

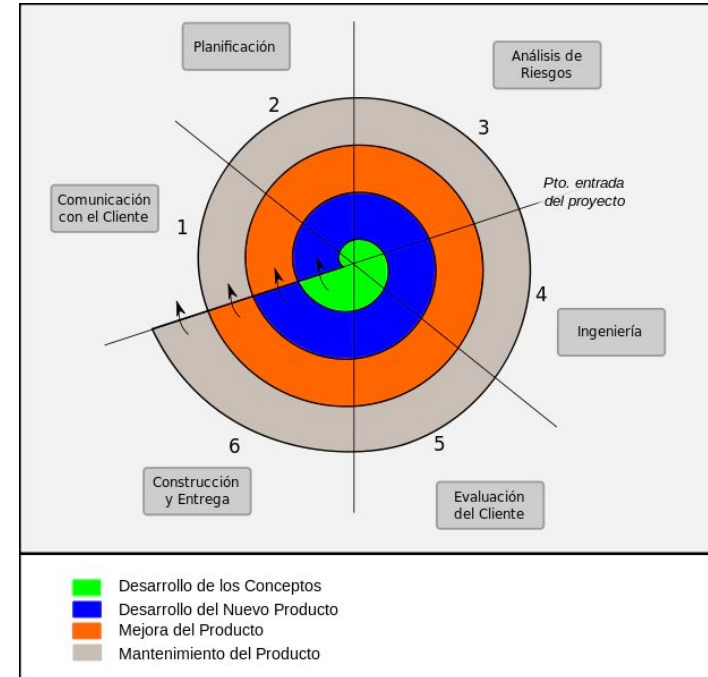
# Modelo Espiral

# Modelo en Espiral

Es un modelo ágil, su fuerte es la seguridad (prevención de riesgos), y adaptabilidad del proyecto, para resolver las problemáticas detectadas en los modelos rígidos.

Es una especie de combinación entre el modelo en cascada y el iterativo.

Cada bucle o iteración representa un conjunto de actividades. Las actividades no están fijadas a ninguna prioridad, sino que las siguientes se eligen en función del análisis de riesgo, comenzando por el bucle interior.



# Origen

Fue definido por primera vez en 1986 por Barry Boehm, con la idea de un enfoque distinto al tradicional en Cascada.

Su idea era tener en cuenta fuertemente el riesgo que aparece a la hora de desarrollar software. Para ello, se comienza mirando las posibles alternativas de desarrollo, se opta por la de riesgo más asumible y se hace un ciclo de la espiral.

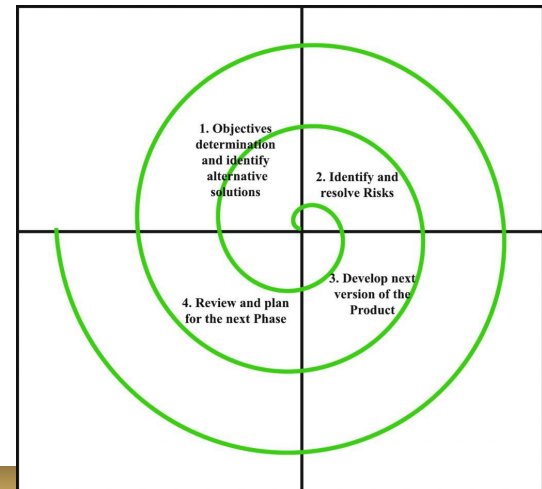
Si el cliente quiere seguir haciendo mejoras en el software, se vuelven a evaluar las distintas nuevas alternativas y riesgos y se realiza otra vuelta de la espiral, así hasta que llegue un momento en el que el producto software desarrollado sea aceptado y no necesite seguir mejorándolo con otro nuevo ciclo.



# Origen

Boehm describía el modelo en espiral como un “modelo de generación de procesos”, donde las elecciones basadas en los riesgos del proyecto generan un modelo de proceso apropiado/correcto para el proyecto.

De esta manera, los procesos en cascada, V, prototipo o incremental, etc., son casos especiales de modelos de proceso en espiral, que se ajustan al patrón de riesgo y la severidad del mismo de acuerdo al proyecto.



