

La implementación de Scrum en organizaciones organizaciones geográficamente distribuidas: El caso de JIRA

*The implementation of Scrum in organizations geographically distributed organizations:
The case of JIRA*

Dahiana Mendoza
Universidad
Latinoamericana de la
Ciencia y Tecnología
dmendezap588@ulacit.ed.c
I

Julio Córdoba
Universidad
Latinoamericana de la
Ciencia y Tecnología
[jcordoba022@ulacit.ed.](mailto:jcordoba022@ulacit.ed.cr)
cr

Gabriel Silva
Universidad
Latinoamericana de Ciencia
y Tecnología
gsilvaa468@ulacit.ed.cr

Fecha de recibido: 05 de marzo 2020

Fecha de aprobado: 25 de marzo de 2020

Resumen- En la ingeniería de sistemas existen múltiples metodologías para el desarrollo de software, las cuales han evolucionado buscando mayor innovación y efectividad. En este sentido, el éxito de todo proyecto depende de diversos factores, sin embargo, las técnicas utilizadas para su ejecución son determinantes para alcanzar los resultados deseados. Las empresas tienen estructuras de equipo diversas, distribuidas geográficamente y con múltiples retos que afrontar, por lo cual los gerentes de proyectos deben hacer uso de herramientas tecnológicas para lograr

cumplir con los objetivos, satisfaciendo las necesidades del negocio. No obstante, estas herramientas nunca podrán estar por encima del equipo de trabajo y su entorno.

Palabras claves: Scrum, Jira, metodología ágil, equipos de desarrollo, ambientes distribuidos.

Abstract- Multiple methodologies exist in the software engineering field, these methodologies have evolved seeking innovation and effectiveness, the project success depends on different aspects, nevertheless, the techniques used in the

project execution will drive its success. The enterprises have diverse team structures, they are geographically distributed, and they face multiple challenges, the project managers should use technology tools in order to achieve the business objectives and needs, but these tools will never been prioritized over the team and their interactions.

Keywords: Agile, Jira, Scrum, development teams.

I. INTRODUCCIÓN

Los equipos de desarrollo de software distribuidos geográficamente deben lograr los objetivos sin importar los múltiples desafíos que la naturaleza virtual posee [1], pues el trabajar en un modo remoto abre portillos hacia problemas de comunicación, desmotivación y falta de sentimiento de pertenencia. El equipo de desarrollo tiene a sus miembros en ubicaciones geográficas diferentes, por lo tanto las zonas horarias, la cultura y el idioma se unen para crear mayores retos. El concepto de colocación de la filosofía ágil, el cual sugiere que el equipo de desarrollo se sitúe en una misma ubicación física,

sencillamente es imposible de implementar [2].

En ágil las personas y sus interacciones están primero, aún sobre los procesos y las herramientas [3], sin embargo, muchas veces las herramientas se ponen en contra del equipo al convertirse estas en la prioridad. De modo que, actitudes tales como el querer mantener el Kanban actualizado antes que fomentar la buena interacción entre desarrolladores pueden llevar a los equipos a sucumbir ante la frialdad del mundo virtual.

Al no contar con los miembros disponibles en el mismo lugar y en el mismo momento, el gerente de proyecto o ScrumMaster debe utilizar diferentes herramientas y estrategias, pues al contar con un equipo de desarrollo con miembros en diferentes regiones geográficas representa un inconveniente al tener que esperar horas e incluso días antes de recibir respuestas para seguir avanzando. Además, al trabajar en modo local los entregables sencillamente se vuelven inmanejables. Las organizaciones modernas dependen de los sistemas

de software en diversas maneras. Al respecto, no importa el tamaño de la empresa, los problemas serán parte de las tareas diarias a superar [4]. En este sentido, JIRA se presenta como una solución al otorgar herramientas de colaboración, mantenimiento de código, administración de requerimientos, trazabilidad, organización del equipo de trabajo, manejo de cargas de trabajo, administración de problemas de aplicación, y un sistema robusto de reportería; todas estas capacidades de la herramienta se pueden desarrollar en la nube, por lo cual calza perfecto para equipos distribuidos en zonas geográficamente lejanas [5].

En un mundo globalizado cada vez es más común encontrar equipos de desarrollo de Scrum con grandes retos para fomentar escenarios que favorezcan su efectividad [6]. Por tal motivo, a lo largo de la investigación se estudiarán diferentes funcionalidades de la herramienta JIRA Software, las cuales son utilizadas por equipos de desarrollo para la implementación de SCRUM.

Dentro de las funciones que tratará la investigación se tienen: el seguimiento de proyectos e incidencias, tableros SCRUM y sus componentes, gestión de la reserva de requerimientos, planeamiento de la iteración utilizando JIRA, flujos de trabajo personalizables, sistema de reportería y filtros personalizables [7]. Estas capacidades de la herramienta se enfocarán en la habilitación de la nube; no obstante, queda excluido de la investigación el uso de funciones a nivel local.

Muchos de los miembros del equipo jamás tendrán un contacto cara a cara con sus compañeros, sin embargo, el unir las diferentes partes: el manifiesto ágil, los principios de Scrum y la herramienta Jira podrían hacer que los equipos logren sus objetivos exitosamente. Esta investigación dará como resultado una referencia introductoria para equipos virtuales ágiles en la búsqueda de una herramienta que les permita solventar sus necesidades para implementar SCRUM.

II. MARCO TEÓRICO

En el mundo de la ingeniería de software diferentes metodologías se han planteado con el fin de buscar las herramientas ideales. Al respecto, métodos tradicionales como la cascada descrito por Winston W. Royce trae una propuesta lineal y secuencial en donde cada etapa debe empezar hasta que la actual termine. Dicho método se adapta muy bien en proyectos pequeños, con tiempos de entrega cortos y requerimientos estáticos [8], sin embargo, en el mundo acelerado y complejo en que las compañías se desarrollan, surgen necesidades que deben ser solventadas con métodos novedosos, ágiles, sencillos de implementar y sobre todo cuyos resultados sean positivos. En el siguiente marco teórico se revisarán los conceptos relacionados con metodología ágil, Scrum y Jira.

