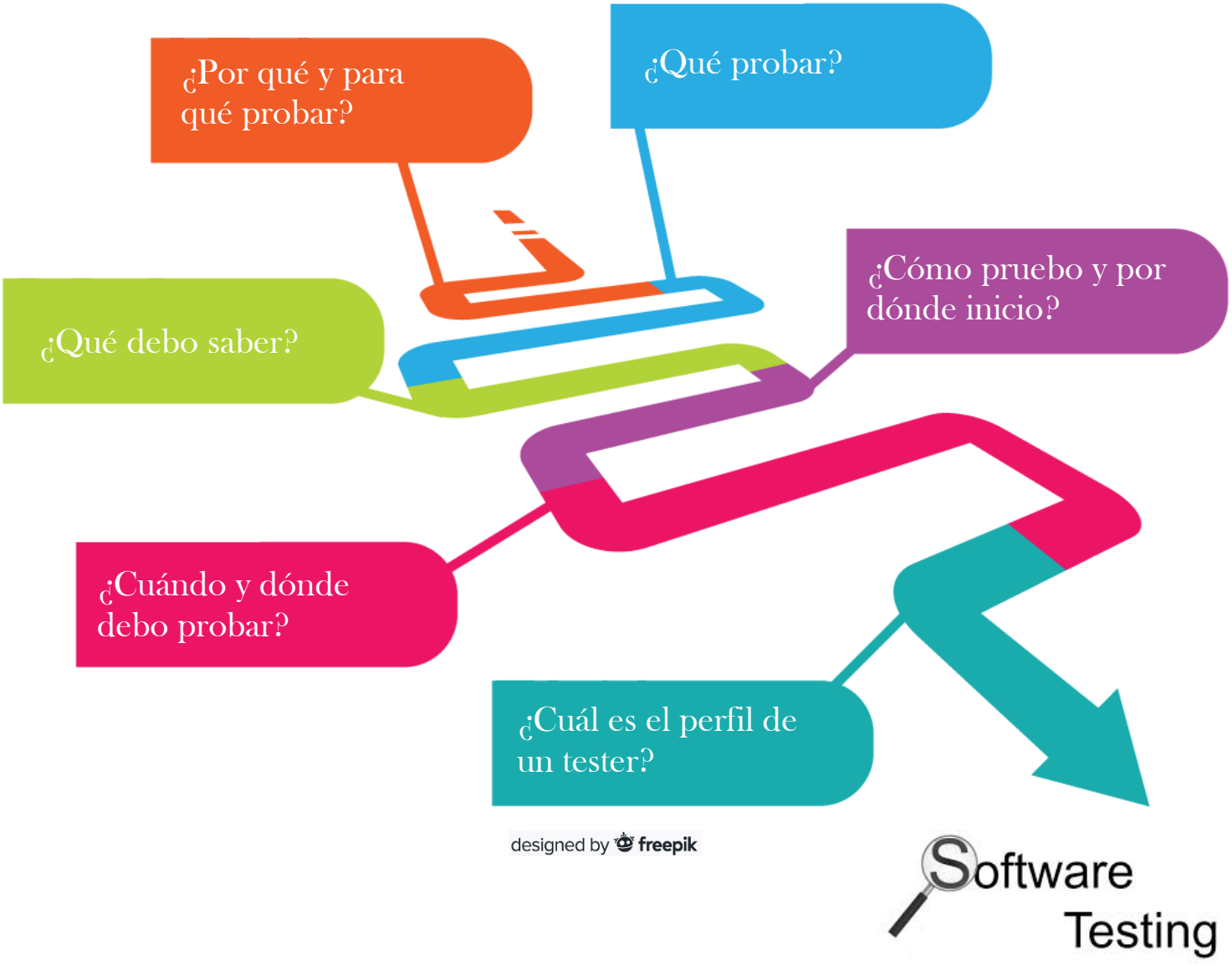


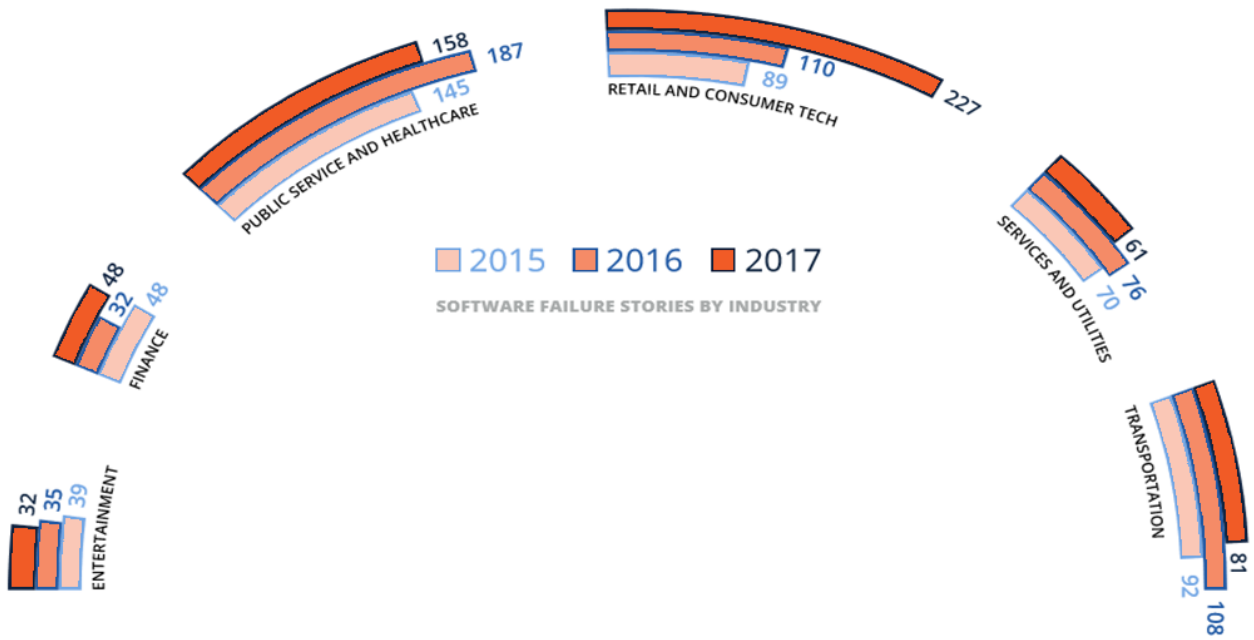
L.I. Liliana Rangel Cano
L.I. Cristhian Eder Alavez Barrita

¿Por qué y para qué probar?

En la actualidad el software se encuentra presente en un gran número de actividades de la vida cotidiana, tenemos por ejemplo: sistemas de información, redes de colaboración, repositorios, aplicaciones, sistemas bancarios, comercio electrónico, entre otros.

Debido a que el software es susceptible a tener defectos[1] por ser desarrollado e implementado por seres humanos, quienes podemos cometer errores[2], surge la necesidad de realizar pruebas, con las que se busca reducir al máximo su presencia, es importante decir que, también al incrementar la funcionalidad, incorporar modificaciones o hacer mantenimiento, es posible que se generen defectos. Si estos errores no son identificados oportunamente, previo a la puesta en producción, existe el riesgo de generar resultados inadecuados, un sistema que no funciona correctamente puede dar lugar a muchos problemas, incluyendo afectación de la imagen de la institución, pérdida de credibilidad, re-trabajo, insatisfacción de los usuarios, pérdida de dinero o daños personales.

De acuerdo con la 5a edición de *Software Fail Watch*, en 2017 se registraron 606 fallas en el software, en 314 empresas, afectando a 3,700 millones de personas, $1.7 billones en activos:



Las pruebas de software, las cuales son "una actividad realizada para evaluar y mejorar la calidad del producto, identificando defectos y problemas" [SWEBOK], permiten disminuir el riesgo de que se presente algún tipo de fallo en la funcionalidad y el desempeño, detectan vulnerabilidades que pudieran ser aprovechadas para realizar acciones tales como ataques, intrusiones o cualquier otro uso indebido, adicionalmente, pueden mejorar la calidad del software, debido a que contribuyen proactivamente a encontrar defectos y a su vez en la toma de decisiones, previo a la puesta en producción.

Las pruebas forman una parte importante en la realización del software, debido a la necesidad de proporcionar a los usuarios, en el caso de la UNAM, a la comunidad universitaria y público en general, productos de calidad y mantenerlos satisfechos.

