

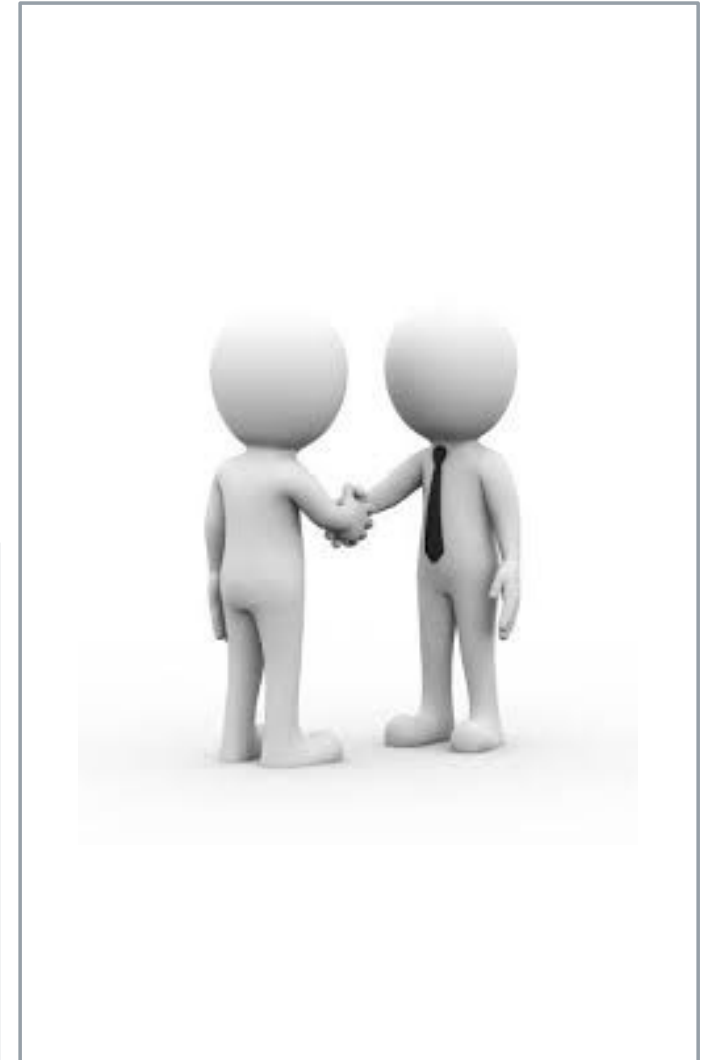
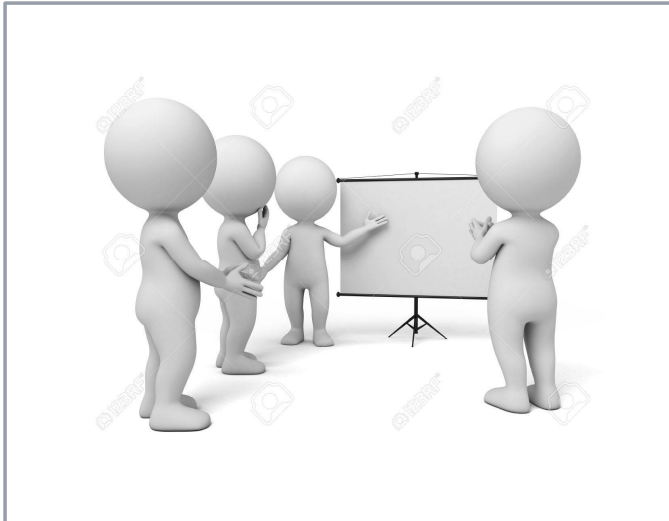
ROLES DEL EQUIPO DE DESARROLLO

■ Cliente

- Poner en marcha el proyecto.
- Pagar las cuentas.
- Definir el resultado final.

■ El Funcional

- Entender las necesidades del cliente.
- Asegurar que la solución que está siendo desarrollada se ajuste a las necesidades.
- Releva requisitos.
- Coordinar reuniones con clientes.
- Redactar las especificaciones funcionales.



