

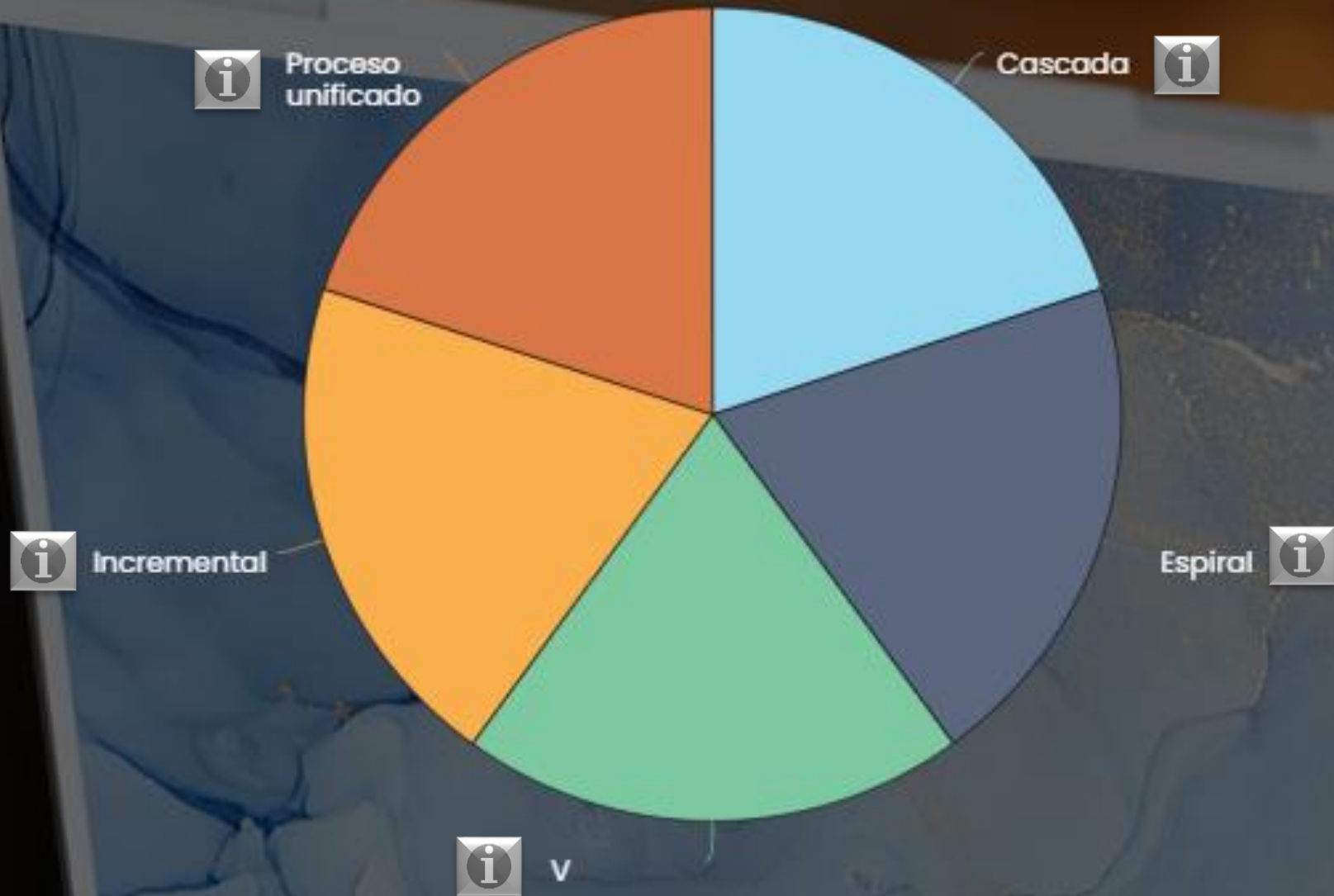
A woman with long brown hair is sitting at a desk, looking at a laptop. She is wearing a white and black striped shirt. The desk has a laptop, some papers, and a small plant. In the background, there are large windows with blinds and several potted plants. The scene is brightly lit, suggesting a sunny day.