1. Un ***defecto*** es la manifestación de un error en el software encontrado porque causa una falla, la cual es una desviación del servicio o resultado esperado.
2. El ***error*** es la acción humana que produce o genera un resultado incorrecto.

¿Qué probar?

La respuesta más cómoda a este cuestionamiento sería, "prueba todo"; sin embargo, *es imposible realizar pruebas exhaustivas*, debido a que no es factible en tiempo, costo y esfuerzo, por lo que se sugiere analizar y priorizar los riesgos de producto[3] y enfocarse en ellos.

De acuerdo con los riesgos asociados al software que se desean reducir, se presentan las siguientes sugerencias que pueden considerarse en las pruebas a aplicar:

Si bien, se presentan las pruebas más comunes y utilizadas en el sector, se deben aplicar las que minimicen los riesgos de mayor impacto en el software.

Del mismo modo, para establecer el alcance de las pruebas es necesario gestionar y priorizar los riesgos de producto donde se identifique el nivel de detalle que requiere la prueba, por ejemplo:

- En las pruebas funcionales, el alcance podría acotarse a verificar que los movimientos se realicen de manera adecuada; o bien, que los datos sean validados con la finalidad de tener calidad en la información; o cuidar hasta el último detalle como el formato de los elementos, la redacción en los mensajes presentados al usuario, entre otros.
- En las pruebas de desempeño, el alcance puede acotarse a conocer los resultados para posteriormente tomar las decisiones pertinentes, o bien, identificar la configuración óptima que permita los mejores resultados.

En caso de requerir la aplicación de más de un tipo de prueba, se sugiere establecer su objetivo, delimitar su alcance, y definir los criterios de aceptación[4], para poder disminuir el tiempo de aplicación, aumentar la variedad de pruebas y lograr la calidad definida.

3. Los ***riesgos de producto*** son eventos que pueden ocurrir posterior al término del proyecto, después de las pruebas de aceptación, amenazan la calidad del producto y afectan a los clientes y/o usuarios.

4. Microsoft Press define a los ***criterios de aceptación*** como "las condiciones que un producto de software debe satisfacer para ser aceptado por un usuario, cliente o stakeholder". Para Google, son "estándares pre-establecidos o requerimiento que un producto o proyecto debe satisfacer".

¿Qué debo saber?

- La calidad del software no está dada en razón de las pruebas aplicadas.

- Las pruebas son el proceso de ejecutar un programa con la intención de encontrar defectos, entre más defectos de mayor impacto se encuentren más valor tiene la actividad.

- Las pruebas de software muestran la presencia de defectos, no su ausencia, por lo que la aplicación de las pruebas no son una evaluación de la calidad que genere un dictamen.

- Las pruebas verifican el cumplimiento de los requerimientos definidos, sin embargo no validan que estos sean la solución a una necesidad.

- Dependiendo del modelo de ciclo de vida de desarrollo de software elegido, del nivel y tipo de pruebas a aplicar, se deberá seleccionar la estrategia y técnicas a utilizar.

- Las pruebas de software no son una actividad improvisada, existen metodologías, estándares, técnicas, estrategias, herramientas, buenas prácticas a seleccionar y diversas referencias que se pueden consultar para su aplicación.

- No hay requerimiento sencillo o con poco valor que no merezca la pena de ser probado, en muchos casos los requerimientos que parecieran inofensivos son los que requieren mayor atención o en donde se engloba el mayor número de defectos, reflejan carencias que afectan en los controles, usabilidad, o tienen una implicación tecnológica.

- Los defectos no sólo se encuentran en el software, también pueden estar en la definición de los requisitos y/o en el diseño.

- Para que los resultados de las pruebas sean objetivos se requiere un nivel de independencia, esto es, que no existan conflictos de intereses, por lo que se sugiere que el personal responsable y a cargo del desarrollo del software no sea el mismo que quien realice las pruebas.

- La principal habilidad de un probador se encuentra en la creatividad para comportarse como lo haría un usuario final que tiene dudas y realiza movimientos imprevistos al utilizar el software.

- Los casos de prueba deben ir cambiando en cada ciclo de pruebas, de lo contrario no se encontrarán nuevos defectos.

