

# Proceso Iterativo - Incremental

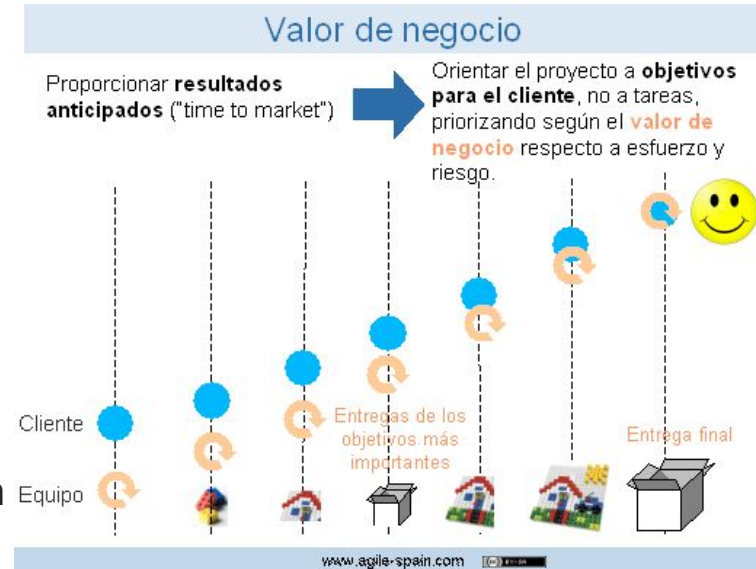
---

# Proceso iterativo/incremental

Se trata de un conjunto de tareas agrupadas en pequeñas etapas repetitivas (iteraciones), se relaciona con estrategias de desarrollo adaptativo.

El modelo consta de diversas etapas de desarrollo en cada incremento, se considera una buena práctica de ingeniería de software.

En un ciclo de vida iterativo e incremental **se van liberando partes del producto periódicamente**, en cada iteración, y cada nueva versión, normalmente, aumenta la funcionalidad y mejora en calidad respecto a la anterior.

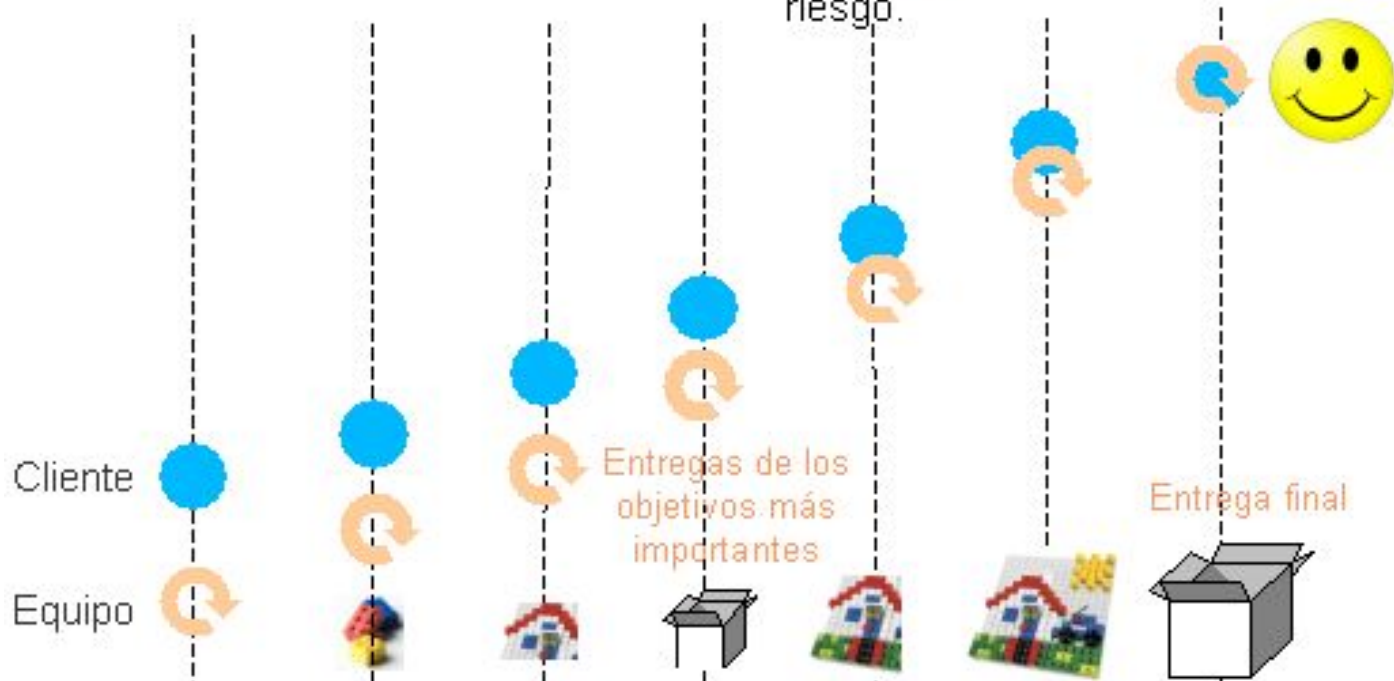


# Valor de negocio

Proporcionar **resultados anticipados** ("time to market")



Orientar el proyecto a **objetivos para el cliente**, no a tareas, priorizando según el **valor de negocio** respecto a esfuerzo y riesgo.

