Unidad 2
Principios, prácticas y procesos clave de las pruebas ágiles

1. Diferencias entre pruebas según un enfoque tradicional o un enfoque ágil

Según se describe en el plan de estudio de Nivel Básico [ISTQB_FL_SYL] y en [Black09], las actividades de prueba están relacionadas con las actividades de desarrollo, y por lo tanto las pruebas varían en los distintos ciclos de vida. Los probadores deben conocer las diferencias entre las pruebas en modelos de ciclo de vida tradicionales (por ejemplo, secuenciales, como el modelo V o iterativas como RUP) y los ciclos de vida ágiles para poder trabajar de una forma eficiente y eficaz. Los modelos ágiles difieren en la forma en la que se integran las actividades de pruebas y desarrollo, los productos de trabajo del proyecto, los nombres, los criterios de entrada y salida utilizados para los distintos niveles de prueba, el uso de herramientas y cómo pueden utilizarse pruebas independientes de forma eficaz.

Los probadores deben recordar que las organizaciones varían considerablemente en cómo abordan la implementación de los ciclos de vida, por lo que desviarse de los ideales de los ciclos de vida ágiles puede suponer una personalización y una adaptación inteligente de las prácticas. La capacidad para adaptarse al contexto de un proyecto dado, incluyendo las prácticas de desarrollo de software actualmente seguidas, constituye un factor de éxito clave para los probadores.

1.1 Actividades de pruebas y desarrollo

Una de las principales diferencias entre los ciclos de vida tradicionales y los ciclos de vida ágiles es el concepto de iteraciones muy cortas, cada iteración da lugar a un software que funciona que proporciona prestaciones de valor para el negocio. Al inicio del proyecto, hay un periodo para la planificación de la entrega. A éste le sigue una secuencia de iteraciones. Al inicio de cada iteración, hay un periodo de planificación de la iteración. Una vez establecido el alcance de la iteración, las historias de usuario seleccionadas se desarrollan, se integran en el sistema y se prueban. Estas iteraciones son muy dinámicas, con actividades de desarrollo, integración y pruebas a lo largo de cada iteración, y con un paralelismo y un solapamiento considerables. Las actividades de prueba se llevan a cabo a lo largo de la iteración, no como una actividad final.

Cada uno de los probadores, desarrolladores y partes interesadas del negocio desempeña una función en las pruebas, al igual que en los ciclos de vida tradicionales. Los desarrolladores llevan a cabo pruebas unitarias a partir de las historias de usuario. A continuación, los probadores prueban estas funcionalidades. Las partes interesadas del negocio también prueban las historias durante la implementación. Las partes interesadas del negocio pueden utilizar casos de prueba escritos, pero también pueden sencillamente experimentar con el software y utilizar la funcionalidad para proporcionar feedback rápido al equipo de desarrollo.

En algunos casos, se realizan iteraciones de robustez o estabilización de forma periódica para resolver defectos residuales y demás formas de deuda técnica. Sin embargo, las mejores prácticas establecen que ninguna prestación se considere "hecha" (finalizada) hasta que se haya integrado y probado con el sistema [Goucher09]. Otra buena práctica es abordar los defectos restantes de la iteración anterior al inicio de la siguiente iteración, como parte del backlog de dicha iteración (conocida como "arreglar los defectos primero"). Sin embargo, algunos se quejan de que esta práctica da lugar a una situación en la que se desconoce el trabajo total a realizar en la iteración y será más difícil estimar cuándo pueden hacerse las funcionalidades restantes. Al final de una secuencia de iteraciones puede haber una serie de actividades de entrega para conseguir que el software esté listo para su entrega, si bien en algunos casos la entrega se produce al final de cada iteración.

Cuando se utilizan pruebas basadas en el riesgo como una de las estrategias de pruebas, se realiza un análisis de riesgos de alto nivel durante la planificación de la entrega, siendo los probadores los que en muchas ocasiones dirigen dicho análisis. Sin embargo, los riesgos de calidad específicos asociados a cada iteración se identifican y evalúan en la planificación de la iteración. Este análisis de riesgos puede afectar tanto a la secuencia de desarrollo como a la prioridad y profundidad de las pruebas de las prestaciones. También afecta a la estimación del esfuerzo de prueba necesario para cada prestación.

