

Procesos en la gestión de la calidad

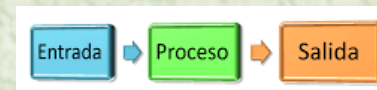
Procesos ISO 21500

Realizar el control de la calidad



Contenidos

Descripción del proceso



Herramientas para el control de la calidad



Descripción del proceso

Realizar el control de la calidad

Objetivos

- ✓ Hacer **seguimiento a los entregables y de los procesos** que se están cumpliendo y **detectar los defectos** mediante el uso de herramientas, procedimientos y técnicas establecidas.
- ✓ **Analizar las posibles causas** de los defectos
- ✓ Determinar las **acciones preventivas y las solicitudes de cambio**
- ✓ **Comunicar** las acciones correctivas y las solicitudes de cambio a los miembros adecuados a la organización del proyecto.



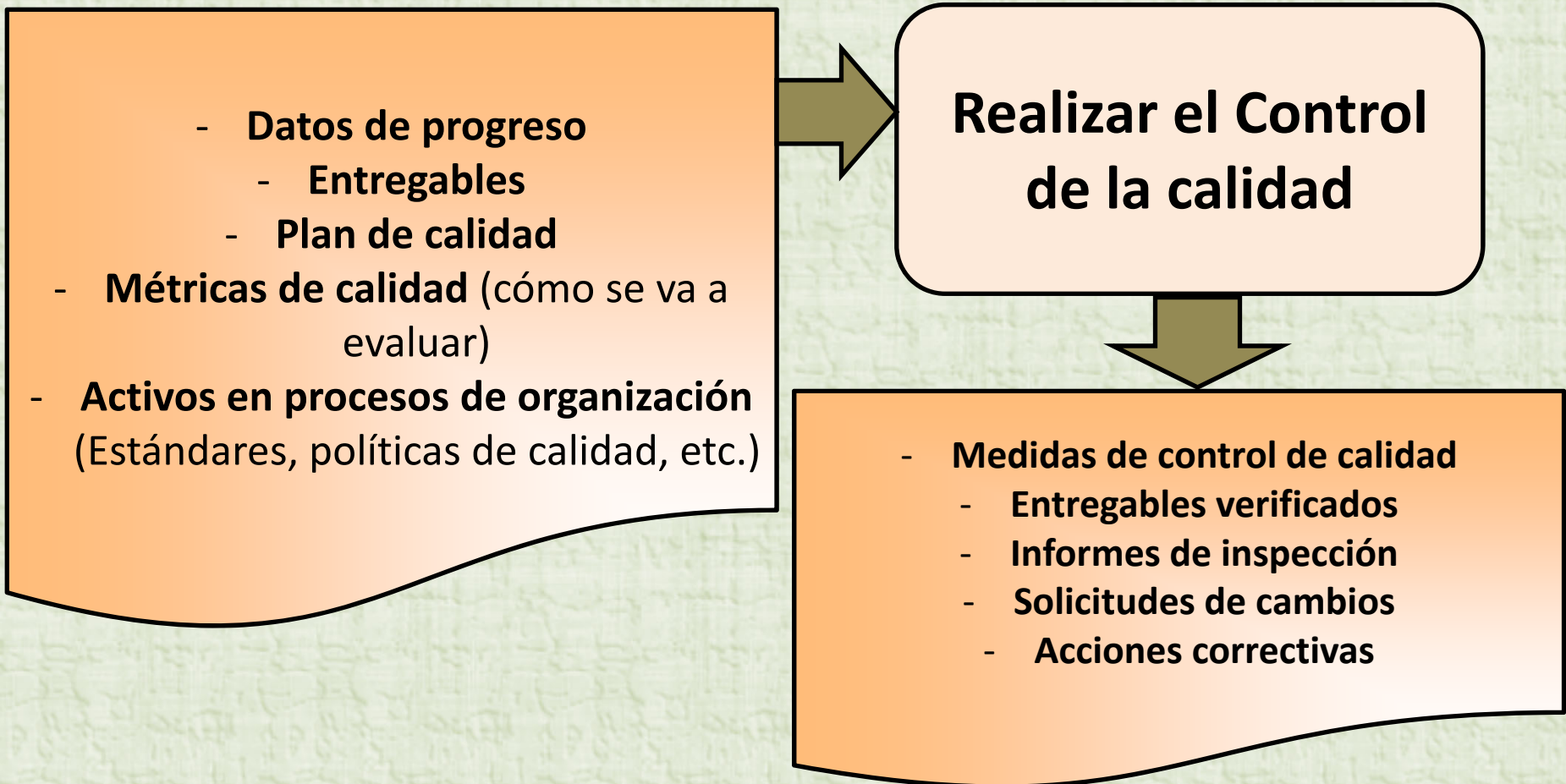
Realizar el control de la calidad