No hay un desarrollo único, ya sea en tecnología o administración técnica, el cual en sí mismo prometa un orden de mejora en una década en productividad, en confiabilidad, en

simplicidad” [9]. Aún en el 2018 seguimos buscando.

En febrero del 2001, 17 expertos en el área del desarrollo de Software se reunieron en Snowbird, Utah, para definir métodos alternativos a los ya popularmente establecidos en el campo. De esta forma nace el famoso Manifiesto Ágil que involucra cuatro postulados y doce principios [3].

El desarrollo ágil es una metodología que involucra cuatro postulados [3]: “Individuos e interacciones sobre procesos y herramientas”, en ágil encontramos que es más importante y prioritario el equipo de desarrollo y cómo sus miembros colaboran entre sí. En este postulado se resalta la importancia de tener un equipo conformado por individuos cuyas habilidades son clave para el éxito del proyecto [10]. En un equipo ágil no se busca tener estrellas en el equipo, lo que se busca es tener un equipo estrella. El segundo postulado: “Software funcionando sobre documentación extensiva.” evita que el equipo de desarrollo invierta esfuerzos exagerados en documentar cada parte y funcionalidad del

sistema, por lo contrario, el equipo de desarrollo se centra en darle al negocio el máximo valor agregado en el periodo más corto posible. En ágil se abraza la documentación, pero no para crear un diagrama y almacenarlo en un repositorio empresarial empolvado [10]. “Colaboración con el cliente sobre negociación contractual”, este tercer postulado insta a los equipos de desarrollo a trabajar de la mano con el cliente en una búsqueda permanente por satisfacer sus necesidades y resolver sus requerimientos, en este trabajo arduo y constante el equipo de desarrollo no se enfoca por hacer valer cláusulas de contratos, su objetivo es liberar en producción productos utilizables que ayuden a la organización con sus objetivos. Ya sea que la negociación se haga con una carta de constitución del proyecto o un contrato legal, el tiempo invertido en negociaciones es sencillamente ineficiente [10]. El último postulado: “Respuesta ante el cambio sobre seguir un plan”, presenta una acción práctica y exitosa; en ágil se planea, pero se reconocen las limitantes de la

planeación en ambientes turbulentos. Las compañías tienen ambientes ambiguos y cambiantes, seguir un plan escrupulosamente traerá sus consecuencias negativas si no hay una reacción al cambio adecuada [10].

Pues bien, estos cuatro postulados son respaldados por 12 principios primordiales [11], de los cuales se destacan: “Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor”; este principio coloca las necesidades del cliente en la cumbre de prioridades, enfocando al equipo de desarrollo en la rápida liberación de entregables como objetivo. “Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible”; este principio introduce el concepto de iteraciones que terminan en incrementos utilizables para el cliente. “Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto”; el concepto de colocación en la metodología ágil es un principio. En un escenario ideal

el equipo debería de estar ubicado en el mismo lugar, con esto los tiempos de respuesta se minimizarían y el trabajo en equipo se daría de una forma más natural. El último principio, pero de suma importancia, es: “A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia”. En ágil la búsqueda por la mejora continua es parte del trabajo, el equipo reconoce la oportunidad de buscar el estado ideal y reinventarse cuantas veces sea necesario.

Tanto el manifiesto ágil como sus principios se enfocan en la importancia de tener el software siempre en un estado construible y utilizable, trabajar directamente con los involucrados y ser por defecto flexibles al cambio [12].

Los creadores de Scrum, Ken Schwaber y Jeff Sutherland, definieron en la guía los roles, eventos, artefactos y reglas que hacen que todos estos componentes se unan entre sí [13]. El manifiesto ágil no especifica cómo llevar un proyecto, en

cambio ofrece guías generales para el desarrollo de software y así mismo permite agilizar la toma de decisiones en el equipo [12].

Creado en 1990, Scrum es un marco de gestión [13] que permite la entrega incremental de productos. Utiliza equipos pequeños que manejan partes independientes del proyecto, pero estos son multifuncionales, autoorganizados y de aproximadamente siete personas. Sus roles, ceremonias, reglas y artefactos se deben de seguir si realmente se está implementando Scrum. Es muy diferente hacer Scrum a pretender hacer scrum [14].

En un estudio realizado por VersionOne (2014), en donde se involucraron más de 3501 participantes, se encontró que Scrum o alguna de sus variantes, son las metodologías ágiles más utilizadas para la elaboración de productos de software en el mundo (73 %) [15].

Ahora bien, para entender Scrum es de suma importancia entender lo siguiente: tres roles están definidos dentro de Scrum; El dueño del

producto (única persona responsable de maximizar el retorno de inversión del proyecto); Equipo de Desarrollo de Scrum (Multifuncional, autogestionado, autónomo); ScrumMaster (Facilitador del proceso, remueve impedimentos, líder, promueve el ambiente adecuado). Todos ellos conforman el equipo de scrum [13].

El dueño del producto es el responsable de maximizar el valor del producto creado por el equipo de desarrollo. La forma en que esto se implementa puede variar de una organización a otra. Además, el dueño del producto es el único encargado de administrar el registro del producto (Product Backlog), en otras palabras, debe de expresar claramente los ítems de la cola de trabajo y ordenarlo basado en prioridades con el fin de lograr las metas de la organización. También, se debe de encargar de guiar al equipo de desarrollo para que los productos trabajados sean los que más agregan valor al negocio; así mismo debe de asegurarse que la cola de trabajo este accesible para todo el equipo de scrum; y finalmente, dentro

de sus responsabilidades, está asegurarse que el equipo de desarrollo entiende la cola de trabajo al nivel en que cada miembro necesita [13].

El equipo de desarrollo son todos los profesionales encargados de entregar productos utilizables al finalizar cada iteración. Además, solamente los miembros del equipo de desarrollo entregarán incrementos del producto en desarrollo. Es un equipo estructurado y empoderado por la organización, son autoorganizados, son multifuncionales y poseen todas las habilidades necesarias para crear incrementos de productos. En scrum no hay títulos o roles definidos para cada miembro, realmente no importa lo que cada persona ejecuta.

