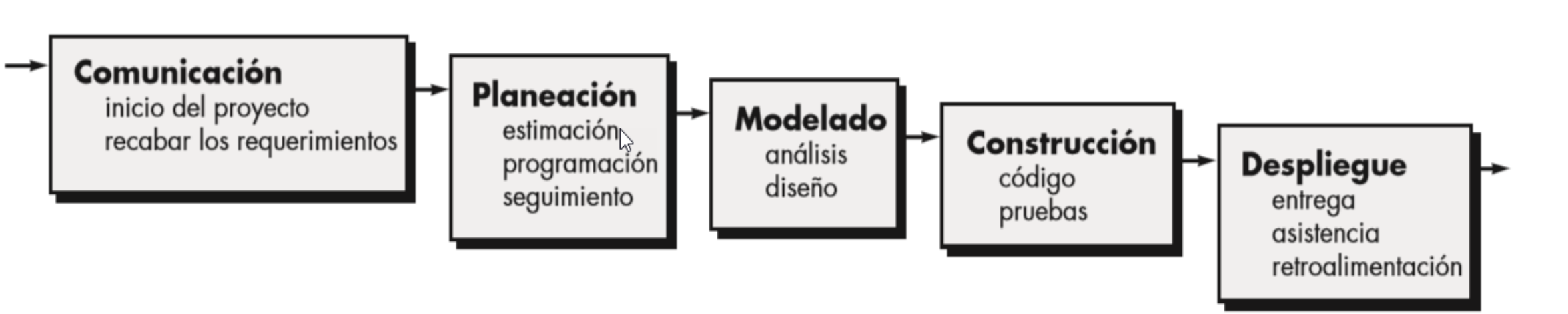

DOCUMENTATION
MAINTENANCE
SOFTWARE
PROJECT
DEVELOPMENT
SEQUENTIAL
WATERFALL
DESIGN
MODEL
PROCESS
PHASES
REQUIREMENTS
ANALYSIS
SUPPORT
CODING
VERIFICATION
TESTING

CARACTERISTICAS

- El Modelo en Cascada fue el primer Modelo de proceso que se introdujo.
- Se lo conoce como un Modelo de ciclo de vida lineal-secuencial.
-Durante el desarrollo del producto se tiene una mínima interacción con el cliente.

- Al final de cada fase, se realiza una revisión para determinar si el proyecto se encuentra en el camino correcto y así continuar o descartar el proyecto.
-Las pruebas de software comienzan solo después de que se completa el desarrollo.

EL PROCESO


Comunicación
inicio del proyecto
recabar los requerimientos
Planeación
estimación
programación
seguimiento
Modelado
análisis
diseño
Construcción
código
pruebas
Despliegue
entrega
asistencia
retroalimentación

VENTAJAS

- Este Modelo es simple, fácil de entender y usar.
- Es fácil de administrar debido a la rigidez del Modelo, cada fase tiene entregables específicos y tiene un proceso de revisión.
- En este Modelo las fases se procesan y completan una a la vez y las fases no se superponen.
- El Modelo en Cascada funciona bien para proyectos pequeños donde los requisitos se entienden muy bien

DESVENTAJAS

Una vez que una aplicación está en la etapa de prueba, es muy difícil volver atrás y cambiar errores en las etapas previas.

No se tiene un software funcionando hasta etapas avanzadas en el ciclo de vida.
Altas cantidades de riesgo e incertidumbre..

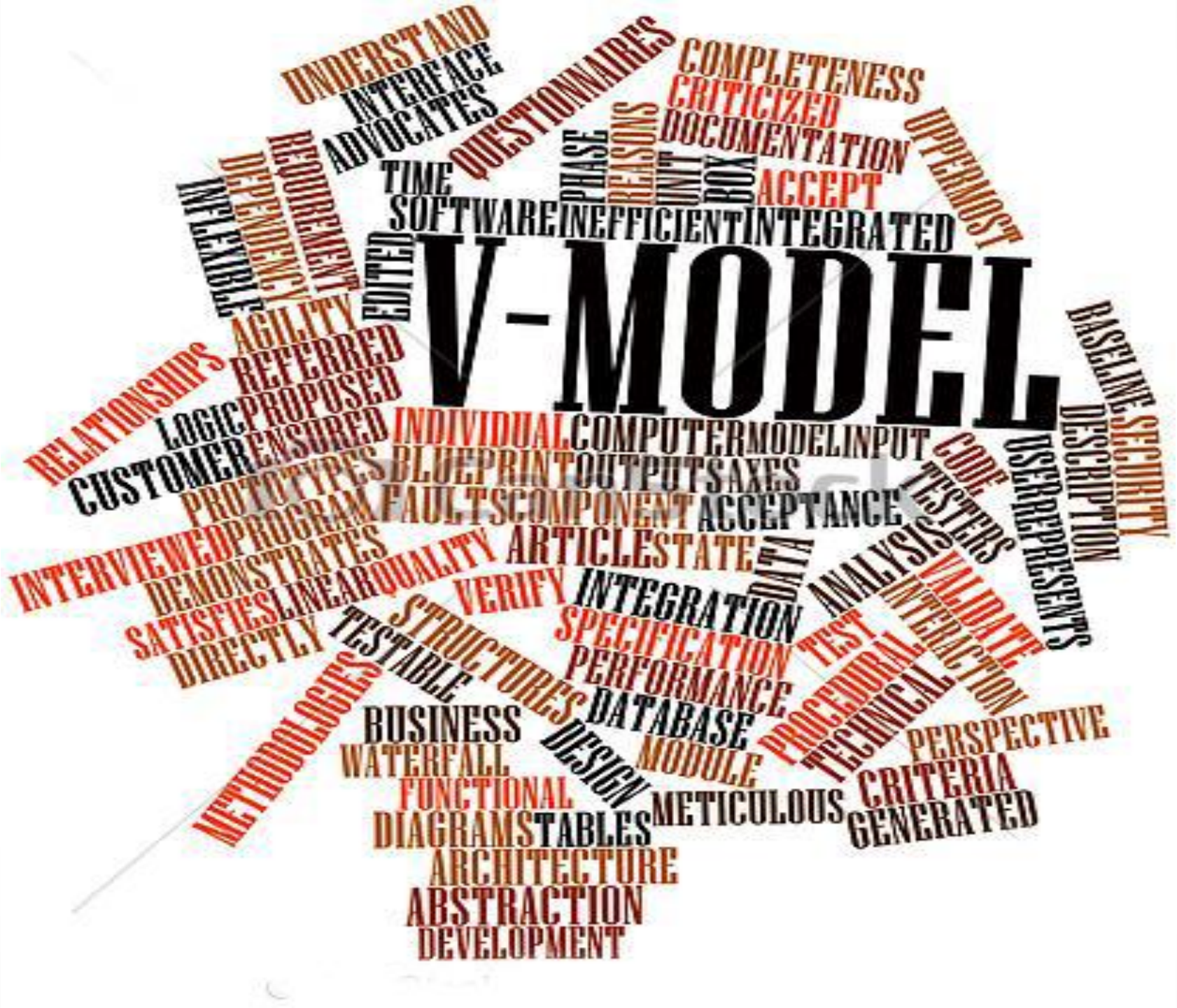
No es un buen modelo para proyectos complejos y orientados a objetos.
Es un Modelo pobre para proyectos largos y continuos.

No es adecuado para los proyectos donde los requisitos tienen un riesgo de cambio medio a alto.

USAR EL MODELO CUANDO …….

- ❑ Los requisitos son muy conocidos, claros y fijos.
- ❑ La definición del producto es estable.
- ❑ Se entiende la tecnología.
- ❑ No hay requisitos ambiguos.
- ❑ El proyecto es corto.