Diferencias entre control y aseguramiento de la calidad

- ✓ Cuando un producto no posee una determinada característica de calidad se dice que tiene un **defecto**.
- ✓ El **control de calidad** tiene como objetivo **detectar los defectos** y realizar acciones para corregirlos
- ✓ Mientras que el **aseguramiento de la calidad** tiene como misión intentar **evitar que se produzcan** los defectos.



Descripción del proceso



Herramientas para el control de la calidad



Dos enfoques desde el punto de vista del software

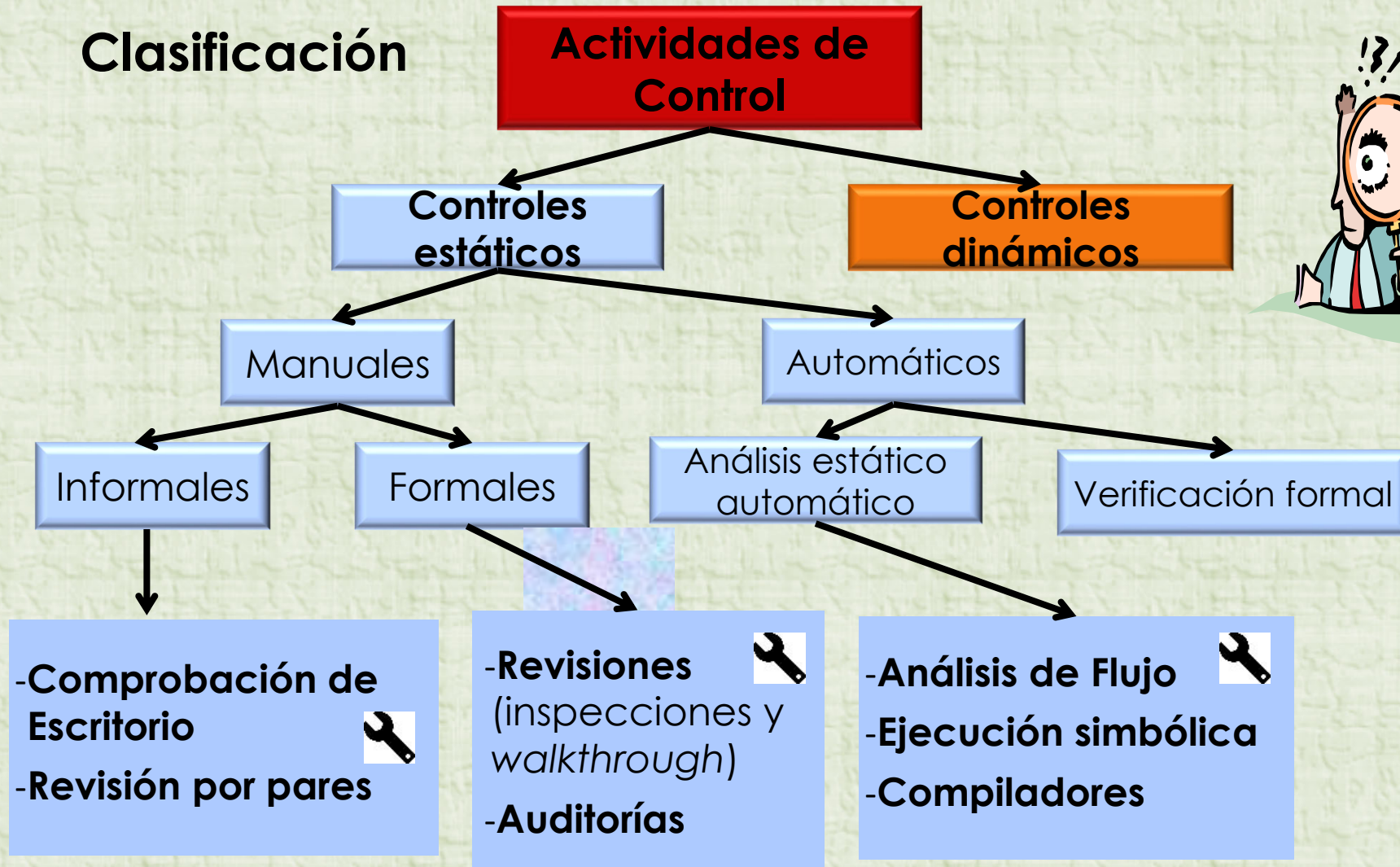
- ✓ **Controles estáticos:** el software, documentación y procesos son revisados por una o más personas.
- ✓ **Controles dinámicos:** aquellos que requieren la ejecución del objeto (artefacto software) que se está probando o de un modelo del mismo.



