

SCM

- **SCM** (Gestión de la configuración de software): Es una especificación de la gestión de configuración (CM). Su propósito: Integridad, visibilidad, trazabilidad, reproducibilidad y control.
- **CM**: Tiene como propósito establecer la integridad ante los cambios y administrar la evolución (Procesos, productos, ambientes).
- **Definiciones básicas**
 - **Configuración**: Componentes, documentos e información que definen una versión determinada del producto a entregar.
 - **Item de configuración**: Elemento involucrado en el desarrollo del producto y que está bajo el control de la gestión de configuración.
 - **Linea base**: Estado de la configuración de un conjunto de ítems en el ciclo de desarrollo que puede tomarse como punto de referencia para una siguiente etapa del ciclo. Ej: Requerimientos, para modificarlos hay que volver al consenso.
- **Etapas**
 - **Administración del proceso SCM**: Que se gestionará en el proyecto y que transversalmente en toda la compañía
 - **Identificación de la configuración**: Items controlados (IC), Baseline, Relaciones IC, Componentes terceros, Librerías,
 - **Control de las configuraciones del SW**: Solicitar, evaluar, aprobar cambios; Planificar cambios, Control y verificación de la implementación.
 - **Estado y Conteo**: Registrar y reportar la información necesaria para administrar la configuración de manera efectiva.
 - **Auditoria de la configuración**: Verificación del estado de la configuración, pueden ser informales (Checklists) o exhaustivas (Planificadas).
 - **Funcional**: Verifica cumplimiento de requerimientos
 - **Física**: Verifica que los IC sean consistentes con la documentación.
 - **De proceso**: Verifica que se haya cumplido el proceso de SCM
 - **Release Management**:
 - **Building**: Combinar versiones correctas de IC en un programa ejecutable.
 - **Release Management**: Administración, identificación y distribución de un producto. Como voy a hacer para evolucionar el aplicativo?
- **SCMP** (Plan de la configuración del proyecto): Resultado de la actividad de planificación del SCM

SQA

- Proporcionar **PPQA**: Evaluación objetiva de procesos, productos y servicios en función de estándares y procedimientos ; Identificar y documentar no-conformidades ; Feedback al personal y gerencia ; Tratar no-conformidades.
- **SQA**: Que es lo que será evaluado, frecuencia, como se ejecutará la evaluación, quienes estarán involucrados en la evaluación. -> Reporte de evaluación, lecciones aprendidas, seguimientos de las no conformidades, mantenimiento de Quality Trends.
- Apunta a buscar el defecto, pruebas estáticas.
- **Revisión de Pares**: Remover defectos de forma temprana en el desarrollo del SW. (No se buscan soluciones, focalizarse en el producto y no en las personas).
 - **Inspección**: Un equipo de personas revisa los componentes del objeto siguiendo un proceso documentado. (Se les hace seguimiento hasta su corrección)(Proceso formal).
 - **Inspección de código**: Detectar defectos de programación, cumplimiento con los requisitos y estándares de construcción. (Complementarias al testing)
 - Equipo: Moderador, Autor, Lector, Inspectores, Registrador
 - Etapas: Planificación de la inspección por el moderador; Reunión general el autor describe los antecedentes del producto; Preparación el inspector examina el producto; Reunión de inspección se debate sobre los defectos encontrados y se llega un acuerdo con los cambios a realizar; El autor realiza los cambios y se hace un seguimiento.
 - **Team Review**: Inspecciones light, sin requerir los roles. (Se distribuye el material a revisar, antes de la sesión. En la sesión se notifican los defectos.)
 - **Walkthrough**: De interés por el autor. Solicita comentarios a grupo de pares. A diferencia de las inspecciones que busca alcanzar los objetivos de calidad, este trata de servir las necesidades del autor. (Informal)(Sin seguimiento, ni procedimientos).
 - **Par de programadores**: Dos desarrolladores trabajen en el mismo producto, al mismo tiempo. No es una técnica de revisión, sino una estrategia de desarrollo.
 - **Peer desckcheck**: El producto es examinado por un revisador, y el autor se desentiende completamente de la revisión y desconoce el criterio con el cual es revisado su trabajo.
 - **Passround**: Desckcheck pero colectivo. Ej: un gdoc donde se ve plasmado un trabajo y varios revisadores van dejando sus anotaciones para que luego el autor se nutra de las mismas.
 - **Ad Hoc**: Che, no tienes 15 minutos y me ayudas a revisar esta función. (Pedir ayuda).
 - Errores comunes de revisión es de SW: Los participantes no entienden el proceso de revisión, critica al autor en vez de al producto (VER ATRÁS)