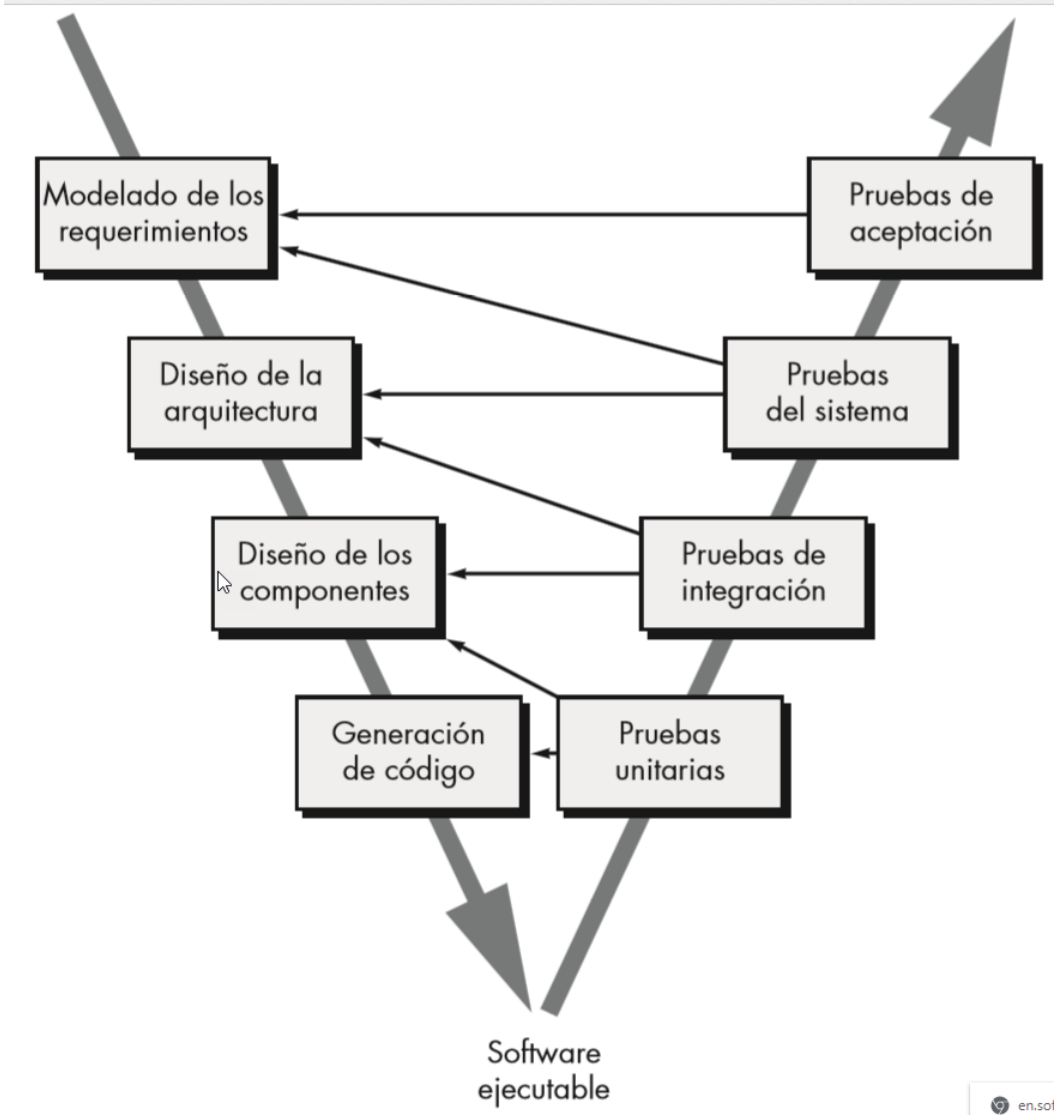

WATERFALL
The first model we want to discuss is the grandfather of
0:01 / 1:14
HD


V-MODEL
UNDERSTAND
INTERFACE
ADVOCATES
QUESTIONNAIRES
COMPLETENESS
CRITICIZED
DOCUMENTATION
UPPERMOST
REASONS
PHASE
UNIT
BOX
ACCEPT
TIME
SOFTWARE
INEFFICIENT
INTEGRATED
REQUIREMENT
DEPENDENCY
INFLEXIBLE
EDITED
AGILITY
REFERRED
PROPOSED
ENSURED
RELATIONSHIPS
LOGIC
CUSTOMER
PROTOTYPES
INDIVIDUAL
COMPUTER
MODEL
INPUT
BLUEPRINT
OUTPUTS
AXES
FAULTS
COMPONENT
ACCEPTANCE
INTERVIEWED
PROGRAM
DEMONSTRATES
SATISFIES
LINEAR
QUALITY
ARTICLE
STATE
VERIFY
DATA
INTEGRATION
DIRECTLY
TESTABLE
STRUCTURES
SPECIFICATION
PERFORMANCE
DATABASE
METHODOLOGIES
BUSINESS
DESIGN
WATERFALL
MODULE
FUNCTIONAL
DIAGRAMS
TABLES
ARCHITECTURE
ABSTRACTION
DEVELOPMENT
METICULOUS
PROCEDURAL
TECHNICAL
TEST
INTERACTION
VALIDATE
ANALYSIS
TESTERS
CODE
USER
REPRESENTS
DESCRIPTION
BASELINE
SECURITY
PERSPECTIVE
CRITERIA
GENERATED

CARACTERÍSTICAS DEL MODELO V

- El Modelo V significa que es un modelo de Verificación y Validación.
- Es un camino en forma de V que ejecuta sus procesos en forma secuencial el cual se asemeja al Modelo en Cascada.
- Se debe completar cada fase antes de que comience la siguiente.
- Las Pruebas del producto se planean en paralelo con una fase del proceso de desarrollo de Software.

LAS FASES DEL MODELO V



REQUISITOS

PRIMERO CONSIDERAR QUE

Así como el Modelo en Cascada, la Especificación de Requisitos del Negocio (BRS) y la Especificación de Requisitos del Software (SRS) son los que comienzan con el Ciclo de Vida de este Modelo.

SEGUNDO

Antes de iniciar un proyecto usando el Modelo V, se debe tener un plan de pruebas del Sistema.

POR ULTIMO ….

El plan de pruebas se debe centrar en cumplir con la funcionalidad especificada en la recopilación de requisitos.

FASES DEL DISEÑO

FASE DE ALTO NIVEL (HLD)

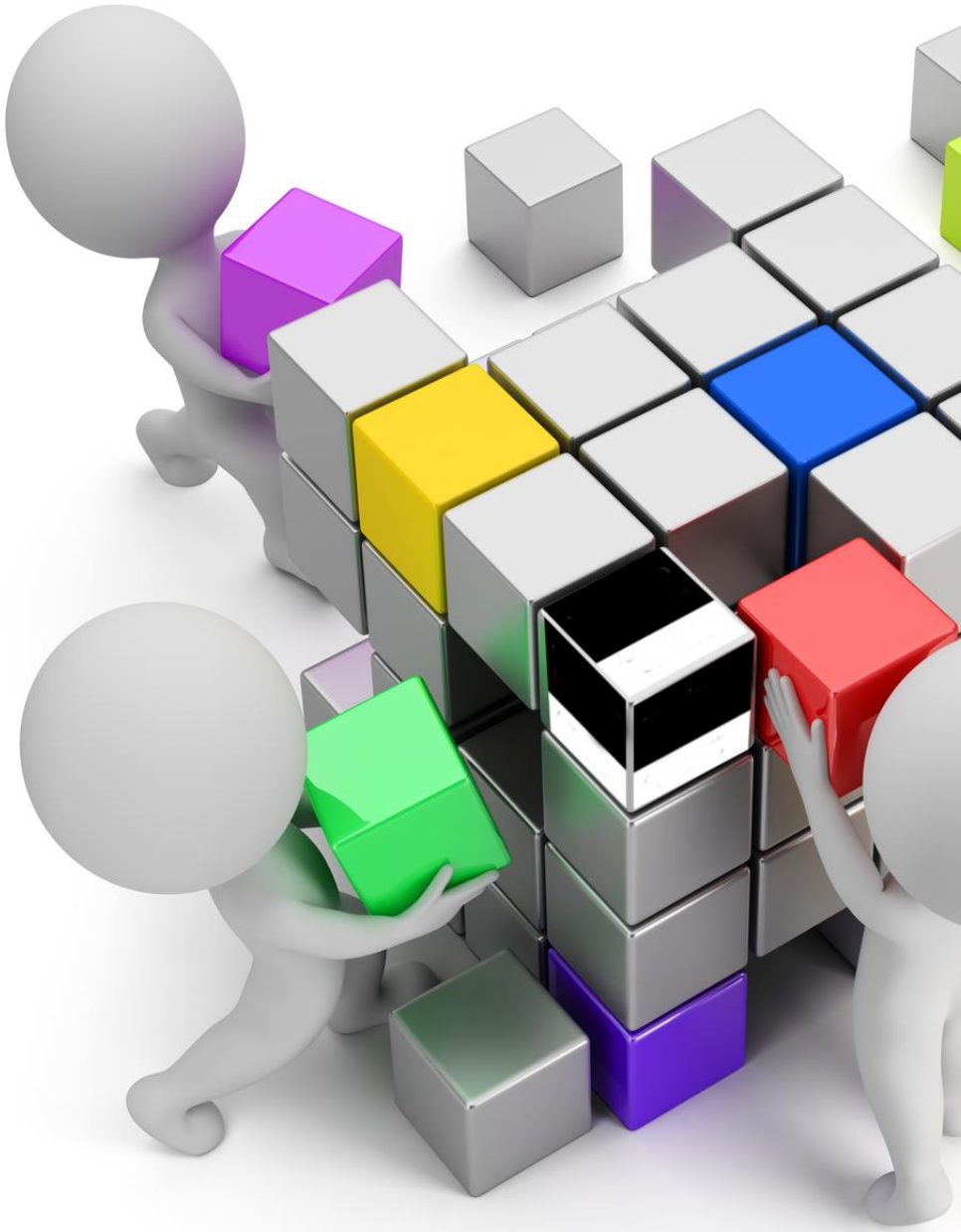
- Se centra en la Arquitectura y el Diseño del Sistema.
- Proporciona una visión general de la solución, plataforma, sistema, producto y servicio / proceso.
- En esta etapa es necesario crear un Plan de Testeo de Integración para probar la integración o las interfaces entre componentes.

