**Gestión de la configuración**

(Redirigido desde «[Gestión de configuración](https://es.wikipedia.org/w/index.php?title=Gesti%C3%B3n_de_configuraci%C3%B3n&redirect=no)»)

La **gestión de la configuración** (y de los activos) 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](https://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n) (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.

Estos dos elementos, el control de cambios y [control de versiones](https://es.wikipedia.org/wiki/Control_de_versiones) de todos los elementos del SI, facilitan también el mantenimiento de los sistemas al proporcionar una imagen detallada del sistema en cada etapa del desarrollo. La gestión de la configuración se realiza durante todas las fases del desarrollo de un sistema de información, incluyendo el mantenimiento y control de cambios, una vez realizada la puesta en producción.

**Elementos de configuración software**

Según la interfaz **gestión de la configuración** definida en [MÉTRICA](https://es.wikipedia.org/wiki/M%C3%89TRICA) v3, los elementos de configuración del [software](https://es.wikipedia.org/wiki/Software) incluyen:

* Ejecutables
* Código Fuente
* Modelos de datos
* Modelos de procesos
* Especificaciones de requisitos
* Pruebas

Y para cada uno de estos elementos se almacenará al menos:

* Nombre
* Versión
* Estado
* Localización

**Gestión de configuración de software**

**Gestión de Configuración de Software** (*Software Configuration Management*, **SCM**) es una especialización de la [gestión de configuración](https://es.wikipedia.org/wiki/Gesti%C3%B3n_de_configuraci%C3%B3n) a todas las actividades en el sector del desarrollo de [software](https://es.wikipedia.org/wiki/Software).

SCM trata y controla:

* la elaboración de [código fuente](https://es.wikipedia.org/wiki/C%C3%B3digo_fuente) por varios [desarrolladores](https://es.wikipedia.org/wiki/Desarrollador_de_software) simultáneamente;
* el seguimiento del estado de las [fases del desarrollo de software](https://es.wikipedia.org/wiki/Fases_del_desarrollo_de_software) (versiones) y sus cambios ([control de versiones](https://es.wikipedia.org/wiki/Control_de_versiones)) y
* la conducción de la integración de las partes del software en un solo producto de software.

Para la realización de la SCM hay diferentes herramientas. Pero herramientas que pretenden ofrecer una solución total al problema, a menudo no cumplen con los requisitos técnicos como:

* apoyo a diferentes plataformas.
* iniciar el proceso de *build*.
* conexión a los bancos de datos existentes.
* integración a la organización existente.

Por esa razón ofrece una mayor flexibilidad una solución que integre herramientas parciales que sean más fáciles de integrar en el proceso existente.

Por ejemplo:

* Uso de un software de administración de versiones como IBM Rational Team Concert, [CVS](https://es.wikipedia.org/wiki/CVS), [Subversion](https://es.wikipedia.org/wiki/Subversion), [SourceSafe](https://es.wikipedia.org/wiki/SourceSafe), [ClearCase](https://en.wikipedia.org/wiki/ClearCase), [Darcs](https://es.wikipedia.org/wiki/Darcs), [Plastic SCM](https://es.wikipedia.org/wiki/Plastic_SCM).
* Introducción de una herramienta para la [documentación](https://es.wikipedia.org/wiki/Documentaci%C3%B3n) comunitaria con una [administración de cambios](https://es.wikipedia.org/wiki/Control_de_versiones), acceso interactivo y foro o alguna plataforma para la comunicación.
* Determinar un entorno para el *build* automático.

