



Ciclo de vida del Desarrollo

¿Qué es el SDLC?

- **Ciclo de Vida del Desarrollo de Sistemas**

- Método que facilita el desarrollo de los sistemas de información
- Guía para los gestores de un proyecto puedan planificar el proceso de diseño y puesta en marcha de cualquier sistema de información que deba reunir ciertos requisitos de cara a su usuario.
- Esquema para concretar los tiempos de desarrollo y la inversión del presupuesto
- Se puede gestionar del modo más efectivo cada tarea y detalle durante todo el proceso de diseño de los sistemas, proporcionándole un calendario de objetivos fundamentales para comunicar todos los involucrados o interesados por el proyecto



<https://www.viewnext.com/el-ciclo-sdlc-en-7-fases/>

Fases del SDLC

1.- Planificación del concepto

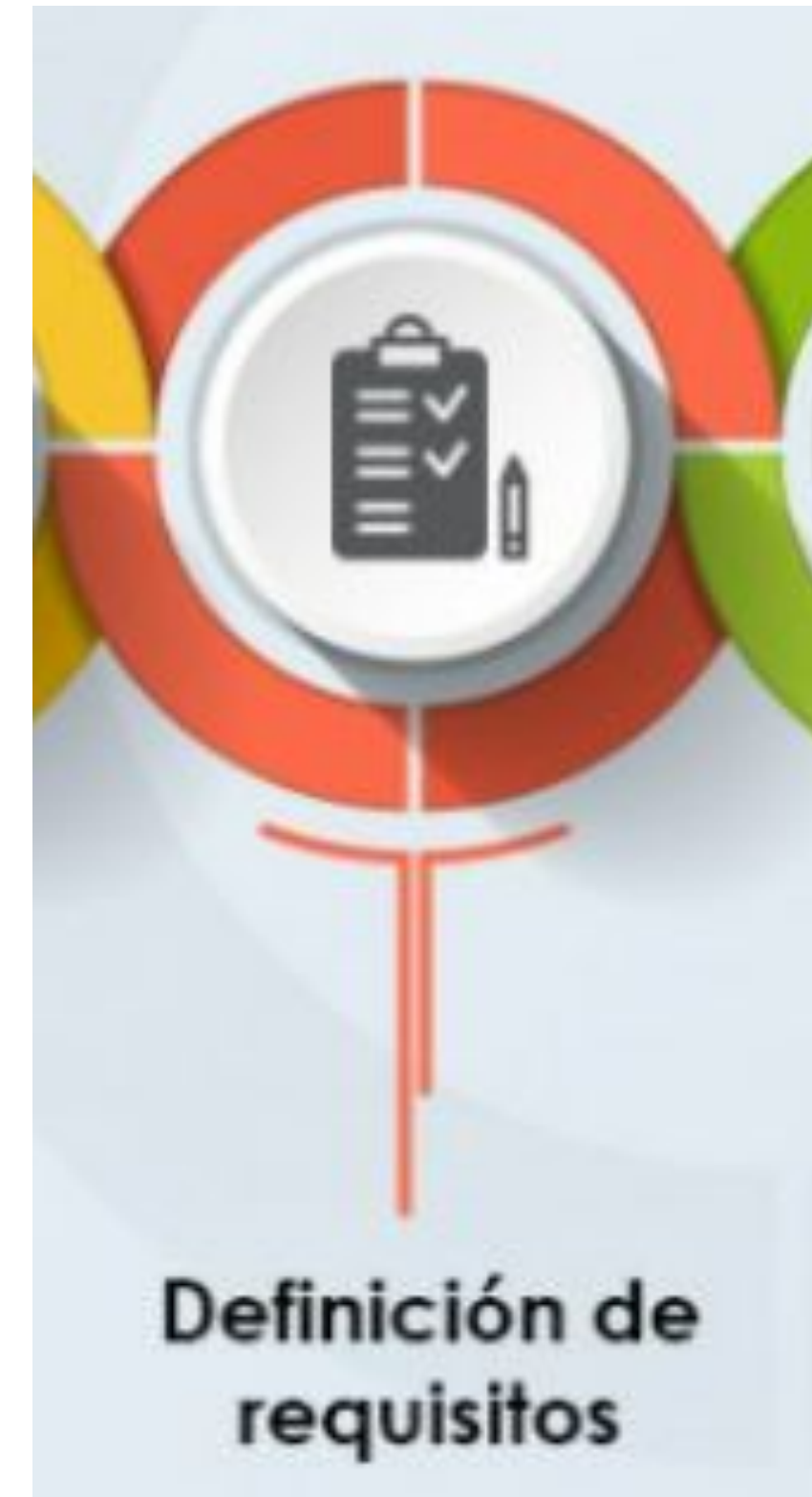
- La fase de planificación en el Ciclo de Vida del Desarrollo de Software (SDLC) es crucial para establecer los cimientos del proyecto.
- Algunas de las tareas que se realiza son:
 - 1. Definición de objetivos:** En esta fase, las organizaciones definen las metas y objetivos del proyecto de software. También determinan los recursos necesarios para completar el proyecto.
 - 2. Recopilación de requisitos:** Se recopilan y documentan los requisitos del usuario para el software. Esto implica comprender las necesidades de los usuarios, las funcionalidades requeridas y las restricciones del sistema.
 - 3. Análisis de costos y beneficios:** Se evalúan los costos y beneficios del proyecto. Esto incluye estimar los gastos, los plazos y los posibles riesgos.
 - 4. Programación y asignación de recursos:** Se crea un plan detallado que establece la programación del proyecto y asigna los recursos necesarios, como personal, herramientas y tecnologías.



<https://www.viewnext.com/el-ciclo-sdlc-en-7-fases/>

Fases del SDLC

- En la fase de recopilación y análisis de **requisitos** del Ciclo de Vida del Desarrollo de Software (SDLC), se llevan a cabo las siguientes actividades cruciales:
 - 1. Investigación exhaustiva:** Se recopilan y analizan los requisitos del usuario. Esto implica comprender las necesidades funcionales y no funcionales del sistema.
 - 2. Discusión con clientes y partes interesadas:** Se mantienen conversaciones con los clientes y otras partes interesadas para alinear las expectativas y comprender sus necesidades específicas.
 - 3. Estudio de viabilidad:** Se evalúa si el proyecto es factible desde el punto de vista técnico, económico y temporal.



