

«Talento Tech»

Testing QA

Clase 12



Clase N°12 | Métricas y su interpretación

Temario:

- Test Management
 - o KPI's de Calidad de Software
 - o Indicadores de Calidad
 - o Indicadores de Velocidad.
 - o Indicadores de Costos.
 - o Uso de Indicadores
- Test Management cómo interpretar los indicadores calculados

Objetivos de la clase

En esta clase vamos a introducirnos en el mundo de las **métricas en QA** y su interpretación. Aprenderemos qué son los **indicadores de calidad, velocidad y costo**, por qué son importantes en la gestión de pruebas y cómo se utilizan para evaluar el estado y la eficiencia de un plan de testing. También vamos a comparar estos indicadores con **estándares de la industria**, lo cual nos va a permitir reflexionar sobre la madurez de un proceso de QA. Finalizaremos con ejercicios prácticos de cálculo, análisis de KPIs y comparaciones con métricas de referencia.

Test Management y métricas

El **Test Management** o gestión del testing no sólo incluye planificar y ejecutar pruebas, sino también **medir resultados y tomar decisiones** en base a esos datos. Para eso se utilizan **indicadores clave de rendimiento (KPIs)**.

¿Qué es un KPI?

Un **KPI (Key Performance Indicator)** es un indicador clave de desempeño. Es una métrica utilizada para evaluar el éxito de una actividad específica, en este caso del proceso de testing. Nos permite cuantificar resultados, detectar desvíos y tomar decisiones informadas.

KPI's de Calidad de Software

Los **KPI (Key Performance Indicators)** o **Indicadores Clave de Desempeño** son métricas cuantitativas (números o porcentajes) que permiten evaluar el rendimiento del proceso de testing. En el contexto de QA, estos indicadores se agrupan principalmente en tres categorías:

- **Indicadores de Calidad**
- **Indicadores de Velocidad**
- **Indicadores de Costos**



A continuación, detallamos los más utilizados en cada categoría junto con ejemplos prácticos:

Indicadores de Calidad

- **Tasa de defectos encontrados:** (Número de defectos encontrados / total de casos ejecutados)
Ejemplo: 12 defectos / 35 casos ejecutados = **34.28%**
- **Defectos por severidad:** % de defectos críticos vs. medios vs. leves.
Ejemplo: 3 críticos de 12 defectos = **25% críticos**
- **Tasa de re-apertura:** Casos donde un defecto se cerró, pero luego se volvió a abrir.
Ejemplo: 2 de 12 defectos se reabrieron = **16.66% de tasa de re-apertura**
- **Densidad de defectos:** Defectos encontrados por módulo, funcionalidad o líneas de código.
Ejemplo: 6 defectos en el módulo de login vs. 2 en el módulo de perfil.

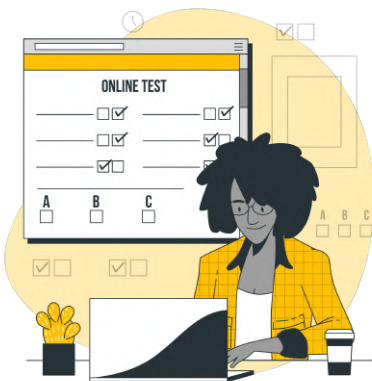
Indicadores de Velocidad

- **Porcentaje de avance:** Casos ejecutados vs. casos planificados.
Ejemplo: $35 / 40 = 87.5\%$ de avance
- **Tiempo promedio por caso de prueba**
Ejemplo: 24 horas / 35 casos ejecutados = **0.685 h por caso**
- **Lead Time:** Tiempo entre el reporte de un bug y su resolución.
Ejemplo: Bug reportado lunes 10:00 y resuelto miércoles 14:00 = **52 horas**
- **Tiempos de ejecución automatizada vs. manual**
Ejemplo: Manual: 7 horas — Automatizada: 2.5 horas

Indicadores de Costos

- **Costo por defecto encontrado:** (Horas invertidas x tarifa) / cantidad de defectos.
Ejemplo: $(24h \times \$50) / 12 = \100 por defecto
- **Horas invertidas vs. presupuestadas**
Ejemplo: $24 / 20 = 120\%$ del tiempo estimado
- **Costo de no calidad:** Costos extra por errores en producción.
Ejemplo: 4 bugs en producción causan 8h de soporte = **\$400**

Uso de Indicadores



Los indicadores permiten gestionar activamente la calidad del proceso. Algunos usos:

- Detectar **áreas críticas** del sistema.
- Verificar cumplimiento de **plazos** y ajustar la planificación.
- Medir la **efectividad de la automatización**.
- Evaluar **costos asociados a errores** y su impacto.
- Facilitar la **toma de decisiones** basadas en datos reales.
- Comparar métricas propias con **estándares de la industria**.
La clave no es sólo calcular números, sino **interpretarlos y accionar en base a ellos**.