**Sistemas de SCM**

Actualmente los siguientes sistemas se ofrecen en el mercado:

* [AccuRev](https://es.wikipedia.org/wiki/AccuRev)
* [Perforce](https://es.wikipedia.org/wiki/Perforce)
* [ClearCase](https://en.wikipedia.org/wiki/ClearCase)
* [Plastic SCM](https://es.wikipedia.org/wiki/Plastic_SCM)
* SpectrumSCM
* Surround SCM
* Sablime
* Smart Bear
* SET-LIBER[1](https://es.wikipedia.org/wiki/Gesti%C3%B3n_de_configuraci%C3%B3n_de_software#cite_note-1)
* [Rational Synergy](https://en.wikipedia.org/wiki/Rational_Synergy) (Telelogic Synergy/CM: Telelogic Synergy (ehem. Synergy/CM, ehem. CM/Synergy, ehem. CCM)
* [Subversion](https://es.wikipedia.org/wiki/Subversion)
* [Git](https://es.wikipedia.org/wiki/Git)
* [Trac](https://es.wikipedia.org/wiki/Trac)
* [Visual Source Safe](https://es.wikipedia.org/wiki/Visual_Source_Safe) (Microsoft)
* Microsoft Team Foundation Server 2010[2](https://es.wikipedia.org/wiki/Gesti%C3%B3n_de_configuraci%C3%B3n_de_software#cite_note-2)
* Microsoft Visual Studio 2010 ALM[3](https://es.wikipedia.org/wiki/Gesti%C3%B3n_de_configuraci%C3%B3n_de_software#cite_note-3)
* Harvest (CA)

**Software configuration management**

From Wikipedia, the free encyclopedia

|  |  |
| --- | --- |
| https://upload.wikimedia.org/wikipedia/commons/thumb/a/a4/Text_document_with_red_question_mark.svg/40px-Text_document_with_red_question_mark.svg.png | This article includes a [list of references](https://en.wikipedia.org/wiki/Wikipedia:Citing_sources), but **its sources remain unclear** because it has **insufficient** [**inline citations**](https://en.wikipedia.org/wiki/Wikipedia:Citing_sources#Inline_citations). Please help to [improve](https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Fact_and_Reference_Check) this article by [introducing](https://en.wikipedia.org/wiki/Wikipedia:When_to_cite) more precise citations. *(September 2010)* *(*[*Learn how and when to remove this template message*](https://en.wikipedia.org/wiki/Help:Maintenance_template_removal)*)* |

|  |
| --- |
| [**IEEE**](https://en.wikipedia.org/wiki/IEEE) **software life cycle** |
| * SQA – [Software quality assurance](https://en.wikipedia.org/wiki/Software_quality_assurance) [IEEE 730](https://en.wikipedia.org/wiki/Software_quality_assurance) * SCM – **Software configuration management** **IEEE 828** * STD – [Software test documentation](https://en.wikipedia.org/wiki/Software_test_documentation) [IEEE 829](https://en.wikipedia.org/wiki/Software_test_documentation) * SRS – [Software requirements specification](https://en.wikipedia.org/wiki/Software_requirements_specification) [IEEE 830](https://en.wikipedia.org/wiki/Software_requirements_specification) * V&V – [Software verification and validation](https://en.wikipedia.org/wiki/Software_verification_and_validation) [IEEE 1012](https://en.wikipedia.org/wiki/Software_verification_and_validation) * SDD – [Software design description](https://en.wikipedia.org/wiki/Software_design_description) [IEEE 1016](https://en.wikipedia.org/wiki/Software_design_description) * SPM – [Software project management](https://en.wikipedia.org/wiki/Software_project_management) [IEEE 1058](https://en.wikipedia.org/wiki/Software_project_management) * SUD – [Software user documentation](https://en.wikipedia.org/wiki/Software_user_documentation) [IEEE 1063](https://en.wikipedia.org/wiki/IEEE_1063) |
| * [v](https://en.wikipedia.org/wiki/Template:IEEE_software_documents) * [t](https://en.wikipedia.org/wiki/Template_talk:IEEE_software_documents) * [e](https://en.wikipedia.org/w/index.php?title=Template:IEEE_software_documents&action=edit) |

