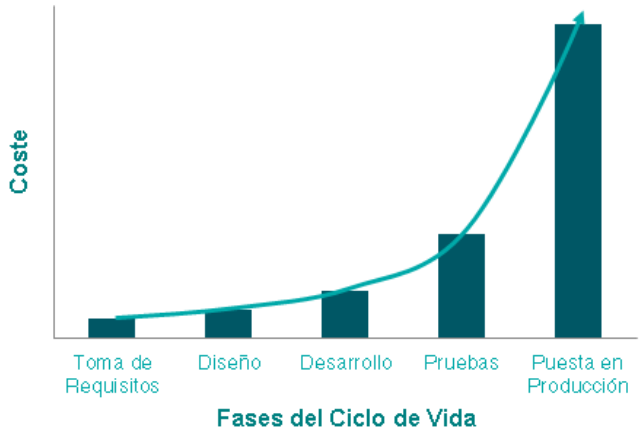
Seguridad en SDLC

El Ciclo de Vida de Desarrollo de Software, o lo que se conoce con sus siglas **SDLC**, es todo el proceso que tiene cualquier desarrollo nuevo en una organización desde que se tiene la idea del mismo hasta que está ya implantado en producción.

En los últimos tiempos lo que se busca es introducir seguridad en este proceso, ya que todo desarrollo tiene sus fases principales (como se muestra en la siguiente imagen en la parte de la izquierda), lo que se intenta introduciendo la seguridad es definir unas fases paralelamente para a la vez que se va ejerciendo el trabajo se vaya considerando la seguridad en el mismo (como se muestra en la imagen de la derecha):

En las imágenes anteriores podemos ver que cada fase tiene su correspondiente fase de seguridad, por ejemplo, en el momento que se están realizando pruebas del correcto funcionamiento de la aplicación, se debería de realizar una auditoría dinámica de la misma en la que se puedan ver las vulnerabilidades antes de la fase de producción.

La **implementación de medidas de seguridad** debe hacerse **desde el inicio** del Ciclo de Vida del Desarrollo del Software, ya que el coste de solucionar cualquier problema de seguridad es mayor cuanto más tarde se detecte, como se observa en el siguiente gráfico:



Al final en muchas organizaciones o departamentos de desarrollo para no implementar seguridad en los mismos, solemos poner muchísimas excusas que obviamente no son validas como por ejemplo las siguientes:

- Nadie sabe cómo funciona, por ende, no la van a atacar. (*Un atacante invertirá el tiempo que necesite para saber como funciona…*)
- Si no se encontraron vulnerabilidades hasta ahora (*Un atacante encontrara una o varias vulnerabilidades con tan solo un vistazo en la misma…*)
- A nadie le interesaría atacar nuestra aplicación. (*Existen millones de bots o sistemas automáticos escaneando constantemente en búsqueda de deficiencias de seguridad…*)
- La aplicación es segura porque corre detrás de un firewall. (*Obviamente si tiene un elemento hardware de seguridad es más segura, pero este hecho no implicara nunca que sea segura…*)
- La aplicación es segura porque usa https. (*En este caso la comunicación es segura pero el desarrollo no tiene porque serlo…*)
- Si no corre como Administrator / root, no puedes hacer nada peligroso. (*Obviamente esto es una buena práctica de seguridad, pero que el usuario tenga pocos privilegios no indica que la aplicación sea segura…*)

Entre todas las excusas posibles para no implementar la seguridad en el SDLC la que más se suele escuchar el "**No hay tiempo para incluir la seguridad**", ya que la mayoría de los desarrollos van muy ajustados de tiempo y es más importante poner algo en producción en el plazo adecuado que incluir la seguridad en cada una de sus partes.

Hay que pensar que "Más vale prevenir que curar", por lo que es mucho más productivo y menos comprometedor evitar o detectar un ataque para el desarrollo que restaurar el estado tras un ataque exitoso.

