

GESTIÓN DE LA CONFIGURACIÓN DE SISTEMAS DE SOFTWARE

AUTORA: HELLEN CUBERO LEDEZMA

NOVIEMBRE: 2020



San Marcos

Introducción

La administración de la configuración es la gestión de un sistema de software en evolución. Cuando se mantiene un sistema se establece un equipo de gestión de la configuración para garantizar que los cambios se incorporen en el sistema de manera controlada y que se mantienen registros con los detalles de los cambios que se implementaron.

Los principales procesos de administración de la configuración se ocupan de la administración del cambio, gestión de versiones, construcción del sistema y gestión de entregas de software (release). (Sommerville, 2011)



Contenido

Introducción y definiciones.....	3
Trazabilidad.....	3
Estándar IEEE 828-1998	4
Identificación de versiones	4
Árboles de características	5
Control de versiones	7
Gestión de entregas.....	7
Conclusiones y recomendaciones.....	9
Referencias bibliográficas	10

Introducción y definiciones

La administración de la configuración (CM, por las siglas de Configuration management) se ocupa de las políticas, los procesos y las herramientas para administrar los sistemas cambiantes de software. Es necesario gestionar los sistemas en evolución porque es fácil perder la pista de cuáles cambios y versiones del componente se incorporaron en cada versión del sistema. (Sommerville, 2011).

La administración de la configuración de un producto de sistema de software comprende cuatro actividades estrechamente relacionadas:

1. **Administración del cambio:** implica hacer un seguimiento de las peticiones de cambios al software por parte de los clientes y desarrolladores, estimar los costos y el efecto de realizar cambios, y decidir si deben implementarse los cambios y cuándo.
2. **Gestión de versiones:** incluye hacer un seguimiento de las numerosas versiones de los componentes del sistema y garantizar que los cambios hechos por diferentes desarrolladores a los componentes, no interfieran entre sí.
3. **Construcción del sistema:** es el proceso de ensamblar los componentes del programa, datos y librerías, y luego compilarlos y vincularlos para crear un sistema ejecutable.
4. **Gestión de entregas (release):** implica preparar el software para la entrega externa y hacer un seguimiento de las versiones del sistema que se entregaron para uso del cliente.

Trazabilidad

La trazabilidad del software según Sanchez (s.f), hace referencia a la creación y gestión de trazas (relación existe entre dos ítems de configuración, por ejemplo un requisito y las clases que lo implementan) entre artefactos de software sujetos a evolución, por ejemplo, ligar requisitos con artefactos de diseño.

Utilidades de la trazabilidad

- Análisis de impacto

- Cobertura de requisitos
- Detección de interacciones indeseadas
- Regeneración de código e ingeniería inversa.

Observe la siguiente matriz de trazabilidad:

▼ Classes Requirements ►	Req 01	Req 02	Req 03	Req 04	Req 05	Req 06
Customer			x	x	x	
Book	x		x		x	
Catalogue				x		x
SystemMng		x				x
CatalogueViewer	x	x			x	
Author				x	x	
PackedBook	x			x		

*Ilustración 1: Matriz de trazabilidad
Fuente: Sánchez, s.f*

Estándar IEEE 828-1998

El estándar IEEE 828-1998 es un estándar que establece un formato requerido y un conjunto de contenidos mínimos para los planes de gestión de configuración de software. Identifica los requisitos para la identificación de la configuración, el control de la configuración, la contabilidad y los informes del estado de la configuración. (IEEE, 1998).

