

SCRUM EN ENTORNOS GEOGRÁFICAMENTE DISPERSOS

Salvador Esquivel

Director/es:

Alejandro Oliveros

**“Trabajo Integrador presentado para obtener el grado de
especialista en Ingeniería de Software”**

Facultad de Informática – Universidad Nacional de La Plata

Marzo, 2016

Índice

Contenido

Resumen	4
Abstract	4
1.- Introducción	6
1.1.- Objetivo	6
1.2.- Motivación/Estado del Arte	6
1.3.- Estructura del trabajo de integración	6
2.- Scrum.....	8
2.1.- Que es Scrum?.....	8
2.2.- Roles de Scrum	10
El Dueño del Producto (Product Owner – PO):	10
El Scrum Master (SM):	10
El Equipo de Desarrollo (Team):	11
2.3.- Eventos en Scrum	11
Reunión de planificación del Sprint (Sprint Planning):	11
Reunión diaria de sprint: (Daily meeting):.....	12
Revisión del Sprint (Sprint Review):	13
Retrospectiva del Sprint (Sprint Retrospective)	13
Refinamiento de la pila del producto (Product Backlog Refinement).....	13
2.4.- Artefactos de Scrum	14
Pila del producto (Product backlog):	14
Lista de pendientes del Sprint (Sprint backlog):.....	14
Incremento del Sprint:.....	15
3.- Entornos geográficamente dispersos (GSD).....	15
4.- Scrum en GSD	21
Problemas de diferencia horaria con poco o nulo solapamiento temporal entre las distintas locaciones:	30
Problemas de comunicación y entendimiento:.....	32

Problemas de falta de documentación formal para los requerimientos:	34
Problemas debido a diferencias culturales:	34
Problemas de capacitación:.....	36
Problemas de infraestructura:.....	37
Problemas de escala:	38
Problemas de falta de confianza:	39
Problema de comprensión y administración de los requerimientos:	39
Problemas de falta de herramientas para soportar el proyecto:.....	40
Problemas de coordinación:.....	40
Problemas de administración y control de proyecto:	41
Problemas varios:	41
Artículos de casos donde no reportan problemas:	42
5.- Conclusiones.....	44
6.-Bibliografía	46

Resumen

La utilización del marco de trabajo Scrum es una tendencia creciente dentro de la industria del software para empresas que buscan formas ágiles de desarrollo. En sus orígenes esta metodología requería que los integrantes de los equipos de trabajo estuvieran establecidos en un mismo ámbito, pues se necesita una gran comunicación y trabajo en conjunto.

Sin embargo, es cada vez más común que los equipos estén en ámbitos geográficamente dispersos, por lo que se hace necesario adaptar y/o buscar la forma en que esta metodología sea apta en estos contextos.

En la industria se están dando muchas experiencias que llevan a analizar los resultados obtenidos con esta forma de trabajo cada vez más expandida en todo el mundo

En el trabajo a desarrollar se investigarán:

- Definición del marco de trabajo Scrum.
- Problemática de entornos geográficamente dispersos: problemas de comunicación, culturales, horarios de trabajo, etc.
- Prácticas utilizadas para resolver los distintos inconvenientes dentro del marco de trabajo Scrum en los entornos geográficamente dispersos.

Abstract

Scrum framework is a growing trend in software industry to companies that are looking for development Agile forms. In their early days, this methodology required that the working teams members were established in an unique room, because it is necessary to have great communication and working together.

However, it is a common increasingly practice that the teams are in geographically dispersed places, which means that it is necessary adapt and/or look for the way that this methodology is suitable in these contexts.

In the industry, there are many experiences that lead to analyze the results obtained using this increasingly extended framework

In this paper, we will analyze:

- Scrum framework definition.

- Global Software Development issues: communication, cultural, working hours problems, etc.
- Practices used to solve the different issues in Scrum framework when applying in Global Software Development.

1.- Introducción

1.1.- Objetivo

Establecer el estado del arte del marco de trabajo Scrum en entornos geográficamente dispersos.

1.2.- Motivación/Estado del Arte

Scrum es una tendencia creciente dentro de la industria del software para empresas que buscan formas ágiles de desarrollo. En sus orígenes este marco de trabajo requería equipos de trabajo pequeños, cuyos integrantes estuvieran establecidos en un mismo ámbito, trabajando cara a cara(1), pues se necesita una gran comunicación y trabajo en conjunto.

Sin embargo, es cada vez más común que los equipos estén en ámbitos geográficamente dispersos, y las empresas precisen continuar obteniendo los beneficios logrados por las metodologías ágiles, tales como reducción del “time to market” y mayor flexibilidad en los proyectos, por lo que se hace necesario adaptar y/o buscar la forma en que esta metodología sea apta en estos contextos, no solamente para equipos pequeños, sino también al escalar a grandes equipos de desarrollo.

En la industria se están dando muchas experiencias que llevan a analizar los resultados obtenidos con esta forma de trabajo cada vez más expandida en todo el mundo

1.3.- Estructura del trabajo de integración

El alcance del presente trabajo contempla la búsqueda de artículos que documenten los casos de estudio que describan los distintos resultados en la utilización del marco de trabajo Scrum dentro del contexto de entornos geográficamente dispersos, comúnmente definidos en la literatura como “Global Software Development” (GSD) .