Para que no ocurra esto que se ha comentado anteriormente y para desarrollar de forma segura, existen varios modelos de seguridad para ser implementados, uno de los más recomendables que se suele implantar es el **Modelo de Madurez para el Aseguramiento de Software** *(**SAMM** – Software Assurance Maturity Model)* es un marco de trabajo abierto y flexible desarrollado por OWASP.

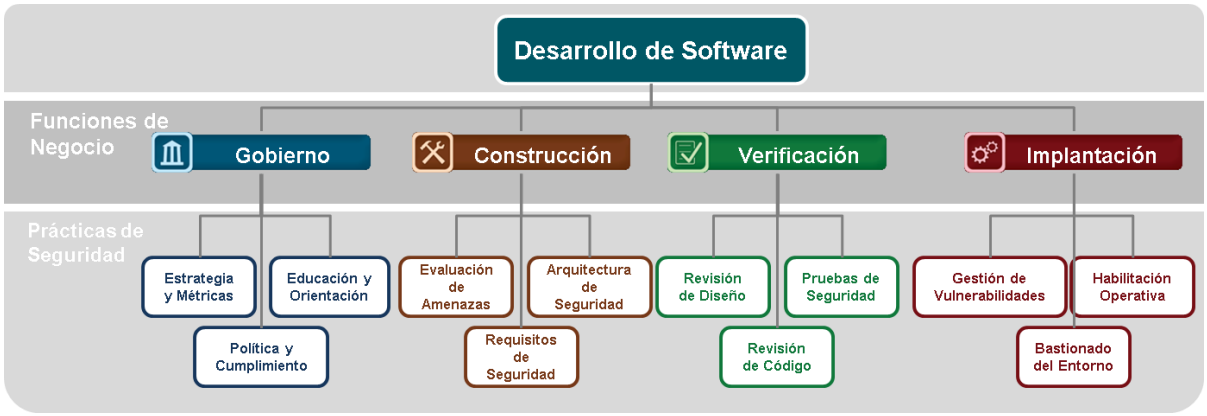
Este modelo sirve para ayudar a las organizaciones a formular e implementar una **Estrategia de Seguridad para el Desarrollo de Software** que sea adecuada a las necesidades específicas de cada organización y tiene, entre otras, las siguientes características:

- Consiste en **varios ciclos de corta duración** (objetivos asequibles) e **iteraciones incrementales** (objetivos finales).
- **Flexible** y **personalizable**, basándose en la tolerancia a riesgos, criticidad de las aplicaciones desarrolladas y la metodología de desarrollo…
- Actividades de Aseguramiento **simples**, **medibles** y **bien definidas**.

El mismo esta diseñado para ayudar a:

- **Evaluar** las Prácticas de Seguridad en el Desarrollo de Software existentes en la organización.
- **Construir** un Programa de Aseguramiento del Software balanceado en iteraciones bien definidas.
- **Demostrar** mejoras concretas en el Programa de Aseguramiento del Software.
- **Definir y medir** las Actividades relacionadas con Seguridad de Desarrollo de Software en la organización.

Todas estas metas se definen según la estructura de la siguiente imagen en la que se pueden observar las diferentes partes que lo componen y la interrelación entre ellos:



Como se puede observar en la imagen esta metodología **OpenSAMM** establece:

- **4 Funciones de Negocio** relacionadas al Desarrollo del Software
- **3 Prácticas de Seguridad** por cada Función de Negocio
- **3 Niveles de Madurez** por cada Práctica de Seguridad
- **2 Actividades de Seguridad** por cada Nivel de Madurez

Por ende, se aconseja a todas las empresas, realizar esta estructura a nivel general siempre como normativa interna y aplicarla a todos los desarrollos que se hagan en las mismas. Una vez que la misma se encuentre implementada, toda la seguridad se realizara de forma autónoma y entrara dentro de los procesos habituales continuos de la compañía.

Para más información sobre este modelo y como es la mejor forma de implementación en las compañías podemos ver la última versión de la misma en castellano en la siguiente dirección web:

http://www.opensamm.org/downloads/SAMM-1.0-es_MX.pdf

Fernando Saavedra

Cybersecurity Manager

Áudea Seguridad de la Información