivos, en el cual se especifican las acciones que han de llevarse a cabo.

Se elabora un diagrama de Gantt para listar e ilustrar las actividades.

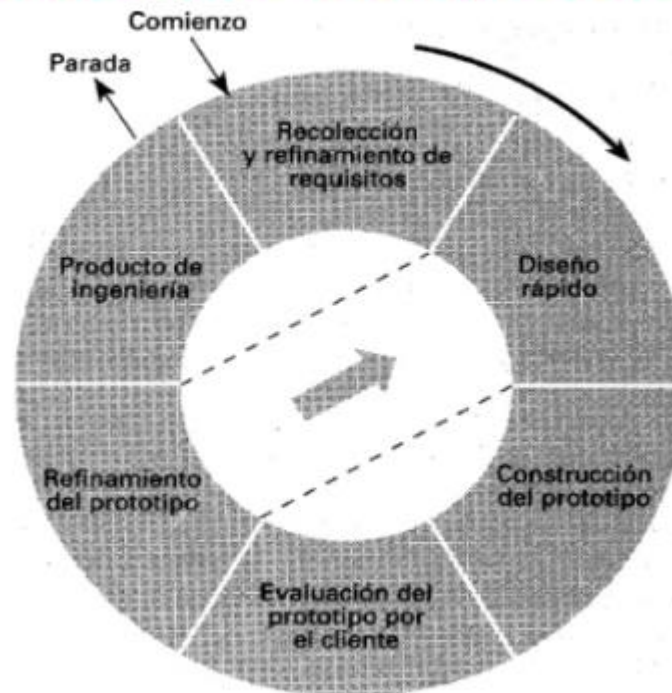
El diagrama de Gantt es una útil herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.



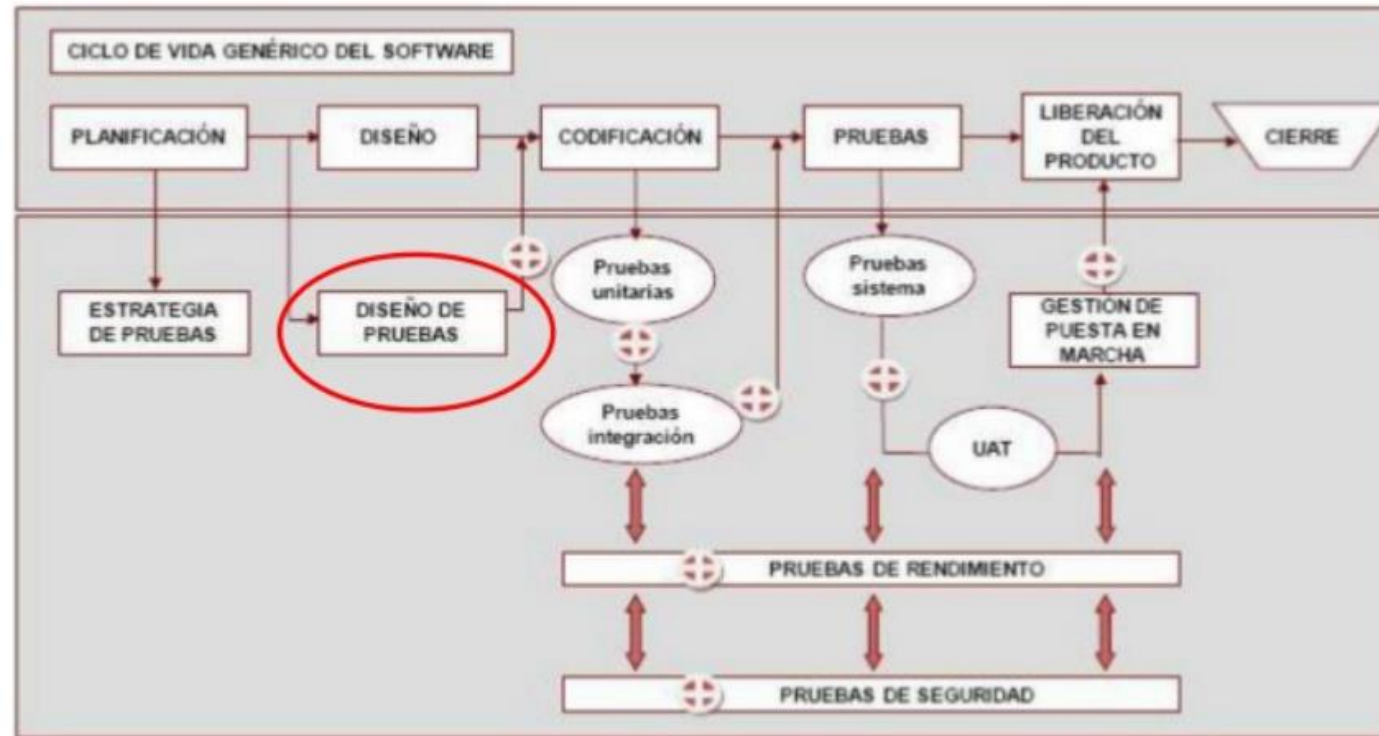
Prototipos

El Modelo de **prototipos**, en Ingeniería de **software**, pertenece a los modelos de desarrollo evolutivo. El **prototipo** debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar muchos recursos.