ROLES DEL EQUIPO DE DESARROLLO

■ El Arquitecto

- Traducir los requisitos, tal como se define por el analista, en una solución técnica.
- Crear un diseño técnico, o simplemente algunos bocetos, de cómo el sistema va a estar estructurado.
- Pensar en el sistema antes de que se desarrolle.
- Decidir qué camino tomar, en base a la arquitectura global que ha elegido.
- Realizar un seguimiento del desarrollo, para ver si todavía se mantiene en consonancia con el diseño general



ROLES DEL EQUIPO DE DESARROLLO

El Desarrollador

- Hacer el seguimiento de su propio progreso,
- Informar al jefe de proyecto de los problemas a los que se enfrenta.
- Implementar las ideas del arquitecto
- Discutir las (in)posibilidades de la implementación con el arquitecto.

El Tester

- Asegurar que el software cumpla con las especificaciones del cliente.
- Asegurar que no haya errores introducidos por los desarrolladores.
- Asegurar que la aplicación o software fabricado sea consistente visualmente.
- Aportar un punto de vista de usuario para lograr una mejor experiencia en el uso de la aplicación

MODELOS DE CICLO DE VIDA DEL DESARROLLO DE SOFTWARE

■ Un **modelo de ciclo de vida de desarrollo de software** describe los tipos de actividad que se realizan en cada etapa de un proyecto de desarrollo de software, y cómo las actividades se relacionan entre sí de forma lógica y cronológica.

Modelo de Desarrollo Secuencial:

- Modelo de Cascada.
- Modelo en "V"

Modelos de Desarrollo Iterativos Incrementales:

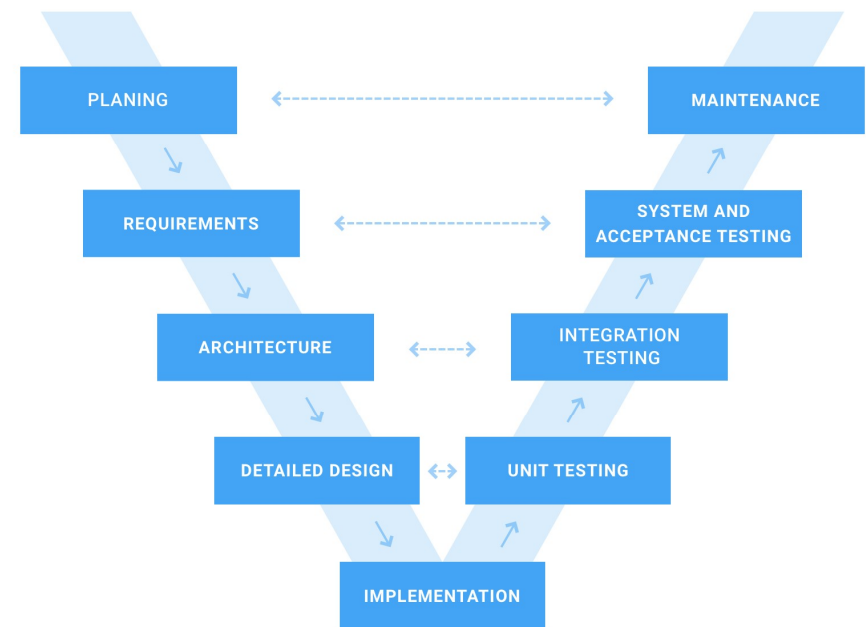
- Desarrollo incremental.
- Desarrollo iterativo.
 - ✓ Scrum
 - ✓ Spotify
 - ✓ DevOps

VISTAZO POR LOS ENFOQUES DE TRABAJO PARA EL DESARROLLO DE SOFTWARE

■ *Enfoque cascada (secuencial)*

- Las actividades ocurren de manera secuencial, significa que cada actividad debe completarse para que inicie la siguiente.
- Bajo este contexto, las pruebas ocurren cuando la documentación funcional y el desarrollo están listos.
- Ventajas
 - Las pruebas cuentan con los insumos completos para trabajar.
 - Es un enfoque que genera pocos cambios de alcance durante el flujo de actividades
- Desventajas
 - Siendo que las actividades son secuenciales, los tiempos de entrega al cliente son muy altos.