El Maestro de Scrum (ScrumMaster) es el responsable de promover y apoyar la ejecución de Scrum tal como la guía lo ha establecido. El ScrumMaster se encarga de ayudar a todos a entender la teoría de Scrum, prácticas, reglas y valores, si bien es un líder del equipo de Scrum, es un servidor del mismo. Es encargado de hacer que las interacciones del equipo

de desarrollo con otros interesados resulten en interacciones que ayuden y agreguen valor.

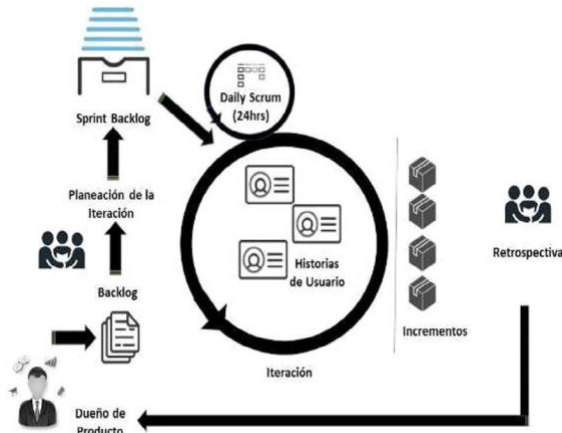


Ilustración 1. Scrum combina todas las actividades de desarrollo en cada iteración, adaptándose a las realidades emergentes, en intervalos fijos

Fuente: Elaboración propia.

En Scrum se han definido cuatro eventos o ceremonias, pero es de suma importancia entender primero que el corazón de Scrum es la iteración [13].

La iteración es un periodo con una duración de menos de un mes donde un incremento del producto final es entregado funcionando y utilizable; una nueva iteración inicia justo cuando la actual termina.

En la ilustración 1 se muestran gráficamente las ceremonias en Scrum:

Planeamiento de la iteración (Sprint Planning): Se ejecuta al inicio de la iteración, tiene una duración de 8 horas para una iteración de 1 mes, si la iteración es más corta así mismo la ceremonia de planeamiento lo será. Dos preguntas deben ser respondidas con esta primera ceremonia: ¿Qué puede ser entregado con el incremento creado en la siguiente iteración? ¿Cómo se logrará el trabajo necesario para lograr el incremento? [13].

Reunión diaria de scrum (Daily Scrum): Son cajas de tiempo de 15 minutos, se celebra diariamente y su principal objetivo es responder tres interrogantes: ¿Qué hice ayer que ayudó al equipo de desarrollo a cumplir la meta de la iteración? ¿Qué haré hoy para ayudar al equipo de desarrollo a cumplir la meta de la iteración?

¿Veo algún impedimento que no me permite ayudar al equipo de desarrollo a lograr la meta de la iteración?

Después de cada Daily Scrum, se reúnen los miembros necesarios para tener una conversación más detallada, adaptarse o replantear las tareas. Quien dirige la ceremonia es el equipo de desarrollo, el ScrumMaster vela porque el equipo de desarrollo mantenga la ceremonia dentro de los 15 minutos y con el formato establecido [13].

Revisión de la iteración (Sprint Review): Se celebra al final de la iteración y su objetivo es revisar el incremento entregado, y de ser necesario, adaptar la cola de trabajo para alinear al equipo con las prioridades del negocio. En esta ceremonia está todo el equipo de Scrum [13]. Retrospectiva de la iteración (Sprint Retrospective): Es una ceremonia de inspección, todo el equipo de scrum participa y juntos crean un plan de mejoras para ser aplicadas en la siguiente iteración, este evento debe ocurrir antes de la siguiente ceremonia de planeamiento y después de la ceremonia de revisión de la iteración actual [13].

Como se mencionó anteriormente se necesitan artefactos, también

conocidas como herramientas para desarrollar un adecuado Scrum, entre ellos se tiene el registro del producto (product backlog), lo cual es un registro de funcionalidades o características registradas por el dueño del producto (product owner). Este registro está ordenado por prioridades que deben de ir alineadas a las necesidades del negocio. Una vez que inicia la iteración se crea el registro de iteración (sprint backlog), el cual refleja la meta a cumplir en cada iteración y además muestra una vista más detallada del trabajo a realizar [16].

Un artefacto realmente importante es la definición de terminado (Definition of Done), pues aquí se detallan las tareas necesarias para crear un incremento al final de la iteración. Este es un artefacto vivo, ya que constantemente mejora gracias a los aportes del equipo de scrum [16].

Las historias de usuario son uno de los artefactos primarios de Scrum; una historia de usuario es una definición de muy alto nivel de un requerimiento, contiene solo la información necesaria para que los desarrolladores puedan

estimar el esfuerzo para poder realizarla. Es importante resaltar que es el dueño del producto el que escribe las historias del usuario, no los desarrolladores y que no reflejan requerimientos funcionales. Las historias de usuario forman parte del backlog y serán acomodadas de acuerdo con prioridades. De la mano de las historias de usuario se encuentran las Historias Épicas (epics), sencillamente son historias de usuario más grandes, más difíciles de implementar y estas no podrán ser resueltas en una sola iteración y por lo tanto necesitan ser segregadas en historias de usuario más pequeñas [17].

Los puntos de historia (story points) son estimaciones del esfuerzo, las cuales son influenciadas por el total de trabajo, complejidad, riesgo e incertidumbre; son una unidad de medida otorgada a cada parte del backlog. Realmente no interesan los números, un equipo puede estimar sus historias con 1,2,3, otro equipo puede estimar basado en 100, 200, 300 y un tercer equipo lo puede hacer en 1 millón, 2 millones, 3 millones, lo

que importa es la proporción de las estimaciones y que estos tres aspectos sean considerados: la cantidad de trabajo, la complejidad del trabajo y el riesgo e incertidumbre al ejecutar las tareas [18].