FASE DE BAJO NIVEL (LLD)

- Se diseñan los componentes reales del software.
- Define la lógica real para cada componente del sistema.
- Incluye un Diagrama de Clases con todos los métodos y la relación entre las clases.
- Se planifican las Pruebas de Componentes.

FASE DE IMPLEMENTACIÓN

- Es donde toda la codificación se lleva a cabo.
- Una vez que se completa la codificación, la ruta de ejecución continúa en el lado derecho de la V donde ahora se ponen en uso los Planes de Prueba desarrollados anteriormente.



FASE DE CODIFICACIÓN

- Está en la parte inferior del modelo V.
- El Diseño del módulo es convertido en código.
- Las Pruebas de Unidad son realizadas por los desarrolladores en su código.

VENTAJAS

- Simple y fácil de usar.
- Las actividades de Prueba como la Planificación, el Diseño de las Pruebas, se realizan antes de la codificación. Esto ahorra mucho tiempo y conlleva a mayor probabilidad de éxito sobre el uso del Modelo en Cascada.
- Tiene un seguimiento proactivo de defectos, es decir, los defectos se encuentran en una etapa temprana.
- Evita el flujo descendente de los defectos.
- Funciona bien para proyectos pequeños donde los requisitos se entienden fácilmente.

DESVENTAJAS

- Simple y fácil de usar.
- Las actividades de Prueba como la Planificación, el Diseño de las Pruebas, se realizan antes de la codificación. Esto ahorra mucho tiempo y conlleva a mayor probabilidad de éxito sobre el uso del Modelo en Cascada.
- Tiene un seguimiento proactivo de defectos, es decir, los defectos se encuentran en una etapa temprana.
- Evita el flujo descendente de los defectos.
- Funciona bien para proyectos pequeños donde los requisitos se entienden fácilmente.

CUANDO SE DEBE USAR EL MODELO EN V?

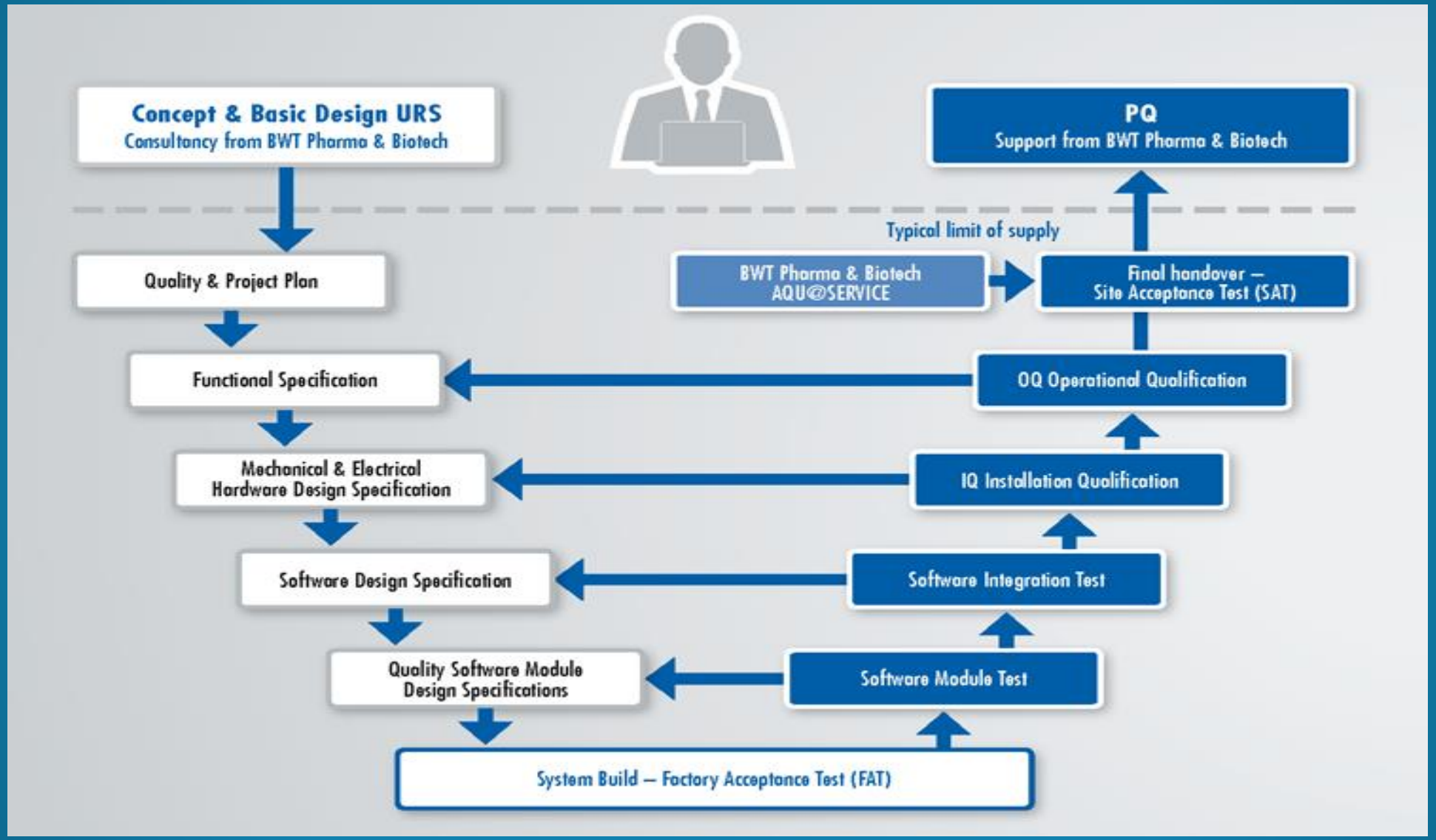
- El modelo V se debe utilizar para proyectos pequeños o medianos donde los requisitos están claramente definidos y son fijos.

- El modelo V se debe elegir cuando se dispone de recursos con amplia experiencia y expertise técnico.