In [software engineering](https://en.wikipedia.org/wiki/Software_engineering), **software configuration management** (**SCM** or **S/W CM**)[[1]](https://en.wikipedia.org/wiki/Software_configuration_management#cite_note-1) is the task of tracking and controlling changes in the software, part of the larger cross-disciplinary field of [configuration management](https://en.wikipedia.org/wiki/Configuration_management).[[2]](https://en.wikipedia.org/wiki/Software_configuration_management#cite_note-2) SCM practices include [revision control](https://en.wikipedia.org/wiki/Revision_control) and the establishment of [baselines](https://en.wikipedia.org/wiki/Baseline_%28configuration_management%29). If something goes wrong, SCM can determine what was changed and who changed it. If a configuration is working well, SCM can determine how to replicate it across many hosts.

The acronym "SCM" is also expanded as **source configuration management process** and **software change and configuration management**.[[3]](https://en.wikipedia.org/wiki/Software_configuration_management#cite_note-3) However, "configuration" is generally understood to cover changes typically made by a [system administrator](https://en.wikipedia.org/wiki/System_administrator).

**Contents**

* [1 Purposes](https://en.wikipedia.org/wiki/Software_configuration_management#Purposes)
* [2 History](https://en.wikipedia.org/wiki/Software_configuration_management#History)
* [3 See also](https://en.wikipedia.org/wiki/Software_configuration_management#See_also)
* [4 References](https://en.wikipedia.org/wiki/Software_configuration_management#References)
* [5 Further reading](https://en.wikipedia.org/wiki/Software_configuration_management#Further_reading)
* [6 External links](https://en.wikipedia.org/wiki/Software_configuration_management#External_links)

**Purposes**

|  |  |
| --- | --- |
| https://upload.wikimedia.org/wikipedia/en/thumb/f/f2/Edit-clear.svg/40px-Edit-clear.svg.png | This section **is in a list format that may be better presented using** [**prose**](https://en.wikipedia.org/wiki/Wikipedia:Prose)**.** You can help by converting this section to prose, if [appropriate](https://en.wikipedia.org/wiki/Wikipedia:Embedded_list). [Editing help](https://en.wikipedia.org/wiki/Help:Editing) is available. *(June 2014)* |

The goals of SCM are generally:[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

* Configuration identification - Identifying configurations, [configuration items](https://en.wikipedia.org/wiki/Configuration_item) and [baselines](https://en.wikipedia.org/wiki/Baseline_%28configuration_management%29).
* Configuration control - Implementing a [controlled change](https://en.wikipedia.org/wiki/Change_control) process. This is usually achieved by setting up a [change control board](https://en.wikipedia.org/wiki/Change_control_board) whose primary function is to approve or reject all change requests that are sent against any baseline.
* Configuration status accounting - Recording and reporting all the necessary information on the status of the development process.
* Configuration auditing - Ensuring that configurations contain all their intended parts and are sound with respect to their specifying documents, including requirements, architectural specifications and user manuals.
* [Build management](https://en.wikipedia.org/wiki/Build_management) - Managing the process and tools used for builds.
* [Process management](https://en.wikipedia.org/wiki/Process_management) - Ensuring adherence to the organization's development process.
* Environment management - Managing the software and hardware that host the system.
* [Teamwork](https://en.wikipedia.org/wiki/Teamwork) - Facilitate team interactions related to the process.
* Defect tracking - Making sure every defect has traceability back to the source.