<https://www.viewnext.com/el-ciclo-sdlc-en-7-fases/>

Fases del SDLC

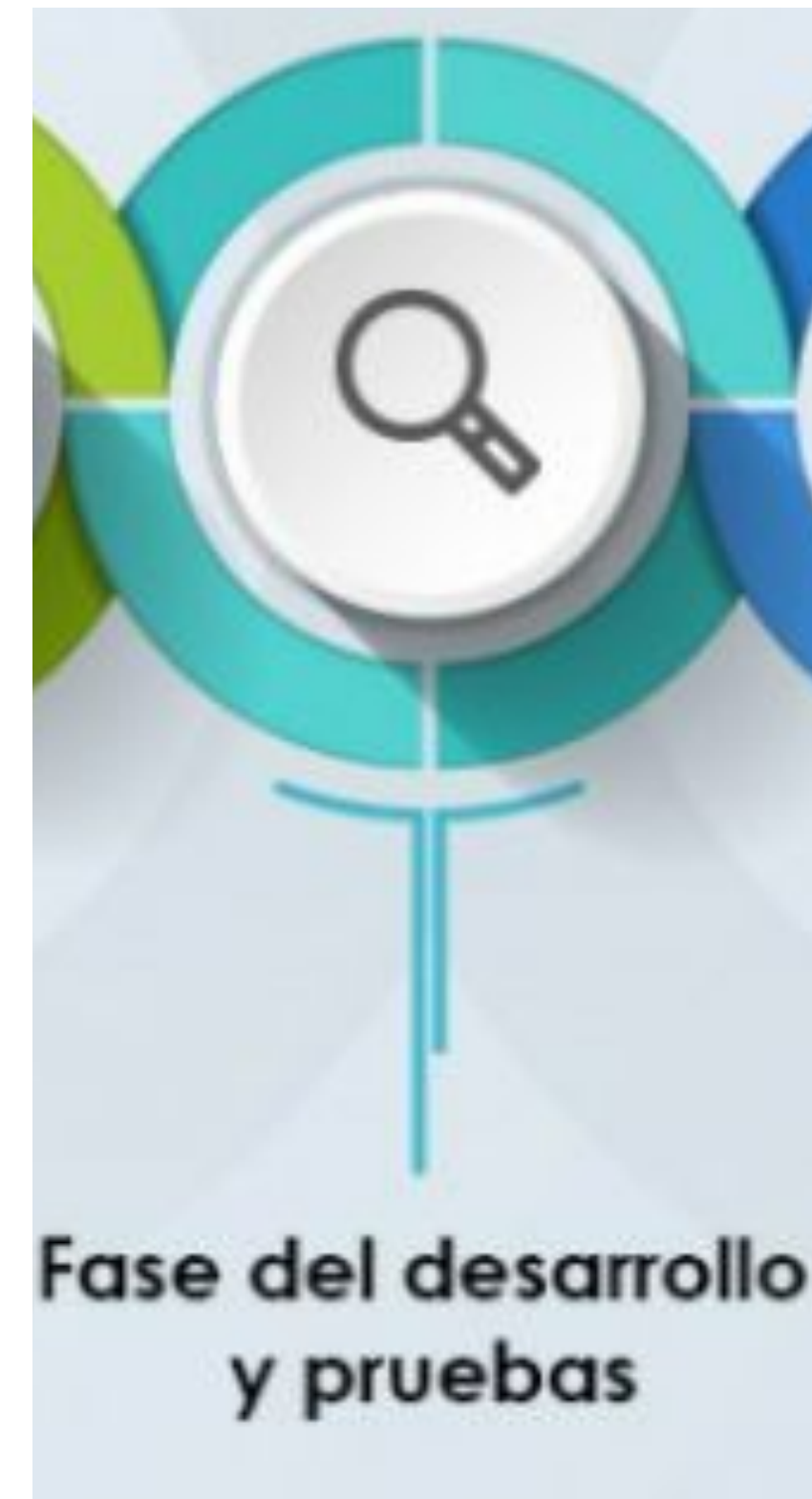
- En la fase de **diseño**, los ingenieros de software realizan varias actividades clave para transformar los requisitos en una solución concreta.
- Aquí están algunas de las tareas comunes en esta etapa:
 - 1. Análisis de requisitos:** Los ingenieros revisan los requisitos del usuario y los traducen en especificaciones técnicas. Esto implica comprender las necesidades funcionales y no funcionales del sistema.
 - 2. Arquitectura del software:** Se define la estructura general del software, incluyendo componentes, módulos y su interacción. Esto ayuda a garantizar que el sistema sea escalable, eficiente y fácil de mantener.
 - 3. Diseño de la interfaz de usuario (UI):** Se crea la apariencia visual y la interacción del software. Esto incluye diseñar pantallas, flujos de navegación y elementos gráficos.
 - 4. Selección de tecnologías:** Los ingenieros eligen las herramientas, lenguajes de programación y frameworks adecuados para implementar la solución.
 - 5. Diseño de base de datos:** Se modela la estructura de la base de datos, definiendo tablas, relaciones y consultas.
 - 6. Diseño de seguridad:** Se establecen medidas de seguridad, como autenticación, autorización y cifrado.
 - 7. Documentación técnica:** Se crea documentación detallada para guiar a los desarrolladores durante la implementación.



<https://www.viewnext.com/el-ciclo-sdlc-en-7-fases/>

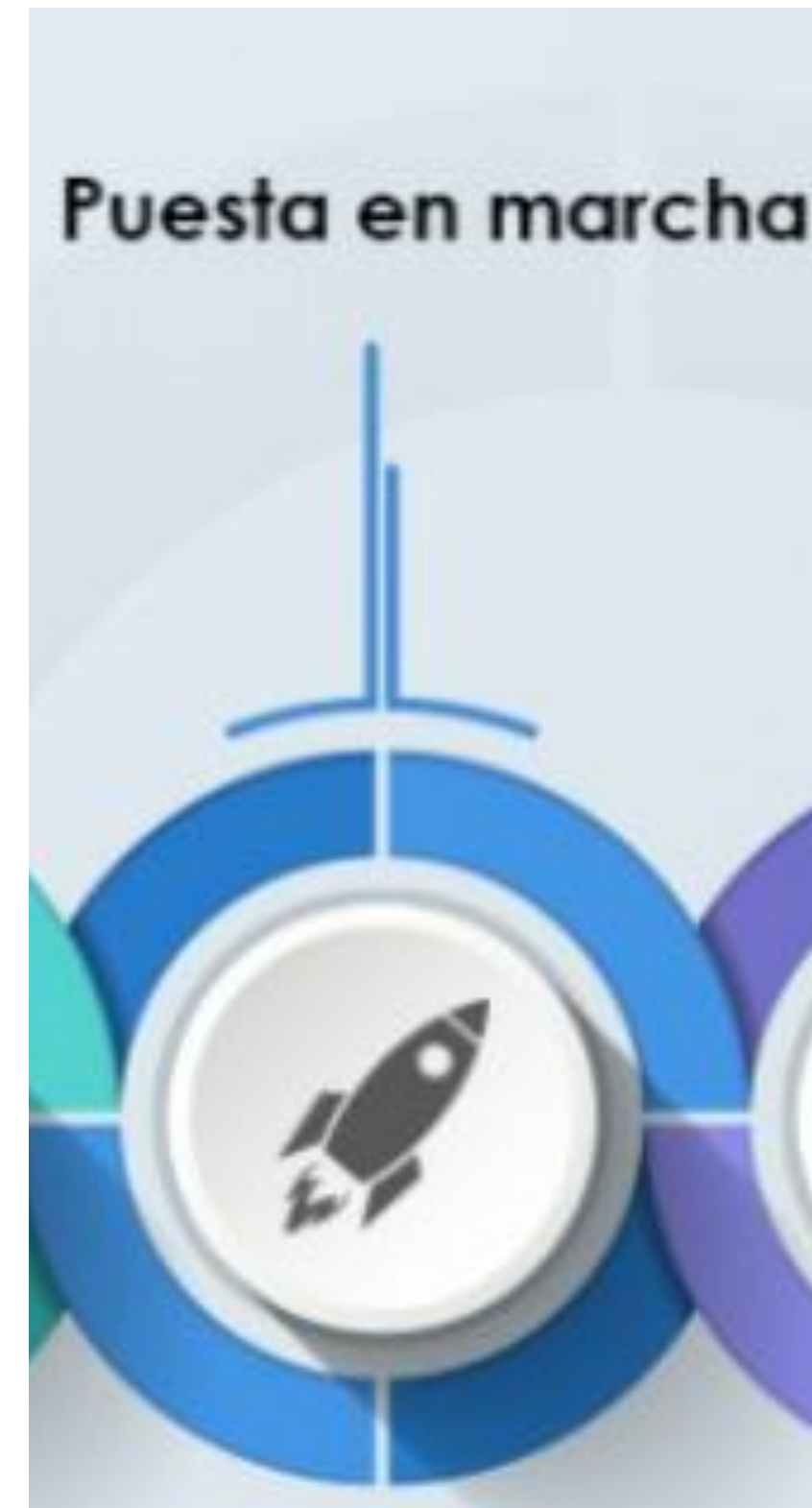
Fases del SDLC

- En el Ciclo de Vida del Desarrollo de Software (SDLC), las actividades en las fases de desarrollo y pruebas son esenciales para crear software de alta calidad.
- Aquí están los detalles:
 - 1. Desarrollo:**
 - I. Codificación:** Los desarrolladores escriben el código fuente del software.
 - II. Integración:** Se combinan los módulos y componentes para formar una aplicación completa.
 - III. Optimización:** Se mejora el rendimiento y la eficiencia del código.
 - IV. Documentación:** Se crea documentación técnica para futuras referencias.
 - 2. Pruebas:**
 - I. Pruebas unitarias:** Se verifican las partes individuales del código.
 - II. Pruebas de integración:** Se evalúa la interacción entre módulos.
 - III. Pruebas funcionales:** Se verifica si el software cumple con los requisitos.
 - IV. Pruebas de rendimiento:** Se evalúa la velocidad y escalabilidad.
 - V. Pruebas de seguridad:** Se identifican vulnerabilidades.
 - VI. Pruebas de aceptación:** Los usuarios validan el software.
- Estas etapas aseguran que el software funcione correctamente y cumpla con las expectativas.