En algunas prácticas ágiles (como por ejemplo, en la programación extrema), se recurre al trabajo por parejas. El trabajo por parejas puede implicar que los probadores trabajen juntos de dos en dos para probar una funcionalidad. El trabajo por parejas también puede suponer que un probador trabaje de forma conjunta con un desarrollador para desarrollar y probar una funcionalidad. El trabajo por parejas puede ser difícil si el equipo de pruebas está deslocalizado, pero los procesos y las herramientas pueden ayudar a realizarlo.

Los probadores también pueden hacer las veces de coaches de pruebas y calidad dentro del equipo, poniendo en común el conocimiento de pruebas y contribuyendo a labores de aseguramiento de la calidad dentro del equipo. Esto promueve un sentimiento de propiedad colectiva de la calidad del producto.

La automatización de las pruebas a todos los niveles tiene lugar en muchos equipos ágiles, y esto puede significar que los probadores dediquen tiempo a crear, ejecutar, hacer

seguimiento y mantener pruebas y resultados automatizados. Dado el uso intensivo de la automatización de pruebas, en los proyectos ágiles tiende a realizarse un porcentaje mayor de las pruebas manuales empleando técnicas basadas en la experiencia y técnicas basadas en defectos, tales como ataques de software, pruebas exploratorias, y predicción de errores.

Mientras que los desarrolladores se concentran en crear pruebas unitarias, los probadores deben concentrarse en crear pruebas automatizadas de integración, de sistema y de integración de sistemas. Esto hace que dentro de los equipos ágiles exista una tendencia a favorecer a probadores que cuentan con una importante experiencia técnica en automatización de pruebas.

Un principio ágil básico es que pueden producirse cambios durante el proyecto. Por lo tanto, en los proyectos ágiles se fomenta la documentación ligera. Los cambios en prestaciones existentes tienen implicaciones en las pruebas, especialmente en las pruebas de regresión. El uso de las pruebas automatizadas es una forma de gestionar la cantidad de esfuerzo de pruebas asociado al cambio. Sin embargo, es importante que el ritmo de cambios no supere la capacidad del equipo para gestionar los riesgos asociados a dichos cambios.

1.2 Productos de trabajo del proyecto

Los productos de trabajo del proyecto de interés inmediato para los probadores ágiles normalmente pueden clasificarse en tres categorías:

1. Los productos de trabajo orientados al negocio que describen qué se necesita (por ejemplo, especificaciones de requisitos) y cómo utilizarlo (por ejemplo, documentación de usuario).
2. Los productos de trabajo de desarrollo que describen cómo se construye el sistema (por ejemplo, diagramas entidad-relación de base de datos), la implementación del sistema (por ejemplo, código) o que evalúan partes individuales de código (por ejemplo, pruebas unitarias automatizadas).
3. Los productos de trabajo de pruebas que describen cómo se prueba el sistema (por ejemplo, estrategias y planes de pruebas), que realmente prueban el sistema (por ejemplo, pruebas manuales y automatizadas), o que presentan los resultados de las pruebas.

En un proyecto ágil típico, es una práctica común evitar generar grandes cantidades de documentación. En su lugar, el foco se centra en disponer de un software que funcione, junto con pruebas automatizadas que demuestren el cumplimiento de los requisitos. La idea de reducir la documentación es aplicable exclusivamente a documentación que no ofrece valor al cliente. En un proyecto ágil de éxito, se logra un equilibrio entre aumentar la eficiencia reduciendo la documentación y facilitar suficiente documentación para dar soporte a las actividades de negocio, pruebas, desarrollo y mantenimiento. El equipo debe decidir durante

la planificación de la entrega qué productos de trabajo son necesarios y qué nivel de documentación de productos de trabajo es necesario.