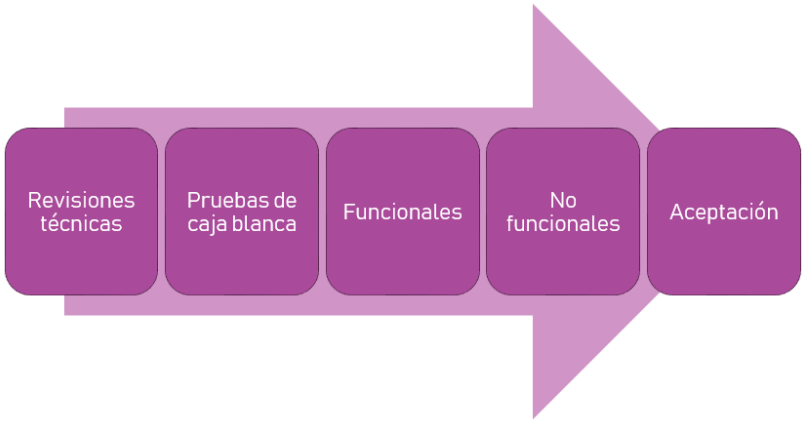
- La principal debilidad de un probador es considerar como “obvia” o ”lógica” la funcionalidad que debe tener el software, por lo cual puede ignorar ambigüedades e inconsistencias en la normatividad o los requerimientos definidos, y no registrar o clarificarlas, omitiendo los defectos en la documentación técnica.

¿Por dónde inicio?

En un proyecto de desarrollo de software, los defectos (bugs) pueden presentarse en cualquier etapa del ciclo de vida, así como, encontrarse en algún elemento de calidad del software como puede ser: en la funcionalidad, el desempeño, la usabilidad, entre otros. Idealmente se deberían realizar pruebas en cada una de las fases del ciclo de vida de desarrollo del software, aplicando diversos tipos de pruebas, sin embargo, esta actividad no terminaría.

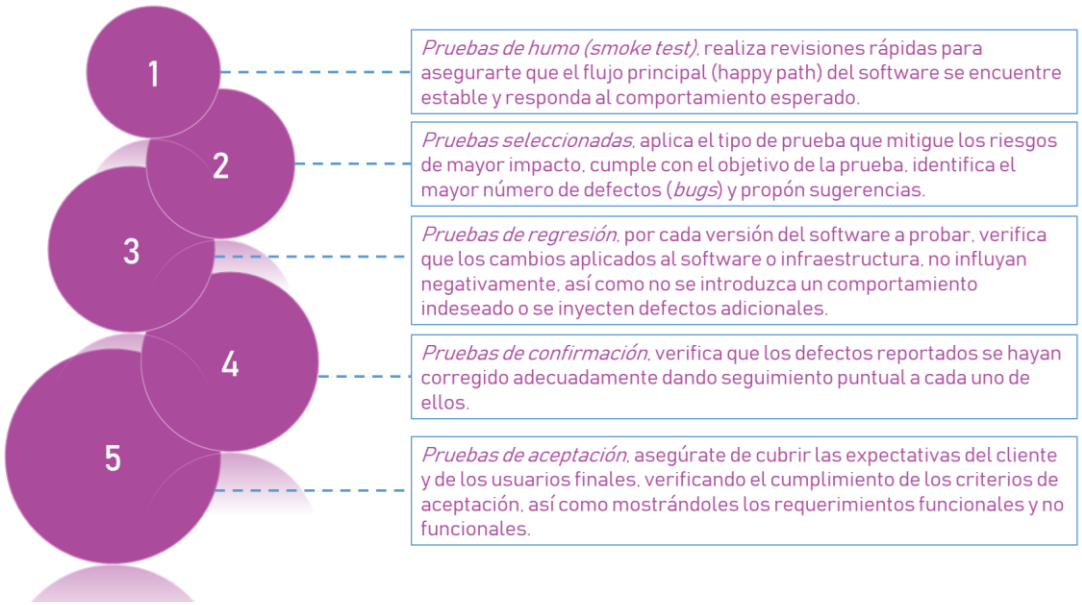
Las pruebas del software adquieren mayor valor e importancia al aplicarse en fases tempranas, debido a que entre más desarrollado se encuentre el software, mayor será el costo de corregir los defectos.

En seguida se presenta el orden sugerido para aplicar las pruebas:



Independientemente del tipo de prueba que se requiera realizar, se sugiere hacer revisiones técnicas para comprender el funcionamiento del sistema, así como aplicar pruebas funcionales para asegurar que el flujo principal del software puede llevarse a cabo sin problemas.