Modelos del ciclo de vida del Desarrollo del Software

Nathaly Sipiran Morillo

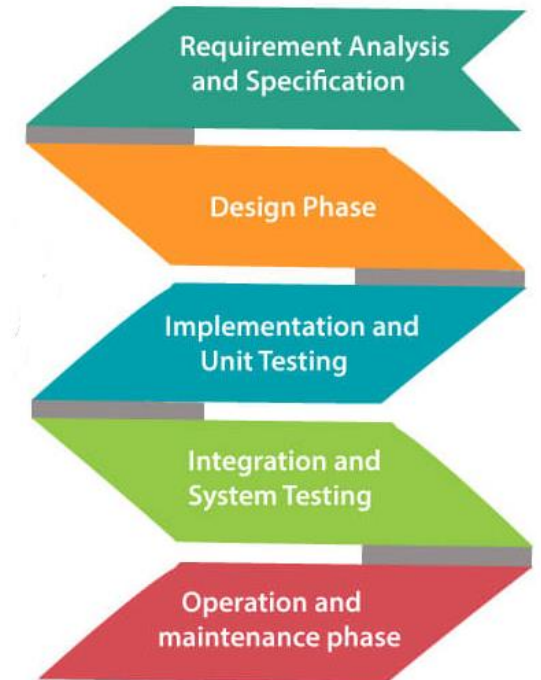
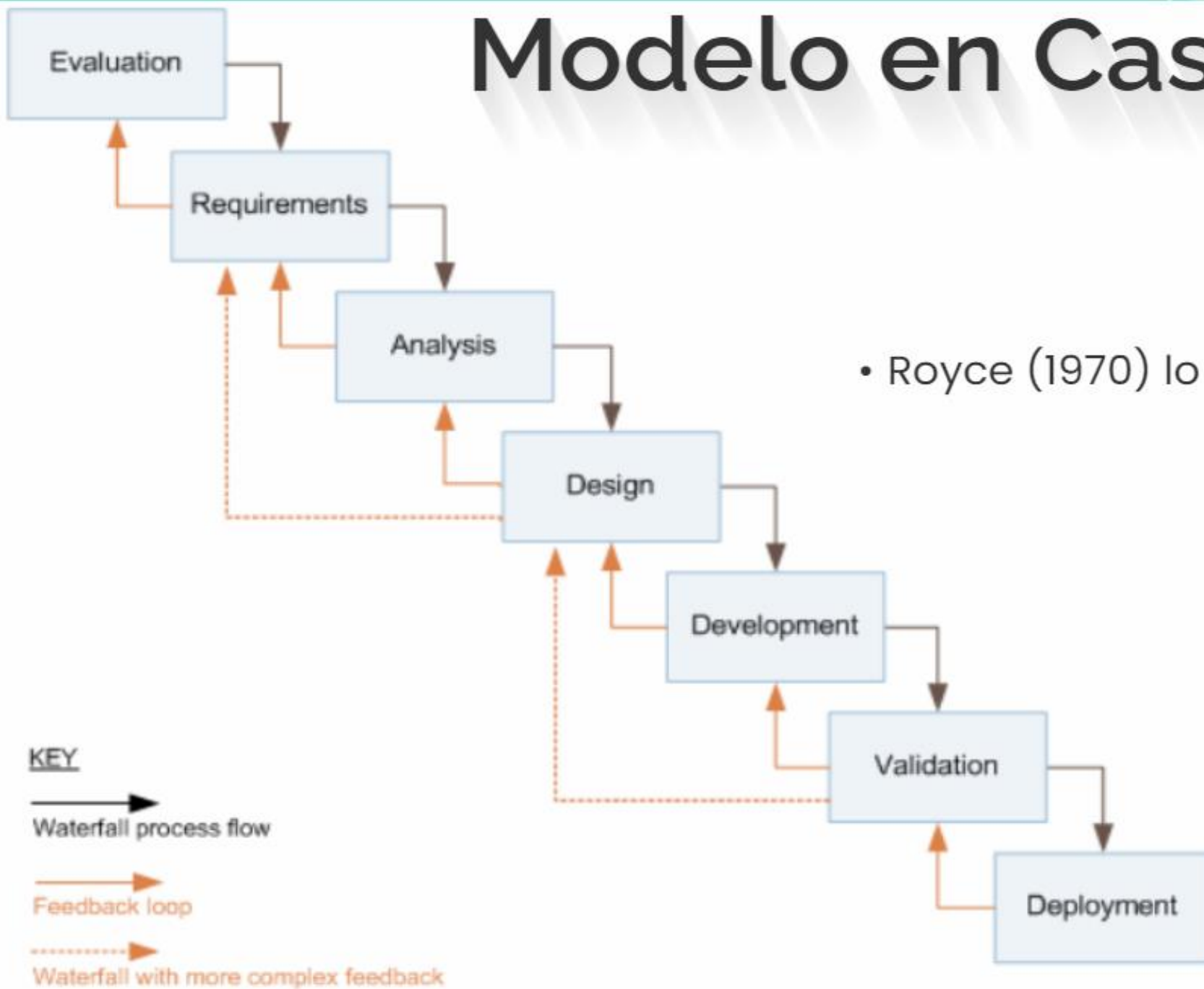
SDLC