With the introduction of [cloud computing](https://en.wikipedia.org/wiki/Cloud_computing) the purposes of SCM tools have become merged in some cases. The SCM tools themselves have become virtual appliances that can be instantiated as virtual machines and saved with state and version. The tools can model and manage cloud-based virtual resources, including virtual appliances, storage units, and software bundles. The roles and responsibilities of the actors have become merged as well with developers now being able to dynamically instantiate virtual servers and related resources.[[4]](https://en.wikipedia.org/wiki/Software_configuration_management#cite_note-4)

**History**

The history of software configuration management (SCM) in computing can be traced back as early as the 1950s, when CM (for Configuration Management), originally for hardware development and production control, was being applied to software development. Early software had a physical footprint, such as [cards](https://en.wikipedia.org/wiki/Punched_card), [tapes](https://en.wikipedia.org/wiki/Punched_tape), and other media. The first software configuration management was a manual operation. With the advances in language and complexity, [software engineering](https://en.wikipedia.org/wiki/Software_engineering), involving configuration management and other methods, became a major concern due to issues like schedule, budget, and quality. Practical lessons, over the years, had led to the definition, and establishment, of procedures and tools. Eventually, the tools became systems to manage software changes.[[5]](https://en.wikipedia.org/wiki/Software_configuration_management#cite_note-5) Industry-wide practices were offered as solutions, either in an open or proprietary manner (such as [Revision Control System](https://en.wikipedia.org/wiki/Revision_Control_System)). With the growing use of computers, systems emerged that handled a broader scope, including [requirements management](https://en.wikipedia.org/wiki/Requirements_management), design alternatives, quality control, and more; later tools followed the guidelines of organizations, such as the Capability Maturity Model of the [Software Engineering Institute](https://en.wikipedia.org/wiki/Software_Engineering_Institute).

# Serena Dimensions CM

Serena Dimensions CM elimina la complejidad del desarrollo en paralelo permitiendo trabajos fluídos, automatizar y ajustar sus tareas, visualizar y minimizar los riesgos ligados a los tiempos considerados/estipulados y eliminar los riesgos que  versión.

Es una solución líder en el mercado que se encuentra enmarcada en la gestión del ciclo de vida de la aplicación (Application Lifecycle Management o ALM), específicamente en cambios en software y gestión de configuración (Software change and configuration management o SCCM).

Serena Dimensiones CM orquesta perfectamente cambios en las aplicaciones, configuraciones y versiones en diferentes plataformas, entornos y equipos. Proporciona una amplia gestión de la configuración de software y capacidades de implementación que ayudan a aumentar la eficiencia, reducir los riesgos de implementación, y asegurar el cumplimiento de las normas.

Dimensions CM ofrece integración de los IDEs más importantes del mercado como Eclipse y Microsoft Visual Studio, visualización de los flujos y en el histórico de cambios, vistas prácticas para validar con completa facilidad los cambios en las diferentes versiones, estrecha relación con las metodologías de desarrollo, flexibilidad para colocar comentarios y compartir vistas de avances y cambios.

Estas características permiten ahorrar tiempo en el desarrollo, mejorar la productividad, disminuir los costos asociados al desarrollo de nuevas funcionalidades y minimizar los riesgos de implantación. Permite automatizar y orquestar sus mejores prácticas de gestión de configuración, el mantenimiento de un control total durante todo el ciclo de vida de toda la aplicación.

**Organización del Proyecto Configurada a su manera**

Con Serena Dimensions CM, usted puede organizar sus proyectos basados en una estructura tradicional, física, o se pueden organizar lógicamente con "piezas de diseño”. Su organización lógica es ideal para el desarrollo basado en componentes, ya que facilita el intercambio y la reutilización de código mediante el intercambio de una parte del diseño en los proyectos.

**Líder del mercado en Seguridad basada en Roles**

Separación de funciones y responsabilidades de seguridad. Serena Dimensions CM facilita esta tarea a través de gran granularidad y autorización basada en privilegio.  Incluye un modelo de seguridad altamente configurable, que le da la máxima flexibilidad para satisfacer las necesidades del negocio sin comprometer la seguridad.

**Más que solo SCCM**

Serena Dimensions va más allá de las capacidades de una solución típica de SCCM. El producto también ofrece la construcción y gestión de la implementación. Este último se usa a menudo para crear una "bóveda de oro" donde almacena los artefactos de software sin comprometer la seguridad.