# Típicos conceptos erróneos

Boehm también identificó varios conceptos equivocados en gente que no está completamente familiarizada con el modelo:

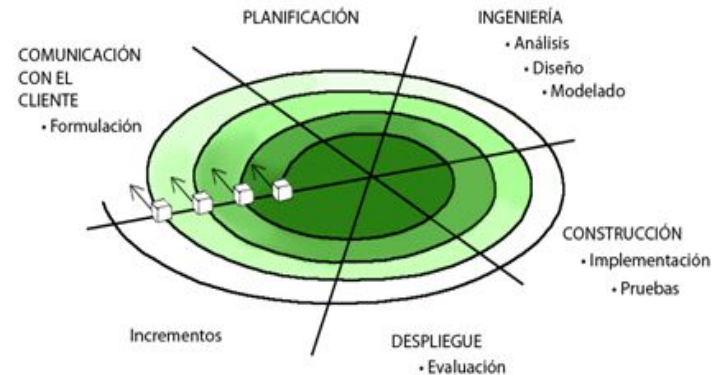
- La espiral **no es simplemente una secuencia incremental en cascada.**
- Todas las actividades del proyecto no siguen una secuencia única en espiral.
- Todas las actividades en el diagrama no necesariamente deben realizarse en el orden que aparecen.



# ¿Cómo se representa el modelo?

El Proceso es representado como una espiral en lugar de una secuencia de actividades con retrocesos, cada giro en la espiral representa una fase en el proceso, no hay fases fijas tales como especificación o diseño (se gira en la espiral dependiendo de qué se requiere).

Los Riesgos son explícitamente identificados y resueltos durante el proceso. Para cada actividad habrá (mínimo) cuatro tareas.

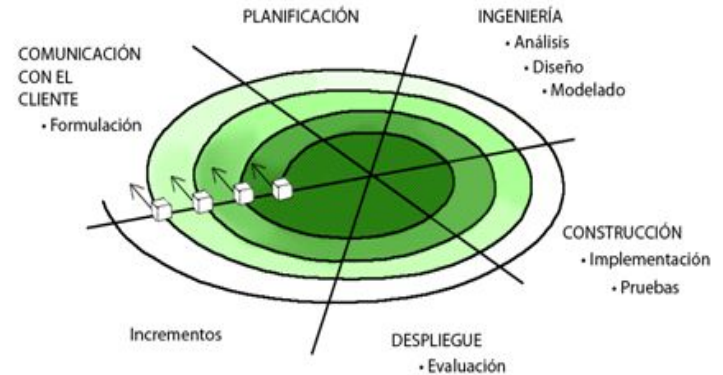


# ¿Cómo funciona el modelo?

Cada proyecto tiene su propio número de bucles. Y por otro lado, cada ciclo de la espiral se llama Fase del proceso de desarrollo de software.

Entonces, este modelo se caracteriza por **iterar repetidamente un conjunto de procesos de desarrollo elemental y eliminar el riesgo, por lo que se está reduciendo activamente**. Todo el proceso de desarrollo pasa repetidamente por ciertas etapas. Cada iteración se llama espiral.

Con cada iteración alrededor de la espiral (comenzando en el centro y siguiendo hacia el exterior), se construyen sucesivas versiones del software, cada vez más completas.

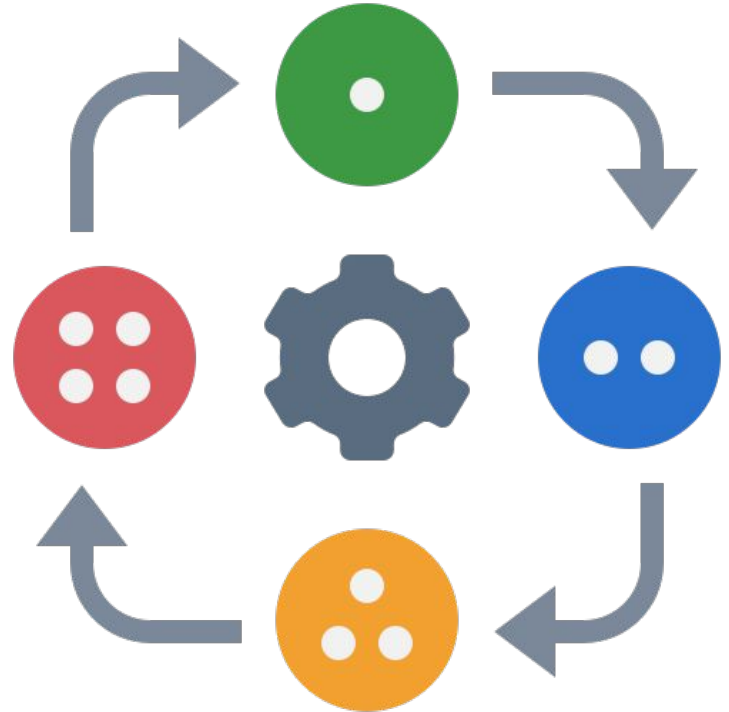


# Etapas

Las cuatro fases principales son:

1. Determinar objetivos
2. Alternativas
3. Restricciones
4. Fase de planificación

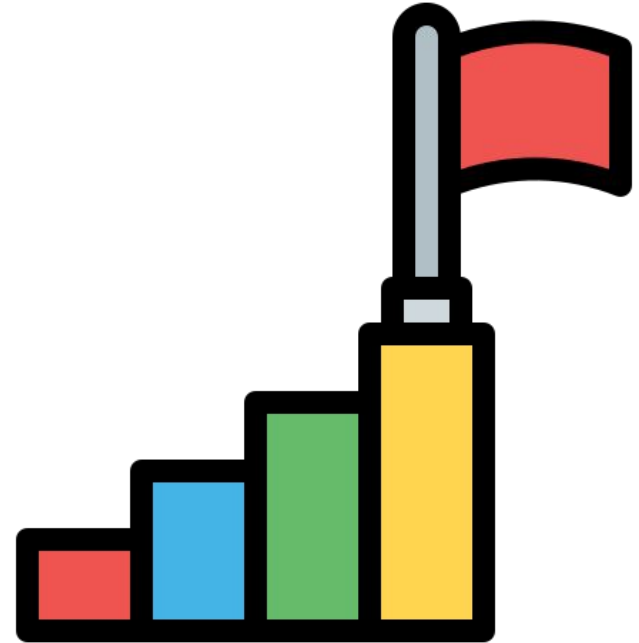
También se suelen nombrar las fases de maneras distintas, como “Determinar objetivos - análisis del riesgo - desarrollo, verificar y validar - planificar”, o incluso “planificación - análisis del riesgo - ingeniería - construcción - adaptación - evaluación del cliente”





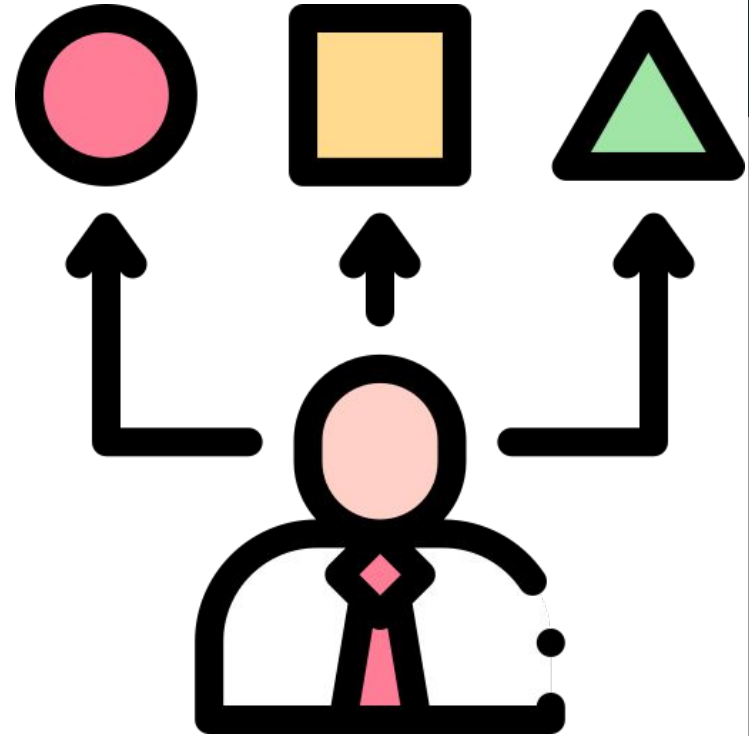
# Determinar objetivos

Un ciclo típico del modelo espiral comienza con la valoración de qué objetivos deben vincularse a cada uno de los pasos del desarrollo de software. Se puede tratar, por ejemplo, de la mejora del rendimiento o de la ampliación de la funcionalidad. Al mismo tiempo, es el momento de definir las alternativas para la implementación (por ejemplo, diseño A vs. diseño B) y determinar las condiciones generales como los costes o la inversión de tiempo.