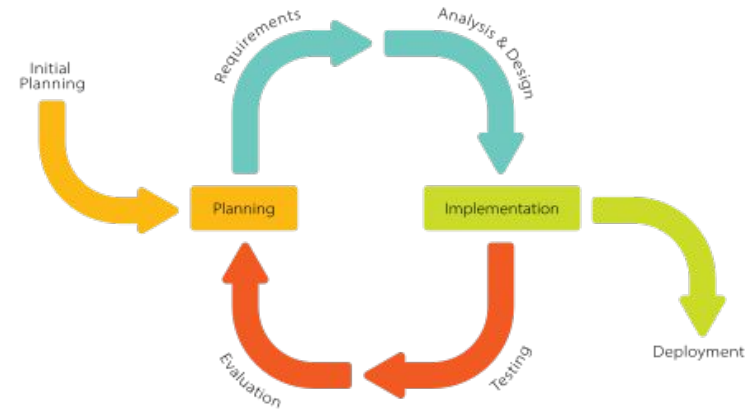


# Origen

El proceso iterativo/incremental es creado en respuesta a las debilidades del modelo tradicional de cascada.

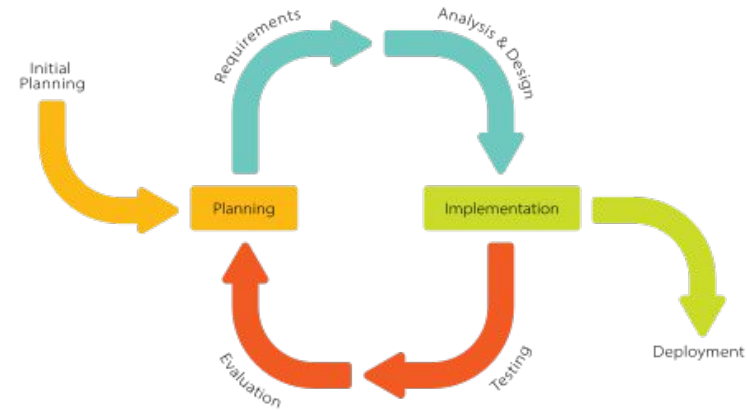
La utilización del término apareció primero en desarrollo de software, combinando los términos *iterativo* e *incremental*, que hasta entonces habían sido ampliamente sugeridos para grandes procesos de desarrollo.

En el año 1985, un estándar del ministerio de defensa de EE.UU mencionó que: “durante el desarrollo de software, más de una iteración del proceso de desarrollo puede llevarse a cabo al mismo tiempo.”.



# Origen

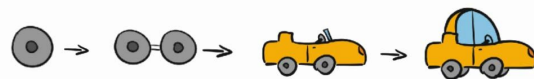
Sin embargo, el ciclo de vida iterativo e incremental es una de las buenas prácticas de ingeniería del software más antiguas, su primer uso en el software se data de los años 50. Además, el ciclo de vida iterativo e incremental es una de las bases de un proyecto basado en metodologías ágiles, más concretamente, con iteraciones cortas en tiempo, de pocas semanas, normalmente un mes y raramente más de dos.



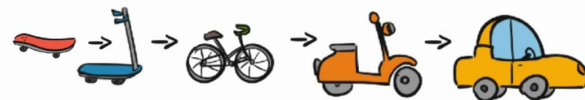
# ¿Qué es iterativo y qué es incremental?

La parte iterativa es simplemente una metodología de diseño que se basa en un proceso cíclico de testeo, prototipo, análisis y refinación de un producto o proceso, y basándose en el testeo de la última iteración de un diseño, se hacen los cambios o refinamientos necesarios.

Por otra parte, la parte incremental se la llama propiamente “proceso de construcción incremental” o “incremental build model”, e indica la implementación y testeo incrementales del proyecto hasta que es terminado. Incluye desarrollo y mantenimiento, y aplica el modelo de cascada incrementalmente.