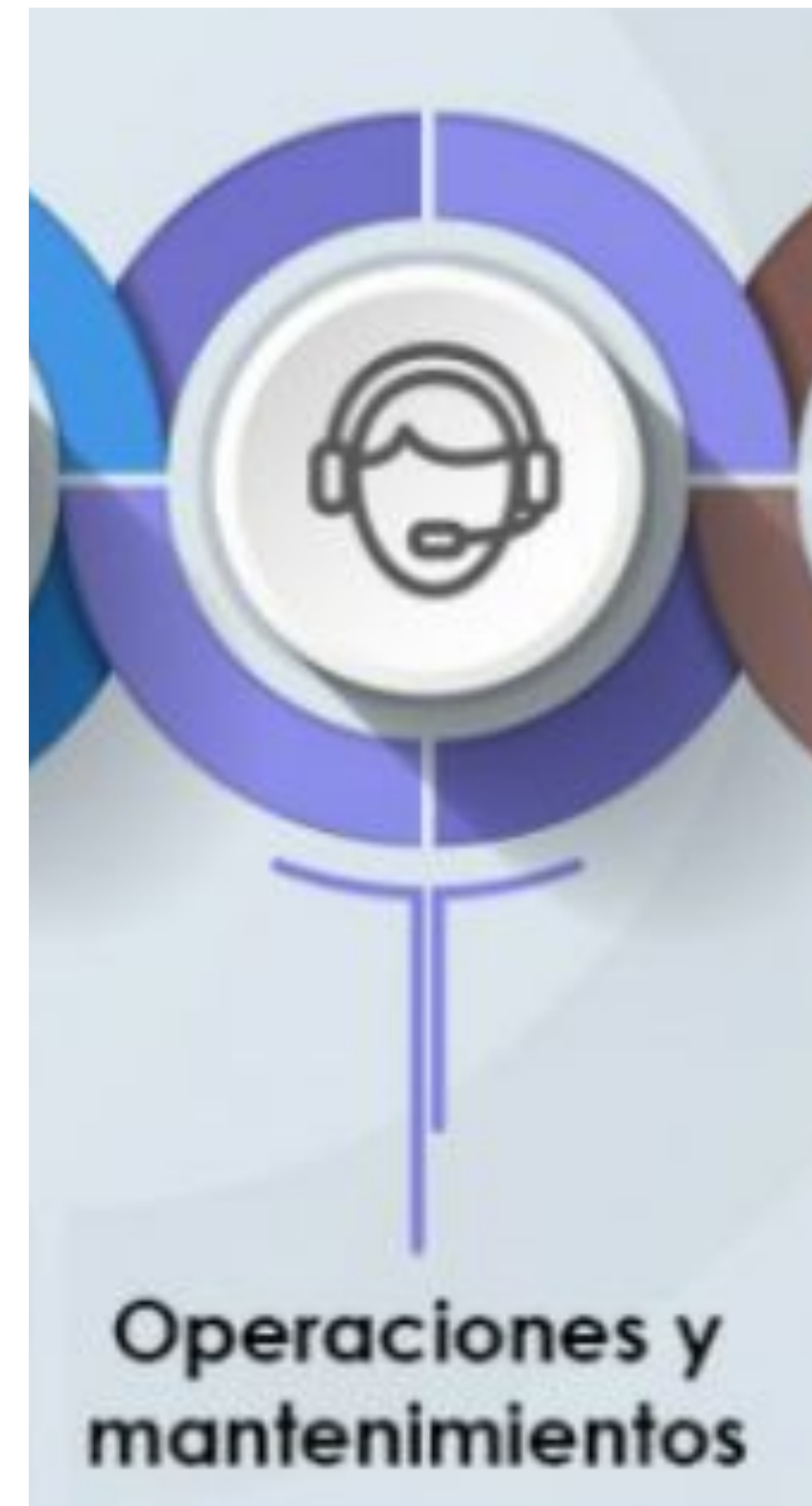
Fases del SDLC

- La fase de implementación en el Ciclo de Vida del Desarrollo de Software (SDLC) es donde el software se pone en marcha.
- Aquí están las actividades clave:
 1. **Instalación del software:** Se implementa el software en el entorno de producción o en los servidores del cliente.
 2. **Configuración y ajustes:** Se configuran parámetros específicos según las necesidades del cliente.
 3. **Pruebas finales:** Se realizan pruebas exhaustivas para asegurar que el software funcione correctamente.
 4. **Capacitación de usuarios:** Se capacita a los usuarios para que utilicen el software de manera efectiva.
- Una vez completada esta fase, el software está listo para su uso.



Fases del SDLC

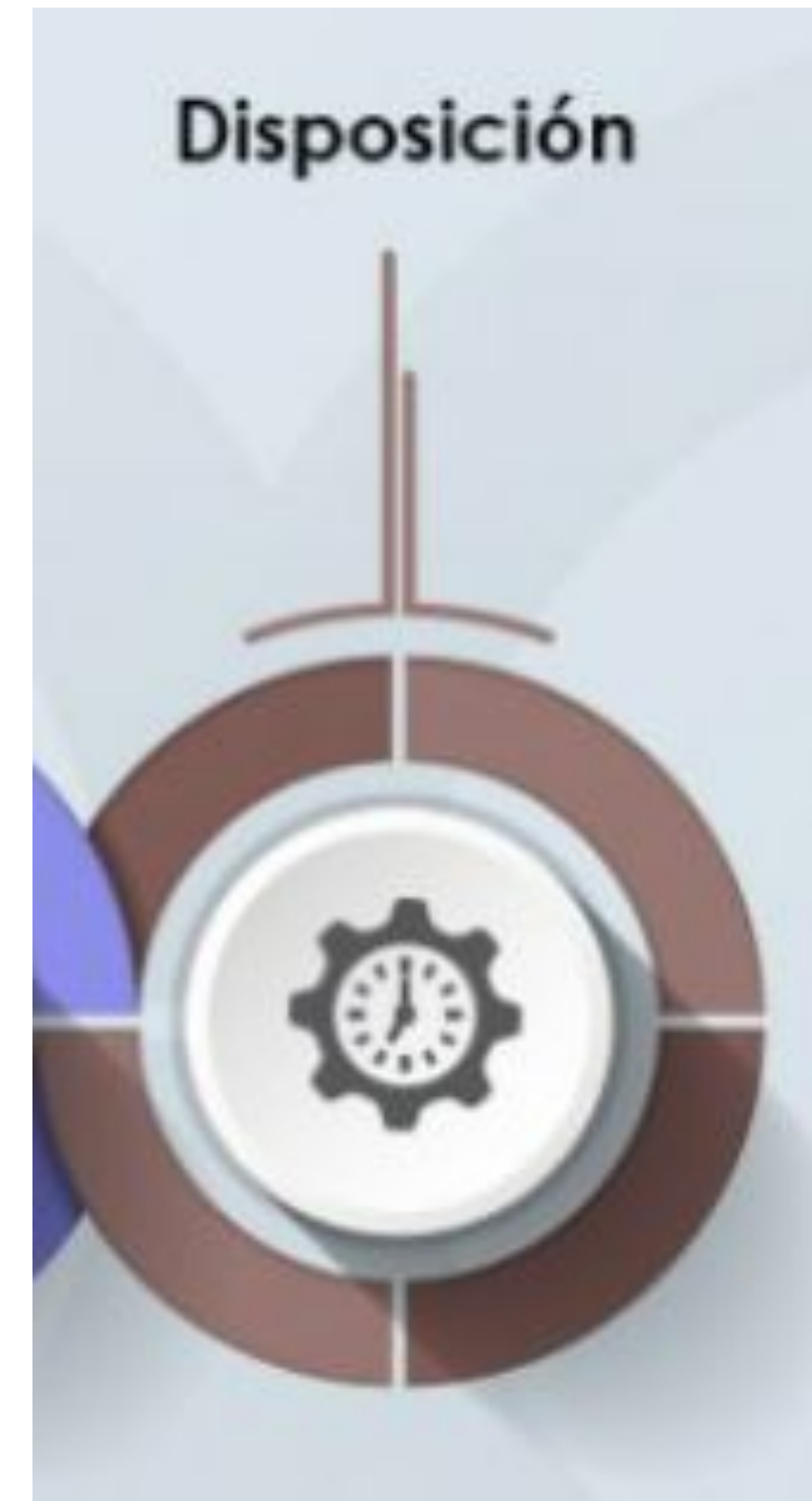
- Durante una fase de **operación total y mantenimientos**, los expertos desarrolladores controlan el sistema para asegurar que cumple los requisitos de negocio solicitados antes del diseño.
- Se ofrece un servicio de mantenimiento y de soporte a los usuarios para garantizar que el sistema sigue funcionando correctamente.
- Algunas de las tareas que se realizan son:
 - **Configuración y ajustes:** Se configuran parámetros específicos según las necesidades del cliente.
 - **Capacitación de usuarios:** Se capacita a los usuarios para que utilicen el software de manera efectiva.
 - **Actualizaciones y correcciones:** Se realizan actualizaciones para agregar nuevas funcionalidades o corregir errores.
 - **Optimización:** Se mejora el rendimiento y la eficiencia del software.
 - **Gestión de errores:** Se solucionan problemas y se realizan ajustes según sea necesario.



