

Universidad Tecnológica Nacional
Facultad Regional Córdoba
Cátedra de Ingeniería de Software
Docentes: Judith Meles, Daniel Battistelli

Software Configuration Management SCM

(o más allá del Commit, Update)

Cambios en el Software

Tienen su origen en:

- Cambios del negocio y nuevos requerimientos
- Soporte de cambios de productos asociados
- Reorganización de las prioridades de la empresa por crecimiento
- Cambios en el presupuesto
- Defectos encontrados a corregir
- Oportunidades de mejora

Un poco de Historia



Tiene su origen a mediados de 1950s, cuando CM (por Configuration Management) originalmente utilizado para desarrollo de hardware y control de producción, fue utilizado en el desarrollo de software.

Definición

Una disciplina que aplica dirección y monitoreo administrativo y técnico a: identificar y documentar las características funcionales y técnicas de los ítems de configuración, controlar los cambios de esas características, registrar y reportar los cambios y su estado de implementación y verificar correspondencia con los requerimientos

(ANSI/IEEE 828, 1990)

Por qué deberíamos usarlo

Su propósito es establecer y mantener la integridad de los productos del proyecto de software a lo largo de su ciclo de vida.

Involucra para la configuración:

- Identificarla en un momento dado
- Controlar sistemáticamente sus cambios
- Mantener su integridad y origen

Problemas en el manejo de componentes

- Pérdida de un componente
- Pérdida de cambios (el componente que tengo no es el último)
- Sincronía fuente - objeto – ejecutable
- Regresión de fallas
- Doble mantenimiento
- Superposición de cambios
- Cambios no validados

Integridad del Producto

Un Producto tiene Integridad cuando:

Satisface las necesidades del usuario

Puede ser fácil y completamente rastreado durante su ciclo de vida

Satisface criterios de performance

Cumple con sus expectativas de costo

SCM como disciplina de gestión

Es una actividad “paraguas”, transversal a todo el proyecto con aplicación en las diferentes disciplinas.



Elementos de SCM

1. Identificación de Items de Configuración

Se trata de establecer estándares de documentación y un esquema de identificación de documentos

2. Control de cambios

Consiste en la evaluación y registro de todos los cambios que se hagan de la configuración software

3. Auditorías de configuraciones

Sirven, junto con las revisiones técnicas formales para garantizar que el cambio se ha implementado correctamente

4. Generación de informes

Identificación de ítems de configuración

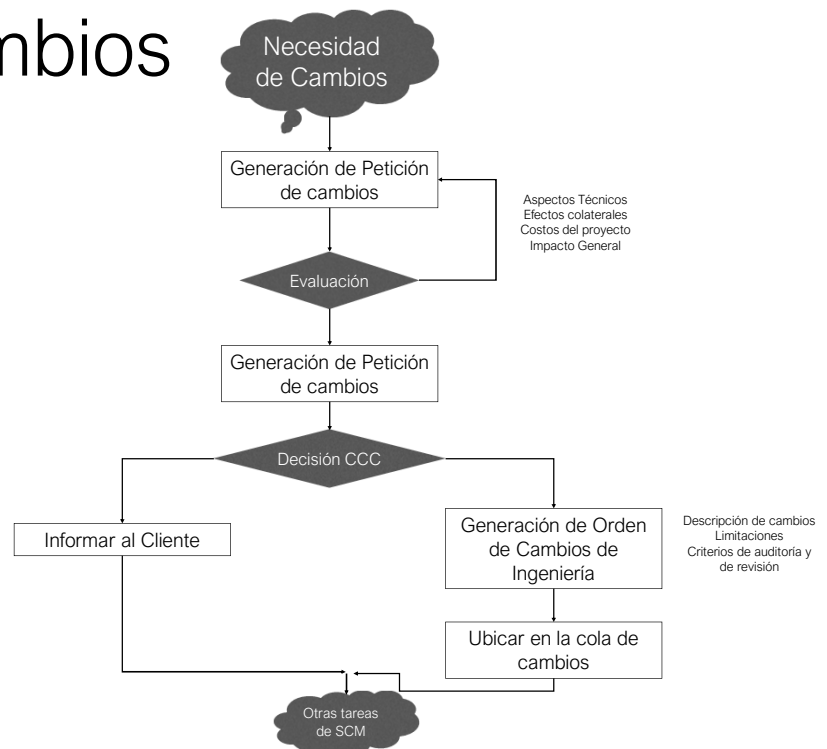
- Todos aquellos artefactos que forman parte del producto o participan de la gestión del proyecto que pueden sufrir cambios o necesitan ser compartidos entre los miembros del equipo y sobre los cuales necesitamos conocer su estado y evolución.
- Pueden ser requerimientos, documentos de diseño, código fuente, código ejecutable, etc.