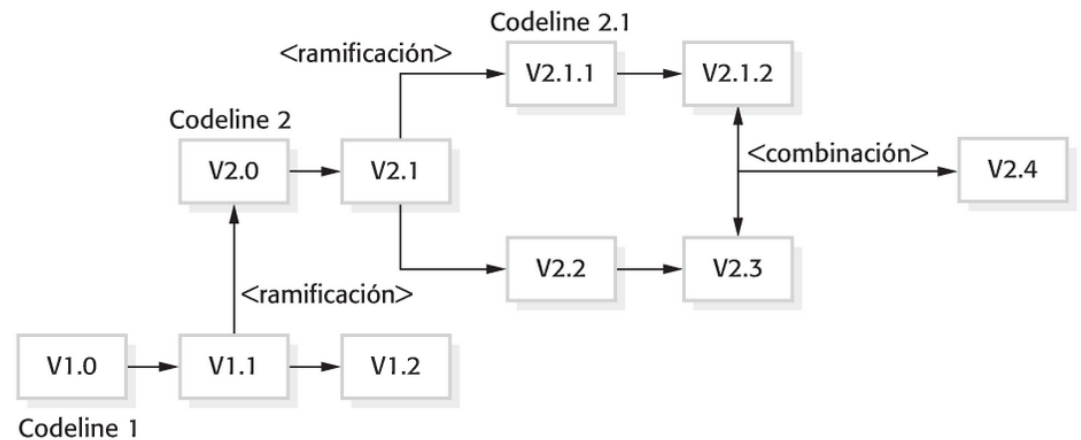
Actualmente fue reemplazado por el estándar IEEE 828-2005 y posee el mismo nombre, estándar para planes de administración de la configuración de software.

Identificación de versiones

Para evitar que dos o más componentes sean modificados al mismo tiempo y uno sobre escriba el trabajo del otro, los sistemas de gestión de versiones utilizan repositorio público y un espacio de trabajo privado. Los desarrolladores sacan componentes del repositorio público hacia su espacio de trabajo privado. Si se extrae un componente, el sistema de gestión de versiones advierte a los usuarios que deseen sacar dicho

componente que alguien más lo está usando. El sistema también garantiza que al ingresar los componentes modificados a las distintas versiones, se les asignen diferentes identificadores de versión y se almacenen por separado. Una consecuencia del desarrollo independiente del mismo componente es que las líneas de código pueden ramificarse (branch). También, si es necesario, se puede combinar ramificaciones de líneas de código para crear una nueva versión de un componente que incluya todos los cambios realizados. (Sommerville, 2011).

Observe el siguiente diagrama:



*Ilustración 2: Ramificación y combinación de líneas de código
Fuente: Sommerville, 2011*

Árboles de características

Los árboles de características según Sánchez (s.f), es una representación gráfica que muestra la especificación de atributos y posibles configuraciones. Observe la siguiente figura:



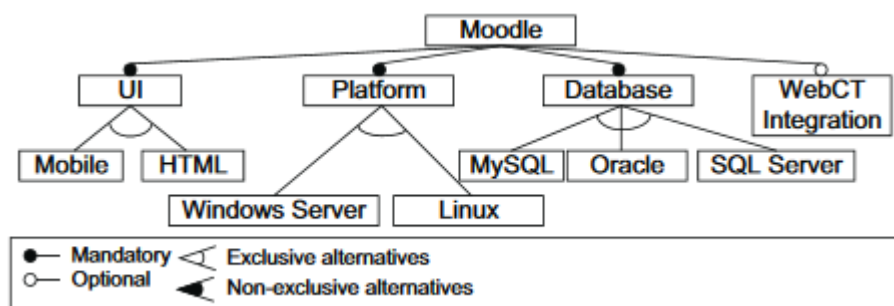


Ilustración 3: Árboles de características
Fuente: Sánchez, s.f

Especificación de restricciones

Por ejemplo: Optar por una base de datos SQLServer => Windows Server (debe implementar el sistema operativo Windows Server).

Las restricciones más usables son:

- Dependencias: $A \Rightarrow B$
- Exclusión Mutua: $(A \Rightarrow \neg B) \wedge (B \Rightarrow \neg A)$

Observe el siguiente ejemplo de configuración:

Ejemplo de configuración

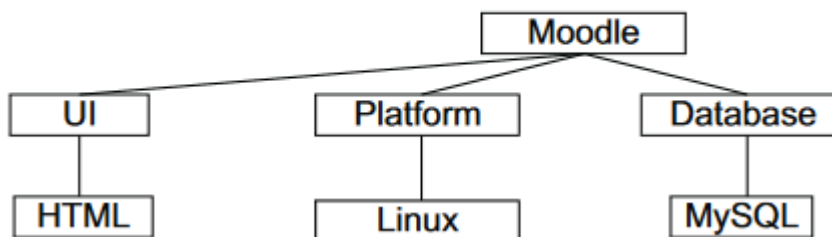


Ilustración 4: Árbol de ejemplo de configuración
Fuente: Sánchez, s.f

Para ser correcto debe satisfacer las restricciones.

