

# Planificación y Gestión de Proyectos Informáticos

Gestión de la configuración del software

Jose M. Moyano

`jmoiano@ugr.es`

**Departamento de Ciencias de la Computación e Inteligencia Artificial**

**Universidad de Granada**

Curso 2022-23



**DaSCI**

Instituto Andaluz de Investigación en  
Data Science and Computational Intelligence



**UNIVERSIDAD  
DE GRANADA**

# Índice

Gestión de la configuración del software

Terminología

Control de cambios

Control de versiones

# Gestión de la configuración del software

## Software Configuration Management (SCM)

SCM no es solo disponer de una herramienta que registre quién hizo qué cambio al código o a la documentación y cuando lo hizo.

SCM también incluye la creación de convenciones sobre el código, políticas y procedimientos que aseguren que todos los “stakeholders” relevantes están involucrados en la realización de cambios sobre el software:

**Conjunto de actividades realizadas para gestionar los cambios a lo largo del ciclo de vida del software.**

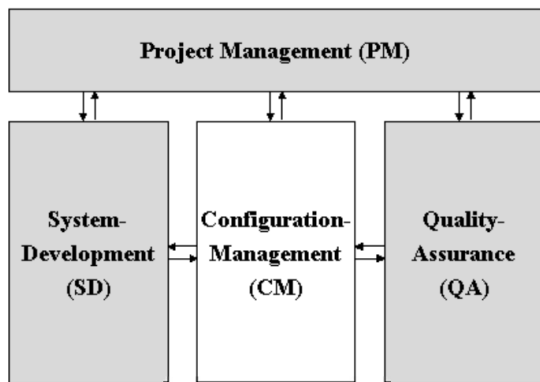
# Gestión de la configuración del software

- ▶ Cómo informar de un problema con el software.
- ▶ Cómo solicitar la incorporación de un nuevo requisito.
- ▶ Cómo informar de los cambios solicitados y pedir sus opiniones a los “stakeholders” involucrados.
- ▶ Cómo priorizar y planificar las solicitudes de cambios.
- ▶ Cómo mantener bajo control las distintas versiones de los productos, es decir, impedir que puedan cambiarse sin seguir los procedimientos apropiados.

# Gestión de la configuración del software

## Propósito

Mantener la integridad de los productos de un proyecto de desarrollo de software a lo largo de su ciclo de vida.



# Gestión de la configuración del software

Aborda 4 tipos de problemas habituales:

- ▶ Comunicación: el número de vías de comunicación aumenta cuadráticamente con el tamaño del equipo.
- ▶ Cambios: problemas cuando un programador no es consciente de los últimos cambios realizados sobre un ítem compartido.
- ▶ Actualizaciones: problemas asociados a la actualización de ítem compartidos.
- ▶ Mantenimiento: problemas asociados al mantenimiento de múltiples versiones.

# Gestión de la configuración del software

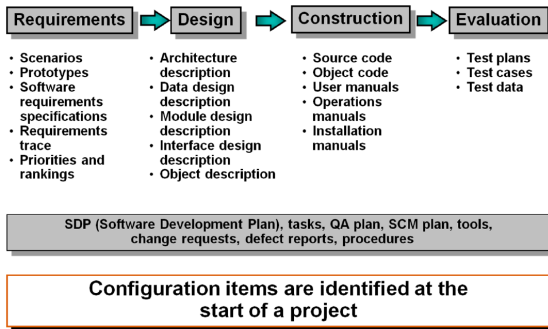
## Beneficios

- ▶ Se evitan problemas asociados a la realización de cambios y se facilitan las tareas de mantenimiento.
- ▶ Aumento de productividad.
- ▶ Automatización de tareas, p.ej. Seguimiento de defectos (defect logging & tracking).

# Terminología

## SCI (Software Configuration Item):

Unidad fundamental en un sistema de gestión de la configuración, p.ej. ficheros de código, planes, modelos especificaciones, documentos, procesos...