Algunos ejemplos de CI

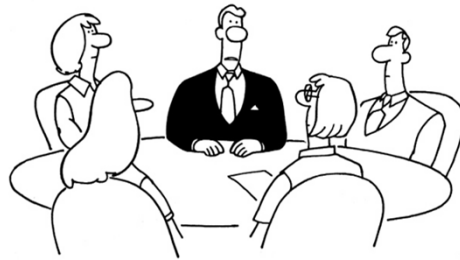
- **Plan de CM**
- Propuestas de Cambio
- Visión
- 10 Riesgos principales
- Plan de desarrollo
- Prototipo de Interface
- Manual de Usuario
- Requerimientos
- Plan de Calidad
- Arquitectura del Software
- Plan de Integración
- Planes de fases
- Estándares de codificación
- Casos de prueba
- Código fuente
- Gráficos, iconos, ...
- Instructivo de ensamble
- Programa de instalación
- Documento de despliegue
- Formulario de aceptación
- Registro del proyecto

Control de Cambios

- Tiene su origen en un Requerimiento de Cambio a uno o varios ítems de configuración que se encuentran en una **línea base**.
- Es un Procedimiento formal que involucra diferentes actores y una evaluación del **impacto** del cambio



El Comité de Control de Cambios



"Whew! That was close!
We almost decided something!"

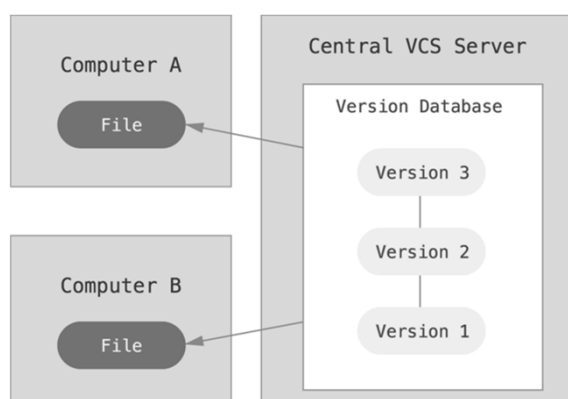
- Está formado por representantes de todas las áreas involucradas en el desarrollo:
- Análisis, Diseño
- Implementación
- Testing
- Otros interesados

El Repositorio



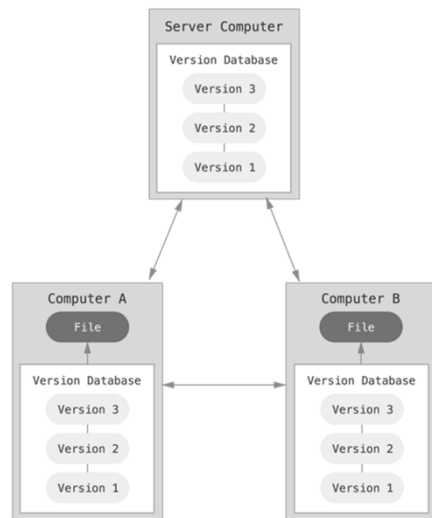
- Un depósito de información conteniendo los ítems de configuración (CIs)
- Mantiene la historia de cada CI con sus atributos y relaciones (**metainformación**)
- Usado para hacer **evaluaciones de impacto** de los cambios propuestos
- Pueden ser una o varias bases de datos
- Posee herramientas de automatización, integración y generación de informes

Repositorios Centralizados



- Un servidor contiene todos los archivos con sus versiones
- Los administradores tiene mayor control sobre el repositorio
- Falla el servidor y estamos al horno