MÓDELO ITERATIVO



MÓDELO ITERATIVO E INCREMENTAL

**Iterativo (espiral)**  
(orden de iteraciones  
basadas en riesgo)



- Ruedas
- Motor
- Aire acondicionado
- Radio
- Ventanas
- Baúl
- Asientos
- Puertas
- Volante y cambios
- Carcasa
- Parabrisas
- Luces

**Niveles de riesgo:**

1. Lo que más riesgo tiene es la carcasa del auto
- ...

1. Que podamos movernos (no a pie)
2. Que se mueva más rápido que velocidad a pie
3. Que podamos andar sentados
4. Que podamos hacer uso de un motor
5. Que podamos andar en la ruta
6. Que no tengamos calor (?)
7. Que podamos andar cuando haya lluvia

**Incremental**  
(orden de incrementos  
basados en funcionalidad)

- Ruedas y tabla 
- Monopatín 
- Asientos
- Frenos 
- Motor (scooter)
- Motor mejor
- Luces 
- (...el resto de las cosas...) 

- Mes 1: ya funciona el vehículo
- Mes 2: ya andamos más rápido que a pie
- Mes 4: ya podemos andar sentados
- Mes 5: ya podemos andar por la ciudad bastante rápido
- Mes 7: ya podemos andar por la ruta
- Mes 12: tengo el auto completo

- Mes 7: ya funciona el auto (tiene ruedas y motor)
- Mes 8: puedo andar sentado en el auto
- Mes 10: puedo andar por la calle
- Mes 12: tengo el auto completo

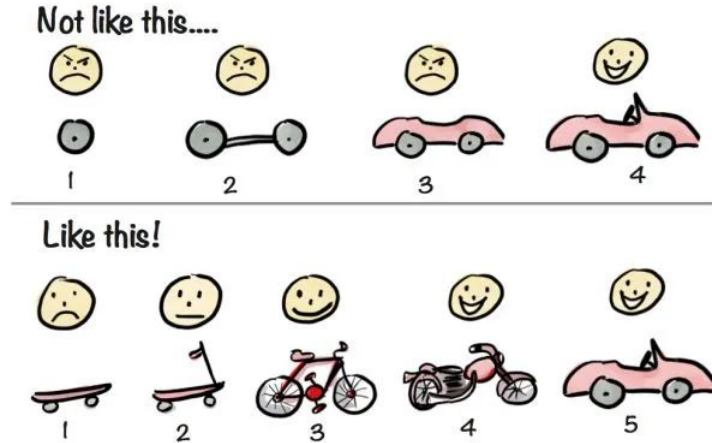


# Incremental

Este modelo aplica el proceso de cascada de una manera incremental.

La serie de lanzamientos o “releases”, se las refiere como “incrementos”, y cada incremento provee más funcionalidad al usuario.

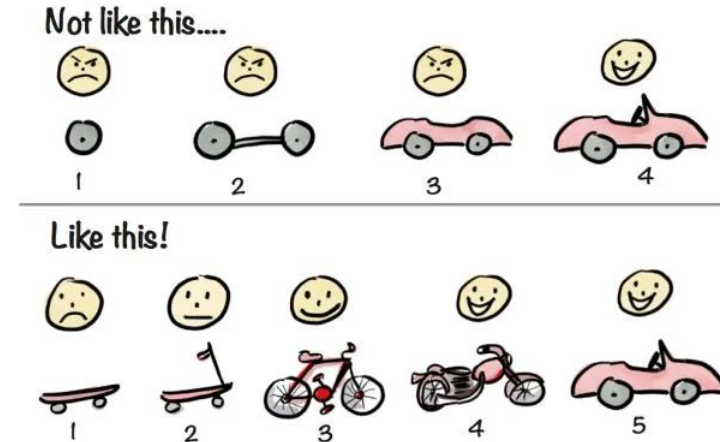
Luego del primer incremento, se entrega un “producto principal”, que ya puede ser usado por el usuario, y basándose en la retroalimentación del usuario, un plan es desarrollado para los siguientes incrementos, y se hacen modificaciones acorde a la retroalimentación.





# Etapas del modelo incremental

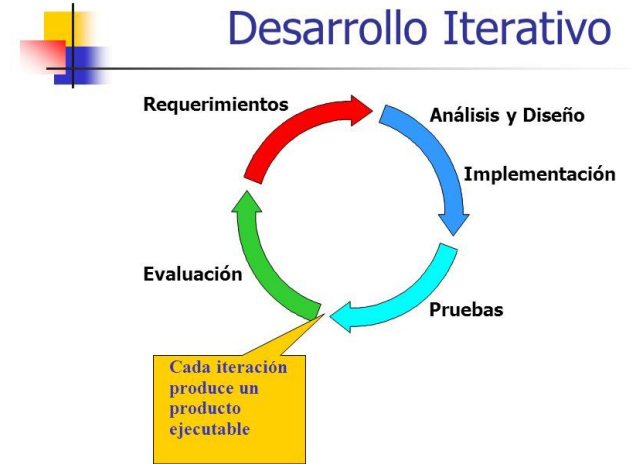
- Se descompone el sistema en varios mini proyectos de desarrollo.
- Se utilizan sistemas parciales para producir el sistema final.
- Se llevan a cabo primero los requerimientos de mayor prioridad.
- El requerimiento de una parte se congela una vez que se desarrolla la porción incrementada.