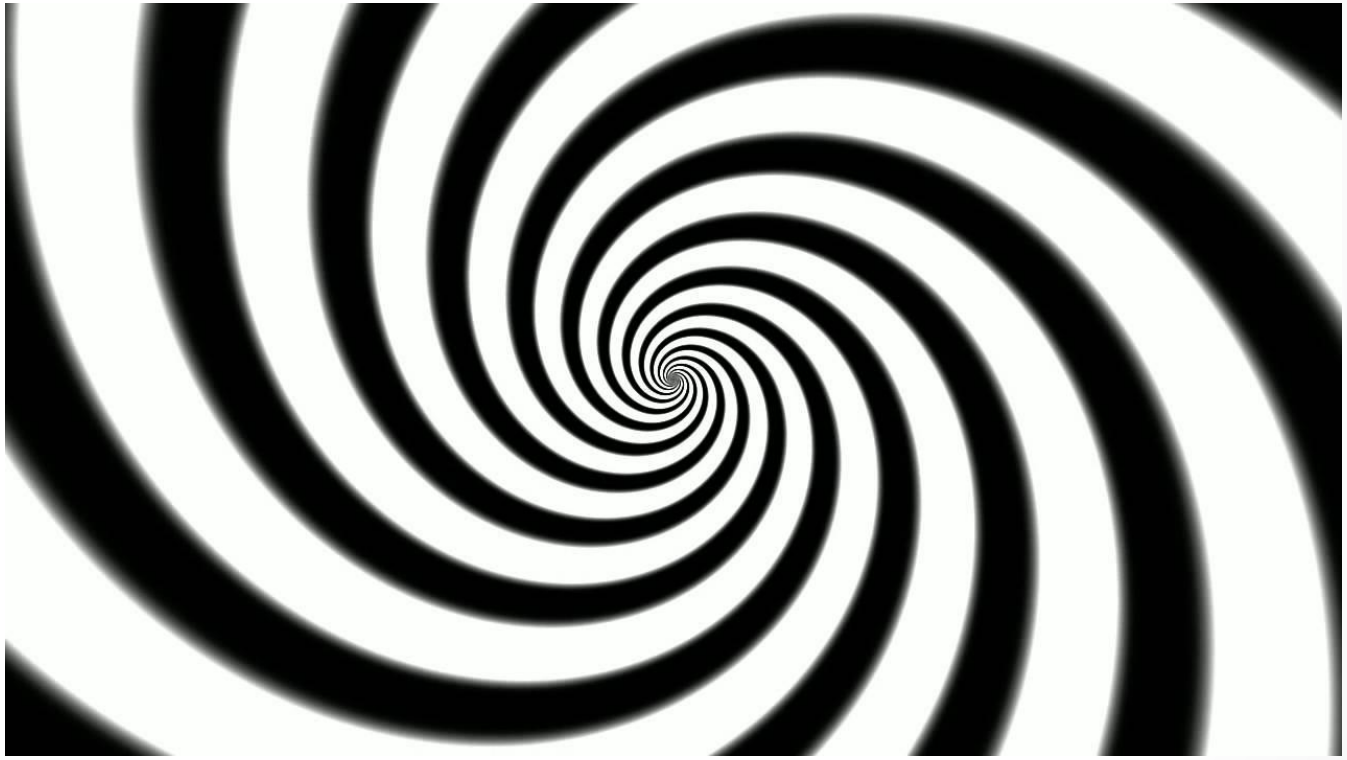
RECOMENDACIONES

- Se requiere una alta confianza del cliente para elegir el enfoque del modelo en forma de V.

- Al no producirse prototipos, existe un riesgo muy alto en el incumplimiento de las expectativas de los clientes.


Concept & Basic Design URS
Consultancy from BWT Pharma & Biotech
Quality & Project Plan
Functional Specification
Mechanical & Electrical Hardware Design Specification
Software Design Specification
Quality Software Module Design Specifications
System Build – Factory Acceptance Test (FAT)
Software Module Test
Software Integration Test
IQ Installation Qualification
OQ Operational Qualification
Final handover – Site Acceptance Test (SAT)
BWT Pharma & Biotech AQU@SERVICE
Typical limit of supply
PQ
Support from BWT Pharma & Biotech

Evolutivo: ESPIRAL



CARACTERISTICAS

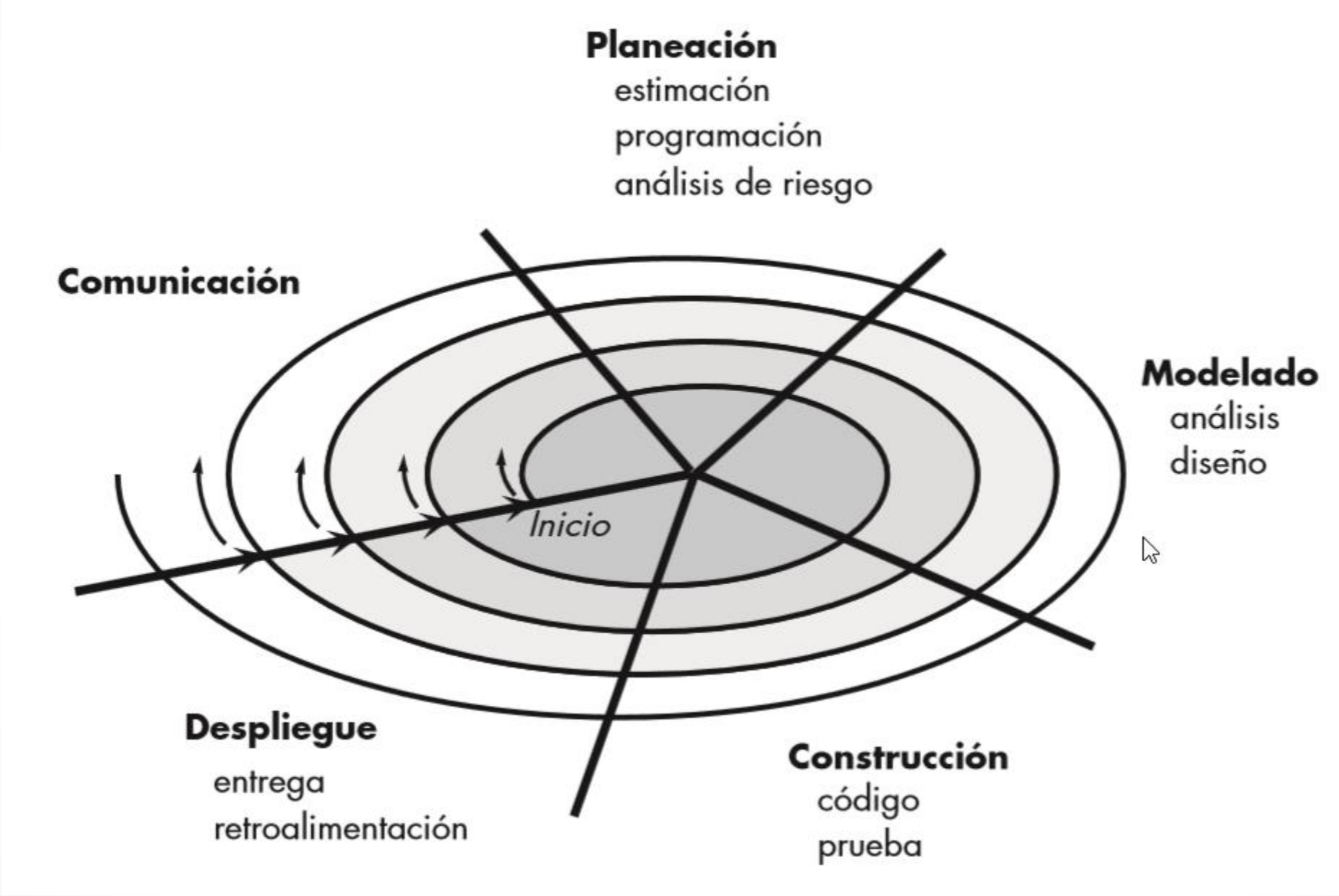
- ✓ **El Modelo en Espiral es similar al Modelo Incremental, con más énfasis en el Análisis de Riesgo.**
- ✓ **Este Modelo tiene cuatro fases: Planificación, Análisis de Riesgos, Ingeniería y Evaluación.**
- ✓ **Un proyecto de software pasa repetidamente por estas fases en iteraciones (llamadas Espirales).**

- ✓ En la espiral de línea de base, que comienza en la fase de Planificación, se reúnen los requisitos y se evalúa el riesgo.
- ✓ Cada espiral posterior se construye en la línea de base espiral.

PROCESO

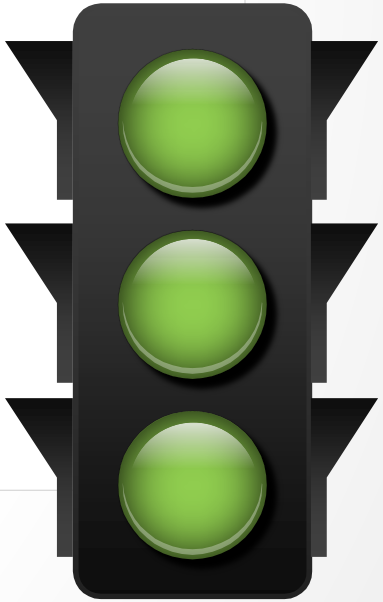
Fase de planificacion	Analisis de riesgo	Fase de ingenieria	Fase de evaluacion
Los requisitos se toman durante la fase de planificación (BRS y SRS).	Se lleva a cabo un proceso para identificar riesgos y soluciones alternativas. Se produce un prototipo al final de la fase. Si se encuentra algún riesgo durante el análisis de riesgo, se sugieren e implementan soluciones alternativas	En esta fase se desarrolla el software, junto con las pruebas al final de la fase. Por ello es que en esta fase se realiza el desarrollo y las pruebas.	Esta fase le permite al cliente evaluar el lanzamiento del proyecto hasta la fecha antes de que el proyecto continúe en la próxima espiral.