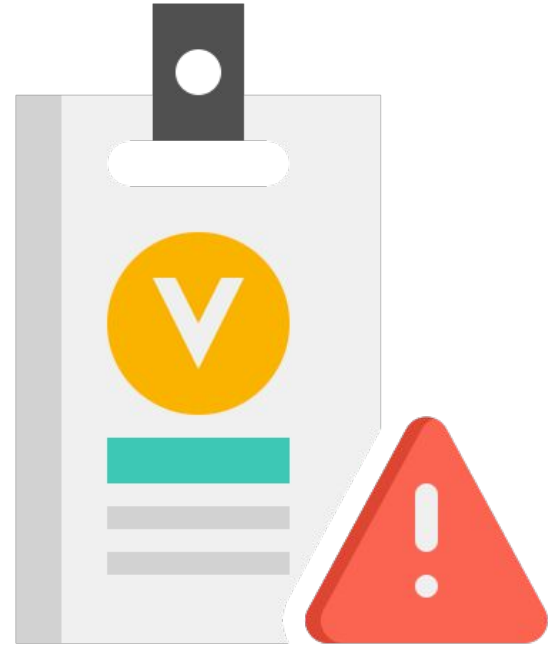
# Alternativas

En este paso, es hora de evaluar las alternativas, momento en el que los objetivos y las condiciones generales serán valores de referencia decisivos. En esta fase del ciclo del desarrollo en espiral, deberán identificarse los ámbitos de inseguridad que presenten un riesgo significativo para el progreso del proyecto de software. Después debe seguir la elaboración de las estrategias que presenten menos riesgo y que sean más rentables.



# Restricciones

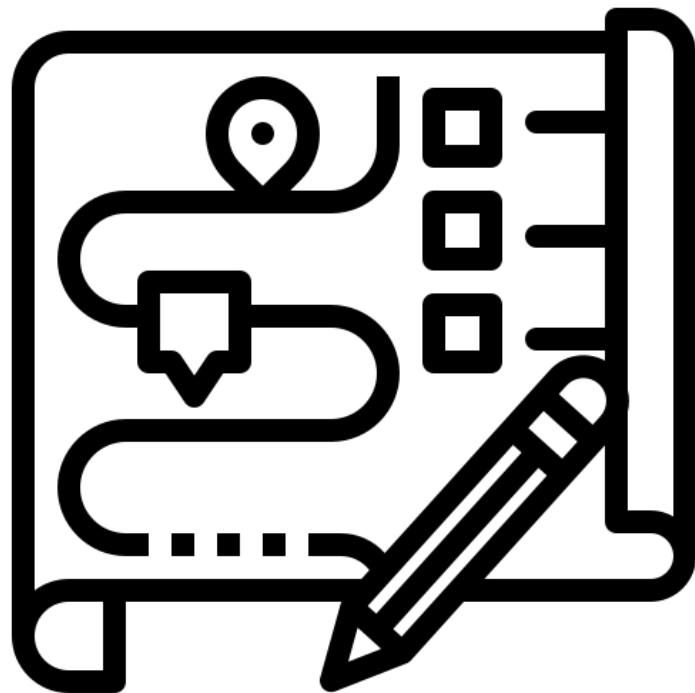
También se le conoce como fase de ingeniería o desarrollo. Mientras el producto se planifica se desarrollan pruebas. En la primera espiral, cuando los requisitos generales no son tan claros, se crea la llamada **Prueba de concepto (POC por sus siglas en inglés)** para obtener los comentarios del cliente. Luego en los siguientes espirales se puede desarrollar una versión funcional de un producto llamado “build” y enviarla al cliente para obtener una nueva y más detallada retroalimentación. Con este enfoque se puede obtener una mejor perspectiva de los requisitos.