# Proceso iterativo/incremental

Es uno de los más utilizados en los últimos tiempos ya que, cómo se relaciona con estrategias de desarrollo adaptativo es empleado en metodologías diversas.

Lo que se busca es que en cada iteración los componentes logren evolucionar el producto dependiendo de los completados de las iteraciones antecesoras, agregando más opciones de requisitos y logrando así un mejoramiento mucho más completo.



# Construcción del proyecto con Proceso iterativo/incremental

Se planifica un proyecto en distintos bloques temporales que se le denominan iteración. En una iteración se repite un determinado proceso de trabajo que brinda un resultado más completo para un producto final, de forma que quien lo utilice reciba beneficios crecientes (prestaciones nuevas o más completas) cuando se avanza con el proyecto.

Lo que se busca es que en cada iteración, los componentes o módulos logren evolucionar al producto, teniendo como punto de partida a las iteraciones antecesoras, cumpliendo con los requisitos y objetivos, logrando así avances para obtener un producto mucho más completo.



# Construcción del proyecto con Proceso iterativo/incremental

Una forma de dirigir al proceso iterativo incremental de forma correcta es priorizar los objetivos y requerimientos en función del valor que ofrece al cliente.

El desarrollo incremental e iterativo es también una parte esencial del tipo de programación soportada por frameworks de desarrollo rápido de software.

La idea principal detrás del modelo iterativo es desarrollarlo de manera incremental, permitiéndole al desarrollador sacar ventaja de lo que se ha aprendido a lo largo del desarrollo anterior, incrementando versiones entregables del sistema.



# Premisas y etapas

El aprendizaje para la incrementación en la siguiente iteración viene de dos vertientes: el desarrollo del sistema, y su uso (mientras sea posible). La lógica básica para comenzar con este modelo plantea una implementación simple de los requerimientos del sistema, e iterativamente mejorar al sistema en una secuencia evolutiva de versiones, hasta que el sistema completo esté implementado. En cada iteración, se realizan cambios en el diseño y se agregan nuevas funcionalidades y capacidades al sistema.



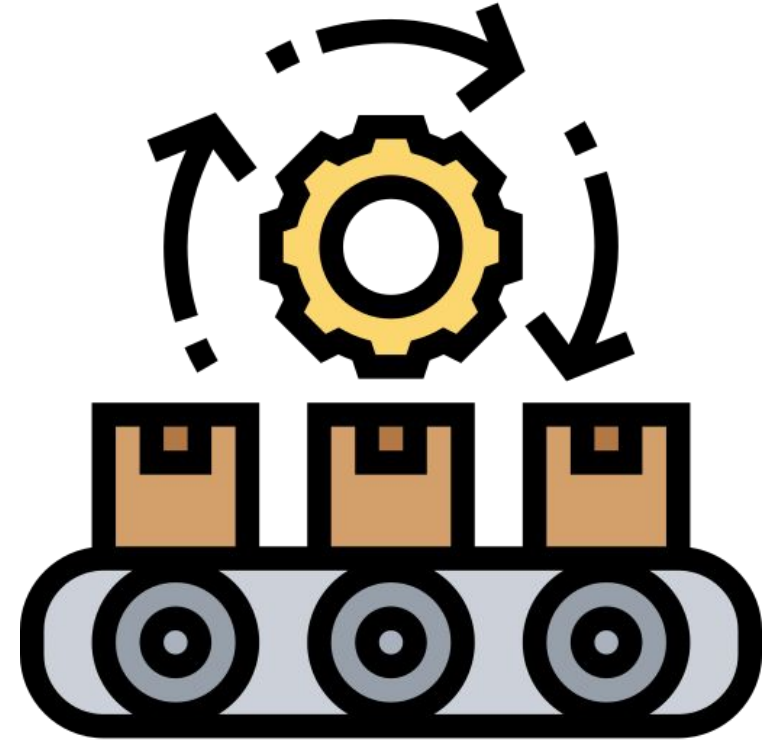
# Premisas y etapas

El modelo se basa en 2 premisas:

- Los usuarios nunca saben bien qué es lo que necesitan para satisfacer sus necesidades.
- En el desarrollo, los procesos tienden a cambiar.