Test Management: cómo interpretar los indicadores

Si la tasa de defectos críticos es alta:

Acción sugerida: Revisar requisitos funcionales, calidad del código y definición de criterios de aceptación.

Ejemplo práctico: Se detectaron 3 defectos críticos en el módulo “Carga de CV”. Se organiza una reunión con el Product Owner (PO) para redefinir los criterios de aceptación, se agregan pruebas automatizadas de validación de formato de archivo y se realiza revisión cruzada del código con el equipo de desarrollo.

 **Si el avance de ejecución es lento:**

Acción sugerida: Evaluar si hay sobrecarga de tareas o pruebas poco optimizadas. Repriorizar o redistribuir tareas.

Ejemplo práctico: Sólo se ejecutó el 60% de los casos planificados. Se detecta que un tester estaba dedicado a pruebas manuales repetitivas. Se automatizan 10 casos y se distribuye parte de la carga al segundo tester.

 **Si el Lead Time es alto**

Acción sugerida: Analizar los cuellos de botella en el flujo de resolución de bugs. Escalar si es necesario.

Ejemplo práctico: Un bug crítico tardó 72 horas en resolverse. Se identifica que el ticket pasó 48 horas en espera sin asignación. Se acuerda en daily que los bugs críticos deben tener un responsable asignado en menos de 12 horas.

 **Si el costo por defecto es elevado**

Acción sugerida: Evaluar automatizar pruebas, mejorar la cobertura o identificar errores más temprano en el ciclo.

Ejemplo práctico: El costo por defecto fue de \$100, superando el estándar (\$80). Se decide automatizar los casos funcionales del módulo de “Login”, reduciendo el tiempo de ejecución de 6 a 2 horas por sprint.

Si se usan herramientas como Postman para APIs

Acción sugerida: Aprovechar los datos generados en las ejecuciones (por ejemplo con Newman) para obtener métricas como tiempo de respuesta promedio, tasa de fallos o cobertura de endpoints.

Ejemplo práctico: Se ejecutan 40 requests con Postman y Newman. 5 fallan por timeout. Se revisan los tiempos de respuesta y se detecta un endpoint lento que se optimiza agregando cacheo en el backend.

Estándares de la industria

Para que los indicadores sean realmente útiles, deben compararse con **valores de referencia** aceptados por la industria. Algunos ejemplos típicos incluyen:

- **Tasa de defectos esperada:** Menor al 20% en ambientes de testing funcional.
- **Tiempo promedio por caso de prueba:** 0.5 horas.
- **Porcentaje de ejecución óptimo en un sprint:** Mayor al 90%.
- **Costo por defecto aceptable:** Igual o menor a \$80 USD.

Estos estándares pueden variar según el tipo de industria, el tamaño del equipo o la criticidad del sistema, pero sirven como guía para evaluar la **madurez y eficiencia** del proceso de testing.

Modelo de Reporte de Métricas de QA — Ejemplo Aplicado

Contexto del Proyecto

En la empresa **Talento Digital**, se está desarrollando una plataforma web llamada **"JobMatch"** que conecta postulantes con ofertas de trabajo. Actualmente se está trabajando en el **Sprint 5**, enfocado en mejorar la funcionalidad de carga y gestión de CVs.

El equipo de QA ejecutó pruebas funcionales manuales y algunas automatizadas sobre los módulos de "Registro", "Carga de CV" y "Recomendaciones".