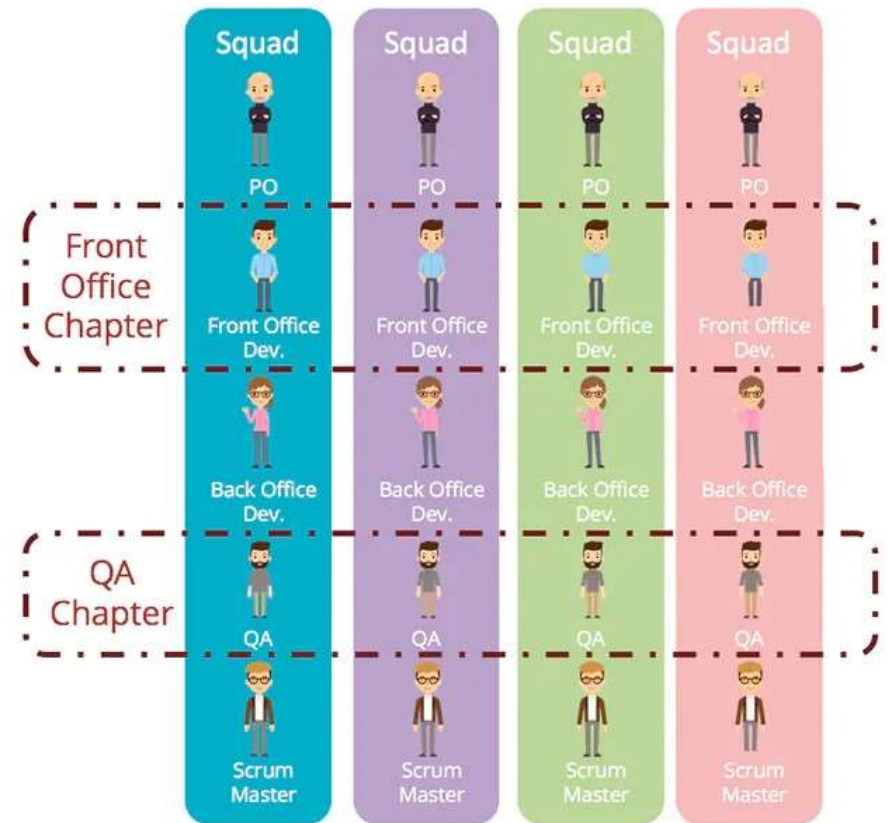
V-Model



VISTAZO POR LOS ENFOQUES DE TRABAJO PARA EL DESARROLLO DE SOFTWARE

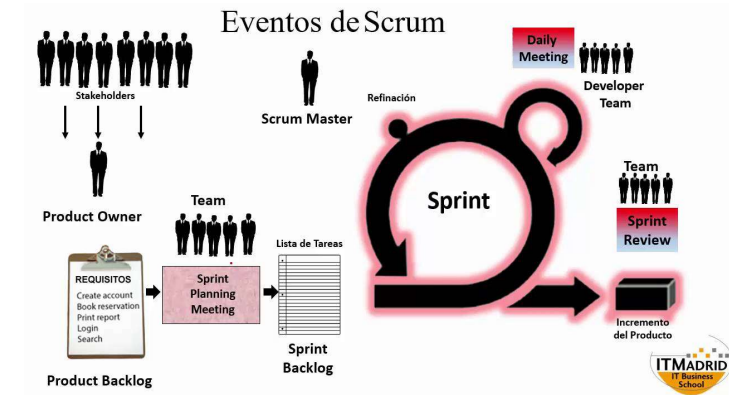
■ Spotify

- Enfoque de trabajo auto-organizado donde las estructuras tradicionales son reemplazadas.
- En general, las decisiones asociadas a los entregables cuentan con la participación de todo el equipo durante todo el proceso de desarrollo.
- Equipos de trabajo
 - Tribus a cargo de áreas de negocio
 - Squad a cargo de funcionalidades o pequeños proyectos
 - Chapters a cargo de grupos de interés.
- Ventajas
 - Permite trabajar específicamente en la funcionalidad del negocio en la que está asignado
 - Participación de todo el equipo
- Desventaja
 - Segmentación de los equipos