Este trabajo integrador se concentrará en obtener las distintas experiencias en la industria en este contexto de trabajo. El resto de este trabajo está organizado de la siguiente forma

En el capítulo 2 se define el marco de trabajo Scrum, sus características y conceptos: roles, ceremonias

En el capítulo 3 se describen las distintas problemáticas y desafíos inherentes a GSD

En el capítulo 4 se describen las distintas posibilidades en la que la metodología Scrum se utiliza en los proyectos con estructura GSD. Asimismo, obtendremos los distintos resultados obtenidos en la revisión de la documentación relevada para estos casos

En el capítulo 5 se formulan conclusiones preliminares como resultado de la descripción del capítulo anterior y se mencionan líneas de investigación o trabajos que se desprenden como parte de este estudio.

1.4.- Método de recolección de la información a analizar.

En una primera instancia, se hizo una búsqueda en Internet tomando la información resultante de las palabras “Scrum”, “Agile” y “Global Software Development” utilizando el buscador www.google.com

De esa primera búsqueda se obtuvieron blogs y artículos referidos a estos temas en conjunto y/o individualmente.

Se hizo una primera depuración descartando los blogs.

Para cada uno de los artículos obtenidos, se recorrieron sus las referencias y fueron buscadas recursivamente utilizando el buscador <https://scholar.google.com/>

En una segunda depuración, se descartaron aquellos artículos en donde Scrum no estuviera en el contenido de los mismos.

Se revisaron los que quedaron, clasificándolos según el objetivo del artículo en:

“casos de estudio”

“Descripción de problemas detectados”

“Marco de trabajo para investigación - Revisión de la literatura existente”

Llegando a la cantidad de 40 artículos que van a ser utilizados para revisar la experiencia de la industria para el tema que estamos evaluando

2.- Scrum

2.1.- Que es Scrum?

Takeuchi y Nonaka (2) publican un estudio en Harvard Business Review en el año 1986, en donde formulan que los proyectos pequeños y multidisciplinarios, similares a la formación de “scrum” en el Rugby, generan los mejores resultados.

Según la guía de Scrum(3), “Scrum es un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente.

Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos.

Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar.

El marco de trabajo Scrum consiste en los Equipos Scrum, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso.

Las reglas de Scrum relacionan los eventos, roles y artefactos, gobernando las relaciones e interacciones entre ellos. “

Como metodología ágil, sus conceptos fundamentales están descritos en el “Manifiesto por el Desarrollo Ágil de Software”(4), en donde se plantea que la comunidad ágil valora:

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta ante el cambio sobre seguir un plan

Se popularizó como una forma de reducir el “time to market”, aumentando la productividad, mejorando la calidad, y reduciendo costos eficientemente, adicionalmente con la posibilidad de manejar la volatilidad de los requerimientos a través del ciclo de vida del desarrollo, la promoción de la colaboración entre los clientes y los desarrolladores, y la entrega frecuente de incrementos de producto (5).

El marco de trabajo está basado en la ejecución de iteraciones (“Sprints”) de corta duración, entre 2 y 4 semanas. En cada iteración el equipo se auto planifica al principio de la misma, y al finalizar se revisan y evalúan los resultados obtenidos entre todos los interesados.

Podemos resumir el ciclo de desarrollo de Scrum en el siguiente gráfico, tomado de “ Información básica de SCRUM” (6):



Podemos observar a partir del gráfico, que el dueño del Producto es el encargado de armar la pila de producto, donde define los requerimientos en función de las necesidades de los clientes, equipo e interesados en el proyecto. Estos requerimientos están usualmente en formato de “historias de usuario” en una lista ordenada por prioridad. En la reunión de planificación del sprint, el equipo elige los ítems de la pila que se comprometerá a desarrollar en el sprint, para luego determinar y estimar las distintas tareas necesarias para la ejecución de los mismos, y en simultáneo auto asignárselas. Luego comienza la ejecución del sprint, que puede ser entre 1 a 4 semanas, desarrollándose diariamente la reunión de sincronización entre los miembros del equipo. Al finalizar, el equipo revisa el incremento de producto potencialmente entregable junto con los interesados en el proyecto, para luego en la reunión de retrospectiva analizar los inconvenientes que pudieran haber tenido durante el proceso y proponer las correcciones o mejoras necesarias.

2.2.- Roles de Scrum

Se definen en Scrum los siguientes roles:

- El Dueño del Producto (Product Owner – PO)
- El Scrum Master (SM)
- El Equipo de Desarrollo (Team)

El Dueño del Producto (Product Owner – PO):

Es el representante del cliente en el equipo de Desarrollo. Es el responsable de definir las prioridades de ejecución de los distintos requerimientos definidos, responsable de estas definiciones, y de responder a las distintas preguntas y dudas del equipo acerca de las mismas. Es el encargado de aprobar los incrementos de producto que se muestran en la reunión de revisión del sprint, al final del mismo.

Las definiciones de requerimientos tienen el formato de “historias de usuario”, con el sentido de una orientación o propuesta hacia el equipo de desarrollo acerca de las necesidades del cliente, para luego en reunión en conjunto, analizar y refinarla. De todas maneras, el dueño del producto sigue siendo responsable de estas definiciones y de su priorización. Es una lista dinámica que puede variar su contenido dentro del proyecto. Los ítems de producto de mayor prioridad tienen un mayor grado de detalle de definición, y los integrantes del equipo de desarrollo participan de las mismas, acordando con el Dueño del Producto el alcance a desarrollar, en las reuniones de “refinamiento”.

El Scrum Master (SM):

Es el responsable de velar por la correcta ejecución de los procesos de Scrum. Es un “líder servil”, es decir que está al servicio del equipo para resolver todos los impedimentos que puedan surgir en el marco del proyecto. Generalmente es la persona experta en el proceso Scrum, y que aconseja al equipo en la forma de proceder.

