CHOUCAIR®
Business Centric Testing

La automatización es una estrategia que complementa el proceso de ejecución de las pruebas y para su implementación es necesario definir y disponer de un modelo que lo soporte.

Por lo anterior, antes de emprender el reto de automatizar sugerimos realizarse las siguientes preguntas que nos permitan reflexionar y escoger el modelo más adecuado.

- ¿ Que es la automatización de pruebas?
- ¿Cuales son los Beneficios de Automatizar?
- Desventajas de la Automatización de Pruebas
- Proceso de Automatización
- ¿Cómo identificar que automatizar?
- ¿Cuales son los Riesgos Asociados a la Automatización ?
- ¿Qué es la Viabilidad Técnica?
- ¿Conoces las Limitaciones de la Automatización?
- ¿Cuales son los Problemas de la Automatización?
- ¿Que es el Contexto y racionalidad?
- Programación Requerida
- ¿Cómo Alcanzar el éxito en la Automatización?
- ¿Tiene mi aplicativo la Capacidad de Ser Automatizado?
- Herramientas
- ¿Qué es el Retorno de la Inversión (ROI)?
- ¿Cuales son los Beneficios de la Automatización en Pruebas Especializadas?
- DEMO

CHOUCAIR
Business Centric Testing

Que es la Automatización de Pruebas?

- La automatización de la prueba es una ejecución automática de pruebas funcionales apoyados en un conjunto de herramientas.
- La automatización busca simular a un humano en el proceso de ejecución manual de la pruebas.
- Es claro que la ejecución se lleva a cabo a traves de estas herramientas pero el proceso de análisis, el diseño e implementación continuaran realizandose de forma manuales.
- La automatización en si necesita otros aspectos para que se desarrolle de manera exitosa:
  - Un conjunto de pruebas que se ejecutadas a traves de distintas herramientas que se pueden combinar
  - Librerias de codigo que se incorporan a nuestra automatización
  - Conjunto de casos de pruebas definidos por el analista
  - Evaluación de los resultados objenidos de la ejecución

"La automatización de pruebas busca incrementar la cobertura de las pruebas, No eliminar a los Analistas de pruebas"

"La automatización de pruebas es una táctica de ejecución (Estrategia), No es un tipo de prueba."

CHOUCAIR®

Business Centric Testing

Para ***Choucair***, la Automatización de Pruebas se resume en, **generar eficiencia en los procesos relacionados a pruebas de software**, mediante el uso de *herramientas*, *modelos* y *estrategias* que den ***mayor velocidad a la ejecución***, la misma que se basa en métodos formales que dan soporte a los criterios de calidad de construcción de los scripts y preservando los conceptos de reutilización al construir netamente lo necesario para suplir la necesidad.

Beneficios de Automatizar

CHOUCAIR®
Business Centric Testing

- *Incrementar la velocidad de la ejecución <Eficiencia>*
- *Aumentar la cobertura de las pruebas*
- *Disminuir el riesgo de fallas y faltas en producción*
- *Disminuir el esfuerzo durante el proceso de ejecución (Tiempo)*
- *Apoyar en la generación y/o consecución de datos de prueba*
- *Apalancar el cumplimiento del Date To Market y Time To Market de los proyectos*
- *Obtener feedback temprano del desarrollo*
- *Aumentar la capacidad y frecuencia de ejecución (Disponibilidad 7x24)*
- *Disponer de ejecuciones de procesos regresivos de forma desatendida e impulsar las pruebas tempranas*
- *Apoyar procesos de IC, TC, EC, DC y DEVOPS*
- *Mejorar la eficiencia de las pruebas reduciendo el costo de cada realización de una prueba*

Beneficios Indirectos

CHOUCAIR®
Business Centric Testing

- ✓ **Enfoque:** Los analistas de prueba o personas que estaban a cargo de las tareas que fueron automatizadas, tienen la posibilidad de mejorar el enfoque en tareas que exploten mejor sus capacidades.

- ✓ **Capacidades:** Mejora las capacidades analíticas de las personas al revisar los resultados y los comportamientos del proceso automatizado en base a sus a los datos obtenidos en las ejecuciones.

- ✓ **Salud:** Reduce los niveles de estrés, cansancio y frustración por ejecución de tareas repetitivas.

- ✓ **Clima laboral:** Gestionado de la mejor manera, apoya a tener un clima laboral mejor y disminuye la posibilidad de conflictos provocados por estrés, cansancio y/o frustración.

- ✓ **Sobre esfuerzo:** Reduce y en la mayoría de los casos elimina el hecho de recurrir a realizar horas adicionales que generan desgaste y por ende que se afecte el mejor desempeño de las personas.

En conclusión,

las personas responsables de los procesos que han sido automatizados, tienden a tener un mejor desempeño en actividades más estratégicas y disponen de una mejor calidad de tiempo para poder usarlos de la mejor forma posible.

Desventajas de la Automatización de la Prueba

CHOUCAIR
Business Centric Testing

- Se generan un conjunto de costos adicionales (incluidos altos costos de inicio)
- Las nuevas tecnologías deben ser dominadas por el grupo de prueba, el cuál puede no tener la experiencia necesaria
- En el peor de los casos, la complejidad puede ser abrumadora
- Crecimiento inaceptable donde la automatización se vuelve más grande que el aplicativo al cual se le esta realizando la misma.
- Habilidades de clase de desarrollo necesarias para el equipo de prueba o para brindarle un servicio al equipo de prueba
- Se requiere un mantenimiento considerable de las herramientas, entornos y recursos de prueba
- La deuda técnica puede crecer muy fácilmente y su disminución se vuelve compleja de reducir.
- La automatización se ha convertido en un proyecto de desarrollo mas y por tal motivo en necesario poner en práctica los procesos y disciplinas del desarrollo de software.