Velocidad, este concepto se refiere a la cantidad de trabajo que un Scrum Team puede completar en un determinado sprint, esta métrica es calculada al final de la iteración al sumar la totalidad de puntos de cada historia de usuario [19].

Scrum es la forma más popular de introducir la filosofía ágil en las organizaciones y esto es debido a su simplicidad y flexibilidad [12]. Además, es de vital importancia que el equipo siga las ceremonias de Scrum, la retrospectiva es una excelente oportunidad para la mejora continua.

Atlassian es una empresa fundada en el 2002, originaria de Australia, su mercado de negocio es la creación de aplicaciones para empresas desarrolladoras de software [5]. En la última encuesta de popularidad realizada en mayo del 2018, en donde 93 herramientas de control de

defectos fueron analizadas, Jira ocupó el primer lugar [20], posicionándose como líder sobre otras herramientas tales como Primavera, Zoho o Workfront.

Según la empresa Atlassian, Jira es utilizado no solamente para administración de proyectos, sino también para el control de defectos por más de 75.000 clientes en más de 122 países alrededor del mundo. A continuación, se citarán algunos casos de éxito al utilizar Jira.

DHI Group es una empresa de medios digitales que opera Dice.com, un sitio web para profesionales en tecnología, además opera muchísimos sitios web más alrededor del globo. Dice.com posee más de 80.000 ofertas de empleo, más de 3 millones de profesionales registrados y alrededor de 2.4 millones de usuarios diferentes por mes [21]. El reto: DHI se expandió rápidamente, en gran parte debido a adquisiciones, lo cual ayudó a tener presencia con más de 10 marcas importantes. Sin embargo, con esto también se heredaron diversas combinaciones de herramientas de colaboración y comunicación, así

como múltiples plataformas que imposibilitaron que los empleados pudieran laborar fluidamente entre equipos y departamentos; los grandes afectados: eficiencia e innovación. Si este escenario no era lo bastante complicado, se sumó la desactualización desmedida de los procesos de gobernanza y desarrollo de las Tecnologías de Información de DHI, sus departamentos de soporte al cliente todavía utilizaban sistemas físicos en papel para darle seguimiento a los casos, esto imposibilitaba hacer búsquedas automatizadas, compartir los incidentes y toda la información importante quedaba expuesta en un medio tan arcaico. La solución: DHI gradualmente se fue migrando a Atlassian como su única plataforma de trabajo, tanto la comunicación, eficiencia, colaboración y trabajo en equipo mejoraron drásticamente. Muy pronto equipos, departamentos y geografías estaban trabajando con resultados de alto rendimiento, los equipos de desarrollo de software utilizaban Jira y los equipos de soporte rápidamente migraron de Service Now a Jira Service Desk. Estas

migraciones le dieron grandes ahorros a DIH en término de costos de licenciamiento, pero más importante aún es que una cultura de apertura se implantó en la empresa, ahora la colaboración pasa naturalmente, lo cual permite que los equipos sean más productivos [21]. “Tener equipos más fuertes y más colaborativos con diferentes grupos nos ayuda a liderar nuevas ideas de productos de una manera más rápida hacia el mercado Atlassian nos ayudó a conseguir esto”, palabras de Brian Hostetter director de arquitectura del grupo DHI.

Otro caso de éxito es la corporación multinacional de alta tecnología Citrix, fundada en 1989, la cual provee servicios de virtualización de servidores, conexión de red, software como servicio e informática en la nube [22]. GetGo Communications, una división de Citrix, es la encargada de proveer los servicios de telefonía y mensajería tanto locales como larga distancia en los mercados de los Estados Unidos. El reto: cuando un incidente se levantaba en GetGo tres equipos se involucraban en la solución: Cuidado del Cliente

(customer care), Operaciones Técnicas (technical operations) e Ingeniería (engineering), desafortunadamente los equipos tenían sistemas y herramientas de comunicación dispares y engorrosas que hacían que minutos valiosos se desperdiciaran al tener que ir y venir mientras se resolvían los incidentes. La solución: GetGo Technical Operations y Customer Care implementaron Atlassian para su sistema de alertas, monitoreo y sistema de comunicaciones; Engineering implementó Jira, lo cual les permitió mejorar los lanzamientos a producción, comunicación y colaboración. “Nuestra tasa de resolución de incidentes ha aumentado más de un 13% desde que empezamos a utilizar Atlassian”, palabras de Doug Squires, Senior Manager, Customer Care de GetGo [23].

Jira es la herramienta de desarrollo de software líder de los equipos ágiles, los mejores equipos de software entregan pronto y frecuentemente [7]. Con Jira se puede planificar el proyecto a través de la creación de

historias de usuario e incidencias, la planificación de las iteraciones se realiza fácilmente y el equipo de desarrollo logra distribuir las tareas de una forma ágil.

El ScrumMaster puede supervisar el trabajo del equipo aterrizado a un contexto en específico y con una visión de principio a fin. Cuando llega el tan anhelado momento del lanzamiento, Jira ofrece capacidades en su sistema para que el ScrumMaster siempre esté seguro que la información que utiliza es la más actualizada y confiable.

De acuerdo con el proyecto y equipo de trabajo se necesita personalizar los flujos de trabajo; al respecto, Jira ofrece flujos de trabajo predefinidos con el fin de agilizar su utilización, sin embargo, se puede personalizar dicho flujo para solventar las necesidades del trabajo.

III. METODOLOGÍA

La metodología utilizada en esta investigación será de tipo cualitativa y explicativa [24] en donde se busca crear una guía de implementación de

Scrum a través de Jira para equipos de desarrollo de software distribuidos geográficamente. Se va a utilizar literatura existente con el fin de identificar qué características de Jira implementan Scrum.

Más adelante, para el análisis de los datos se contará con casos de éxito documentados y literatura técnica extraída de fuentes oficiales. Se utilizarán diferentes recursos de referencia, entre ellos libros, artículos de investigación, videos y sitios en la web. En la ingeniería de software el desarrollo de sistemas, en conjunto con la gestión del proyecto, son partes fundamentales y esta investigación presente introducir a la audiencia en conceptos primordiales sobre cómo utilizar Jira con Scrum en sus organizaciones.