Información del Sprint / Iteración

- **Nombre del proyecto:** JobMatch
- **Sprint / Iteración N°:** Sprint 5
- **Fecha de ejecución:** 1 al 5 de abril de 2025
- **Tester responsable:** Ana Martínez

Indicadores Calculados

- Casos planificados: 40
- Casos ejecutados: 35
- % de ejecución: 87.5%
- Defectos encontrados: 12
 - Críticos: 3
 - Medios: 5
 - Leves: 4
- Tasa de defectos: 34.28%
- Tasa de re-apertura: $2/12 = 16.66\%$
- Tiempo invertido: 24 horas
- Horas presupuestadas: 20 horas
- Costo por defecto: $(24h \times \$50) / 12 = \100

Comparativa con Estándares

Indicador	Resultado	Estándar industria	de	la	¿Cumple?
% Ejecución	87.5%	> 90%			✗
Tasa de defectos	34.28%	< 20%			✗
Tiempo por caso	0.685 h	0.5 h			✗
Costo por defecto	\$100	≤ \$80			✗

Análisis e Interpretación

- Se ejecutó la mayoría de los casos, pero no se llegó al 90% esperado. Esto podría deberse a desvíos en la priorización o falta de tiempo.
- La tasa de defectos fue alta, con un 25% de errores críticos, lo que sugiere que la funcionalidad aún no está madura.
- El tiempo por caso fue superior al estándar, lo que indica pruebas manuales lentas o test cases poco optimizados.
- El costo por defecto está por encima del ideal, lo que afecta la eficiencia del equipo.

Recomendaciones

- Automatizar al menos los casos repetitivos en el módulo de “Registro”.
- Priorizar revisión de requerimientos con el Product Owner para reducir bugs críticos.
- Reestructurar la planificación de pruebas para llegar al 90% de ejecución en próximos sprints.
- Mejorar la documentación de test cases para acortar tiempos de ejecución.

Material Complementario

- [¿Qué es un KPI \(en inglés\)? – Atlassian Glossary](#)
- [Postman + Newman – Generar reportes de ejecución](#)
- [Guía práctica de métricas en QA](#)

Próximos Pasos

En la **próxima clase**, nos adentraremos en uno de los campos más relevantes hoy en QA: el testing en dispositivos móviles.

Lo que veremos en la Clase 13:

- ¿Qué es Mobile Testing? y ¿por qué es tan importante en la actualidad?
- Las diferencias entre testear en escritorio, web y mobile.
- ¿Cómo adaptar pruebas y criterios de validación a diferentes tipos de pantallas, resoluciones y sistemas operativos (Android, iOS)?
- Introducción a herramientas específicas como **Appium**, **BrowserStack**, y el uso del inspector del navegador para simulaciones.
- Caso práctico con Talento Lab: acompañar el despliegue de una nueva app mobile de la plataforma.

Como verás, lo que aprendimos sobre KPIs, organización de pruebas y seguimiento en Jira también será aplicable en contextos móviles. Además, comenzaremos a pensar cómo automatizar algunas validaciones básicas en entornos diversos y cómo mantener la calidad del producto en múltiples dispositivos.

¡Otro día en Talento Lab!



Silvia y Matías te entregan una planilla con los resultados del sprint. Tu tarea ahora es **analizar métricas**, generar un reporte y compararlo con valores estándar.

"Necesitamos saber si el plan de pruebas está funcionando y dónde debemos mejorar", te dice Matías.



Te piden que calcules métricas clave, interpretes los datos y prepares recomendaciones para el equipo.

Ejercicios Prácticos

Ejercicio 1: Métricas propias

Objetivo: Incorporar métricas.

Instrucciones:

1. Escenario:
 - Casos planificados: 40
 - Casos ejecutados: 35
 - Defectos encontrados: 12 (3 críticos, 5 medios, 4 leves)
 - Tiempo invertido: 24h
 - Horas presupuestadas: 20h
2. Analiza qué indicadores podrías usar para evaluar el plan.
3. Calcula:
 - Porcentaje de ejecución
 - Tasa de defectos
 - Distribución por severidad
 - Costo por defecto (asumir \$50/h)
 - Desviación del tiempo estimado
4. Reporta los indicadores con interpretación.

Ejercicio 2: Comparación con estándares de la industria

Objetivo: Incorporar comparativa con los estándares de la industria.

Instrucciones:

1. Usa los indicadores calculados.
2. Compáralos con estos estándares:
Tasa de defectos: $< 20\%$
 - Tiempo por caso: 0.5h
 - Ejecución esperada: $> 90\%$
 - Costo por defecto: $\leq \$80$
3. Analiza diferencias entre tus métricas y los estándares:
 - Calidad
 - Velocidad
 - Costos
4. Reflexiona y propone acciones de mejora.



Buenos Aires
aprende

Agencia de Habilidades para el Futuro

BA Buenos
Aires
Ciudad