Repositorios Descentralizados



- Cada cliente tiene una copia **exactamente igual** del repositorio completo
- Si un servidor muere solo es cuestión de “copiar y pegar”
- Posibilita otros **workflows** no disponibles en el modelo centralizado

Workflows

In case of fire



1. git commit



2. git push



3. leave building

Auditorías de Configuración

- **Auditoría física de configuración (PCA)**
Asegura que lo que está indicado para cada SCI en la línea base o en la actualización se ha alcanzado realmente.
- **Auditoría funcional de configuración (FCA)**
Evaluación independiente de los productos de software, verificando que la funcionalidad y performance reales de cada ítem de configuración sean consistentes con la **especificación de requerimientos**.

Características

- Las auditorías cuestan tiempo y dinero
- Deben realizarse desde la primeras etapas de desarrollo. Su postergación a etapas posteriores puede llegar a hacer fracasar el proyecto
- Suministra **visibilidad** y **rastreabilidad** del ciclo de vida del producto de software

Registro e Informe de Estado

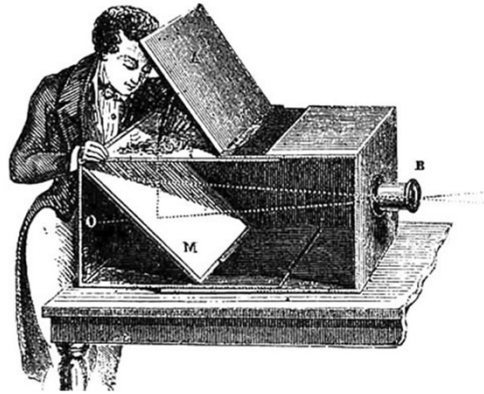
- Se ocupa de mantener los registros de la evolución del sistema
- Maneja mucha información y salidas por lo que se suele implementar dentro de **procesos automáticos**
- Incluye reportes de rastreabilidad de todos los cambios realizados a las **líneas base** durante el ciclo de vida.

Algunas preguntas que podría responder

- ¿Cuál es el estado del ítem?
- ¿Un requerimiento de cambio ha sido aprobado o rechazado por el CCB?
- ¿Qué versión de ítem implementa un requerimiento de cambio aprobado (saber cuál es el componente que contiene la mejora)?
- ¿Cuál es la diferencia entre una versión y otra dada?

La Configuración

Un conjunto de ítems de configuración con su correspondiente versión en un momento determinado



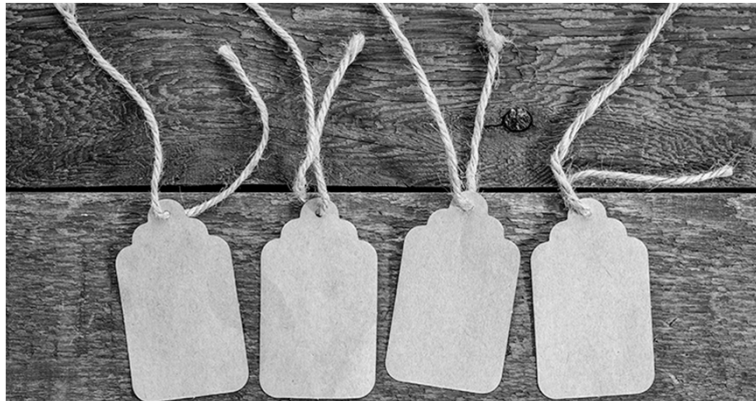
La Línea Base



- Una **configuración** que ha sido revisada formalmente y sobre la que se ha llegado a un acuerdo
- Sirve como base para desarrollos posteriores y puede cambiarse sólo a través de un **procedimiento formal de control de cambios**
- Permiten ir atrás en el tiempo y reproducir el entorno de desarrollo en un momento dado del proyecto

Identificación de la Línea Base

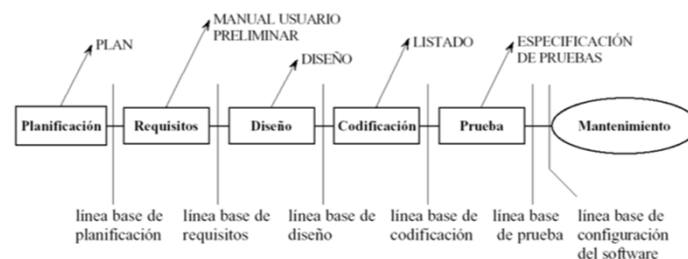
- Se utilizan **etiquetas** para “marcar” las baseline
- No confundir con la versión del Producto x.y.z (semver)