IV. ANÁLISIS Y RESULTADOS

Las empresas tienen diversas herramientas para migrar su negocio a Scrum, sin embargo, Atlassian provee una suite completa para implementar este marco de trabajo y su adopción está en constante aumento. En el último estudio realizado en mayo del

2018 sobre herramientas de administración de proyectos, Jira ocupa el segundo lugar en popularidad, perdiendo solamente ante MS Project; más de 229 herramientas fueron analizadas [25].

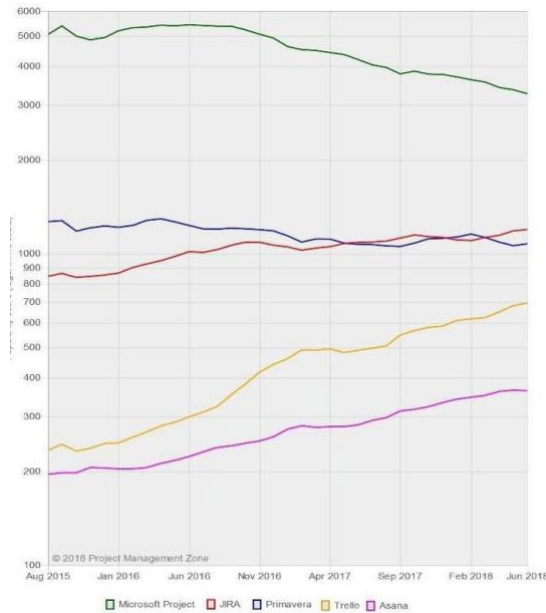


Ilustración 2. Tendencia en popularidad de herramientas de planeación de proyectos [25]

Atlassian ofrece dos opciones para trabajar con Jira Software: la versión en la nube y el servidor local.

Pues bien, para equipos distribuidos geográficamente la versión en la nube es la ideal por las siguientes razones: la empresa puede despreocuparse de comprar y mantener su propia

infraestructura, en la nube la configuración e instalación corre por Atlassian, la suscripción incluye actualizaciones de software, soporte técnico, costos de hosting, ancho de banda y sistema de administración. Además, agregar y quitar aplicaciones es fácil de realizar, los equipos pueden activar una aplicación, probarla y volverla a apagar si el equipo aún no está listo para usarla, realmente es un pago por uso (pay per use). Si el equipo utiliza aplicaciones de Google estas pueden ser integradas de forma sencilla. La nube es segura, todos los sitios de Atlassian usan SSL por defecto y están alojados en instalaciones seguras. La nube es confiable, Atlassian ofrece un 99,9 % de uptime con un modelo de soporte 24x7 [26]. El precio de la nube es asequible, en la siguiente figura se muestran los costos:

Número de usuarios	Tarifa	Costo
1-10 Usuarios	Mensual	10 USD tarifa plena
11-100 Usuarios	Mensual	7 USD por usuario
100-200 Usuarios	Mensual	5.50 USD por usuario
200-500 Usuarios	Mensual	3.10 USD por usuario

Ilustración 3. Jira Software Precios [27]

Al iniciar a trabajar con Jira Software es importante realizar cinco tareas primordiales que llevan a la práctica la teoría de Scrum:

Configurar los miembros del equipo en la herramienta:

Como se explicó en el marco teórico, la guía de Scrum establece un equipo de Scrum, el cual está conformado por el ScrumMaster, El dueño del Producto y el Equipo de Desarrollo [16], todos estos involucrados deberán ser agregados por el administrador de Jira, se les deberá otorgar los niveles de permiso adecuados basados en tres aspectos: los roles que desempeñarán en el equipo, las tareas a ejecutar en el proyecto y el nivel de permisos que los miembros necesitan para ejecutar sus labores en el proyecto. Básicamente no existe un método fijo para configurar los permisos, lo más importante es que el Scrum Team sea capaz de ejecutar las acciones necesarias [7].

Planificación del trabajo para el equipo: El backlog, artefacto de Scrum se implementa en Jira, en él se mostrarán las historias de usuario

creadas por el dueño del producto, así como los defectos de la aplicación en desarrollo. Toda la gestión del backlog se realiza desde Jira, fácilmente se pueden crear grupos de historias de usuario para las iteraciones y así reflejar el sprint backlog. En Jira también se implementa el concepto de Epics, todas las funcionalidades del backlog requerirán permisos especiales otorgados a los miembros para poder ser ejecutadas. El backlog de Jira es sumamente robusto y permite además la estimación de las historias de usuario y epics, la estimación será basada en Story Points. Otra característica presente en Jira es la de crear subtareas, esta funcionalidad es bastante favorable, ya que permite un nivel de monitoreo más preciso al dar visibilidad de seguir el progreso de las historias de usuario y sus subtareas.

Al ejecutar el evento del Sprint Planning el equipo de Scrum podrá utilizar Jira para correr la ceremonia, es tan fácil de realizar que lo único que tendrán que hacer es clic en la opción de crear una nueva iteración (create sprint) y arrastrar las historias de

usuario que serán trabajadas en dicha iteración [7]. Es importante recalcar que durante esta ceremonia el dueño del producto deberá de haber creado con anterioridad las historias de usuario en el backlog, las cuales estarán ordenadas por prioridades y únicamente durante la planeación. El equipo de desarrollo se enfocará en analizar y entender el trabajo requerido, así como estimar y delimitar el alcance para la iteración. Una vez que la planificación de la iteración está lista el equipo dará clic en Iniciar Iteración (start sprint) y listo, las historias serán automáticamente movidas a la iteración activa (active sprint) [7].