Desventajas de la Automatización de la Prueba

CHOUCAIR
Business Centric Testing

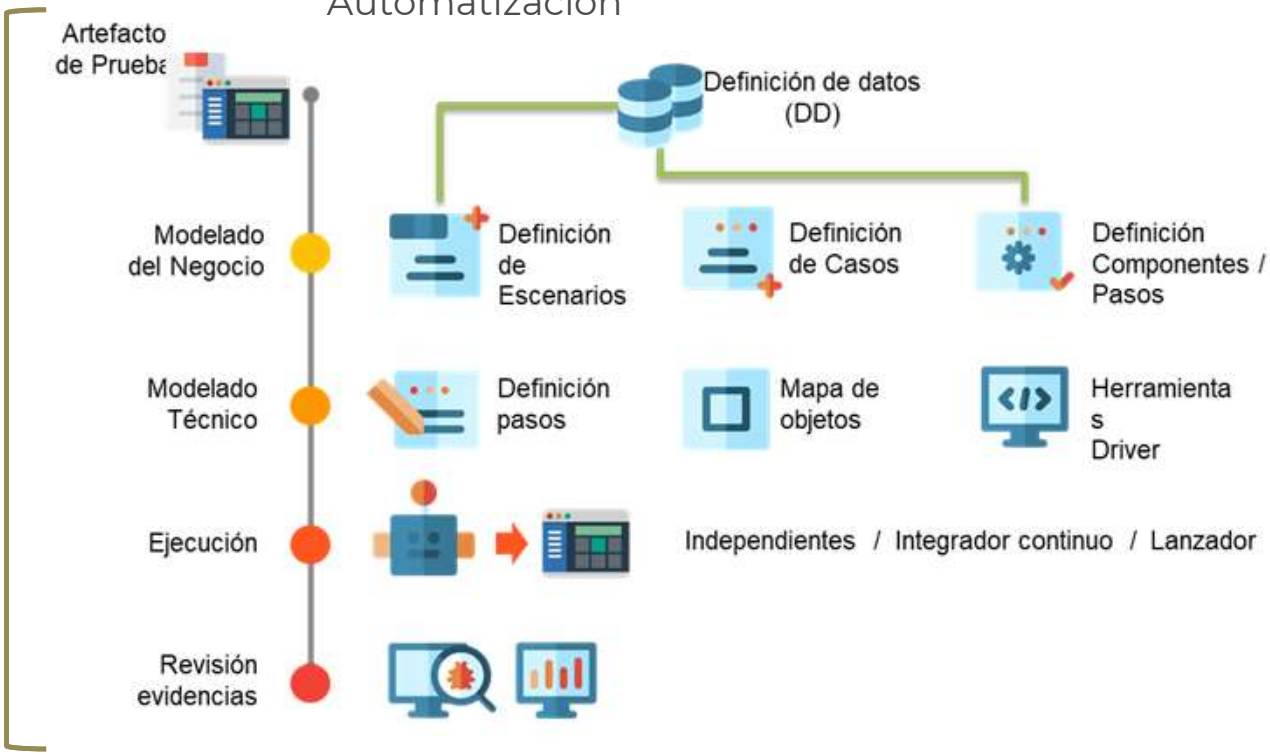
- Concentrarse en la automatización puede hacer que los probadores se olviden de la gestión de riesgos para el proyecto
- La paradoja del pesticida aumenta cuando se usa la automatización, ya que exactamente las mismas pruebas se ejecutan cada vez
- Las fallas de automatización a menudo no son fallas de la aplicación que se esta probando; están en la propia automatización (falsos positivos)
- Sin una programación inteligente en las pruebas, las herramientas son de mentalidad literal y estúpidas; los probadores no lo son.
- Las herramientas tienden a ser de un subproceso único, es decir, solo buscan un resultado para un evento cuando en realidad hay muchos resultados posibles que pueden ocurrir

Proceso de Automatización

CHOUCAIR
Business Centric Testing

Antes de iniciar un proceso continuo de automatización de un set de casos de prueba de un sistema, se hace necesario adelantar unas tareas previas que nos ayuden a determinar la conveniencia de dar inicio al proceso, como son, la viabilidad Funcional, Técnica y determinación del costo / beneficio de la inversión a realizar.

Proceso continuo de Automatización



¿Cómo identificar que Automatizar?

*Es importante determinar la Viabilidad Funcional, teniendo en cuenta el **costo beneficio** que se obtiene al automatizar un proceso.*

NO se tiene que pensar, que, por solo tener la capacidad de Automatizar algo necesariamente sea conveniente hacerlo.

A continuación, se listan algunos de los criterios para tener en cuenta al momento de elegir que automatizar en el proceso de pruebas:

- *Procesos críticos que impacten al negocio.*
- *Procesos que toman un tiempo considerable medio o alto en completarse y que tienen que ser ejecutados por usuarios (Negocio, analistas de prueba, otros).*
- *Procesos de prueba que ejecutan el mismo flujo de trabajo en diferentes configuraciones (Ejemplo, ambientes).*
- *Pruebas orientadas por datos, flujos ejecutados con diferentes datos cada vez.*
- *Procesos que implican el ingreso de grandes volúmenes de datos, ejemplos formularios de registro.*
- *Pruebas que en el proceso de ejecución requiere verificar múltiple información.*
- *Tareas repetitivas que intervienen en el proceso de pruebas.*

¿Cómo identificar que Automatizar?