<https://www.viewnext.com/el-ciclo-sdlc-en-7-fases/>

Fases del SDLC

- La fase de **disposición** en el Ciclo de Vida del Desarrollo de Software (SDLC) se refiere a la retirada o finalización del software.
- Esta fase comprende el fin del ciclo de vida del sistema y su retiro del funcionamiento.
- Se deben seguir unos pasos sistemáticos para finalizar el sistema en un entorno de seguridad, que permita conservar toda información útil o sensible de cara a continuar con los negocios en un sistema nuevo.
- Aunque no es una fase comúnmente mencionada, aquí están algunas actividades relacionadas:
 - **Retiro del software:** Si el software ya no es útil o relevante, se retira del uso. Esto puede implicar la desactivación de servidores, la eliminación de datos o la notificación a los usuarios.
 - **Documentación final:** Se crea una documentación completa que describe cómo desactivar y retirar el software de manera adecuada.

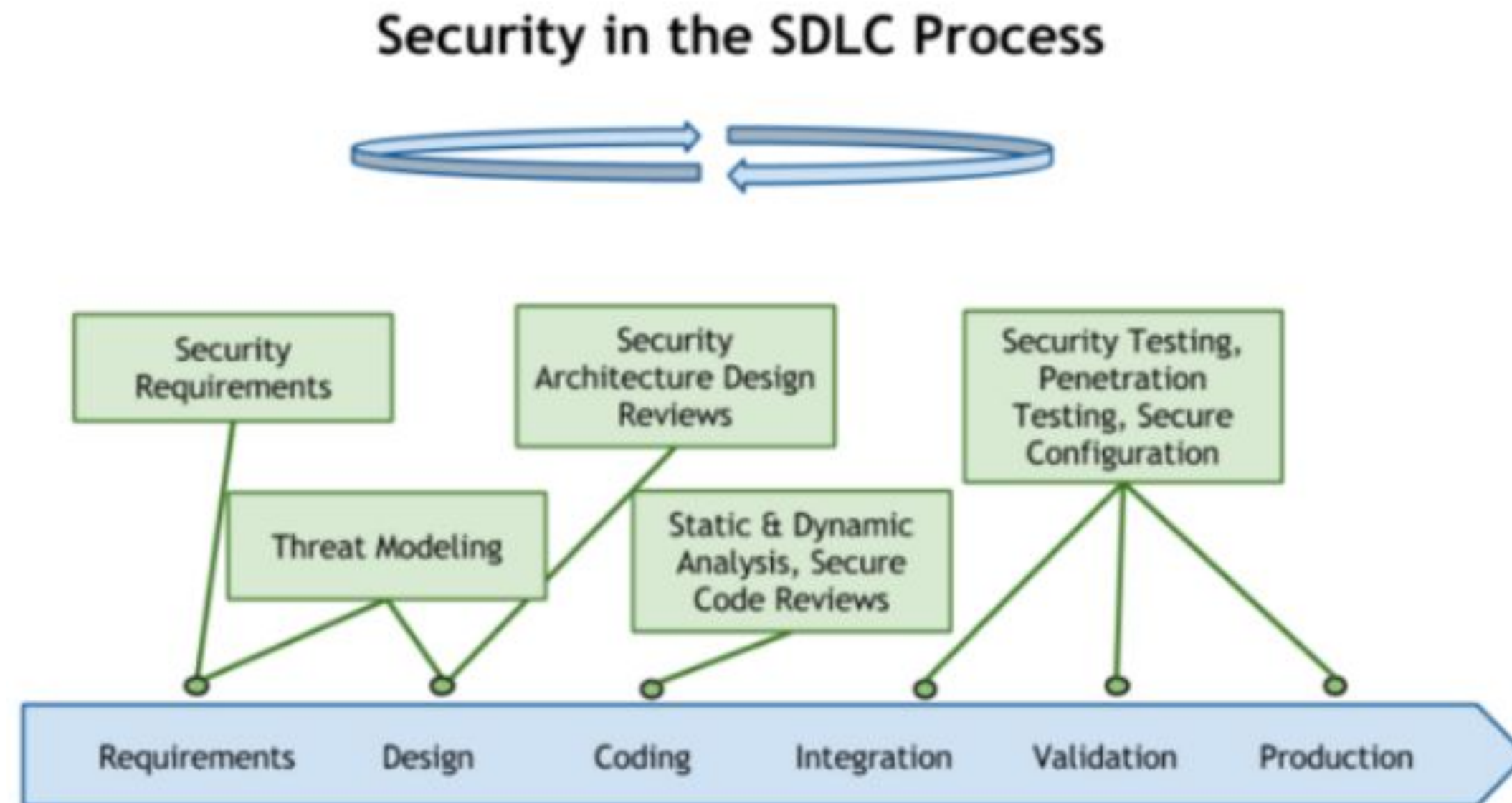


<https://www.viewnext.com/el-ciclo-sdlc-en-7-fases/>

¿Qué es el SSDLC?

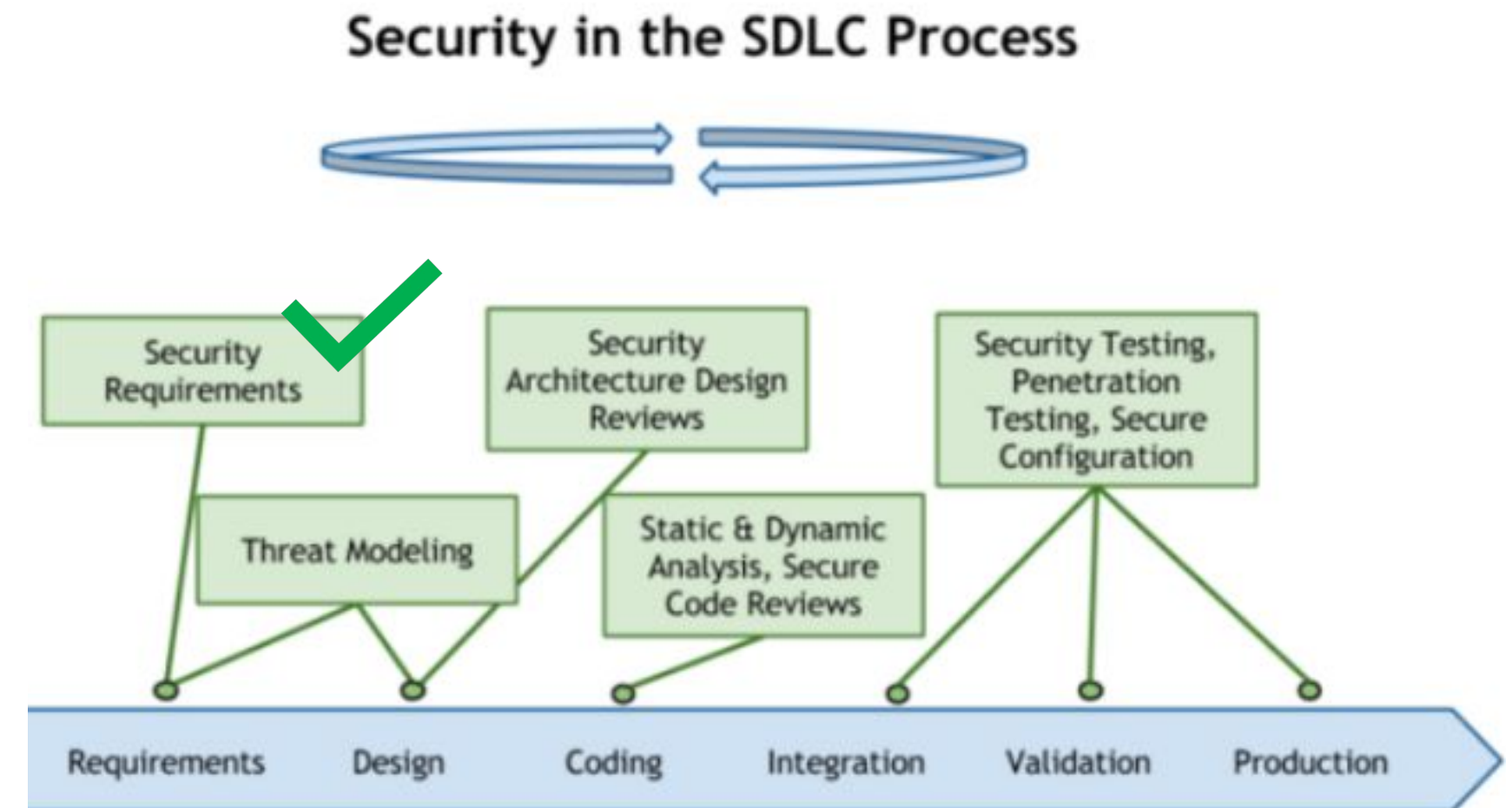
- **Ciclo de vida de desarrollo seguro (S-SDLC)**

- El **S-SDLC (Ciclo de Vida de Desarrollo Seguro)** es una variante que se enfoca en la seguridad en cada etapa del ciclo de vida de las aplicaciones a desarrollar.
- Su objetivo es crear aplicaciones lo más seguras posible desde el inicio del proceso de diseño hasta su publicación.
- Esta metodología garantiza la calidad de las creaciones y se concibe como una herramienta de mantenimiento.