Los productos de trabajo típicos orientados al negocio incluyen historias de usuario y criterios de aceptación. Las historias de usuario constituyen la versión ágil de las especificaciones de requisitos, y deberían explicar cómo debe comportarse el sistema por lo que respecta a una prestación o funcionalidad única y coherente. Una historia de usuario debe definir una funcionalidad que sea lo suficientemente pequeña como para completarse en una única iteración. Las colecciones más grandes de funcionalidades relacionadas, o una colección de funciones que componen una sola funcionalidad compleja, pueden denominarse épicas. Las épicas pueden incluir historias de usuario para equipos de desarrollo diferentes. Por ejemplo, una historia de usuario puede describir lo que se necesita a nivel de API (middleware) mientras que otra historia describe lo que se necesita a nivel de UI (aplicación). Estas colecciones pueden desarrollarse a lo largo de una serie de iteraciones. Cada épica y sus historias de usuario deben tener criterios de aceptación asociados.

Entre los productos de trabajo típicos del desarrollador en proyectos ágiles se encuentra el código. Asimismo, los desarrolladores ágiles a menudo crean pruebas unitarias automatizadas. Estas pruebas pueden crearse después del desarrollo del código. En algunos casos, no obstante, los desarrolladores crean pruebas de forma incremental, antes de escribir cada una de las partes del código, con vistas a proporcionar una forma de verificar, una vez escrita dicha parte del código, si esta funciona según lo esperado. Si bien este enfoque se conoce como desarrollo de probar primero o desarrollo guiado por pruebas, en realidad las pruebas más que pruebas son una especie de especificaciones ejecutables de diseño de bajo nivel [Beck02].

Los productos de trabajo típicos del probador en proyectos ágiles incluyen pruebas automatizadas, además de documentos como planes de pruebas, análisis de riesgos de calidad, pruebas manuales, informes de defectos y resultados de las pruebas. Los documentos se crean de la manera más ligera posible, lo que a menudo también es cierto en ciclos de vida tradicionales. Los probadores también generarán métricas de pruebas a partir de los informes de defectos y de los resultados de las pruebas, y de nuevo se pone énfasis en un enfoque ligero.

En algunas implementaciones ágiles, en proyectos y productos especialmente regulados, críticos para la seguridad o muy complejos, se requiere una mayor formalización de estos productos de trabajo. Por ejemplo, algunos equipos transforman las historias de usuario y los criterios de aceptación en especificaciones de requisitos más formales. Pueden elaborarse informes de trazabilidad vertical y de trazabilidad horizontal para satisfacer auditorías, normativas y otros requisitos.

1.3 Niveles de prueba

Los niveles de pruebas son actividades de prueba que están relacionados de forma lógica, a menudo por la madurez o integridad del elemento objeto de la prueba.

En los modelos de ciclo de vida secuenciales, los niveles de pruebas a menudo se definen de forma que los criterios de salida de un nivel forman parte de los criterios de entrada para el siguiente nivel. En algunos modelos iterativos, esta regla no es aplicable. Los niveles de pruebas se solapan. Las actividades de especificación de requisitos, especificación de diseño y actividades de desarrollo pueden solaparse con los niveles de prueba.

En algunos ciclos de vida ágiles, el solapamiento se produce porque pueden producirse cambios en requisitos, diseño y código en cualquier punto de una iteración. Mientras que Scrum, en teoría, no permite cambios en las historias de usuario después de la planificación de la iteración, en la práctica, a veces estos cambios sí se producen. Durante una iteración, cualquier historia de usuario normalmente avanzará de manera secuencial a través de las siguientes actividades de prueba:

- Pruebas unitarias, típicamente realizadas por el desarrollador
- Pruebas de aceptación de prestaciones, que a veces se dividen en dos actividades:
    - Las pruebas de *verificación* de prestaciones, que a menudo se automatizan, las pueden realizar los desarrolladores o los probadores y suponen la realización de pruebas contra los criterios de aceptación de la historia de usuario.
    - Las pruebas de *validación* de prestaciones, que normalmente son manuales y pueden afectar a desarrolladores, probadores y partes interesadas del negocio que trabajan conjuntamente para establecer si la prestación es apta para su uso, para mejorar la visibilidad del progreso conseguido y para recibir feedback real de las partes interesadas del negocio.

Además, a menudo hay un proceso paralelo de pruebas de regresión que se produce durante la iteración. Esto supone ejecutar de nuevo las pruebas unitarias automatizadas y las pruebas de verificación de las funcionalidades a partir de la iteración actual y de las iteraciones anteriores, generalmente a través de la integración continua.

