

TEMA 100. LA MIGRACIÓN DE APLICACIONES EN EL MARCO DE PROCESOS DE AJUSTE DIMENSIONAL Y POR OBSOLESCENCIA TÉCNICA. GESTIÓN DE LA CONFIGURACIÓN Y DE VERSIONES. GESTIÓN DE ENTORNOS.

Actualizado a 09/10/2020

1. LA MIGRACIÓN DE APLICACIONES EN PROCESOS DE AJUSTE DIMENSIONAL Y POR OBSOLESCENCIA

El **ajuste dimensional**, consiste en aligerar la carga de trabajo que soporta un gran ordenador (*mainframe*) trasladándola a otro tipo de ordenadores de menor coste.

Downsizing: Mover una aplicación o parte de ella de un *mainframe* a una plataforma menos costosa.

Upsizing: Integración en entornos de red de aplicaciones y ordenadores que estaban aislados, para compartir datos. Ej integración de BBDD aisladas en un servidor único de BD.

Rightsizing: Elección de la plataforma más apropiada para las actividades informáticas de una organización (*mainframe*, arquitectura cliente-servidor, etc).

Las **aplicaciones** obsoletas o con riesgo de obsolescencia suelen denominarse aplicaciones **legacy**, legadas, o heredadas.

Causas de obsolescencia: obsolescencia hardware, fin del soporte del fabricante, código fuente no adaptable, complejidad del código, reducción de costes al pasar de aplicación propietaria a sw libre.

Tipos de migraciones de aplicaciones:

- **Refronting o refacing:** modificar la interfaz gráfica sin reescribir la aplicación completa.
- **Replacement** (sustitución): sustituye la aplicación completas por una o varias aplicaciones.
- **Rehosting** (re-alojamiento): mover la aplicación desde un entorno *hardware legacy* a un entorno más moderno, sin modificar el código de la aplicación o su arquitectura.
- **Rearchitcting** (re-arquitectura o reescritura): desarrollar de nuevo la aplicación utilizando nuevos paradigmas de programación, (pj.: desarrollo web, orientación a objetos, etc).
- **Interoperation** (interoperabilidad) o wrapping: encapsular la aplicación o algunos de sus componentes para que sean utilizados por otras aplicaciones o infraestructura con tecnología más moderna.
- **Retirada.**
- **Migración a la nube:** engloba a varias de las anteriores, pj.: un *re-hosting* a un entorno *cloud*, o una sustitución por una aplicación SaaS.

2. GESTIÓN DE LA CONFIGURACIÓN

Mantiene una imagen detallada de la situación de los activos y de sus relaciones a lo largo del tiempo.

2.1. MÉTRICA V3 – INTERFAZ GC

La Gestión de la Configuración es una de las cuatro interfaces de la Metodología Métrica v3. Existen tres escenarios en los que entra en juego la Gestión de la Configuración en Métricav3:

- **Estudio de Viabilidad del Sistema (EVS):**
 - Durante la actividad EVS3 Definición de Requisitos del Sistema, tiene lugar la actividad **EVS-GC1 Definición de los requisitos de Gestión de la Configuración**.
 - La actividad **EVS-GC2 – Establecimiento del plan de Gestión de la Configuración** tiene lugar tras la tarea EVS6 – Selección de la Solución.

- Durante los procesos de ASI, DSI, CSI e IAS
 - La actividad **GC 1 Identificación y registro de productos**
 - La actividad **GC 2 Identificación y registro del producto global**
- **Mantenimiento del Sistema de Información (MSI)**
 - La actividad MSI-GC 1 Registro del cambio en el sistema de gestión de la configuración.

2.2.ITIL 4 – PRÁCTICA GESTIÓN DE LA CONFIGURACIÓN

Propósito: garantizar que disponemos de información precisa y confiable sobre la configuración de servicios, de los elementos de configuración (CI) que los respaldan y sobre sus relaciones y dependencias.

Términos relevantes:

- **CI:** cualquier componente que requiera administración y sea necesario para proporcionar un servicio de TI: hw, sw, redes, edificios, personas, proveedores y documentación. Los servicios también se tratan como elementos de configuración.
- **CMS:** Sistema de Gestión de la configuración. Grupo de herramientas, datos e información que se utiliza para apoyar la práctica de gestión de configuración del servicio.
- **CMDB:** Base de datos usada para guardar los registros de configuración a lo largo de su ciclo de vida. Mantiene también las relaciones entre los registros de configuración.
- **Línea base de configuración:** Es la configuración de un producto, servicio o infraestructura que se ha revisado formalmente. Sirve como base para otras actividades como las de desarrollo y planificación.