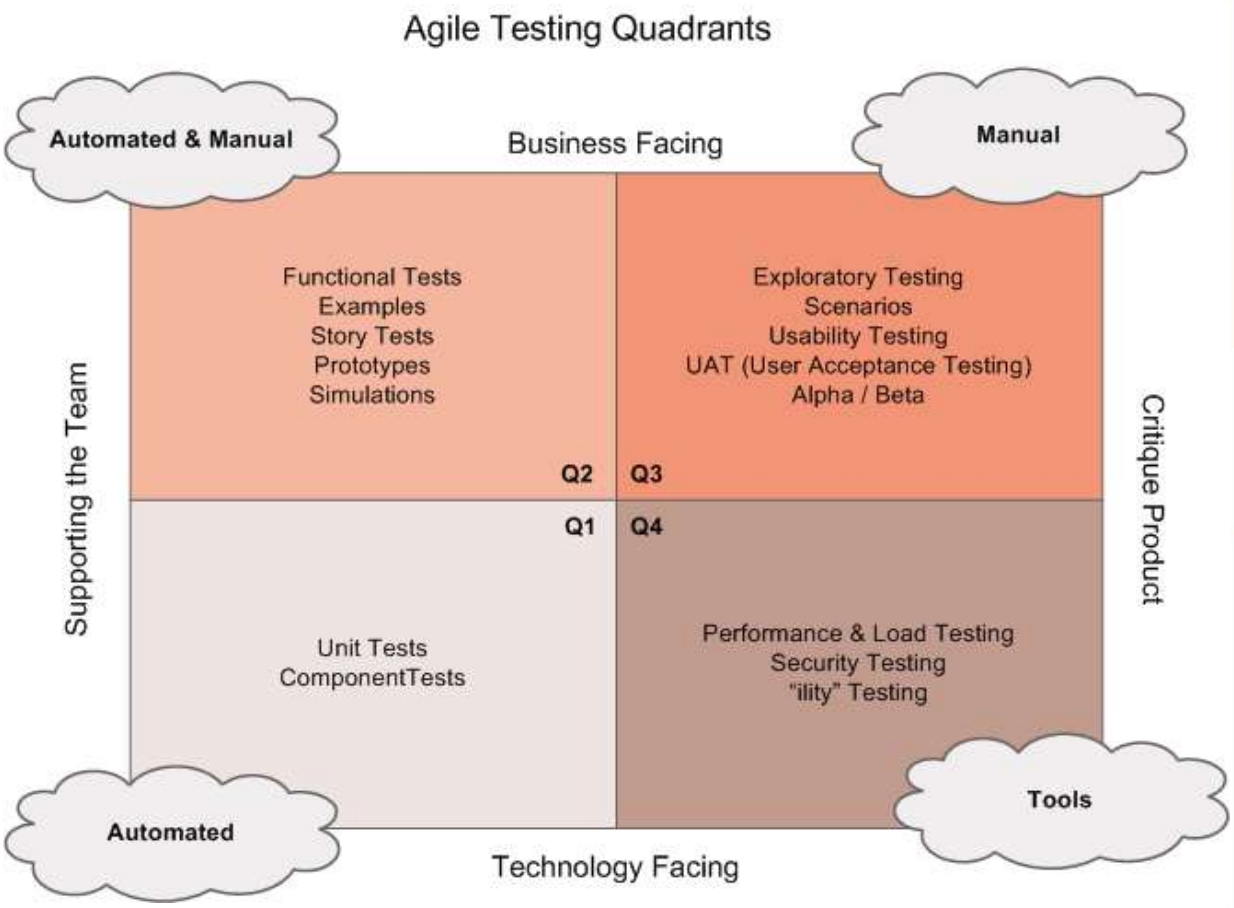
CHOUCAIR
Business Centric Testing
Viabilidad Funcional | Viabilidad Técnica | Determinar ROI

Enfoques de Automatización

Los cuadrantes **representan los diferentes propósitos y tipos de pruebas** de software que podemos realizar en un entorno Ágil.

Sin embargo, también nos ayuda a identificar el momento adecuado para pensar en la aplicación de estrategias de automatización.

Como vemos, esto sucede en los cuadrantes 1 y 2, Estas pruebas primeramente guían el desarrollo de la funcionalidad, y luego cuando se automatizan sirven para apoyar la refactorización y la inclusión de nuevo código sin causar resultados inesperados en el comportamiento del sistema.



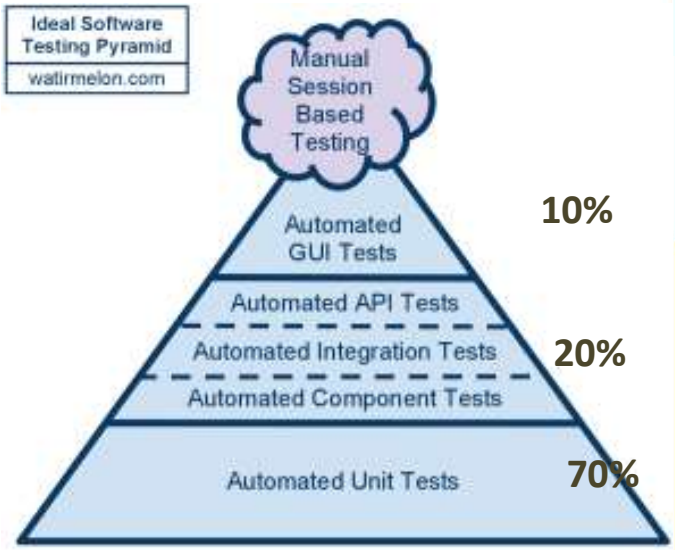
¿Cómo identificar que Automatizar?

CHOUCAIR®
Business Centric Testing
Viabilidad Funcional | Viabilidad Técnica | Determinar ROI

Niveles de Automatización

La pirámide de pruebas de ***Mike Cohn***, descrita en su libro *Succeeding with Agile*, ha sido un referente en este campo durante mucho tiempo. En ella ***Cohn*** establece que hay varios niveles de pruebas, y señala el grado en el que deberíamos automatizarlas. Lo ideal sería:

- **Muchos tests unitarios automáticos**, porque un primer punto primordial para detectar fallos es a nivel de desarrollador. Si una funcionalidad en este punto falla, podrían fallar pruebas de los siguientes niveles: integración, API etc.
- **Bastantes tests a nivel de API, integración de componentes, servicios**, que son los más estables y candidatos a automatizar.
- **Menos tests de interfaz gráfica automatizados**. Ya que estos tests son variables, lentos en su ejecución y con muchas dependencias con otros componentes.
- **Un nivel estable de pruebas automáticas**, que vayan detectando los testers manuales y se vayan automatizando paulatinamente, para que llegue un momento en el que invirtiendo los mismos recursos logremos cada vez una cobertura mayor de pruebas.



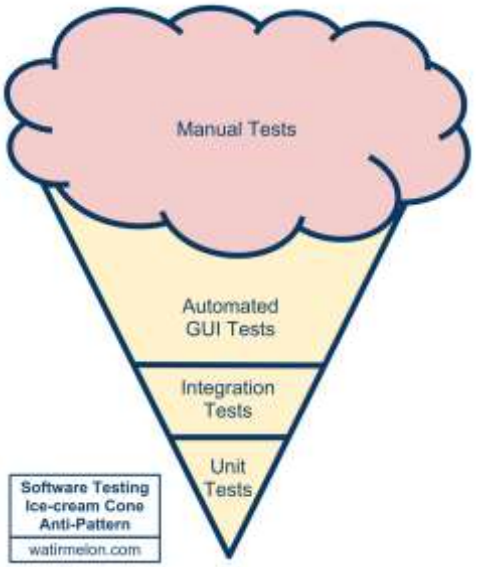
¿Cómo identificar que Automatizar?

CHOUCAIR®
Business Centric Testing

Viabilidad Funcional | Viabilidad Técnica | Determinar ROI

En ciertas ocasiones se tiende a perder el foco y a invertir en automatización en el nivel y grado no adecuado. Una mala estrategia en estos casos, es lo que llamamos el anti patrón "**Ice Cream Cone (Cono de helado)**".