En algunos proyectos ágiles, puede existir un nivel de pruebas de sistema, que se inicia una vez que la primera historia de usuario está lista para dichas pruebas. Puede implicar la ejecución de pruebas funcionales, así como pruebas no funcionales para pruebas de rendimiento, fiabilidad, usabilidad y demás tipos de pruebas relevantes.

Los equipos ágiles pueden emplear varias formas de pruebas de aceptación (empleando el término según se explica en el plan de estudios de Nivel Básico [ISTQB_FL_SYL]). Pueden realizarse pruebas alfa internas y pruebas beta externas, bien al cierre de cada iteración, tras la finalización de cada iteración, o tras una serie de iteraciones. También pueden realizarse

pruebas de aceptación de usuario, pruebas de aceptación operativas, pruebas de aceptación normativas, y pruebas de aceptación contractuales, bien al cierre de cada iteración, bien tras la finalización de cada iteración, o tras una serie de iteraciones.

1.4 Pruebas y gestión de la configuración

Los proyectos ágiles a menudo implican un uso intensivo de herramientas automatizadas para desarrollar, probar y gestionar el desarrollo de software. Los desarrolladores utilizan herramientas para el análisis estático, las pruebas unitarias y la cobertura de código. Los desarrolladores comprueban continuamente el código y las pruebas unitarias en un sistema de gestión de la configuración a través de marcos de trabajo automatizados de construcción y pruebas. Estos marcos de trabajo permiten la integración continua del nuevo software en el sistema junto con el análisis estático y las pruebas unitarias que se ejecutan repetidamente a medida que se hace la subida del nuevo software [Kubaczkowski].

Estas pruebas automatizadas también pueden incluir pruebas funcionales a nivel de integración y sistema. Este tipo de pruebas funcionales automatizadas pueden crearse utilizando arneses de prueba funcionales, herramientas de código abierto o comerciales, pruebas funcionales de interfaz de usuario, y pueden integrarse en las pruebas automatizadas que se ejecutan como parte del marco de trabajo de la integración continua. En algunos casos, a causa de la duración de las pruebas funcionales, las pruebas funcionales se separan de las pruebas unitarias y se ejecutan con menos frecuencia. Por ejemplo, las pruebas unitarias pueden ejecutarse cada vez que se hace una subida de código nuevo, mientras que las pruebas funcionales más largas solo se ejecutan cada pocos días.

Un objetivo de las pruebas automatizadas es confirmar que el software construido está funcionando y es instalable. Si alguna prueba automatizada falla, el equipo debe arreglar el defecto subyacente a tiempo para la siguiente subida de código. Esto requiere una inversión en informes de pruebas en tiempo real para ofrecer una buena visibilidad de los resultados de las pruebas. Este enfoque ayuda a reducir ciclos caros e ineficientes de "construir-instalar-fallo-reconstruir-reinstalar" que se dan en muchos proyectos tradicionales, ya que los cambios que rompen el software construido o hacen que el software falle al instalarse se detectan rápidamente.

Las pruebas automatizadas y las herramientas de construcción permiten gestionar el riesgo de regresión asociado a los cambios frecuentes que a menudo se dan en los proyectos ágiles. Sin embargo, confiar excesivamente solo en las pruebas unitarias automatizadas a la hora de gestionar estos riesgos puede suponer un problema, ya que las pruebas unitarias a menudo tienen una eficacia limitada a la hora de detectar defectos [Jones11]. También son necesarias pruebas automatizadas en los niveles de integración y sistema.

1.5 Opciones organizativas para las pruebas independientes

Según se describe en el plan de estudios de Nivel Básico [ISTQB_FL_SYL], los probadores independientes a menudo son más efectivos a la hora de encontrar defectos. En algunos equipos ágiles, los desarrolladores crean muchas de las pruebas en forma de pruebas automatizadas. Uno o más probadores pueden integrarse en el equipo, llevando a cabo muchas de las tareas de prueba. Sin embargo, dada la posición de esos probadores dentro del equipo, existe un riesgo de pérdida de independencia y evaluación objetiva.