Testing

- Asegurar la calidad vs Controlar la calidad
- Testing: Técnica de autoría funcional dinámica para encontrar fallas. Debo tener definido el requerimiento.
- Proceso del testing (VER ATRÁS)
- Incidente de Testing: Evento durante ejecución del test que requiere investigación. (No toda incidencia es una falla).
- **Equivocación**: Acción humana que produce un resultado incorrecto.
- **Defecto**: Paso, proceso o definición de dato incorrecto.
- **Falla**: Resultado de una ejecución incorrecta. Manifestación de un defecto.
- El defecto está dado por una equivocación en el código, la falla en la hora de ejecución. (Un defecto puede o no generar una falla).
- **Condiciones de prueba**: Descripciones de situaciones que quieren probarse.
- **Casos de prueba**: Instancia de una condición de prueba.
- **Criterio de selección**: Condición para seleccionar un conjunto de casos de prueba.
- **Clase de equivalencia**: Posibles casos de prueba divididos en clases, todos los casos de una clase son equivalentes si detectan los mismos defectos.
- **Depuración**: Eliminar un defecto. La depuración no es una tarea de prueba aunque es consecuencia de ella. (Detectar, depurar, volver a probar, aprender para el futuro)
- Como nunca se puede demostrar que un programa es correcto, continuar probando es una decisión económica. Cuando parar de probar?
 - Pasa exitosamente el conjunto de pruebas diseñado
 - Cierta cantidad de fallas no críticas es aceptable
 - Cantidad de fallas detectadas es similar a la cantidad de fallas estimadas
- **Tipos de prueba**
 - **Prueba unitaria**: Caja negra y Caja blanca
 - **Prueba de integración**: Verifica que los bloques aislados que funcionan bien, también lo hagan al acoplarse al sistema. (No incrementales, Incrementales).
 - **Prueba de aceptación de usuario**: (Prueba de caja negra) realizada por los usuarios que verifican que se ajusta a sus requerimientos.
 - **Prueba de stress**: Se somete al sistema excediendo los límites de su capacidad (CPU, DISCO).
 - **Prueba de Volumen y Performance**: Verificar que el sistema soporta los volúmenes definidos en los requerimientos.
 - **Prueba de Regresión**: Probar lo viejo, luego de introducir un cambio.
 - **Prueba tipo alfa y beta**: Primera versión al usuario para que lo pruebe. Alfa: el usuario lo hace en mis instalaciones; Beta: el usuario lo hace en sus instalaciones.
- Testing y Ciclo de vida (VER ATRÁS): Modelo del ciclo de vida en V
- **Prueba Funcional**

- Llamada Prueba caja negra: Prueba funcional producida por los datos o producida por la entrada/salida.
 - **Clases de equivalencias**
 - Entradas: Clases validas, Clases Invalidas
 - Rango de valores: una valida y dos invalidas.
 - Conjunto de valores: Una valida y una inválida.
 - Debe ser: una valida y una inválida.'
 - **Condiciones de borde**
 - **Clases inválidas:** Se prueba valores de otro tipo. Ej: números en campos alfabéticos. Si el estado civil es divorciado, los datos del cónyuge se deben ignorar.
 - **Conjetura de errores:** Probamos lo que sospechamos que puede andar mal.
- **Prueba estructural**
 - Llamada Prueba caja blanca: Prueba lo que el software hace, se basa en como está estructurado internamente y su definición.
 - Grado de cobertura: Sentencias (Bloques), Decisiones, Condiciones, Camino básico (Independientes) (Ver atrás diagrama).
 - Complejidad ciclomatica: Medición cuantitativa de la complejidad lógica de un programa. Cantidad de caminos independientes, numero de regiones del grafo, Aritas – Nodos + 2
- Prueba caja gris: Combina elementos de la caja negra y la caja blanca. Se conoce parte de la implementación o estructura interna, el conocimiento es parcial no total.
- Las pruebas de caja blanca se complementa con las pruebas de caja negra.
- Las pruebas de caja gris prueba el SW como si fuera caja negra, pero suman condiciones y casos adiciones derivados del conocimiento de la operación.
- Un requerimiento que no es testeable no es implementable

Métricas

- **Medida:** Indicación cuantitativa de algún atributo.
- **Métrica:** Medida cuantitativa del grado que un sistema posee de algún atributo dado.
- **Indicador:** Combinación de métricas que proporcionan una visión más profunda.
- **Como especificar una métrica**
 - Objetivo
 - Descripción de la métrica
 - Método de medición (Fuente, Formula, Visualización)
 - Frecuencia de cálculos
 - Valores esperados
- **Como implementar un programa de métricas**
 - Identificar objetivos
 - Definir las métricas
 - Recolectar datos históricos
 - Automatizar el proceso
 - Utilizar las métricas en la toma de decisiones
- **GQM: Goal Question Metric**
 - Objetivo: Definir que métricas implementar y como utilizarlas.
 - Se basa en una jerarquía de tres niveles
 - Conceptual (Goal): meta, aquello que la organización quiere alcanzar
 - Operacional (Question): Las preguntas que debe responder las métricas para definir el cumplimiento de las metas.
 - Cuantitativo (Metrics): Las mediciones necesarias para ayudar a responder a la pregunta.
 - Metas -> Preguntas -> Métricas -> Recolectar, validar, analizar los datos obtenidos
- **Clasificación:**
 - Métricas del proceso: Métricas de gestión de proyectos
 - Duración promedio proyecto
 - Promedio de desvío en proyectos
 - Métricas del producto SW: Tamaño, calidad del producto
 - Ej: Cantidad líneas de código, Puntos de función, Acoplamiento, Calidad.