Como ves es lo contrario a la pirámide de Cohn: centramos el foco en muchas pruebas manuales y en automatizar pruebas de interfaz de usuario, y nada de pruebas unitarias. Con todos los problemas que acarrea no detectar errores en los otros niveles y dejarlo todo para la parte más visible de la aplicación.



¿Cómo identificar que Automatizar?

CHOUCAIR
Business Centric Testing

Con lo visto hasta el momento, podríamos decir:

¿Conviene automatizar?

- ✓ Pruebas Unitarias y de componente
- ✓ Pruebas a nivel de API (Servicios, Microservicios)
- ✓ Procesos de prueba altamente repetitivos
- ✓ Smoke test – pruebas de recepción
- ✓ Set de pruebas orientados a regresiones parciales o totales
- ✓ Flujos end to end con orientación al negocio
- ✓ Procesos de configuración necesarios.
- ✓ Procesos de Generación, Creación y/o Extracción de datos.
- ✓ Procesos de Migración de datos.
- ✓ Tareas de apoyo repetitivos.
- ✓ Tareas de Integración, Entrega y Despliegue Continuo.

¿No conviene tanto?

- ✗ Pruebas exploratorias, pruebas adhoc, pruebas aleatorias o de dominio específico.
- ✗ Procesos que se ejecutan una única vez, esto varia en el caso que se repita para cada proyecto, por ejemplo.
- ✗ Procesos sin resultados predecibles, por tener un alto grado de análisis y decisión sobre los resultados obtenidos.
- ✗ Procesos que por su complejidad demandan una inversión de tiempo y dinero irrecuperable al automatizarlo.
- ✗ Sistemas inestables.
- ✗ Sistemas con proyección de ser modificados en un alto porcentaje o que constantemente sufren modificaciones de alto impacto.

¿Cómo identificar que Automatizar?

CHOUCAIR
Business Centric Testing

Consideraciones

- Automatización es más que capture and replay
- No se trata de escribir un programa para probar otro programa
- Automatización es más que ejecución de pruebas
- Automatización de pruebas es estratégico
- No todo puede o debe ser automatizado
- Reinvertir el tiempo ahorrado
- Necesidad de apoyarse en framework´s
- La automatización requiere mantenimiento continuo
- Requiere de una infraestructura que la soporte

Riesgos Asociados

CHOUCAIR®
Business Centric Testing

La implementación de cualquier solución de pruebas automatizadas no es sencilla, incluso teniendo la mejor herramienta a disposición. La automatización puede fallar posiblemente debido a los siguientes riesgos asociados a la misma:

1 **Falta de lineamientos de desarrollo**
La programación de aplicaciones sin buenas prácticas que faciliten la automatización de las pruebas, ejemplo, que los elementos de la pantalla contengan un ID que los identifique o seguir los estándares de la wc3.

2 **Falta de conocimientos en codificación**
Una automatización eficiente no es tan simple. Los guiones de prueba generados automáticamente por la herramienta durante la grabación deben ser modificados manualmente, lo que requiere el conocimiento de secuencias de comandos, a fin de hacer las secuencias de comandos robustas, reutilizables, y fácil de mantener. Para poder modificar los scripts, el Tester debe ser entrenado en la herramienta y en el lenguaje de scripting.

3 **Datos de prueba**
Al igual que para el testing manual, los datos de prueba son un pilar importante en las pruebas automatizadas. Hay que tener en cuenta que una buena administración de datos de prueba es la base de la automatización con éxito, ya que al no hacer un seguimiento de los datos con los que corremos las pruebas o los datos que estas generan puede provocar un gran numero de reportes de errores los cuales se produzcan por datos corruptos en la base de datos y no por la aplicación bajo pruebas.

4 **Inestabilidad en el Ambiente de Pruebas**
Así como la inestabilidad o no disponibilidad de ambiente adecuado para la ejecución de las pruebas afecta a las ejecuciones manuales, mucho más afecta al tratarse de procesos automáticos, lo anterior acompañado de un manejo inadecuado de mecanismos de autorrecuperación y control de excepciones, puede ocasionar reproceso y perdidas en la oportunidad y eficacia de los test.

Viabilidad Técnica

CHOUCAIR®
Business Centric Testing

Viabilidad Funcional — Viabilidad Técnica — Determinar ROI

Así como es importante determinar la Viabilidad Funcional, es decir determinar el alcance con base en la importancia, beneficios, criticidad y costo de cada una de las pruebas y procesos seleccionados. También lo es determinar la Viabilidad Técnica del proceso.

Viabilidad Técnica, consiste en el proceso de evaluación del nivel de adherencia o de interacción de la aplicación o sistema bajo prueba a la herramienta o framework seleccionado.

Dicho análisis nos va permitir establecer el nivel de complejidad y esfuerzo en la construcción de los script de prueba, elementos clave para determinar el costo beneficio del proceso.

Viabilidad Técnica

CHOUCAIR®
Business Centric Testing

Viabilidad Funcional | Viabilidad Técnica | Determinar ROI

- Seleccione una muestra de elementos que se utilizaran en la prueba, escoja objetos no convencionales ejemplo: calendarios, menús, objetos dinámicos, ventanas emergentes, mensajes de validación, etc.
- Un segundo criterio, es identificar si la aplicación utiliza mecanismos de seguridad necesarios para completar transacciones, ejemplo: captcha, tokens virtuales o SMS, biometría, etc. (consulte si dichos métodos pueden ser simulados)
- Otro criterio es el nivel de uso de los elementos y su importancia dentro de la aplicación, esto nos ayudará a determinar el impacto en la automatización al no poder interactuar con el.