Modelos



Modelo en Cascada

- Modelo base
- Bennington (1956)
- Royce (1970) lo mejoró dos veces con retro-alimentaciones



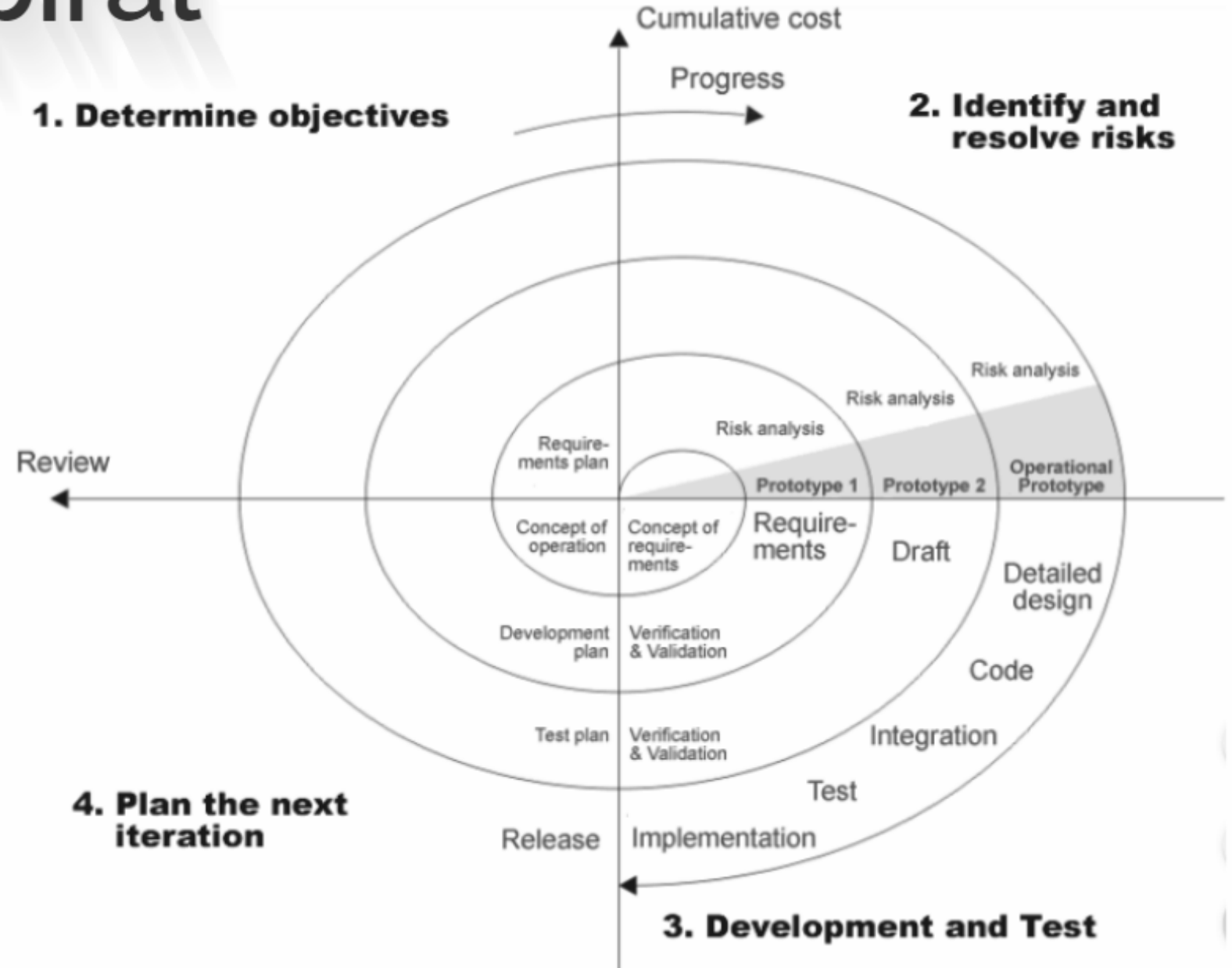


Modelo en cascada

Royce mejoró este modelo al proporcionar un ciclo de retroalimentación para que cada etapa anterior pudiera revisarse. Pero Royce también sintió que esta disposición podría resultar inadecuada ya que la iteración puede necesitar trascender la iteración del par de etapas sucesivas anteriores.

Modelo Espiral