Otros equipos ágiles conservan equipos de pruebas totalmente aparte e independientes y asignan probadores a demanda en los últimos días de cada sprint. Esto puede preservar la independencia, y estos probadores pueden ofrecer una evaluación objetiva e imparcial del software. No obstante, las presiones de tiempo, la falta de entendimiento de las nuevas prestaciones del producto, y los problemas de relación con las partes interesadas del negocio y los desarrolladores a menudo dan lugar a problemas con este enfoque.

Una tercera opción es tener un equipo de pruebas independiente y aparte en el que los probadores son asignados a equipos ágiles a largo plazo, al inicio del proyecto, lo que les permite mantener su independencia y obtener un buen conocimiento del producto, además de establecer relaciones sólidas con otros miembros del equipo. Además, el equipo de pruebas independiente puede tener probadores especializados externos a los equipos ágiles para trabajar a largo plazo y/o en actividades independientes de la iteración, como desarrollar herramientas de pruebas automatizadas, llevar a cabo pruebas no funcionales, crear y soportar entornos y datos de prueba, y realizar niveles de pruebas que pueden no adecuarse a un sprint (por ejemplo, pruebas de integración de sistemas).

2. Estado de las pruebas en los proyectos ágiles

En los proyectos ágiles los cambios se producen rápido. Estos cambios pueden suponer evoluciones constantes en el estado de las pruebas, el progreso de las pruebas y la calidad del producto, y los probadores deben buscar la forma de hacer llegar esa información al equipo para que puedan adoptar las decisiones oportunas para seguir completando cada iteración con éxito. Además, los cambios pueden afectar a prestaciones existentes procedentes de iteraciones previas. Por lo tanto, deben actualizarse las pruebas manuales y automatizadas para hacer frente de manera eficaz al riesgo de regresión.

2.1 Comunicación del estado de la prueba, progreso y calidad del producto

Los equipos ágiles avanzan produciendo software que funciona al final de cada iteración. Para determinar cuándo tendrá el equipo software que funciona, es preciso hacer un seguimiento del progreso de todos los elementos de trabajo de la iteración y de la entrega. Los probadores de equipos ágiles utilizan varios métodos para registrar el progreso y el estado de las pruebas, incluyendo los resultados de la automatización de las pruebas, la progresión de tareas e historias en el tablero de tareas ágil, y los gráficos Burndown que muestran el progreso del equipo. Estos pueden comunicarse al resto del equipo a través de medios como cuadros de mando tipo wiki y correos electrónicos tipo cuadros de mando, además de verbalmente en reuniones diarias de pie. Los equipos ágiles pueden utilizar herramientas que generan informes de estado automáticamente a partir de los resultados de las pruebas y el progreso de las tareas, que a su vez actualizan los cuadros de mando tipo wiki y correos electrónicos. Esta forma de comunicación también reúne métricas del proceso de pruebas, que pueden utilizarse para la mejora del proceso. Comunicar el estado de las pruebas de esta forma automatizada también deja más tiempo y libera a los probadores para concentrarse en diseñar y ejecutar más casos de prueba.

Los equipos pueden utilizar gráficos Burndown para hacer un seguimiento del progreso en toda la entrega y dentro de cada iteración. Un gráfico Burndown [Crispin08] representa la cantidad de trabajo pendiente de realizar en comparación con el tiempo asignado para la entrega o la iteración.

Para ofrecer una representación visual instantánea y detallada del estado actual de todo el equipo, incluyendo el estado de las pruebas, los equipos pueden recurrir a tableros de tareas ágiles. El tablero de tareas refleja las tarjetas de las historias, las tareas de desarrollo, las tareas de pruebas y demás tareas creadas durante la planificación de la iteración , a menudo empleando tarjetas de colores para establecer el tipo de tarea. Durante la iteración, el progreso se gestiona moviendo estas tareas por el tablero de tareas en columnas del tipo trabajo pendiente, trabajo en curso, por verificar y hecho. Los

equipos ágiles pueden utilizar herramientas para mantener sus tarjetas de historias y tableros de tareas ágiles, y pueden automatizar la actualización de los tableros y de los estados.