- Elabore un inventario de objetos, catalogados por el nivel de respuesta de la herramienta seleccionada (Alta, Media, Baja o No tiene adherencia)
- Obtenga el total de objetos de prueba y de cada categorización.
- Determine el peso (%) de cada categoría, y realice el calculo:

  total de la categoría / total de objetos x Peso(%)
- Por último calcule la adherencia total, sumando los porcentajes obtenidos por cada categoría

		100%	75%	50%	0%
	Objeto	Alta	Media	Baja	No tiene
	objeto 1	10	0	0	0
	objeto 2	10	0	0	0
	objeto 3	0	2	0	0
	objeto 4	0	0	5	0
	objeto 5	0	0	0	4
Total Objetos	31	20	2	5	4
Adherencia Total	77%	65%	5%	8%	0%

Limitaciones de la Automatización

CHOUCAIR®
Business Centric Testing

- Concentrarse en la automatización puede hacer que los probadores se olviden de la gestión de riesgos para el proyecto
- La paradoja del pesticida aumenta cuando se usa la automatización, ya que exactamente las mismas pruebas se ejecutan cada vez
- Las fallas de automatización a menudo no son fallas de la aplicación que se esta probando; están en la propia automatización (falsos positivos)
- Sin una programación inteligente en las pruebas, las herramientas son de mentalidad literal y estúpidas; los probadores no lo son.
- Las herramientas tienden a ser de un subproceso único, es decir, solo buscan un resultado para un evento cuando en realidad hay muchos resultados posibles que pueden ocurrir

Limitaciones de la Automatización

CHOUCAIR®
Business Centric Testing

- Se requerirán pruebas manuales (pruebas exploratorias, etc.)
- El análisis, el diseño y la implementación probablemente aún son manuales
- Los seres humanos encuentran la mayoría de los bugs; la automatización solo puede encontrar lo que está programado para encontrar
- Falso sentido de seguridad cuando se ejecutan grandes cantidades de pruebas pero se descubren pocos fallos
- Problemas técnicos para el proyecto al estar a la vanguardia de las herramientas y técnicas
- Requiere cooperación con el equipo de desarrollo y esto puede crear problemas al interior de las compañías.

Problemas de la Automatización

CHOUCAIR®
Business Centric Testing

- Cuando se inicio con la automatización se obtuvieron tasas de fallos muy altas
- Se identifica como problema principal que la contrucción de las pruebas no se llevo a cabo de manera correcta:
  - Cada prueba se ejecutó exactamente igual siempre
  - Los test se trabajan como tareas de poca habilidad (es decir, cualquier mono puede hacerlo)
  - El cambio a la GUI ocurre a menudo, rompiendo los guiones
- De hecho, cada prueba realizada manualmente desencadena muchas decisiones instantáneas del probador; requiere de mucha inteligencia para que funcione
- Los guiones de prueba, en su versión más básica, contienen tres columnas
  - Una tarea abstracta a realizar
  - Los datos a utilizar para la tarea abstracta
  - El comportamiento esperado

Contexto y Racionalidad

CHOUCAIR®
Business Centric Testing

- El conjunto de casos de prueba en un archivo u otro medio por sí solo no es muy útil
- Debe estar integrado con un probador manual para agregarle valor
- El probador manual le añade
  - Contexto: un entendimiento del del sistema, cuándo, dónde, cómo y qué debe hacerse
  - Racionalidad: ¿Hizo lo que hizo el sistema, y el tiempo que tomó tiene sentido en el contexto en el que nos encontramos?
- Las herramientas automatizadas y los script carecen de “inteligencia”: tienen un diminuto sentido del contexto y poca racionalidad
- Sin embargo, cada herramienta de automatización es, en el fondo, un lenguaje de programación
- Los desarrolladores(Automatizadores) utilizan el lenguaje de programación para añadir contexto y racionalidad a los script de automatización

Programación Requerida

CHOUCAIR
Business Centric Testing

- La automatización funcional solo puede tener éxito cuando la inteligencia de los probadores manuales se agrega a los guiones mediante la programación
- Esto requiere
  - Análisis de lo que está sucediendo realmente en la prueba
  - Diseño de cómo resolver los problemas presentados
  - Implementación utilizando herramientas y bibliotecas invocables
  - Documentación
  - Mantenimiento cuando ocurre un cambio
- Un script automatizado es un programa que se escribe para probar otro programa.

Cómo Alcanzar el Éxito en la Automatización

CHOUCAIR®
Business Centric Testing

- Es claro que la automatización no tendrá éxito por simple casualidad
- Este éxito solo se alcanza cuando ponemos en marcha las mismas cosas que ayudan a que un proyecto de software se culmine de manera exitosa:
  - Un plan a largo plazo que se alinea con las necesidades comerciales
  - Buena gestión y atención a los detalles
  - Madurez del proceso
  - Una arquitectura formalizada y un marco para la automatización
  - Capacitación adecuada en ingeniería de prueba y de desarrollo
  - La capacidad de crear y usar buena documentación
- Lo anterior no va a garantizar un éxito rotundo, pero la falta de ello sin duda puede dar un empujoncito hacia el fallo

Capacidad de Ser Automatizado

CHOUCAIR®
Business Centric Testing

- El proceso de La automatización difícilmente tendrá éxito si se realiza en una isla
- Obtener el apoyo y la ayuda del desarrollo realmente pueden ayudar al éxito
- Una Sistema Bajo Pruebas que está diseñado con la capacidad de ser probado puede ayudar
- Existen varios niveles diferentes de interfaz para la automatización
  - El nivel de GUI (a menudo frágil y propenso a fallos)
  - Interfaz de Programación de Aplicaciones (IPA) para uso público y protegido
  - Nivel de protocolo (HTTP, TCP, UD, etc.)
  - Nivel de servicio (SOAP, REST, etc.)
- Cuanto más bajo esté en esa lista, la automatización probablemente será menos frágil y más robusta

Factores de Éxito

CHOUCAIR®
Business Centric Testing

- Administración que ha sido educada para comprender lo que es y lo que no es posible (las expectativas poco realistas de la administración son una de las principales razones por las que fallan los proyectos de automatización de pruebas de software)
- Un equipo de desarrollo para los Sistemas Bajo Pruebas que entiende la automatización y está dispuesto a trabajar con los probadores
- Sistemas Bajo Pruebas que están diseñados para la capacidad de ser automatizados
- Desarrollar un estudio de viabilidad comercial(ROI) a corto, mediano y largo plazo
- Tener las herramientas adecuadas para trabajar en el entorno y con los Sistemas Bajos Pruebas.