En lo posible tiene experiencia técnica con lo cual le permite resolver los problemas que pueda tener el equipo por sus propios medios, y además es conveniente que tenga un rol destacado dentro de la organización, para poder, a través de sus relaciones informales, acceder a la personas que puedan ayudarlo cuando los impedimentos superen su radio de acción directo. Es lo más

cercano al gerente de proyecto del método en cascada, pero con un rol completamente distinto, ya que no toma ninguna decisión ni efectúa ninguna asignación de tareas, si no que lo hace el equipo en su conjunto. Este cambio de rol es uno de los más difíciles de asumir cuando se modifica el esquema de desarrollo de cascada a Scrum.

El Equipo de Desarrollo (Team):

El equipo es el encargado del desarrollo del incremento de producto. Es auto organizado, es decir que el equipo en reunión de planificación es el que define responsabilidades, estimaciones y cuál va a ser el resultado del sprint siguiendo las prioridades definidas por el PO. Refina las definiciones de requerimientos junto con el Scrum Master y el Dueño de producto en las reuniones de refinamiento de la lista de producto.

Durante la ejecución del sprint, se reúne diariamente para sincronizar esfuerzos y plantear los distintos impedimentos al Scrum Master, encargado de canalizarlos y resolverlos. Se necesita que los participantes del equipo tengan una participación activa en el proyecto y se hagan dueños del mismo para que pueda tener éxito, siendo este requisito complejo de lograr, considerando las naturales características introvertidas de la mayoría de los profesionales de sistemas, donde suelen esperar a que les asignen las tareas para luego realizarlas. No hay diferencias de roles entre los miembros del equipo, salvo los naturales conocimientos y capacidades de cada uno, pero en principio todos están habilitados para hacer cualquier actividad. Todos son considerados “desarrolladores”

2.3.- Eventos en Scrum

Los eventos definidos son:

- Reunión de planificación del Sprint (Sprint Planning)
- Reunión diaria de sprint: (Daily Meeting)
- Revisión del Sprint (Sprint Review)
- Retrospectiva del Sprint (Sprint Retrospective)
- Refinamiento de la pila del producto (Product Backlog Refinement)

Reunión de planificación del Sprint (Sprint Planning):

La reunión tiene como información de entrada la pila del producto, elaborada por el Dueño del Producto.

Se divide en dos partes:

En la primera se hace el análisis y la selección de los requerimientos que se van a desarrollar en la iteración. Estos requerimientos en general están definidos según el formato de “historias de usuario”. El equipo negocia con el Dueño del Producto cuales son los ítems que se van a comprometer a entregar al final del sprint. Se hace una revisión de la estimación de los ítems comprometidos, en general medido en “puntos de historia” (story points). En este momento se determina el “Objetivo del sprint”, una síntesis de lo que se va a desarrollar en una oración, que tenga un significado para todos los participantes del proyecto

En la segunda parte el equipo elabora la lista de tareas necesarias para desarrollar las historias seleccionadas, las estima y los miembros del equipo se las autoasignan. Queda definido así el “sprint backlog” como resultado de este proceso, que es la base para utilizar en las sucesivas reuniones diarias de sprint

Reunión diaria de sprint: (Daily meeting):

Es una reunión de sincronización de duración máxima de quince minutos, es donde cada miembro del equipo responde las siguientes preguntas:

¿Qué hice desde la última reunión de sincronización?

¿Qué voy a hacer hasta la próxima reunión?

¿Tengo o voy a tener algún impedimento?

Se revisa el sprint backlog, y eventualmente se lo modifica si es necesario agregar y quitar alguna tarea, efectuando la asignación o des asignación correspondiente.

Es necesario revisar en este momento la carga de trabajo de cada miembro del equipo para balancearla, eliminando la sobre asignación, pasando tareas a otros que tengan disponibilidad.

Estrictamente deberá durar no más de quince minutos; es muy común que en esta reunión se intente debatir o discutir algún tema, por lo que el Scrum Master deberá no permitirlo.

Usualmente se coordina una reunión inmediatamente posterior para resolver estas cuestiones con los involucrados.

Revisión del Sprint (Sprint Review):

Al finalizar la iteración, el equipo presenta a los distintos interesados el incremento de software desarrollado. Deberá ser una o varias funcionalidades potencialmente entregables al usuario. El equipo en su conjunto deberá definir al principio del sprint que quiere significar “hecho” o “terminado” para el proyecto. Esta presentación es una demo informal del software, no una presentación en power point o similar. Puede llegar a estar soportada por una presentación, válida en los casos donde los asistentes a la reunión necesiten algún tipo de introducción al tema o puesta en contexto, para entender los que se va a mostrar.

En función de lo mostrado, los participantes aportan sugerencias, mejoras, las cuales el dueño del producto las considera para su incorporación o no en la lista de pendientes del producto y su priorización para incorporarlas en futuras iteraciones.

El dueño del producto aprueba o rechaza en este momento lo desarrollado por el equipo.

Retrospectiva del Sprint (Sprint Retrospective)

El equipo analiza la forma de trabajo y los distintos inconvenientes para poder mejorar la productividad, definiéndose “action items” con responsable y eventualmente se las incorpora como tarea en la siguiente iteración. Tiene una cierta similitud con las reuniones post mortem de los proyectos en cascada, siendo en este caso considero más valiosas porque las correcciones y mejoras de los procesos detectadas están determinadas durante el proyecto, al fin de cada iteración, permitiendo la posibilidad de implementarlas en las iteraciones siguientes, dentro del marco del proyecto actual, y no cuando el mismo ya terminó.