Los **factores clave** en esta práctica son:

- Asegurar que la organización dispone de la información de configuración relevante sobre sus productos y servicios.
- Asegurar que el coste de proveer la información de configuración se optimiza de forma continua.

2.3.GESTIÓN DE LA CONFIGURACIÓN EN ESTÁNDARES Y METODOLOGÍAS

Algunos estándares y metodologías generales relativos a la gestión de la configuración:

- CMMI – Configuration Management.
- ISO 10007:2017 Quality management - Guidelines for configuration management.
- 828-2012 IEEE Standard for Configuration Management in Systems and Software Engineering.

3. GESTIÓN DE VERSIONES

3.1.ITIL 4 – PRÁCTICA GESTIÓN DE LIBERACIONES O VERSIONES

Propósito: hacer los servicios nuevos y modificados adecuados y disponibles para su uso por los usuarios. Es necesario acordar con el cliente una correcta planificación de las versiones.

La práctica de gestión de liberaciones puede adoptar un enfoque tradicional, donde la gestión de versiones y despliegue se combinan en un único proceso, o bien un enfoque ágil, donde el despliegue se

hace en pequeños incrementos de la versión. En un entorno DevOps, la gestión de versiones está integrada con la integración continua y las herramientas para la entrega continua.

3.2.GESTIÓN DE VERSIONES Y RELEASES EN EL DESARROLLO SOFTWARE

Terminología asociada a la gestión de versiones:

- **Repositorio:** servidor de versiones y ficheros que conforman el código fuente de la aplicación.
- **Check-out:** descargar una copia del código fuente a un equipo local desde el servidor.
- **Update:** actualizar la versión local del código con la última versión disponible en el servidor.
- **Commit (o check-in):** convertir cambios temporales en definitivos, consolidando en el servidor.
- **Diff (o Delta):** diferencias entre dos versiones (código añadido, eliminado, modificado).
- **Delta compression:** se almacenan en el repositorio únicamente las diferencias entre versiones.
- **Branch (rama):** Familia de modificaciones independiente de otra. Cuando se crea una nueva rama, se duplica la versión actual del código fuente, y esta se modifica de forma independiente.
- **Merge:** fusionar una rama con otra rama (generalmente con la rama principal).
- **Fork (bifurcación):** crear una nueva rama que nunca volverá a ser fusionada con la principal.
- **Push:** enviar el código a otros servidores tras hacer el commit en local.
- **Pull:** descargar una copia de código de un servidor y sustituir a la local.
- **Fetch:** descargar una copia de código del servidor al equipo local sin sustituir la copia local.

Integración continua: Permite obtener versiones de la aplicación con mejoras incrementales en periodos cortos de tiempo. Hace uso de herramientas de automatización de la compilación y pruebas como Jenkins o Hudson, que pueden integrarse con herramientas de gestión de versiones.

3.3.HERRAMIENTAS PARA GESTIÓN DE CONFIGURACIÓN Y DE VERSIONES

Herramientas orientadas al gobierno de los sistemas de información (ITIL):

- **BMC Remedy:** Cuenta con funcionalidad completa de CMDB (BMC Atrium) y otros módulos.
- **EasyVista Service Manager:** La CMDB forma parte de una suite ITSM.
- **ServiceNow:** Igualmente la CMDB como parte de la suite de módulos ITSM.

Herramientas orientadas al desarrollo:

- De **gestión centralizada o modelo cliente-servidor:**
 - **CVS** (Concurrent Versions System): Escrito en C y con licencia GPL.
 - **SVN** (Apache Subversion): Escrito en C y con licencia Apache. Más moderno que CVS.
 - **IBM Rational Clear Case:** Licencia propietaria.
 - **Perforce Helix:** Licencia propietaria.
- De **modelo distribuido:**
 - **Git:** Escrito en múltiples lenguajes, licencia GPL y LGPL. Goza de gran popularidad.
 - **Mercurial.** Escrito en Python y C, licencia GPL.
 - **GNU Bazaar:** Escrito en Python y C, licencia GPL. Permite modo distribuido y modo cliente-servidor.

4. GESTIÓN DE ENTORNOS

- Tipos de entornos: Entorno local, entorno de desarrollo, entorno de integración, entorno de pruebas, entorno de pre-producción, entorno de producción.
- Actividades típicas de la gestión de entornos: reserva de entornos, limpieza de entornos, coherencia entre entornos, gestión de la capacidad de los entornos.