Factores de Éxito

CHOUCAIR®
Business Centric Testing

- La capacitación correcta para tareas de prueba y de automatización.
- Tener una arquitectura y un marco bien diseñados y bien documentados para admitir los script´s de pruebas.
- Tener una estrategia de automatización de pruebas bien documentada, financiada y aceptada por la administración
- Un plan formal y bien documentado para el mantenimiento de la automatización
- Automatización en el nivel de interfaz correcto para el contexto de las pruebas necesarias

Herramientas / Frameworks

CHOUCAIR
Business Centric Testing

En el mercado existe una gran variedad de herramientas y frameworks de automatización y de apoyo que al integrarlas conforman un ecosistema de desarrollo. Dentro de esa variedad se encuentran herramientas comerciales (requieren de un licenciamiento con costo) y otras de libre uso denominadas Open Source. A continuación se presenta una relación categorizada:

Herramientas BDD Open Source

- cucumber
- jbehave
- specflow

Herramientas / Frameworks de Automatización Open Source

- Choucair Libraries
- Selenium
- Serenity BDD
- appium
- JUnit
- TestNG
- WinAppDriver
- Karate
- REST-assured

Plataformas Comerciales

- TRICENTIS
- TOSCA TESTSUITE
- Grupo HDI
- Katalon
- TestComplete

Herramientas de Apoyo

- Jenkins
- GitLab
- Bitbucket
- Azure DevOps

Arquitectura >> Herramientas por Enfoque

CHOUCAIR
Business Centric Testing

Herramientas de apoyo para BDD (Behavior Driven Development): Antes de nombrar las herramientas que apoyan BDD, recordemos que es; *BDD es una estrategia de desarrollo*, no es una técnica de testing, se define como un idioma común entre todos los ***stakeholders***, con el fin de mejorar la comunicación y el entendimientos entre equipos técnicos y no técnicos, dicha comunicación se apoya en el lenguaje Gherkin.

Gherkin es un DSL o Lenguaje Específico de Dominio (Domain-Specific Languaje), es decir, un lenguaje que está creado para resolver un problema y ser entendible y legible por el área de negocio, soporta más de 60 idiomas y sirve simultáneamente como documentación de apoyo al desarrollo y a las pruebas automatizadas.

Ahora, existe una gran variedad de herramientas que apoyan BDD mediante el análisis e interpretación del lenguaje gherkin, y cumpliendo los siguientes propósitos:

- Documentación del sistema a nivel de su comportamiento.
- Apoyar a la generación de test automatizados
- Proporcionar especificaciones ejecutables comprensibles para todos.

Aunque ***Cucumber*** es la herramientas más conocida, existen otras disponibles y que apoyan diferentes lenguajes:

Specflow, NSpec, NBehave para .NET
Jbehave, Jdave, Instinct para Java

Arquitectura >> Herramientas por Enfoque

CHOUCAIR
Business Centric Testing

Herramientas / Frameworks de Automatización de Pruebas

Pruebas Unitarias: es una forma de comprobar el correcto funcionamiento de una unidad de código. Por ejemplo en diseño estructurado o en diseño funcional una función o un procedimiento, en diseño orientado a objetos una clase. Esto sirve para asegurar que cada unidad funcione correctamente y eficientemente por separado. Además de verificar que el código hace lo que tiene que hacer, verificamos que sea correcto el nombre, los nombres y tipos de los parámetros, el tipo de lo que se devuelve, que si el estado inicial es válido, entonces el estado final es válido también.

framework open source **para la automatización de las pruebas** (tanto unitarias como de integración) en los proyectos software. El framework le provee al usuario herramientas, clases y métodos que le facilitan la tarea de realizar pruebas en su sistema y así asegurar su consistencia y funcionalidad. También sirve como **herramienta para realizar las pruebas de regresión,** que se ejecutan cuando una parte del código ha sido modificada y sea necesario comprobar que se sigue cumpliendo con todos los requisitos.

A continuación algunos frameworks:
Junit, **TestNG** para Java
Nunit para .NET
Typemock, CPPUnit, Cunit para C/C++
PyUnit para Python
PHPUnit, Simple Test para PHP

Arquitectura >> Herramientas por Enfoque

CHOUCAIR
Business Centric Testing

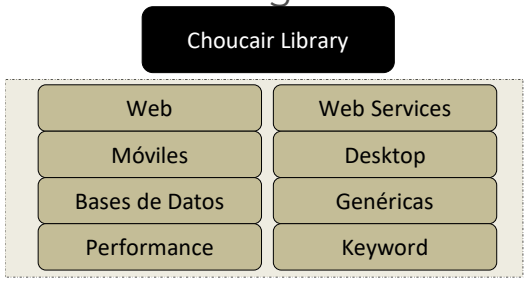
Herramientas / Frameworks de Automatización de Pruebas

Pruebas funcionales: son pruebas basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software (requisitos funcionales y/o HU).

Las pruebas funcionales se pueden realizar sobre distintas tecnologías, a continuación relacionamos algunas Open Source:

Selenium, Serenity, para Web.
Appium, para moviles.
Winium, WinAPP, SikuliX, WinAppDriver para aplicaciones Desktop, aplicaciones Windows
Soap UI, Jmeter, Postman, Rest assured, Karate para API, Rest, SOAP.

Choucair Library, Choucair, cuenta con una librería propia que brinda un conjunto de utilidades (funcionalidades, clases y métodos) genéricas que con su reúso facilitan el proceso de construcción de Scripts de automatización para distintas tecnologías.



Arquitectura >> Herramientas por Enfoque

Herramientas / Frameworks de Automatización de Pruebas

Además de las herramientas Open Source, existe una gran variedad de herramientas o suites Comerciales robustas que brindan entre sus características, la automatización multiplataforma, y actividades de gestión de proyectos, además de no requerir de conocimientos técnicos.

SilkTest

Es una herramienta de automatización de pruebas funcionales y de regresión para aplicaciones empresariales, móviles, nativas y web.

Katalon

es una solución de prueba de automatización desarrollada por Katalon LLC. El software está construido sobre los marcos de automatización de código abierto Selenium, Appium con una interfaz IDE especializada para pruebas de aplicaciones web, API, móviles y de escritorio.

Ranorex Studio, es un marco de automatización de pruebas GUI proporcionado por Ranorex GmbH , una compañía de desarrollo de software. El marco se utiliza para probar aplicaciones de escritorio, basadas en web y móviles.

TOSCA de Tricentis

Plataforma de pruebas continuas, acelera las pruebas con un enfoque sin código y sin script para la automatización de pruebas de extremo a extremo. Con soporte para más de 160 tecnologías y aplicaciones empresariales, Tosca proporciona automatización de prueba resistente para cualquier caso de uso.