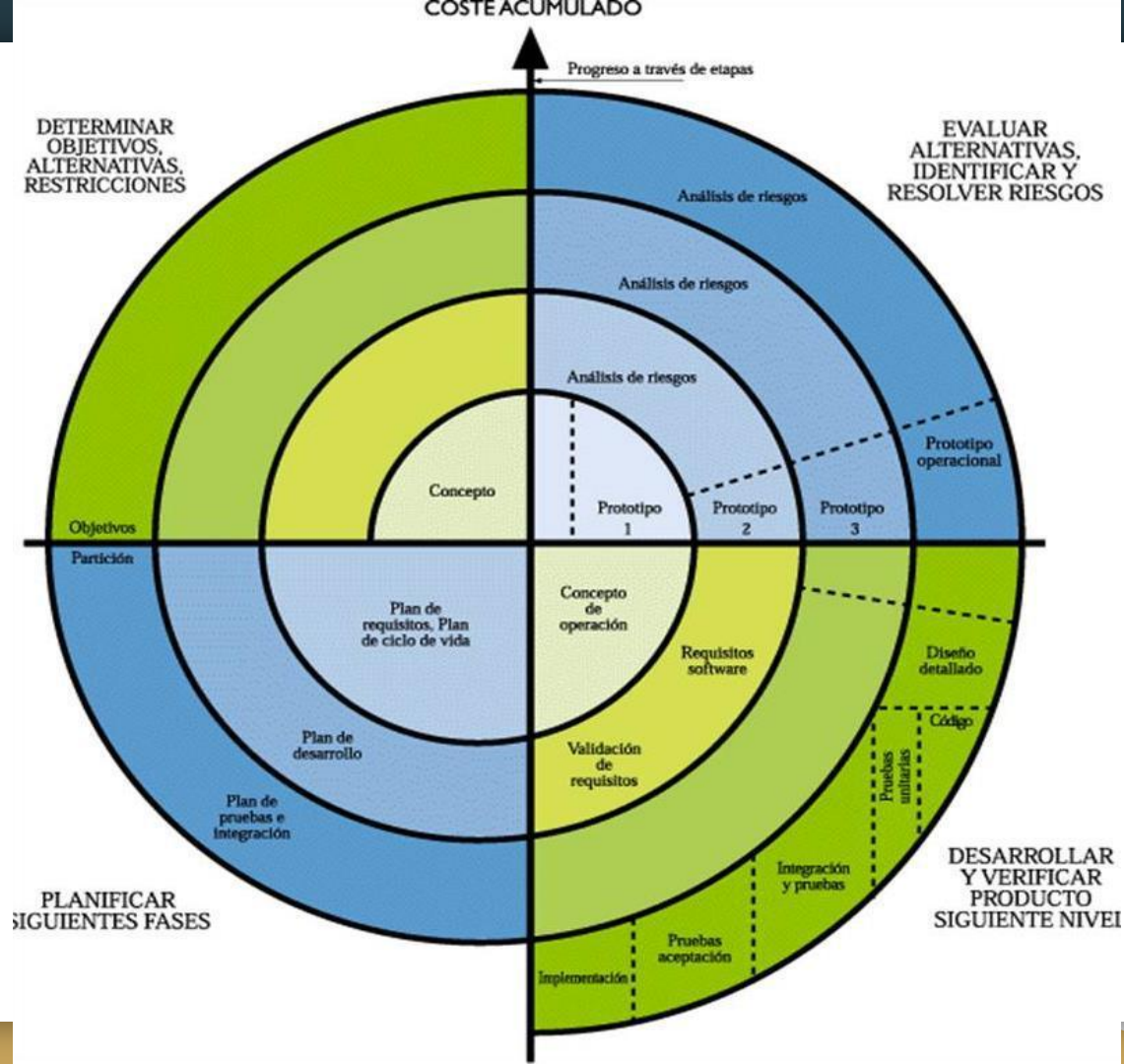


# Fase de planificación

También llamada fase de evaluación. Consiste en planificar las próximas fases y evaluar la salida del proyecto hasta la fecha antes de que el proyecto continúe a la siguiente espiral.

Si se producen errores, entonces vamos a ir buscando soluciones, y si una alternativa es una mejor solución, se prefiere en el siguiente ciclo.





