

FALENCIAS DEL DISEÑO DE APLICACIONES

El diseño de muchas aplicaciones de software comienza como una imagen o idea en la mente de sus diseñadores. Este diseño puede tener falencias que tiende a degradarse. Hay cuatro síntomas primarios a tener en cuenta los cuales nos indican cuando un diseño de software tiende a degradarse:

- a. **Rigidez:** Tendencia del software a ser difícil de cambiar. Un diseño rígido, involucra que al cambiar los requerimientos del diseño deberán realizarse cambios muy grandes del mismo. Significa que el diseño no tiene la capacidad de separarse en módulos independientes.
- b. **Fragilidad:** Cuando al producir un cambio en alguna parte de un software, dicho cambio ocasiona cambios en otros sectores del software. Se dice que el software es frágil. Tal software causa la sospecha de administradores y consumidores de que los desarrolladores han perdido control de su software. Genera la desconfianza y se pierde la credibilidad.
- c. **Inmovilidad:** Inhabilidad de reusar software. Es decir el software debe ser lo suficientemente flexible como para promover el reuso del software. Por ejemplo: un ingeniero descubre que puede utilizar módulos que ya ha diseñado pero debido a la inmovilidad de su diseño deberá volver a reescribir dicho modulo para este caso específico. Cuando un modulo resuelve un problema que puede llegar a aparecer en otro momento es muy importante tener en cuenta el concepto de inmovilidad.
- d. **Viscosidad:** Enfrentados a un cambio, los ingenieros usualmente encuentran más de una forma de realizarla. Cuando los métodos de preservar el diseño son más difíciles de emplear que los métodos que no preservan el diseño (haks), la viscosidad del diseño es elevada. Los requerimientos van cambiando en formas que el diseño original no anticipaba. Se debe encontrar alguna manera de que los diseños sean resistentes a tales cambios y protegerlos de la decadencia. Se puede también encontrar o definir la viscosidad del entorno que se da cuando el medio es lento e ineficiente. Ejemplo: Si un modulo tiene largos tiempos de compilación y un ingeniero desea hacer cambios sobre este, dichos cambios no tienen que traer largas recompilaciones. Si un proceso necesita mucho tiempo para chequear unos pocos archivos, entonces si un ingeniero debe introducir algún cambio, el mismo debe realizarse con el mínimo de chequeos posibles.

En orden de prevenir la degradación de la arquitectura de dependencia, las dependencias entre módulos en una aplicación deben ser manejadas mediante una buena limitación o restricción de las dependencias entre módulos para así evitar que la decadencia se propague. Los módulos deben ser lo más cohesivos (unidos) posible y lo menos dependientes posible. Los cambios de requerimiento conllevan a consecuencias no anticipadas, por lo cual debemos intentar que nuestros diseños eviten los síntomas de decadencia mencionados. Con esto lograremos que si es necesario algún cambio del mismo, pueda ser realizado en forma rápida y sin necesidad de estar familiarizado con la totalidad del diseño.