Fases SSDLC

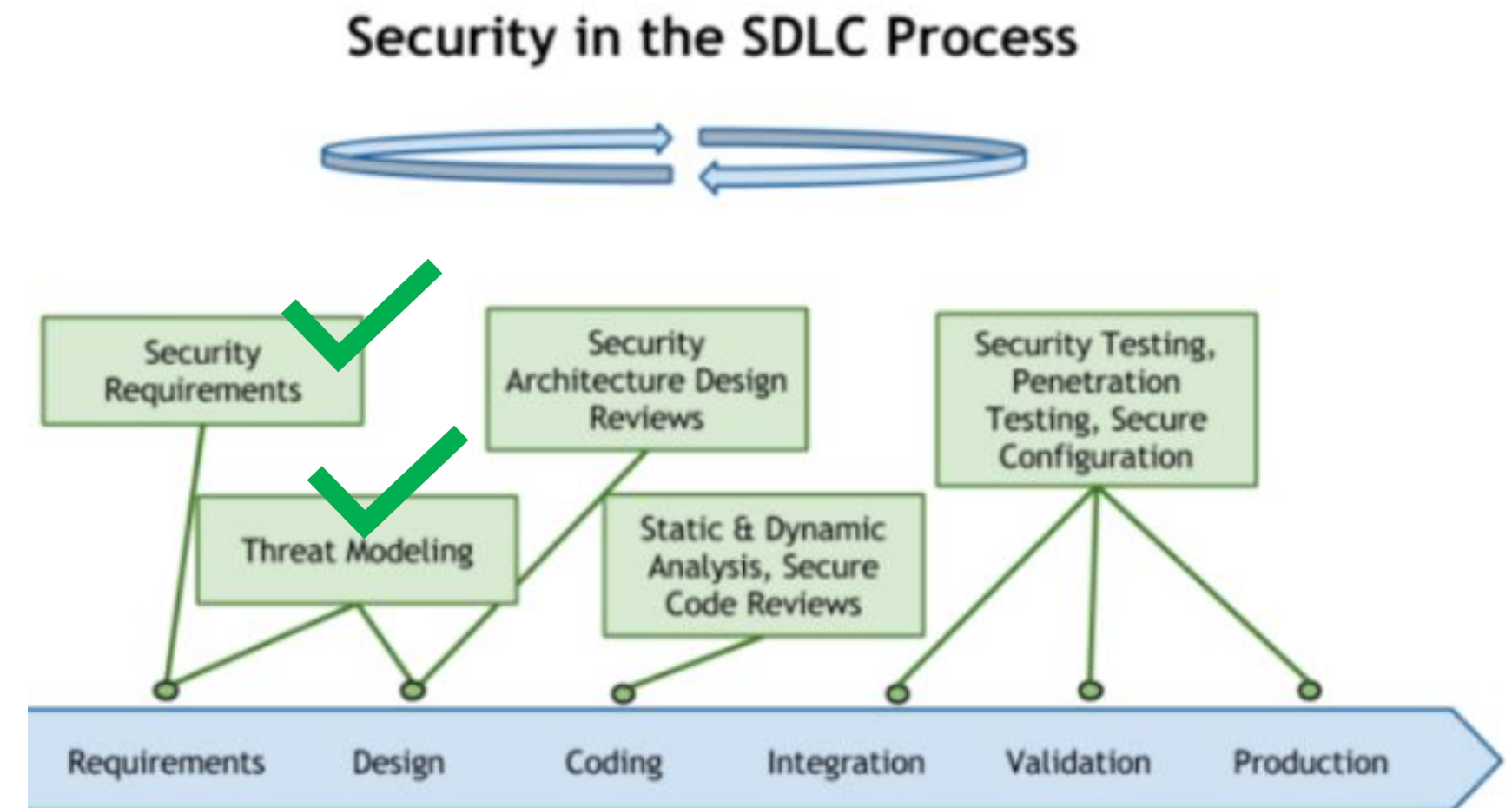
- El primer paso sería establecer requerimientos y controles de seguridad (**Security requirements**) para el ciclo de desarrollo de software los cuales tienen que poder cuantificarse.
- Control de autenticación**
 - creación de usuarios con contraseña robusta.
- Control de roles y privilegios**
 - cada usuario tan solo debe tener acceso a lo necesario dentro de la aplicación.
- Requerimientos orientados al riesgo.**
 - requerir contraseñas y métodos de autenticación fiables.
- Aprobación de privilegios**
 - mínimo privilegio, autorizar a cada usuario el acceso imprescindible
- Documentos con los requerimientos de seguridad.



<https://betabeers.com/blog/ciclos-vida-del-software-seguro-s-sdlc-346/>

Fases SSDLC

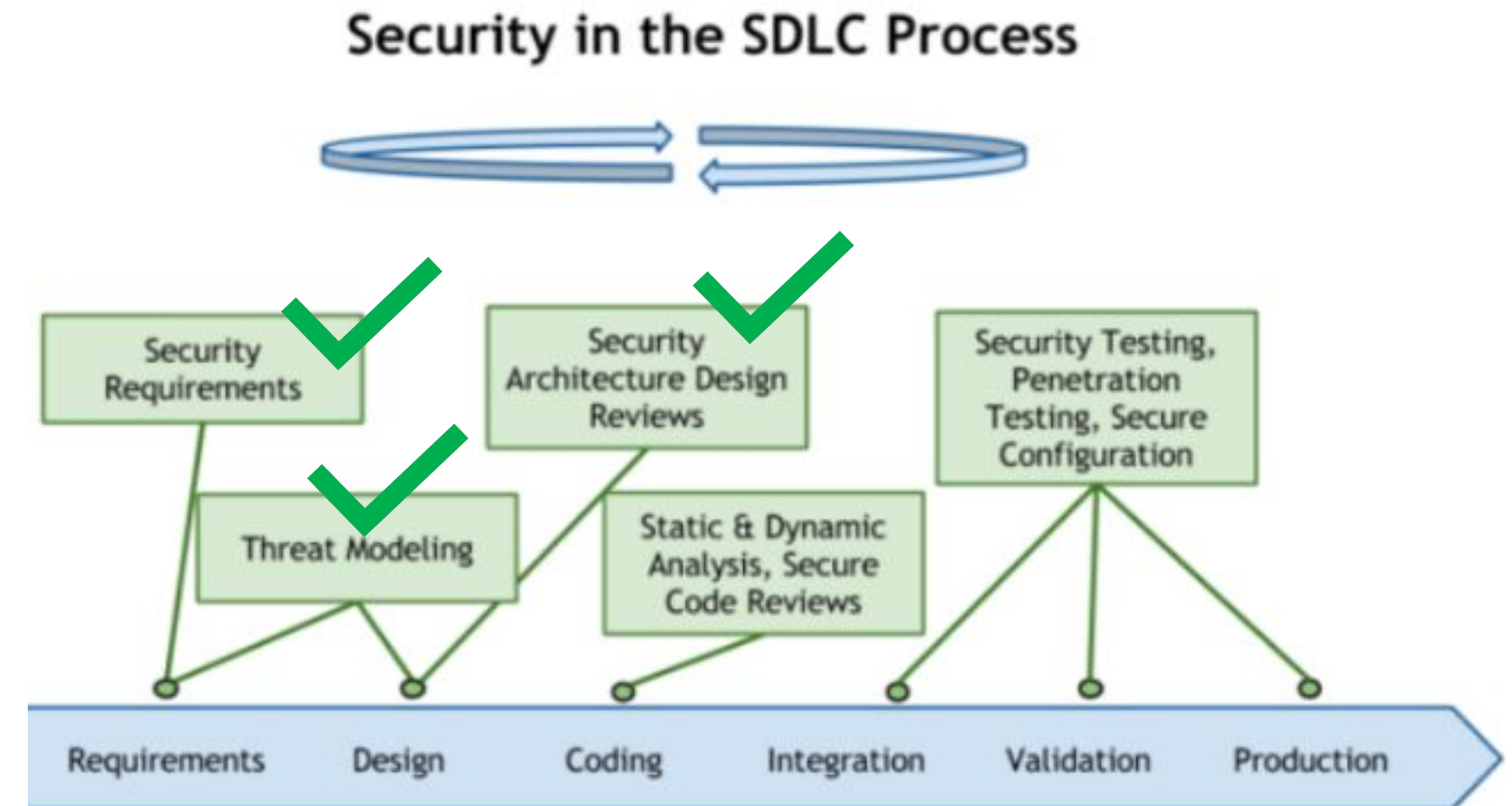
- La siguiente fase sería de análisis y diseño de las amenazas (**Threat Modeling**) puesto que muchos fallos de seguridad en las aplicaciones fueron detectados en la fase de diseño.
- Diseño seguro de mensajes de error**
 - no se debe poder interactuar con los mensajes de error por que pudieran dar acceso a la aplicación.
- Diseño seguro de autenticación**
 - para evitar errores como los de tener acceso con 1=1 y descartar eso datos de entrada.
- Diseño de autoría y login**
 - Sería un diseño centrado en usabilidad, pero sin dejar de lado el no acceso sin contraseña a los servicios.
- Análisis de riesgo**
 - analizar la aplicación en busca de posibles riesgos que pudieran aparecer en esta fase de análisis y diseño.
- Documentos con el análisis y pruebas de seguridad