A partir de las premisas, el proceso se compone por 3 etapas:

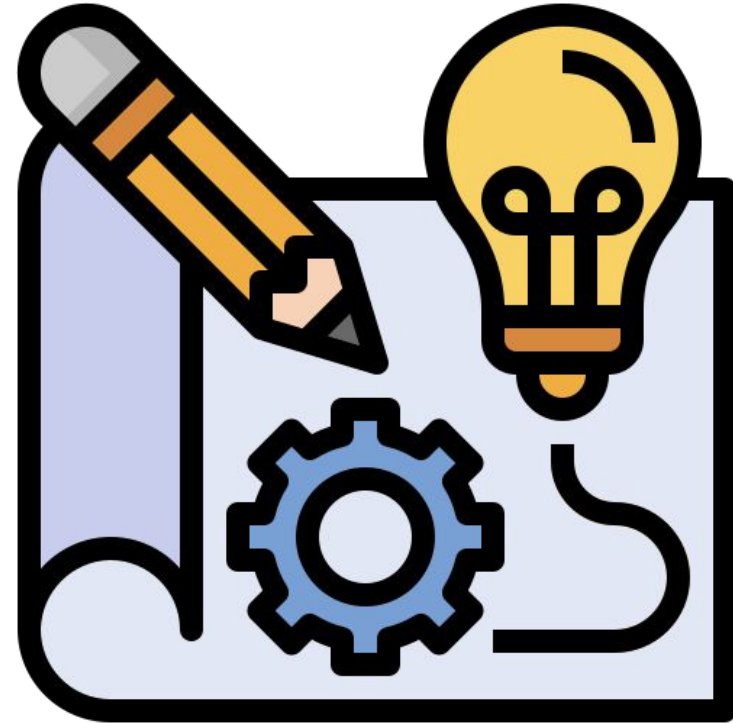
- Etapa de inicialización o creación.
- Etapa de iteración
- Etapa de control de proyecto.



# Etapa de creación

Se crea una versión del sistema. La meta de esta etapa es crear un producto con el que el usuario pueda interactuar, y por ende, retroalimentar el proceso. Debe ofrecer una muestra de los aspectos claves del problema y proveer una solución lo suficientemente simple para ser comprendida e implementada fácilmente.

Para guiar el proceso de iteración se crea una lista de control de proyecto, que contiene un historial de todas las tareas que se necesitan ser realizadas. Incluye cosas como nuevas funcionalidades para ser implementadas, y áreas de rediseño de la solución ya existente. Esta lista de control se revisa periódicamente.





# Etapa de iteración

Esta etapa involucra el rediseño e implementación de una tarea de la lista de control de proyecto, y el análisis de la versión más reciente del sistema. La meta del diseño e implementación de cualquier iteración es ser simple, directa y modular, para poder soportar el rediseño de la etapa o como una tarea añadida a la lista de control del proyecto.

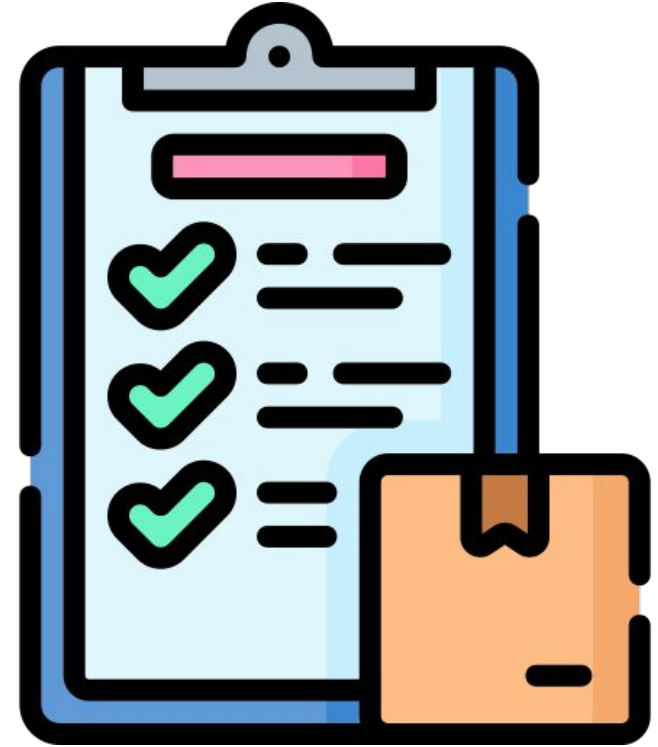
El código puede, en ciertos casos, representar la mayor fuente de documentación del sistema. El análisis de una iteración se basa en la retroalimentación del usuario y en el análisis de las funcionalidades disponibles del programa. Involucra el análisis de la estructura, modularidad, usabilidad, confiabilidad, eficiencia y eficacia (en resumen, alcanzar las metas). La lista de control del proyecto se modifica bajo la luz de los resultados del análisis.