El modelo en espiral representa un cambio de paradigma del enfoque impulsado por la especificación de la caída de agua a uno basado en el riesgo. Cada ciclo atraviesa cuatro cuadrantes, como se ilustra en la Figura 4. Estos son:



Modelo espiral

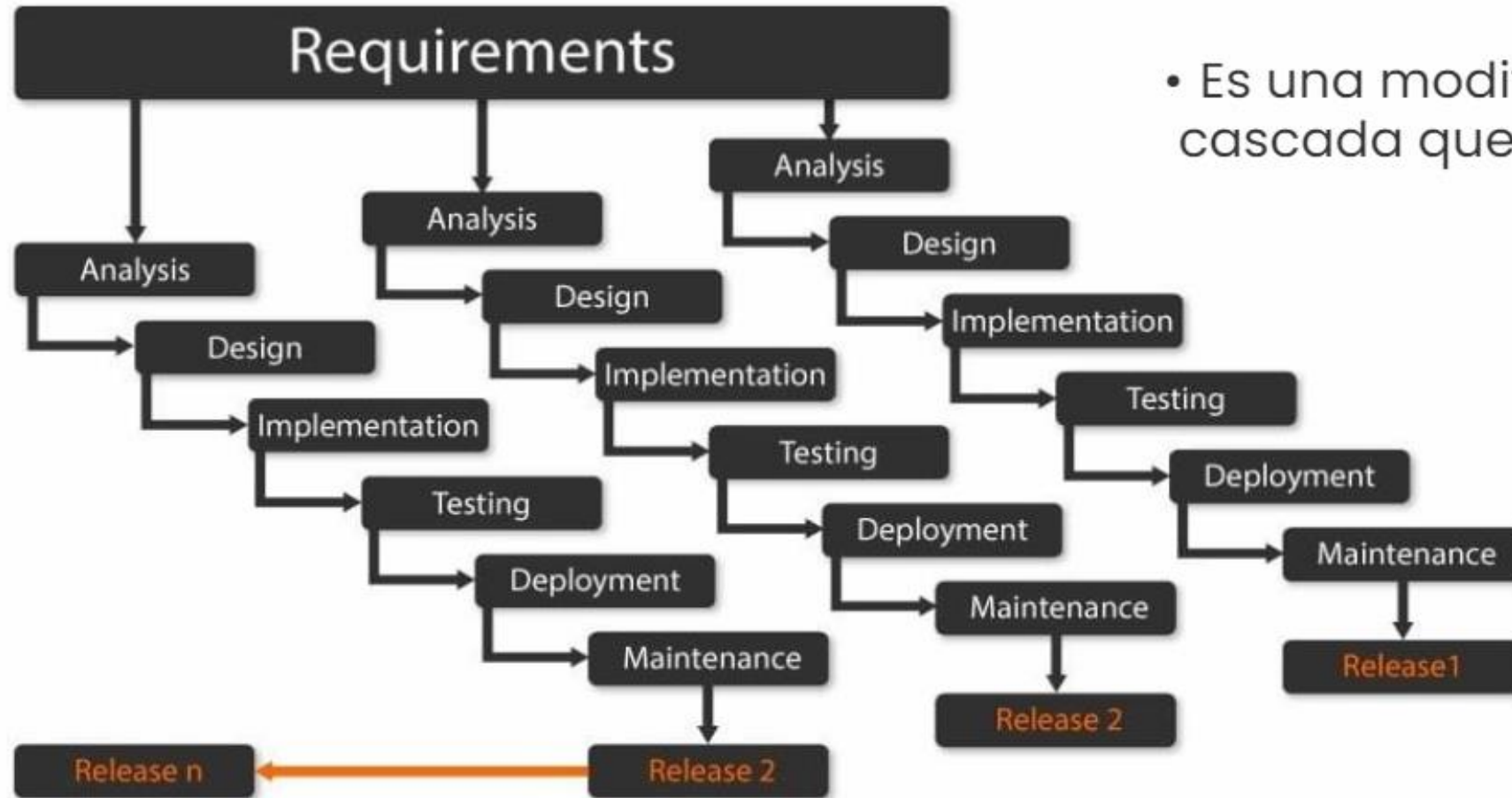
Establecimiento de objetivos: cada ciclo en la espiral comienza con la identificación del propósito para ese ciclo, las diversas alternativas que son posibles para alcanzar los objetivos y las limitaciones que existen.