Refinamiento de la pila del producto (Product Backlog Refinement)

También conocido como “backlog grooming” o “story time”. Si bien originariamente no estaba establecida como parte de Scrum, se llegó a la conclusión que era necesaria implementarla; de esa manera se incorpora una nueva ceremonia para la revisión y refinamiento de la definición de requerimientos, en donde el equipo, en reunión con el PO, analiza los ítems del product backlog que van a desarrollarse en futuros sprints, con el objetivo de tenerlos claramente definidos al momento de la reunión de planificación. Es conveniente hacer una reunión semanal de al menos dos horas, donde en función de la prioridad, se van determinando y ajustando los detalles de los

próximos requerimientos, el equipo se va involucrando acerca del contenido de los mismos, y estará de esta manera en condiciones de estimarlos.

Considero esta actividad muy importante para que al tiempo de la reunión de planificación, esta sea bien efectiva y no se dilate por problemas de ambigüedades o desconocimiento de las definiciones.

2.4.- Artefactos de Scrum

Los artefactos de Scrum son:

- Pila del producto (Product backlog)
- Lista de pendientes del Sprint (Sprint backlog)
- Incremento del Sprint

Pila del producto (Product backlog):

Es una lista priorizada de funcionalidades definidas por el dueño del Producto con el formato de “historias de usuario” y estimadas, en general, en “puntos de historia”. El responsable de su realización y mantenimiento es el Dueño del producto, si bien pueden participar los miembros del equipo del proyecto en la elaboración, aportes de sugerencias etc. Se revisa en las reuniones de refinamiento, y se utiliza como información de entrada en la reunión de planificación del sprint, donde, en función de la prioridad y del volumen de trabajo necesario, se define un subconjunto de esta pila como alcance de la iteración.

Lista de pendientes del Sprint (Sprint backlog):

Es el subconjunto de la pila del producto que el equipo se compromete a desarrollar en la iteración, junto con las tareas necesarias para lograrlo. Surgen como resultado de la reunión de planificación, y el equipo de desarrollo en su conjunto es el responsable de la misma. Esta lista es la que se revisa diariamente en las reuniones del equipo, y eventualmente se pueden agregar tareas en cualquier momento en que sean descubiertas, así como eliminar las que se considere que no son necesarias. Es recomendable que las tareas sean de una duración no mayor de 16 horas, pues siendo superiores, no tiene sentido la revisión diaria

Incremento del Sprint:

Son los distintos componentes de software desarrollados durante el sprint y potencialmente entregables, que se muestran en la reunión de revisión del sprint a través de una demo. El equipo de desarrollo deberá definir para el caso que es que el software esté “terminado” (done).

Es fundamental para el marco de trabajo que como resultado de la iteración el equipo pueda lograr tener algo “mostrable”, o sea software trabajando, ejecutando la funcionalidad definida.

3.- Entornos geográficamente dispersos (GSD)

Según E. Carmel (7), la lista de características que distingue a los equipos de software globales respecto de los “normales” es corta y precisa:

a) Distancia (la distancia entre los desarrolladores entre sí y con los clientes o usuarios finales):

La distancia geográfica (dispersión) entre los lugares de desarrollo tiene un impacto directo en todas las formas de comunicación. Esta es menos frecuente y más restringida. La distancia impacta en la comunicación entre el diseñador y el cliente, entre dos desarrolladores, y entre los equipos de desarrollo y sus gerentes remotos. Las restricciones de comunicación se vuelven cada vez más significativas para los equipos de desarrollo, si es que están en la misma área metropolitana, dentro del mismo país o cruzando los océanos

La distancia afecta todas las maneras de coordinación y control

La distancia fuerza a la mayoría de las comunicaciones a canales electrónicos de varios anchos y colores. Estos canales no son tan ricos como la comunicación cara a cara.

b) Diferencias de huso horario;

Exacerba los problemas de comunicación. Cuando los desarrolladores en California e India no tienen una superposición natural de horarios de trabajo para conversaciones o video, uno de los lados siempre tiene que transigir. Casi todas las comunicaciones se canalizan a través de tecnologías asincrónicas, como e-mail o reuniones formales programadas.

c) Cultura nacional (incluyendo lenguaje, tradiciones nacionales, costumbres y normas de conducta).

Quizás sea el tema más confuso. Abarca tradiciones nacionales y étnicas, costumbres, normas de conducta, así como lenguaje. Los equipos con mezcla de culturas tienen más potencial de productividad y de problemas, en relación a los grupos más homogéneos culturalmente. Los problemas pueden provenir de desconfianza, falta de comunicación, and falta de cohesión. Los managers globales se enfrentan a un acto de equilibrio: promover un saludable conflicto de ideas y opiniones mientras controla diferencias culturales entre los miembros del equipo(7)

Para E. Carmel (7), existen dos fuerzas propulsoras para la globalización del software.

La primera es la propagación de las actividades de desarrollo a las nuevas naciones industrializadas, emergentes, que hasta hace poco permanecían afuera del campo de juego del software.

La segunda es la transición del desarrollo del software saliendo del tradicional centralizado, en una misma locación, a una forma en la cual los equipos de desarrollo globales colaboran a través de los límites de las naciones, trabajando en un mismo proyecto de software o producto o sistema.