Organizar el trabajo en tableros: Otra de las poderosas funcionalidades de Jira son los tableros (boards), aquí el equipo de Scrum podrá rastrear el trabajo que tienen asignado de una forma visual. Los tableros de Jira son altamente personalizables, desde los flujos de trabajo, colores y filtros. Es de vital importancia que el equipo de Scrum analice cuál es el tablero que mejor se ajusta a sus necesidades. En Jira los tableros de Scrum poseen su

propio backlog. Únicamente el administrador del proyecto y el administrador del tablero podrán configurar las propiedades del mismo. Cada tablero será parte de un proyecto y esto se deberá especificar en la propiedad ubicación (location). Los tableros pueden ser renombrados en cualquier momento, además, de ser necesario, podrán ser borrados. Es importante recalcar que los tableros se pueden borrar, pero las historias de usuario permanecerán intactas, esto es porque los tableros son una vista de las historias o defectos, no son los user stories o bugs per se [7].

Controlar el progreso del trabajo: Jira Software ofrece una amplia gama de reportes que permiten conocer el estado y progreso del trabajo. Con el gráfico de velocidad los equipos de Scrum podrán conocer qué tan precisas son sus estimaciones y qué tanto están logrando completar en cada iteración. Este reporte será utilizado durante las revisiones de la iteración, lo cual promoverá oportunidades de mejorar en las estimaciones futuras [7].

En la Ilustración 4 se puede apreciar cómo el gráfico de velocidad permite ver la cantidad de story points comprometidos versus los story points que realmente se pudieron completar [7].

El gráfico de velocidad permite que el equipo de Scrum determine si está comprometido más allá de su capacidad, o por el contrario no está llegando a su capacidad. Al mostrar puntos atípicos (outliers) el equipo podrá identificar los errores cometidos, por ejemplo, al sobreestimar una historia de usuario al convertirla en una historia épica. Con este gráfico los equipos de Scrum podrán determinar su velocity con las primeras iteraciones, pues al iniciar a colaborar los miembros de un equipo nuevo difícilmente podrían determinar de una forma certera su velocidad; todos los miembros son nuevos. Sin embargo, con este gráfico podrán ver datos históricos que les permitirá establecer esta métrica como equipo.

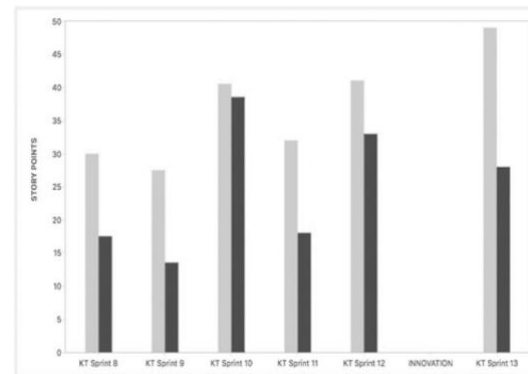


Ilustración 4. Gráfico de Velocidad en Jira Fuente: [28]

Los gráficos de trabajo pendiente (burdown charts) son de gran utilidad para los líderes de equipos, los cuales pasan la mitad del tiempo pensando en el trabajo venidero y la otra mitad la pasan trabajando en la iteración actual. Los líderes del equipo necesitan saber cómo van marchando las cosas, el burndown chart es de referencia diaria, durante los Daily Scrum, el gráfico puede ser utilizado, permitiendo que el equipo pueda concentrarse en seguir avanzando a la meta final. Con este gráfico el equipo podrá predecir la probabilidad de poder cumplir con todo el trabajo acordado, es una gran herramienta para evitar el famoso scope creep en los proyectos. La disciplina es vital por parte del equipo de desarrollo.

Diariamente los integrantes deberán de actualizar las historias de usuario para mantener datos actualizados en el gráfico. Este gráfico es sumamente poderoso, es una fuente para que el equipo pueda tomar las acciones necesarias para corregir los errores que están poniendo en riesgo el proyecto.

La empresa global BlackRock, dedicada a la gestión de inversiones, administra más de 4.77 trillones de dólares, debido a su magnitud poseen grupos de dueños de producto, todos con necesidades diversas y constantes. En números, BlackRock es capaz de hacer liberaciones a producción más de 500 veces a la semana, su cartera al 2015 era de 65 clientes, 81 millones de líneas de código y un equipo técnico conformado por más de 4000 miembros [28].

En esta empresa tienen releases de bajo y alto riesgo y cada una de estas liberaciones debe ser aprobada por el equipo técnico, cuentan con múltiples ambientes, en general es un ambiente sumamente complejo. En la búsqueda de una herramienta holística y

después de tres intentos fallidos con otras herramientas, finalmente Jira fue la solución ideal. Se definieron cinco procesos clave que se debían cubrir: extensibilidad, la compañía necesitaba que todos los equipos tanto los ágiles como los que utilizaban cascada pudieran beneficiarse, una interface con los procesos existentes era necesaria. El segundo proceso era relaciones y dependencias, se buscaba obtener un solo release a través de múltiples deployments, teniendo la posibilidad de trazar de principio a fin y poder hacer rollbacks. Luego, visibilidad era el tercer proceso clave; al ser una empresa con clientes externos la transparencia hacia ellos es vital. La escalabilidad se definió como cuarto proceso clave, sencillamente si es necesario crecer necesitan que la herramienta lo pueda hacer. Finalmente, el quinto proceso definido fue el de control, Jira les permitió tener revisiones a diferentes niveles, las aprobaciones fueron implementadas fácilmente [29].

Para Release Management Jira logra resolver los cinco procesos clave con las siguientes funcionalidades:

Proceso clave	Jira Software
Extensibilidad	SDK, Librería Plugin, APIs
Relaciones y dependencias	Tareas, Subtareas, Enlaces
Visibilidad	Tableros, Filtros, Suscripciones, Notificaciones
Escalabilidad	Flexibilidad y escalabilidad para el volumen de historias de usuario, <i>issues</i> y usuarios
Control	Flujos de trabajo, tipos de <i>issues</i> , reportes, vistas

Ilustración 5. Gestión de Liberaciones en Jira Fuente: [23]

Finalmente, queda claramente comprobado que Jira es una herramienta ideal para la implementación de Scrum. Un claro ejemplo es la siguiente empresa, la cual con éxito logró transicionar a dicha herramienta:

Audi es una empresa alemana fabricante de vehículos de alta gama con presencia a nivel global. Esta empresa se distingue por tener la innovación en su corazón, sus cuatro anillos en el logo representan los principios de Vorsprung durch Technick que significa “avanzar a través de la tecnología”, desde autos híbridos hasta vehículos piloteados automáticamente, todos sus proyectos

de innovación se centran en el centro de investigación y desarrollo en Ingolstadt y Neckarsulm en Alemania [30].