Las tareas de pruebas en el tablero de tareas se ven reflejadas en los criterios de aceptación definidos para las historias de usuario. A medida que los scripts de automatización de pruebas, las pruebas manuales y las pruebas exploratorias de una tarea de prueba pasan a un estado aprobado, la tarea pasa a la columna "hecho" del tablero de tareas. El equipo completo revisa el estado del tablero de tareas periódicamente, a menudo en reuniones diarias de pie, para asegurarse de que las tareas se mueven por el tablero a un ritmo aceptable. Si alguna tarea (incluidas las tareas de pruebas) no se mueve o se mueve con demasiada lentitud, el equipo revisará y solucionará cualquier problema que pueda estar bloqueando su progreso.

La reunión diaria de pie incluye a todos los miembros del equipo ágil, también a los probadores. En esta reunión, todos comunican su estado actual. La agenda de cada miembro es [Guía de la Alianza Ágil]:

- ¿Qué ha hecho desde la última reunión?
- ¿Qué tiene previsto hacer antes de la próxima reunión?
- ¿Algún impedimento?

En las reuniones diarias de pie se comunica cualquier problema que pueda bloquear el progreso de las pruebas, de manera que todo el equipo esté al corriente de los problemas y pueda solucionarlos en consecuencia.

Para mejorar la calidad general del producto, muchos equipos ágiles llevan a cabo encuestas de satisfacción para recibir feedback sobre si el producto cumple las expectativas del cliente. Los equipos pueden utilizar otras métricas parecidas a las obtenidas en metodologías de desarrollo tradicionales, tales como la tasa de pruebas superadas/fallidas, la tasa de detección de defectos, los resultados de las pruebas de confirmación y regresión, la densidad de los defectos, los defectos encontrados y arreglados, la cobertura de requisitos, la cobertura de riesgo y la cobertura de código para mejorar la calidad del producto. Como con cualquier ciclo de vida, las métricas obtenidas y reportadas deben ser relevantes y ayudar a la toma de decisiones. Las métricas no deben utilizarse para recompensar, penalizar o aislar a ningún miembro del equipo.

2.2 Gestión de riesgos de regresión con casos de prueba manuales y automatizados

En un proyecto ágil, el producto crece a medida que van completándose las iteraciones. Por lo tanto, el alcance de las pruebas también crece. Además de probar los cambios de código realizados en la iteración actual, los probadores también tienen que comprobar que no se ha introducido ninguna regresión en las funcionalidades que han sido desarrolladas y

probadas en iteraciones anteriores. El riesgo de introducir regresión en los desarrollos ágiles es alto debido a la amplia rotación de código (líneas de código añadidas, modificadas o eliminadas de una versión a otra). Dado que dar respuesta al cambio es un principio clave de los procesos ágiles, también pueden hacerse cambios a funcionalidades ya entregadas para ajustarse a las necesidades del negocio. Para mantener la velocidad sin incurrir en una gran cantidad de deuda técnica, es fundamental que los equipos inviertan en la automatización de las pruebas en todos los niveles de pruebas lo antes posible. También es fundamental que todos los activos de las pruebas, tales como las pruebas automatizadas, los datos de prueba y otros artefactos de pruebas se mantengan actualizados con cada iteración. Se recomienda que todos los activos de las pruebas se mantengan en una herramienta de gestión de la configuración para permitir el control de versiones, garantizar la facilidad de acceso por parte de todos los miembros del equipo y facilitar la realización de los cambios necesarios a causa de cambios en la funcionalidad sin perder la información histórica de los activos de las pruebas.

Dado que casi nunca se pueden repetir íntegramente todas las pruebas, especialmente en proyectos ágiles con plazos ajustados, los probadores tienen que asignar tiempo en cada iteración a revisar los casos de prueba manuales y automatizados procedentes de iteraciones previas y actuales para seleccionar casos de prueba que puedan aspirar a formar parte del juego de pruebas de regresión, y retirar los casos de prueba que ya no son relevantes. Los casos de prueba de iteraciones anteriores que verificaban prestaciones muy específicas pueden tener escaso valor en iteraciones posteriores debido a cambios en las prestaciones o a nuevas prestaciones que alteran la forma en la que se comportan las prestaciones anteriores.