E.Carmel (7) considera los siguientes seis factores catalizadores:

- Talento especializado: Las compañías de software quieren disponer de los mejores diseñadores y desarrolladores del mundo independientemente de donde se encuentren.
- Adquisiciones: las compañías de software van ampliando sus carteras de productos a través de adquisiciones. Las compañías más pequeñas no pueden sobrevivir contra los conglomerados de software más grandes. Estamos en la era de los “mega-mergers”
- Reducción de costos de desarrollo, al buscar mercados laborales donde los valores horarios son sustancialmente menores, como por ejemplo la India o China, Rusia, Filipinas, y algunos países de Europa Central.
- Presencia global: Estratégicamente, los ejecutivos de software precisan posicionar sus empresas como “firmas globales”. Esto implica para las compañías la necesidad de promover relaciones con determinados países que implican inversiones en esos lugares.
- Reducción del “time to market”. Dadas las diferencias horarias, el equipo puede ser productivo con desarrollo “follow the sun”: cuando un equipo está durmiendo, el otro continúa la tarea que deja el primero.
- Proximidad con el cliente: el lema de las mejores corporaciones es estar cerca del cliente. En consecuencia las organizaciones de software necesitan instalar lugares de trabajo cerca de sus clientes, en donde sea que estén

Luego, Carmel considera los siguientes factores sustentadores:

- Rigor en el proceso de desarrollo. La distancia deriva en mayores formalismos.

- Frescura interna: la diversidad trae nueva creatividad e inspiración para la organización de software.
- Distancia de las distracciones: las unidades distantes están lejos de las numerosas distracciones de las casas centrales.
- Plantel profesional de “globalizadores” de software
- Experiencia: en la mayoría de los casos los lugares remotos and pasado la curva de conocimiento y están teniendo buena performance

Por otro lado, considera los siguientes factores de tamaño:

- Escala: En algún punto la los centros de desarrollo locales se vuelven demasiado grandes y difíciles de manejar
- Sinergia evolutiva de escala: la escalabilidad provoca oportunidades para colaboración global que son imprevistas.

Finalmente, plantea los siguientes factores de “visión”:

- Transparencia en la locación: se elimina la percepción de distancia a través de la tecnología.
- Organización virtual: se arman entidades de diferentes organizaciones semejantes a una red que tiene jerarquía y centralización débil

La experiencia profesional del autor de este trabajo de especialización en el campo de GSD sugiere agregar el siguiente factor no contemplado por E. Carmel:

- Beneficios para el empleado, al facilitar la modalidad de trabajo desde su casa en lugar de concurrir a la oficina (“home office”).

Con respecto al home office, en (8) se plantea que la diferencia respecto de otras modalidades de modelos distribuidos, que el grupo está individualmente y completamente distribuido, mientras que en los otros casos se encuentran pequeños grupos integrados en un mismo lugar físico. Una ventaja del home office es la disminución de población en las ciudades, economía del costo del espacio físico en las empresas, dar la chance a gente que está alejada de poder trabajar, que de otra manera no podría hacerlo. Como desventaja podemos anotar pérdida de la comunicación, y una exclusión de la sociedad. Dentro del campo del desarrollo de software, esta modalidad se va haciendo cada vez más extendida, con el crecimiento de las comunidades de código abierto (“Open Source communities”), donde la gente hace su aporte durante su tiempo libre en su casa.

Por su parte, James D. Herbsleb y Deependra Moitra(9) coinciden en afirmar que en los últimos años el software se ha vuelto en un componente vital en casi todos los negocios, el éxito en los

negocios depende cada vez más de usar software como un arma competitiva. Hace más de una década, impulsadas por la búsqueda de costos más bajos y el acceso a recursos capacitados, varias organizaciones (especialmente corporaciones multinacionales) comenzaron a experimentar con centros de desarrollo ubicados remotamente y con subcontratación. Algunos factores que han acelerado esta tendencia:

- La necesidad de aprovechar el pool de recursos globales para usar recursos escasos en forma satisfactoria y competitiva en costos
- Las ventajas de negocio de proximidad al mercado, incluyendo conocimiento del cliente y condiciones locales,
- La rápida formación de corporaciones virtuales para explotar oportunidades de mercado
- Severa presión para mejorar el time-to-market haciendo uso de las diferencia horarias con el desarrollo “follow the sun” y la necesidad de capitalizar oportunidades de adquisiciones y fusiones de empresas en cualquier lugar donde ellas se presenten

En consonancia, en(10)se plantea que los equipos virtuales:

- Reducen costos, recortando gastos de viaje y tiempo, creando nuevas “e-economies” de escala, y diseñando mejores procesos digitales.
- Disminuyen el ciclo de desarrollo pasando de procesos en serie a en paralelo, estableciendo mejores comunicaciones y generando mayor confianza extendida
- Aumentan la motivación permitiendo participación más diversa, estimulando la creatividad de procesos y productos y alentando sinergia de desarrollo de nuevos negocios.
- Impulsan aprendizaje capturando conocimiento en el curso natural del trabajo, y compartiendo las mejores prácticas.

Adicionalmente una ventaja más, que indica (11), al utilizar “off shoring” para el desarrollo de software se posibilita aumentar y disminuir fácilmente el tamaño de los equipos sin tener que recurrir a los despidos. Así es que empresas de desarrollo de USA, Europa, Japón utilizan esta práctica con Europa del Este, Rusia o el lejano Oriente (India, China, Malasia), siendo típicamente los costos por hora de desarrollo “offshore” el 30 % de “onshore”.