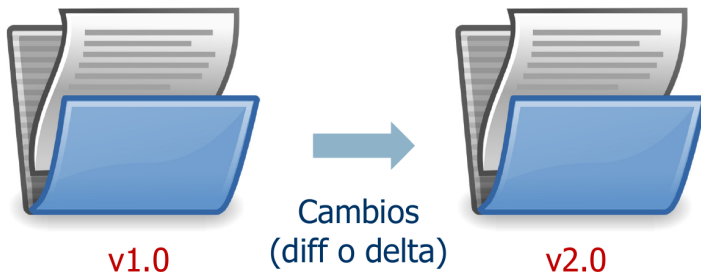




# Terminología

## Versión

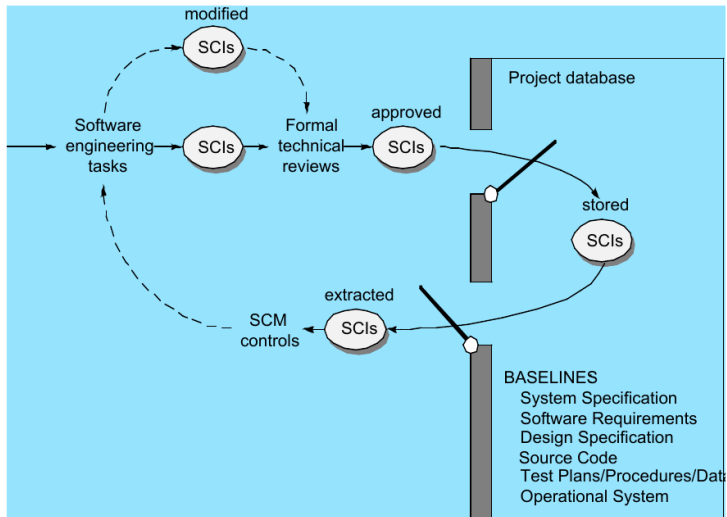
Versión específica de un SCI:



# Terminología

## Baseline

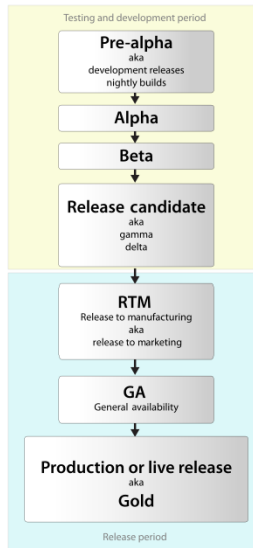
Versión aprobada de un producto en un momento determinado que sirve de base para definir los cambios.



# Terminología

## Release

Distribución formal de una versión aprobada. Ej.: versión alfa/beta.



## Branch (rama)

Duplicación de un conjunto de SCI's de forma que las modificaciones puedan realizarse en paralelo.

- ▶ “trunk”: línea de desarrollo que no es una rama, a.k.a. baseline, mainline o master.
- ▶ Conjunto de ramas de un proyecto: árboles de versiones (trees), streams o “codelines”.

# Control de cambios

“Software is never done...”

- ▶ CR (Change Requests).
- ▶ Change “triage”.
- ▶ CCB (Change/configuration control board). Informal en proyecto pequeños, 7 niveles en FAA NAS (National Airspace System).

# Control de cambios

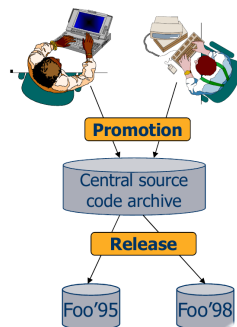


Tipos de cambio:

- ▶ Promotion: interno a la organización de desarrollo.
- ▶ Release: visible fuera de la organización de desarrollo.

# Control de cambios

- ▶ Programmer's directory: bajo el control del programador.
- ▶ Master directory: contiene la línea base actual, sólo con los cambios autorizados.
- ▶ Software repository: archivo de las versiones publicadas.



# Control de versiones

El almacén (a.k.a. repositorio) SCM proporciona el conjunto de mecanismos y estructuras de datos que permiten gestionar los cambios de forma efectiva.