Muchas líneas base

Pueden ser:

- De especificación (Requerimientos, Diseño)
- De productos que han pasado por un control de calidad definido previamente

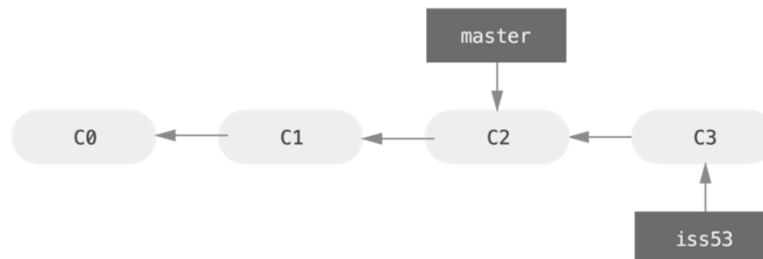




Ramas

Creación de ramas

- Existe una **rama principal** (trunk, master)
- Sirven para bifurcar el desarrollo
- Pueden tener razones de creación con semántica
- Permiten la experimentación
- Pueden ser **descartadas** o **integradas**



Integración de ramas



- La operación se llama **merge**
- Lleva los cambios a la rama principal
- Pueden surgir **conflictos** (resolvemos con diff)
- Todas las ramas deberían eventualmente integrarse a la principal o ser descartadas



Plan de Gestión de Configuración

También se planifica! Qué debería incluir el plan?

- Reglas de nombrado de los CI
- Herramientas a utilizar para SCM
- Roles e integrantes del Comité
- Procedimiento formal de cambios
- Plantillas de formularios
- Procesos de Auditoría

SCM en entornos Agile

- **Individuos e interacciones sobre procesos y herramientas**
Los procesos y herramientas de SCM deben adaptarse al equipo y su forma de trabajo y no al revés.
- **Software funcionando sobre documentación extensiva**
Es posible minimizar la dependencia del equipo de trabajo en procedimientos documentados automatizando tareas propias de la gestión de SCM con herramientas.
- **Colaboración con el cliente sobre negociación contractual**
El uso de herramientas de SCM apropiadas puede proveer mayor visibilidad a los stakeholders del estado del proyecto y mejorar la comunicación.
- **Respuesta ante el cambio sobre seguir un plan**
El uso de las estructuras y políticas de SCM adecuadas debe facilitar y propiciar el cambio en vez de prevenirlo.

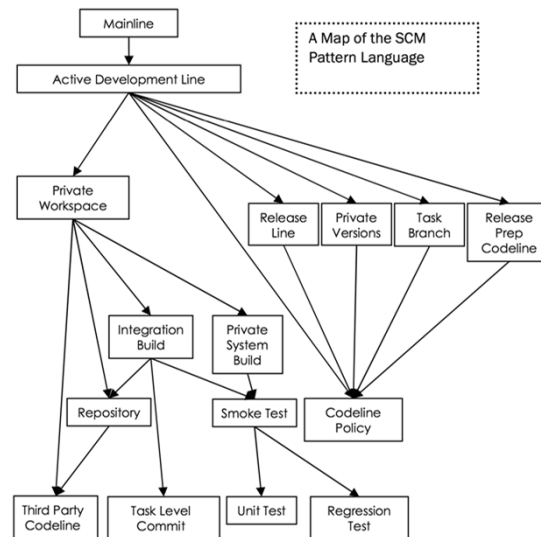
SCM en Agile

- Qué pasa con el Comité de Control de Cambios?
- Algunos de los ítems que podemos identificar, por ejemplo en el framework Scrum:
 - Fotos del Burndown Chart
 - Correos de aprobación de Sprint Review
 - User Stories
 - Backlogs
 - Conclusiones de Retrospectivas

Y hay mucho más...