Control de versiones

La gestión de versiones (VM, por las siglas de Version Management), es el proceso de hacer un seguimiento de las diferentes versiones de los componentes. También incluye asegurar que los cambios hechos a dichas versiones por los diferentes desarrolladores no interfieran unos con otros. Por lo tanto, se puede considerar a la gestión de versiones como el proceso de administrar líneas de código y líneas base.

Una línea de código es una secuencia de versiones de código fuente con las versiones más recientes en la secuencia derivadas de las versiones anteriores. Una línea base es una definición de un sistema específico, especifica la versión del componente que se incluye en el sistema más una especificación de librerías usadas, archivos de configuración, entre otros. (Sommerville, 2011)

Actualmente existen varios sistemas que permiten a las empresas establecer un control de versiones, por ejemplo Git, es un software de control de versiones diseñado por Linus Torvalds, con el objetivo de brindar eficiencia y confiabilidad al mantenimiento de versiones de aplicaciones.

Gestión de entregas

Una entrega (release) de sistema es una versión de un sistema de software que se distribuye a los clientes. Para software de mercado masivo es posible identificar por lo general dos tipos de entregas: release mayor, que proporciona funcionalidad significativamente nueva, y release menor, que repara bugs y corrige problemas reportados por el cliente.

Para software a la medida o líneas de producto de software, la gestión de las entregas del sistema es un proceso complejo. Es posible que deban producirse entregas especiales para cada cliente.

Para documentar una entrega, es necesario registrar las versiones específicas de los componentes de código fuente que se usaron en la creación del código ejecutable. Hay que conservar copias de los

archivos de código fuente, los ejecutables correspondientes y todos los datos y archivos de configuración. También hay que registrar las versiones del sistema operativo, librerías, compiladores y otras herramientas utilizadas para construir el software. (Sommerville, 2011)

Una entrega de sistema no solo es el código ejecutable del sistema, también puede incluir:

- Archivos de configuración que definan cómo debe configurarse la entrega (release) para instalaciones particulares;
- Archivos de datos, como los archivos de mensajes de error, necesarios para la operación exitosa del sistema;
- Un programa de instalación para ayudar a instalar el sistema en el hardware objetivo;
- Documentación electrónica y escrita que describa al sistema;
- Empaquetado y publicidad asociada diseñados para dicha entrega. (Sommerville, 2011).

**ESTA LECTURA ESTÁ BASADA
CASI EN SU TOTALIDAD EN EL
CAPÍTULO 25: ADMINISTRACIÓN
DE LA CONFIGURACIÓN, DEL
LIBRO SOMMERVILLE, 2011,
DISPONIBLE EN LA PLATAFORMA
ELIBRO DE LA UNIVERSIDAD.**

Conclusiones y recomendaciones

La administración de la configuración implica enfrentar un gran volumen de información, por lo que es necesario utilizar una herramienta de administración de la configuración, existen desde las más simples herramientas que apoyan una sola tarea de administración, hasta complejos y costosos conjuntos de herramientas integradas para apoyar todas las actividades de la administración de la configuración.

Por otra parte, el software debe reconstruirse frecuentemente y probarse de inmediato después de construir una nueva versión. Esto facilita la detección de bugs y problemas introducidos desde la última construcción.

Además, las entregas de sistema contienen código ejecutable, archivos de datos, archivos de configuración y documentación. La gestión de entregas incluye tomar decisiones referentes a las fechas de entrega del sistema, preparar toda la información para su distribución y documentar cada entrega del sistema. (Sommerville, 2011).

A modo de recomendación, para obtener más información observe el capítulo 25 del libro Ingeniería de Software (9a. ed.), disponible en este [enlace](#).

Referencias bibliográficas

Sommerville, I. (2011). Ingeniería de software (9a. ed.). Pearson Educación.
<https://elibro.net/es/ereader/usanmarcos/37857?page=702>

"IEEE Standard for Software Configuration Management Plans," in *IEEE Std 828-1998*, vol., no., pp.1-24, 27 Oct. 1998, doi: 10.1109/IEEESTD.1998.88281.

Sánchez, P. (s.f). Gestión de la configuración de sistemas de software. Recuperado de <https://ocw.unican.es/pluginfile.php/1408/course/section/1803/tema9-gestionConfiguracion.pdf>



www.usanmarcos.ac.cr

San José, Costa Rica