# Listas de control del proyecto

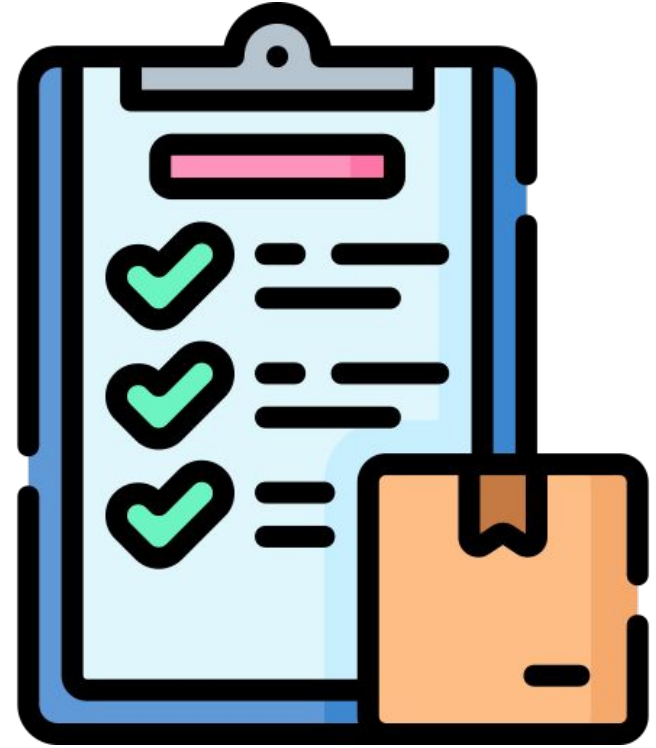
Las listas de control del proyecto que guían la implementación y el análisis incluyen:

- Cualquier dificultad en el diseño, codificación y prueba de una modificación que debería apuntar a la necesidad de rediseñar o recodificar.
- Las modificaciones deben ajustarse fácilmente a los módulos fáciles de encontrar y a los aislados (recordemos que el desarrollo es modular, y en algunos casos, los módulos, no están tan definidos). Si esto no se cumple, resultará conveniente analizar si se requiere algún grado de rediseño en la estructura del proyecto para aplicar las modificaciones necesarias de forma ordenada.
- Las modificaciones deben ser más fáciles de hacer conforme avanzan las iteraciones. Si no es así, hay un problema primordial usualmente basado en un diseño débil o en la proliferación excesiva de parches al sistema.



# Listas de control del proyecto

- Los parches normalmente deben permanecer solo por una o dos iteraciones. Se hacen necesarios para evitar el rediseño durante una fase de implementación.
- Es recomendable que la implementación existente sea analizada frecuentemente para determinar qué tal se ajusta a las metas del proyecto.
- Las facilidades para analizar el programa deben ser utilizadas cada vez para ayudar en el análisis de implementaciones parciales.
- La opinión del usuario debe ser solicitada y analizada para identificar deficiencias en la implementación.



## Incrementos



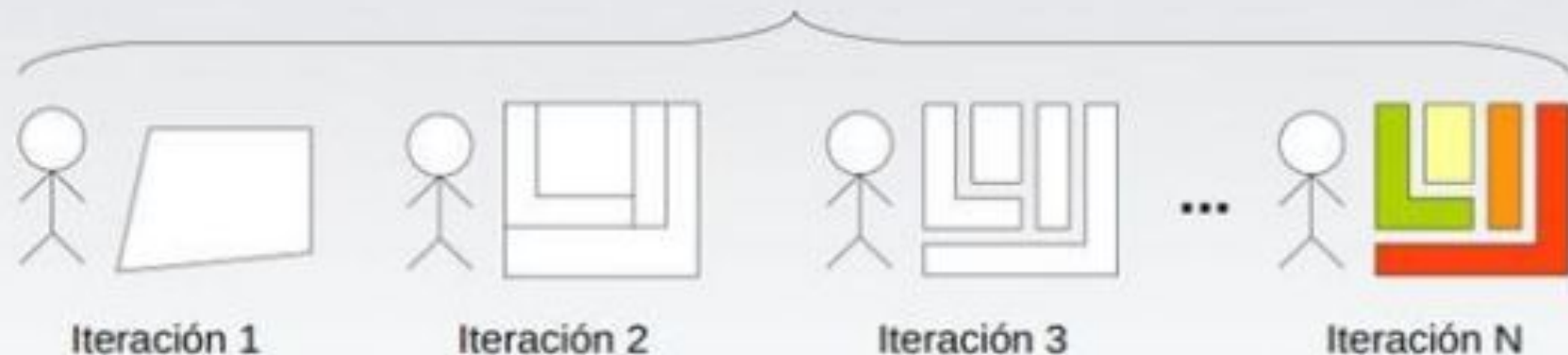
Las fases se dividen en incrementos, en cada incremento se desarrolla una parte de la funcionalidad y se validan los productos resultantes