Mientras revisan los casos de prueba, los probadores deben tener en cuenta su idoneidad para la automatización. El equipo tiene que automatizar el máximo número de pruebas de iteraciones anteriores y actuales. Esto permite que las pruebas de regresión automatizadas reduzcan el riesgo de regresión con menos esfuerzo que con las pruebas de regresión manuales. Este menor esfuerzo de pruebas de regresión libera a los probadores para que prueben nuevas funcionalidades y funciones de forma más exhaustiva en la iteración actual.

Es fundamental que los probadores tengan la capacidad de identificar rápidamente y actualizar casos de prueba de iteraciones y/o entregas anteriores que se ven afectadas por los cambios realizados en la iteración actual. La definición de cómo el equipo diseña, escribe y guarda los casos de prueba debería realizarse durante la planificación de la entrega. Las buenas prácticas de diseño de pruebas e implementación de pruebas deben adoptarse de forma temprana y aplicarse de manera consistente. El menor tiempo para pruebas y el cambio constante en cada iteración aumentará el impacto de utilizar malas prácticas de diseño e implementación de pruebas.

El uso de la automatización de pruebas, en todos los niveles de prueba, permite a los equipos ágiles ofrecer feedback rápido sobre la calidad del producto. Unas pruebas automatizadas

bien escritas constituyen un documento vivo de la funcionalidad del sistema [Crispin08]. Mediante la comprobación de las pruebas automatizadas y sus correspondientes resultados de pruebas en el sistema de gestión de la configuración, alineados con el versionado de los builds del producto, los equipos ágiles pueden revisar la funcionalidad probada y los resultados de las pruebas para cualquier versión dada en cualquier momento.

Las pruebas unitarias automatizadas se ejecutan antes de que el código fuente se registre en el sistema de gestión de la configuración para garantizar que los cambios del código no rompen la versión del software. Para reducir las roturas de la versión, que pueden ralentizar el progreso de todo el equipo, no debe hacerse el check-in del código a menos que se hayan superado todas las pruebas unitarias automatizadas. Los resultados de las pruebas unitarias automatizadas ofrecen feedback inmediato sobre la calidad del código y de la versión, pero no sobre la calidad del producto.

Las pruebas de aceptación automatizadas se ejecutan periódicamente como parte de la integración continua de la construcción (build) de todo el Sistema. Estas pruebas se ejecutan contra una versión (build) del sistema completo, pero normalmente no se ejecutan en cada check-in de código ya que tardan más tiempo en ejecutarse que las pruebas unitarias automatizadas y podrían ralentizar los check-in de código. Los resultados de prueba procedentes de pruebas de aceptación automatizadas proporcionan feedback sobre la calidad del producto en relación con la regresión desde la última versión, pero no establecen un estado de la calidad general del producto.

Las pruebas automatizadas pueden ejecutarse de manera continua contra el sistema. Inmediatamente después de desplegar una nueva versión (build) en el entorno de pruebas, debe crearse un subconjunto de pruebas automatizadas para cubrir los puntos críticos de funcionalidad e integración del sistema. Estas pruebas se conocen como pruebas de verificación de la versión (build). Los resultados de las pruebas de verificación de la versión proporcionarán un feedback instantáneo sobre el software después del despliegue, de manera que los equipos no pierden tiempo probando una versión inestable.

Las pruebas automatizadas incluidas en el conjunto de pruebas de regresión normalmente se ejecutan como parte de la construcción principal diaria en el entorno de integración continua, y de nuevo cuando se despliegan una nueva versión en el entorno de pruebas. En cuanto una prueba de regresión automatizada falla, el equipo se detiene e investiga los motivos de la prueba fallida. La prueba puede haber fallado por cambios funcionales legítimos en la iteración actual, en cuyo caso es posible que deba actualizarse la prueba y/o la historia de usuario para reflejar los nuevos criterios de aceptación. De forma alternativa, es posible que la prueba deba retirarse si se ha construido otra prueba para cubrir los cambios. Sin embargo, si la prueba ha fallado por un defecto, es una buena práctica que el equipo arregle el defecto antes de avanzar con nuevas prestaciones.

Además de la automatización de las pruebas, también podrán automatizarse las siguientes tareas:

- Generación de datos de prueba
- Carga de datos de prueba en los sistemas
- Despliegue de las versiones construidas en los entornos de pruebas
- Restauración de un entorno de pruebas (por ejemplo, los ficheros de datos de base de datos o página web) a una versión anterior
- Comparación de los datos de salida

La automatización de estas tareas reduce los costes y permite al equipo dedicar tiempo a desarrollar y probar nuevas prestaciones.