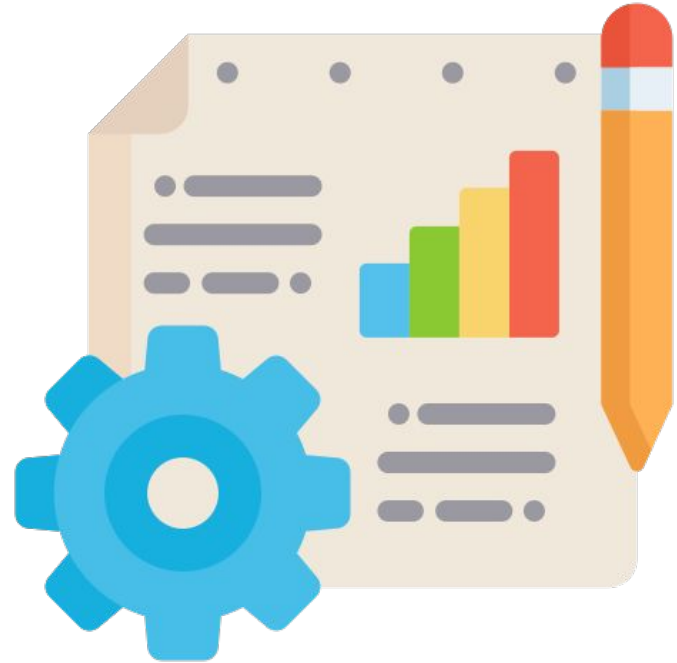
# Ventajas del modelo en espiral

- Manejo del riesgo
- Modificación del sistema
- Flexibilidad
- Menor dificultad de gestión
- Estimación precisa del cronograma
- Gran control de la documentación



# Manejo del riesgo

Este modelo permite el manejo de riesgo. Es una de las características que lo hacen más atractivo especialmente para administrar proyectos grandes y costosos. Además, este enfoque hace que el proyecto sea más transparente porque, por diseño, cada espiral debe ser revisada y analizada. Y finalmente, ese análisis y revisión constante hace que se puedan tratar los riesgos en cada fase. Esto hace que sean ventajosos en proyectos muy grandes, y en entornos técnicos modernos.



# Modificación del sistema en etapas tempranas

El éxito de uso del prototipo depende de qué tan pronto y con qué frecuencia se reciba la retroalimentación del usuario para hacer cambios y adecuarlos a las necesidades actuales.

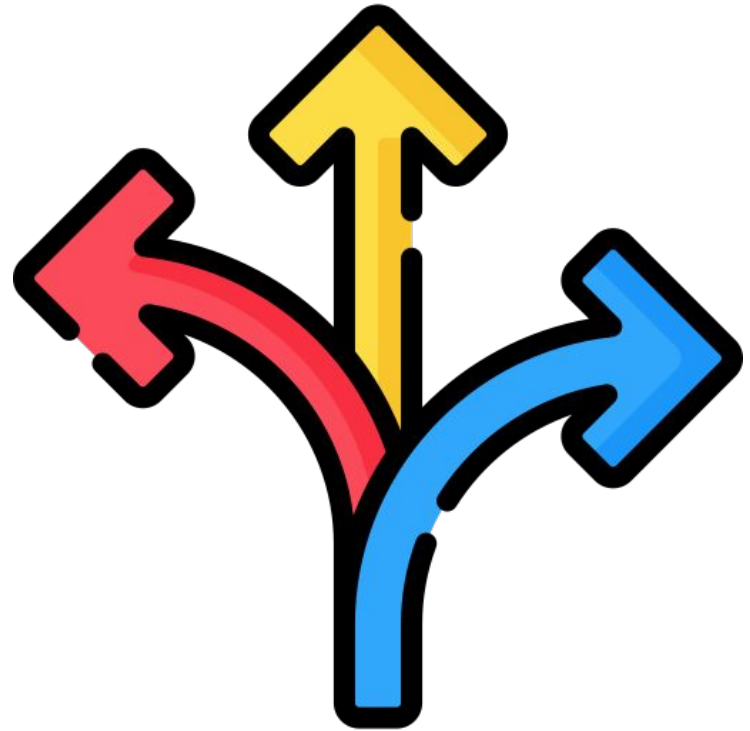
Los cambios iniciales durante el desarrollo de un proyecto son menos costosos que si se realizan en etapas tardías, como el prototipo puede cambiar varias veces, la flexibilidad y adaptabilidad son su esencia.





# Flexibilidad

El modelo en espiral es altamente adaptable a cambios. Debido a que el modelo está diseñado para ser iterativo e incremental, los cambios y mejoras pueden incorporarse en el proceso de desarrollo en cualquier momento, incluso después de que se haya iniciado el proceso.



# Menor dificultad de gestión

El proyecto se puede dividir en varias partes y se pueden evaluar los riesgos de cada una de ellas.

Esto ayuda a disminuir las dificultades de gestión al permitir al equipo de desarrollo abordar cada parte del proyecto de manera individual y centrarse en los riesgos específicos que se presentan en cada fase.