Cliente

En general, una vez los productos de una fase se consideran listos, estos no se modifican más a lo largo de las siguientes fases

## Iteraciones



Cada iteración **refina** lo realizado en la iteración anterior. De esta forma se produce una dinámica en la que se van mejorando los productos (entregables) obtenidos en la iteración anterior. Eventualmente se realizarán todas las iteraciones planificadas, o se llegará al nivel de refinamiento deseado

# Ventajas del desarrollo incremental

- Los usuarios no tienen que esperar hasta que el sistema completo se entregue para hacer uso de él. El primer incremento cumple los requerimientos más importantes de tal forma que pueden utilizar el software al instante.
- Los usuarios pueden utilizar los incrementos como prototipos y obtener experiencia sobre los requerimientos de los incrementos posteriores del sistema.
- Existen muy pocas probabilidades de riesgo en el sistema. Aunque se pueden encontrar problemas en algunos incrementos, lo normal es que el sistema se entregue sin inconvenientes al usuario.



# Ventajas del desarrollo incremental

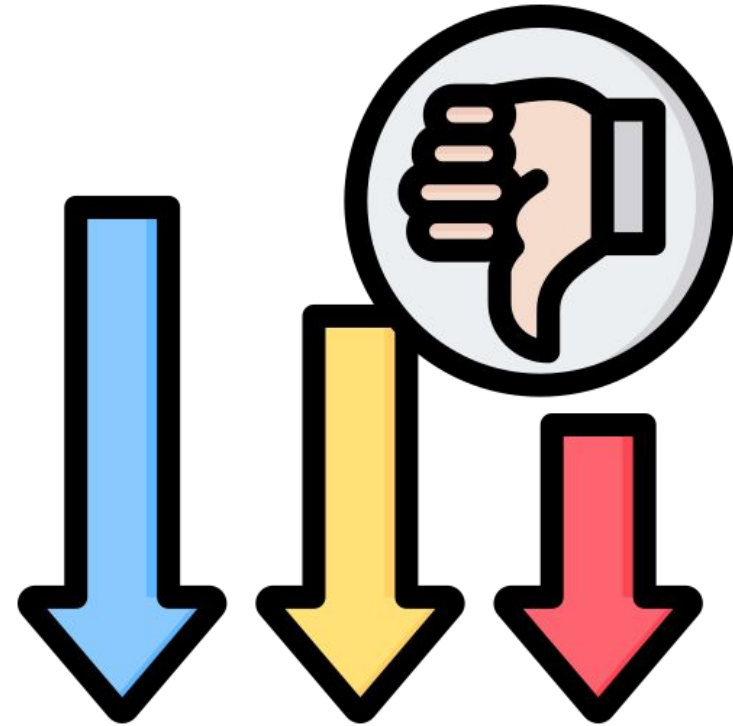
- Lo más importante del sistema se entrega primero, y los incrementos posteriores se integran. Esto quiere decir que es menos probable que los usuarios encuentren fallas de funcionamiento del software en las partes más importantes del sistema.
- En el desarrollo de este modelo se recibe retroalimentación de los usuarios desde el inicio de las iteraciones.
- Permite separar la complejidad del proyecto, gracias a su desarrollo por parte de cada iteración o bloque.
- El producto es consistente y puntual en el desarrollo.
- Los productos desarrollados con este modelo tienen una menor probabilidad de fallar.
- Se obtiene aprendizaje en cada iteración, que es aplicado en el desarrollo del producto.