- Patrones de SCM
- Antipatrones
- DevOps
- Continuous Integration (CI)
- Continuous Delivery & Deployment
- High Availability

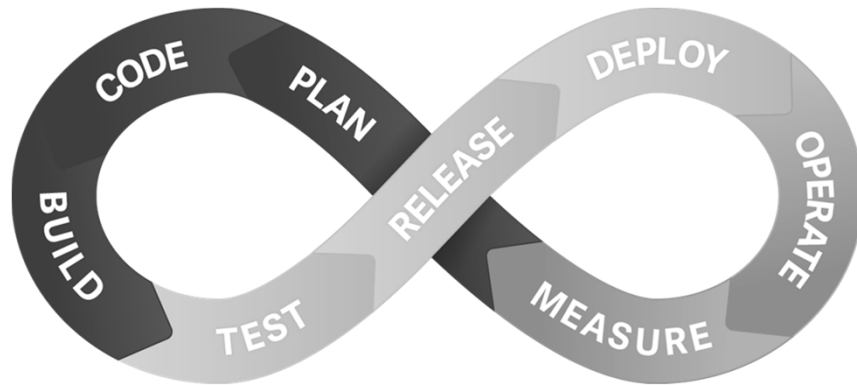
El Mapa de los Patrones



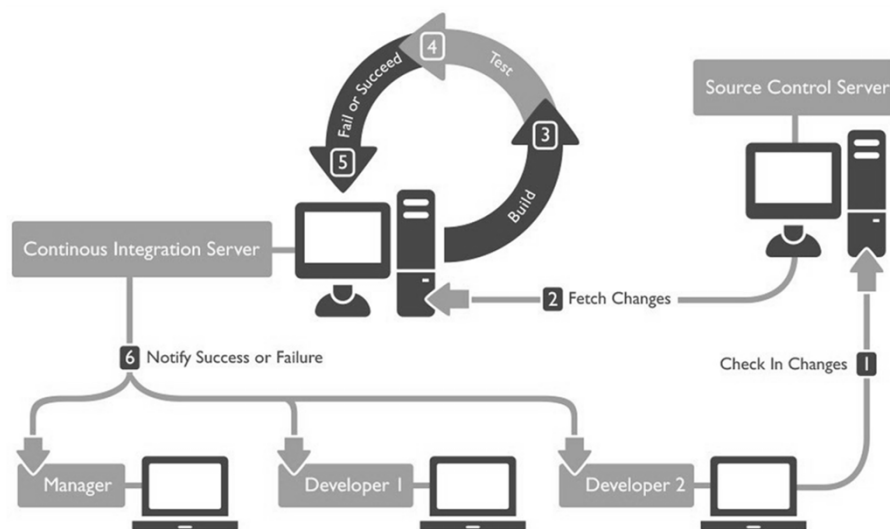
Y los Antipatrones

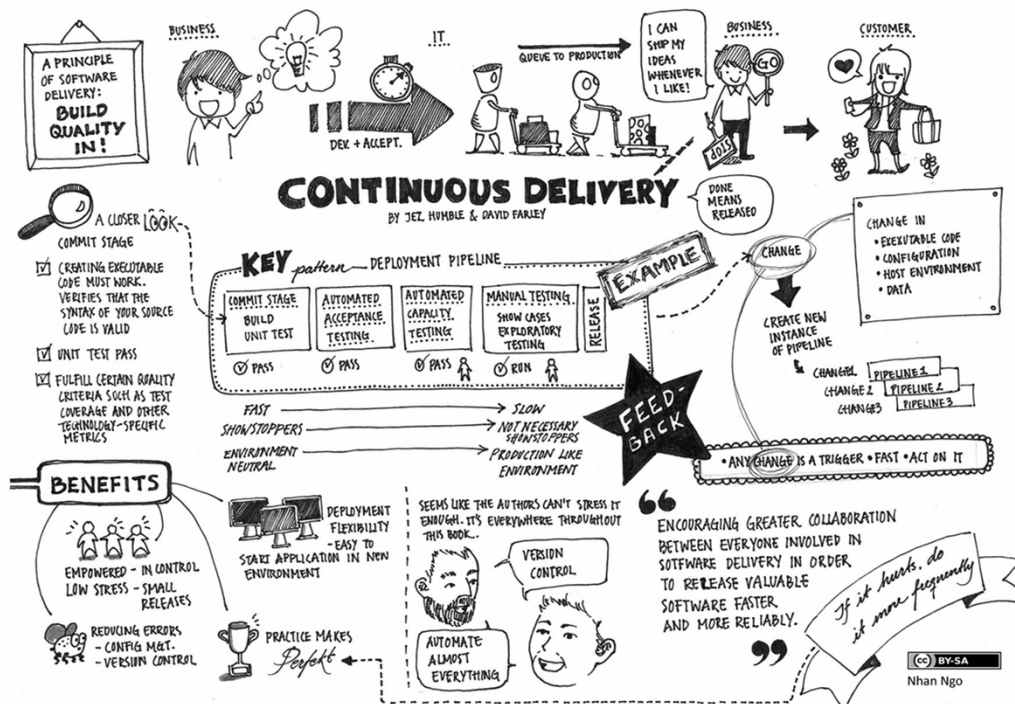
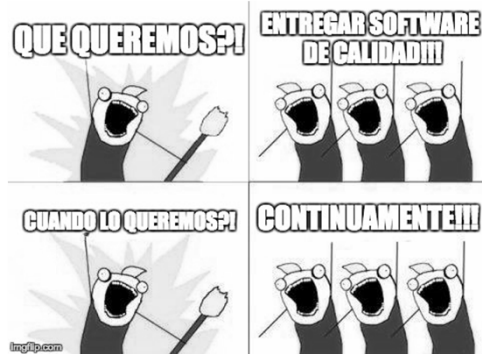
- **Merge-Paranoia**
Evitar la fusión a toda costa, por lo general a causa de un temor a las consecuencias.
- **Merge-Mania**
Gastar demasiado tiempo la fusión de configuración, en lugar de su desarrollo.
- **Big Bang Merge**
Aplazar la fusión hasta el final de las actividades de desarrollo y de intentar fusionar todas las ramas simultáneamente.
- **Development Freeze**
Detener desarrollo mientras que ejecutan ramificaciones, fusiones, o la construcción de nueva línea base.
- **Never-Ending Merge**
Fusiones continuas, porque siempre queda algo por integrar.
- **Wrong-Way Merge**
Fusionar una versión de componentes con una versión obsoleta.
- **Branch Mania**
Generar ramificaciones sin una versión aparente.
- **Cascading Branches**
Generar ramificaciones, pero nunca actualizar la línea base.
- **Berlin Wall**
Las ramificaciones dividen al equipo de desarrollo, en lugar de dividir su trabajo