<https://betabeers.com/blog/ciclos-vida-del-software-seguro-s-sdlc-346/>

Fases SSDLC

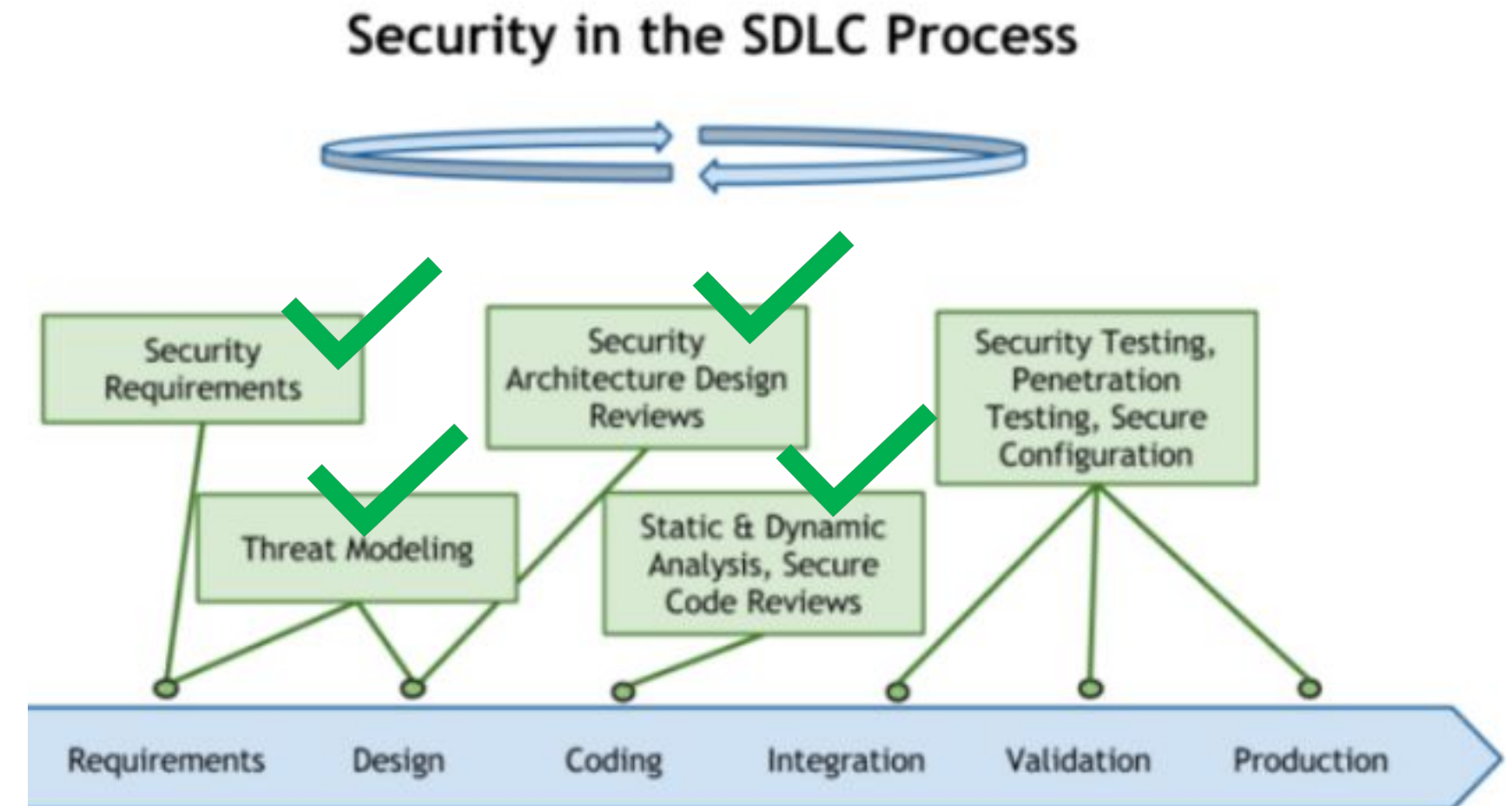
- Las faltas en la codificación pueden comprometer la aplicación y los datos, pero hacer una revisión de la arquitectura en la que va a estar instalada es imprescindible (**Security architecture design review**) por lo que un mal diseño físico del ambiente TI en el que estará implementada la solución es una debilidad de este.
- Alguna de las revisiones que se realizan son:
 - Segregación de componentes (redes, Servidores, etc.).
 - Validar los datos de entrada antes de procesarlos
 - Controlar el tamaño y tipos de datos de entrada que se ingresan en formularios
 - Eliminar caracteres especiales.
 - No mezclar datos con código para evitar errores de procesamiento.
 - Reemplazar SQL dinámicas por Stored procedures.
- Documento con las reglas a cumplir en el desarrollo y la arquitectura.