Scrum

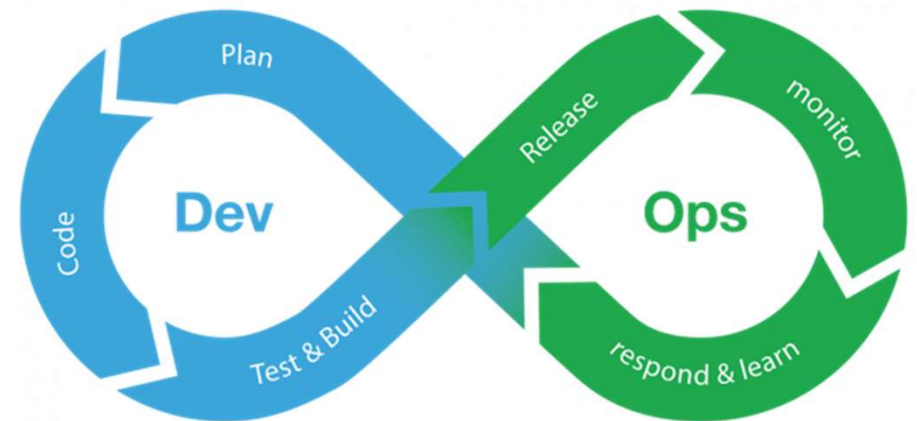
- Enfoque de trabajo en equipos auto-organizados donde cada actividad se define en consenso.
- Los entregables al cliente ocurren en un tiempo fijo y periódico (entre 2 a 8 semanas).
- El trabajo parte de historias de usuario que deben desarrollarse con la participación de desarrollo y testing.
- Este enfoque se apoya mucho en la automatización de las pruebas.
- Ventajas
 - Participación de todo el equipo
- Desventajas
 - Finalmente, y según su orientación puede terminar en 'mini-cascada'.



VISTAZO POR LOS ENFOQUES DE TRABAJO PARA EL DESARROLLO DE SOFTWARE

■ Devops

- Enfoque de trabajo en equipo que tiene como norte llevar los desarrollos a producción lo más pronto posible y de forma continua.
- Las decisiones se toman en equipo porque participa durante todo el proceso de desarrollo.
- Impulsa la automatización de actividades, entre ellas las pruebas, a su máximo nivel.
- El equipo de Testing debe ser mucho más técnico para convertirse en un equipo de Testing Continuo.
- Ventajas
 - Liberaciones a producción continuas y por ende, de menor impacto que agregan valor periódicamente.
- Desventajas
 - Riesgo de subestimar el esfuerzo manual del Testing.



TIPOS DE PRUEBAS

Funcionales

No Funcionales

Asociadas al cambio



TIPOS DE PRUEBAS

- **Funcionales:** Son las que se evalúan **qué** hace el sistema
 - Adecuación: ¿el sistema es adecuado a la audiencia a la cual va dirigido?
 - Precisión: ¿el sistema cuenta con la exactitud en los cálculos que se espera?
 - Cumplimiento de la funcionalidad: ¿el sistema contiene todas las funcionalidades esperadas?
 - Interoperabilidad: ¿el sistema interactúa con otros sistemas tal como se espera?
 - Seguridad: ¿el sistema cuenta con mecanismos de prevención de accesos indebidos/no autorizados?

TIPOS DE PRUEBAS

- **No Funcionales:** Son las que evalúan cómo trabaja el sistema
 - Fiabilidad: ¿el sistema es confiable y está disponible cuando se espera?
 - Usabilidad: ¿el sistema es amigable/intuitivo?
 - Eficiencia (Desempeño): ¿el sistema tiene los tiempos de respuesta esperados? ¿el sistema responde adecuadamente cuando hay una concurrencia de usuarios?
 - Mantenibilidad: ¿el sistema es fácil de mantener/actualizar ante cambios?
 - Portabilidad: ¿el sistema funciona bajo las plataformas esperadas?

TIPOS DE PRUEBAS

- **Asociadas al Cambio:** Son aquellas que se derivan de los cambios de mantenimiento al sistema
 - Regresiones: ejecutar un set de pruebas con el objetivo de detectar algún efecto secundario producto de una modificación en el código.
 - Re-pruebas: volveré a probar un caso de prueba que falló y fue corregido por el equipo de desarrollo.

NIVELES DE PRUEBAS

Componente

Integración

Sistemas

Aceptación



NIVELES DE PRUEBAS

Componente

- También llamadas unitarias, son las pruebas que evalúan un método, una clase, un entregable de código atómico.
- Generalmente se prueba desde el IDE y lo hace el desarrollador.
- Debería contemplar cualquiera de los tipos de prueba.
- Se espera detectar defectos y corregirlos.

