



## **TEMA 102. GESTIÓN DE CAMBIOS EN PROYECTOS DE DESARROLLO DE SOFTWARE. GESTIÓN DE LA CONFIGURACIÓN Y DE VERSIONES. GESTIÓN DE ENTORNOS.**

Actualizado a 28/09/2020

## 1. GESTIÓN DE CAMBIOS

Es la forma de dar respuesta nuevas necesidades y modificaciones al SW actual, suponiendo un proceso de **mejora del SW**. Está motivado por una modificación los requisitos y limitaciones iniciales, o debido a errores operaciones o funcionales, y otras veces por necesaria adecuación del software a normativa legal.

### MÉTRICA V3

Se debe distinguir:

- Interfaz de Gestión de Proyectos, dentro de su actividad *Seguimiento y Control* (GPS), se encuentra la Gestión de Cambios de Requisitos ante variaciones de los requerimientos a posteriori del proceso de Análisis (ASI). Con varias actividades, entre las que se destacan:
  - **GPS 5 - Petición de Cambio de requisito**
  - **GPS 6 - Análisis de la Petición de Cambio de Requisitos**
  - **GPS 9 - Registro del Cambio de Requisito**
- Proceso de Mantenimiento de SS. II, con la actividad **MS1 - Registro de Petición**, en la que se establece un sistema estandarizado de registro de información para las peticiones de mantenimiento

### ITIL 4

La práctica de control de cambios está orientada a conseguir el máximo número de cambios exitosos en servicios, asegurando la gestión adecuada de los riesgos de cada cambio, mediante la existencia de una autoridad de cambio que los analice y valide, y gestionando la planificación de los cambios mediante los recursos necesarios e informando a las partes implicadas.

Se distinguen tres tipos de cambios con impacto en los servicios (adición, modificación o eliminación):

- **Estándar:** son de bajo riesgo, cambios pre-autorizados, documentados y ejecutados sin necesidad de autorización. Pueden ser operacionales o peticiones
- **Normales:** deben ser planificados, analizados y autorizados por el CAB (Comité Asesor del Cambio). Son iniciados a través de peticiones de cambio.
- **Emergencia:** deben ser implementados lo antes posible, se priorizan en la planificación respecto a los dos anteriores, se autorizan de manera más ágil por parte de una autoridad diferente ECAB y deben sufrir un análisis de riesgos.

## 2. GESTIÓN DE LA CONFIGURACIÓN Y DE VERSIONES

### 2.1. CONFIGURACIÓN

La **gestión de la configuración** es el conjunto de procesos destinados a asegurar la calidad de todo producto obtenido durante cualquiera de las etapas del desarrollo de un sistema de información (SI), a través del estricto control de los cambios realizados sobre los mismos y de la disponibilidad constante de una versión estable de cada elemento para toda persona involucrada en el citado desarrollo. Permite verificar que todos los componentes necesarios para ejecutar la aplicación estén correctamente identificados, almacenados y versionados. Se basa en el control de cambios y en el control de versiones.

### MÉTRICA V3

La Interfaz de la gestión de la configuración consiste en la aplicación de procedimientos administrativos y técnicos (identificar, definir, proporcionar información y controlar los cambios en la configuración del sistema, así como las modificaciones y versiones de estos) durante el desarrollo del sistema de información y su posterior mantenimiento. Entre los elementos de configuración del SW se incluyen ejecutables, código fuente, modelos de datos, modelos de procesos, especificaciones de requisitos y especificaciones de pruebas.

Las actividades implicadas son:

- *Estudio de Viabilidad del Sistema - EVS*
  - **EVS-GC1 – Definición de los requisitos de Gestión de la Configuración**
  - **EVS-GC2 – Establecimiento del plan de Gestión de la Configuración**
- *Análisis, Diseño, Construcción e implantación - ASI, DSI, CSI, IAS*
  - **GC 1: Identificación y Registro de Productos.**
  - **GC 2: Identificación y Registro del Producto Global**
- *Mantenimiento del Sistema de Información – MSI*
  - **MSI-GC 1: Registro del Cambio en el Sistema de Gestión de la Configuración**

#### ITIL 4

La práctica de la gestión de la configuración del servicio permite garantizar que se dispone de la información necesaria y confiable de la configuración de servicios, de sus relaciones y de los elementos de configuración (**CI o Configuration Item**), asegurando que esta información esté siempre disponible cuando y donde sea necesaria.