Modelo ciclo de vida prototipado



Diseño de Pruebas



Casos de prueba

En esta última etapa, se generan los casos de pruebas. Para ello, se considera como referencia la tabla de condiciones de entrada, indicando en cada caso de prueba las clases de equivalencia creadas.

Por ejemplo, para el caso bancario se tendría lo siguiente:

ID CP	Clases de equivalencia	CONDICIONES DE ENTRADA					Resultado esperado
		Código de banco	Código de sucursal	Número de cuenta	Clave personal	Orden	
CP1	CEV<02>, CEV<03>, CEV<04>, CEV<05>, CEV<07>	200	1000	10000	Aaaaa	"Talonario"	Mensaje "Envío de talonarios"
CP2	CEV<01>, CEV<03>, CEV<04>, CEV<05>, CEV<08>	820	9999	99999	Zzzzz	"Movimientos"	Mensaje "Envío de movimientos "
CP3	CEV<02>, CEV<03>, CEV<04>, CEV<05>, CEV<06>	999	1001	12345	A1b2c	"Seleccione Orden"	Mensaje "Envío de talonarios y movimientos"
CP4	CENV<01>, CEV<03>, CEV<04>, CEV<05>, CEV<07>	30A	1989	12345	1a2b3	"Seleccione Orden"	Mensaje "Código de banco incorrecto"
CP5	CENV<04>, CEV<03>, CEV<04>, CEV<05>, CEV<07>	210	999	12345	1a2b3	"Seleccione Orden"	Mensaje "Código de sucursal incorrecto"
CP6	CENV<07>, CEV<03>, CEV<04>, CEV<05>, CEV<07>	210	1989	123	1a2b3	"Seleccione Orden"	Mensaje "Número de cuenta incorrecto"
CP7	CENV<09>, CEV<03>, CEV<04>, CEV<05>, CEV<07>	210	1989	12345	""	"Seleccione Orden"	Mensaje "Clave incorrecta"

Plan de prueba - Artefacto

El propósito del plan de pruebas es explicitar el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos de un proceso de pruebas. Note que puede haber un plan global que explicita el énfasis a realizar sobre los distintos tipos de pruebas (verificación e integración).

Un plan de pruebas incluye:

1. Identificador del plan.

Preferiblemente de alguna forma mnemónica que permita relacionarlo con su alcance.

Por ejemplo:

- TP-Global (plan global del proceso de pruebas)
- TP-Req-Seguridad1 (plan de verificación del requerimiento 1 de seguridad)
- TP-Contr-X (plan de verificación del contrato asociado al evento de sistema X).

Plan de prueba

2. Alcance.

Indica el tipo de prueba y las propiedades/elementos del software a ser probado.

3. Ítems de prueba

Indica la configuración a probar y las condiciones mínimas que debe cumplir para comenzar a aplicarle el plan.

4. Estrategia

Describe la técnica, patrón y/o herramientas a utilizarse en el diseño de los casos de prueba.

Por ejemplo, en el caso de pruebas unitarias de un procedimiento, esta sección podría indicar: "Se aplicará la estrategia caja-negra de pruebas funcionales" o "Ejercicio de los caminos ciclomáticos válidos".

En lo posible la estrategia debe precisar el número mínimo de casos de prueba a diseñar, por ejemplo: 80% de funcionalidades del documento de requisitos, 60% de los caminos ciclomáticos.

La estrategia también explicita el grado de automatización que se exigirá, tanto para la generación de casos de prueba como para su ejecución.

Plan de prueba

5. Categorización de la configuración

Explicita las condiciones bajo las cuales, el plan debe ser:

- Suspendido
- Repetido
- Culminado

6. Tangibles

Explicita los documentos a entregarse al culminar el proceso previsto por el plan.

Ejemplo: Sub planes, especificación de pruebas, casos de prueba, resumen gerencial del proceso y bitácora de pruebas.

7. Procedimientos especiales

Identifica el grafo de las tareas necesarias para preparar y ejecutar las pruebas, así como cualquier habilidad especial que se requiere.

Plan de prueba

8. Recursos

Especifica las propiedades necesarias y deseables del ambiente de prueba, incluyendo las características del hardware, el software de sistemas (Ejemplo: Sistema operativo), cualquier otro software necesario para llevar a cabo las pruebas, así como la colocación específica del software a probar (módulos se colocan en qué máquinas de una red local) y la configuración del software de apoyo. La sección incluye un estimado de los recursos humanos necesarios para el proceso.

9. Calendario

Esta sección describe los hitos del proceso de prueba y el grafo de dependencia en el tiempo de las tareas a realizar.

10. Manejo de riesgos

Explicita los riesgos del plan, las acciones mitigantes y de contingencia.

11. Responsables

Especifica quién es el responsable de cada una de las tareas previstas en el plan.

Gracias

¿ALGUNA PREGUNTA?