Herramientas para el control de la calidad



Clasificación



Herramientas para el control de la calidad



- **Revisiones técnicas formales**
- **Inspecciones**
- Auditorías
- Walkthrough (visita guiada)
- Revisión por pares
- Análisis de flujo de datos



Revisiones Técnicas Formales



Las **revisiones** son **técnicas estáticas** que sirven para detectar defectos que puedan así ser eliminados.

Objetivos:

- ✓ Descubrir errores en la función, lógica o implementación de cualquier representación del software.
- ✓ Verificar el cumplimiento de los requisitos
- ✓ Garantizar el cumplimiento de los estándares.
- ✓ Conseguir un desarrollo uniforme del software
- ✓ Obtener proyectos que hagan más sencillo los trabajos técnicos (análisis que permitan buenos diseños, diseños que permitan implementaciones sencillas, estrategias de efectivas, etc.)

Revisiones Técnicas Formales



Se revisa un producto

(especificación, módulo, listado,...)

Poca gente, preparación y duración breves



Decisión final:

- Aceptación
- Rechazo
- Aceptación condicionada a pequeñas modificaciones

Participantes: jefe de revisión, revisores (ingenieros, programadores,...) y productor

Revisiones Técnicas Formales



Se aplican en diversos momentos del desarrollo para detectar defectos.

Las revisiones son un método de control de calidad eficaz en las fases iniciales del desarrollo.

- ✓ Diseño: entre el 50 y el 60% de los errores del desarrollo.

La diversidad de un grupo de personas para:

- ✓ Señalar la necesidad de mejoras en el producto software (diagramas del análisis, diccionario de datos, diseño, código, estrategia de pruebas,...)
- ✓ Confirmar las partes en las que no es necesaria una mejora.
- ✓ Conseguir un trabajo técnico de calidad más uniforme.

Efectividad: se calcula que son efectivas en un 75%.

Herramientas para el control de la calidad



- **Revisiones técnicas formales**

- **Inspecciones**

Revisiones en las que los participantes van leyendo el documento y comprobando en cada a paso el cumplimiento de los criterios en una lista de comprobación

- **Auditorías**

Una auditoría consiste en realizar una investigación para certificar conformidad e identificar desviaciones

- **Walkthrough (visita guiada)**

Se demuestra la funcionalidad del objeto revisado mediante la simulación de su funcionamiento con casos de prueba y ejemplos

- **Revisión por pares**

Consiste en la revisión del código de un programador por otros programadores (sus pares)

- **Análisis de flujo de datos**

Analizar el comportamiento de un programa. Se basa en una representación de grafos (los nodos representan sentencias o segmentos y los arcos transiciones)