TestComplete

es una plataforma funcional de pruebas automatizadas desarrollada por SmartBear Software. TestComplete ofrece a los evaluadores la capacidad de crear pruebas automatizadas para aplicaciones de Microsoft Windows, Web, Android e iOS

STARC

Modelo apoyado en la plataforma **STARC** (Sistema de Tests Automatizado con Reutilización de Código). En éste se modelan las funcionalidades y requisitos de la aplicación (flujos principales y alternativos), y se suministran los detalles de su arquitectura (banco de datos, servidores de aplicación, lenguajes).

A partir del modelo, STARC propone casos de test para los flujos modelados, cruzando diversas informaciones suministradas con sus bancos de conocimiento y, automáticamente genera scripts de test para ejecución en las principales herramientas del mercado.

Los resultados y evidencias son recolectados y los reportes de calidad se generan automáticamente y así se compone la documentación del proceso de testing, generando feedback para las áreas de negocio y de TI.

Herramientas de Apoyo

El proceso de Automatización es similar al de un proceso de desarrollo de aplicaciones, por lo mismo es necesario contar con un ecosistema que lo soporte, facilitando el mantenimiento, versionado, orquestación de ejecuciones e integración continua.

Control de Versiones, se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación, las herramientas más utilizadas son **git** y **subversion**.

Repositorios, servicio de alojamiento para los proyectos que usan controles de versión. Algunas herramientas son Github, Bitbucket,

Jenkins, es por definición, un servidor open source de integración continua (CI). Es el software de automatización más usado de todos, escrito en Java. Es muy conveniente al contar con más de 14.000 plugins para soportar la automatización de todo tipo de tareas.

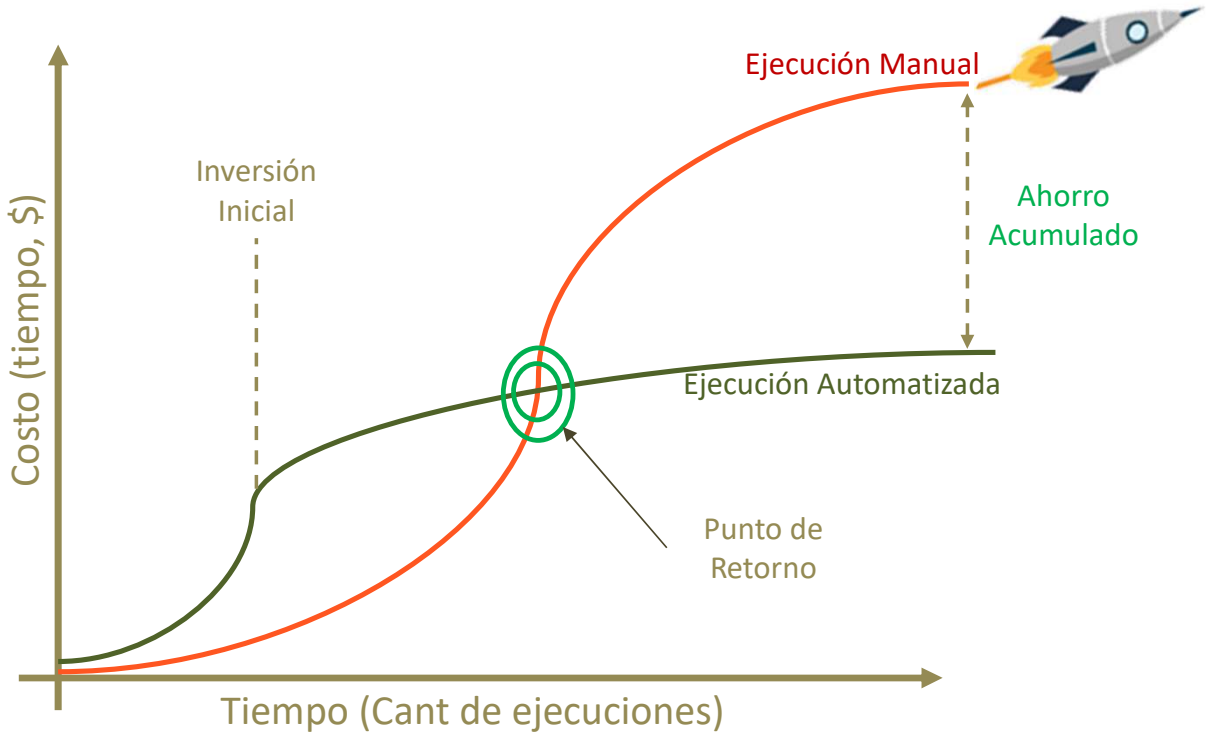
Azure Devops, plataforma para la planeación, colaboración y gestión de proyectos, cuenta con un moderno conunto de servicios de desarrollo como son:

- Test Plans: Probar y distribuir soluciones con confianza usando herramientas de pruebas manuales y exploratorias.
- Pipelines: Compilar, probar e implementar código con CI/CD que funciona con cualquier lenguaje, plataforma y nube.
- Repository: Brinda un número ilimitado de repositorios GIT hospedados en la nube.
- Artifacts: Crear, hospedar y compartir paquetes.

Retorno de la Inversión - ROI

CHOUCAIR®
Business Centric Testing

Viabilidad Funcional — Viabilidad Técnica — Determinar ROI

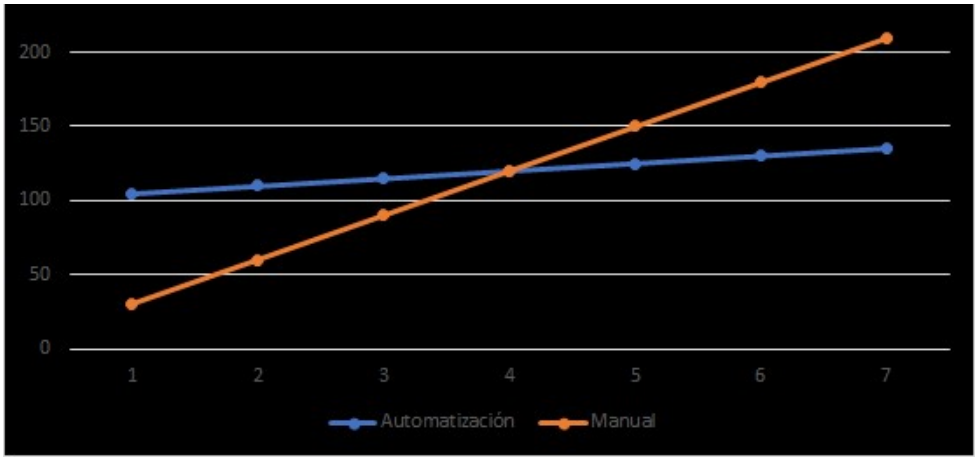


Retorno de la Inversión (ROI)