Incluye los procesos para:

- Identificar nuevos elementos de configuración y agregarlos al CMS y a la CMDB.
- Actualizar datos de configuración cuando se implementan cambios
- Verificar que los registros de configuración sean correctos.
- Auditoría de aplicaciones e infraestructura para identificar aquellas que no están documentadas

## 2.2. VERSIONES

Según ITIL 4, una **versión** es una publicación de un servicio u otro elemento de configuración, o una colección de elementos de configuración, que está disponible para su uso. Incluye todos los cambios SW y HW validado para su instalación en el entorno de producción.

La práctica de gestión de liberación de versiones puede realizarse con un enfoque tradicional o bien un enfoque ágil, donde el despliegue se hace en pequeños incrementos. En un entorno DevOps, la gestión de versiones está integrada con la integración continua y las herramientas para la entrega continua.

## 2.3. DESPLIEGUES

Un **despliegue** incluye los componentes de infraestructura y aplicaciones que posibilitan la entrega de desarrollo o mantenimiento SW, incorporando nuevas funcionalidades o correcciones. La **release** posibilita la activación de las nuevas funcionalidades para los usuarios finales.

Tendencias: tradicional con *plan de despliegue* o entrega continua mediante *infraestructura como código*.

Existen distintos tipos de enfoques de despliegues:

- Big Bang: la administración e implementación se ejecuta como un solo proceso una única vez.
- Por fases: se fragmenta: por módulos o funcionalidades implementadas y probadas.
- Entrega continua: en cualquier momento de manera automatizada.
  - Pull: un proceso detecta nueva versión y lo despliega de manera automatizada
  - Push: el servidor de integración continua realiza el despliegue.
- Blue/green deployment: primero se despliega (blue), después se libera para el usuario (green)
- Canary release: se despliega y se libera para un reducido nº usuarios, que se va incrementando.

### 3. GESTIÓN DE ENTORNOS

- Definir y normalizar los distintos entornos necesarios
- Establecer el procedimiento de despliegue y configuración en cada uno de los entornos
- Mantener los entornos con el SW y HW necesario para su correcta ejecución
- Promocionar y desplegar versiones en cada entorno

#### 3.1. TIPOS DE ENTORNOS

Los entornos pueden ser virtuales o reales, y según su finalidad de uso son clasificados:

- **Entorno local**
- **Entorno de desarrollo**
- **Entorno integración**
- **Entorno de pruebas o cualificación**
- **Entorno preproducción o staging**
- **Entorno de producción**

### 4. HERRAMIENTAS

#### GESTIÓN DE VERSIONES

- Control de versiones: permite guardar registro histórico de las modificaciones realizadas, gestionar líneas de base, recuperar versiones específicas y almacenar el código fuente
  - Modelos:
    - Cliente-Servidor: Apache Subversion o SVN (abierto), CVS, Visual SourceSafe (propietario), IBM Rational Rose (propietario), Perforce Helix(propietario).
    - Distribuido: Git, GitLab, Github, BitBucket, Fossil, Mercurial, Monotone, Gerrit
- Repositorio de artefactos (archivos binarios, librerías, paquetes compilados) -> Nexus, JFrog Artifactory, Archiva, npm, Docker Register

#### GESTIÓN DE CONFIGURACIÓN

- De servicios: BCM Remedy, EasyVista, ServiceNow

#### GESTIÓN DE DESPLIEGUES

- Sistemas virtuales: VirtualBox, Vagrant, Docker
- Sistemas en la nube (PaaS): OpenShift
- De infraestructura: Ansible, Chef, Puppet, SaltStack, Terraform, CloudFormation