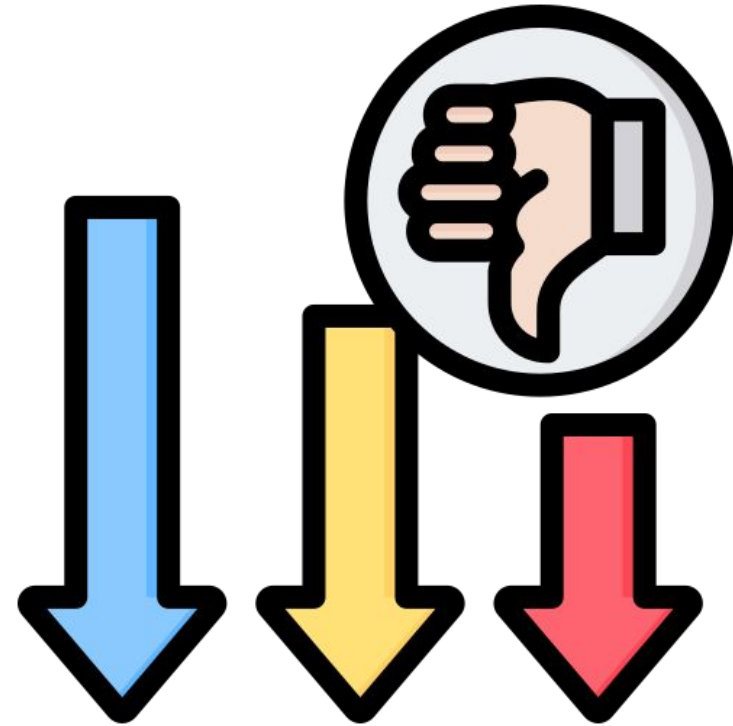
# Desventajas del desarrollo incremental

- La entrega temprana de los proyectos produce la creación de sistemas que pueden resultar simples y a veces se ven un poco monótonos a los ojos del personal que lo recibe.
- La mayoría de los incrementos se harán en base de las necesidades de los usuarios. Los incrementos en sí ya son estipulados desde antes de la entrega del proyecto, sin embargo, hay que ver cómo se maneja el producto por si necesita otros cambios además de los estipulados antes de la entrega del proyecto. Este problema no se ve frecuentemente ya que la mayoría de las veces los incrementos estipulados suplen satisfactoriamente al usuario.



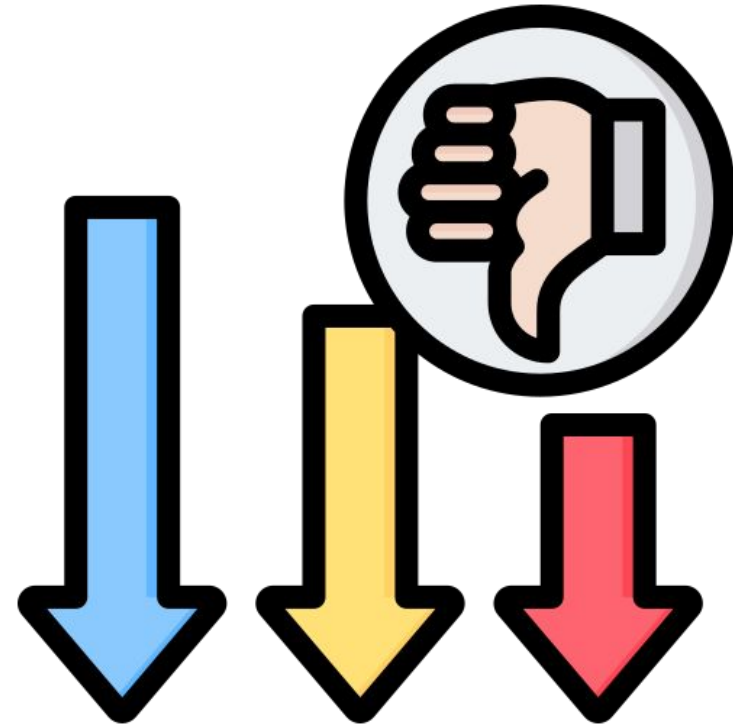
# Desventajas del desarrollo incremental

- Los incrementos no deben constar de muchas líneas de código ya que la idea de los incrementos es agregar accesorios al programa principal (o funcional), para que este tenga una y mil formas de desenvolverse en su tarea; llenar los incrementos de muchas líneas de código provocaría que se perdiera la objetividad o base de lo que se trata el desarrollo incremental. (no es una desventaja necesariamente).
- Requiere de un cliente involucrado durante todo el curso del proyecto. Hay clientes que simplemente no estarán dispuestos a invertir el tiempo necesario.
- El trato con el cliente debe basarse en principios éticos y de colaboración, más que trabajar cada parte independientemente, defendiendo sólo su propio beneficio.



# Desventajas del desarrollo incremental

- La entrega de un programa parcial pero funcional, puede hacerlo vulnerable debido a la falta de robustez y desarrollo, provocando que amenazas de seguridad puedan interferir con el correcto funcionamiento del proyecto.
- Infunde responsabilidad en el equipo de desarrollo al trabajar directamente con el cliente, requiriendo de profesionales con capacidades técnicas y de gestión sobre el promedio.



# ¿Para qué proyectos es adecuado?

- Proyectos donde los usuarios no saben bien qué es lo que necesitan para satisfacer sus necesidades.
- Proyectos Grandes – medianos - pequeños.
- Proyectos donde los procesos tienden a cambiar.