NIVELES DE PRUEBAS

Integración

- Son las pruebas que evalúan la interacción entre varios componentes, por ejemplo un componente que llama a otro.
- Generalmente su mejor representante son los servicios web (APIs) que representan una funcionalidad del backend que va a base de datos.
- Usualmente la suelen ejecutar los desarrolladores pero resulta muy útil para los testers más técnicos para detectar potenciales fallas o cuando las entregas del sistema están demoradas.
- Debería contemplar cualquiera de los tipos de prueba.
- Se espera detectar defectos para reportarlos y puedan ser corregidos.

NIVELES DE PRUEBAS

Sistemas

- Son las pruebas que mayormente ejecutan los testers donde evalúan el sistema desde el punto de vista del usuario.
- Debería contemplar cualquiera de los tipos de prueba.
- Se espera detectar fallas para reportarlas y que puedan ser corregidas.

NIVELES DE PRUEBAS

Aceptación

- Con las pruebas de comprobación del requisito funcional, ejecutadas por los usuarios (o sus representantes) para generar confianza en el sistema cuando pase a producción.
- No se espera detectar fallas.

PROCESO DE PRUEBAS



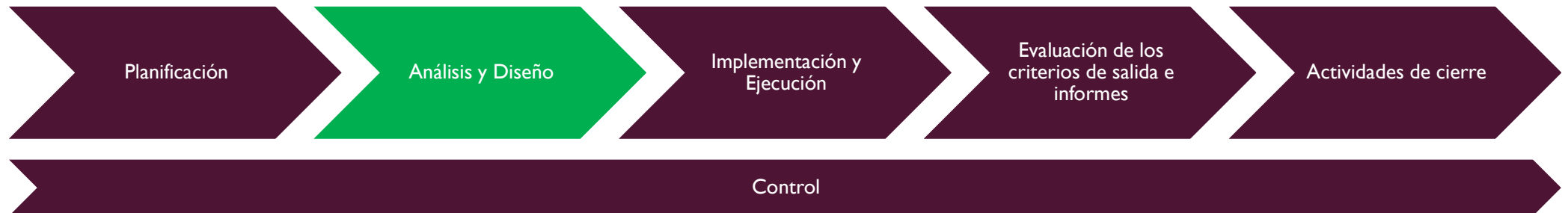
PROCESO DE PRUEBAS

- A partir de los insumos (base de pruebas) se arma el documento Plan de Pruebas que tenga en cuenta el alcance a cubrir en las pruebas: qué probar, cómo se va a probar, qué condiciones se requieren, qué suposiciones y restricciones se tienen para así tener una estimación de esfuerzo del proceso de pruebas.



PROCESO DE PRUEBAS

- Se realiza la revisión detallada de la base de pruebas para definir las condiciones de prueba a tener en consideración en el proceso de pruebas.
- A partir de las condiciones de prueba, se desarrollan los casos de prueba teniendo en cuenta la prioridad de cada uno de ellos e información detallada que luego facilite la ejecución correspondiente. El entregable de esta fase es la Matriz de Pruebas



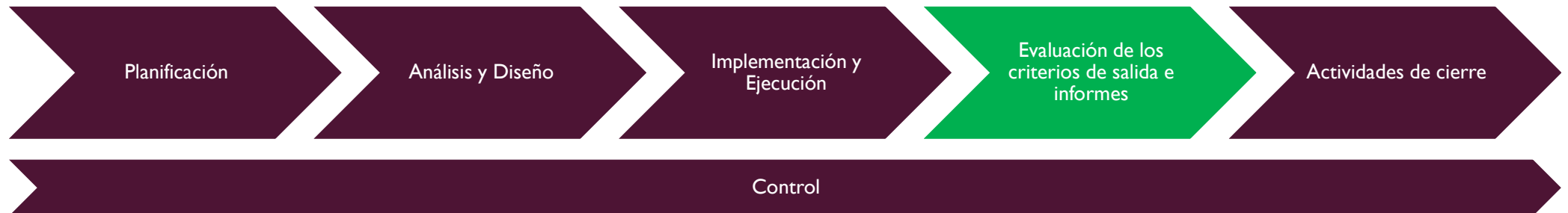
PROCESO DE PRUEBAS

- Consiste en las actividades de preparación de datos y del entorno justo antes de recibir la entrega a probar.
- Al recibir el desarrollo se procede a la ejecución de los casos de prueba en función a la prioridad, se toman evidencias de las pruebas exitosas y se reportan las fallas que se detecten. El entregable de esta fase es el Documento de Evidencias y Reportes de Fallas.