En paralelo, esta forma de trabajo genera distintos tipos de desafíos y problemas en el contexto de trabajo en equipo para el desarrollo de Software, como por ejemplo:

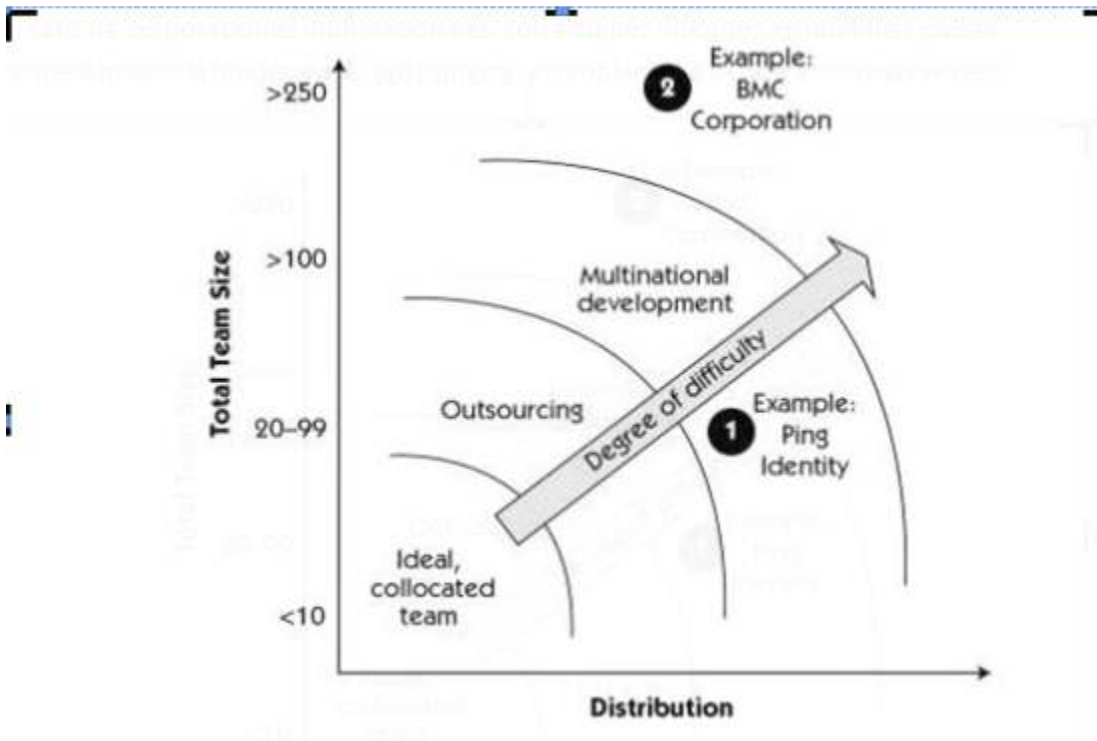
- Problemas de comunicación debido a distintos factores, a saber:
 - Diferencias horarias. Si los integrantes del equipo están en países con distintos husos horarios, solamente van a poder tener comunicación inmediata en las franjas horarias coincidentes, lo que genera retraso en las respuestas, dificultades para coordinar reuniones de equipo, menos posibilidades de colaboración entre los integrantes.

- Dispersión geográfica entre los miembros de los equipos. Imposibilita las reuniones “cara a cara”, contactos informales, reducción de confianza
 - Diferencias culturales. Las distintas regiones tienen distintas problemáticas de actitud ante las mismas circunstancias, por lo que puede dar como resultado malas interpretaciones, problemas de relación, diferentes percepciones de la autoridad, falta de mecanismos para evitar falta de entendimiento y reducción de la cooperación (12)
 - Problemas tecnológicos: el hecho de que la tecnología no está lo suficientemente desarrollada en los países emergentes, hace que tengan problemas de conectividad, banda ancha, cortes de electricidad, etc., con los consiguientes perjuicios.
 - Diferencias en el lenguaje. No solamente las diferencias idiomáticas sino también el acento regional dificultan el entendimiento e incluso dan lugar a interpretaciones contradictorias.
 - Pérdida de la comunicación no verbal. La mayor proporción de la comunicación se da a través de lo no verbal, es decir gestos, postura corporal, etc. Esto lamentablemente se pierde.
-
- Pérdida de confianza, pues el jefe deja de tener un control visual en los empleados, depende exclusivamente de la voluntad del empleado de hacer su trabajo.
 - Moral del equipo decreciente. Al no estar en un mismo lugar, se pierde la noción de equipo y por consiguiente se dificulta la capacidad de búsqueda de un objetivo común a conseguir. Los miembros del equipo pueden llegar a sentir que no son lo suficiente valorados en su trabajo.
 - Problemas de coordinación y control: como consecuencia de los anteriores
 - Problemas de colaboración entre los miembros del equipo

Dean Leffingwell (13) plantea que en escala, todo desarrollo es desarrollo distribuido; la oportunidad de comunicarse informalmente decrece con la distancia: la gente ubicada a 100 metros o más, tiene apenas el 5% de chance de hablar con otra. Al incrementar el tamaño del equipo, trabajar en un mismo ámbito no es una realidad práctica para la mayoría de las empresas.

Aunque las empresas con gran cantidad de desarrolladores estén en una única ciudad, el tamaño de la organización es tal que la gente no está en el mismo espacio de trabajo, el mismo piso, el mismo edificio o el mismo campus.

En el caso de corporaciones multinacionales, con equipos ubicados en múltiples países, incrementándose la tendencia de outsourcing, el problema se va volviendo más severo



En la figura se representa como el grado de dificultad aumenta con la distribución y el tamaño del equipo de desarrollo

Asimismo(14) , factores contextuales de especificidades claves del proyecto, del equipo y distancia pueden también impactar en la efectividad del equipo. Esto incluye, por ejemplo, la naturaleza contractual, el dominio de aplicación, la volatilidad de los requerimientos, el personal del proyecto, la distribución de las locaciones, las distancias geográficas, socio-cultural y geográfica entre los participantes

4.- Scrum en GSD

Gran parte del éxito de Scrum está en que los miembros de los equipos se sitúan en un mismo ámbito geográfico (12), siendo la interacción entre los miembros del equipo dificultosa en GSD. La comunidad ágil aboga por la importancia de la proximidad cercana y la relación entre los miembros del equipo(15), lo cual hace presumir complicada la implementación de estos conceptos.