## Funciones

- ▶ Integridad de las distintas versiones.
- ▶ Gestión de dependencias entre elementos SCI.
- ▶ Compartir datos entre los miembros del equipo.
- ▶ Integración con las herramientas de desarrollo.
- ▶ Aplicación de mecanismos de gestión de cambios (p.ej. aprobación en revisiones técnicas formales).
- ▶ Trazabilidad.
- ▶ Realización de auditorías.



# Control de versiones

Un sistema de control de versiones implementa o se integra directamente con:

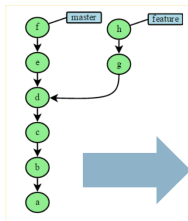
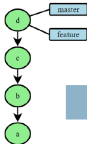
- ▶ Una base de datos (o repositorio) del proyecto.
- ▶ Una herramienta de gestión de versiones que maneja todas las versiones de cada SCI.
- ▶ Una herramienta de compilación (make/build) que permite automatizar la creación de versiones específicas.
- ▶ Una herramienta de seguimiento (issue/bug tracking) que permite registrar y controlar todos los hechos o problemas asociados a los distintos SCI.

# Control de versiones

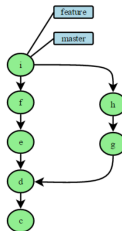
## Git

### Branching & merging

```
$ git branch feature # Create the branch
$ git checkout feature # Switch to the new branch
# or in a single command
$ git checkout -b feature
```



```
$ git merge feature
```

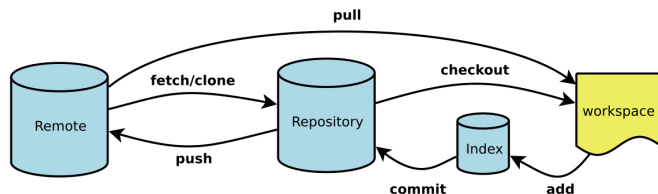


<https://illustrated-git.readthedocs.org/en/latest/>

# Control de versiones

## Git

### Remote repositories



<https://illustrated-git.readthedocs.org/en/latest/>

# Ejercicios

Busque información sobre herramientas concretas que faciliten la realización de las siguientes tareas:

- ▶ Control de versiones.
- ▶ Compilación (daily builds).
- ▶ Pruebas automatizadas.
- ▶ Seguimiento de bugs (bug tracking).

Para cada categoría, identifique al menos 3 herramientas que se utilicen actualmente.

# Bibliografía

## Libros de texto:

- ▶ Roger S. Pressman: **Software Engineering: A Practitioner's Approach** McGraw-Hill, 8th edition, 2014. ISBN 0078022126
- ▶ Shari Lawrence Pfleeger & Hoanne M. Atlee: **Software Engineering: Theory and Practice** Prentice Hall, 4th edition, 2009, ISBN 0136061699
- ▶ Ian Sommerville: **Software Engineering** Pearson, 10th edition, 2015. ISBN 0133943038

# Bibliografía

## Lecturas recomendadas

- ▶ Dwayne Phillips: **The Software Project Manager's Handbook: Principles That Work at Work** Wiley / IEEE Computer Society, 2nd edition, 2004, ISBN 0471674206
- ▶ Donald J. Reifer (editor): **Software Management** Wiley / IEEE Computer Society, 7th edition, 2006 ISBN 0471775622
- ▶ Richard H. Thayer (editor): **Software Engineering Project Management** Wiley / IEEE Computer Society, 2nd Edition, 2000, ISBN 0818680008

# Planificación y Gestión de Proyectos Informáticos

Gestión de la configuración del software

Jose M. Moyano

[jmoyano@ugr.es](mailto:jmoyano@ugr.es)

**Departamento de Ciencias de la Computación e Inteligencia Artificial**

**Universidad de Granada**

Curso 2022-23



**DaSCI**

Instituto Andaluz de Investigación en  
Data Science and Computational Intelligence



**UNIVERSIDAD  
DE GRANADA**