DIAGRAMA DEL MODELO ESPIRAL



VENTAJAS

- Debido a la gran cantidad de Análisis de Riesgo , se mejora la probabilidad de riesgo.
- Bueno para proyectos grandes y de misión crítica.
- Funcionalidad adicional puede ser agregada a futuro (fecha posterior).
- El software se produce al principio del ciclo de vida del software.



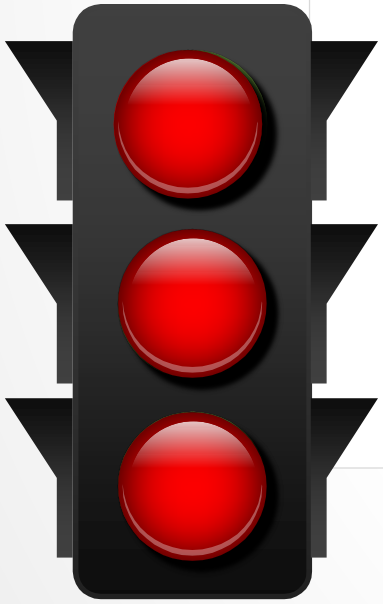
DESVENTAJAS

Puede ser un modelo costoso de usar.

El Análisis de Riesgos requiere una experiencia altamente específica.

El éxito del proyecto depende en gran medida de la fase de análisis de riesgos.

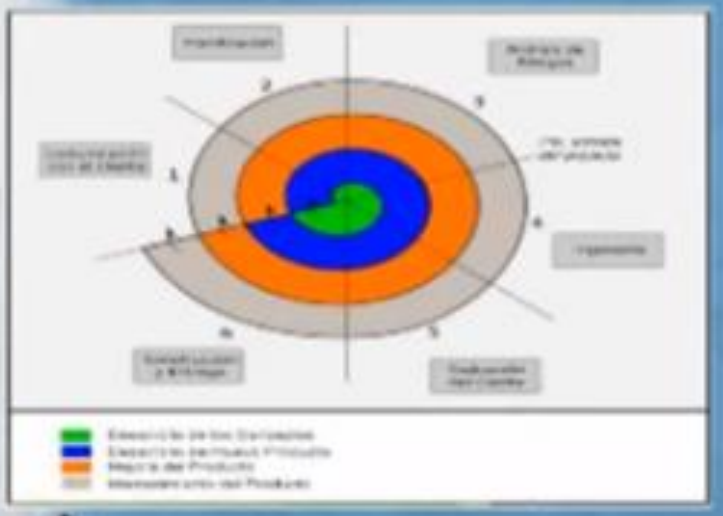
Puede ser un modelo costoso de usar.



CUANDO USAR EL MODELO ESPIRAL?

- Cuando los costos y la evaluación de riesgos son importantes.
- Para proyectos de riesgo medio a alto.
- Compromiso a largo plazo del proyecto imprudente debido a posibles cambios en las prioridades económicas
- Los usuarios no están seguros de sus necesidades.
- Los requisitos son complejos.
- Se esperan cambios significativos en el producto (investigación y exploración).

MODELO ESPIRAL



0:04 / 4:24

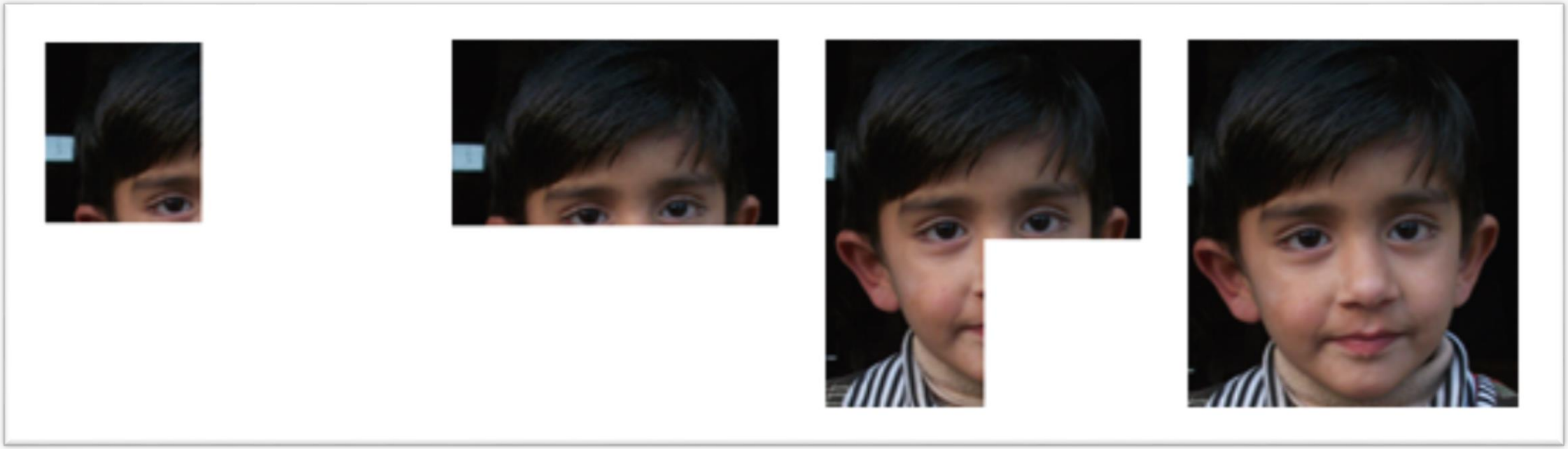

Incremental build model
increment
long
agile
philosophy
utilization
delivered
designed
software
component
system
element
prototyping
finishedsatisfies
iterative
functionality
termed
helps
development
outlay
traumatic
introducing
exceed
decomposed
regression
evolution
time
involves
incremental
separately
modeling
iteration
complete
waterfall
reuse
integration
added
debug
process
implemented
subsequent
incrementally
problems
rigorous

CARACTERISTICAS

- ✓ **En el Modelo Incremental, todo el requisito se divide en varias construcciones.**
- ✓ **Es un Modelo que utiliza múltiples Ciclos de Desarrollo, lo que hace que el Ciclo de Vida sea un ciclo de "Cascada Múltiple".**
- ✓ **Los Ciclos son divididos en módulos más pequeños y más fáciles de gestionar.**

- ✓ En este Modelo, cada Módulo pasa por las Fases de Requisitos, Diseño, Implementación y Prueba.
- ✓ Se produce una versión del trabajo de software realizado durante el primer módulo, por lo que desde un principio se tiene un software de trabajo durante el ciclo de vida del software.
- ✓ Cada versión posterior del módulo agrega funcionalidad a la versión anterior.
- ✓ El proceso continúa hasta que se logra el sistema completo.

EJEMPLO



INCREMENTO 1 → INCREMENTO 2 → INCREMENTO 3 → INCREMENTO 4

Cuando se trabaja con el Modelo Incremental, se agrega pieza por pieza, pero se espera que cada pieza esté completamente terminada. Así se puede continuar añadiendo las piezas hasta que se complete el producto.

EJEMPLO

1 INCREMENTO 1

Se tiene un producto pensado

2 INCREMENTO 2

Se empieza a construirlo y, en la primera iteración, el primer módulo de la aplicación/producto está totalmente listo y se puede mostrar a los clientes.

