

Modernización de la Arquitectura

1. Autoría

- Claudia Nistal Martínez UO294420
- Marcelo Díez Domínguez UO293820
- Marcos Llanos Vega UO218982

2. Modernización de la Arquitectura

A medida que se desarrollan los sistemas software, estos se van volviendo obsoletos, además de que van apareciendo nuevas tecnologías. La modernización de la arquitectura trata de eliminar las desventajas de los sistemas antiguos, mediante el uso de las prácticas modernas.

■ Modernización de la Arquitectura en Empresa

Existen dos tipos de procesos de modernización:

- El ideal, donde se hacen cambios incrementales de manera continua, controlando así una baja deuda técnica.
- El real mayoritariamente, donde se hacen cambios a gran escala, requiriendo una pausa del desarrollo para la modernización forzada y bajo un alto coste.

■ Proceso de Modernización

La modernización no se trata de un tema “fijo”, sino que es un proceso cambiante. Es decir, cuando un sistema se convierte en un cuello de botella, se plantea la modernización para solucionar problemas, y cuando estos se han solventado, la modernización pasa a un “segundo plano”.

Como todas las empresas quiere tener un sistema actualizado con las mejores implementaciones, el proceso de modernización se convierte en una inversión y no en un costo.

3. Arquitectura del Software Sociotécnica

Dentro del ámbito de la modernización de la arquitectura aparecen dos tipos de aspectos importantes, los sociales y los técnicos. Ambos aspectos deben estar bien desarrollados, derivando el problema real a saber encontrar una optimización conjunta que abarque la organización y el software.

■ Ley de Conway

“La forma de la estructura en equipo sigue la forma del sistema que se está construyendo”

4. Análisis del Sistema a Modernizar

Se deben revisar todos los aspectos del sistema. Para ello, se analiza: tecnología, diseño, funciones, estructura y habilidades del equipo. Es importante hacerse la siguiente pregunta: “Si modernizamos todo el sistema, ¿cuánto valor generaríamos y cuánto nos costaría?” Para calcularlo se hace uso del Retorno de Inversión (ROI). No hay unas reglas fijas que seguir para la modernización.

■ Puntos a tener en cuenta para la modernización

Se deben de tener en cuenta los siguientes puntos: estrategia (mapeo de Wardley), el estado actual, el estado futuro, desacoplar la arquitectura y migrar el estado actual a futuro. Para modernizar primero se debe conocer bien qué partes se deberían cambiar y por qué, y después ya avanzar hacia el cómo se va a hacer.

5. Ingeniería de Datos y Arquitectura

Los datos son imprescindibles en cualquier arquitectura, ya que sin ellos no se podría hacer nada. Por tanto, de nada sirve modernizar el resto del sistema, si los datos se dejan sin modernizar.

A diferencia de otros equipos de desarrollo, los de ingeniería de datos suelen ser equipos muy centralizados y su trabajo no es escalable. En ellos, los aspectos en los que se profundiza en otras partes del software no se están aplicando lo suficiente.

Sin embargo, poseen gran impacto, ya que hay veces donde los datos son la motivación del cambio, como por ejemplo cuando una empresa migra su instalación local a la nube y necesita hacer cambios ya que el proceso que estaba optimizado para la instalación local, puede no serlo para la nube.

6. Razones para modernizar o no un sistema

No siempre modernizar un sistema ya existente es la mejor opción. A continuación, se pueden ver algunas razones que llevarían a realizar este proceso y otras que no.

| Razones para SÍ modernizar | Razones para NO modernizar |
|---|---|
| Ahorrar dinero porque los costes diarios de soporte y mantenimiento son elevados | No se quiere perder tiempo ni dinero en algo que nos sea añadir nuevas funciones al sistema |
| Ganar más dinero porque no se pueden añadir nuevas funcionalidades ya que requieren mucho tiempo y dinero | Es más fácil, rápido e incluso barato deshacerse del sistema viejo y adquirir uno nuevo |
| Obligación legal por entidades reguladoras como la RGPD o los datos sanitarios | |

7. Arquitectura evolutiva

Es un tipo de arquitectura que, idealmente, permite crear sistemas que se ajustan para adaptarse a las necesidades cambiantes del negocio. Estos sistemas tienen un acoplamiento mínimo lo cual permite realizar cambios con pequeñas modificaciones sin afectar al resto.

En la realidad muy pocas empresas lo aplican, solo un 3%, y además de éstas que lo aplican la mayoría lo hacen sólo parcialmente en alguna de sus partes, pero no en todas. Estas empresas que sí la aplican poseen todas unas características comunes:

- Tienen equipos de desarrollo autónomos.
- Utilizan prácticas de programación extrema con TDD y programación por parejas.
- Están centrados en la mejora continua de su trabajo.
- Tratan de refactorizar el código continuamente.
- Sus trabajadores intentan aprender cosas nuevas.
- La empresa les aporta nuevas formaciones a los mismos.

Resumiendo, cuentan con equipos de desarrollo que aplican la máxima calidad a lo que hacen, pero no se limita a ellos. El CTO tiene que entender y aplicar esa calidad y el CEO debe confiar en ellos. Al principio debe confiar, después cuando surge un problema y se soluciona con pocos cambios rápidamente, son ya los resultados los que avalan esta forma de trabajo. Así no se necesitan luego hacer grandes labores de modernización del sistema, que paren el resto de trabajos.

8. Equilibrio entre la modernización y desarrollo del resto del negocio

Lo primero a tener en cuenta es que todo el tiempo que se dedique a tareas de modernización no se podrá dedicar a otras labores como puede ser añadir nuevas funcionalidades al sistema. Por tanto, hay que encontrar el equilibrio entre ambas cosas. Se trata de pensar más en el futuro a largo plazo y no limitarse al presente. Además, es importante que toda la empresa esté de acuerdo y reme en la misma dirección, ya que si no fracasarán.

9. Éxito o fracaso de la modernización

En todos los procesos de modernización se dan unos casos que normalmente llevan al éxito o al fracaso del mismo.

| Causas de éxito | Causas de fracaso |
|--|--|
| Hay una necesidad clara e inevitable de modernizar el sistema | No hay necesidad y se va posponiendo para realizar otras funciones |
| Todas las partes involucradas la apoyan | Hay partes que no la apoyan o no la saben aplicar |
| Se ha analizado detenidamente el sistema inicial previamente | No se analizó el sistema original y luego aparecieron problemas por el camino |
| Se planifico con todo detalle el proceso de migración, para completar el proceso y no dejarlo a medias | No se planifico, y al aparecer problemas, se deja a medias, para seguir el desarrollo de otras funciones |

Si no se completa el proceso y se deja a medias, no sólo deja de ser útil, sino que además es contraproducente, ya que, en lugar de un sistema antiguo, habrá dos sistemas (una parte del antiguo y otra del nuevo), haciendo que sea peor aún que el original.

10. Referencias

Para más información se puede escuchar el podcast de Software Engineering Radio Podcast donde Giovanni Asproni entrevista a los autores del libro "Architecture Modernization" Nick Tune y Jean-Georges Perrin (<https://se-radio.net/2024/10/se-radio-638-nick-tune-and-jean-georges-perrin-on-architecture-modernization/>).