De acuerdo con las pruebas a aplicar, se recomienda realizar diferentes ciclos de pruebas[1] donde se consideren las siguientes:



5. Se considera como un ***ciclo de pruebas*** a la aplicación de un tipo de prueba (como prueba funcional, confirmación, regresión, entre otras), a una versión de software, con un alcance establecido (uno o más módulos) en un periodo de tiempo continuo.

¿Cuándo y dónde debo probar?

El momento de intervención del equipo de pruebas en el proyecto depende del modelo de ciclo de vida del software utilizado para el desarrollo, sin embargo,

con la finalidad de identificar oportunamente los defectos y disminuir el costo de su corrección, se sugiere iniciar las pruebas durante la fase de análisis, realizando revisiones técnicas, y participar en cada una de las fases, revisando y retroalimentando los productos de trabajo.

Respecto al cuestionamiento de ¿en dónde probar?, es recomendable que el equipo de pruebas tenga su propio ambiente tecnológico análogo al ambiente de producción, para que los probadores tengan el control completo de los datos y no se limiten a realizar pruebas con temor a interrumpir el trabajo de los desarrolladores, así como, reporten defectos causados por un cambio en el software, por datos insertados directamente a la base de datos y no cumplan con las especificaciones, entre otras cuestiones.

Se sugiere que el ambiente tenga una versión estable del software, esto significa que no debe cambiarse el software o realizar ajustes a configuraciones durante la aplicación del ciclo de pruebas; y se encuentre vigente dicha versión, para asegurar que los defectos identificados sean oportunos y no se hayan reportado previamente.

Para que los resultados de las pruebas de desempeño sean confiables, es importante aplicar las pruebas en el ambiente de producción y en la última versión del software.

Para la aplicación de las pruebas de desempeño y de seguridad es recomendable que se realicen en el ambiente de producción, previo a su liberación, sin embargo, en ocasiones las pruebas deben llevarse a cabo una vez que el software se encuentra en funcionamiento, por lo cual se aconseja tener un ambiente alterno que posea las mismas características del ambiente de producción y los mismos parámetros de configuración.

¿Cuál es el perfil de un tester?

Debido a las actividades que realiza el probador, las habilidades sugeridas que debe tener son:

Comunicación asertiva, puesto que la principal función de un probador es identificar y reportar defectos, es indispensable para este rol expresarse de manera consciente, congruente, directa y equilibrada, siendo claro, preciso y tener el tacto suficiente para informar tanto los defectos como el estado general del software.

Analítico, el probador debe comprender el proceso a automatizar mediante el software y las reglas de negocio para poder verificar el cumplimiento de los requerimientos y que el software contemple los controles necesarios y realice de forma transparente los flujos de información de acuerdo con la necesidad a cubrir.

Inquisitivo, con el objetivo de identificar los defectos en el flujo de información del software, el probador debe investigar a detalle los resultados esperados para las diversas entradas, proceso, interacción entre secciones, acciones, estatus.

Crítico, para evaluar el software con un enfoque de especialista en procesos y de usuario final, para hacer deducciones y vincular lo observado con los criterios de calidad.

Objetivo, todo probador debe ser imparcial en el estado en que se encuentra el software, sin maximizar o minimizar los defectos identificados, reconociendo por cada uno de ellos su impacto y el efecto que pudiera tener al no ser atendido. Adicionalmente el probador no debe tomar partido entre el equipo de trabajo y/o el cliente, sino basarse en la normatividad, documentación técnica del proyecto y buenas prácticas del sector.

Perceptivo, en las necesidades de los clientes, usuarios, del equipo de trabajo líder, analistas, diseñadores, desarrolladores, para enfocar las actividades de pruebas en ello.

Creativo y curioso, para generar casos y escenarios de prueba e imaginar problemas que podrían existir, descubrir escenarios y flujos que no se le ocurren a quién lo implementó.

Negociador al dar seguimiento a los defectos reportados, sobre todo aquellos en los que, el responsable de realizar las correcciones, indica que no son defectos o están fuera del alcance del proyecto.

Adicionalmente a estas habilidades, el probador debe tener conocimientos técnicos en ingeniería de software y programación, para la automatización de pruebas funcionales, y aplicación de otro tipo de pruebas como de desempeño, seguridad, entre otras.