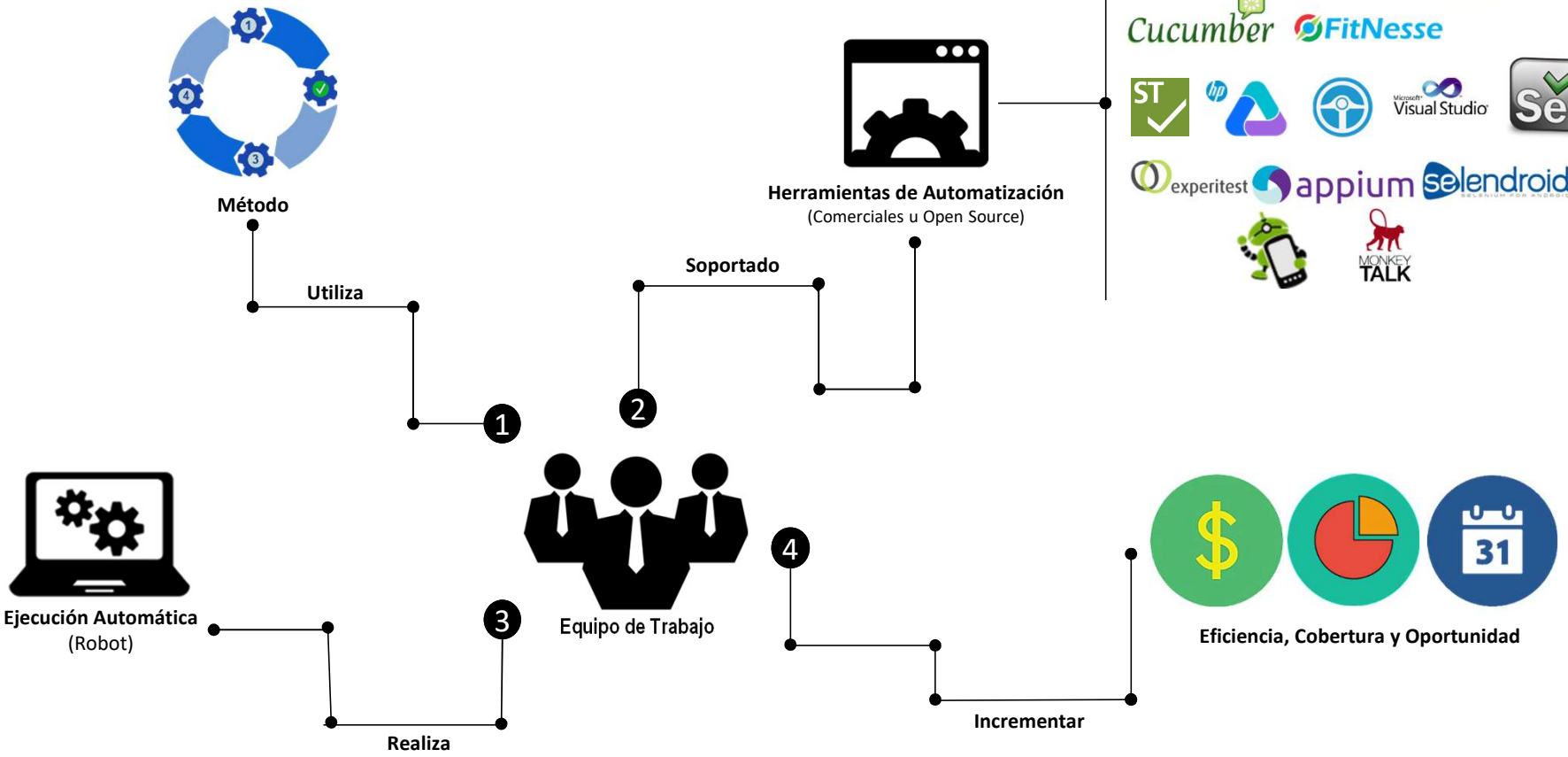
Como todos sabemos, automatizar requiere un alto costo e inversión el cual debe ser tenido en cuenta al momento de evaluar el costo / beneficio de la automatización.
En ocasiones el beneficio obtenido no es suficientemente significativo como para recuperar la inversión realizada, en ocasiones este tipo de análisis ayuda a determinar si es viable o no embarcarse en el proyecto.
Uno de los aspectos a tener en cuenta es el nivel de uso, es decir la cantidad de ejecuciones estimada.

Para el ejemplo, estimamos que la ejecución manual del proceso a automatizar tarda 30 horas, suponemos una inversión de 100 horas en la construcción y un estimado de ejecución con automatización de 5 horas, si hacemos el siguiente cálculo:
Inversion / (Ejecución Manual – Ejecución con Auto), obtenemos el punto de equilibrio o ROI.
Si proyectamos y graficamos las ejecuciones manual y automatizada, vemos el punto de cruce (ROI) y el comportamiento en el ahorro de las ejecuciones

Ejecución Manual	30
Inversion	100
Ejecución con Auto.	5
ROI	4



Resumiendo …..


Método
Utiliza
1
Ejecución Automática
(Robot)
Realiza
3
Herramientas de Automatización
(Comerciales u Open Source)
Soportado
2
Equipo de Trabajo
4
Incrementar
Eficiencia, Cobertura y Oportunidad
CHOUCAIR
Business Centric Testing
jbehave
JUnit
N unit
Cucumber
FitNesse
ST
Visual Studio
Se
experitest
appium
selendroid
MONKEY TALK

Beneficios de la Automatización en las Pruebas Especializadas de Datos

Verificar que los datos cumplan con criterios de exactitud, completitud, integridad, consistencia, disponibilidad y confiabilidad

Pruebas de Migración de Datos

Pruebas Automatizadas que cubren el 100% del set de datos para la prueba y reglas de negocio

Datos completos y sin ambigüedad

Pruebas de Business Intelligence

Modelos de datos precisos, disponibles, acordes con las necesidades, reglas y objetivos de negocio de forma ágil.

Cumplimiento de objetivos y eficiencia de costos, optimizando su Time to market .

Apoyan el descubrimiento de fallas y la medición en la Calidad de sus Datos