Aproximadamente seis mil personas entre ellos ingenieros, diseñadores, desarrolladores de software y equipos de aseguramiento de calidad trabajan en los centros. Este personal se relaciona con proveedores a nivel mundial. El reto que se presenta es el de mantener a todos calibrados hasta el más mínimo detalle, con acceso instantáneo y fácil a la información actualizada y relevante, toda esta capacidad debe darse sin enlentecer el impulso del equipo.

En el 2007, el centro de investigación y desarrollo inició su transición a Jira, explotando primeramente sus características de seguimiento de problemas (issue tracker). Sin embargo, el éxito ha sido tal que actualmente es la herramienta encargada de recolectar miles de requerimientos de funcionalidades; equipos encargados de las pruebas de los automóviles han iniciado a gestionar los bugs de los programas de los carros durante las pruebas de

los vehículos. Mientras tanto otros equipos explotan otras características de la suite de Jira a través de Confluence en donde almacenan el conocimiento, protocolos de departamentos, minutas de reuniones y políticas. Audi ha logrado integrar sus sistemas antiguos tales como LDAP (Lightweight Directory Access Protocol) y IBM Application Middleware, todo esto de una forma ágil y fácil. El trabajo se ha simplificado tanto que los usuarios simplemente comunican sus requerimientos, se crea un proyecto, luego un tablero y listo, los clientes reciben lo que necesitan. El éxito ha sido tal que todos los proyectos se manejan con Jira, no importa si son carros, software para carros o software para el negocio. Florian Banzer y Martin Binder analistas de sistemas de Audi aseguran “Al soportar el proceso de desarrollo y sistemas con Jira y Confluence se obtiene un impacto positivo en cada uno de los automóviles que salen de nuestro centro de manufactura” [30].

V. CONCLUSIONES Y

RECOMENDACIONES

Ágil es sin duda alguna una herramienta valiosísima para los equipos de desarrollo. Los equipos han ido evolucionando hacia metodologías más innovadoras que les permitan lograr sus objetivos en el menor tiempo posible, potencializando al máximo el retorno de inversión para las organizaciones.

Con Scrum, los equipos tienen a mano un marco de trabajo que les permite no solamente agregar valor al negocio, sino que también por su naturaleza, los equipos logran llegar a un nivel óptimo de rendimiento. Si bien es cierto el concepto de colocación está intrínseco en ágil y Scrum, en un mundo globalizado los equipos virtuales geográficamente distribuidos pueden alcanzar el éxito y eliminar las barreras de cultura, tiempo y espacio.

Jira es la herramienta por elección de los equipos de desarrollo de software, [7] sus diversas capacidades hacen que implementar Scrum sea 100 % realizable y son muchas las empresas

que actualmente han visto sus indicadores de productividad beneficiados gracias a esta herramienta. Además, con Jira no solamente los desarrolladores ganan, cada día son más los negocios de diversas áreas que adoptan Jira para llevar sus proyectos.

Es importante rescatar que el recurso humano seguirá siendo imprescindible para el éxito de los proyectos. El ScrumMaster tiene a cargo una labor primordial al habilitar el ambiente de trabajo que permita que los miembros puedan interactuar y ejecutar sus tareas de la mejor manera [16].

Un gerente de proyectos que haga uso de herramientas como Jira, aunado a un equipo calificado y altamente motivado, podrá ejecutar con el más alto nivel de rendimiento. Sumado a ello, las organizaciones que cuenten con estos elementos podrán gozar de una ventaja competitiva clara y tangible.

VI. TRABAJOS FUTUROS

El presente trabajo se ha enfocado a nivel macro, desarrollando las

principales características y funcionalidades de Jira que implementan Scrum, sin embargo, para futuros trabajos se podría explorar más a fondo las múltiples compatibilidades que tiene Jira con otras herramientas de Atlassian. Existen herramientas para equipos de desarrollo que podrían ser utilizadas fácilmente con Jira; la gestión del conocimiento se podría llevar a cabo con Confluence, la integración continua a través de Bamboo, el manejo del código con Bitbucket y la colaboración en tiempo real con Stride. Estas son apenas algunas herramientas poderosas que podrían servir para futuras investigaciones.

VII. REFERENCIAS

- [1] J. G. J. S. S. A. A. E. A. S. Guzmán, «How to get mature global virtual teams: a framework to improve team process management in distributed software teams.,» Software Quality Journal, vol. 18, no. 4, pp. 409-435, December 2010.
- 2] G. Zilberfeld, «How to Make Collocation Work for You,» Agile Connection, 12 Mayo 2013. [En línea].

Available:
<https://www.agileconnection.com/article/how-make-collocation-work-you>.
[Último acceso: 5 Mayo 2018].

[3] M. B., A. v. B. A. C. W. C., M. F., G., J. H. A. H.

Kent Beck, «Manifiesto for Agile Software Development,» Agile Manifesto Org, 2001.[En línea].Available: <http://agilemanifesto.org/>. [Último acceso: 11 Mayo 2018].

[4] F.-C.B.A.B.I.-B.B. Marius-Constantin BRAD, «A Comparative Study of Agile Project Management Software Tools,» Economy Informatics, vol. vol XVI, n° 1, pp. 27-39, 2016.

[5] «Jira Software: Funcionalidades para el Desarrollo de Software,» Atlassian, 16 Mayo 2018. [En línea]. Available: <https://es.atlassian.com/software/jira/features>. [Último acceso: 18 Mayo 2018].

[6]J.Lim, «The challenges of scaling software development teams globally: pt. 1,» Atlassian, 1 Junio 2015. [En

línea]. Available:
<https://www.atlassian.com/blog/software-teams/challenges-scaling-software-development-teams-globally-part-1>.
[Último acceso: 5 Mayo 2018].