Evaluación y reducción de riesgos: La siguiente fase del ciclo es calcular estas diversas alternativas en función de los objetivos y las limitaciones. El enfoque de la evaluación en esta etapa se encuentra en la percepción de riesgo del proyecto.

Desarrollo y validación: la siguiente fase es desarrollar estrategias que resuelvan incertidumbres y riesgos. Este proceso puede incluir actividades como evaluación comparativa, simulación y creación de prototipos.

Planificación: Finalmente, se planifica el siguiente paso. Se revisa el proyecto y se elige si continuará con otro período de la espiral. Si se determina que se mantiene, se elaboran planes para el siguiente paso del proyecto.

Modelo Incremental



- Es una modificación al sombrero de cascada que se aproxima al modelo espiral.

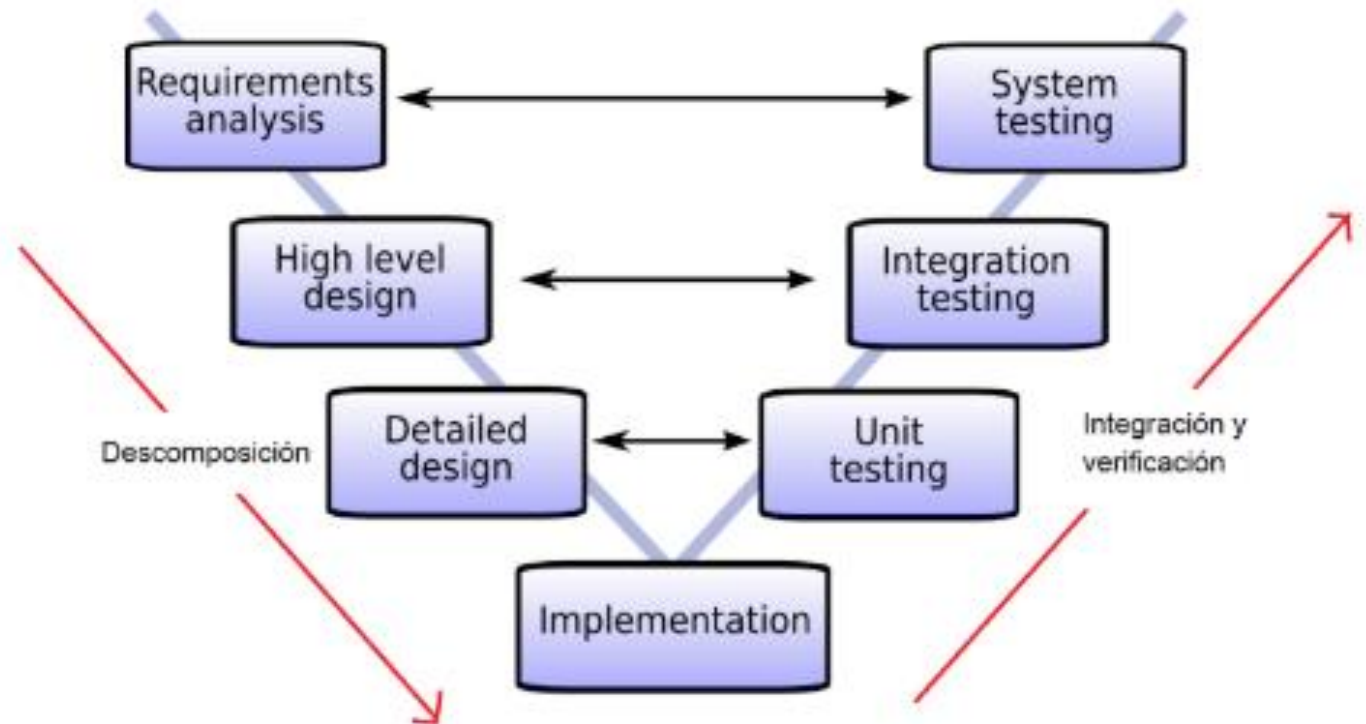
Modelo incremental

Conocido como modelo de cascada iterativo, puede verse como una representación tridimensional del modelo de cascada. El eje z contiene una serie de modelos en cascada para representar el número de iteraciones que se realizarían para mejorar la funcionalidad del producto final de forma incremental. Las principales fortalezas de este modelo son:

- i) La retroalimentación de las iteraciones anteriores se puede incorporar en la iteración actual.
- ii) Las partes interesadas pueden participar a través de las iteraciones y, por lo tanto, ayudan a identificar los riesgos arquitectónicos antes.
- iii) Facilita la entrega del producto con versiones tempranas e incrementales que evolucionan a un conjunto completo de características con cada iteración.
- iv. La implementación incremental permite monitorear los cambios y aislar y resolver los problemas para mitigar los riesgos.

Modelo V

- El modelo es una variación del modelo de cascada en forma de V doblada por la mitad