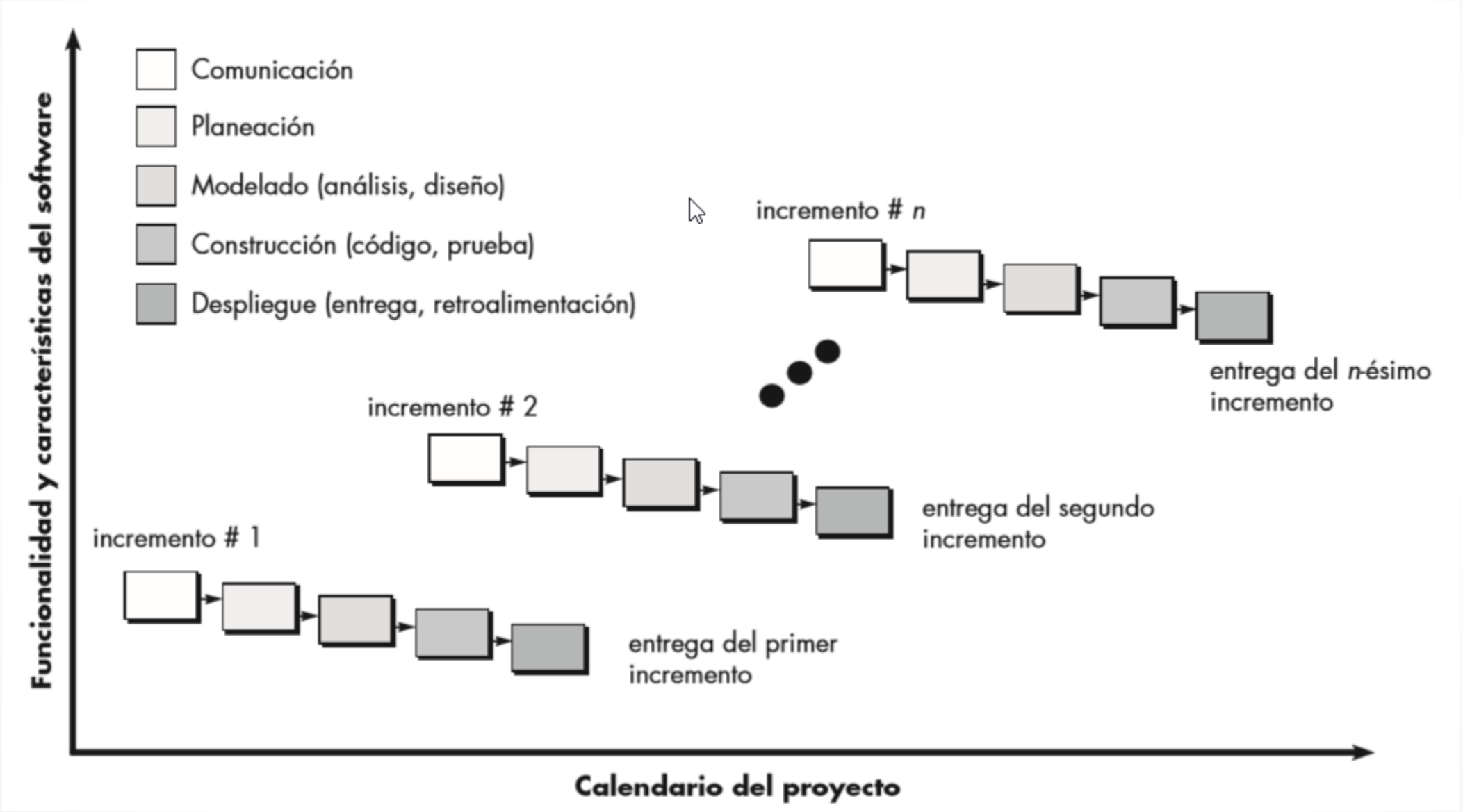
3 INCREMENTO 3

Del mismo modo, en la segunda iteración, el otro módulo está listo e integrado con el primer módulo.

4 INCREMENTO 4

En la tercera iteración todo el producto está listo e integrado.

DIAGRAMA DEL MODELO INCREMENTAL



VENTAJAS

- Genera software de trabajo (working software) de forma rápida y temprana durante el ciclo de vida del software..
- Este modelo es más flexible à menos costoso para cambiar el alcance y los requisitos.
- Es más fácil probar y depurar una iteración pequeña y reduce el costo de entrega inicial.
- Más fácil de administrar el riesgo porque las piezas de riesgo se identifican y manejan durante su iteración.

DESVENTAJAS

Necesita buena Planificación y Diseño.

Necesita una definición clara y completa de todo el sistema antes de poder descomponerlo y construirlo de manera incremental.

El costo total es más elevado que al utilizar el Modelo en Cascada.

CUANDO USAR EL MODELO INCREMENTAL ?

- Este modelo se puede utilizar cuando los requisitos del sistema están claramente definidos y comprendidos.
- Los requisitos principales deben ser definidos; Sin embargo, algunos detalles pueden evolucionar con el tiempo.
- Cuando existe una necesidad de ver la factibilidad temprana del producto.
- Se está utilizando una nueva tecnología.
- Hay algunas características y objetivos de alto riesgo.


ITERATIVE AND INCREMENTAL
We just saw two of the three distinguishing aspects of
0:01 / 1:55


requirements
flowchart
programming
deployment
project
methodology
business
work
engineering
diagram
iterative
cycle
model
agile
analysis
incremental
sprint
development
integration
software
plan
scrum
design
system
method
lifecycle
web
flow
team
agility
data
life
process
strategy
information
management
implementation
quality
meeting

CARACTERISTICAS

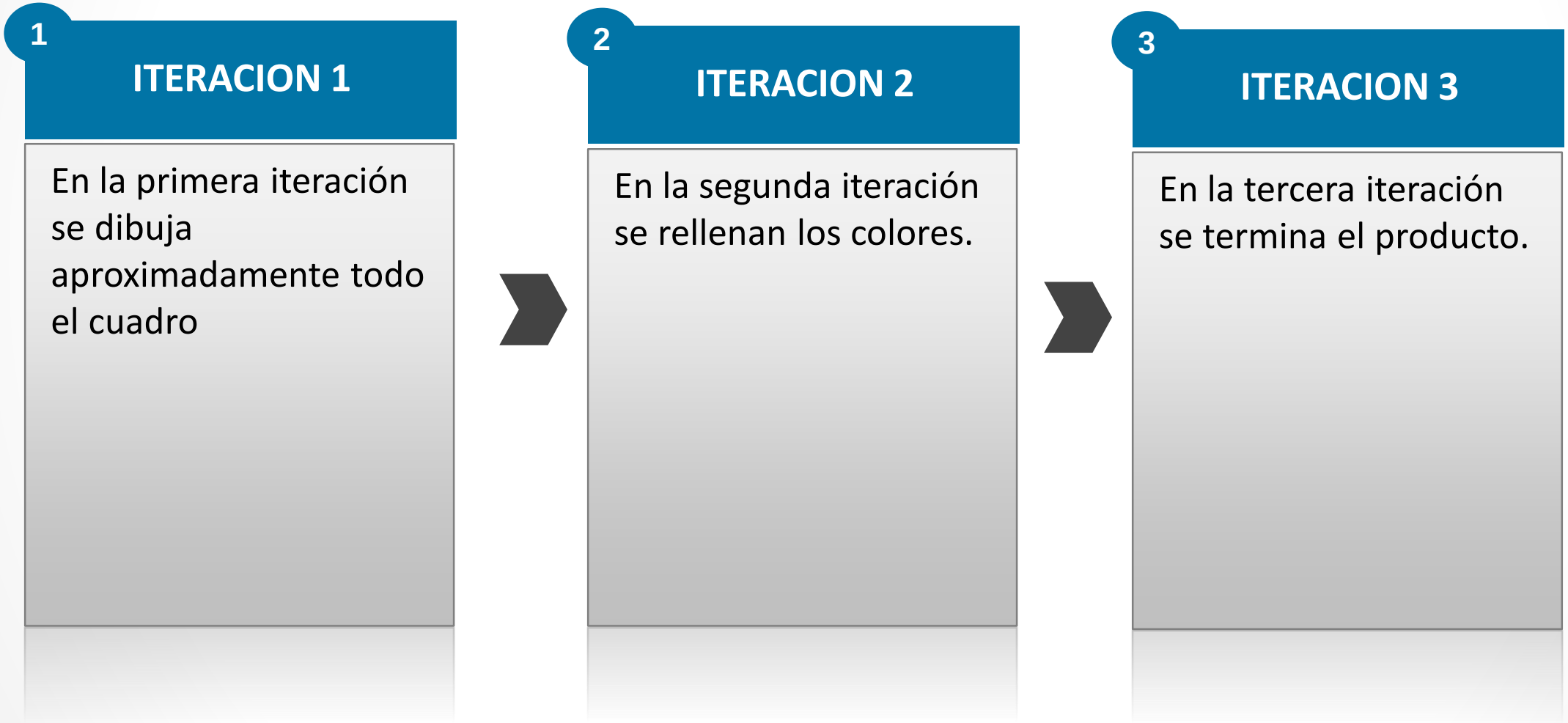
Un modelo de ciclo de vida iterativo no intenta comenzar con una especificación completa de los requisitos. En su lugar, el desarrollo comienza especificando e implementando solo una parte del software, que luego se puede revisar para identificar otros requisitos.