<https://betabeers.com/blog/ciclos-vida-del-software-seguro-s-sdlc-346/>

Fases SSDLC

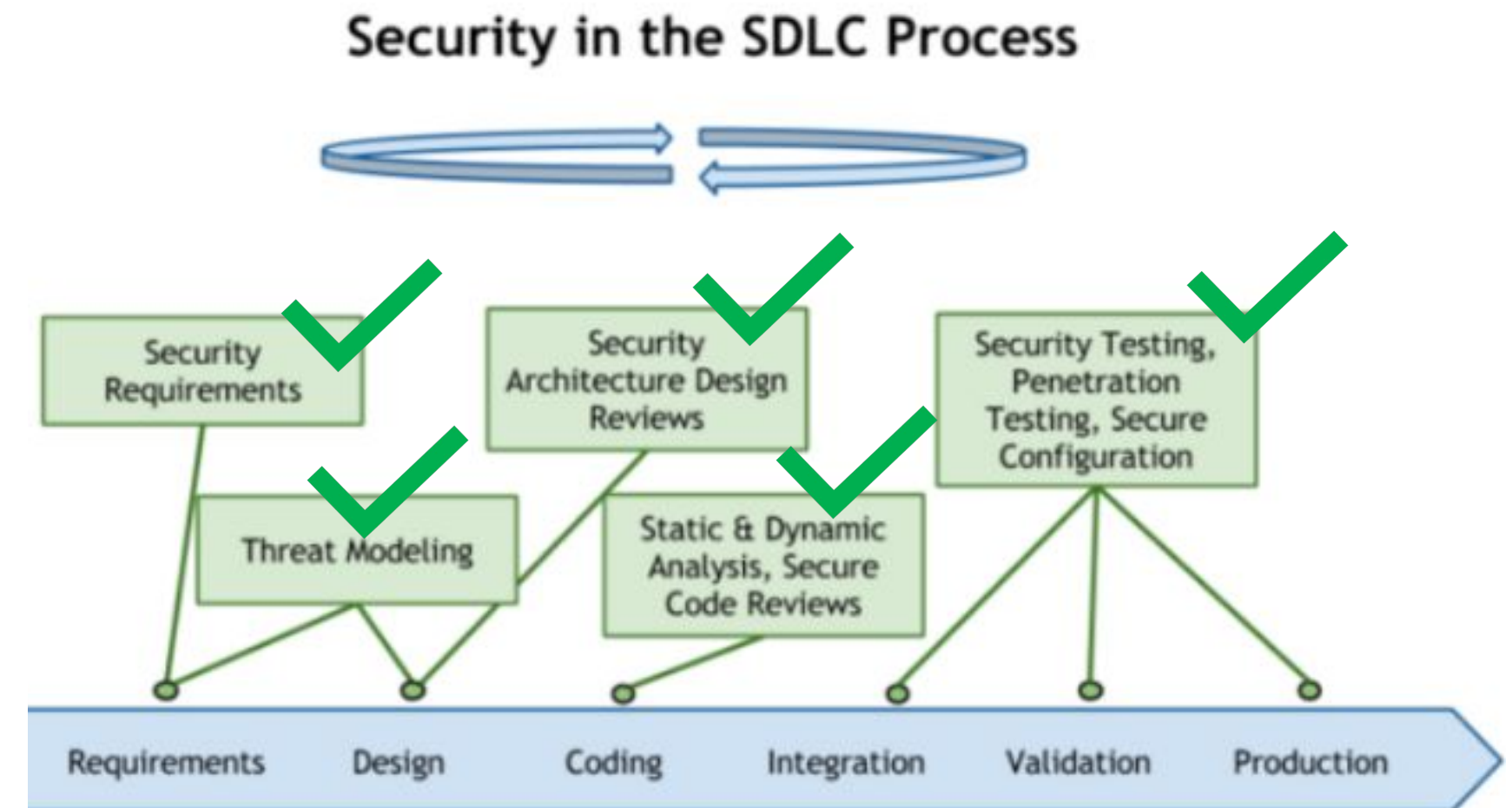
- Las faltas en la codificación pueden comprometer la aplicación y los datos, por lo que el análisis de código estático y Dinámico (**Static & dynamics, analisis secure code reviews**) es importante, ya que trataría como su nombre indica de un análisis de los códigos escritos por los desarrolladores para que esta solución tecnológica pueda funcionar .
- Mirando el código sin que la aplicación se esté ejecutando
- Análisis con la aplicación ejecutándose.
- Subsanar los hallazgos encontrados en las pruebas
- Documentar cambios realizados.



<https://betabeers.com/blog/ciclos-vida-del-software-seguro-s-sdlc-346/>

Fases SSDLC

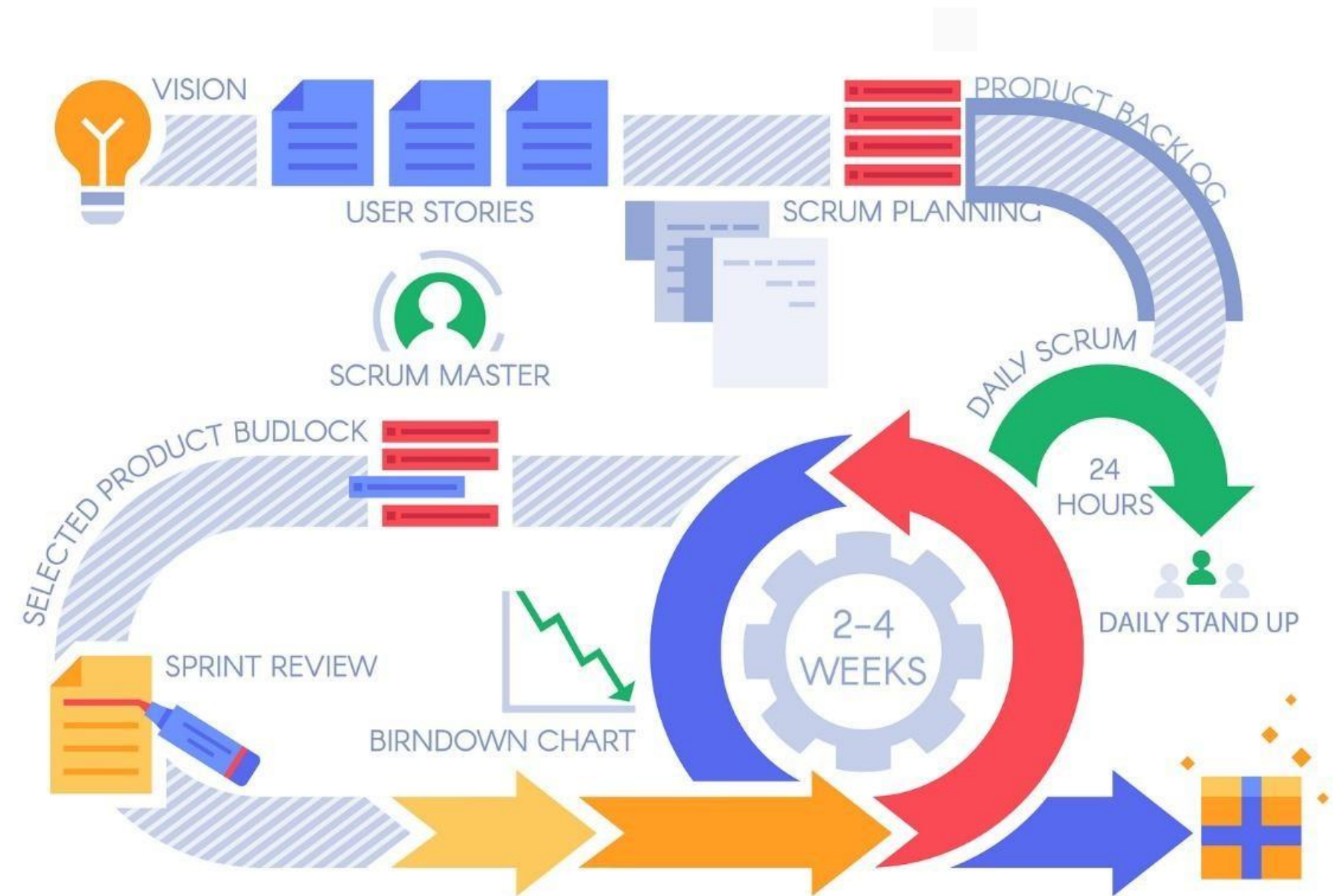
- El Test de penetración y seguridad (**Security testing, Penetration testing, secure configuration**), tiene la finalidad de realizar las pruebas necesarias para establecer que la solución es lo más segura posible y que reduce los riesgos que esta tengan con respecto a ciberseguridad.
- Realizar una auditoría de hacking ético de seguridad con la finalidad de detectar las vulnerabilidades
- Subsanas los hallazgos encontrados en la auditoria, realizando por último una configuración de seguridad.
- Documentar cambios realizados.