DevOps

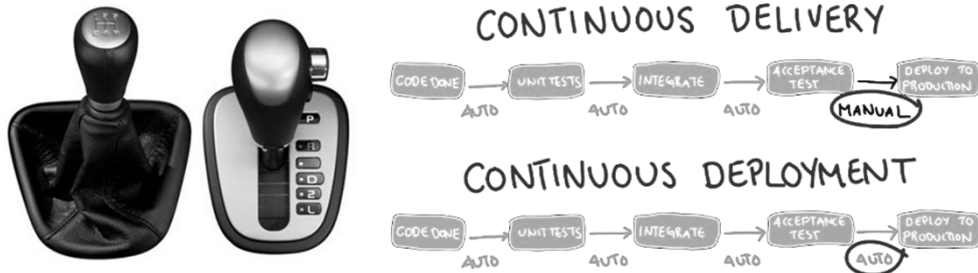


Continuous Integration

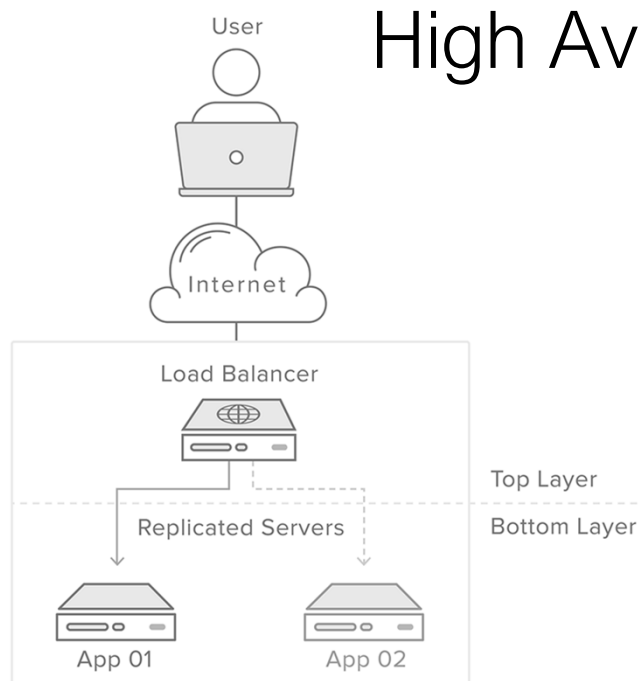




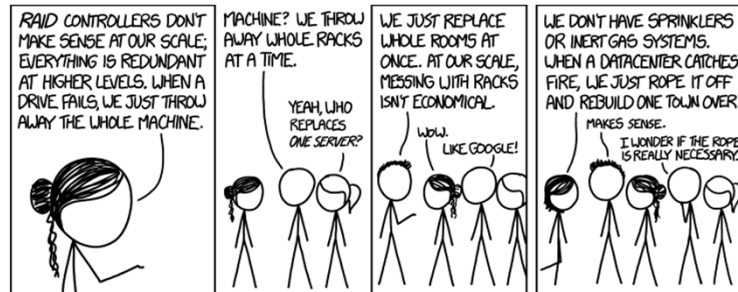
Continuous Deployment



High Availability



High Availability



- Una instancia por vez
- Bloques de instancias (50, 50)
- Todas de una vez

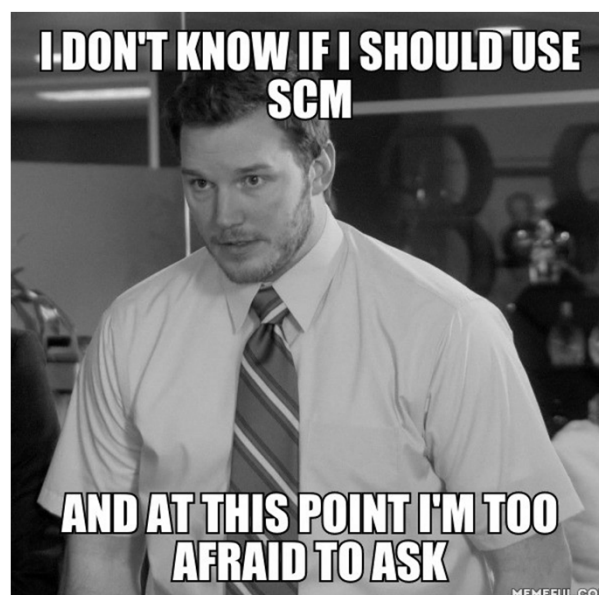
La disponibilidad se expresa con un índice...

Índice de disponibilidad	Duración del tiempo de inactividad
97%	11 días
98%	7 días
99%	3 días y 15 horas
99,9%	8 horas y 48 minutos
99,99%	53 minutos
99,999%	5 minutos
99,9999%	32 segundos

Algunos Tips

- Hacer de la Gestión de Configuración el trabajo de todos
- Crear un ambiente y un proceso de ingeniería que permita la Gestión de Configuración
- Definir y documentar el proceso de CM/Ingeniería , luego seleccionar la/las herramientas que le den soporte al proceso.
- El personal de CM debe contar con Individuos con expertise técnica para dar soporte al desarrollo y mantenimiento del producto
- Los procedimientos y el Plan de SCM debe realizarse en las etapas iniciales del proyectos

Preguntas



Referencias

- **Sommerville, Ian** - INGENIERÍA DE SOFTWARE - Novena Edición (Editorial Addison-Wesley Año 2011).
- Bersoff, E.H., "Elements of Software Configuration Management",
- IEEE Transactions on Software Engineering, vol 10, nro. 1, enero 1984, pp 79-87
- Little Book of Configuration Management – <http://www.spmn.com>
- SCM & the Agile Manifesto - <http://www.scmpatterns.com/agilescm/>