Este proceso luego se repite, produciendo una nueva versión del software para cada ciclo del modelo.

EJEMPLO

IDEA → ITERACION 1 → ITERACION 2 → ITERACION 3

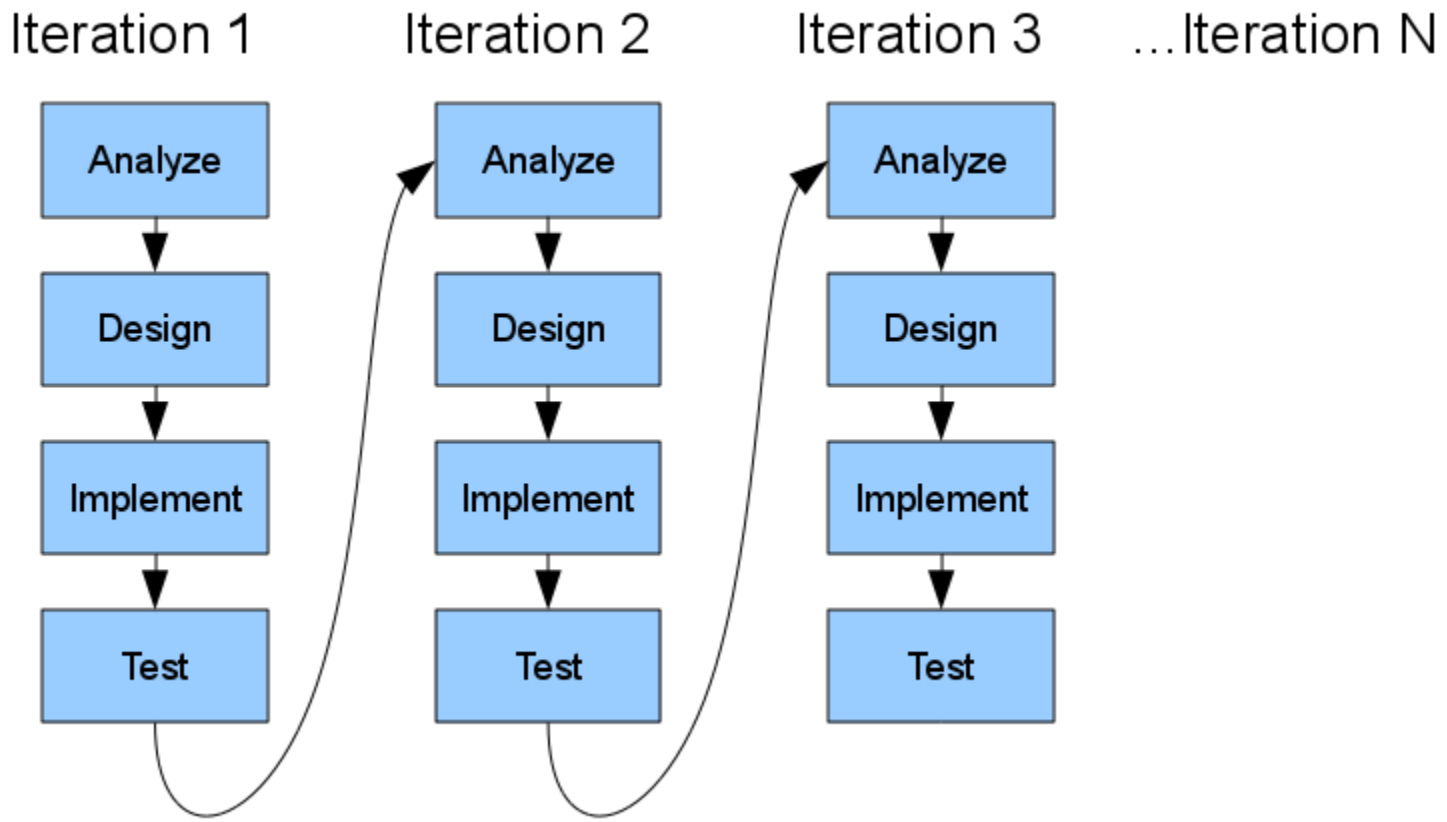
COMO SE VE EN LA IMAGEN :



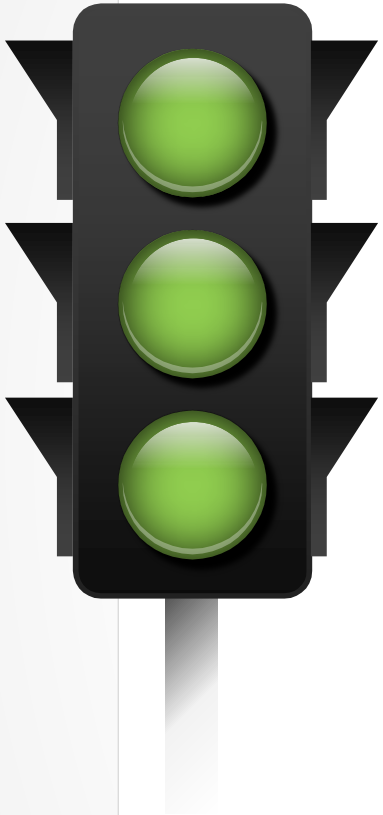
CUANDO TRABAJAMOS DE FORMA INCREMENTAL :

- ❑ Creamos un producto en bruto o pieza de producto en una iteración.
- ❑ Luego lo revisamos y lo mejoramos en la siguiente iteración.
- ❑ Y así sucesivamente hasta que finalice.

DIAGRAMA DEL MODELO ITERATIVO



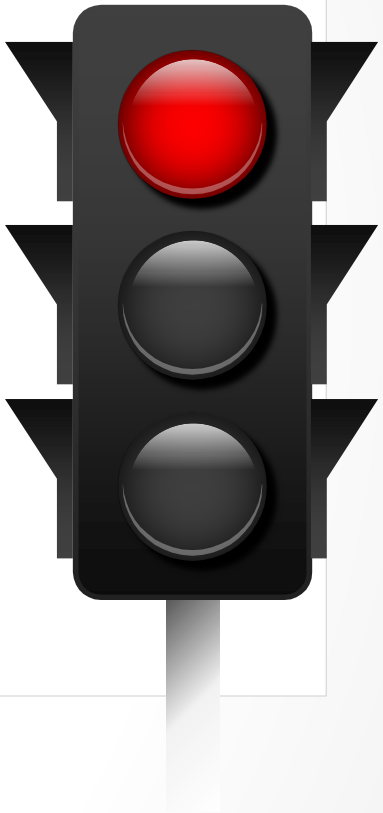
VENTAJAS



- En el modelo iterativo, solo podemos crear un diseño de alto nivel de la aplicación antes de que realmente comencemos a construir el producto y definamos la solución de diseño para todo el producto. Más adelante, podemos diseñar y construir una versión esquelética de eso, y luego evolucionar el diseño basado en lo que se había construido.
- En el modelo iterativo estamos construyendo y mejorando el producto paso a paso. Por lo tanto podemos rastrear los defectos en las primeras etapas. Esto evita el flujo descendente de los defectos.
- En el modelo iterativo podemos obtener la retroalimentación confiable de los usuarios. Cuando presentamos bocetos y planos del producto a los usuarios para sus comentarios, les pedimos que se imaginen cómo funcionará el producto.
- En el modelo iterativo, se dedica menos tiempo a la documentación y se da más tiempo para el diseño.

DESVENTAJAS

- Cada fase de una iteración es rígida sin solapamientos.
- Problemas de diseño o sistemas de arquitectura costosos.
- pueden surgir debido a que no se tienen todos los requisitos.



USAR EL MODELO ITERATIVO CUANDO…

- Los requisitos de todo el sistema están completos, claramente definidos y comprendidos.
- Cuando el proyecto es grande.
- Los requisitos principales deben ser definidos, Sin embargo, algunos detalles pueden evolucionar con el tiempo.