PROCESO DE PRUEBAS

- Al ejecutar las pruebas, ocurrirá que se reciben nuevas versiones con adecuaciones y corresponde hacer re-pruebas y regresiones con lo cual será importante tener en cuenta cuando se cumplan los criterios de salida para así finalizar la ejecución.
- Finalmente al cierre del proyecto/sprint será valioso consolidar la información relevante del proceso de pruebas en un informe final.



PROCESO DE PRUEBAS

- Al ejecutar las pruebas, ocurrirá que se reciben nuevas versiones con adecuaciones y corresponde hacer re-pruebas y regresiones con lo cual será importante tener en cuenta cuando se cumplan los criterios de salida para así finalizar la ejecución.
- Finalmente al cierre del proyecto/sprint será valioso consolidar la información relevante del proceso de pruebas en un informe final.



PROCESO DE PRUEBAS

- Durante todo el proceso de pruebas, debe hacerse un seguimiento que consiste en:
 - Evaluar si lo establecido en el Plan de Pruebas sigue vigente.
 - Medir los resultados que se van obteniendo.
 - Levantar alertas tempranas en caso de retrasos.
- Para ello se suelen llevar tableros de control, reportes de estatus periódicos, entre otros elementos



ESTÁNDARES DE PRUEBA



ISTQB: Principal estándar de prácticas de Testing a nivel mundial e igualmente principal ente certificador (individual) en el área.

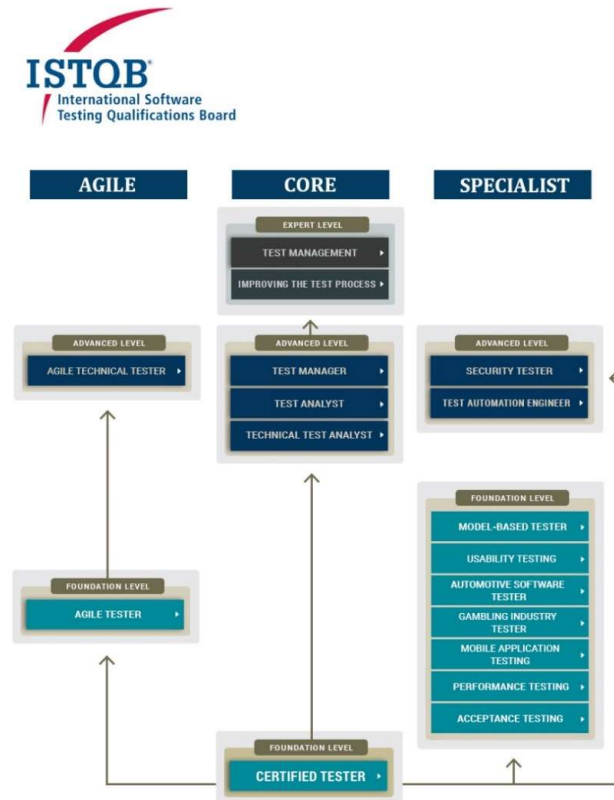


TMMI: Unidad que evalúa y pondera del 1 al 5 el nivel de madurez de un equipo de Testing. Plantea los principios de madurez en cada uno de los niveles.

ISTQB

- Principal estándar de pruebas de software.
- Tercer estándar tecnológico con mayor número de personas certificadas después de ITIL y PMI.
- Provee los conceptos claves sobre el trabajo de prueba muchos de los cuales son repasados en la presenta capacitación.

■ Sitio Oficial: <https://www.istqb.org/>



TMMI

- Modelo de Madurez de equipos de prueba en función a las prácticas, procesos, entregables, roles y objetivos que emplean en su día a día.
- En función a auditorías, un ente certificador determina la madurez en el equipo de pruebas.



NORMAS Y ESTÁNDARES



- ISO 9126 (hoy ISO 25000) Atributos de la calidad
- IEEE 829 Plantillas del proceso de pruebas
- IEEE 1024 Revisiones