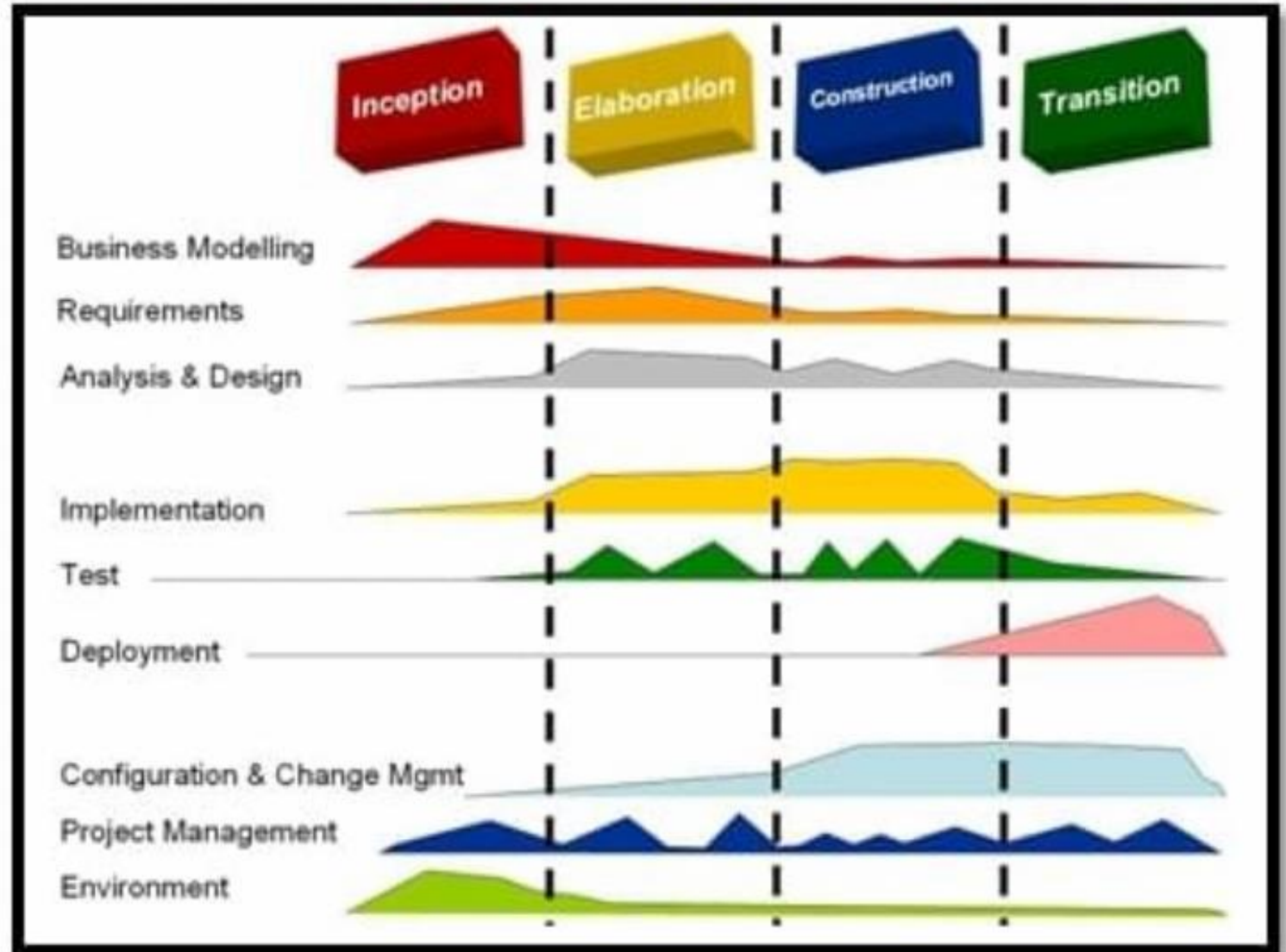


Modelo V

La pierna izquierda de la forma de V representa la evolución de los requisitos del usuario en componentes cada vez más pequeños a través del proceso de descomposición y definición; la pierna derecha representa la integración y verificación de los componentes del sistema en niveles sucesivos de implementación y ensamblaje. El eje vertical representa el nivel de descomposición desde el nivel del sistema, en la parte superior, hasta el nivel de detalle más bajo en el nivel de componente en la parte inferior. Por lo tanto, cuanto más complejo es un sistema, más profunda es la forma de V con un número correspondientemente mayor de etapas.

Modelo Proceso unificado

Mientras que la cascada se basa en las especificaciones y la espiral se basa en el riesgo, el modelo de proceso unificado se basa en el modelo (o la arquitectura) y en el caso de uso



Modelo proceso unificado

creado para abordar los requisitos de desarrollo específicos del software orientado a objetos y su diseño. Fue desarrollado en la década de 1990 por Rational Software; Por lo tanto, se conoce como el modelo Rational Unified Process (RUP). RUP encapsula siete mejores prácticas dentro del modelo:

- i. Desarrolle iterativamente utilizando la gestión de riesgos para impulsar las iteraciones
- ii. Gestionar requisitos.
- iii. Emplear una arquitectura basada en componentes.
- iv. Usa modelos visuales.
- v. Verificar la calidad continuamente.
- vi. Control de cambios.
- vii. Usar personalización.



Referencias

Jaiswal, S. (2018). *Javatpoint*. <https://www.javatpoint.com/software-engineering-sdlc-models>

Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, 35(3), 8–13. <https://doi.org/10.1145/1764810.1764814>