<https://betabeers.com/blog/ciclos-vida-del-software-seguro-s-sdlc-346/>

Metodología AGILE

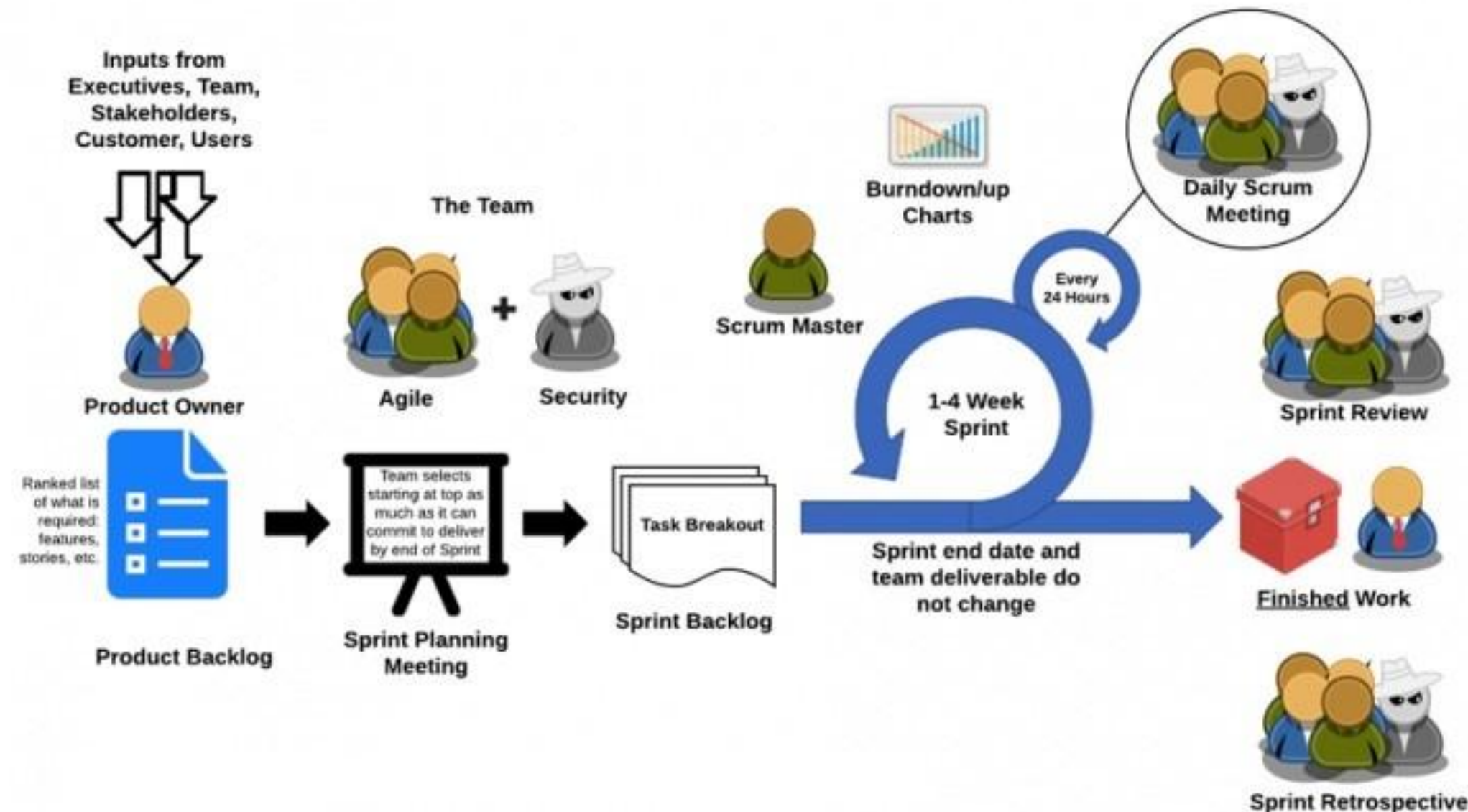
- Las metodologías ágiles son enfoques para el desarrollo de productos que están alineados con los valores y principios descritos en el Manifiesto ágil para el desarrollo de software.
 - Entrega de valor
 - Colaboración de negocio
 - Dinámica del equipo y cultura
 - Retrospectivas y aprendizaje continuo
- El objetivo es entregar el producto, con unas entregas incrementales y frecuente de funcionalidad,
- Se organizan pequeños equipos multifuncionales autoorganizados, permitiendo la retroalimentación frecuente de los clientes.
- Algunas Metodologías Agiles son:
 - **SCRUM**
 - **KANBAN**
 - **LEAN**
- <https://www.digite.com/agile/agile-methodology/>



<https://www.digite.com/agile/scrum-methodology/>

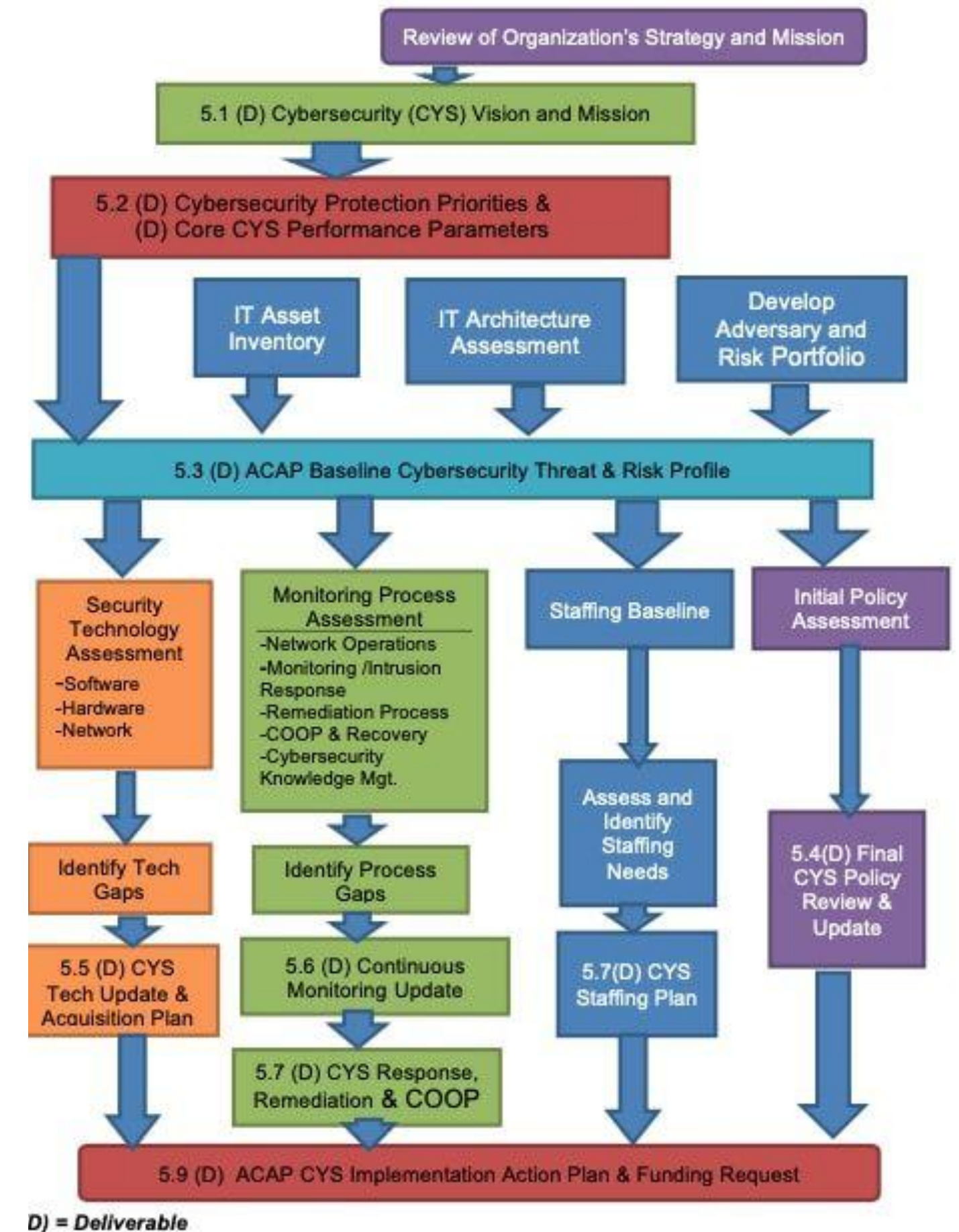
Metodología ágiles y SSDLC

- **ACAP (Agile Cybersecurity Action Planning o plan de acción de ciberseguridad).**
 - **ACAP** sienta las bases para una “Cultura organizativa de seguridad cibernética adaptativa” a los desafíos complejos que surgen continuamente.
 - **ACAP** es un enfoque nuevo, dinámico y holístico para alinear rápidamente la estrategia de ciberseguridad de la organización, la capacidad técnica y organizativa, los procesos y las políticas para abordar el universo de amenazas y riesgos cibernéticos de hoy en día
 - Se repite en ciclos de 1 a 6 meses, al igual que los "**Sprints**" de **Agile Development** o "según sea necesario".



Metodología ágiles y SSDLC

- El proceso **ACAP** utiliza una sofisticada fusión paradójica de conceptos específicos de varios campos:
 - Gestión de riesgos
 - Planificación estratégica adaptativa
 - Mejora de procesos
 - Metodología Agile
 - Buenas Prácticas de Ciberseguridad
 - Colaboración creativa
- El corazón del Proceso **ACAP** es un taller de estrategia de amenaza / riesgo en el que un equipo técnico y de liderazgo de múltiples niveles y funciones:
 - Crear un perfil de amenaza / riesgo en continua evolución
 - Reevaluar rápidamente la infraestructura de ciberseguridad
 - Identificar los deltas / problemas sistémicos antes de que ocurran.
 - Crear un plan de acción para remediar los deltas / problemas a través de cambios en las políticas de ciberseguridad y actualizaciones de estrategias, personal, procesos y tecnología.
- <https://ciberseguridad.blog/ciberseguridad-agil-con-agile-cybersecurity-action-planning-acap/>





Lorem ipsum Dolor sit amet, consectetur Adipiscing Elit. Etiam eget quam

lacus.