La práctica de Scrum en estos entornos se está haciendo cada vez más extensiva, incluso varios plantean (15) que el uso cuidadoso de las prácticas ágiles en proyectos globalizados y distribuidos pueden proporcionar una cantidad de beneficios en términos de comunicación, mejora de la productividad, de administración de proyectos, confianza, motivación del equipo, visibilidad del proyecto.

Se proponen los siguientes modelos de equipos Scrum distribuidos: (10)

- Aislados: los equipos están aislados dispersos en distintas geografías sin contacto entre sí
- Scrum de scrum distribuidos: los equipos están aislados e integrados a través de reuniones de scrum de scrum regulares
- Scrum totalmente distribuidos: los equipos tienen sus miembros dispersos a través de las geografías.

En esta tercera división podríamos considerar dos aspectos distintos:

- Una primera consideración en la cual los equipos, si bien están dispersos, sus componentes están ubicados en una misma oficina para aquellos que están en determinado lugar, es decir que asisten a un espacio común a trabajar.
- Una segunda consideración en donde sus miembros, total o parcialmente, trabajan en forma remota desde sus domicilios (“home office”), sin ningún otro tipo de contacto salvo el “virtual”, o sea a través de internet, teléfono, etc.

En estos contextos, veremos las distintas experiencias documentadas en los casos de estudio de los artículos analizados.

Los casos de estudio analizados se esquematizan en el siguiente cuadro:

Fuente	Empresa/Industria	Proyecto	Composición del equipo
Yahoo! Distributed Agile -Notes from the World Over (17)	Yahoo/software	Podcast	SunnyVale, California Bangalore, India
		Vespa News Search	Product Managers en Sunnyvale y Santa Monica El Team en Trondheim, Noruega.
		Yahoo! Answers	SunnyVale, California (2 equipos) London, UK (2 equipos)
WhyScrumWorks (18)	Danske Bank/Financiera	DELHI	7 personas en 1 scrum team en Dinamarca 8 personas en 1 scrum team en la India
Fully Distributed Scrum - The Secret Sauce for Hyperproductive Offshored Development Teams (16)	Xebia/software	ProRail PUB	1 equipo en Holanda y 3 en la India
Usage of SCRUM Practices within a Global Company (19)	Organización global que mantiene centros de desarrollo de Software en América y Asia	Project A control impositivo global en procesos de negocios financieros	1 PO en USA, 1 SM en USA, 6 dev/testers en Brasil
		Project B logística, procesos de salida y entrada	2 PO en China, 3 dev/testers en USA, 6 dev/testers en Brasil, 1 SM en USA
Back to Basics- The Role of Agile Principles in Success with an Distributed Scrum Team (20)	FAST (software)	no especifica nombre	en Noruega y en Boston. No especifica cantidad ni roles

--	--	--	--

Cross-Continent development using Scrum and XP (21)	SAS Institute (Software)	No especifica nombre	1 equipo de +10 developers en Copenhague Dinamarca y el resto (+80) en Cary, Carolina del Norte USA. No especifica cantidad ni componentes de estos
---	--------------------------	----------------------	---

Implementing Scrum in a Distributed Software Development Organization (22)	BMC Identity Management (software)	no especifica nombre	2 equipos en Francia, Tel Aviv y Florida y 2 equipos en Tel Aviv. Un equipo de Product owners con 4 en Israel, 1 en Francia y 1 en USA
--	------------------------------------	----------------------	--

Distributed Agile Development - Using Scrum in a Large Project (23)	IT Company (no está el nombre)	no especifica nombre	20 personas en Noruega y 20 en Malasia, divididos en 7 equipos. 5 product owners en Noruega, 1 Scrum Master en Malasia y 2 SM en Noruega
---	--------------------------------	----------------------	--

Scrum Practice Mitigation of Global Software Development Coordination Challenges: A Distinctive Advantage? (12)	No especifica nombre – Paper millquality control	PaperSys	1 único equipo (4 onshore Finlandia, 6 offshore no especifica lugar)
	No especifica nombre –Energy production control	EnergySys	1 único equipo (4 onshore Finlandia, 7 offshore no especifica lugar)
	No especifica nombre - Collaboration software	ToolSoft	11 en 4 equipos onshore Sydney, Australia; 4 en 1 equipo offshore en USA
	No especifica nombre - Telecom	TestSoft	1 equipo onshore en Finlandia, 2 equipos

Trabajo final integrador: Scrum en Entornos Geográficamente Dispersos.
Salvador Esquivel

	test platform		offshore: 1 en Brasil y otro en Alemania y la India
How can agile practices minimize global software development co-ordination risks (15)	“Alpha” compañía de desarrollo de software de Australia	“Alpha-Global”	2,5 en Australia, 25 en Malasia. No especifica cuántos equipos Scrum y cómo se conforman
Distributed Scrum in a Research Project Management (23)		Eurace (research project)	Alemania, Francia, Turquía, Italia, UK, USA. No especifica cantidad
The Creation of a Distributed Agile Team (24)	British Telecom (telecomunicaciones)	Web21c SDK	Londres (Inglaterra), Denver (USA) y Bangalore (India)
The Good and Bad of Agile Offshore Development (25)	Check Free (software)	no especifica	Waterloo, Canadá y Atlanta, Georgia, India (offshore). Entre 8 y 15 equipos de 7 personas
Adaptive Engineering of Large Software Projects with Distributed Outsourced Teams (26)	Sirsi Dynix (software)	no especifica	mitad en Provo Utah (USA) y mitad en St. Petersburg (Rusia)
Distributed Scrum: Agile Project Management with Outsourced Development Teams (27)	ídem anterior	ídem anterior	ídem anterior
Facing Fake-to-Fake:	Saxonia Systems AG	no especifica	Dresden y Gorlitz