[7] Atlassian, «La herramienta de desarrollo de software líder de los equipos ágiles,» Atlassian, 16 Mayo 2018. [En línea]. Available: <https://es.atlassian.com/software/jira>. [Último acceso: 16 Mayo 2018].

[8] D. W. W. Royce, «Managing the Development of Large Software Systems,» IEEE Wescon, vol. 26, n° 8, pp. 1-9, 1970.

[9] F. P. B. Jr, «No Silver Bullet Essence and Accident in Software No Silver Bullet - Essence and Accident in Software,» Computer, pp. PP 10-19, Abril 1987.

[10] J. H. Martin Fowler, The Agile Manifesto, Martin Fowler.com, 2006.

[11] M.B.,A.v.B.A.C.W.C.,M.F.,G., J. H. A. H. Kent Beck, «Principios del Manifiesto Ágil,» Agile Manifiesto,[En línea]. Available:

<http://agilemanifesto.org/iso/es/princip>

les.html. [Último acceso: 19 Mayo 2018].

[12] S. B. F. S. A. B. D. S. M. W. Matthew B. Harris, «Nerd Herding: Practical Project Management in the Field,» de Proceedings of the World Congress on Engineering and Computer Science ,WCECS 2015, October 21-23, 2015, San Francisco, USA, 2015.

[13] J. S. Ken Schwaber, The Scrum Guide, Creative Commons, November 2017.

[14] S. P. M. Michael James (MJ), «Scrum Reference Card,» 19 Mayo 2018. [En línea]. Available:<http://scrumreferencecard.com/reference-card-de-scrum/>. [Último acceso: 20 Mayo 2018].

[15] J. R.-R. H. B. C. Christian Pérez-Salazar, El espíritu de Scrum en el desarrollo de software libre en el ámbito académico, México: Universidad Veracruzana, 2014.

[16] G. B. C. L. B. V. Pete Deemer, Scrum Primer Una introducción básica a la teoría y práctica de Scrum V2.0, Scrum Primer.org, 2012.

[17] S. W. Ambler, «User Stories: An Agile Introduction,» Agile Modeling, 2014. [En línea]. Available: <http://www.agilemodeling.com/artifacts/userStory.htm>. [Último acceso: 26 Mayo 2018].

[18] M. Cohn, «What are story point?» Mountain Goat Software, 23 Agosto 2016. [En línea]. Available: <https://www.mountaingoatsoftware.com/blog/what-are-story-points>. [Último acceso: 26 Mayo 2018].

[19] Scruminc, «Velocity,» 26 Mayo 2018. [En línea]. Available: <https://www.scruminc.com/velocity/>. [Último acceso: 26 Mayo 2018].

[20] P. M. zone, «Issue Management Tools-Popularity Ranking,» 22 Mayo 2018. [En línea]. Available:<https://project-management.zone/ranking/category/issue>. [Último acceso: 24 Mayo 2018].

[21] Atlassian, «For transparency across a global team, Dice.com chooses Atlassian,» Atlassian, 22 Mayo 2018. [En línea] Available: <https://www.atlassian.com/customers>. [Último acceso: 25 Mayo 2018].

- [22] Citrix, «Citrix,» 23 Mayo 2018. [En línea]. Available: <https://www.citrix.es/>. [Último acceso: 25 Mayo 2018].
- [23] D. Squires, «Combining Operations and Support: Incident Management with Atlassian - Atlassian Summit 2016,» Atlassian, 10-13 Octubre 2016. [En línea]. Available: <https://www.youtube.com/watch?v=MRcKTcDcJ7s>. [Último acceso: 25 Mayo 2018].
- [24] R. Hernández Sampieri, Metodología de la Investigación, Mexico: McGraw-Hill, 2001.
- [25] P. M. Zone, «Project Planning Tools - Popularity Ranking Trend,» 26 Mayo 2018. [En línea]. Available: <https://project-management.zone/ranking/category/trend/planning>. [Último acceso: 26 Mayo 2018].
- [26] Atlassian, «Pros and Cons of Cloud vs. Server,» 26 Mayo 2018. [En línea]. Available: <https://confluence.atlassian.com/cloudkb/pros-and-cons-of-cloud-vs-server-691011844.html>. [Último acceso: 26 Mayo 2018].
- [27] Atlassian, «Jira Software Pricing,» 26 Mayo 2018. [En línea]. Available: <https://www.atlassian.com/software/jira/pricing#pricing-calculator>. [Último acceso: 26 Mayo 2018].
- [28] J.-P. P. Gabriel Tiu, «Release Management with JIRA at BlackRock - Atlassian Summit 2015,» BlackRock, 4 Noviembre 2015. [En línea]. Available: <https://www.youtube.com/watch?v=Y80-lb4yjDE> [Último acceso: 5 Junio 2018].
- [29] Harvard University, «Release Management - Quick Reference Guide for users of Service Now,» Harvard University Information Technology. [En línea]. Available: https://huit.harvard.edu/files/huit/files/quick_ref_guide_v9-full_0.pdf, Noviembre 2013.
- [30] C. Nyce, «Case study: How Audi builds luxury vehicles with Jira,» Atlassian, 6 Mayo 2014. [En línea]. Available: <https://www.atlassian.com/blog/archives/case-study-audi-builds-luxury-vehicles-jira>. [Último acceso: 31 Mayo 2018].

[31] R. L. P. N. B. ,. I. B. ,. M. C.-D. D. S. D. ,. R. E. O. L. ,.O.P. ,. B. R. B. T. Janet D. Evans, «Managing Distributed Software Development in the Virtual Astronomical Observatory,» Harvard University, San Francisco, CA, USA, March 2012.

VIII. ILUSTRACIONES

Ilustración 1. Scrum combina todas las actividades de desarrollo en cada iteración, adaptándose a las realidades emergentes, en intervalos fijos.....3

Ilustración 2. Tendencia en popularidad de herramientas de planeación de proyectos [25].....5

Ilustración 3. Jira Software Precios [27].....5

Ilustración 4. Gráfico de Velocidad en Jira.....6

Ilustración 5. Gestión de Liberaciones en Jira.....7