ITERATIVE
MODEL
PMPLOUNGE.COM
0:03 / 5:43


MANAGEMENT
VARY
CAPTURE
WORKFLOWS
SCOPE
DIAGRAMS
ARCHITECTURE
INCLUDE
SOFTWARE
IMPLEMENTATION
INTERCHANGEABLY
FUNCTIONALITY
TEAM
ARCHITECTURAL
NOTATION
DEVELOPMENT
ARCHITECTURALLY
SYSTEM
ITERATION
DESCRIBE
BASELINE
ITERATIVE
EXECUTABLE
UNIFIED PROCESS
DISCIPLINES
RELATIVE
DELIVERABLE
RISK
STABILIZED
SIGNIFICANT
CASE
COST
REDUCING
PHASE
ARTIFACTS
PARTIAL
PUBLISHED
GOALS
TRADEOFFS
INCREMENT
SIMPLIFY
ILLUSTRATING
REFINEMENT
SERIES
DIVIDED
EXPECTED
CONCEPTUAL
INCREMENTAL
FOUNDATION
TYPICAL
EXTENSIBLE
SCHEDULE
AUTHORS
FINAL
SCALABILITY
INITIAL
VARIATIONS
EMPHASIS
BUILT
REQUIREMENTS
NUMBER
DELIVERABLES
SPECIFICATION

CARACTERISTICAS

El proceso Unificado es un intento por obtener los mejores rasgos y características de los modelos tradicionales.
Reconoce la importancia de la comunicación con el cliente y los métodos directos para describir su punto de vista respecto de un sistema (el caso de uso).
Hace énfasis en la importancia de la arquitectura del software y “ayuda a que el arquitecto se centre en las metas correctas (que sea comprensible, permita cambios futuros y la reutilización).

Sugiere un flujo del proceso iterativo e incremental, lo que da la sensación evolutiva que resulta esencial en el desarrollo moderno del software.

El modelo iterativo e incremental propuesto por el PU puede y debe adaptarse para que satisfaga necesidades específicas del proyecto.

FASES DEL PROCESO UNIFICADO

FASE INICIACION

- Agrupa actividades de comunicación y planeación con el cliente.
- Se identifican los requisitos del negocio, se describen un conjunto de Casos de Uso.
- Se propone una arquitectura y se desarrolla un plan para la naturaleza Iterativa e Incremental del proyecto.
- Se identifica recursos, evalúa los riesgos principales, define un programa de actividades y establece una base de avance del Incremento del software.

FASE DE ELABORACION

- Modelado del modelo general del proceso.
- Se mejora y amplía los Casos de Uso preliminares (Fase anterior).
- Aumenta la representación de la arquitectura para incluir: los modelos del Caso de Uso, de Requerimientos, del Diseño, de la Implementación y del Despliegue.
- Al terminar esta fase se revisa con cuidado el plan para asegurar que el alcance, riesgos y fechas de entrega siguen siendo razonables.
- Es frecuente que en esta etapa se hagan modificaciones al plan.

FASES DEL PROCESO UNIFICADO

FASE DE CONSTRUCCION

- Teniendo el modelo de arquitectura, se desarrolla o adquiere los componentes del software.
- Se completan los modelos de requerimientos y diseño (comenzados en la fase de elaboración). Para que reflejen la versión final del incremento.
- Se implementa el código fuente.
- Se diseñan y efectúan pruebas unitarias.
- Se realizan actividades de integración (ensamble de componentes y pruebas de integración). Se emplean casos de uso para obtener las pruebas de aceptación.

FASE DE TRANSICION

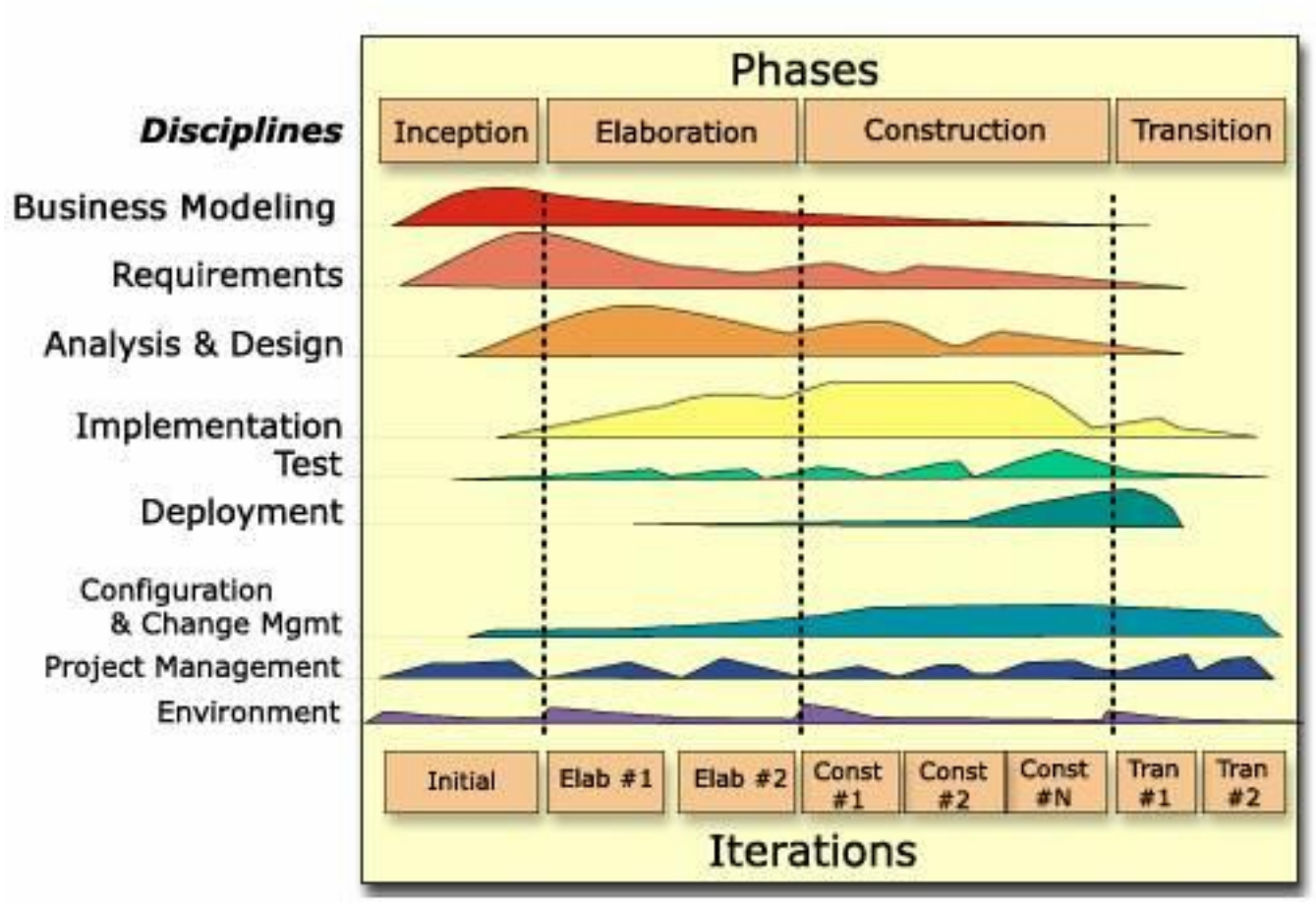
Del PU incluye las últimas etapas de la actividad general de construcción y la primera parte de la actividad de despliegue general (entrega y retroalimentación). Se da el software a los usuarios finales para las pruebas beta, quienes reportan tanto los defectos como los cambios necesarios. Además, el equipo de software genera la información de apoyo necesaria (por ejemplo, manuales de usuario, guías de solución de problemas, procedimientos de instalación, etc.) que se requiere para el lanzamiento. Al finalizar la fase de transición, el software incrementado se convierte en un producto utilizable que se lanza.

FASES DEL PROCESO UNIFICADO

FASE DE PRODUCCION

- Se brinda apoyo para el ambiente de operación (infraestructura).
- Se reportan defectos y solicitudes de cambio para su evaluación.
- Al mismo tiempo que se llevan a cabo las fases de construcción, transición y producción, comienza el trabajo sobre el siguiente incremento.Las cinco fases del PU no ocurren en secuencia sino que concurren en forma escalonada.
- El equipo adapta el proceso (acciones, tareas, subtareas y productos del trabajo) a fin de que cumpla sus necesidades.

DIGRAMA DEL PROCESO UNIFICADO



VENTAJAS

- Acelera el ritmo de desarrollo.
- Se adapta mejor a las necesidades del cliente.
- Al ser un proceso Iterativo e Incremental, da una sensación evolutiva.
- Se adapta a satisfacer las necesidades específicas de un proyecto.


RATIONAL UNIFIED PROCESS (RUP)
There are two more software process models that I want to cover, so
0:01 / 1:25