3. Función y habilidades de un probador en un equipo ágil

En un equipo ágil, los probadores deben trabajar en estrecha colaboración con los demás miembros del equipo y con las partes interesadas del negocio. Esto tiene una serie de implicaciones en términos de las habilidades que un probador debe tener y las actividades que llevan a cabo dentro de un equipo ágil.

3.1 Habilidades de los probadores ágiles

Los probadores ágiles deben tener todas las habilidades citadas en el plan de estudios de Nivel Básico [ISTQB_FL_SYL]. Además de estas habilidades, un probador en un equipo ágil debe ser competente en la automatización de pruebas, el desarrollo guiado por pruebas, el desarrollo guiado por pruebas de aceptación, en pruebas de caja blanca, de caja negra y pruebas basadas en la experiencia.

Dado que las metodologías ágiles dependen mucho de la colaboración, la comunicación y la interacción tanto entre los miembros del equipo como también con las partes interesadas fuera del equipo, los probadores en un equipo ágil deben disponer de buenas habilidades interpersonales. Los probadores en equipos ágiles deben:

- Ser positivos y estar orientados a soluciones con los miembros del equipo y otras partes interesadas
- Demostrar un pensamiento crítico, orientado hacia la calidad y escéptico sobre el producto
- Adquirir activamente información de las partes interesadas (en lugar de confiar íntegramente en las especificaciones escritas)
- Evaluar y reportar con exactitud los resultados de las pruebas, el progreso de las pruebas y la calidad del producto.
- Trabajar junto con los representantes de negocio y las partes interesadas de forma eficaz para definir historias de usuario testeables y especialmente criterios de aceptación,
- Colaborar con el equipo, trabajar por pares con programadores y otros miembros del equipo
- Responder rápidamente al cambio, cambiando, añadiendo o mejorando los casos de prueba
- Planificar y organizar su propio trabajo

El desarrollo continuo de habilidades, incluyendo el desarrollo de las habilidades interpersonales, es fundamental para todos los probadores, incluyendo los de equipos ágiles.

3.2 Funciones de un probador en un equipo ágil

Las funciones de un probador en un equipo ágil incluyen actividades que generan y proporcionan feedback no sólo sobre el estado de las pruebas, el progreso de las pruebas y la calidad del producto, sino también sobre la calidad del proceso. Además de las actividades descritas en otras secciones de este plan de estudios, estas actividades incluyen:

- Conocer, implementar y actualizar la estrategia de pruebas
- Medir y reportar la cobertura de pruebas en todas las dimensiones de cobertura aplicables
- Asegurar el uso adecuado de las herramientas de pruebas
- Configurar, usar y gestionar entornos de pruebas y datos de pruebas
- Reportar defectos y trabajar con el equipo para solucionarlos
- Orientar a otros miembros del equipo en cuanto a aspectos relevantes de las pruebas.
- Asegurar que se planifiquen las tareas de pruebas adecuadas durante la planificación de la entrega y de la iteración
- Colaborar activamente con los desarrolladores y partes interesadas del negocio para aclarar los requisitos, especialmente en términos de facilidad de prueba, consistencia e integridad
- Participar proactivamente en las reuniones retrospectivas de equipo, sugiriendo e implementando mejoras

Dentro de un equipo ágil, cada miembro del equipo es responsable de la calidad del producto y desempeña una función en la ejecución de tareas relacionadas con las pruebas.
Las organizaciones ágiles pueden enfrentarse a algunos riesgos organizativos relacionados con las pruebas:

- Los probadores trabajan tan cerca de los desarrolladores que pierden la mentalidad apropiada de probador
- Los probadores se vuelven tolerantes o silencian prácticas ineficaces, ineficientes o de baja calidad dentro del equipo
- Los probadores no pueden llevar el ritmo de los cambios entrantes en iteraciones limitadas por tiempo

Para mitigar estos riesgos, las organizaciones podrían considerar variaciones para preservar la independencia citada en etapas anteriores.