# Estimación precisa del cronograma

Se pueden obtener estimaciones del proyecto en términos de cronograma, los costos se vuelven cada vez más realistas a medida que avanza el proyecto y se completan los bucles en espiral.



# Gran control de la documentación

Hay una gran capacidad de controlar de manera efectiva la documentación del proyecto, que se controla cuidadosamente y se revisa de manera continua para garantizar que sea precisa y actualizada.

A su vez, gracias al control de la documentación, se reduce el riesgo de que el proyecto se desvíe significativamente del plan original.



# Desventajas del modelo en espiral

- Muy costoso
- Muchísima documentación
- Se requiere mucha experiencia y experimentación
- El dinamismo dilata el producto final
- Involucra constantemente al cliente



# Muy costoso

El monitoreo de riesgos requiere recursos adicionales, por eso es un modelo que se considera costoso. Cada espiral requiere experiencia específica, lo que hace que el proceso de gestión sea más complejo.

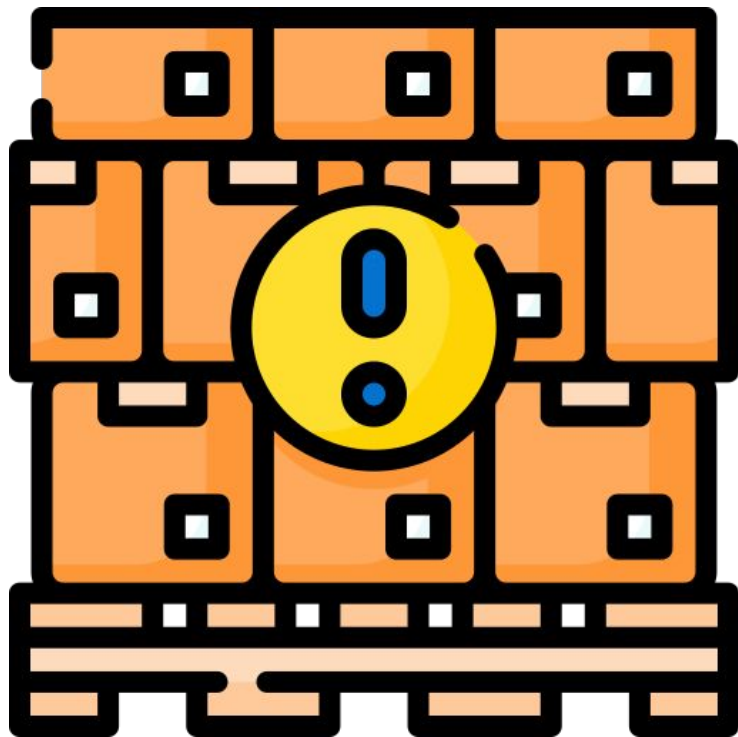
La magnitud del proyecto hace que sea realmente muy costoso en general.



# Muchísima documentación

La excesiva documentación puede ser contraproducente y costosa.

La documentación excesiva puede consumir una gran cantidad de tiempo y recursos. En cada ciclo del modelo en espiral, se espera que los equipos documenten sus decisiones y resultados. Esto puede significar horas adicionales dedicadas a la documentación en lugar de trabajar en el desarrollo real del software.



# Se requiere mucha experiencia y experimentación

En primer lugar, en la mayoría de los casos, un equipo de proyecto típico no tiene la experiencia necesaria para implementar adecuadamente el modelo en espiral.

La experimentación muchas veces puede ser incierta, y puede ser difícil saber qué resultados esperar. En algunos casos, puede haber una gran cantidad de incertidumbre sobre los riesgos asociados con el proyecto.





# El dinamismo dilata el producto final

Las decisiones periódicas y “on-the-fly” pueden dilatar el proceso de desarrollo; el usuario puede entusiasmarse con esta flexibilidad y dinamismo para agregar constantemente funcionalidades nuevas e innecesarias, dilatando el producto final.



# Involucra constantemente al cliente

Involucrar constantemente al cliente puede ser **muy demandante en términos de tiempo y recursos**. Para que el cliente participe de manera efectiva en el proceso de desarrollo, es necesario que esté disponible y dispuesto a proporcionar comentarios y retroalimentación constante. Sin embargo, esto puede ser difícil de lograr si el cliente tiene otras responsabilidades o compromisos y no puede dedicar suficiente tiempo al proyecto.