Lessons learned from Distributed Scrum (28)	(Alemania)		
---	------------	--	--

Fully distributed Scrum - Replicating local Productivity and Quality with Offshore Teams (29)	El mismo caso de (16)		
---	-----------------------	--	--

Scaling Scrum in a Large Distributed Project (30)	large, global product development company		20 equipos (170 personas) en cuatro países: Finlandia (10 equipos de desarrollo), Alemania (2 equipos de testing) Grecia (2 equipos de testing) e India(6 equipos de desarrollo)
---	---	--	--

Outsourcing and Offshoring with Agility: A Case Study (31)	Elegance Technologies, Inc., EXTOL International, Inc. (Software)	UCC net application	2-3 onshore staff and 5-10 offshore staff. No especifica lugar
--	---	---------------------	--

Using Scrum in a Globally Distributed Project: A Case Study (32)	IT company (no especifica nombre)		20 en Noruega y 20 en Malasia
--	-----------------------------------	--	-------------------------------

Lessons learned from an extremely distributed Project (33)	no especifica nombre	no especifica nombre	UK, 8 developers en Sydney, 20 personas en Bangalore
--	----------------------	----------------------	--

--	--	--	--

Experiences in Scaling the Product Owner Role in Large-Scale Globally Distributed Scrum (1)	no especifica nombre	proyecto A	25 equipos en Finlandia, Grecia, Alemania e India
		proyecto B	25 equipos en Finlandia y Hungría

Fully Distributed Scrum Linear Scalability of Production between San Francisco and India (11)	Xebia	TBD.com	San Francisco, India
---	-------	---------	----------------------

Tool Support for Globally Distributed Scrum (34)	no especifica	no especifica	Dinamarca e India. No especifica cantidad
--	---------------	---------------	---

Overcoming the Challenges of Building a Distributed Agile Organization (35)	First American Corelogic	no especifica	Bangalore, India y Santa Ana, CA
---	--------------------------	---------------	----------------------------------

Towards an Understanding of Tailoring Scrum in Global Software Development: A Multi-case Study (14)	maquinaria y sistemas industriales	Paper Info (seudónimo)	4 en Finlandia y 6 en un país offshore distribuido en 3 locaciones
	Energía y refinería	Energy Info (seudónimo)	4 en Finlandia y 7 en un país offshore en 1 locación
	proveedor de software colaborativo	Collaborationsoft (seudónimo)	11 en Sydney Australia y 4 en San Francisco USA
	manufactura de productos de	TestSoft (seudónimo)	1 en Finlandia, 11 en Brasil, 2 en Alemania, 1

Trabajo final integrador: Scrum en Entornos Geográficamente Dispersos.
Salvador Esquivel

	telecomunicaciones		en India
Towards a Framework for Using Agile Approaches in Global Software Development (5)	AusBest (software)	AGI Project	3,5 en Australia y 25 en Malasia
Insights into an Agile Adventure with Offshore Partners (36)	CampusSoft (software)	no especifica	20 onshore en UK, 4 en Rumania y 20 en India
Growing and Sustaining an Offshore Scrum Engagement (37)	Kelley Blue Book (venta de autos por internet)	no especifica	Irvine, CA, USA, 10 en Beijing, China y 5 en Hyderabad, India
Adopting Agile in Distributed Development (38)	Wipro Technologies (software)	no especifica	no especifica
Distributed Scrum for Large-Scale and Mission-Critical Projects (39)	Mission critical Software provider	no especifica	Equipo dividido en dos locaciones dentro de UAS, distanciados 100 millas. No especifica nombres.
Distributed Agile: Growing a Practice Together (40)	Elastic Path Software and Luxoft Canada (software)	No especifica	Vancouver (Elastic Path Office), Vancouver (LuxoftOffice), Moscow, Dubna, yOmsk .3equipos:2 deElasticPathy 1

			deLuxoft. El equipo Luxoft incluye gente nearshore desde la oficina de Vancouver y recursos offshore en Rusia. No especifica cantidad
--	--	--	---

Scrum and Global Delivery: Pitfalls and Lessons Learned (41)	Professional services company. No especifica nombre	NOR1	2-4 Noruega, 18-20 País de Europa del Este
		NOR2	2 Noruega, 5-6 País Báltico

Using Scrum in Distributed Agile Development: A Multiple Case Study (42)	Service company: Desarrollo de una nueva versión de software servicio de impresión para uso interno en nuevos mercados	PrintCo	Onsite: Finlandia (7) Offsite: Latvia, 2 sites (2+1) Alemania (1)
	Compañía Industrial: desarrollo adicional y mantenimiento de una herramienta de administración de la información para uso interno	PulpCo	On site: Finlandia, site principal + 1 consultor subcontratado trabajando on site (9+1) Off site: Russia, subcontratados -6
	Compañía IT: desarrollo adicional y mantenimiento de un producto de software de energía que está en uso en algunas compañías alrededor del	Energy-Software	On site: Noruega (~20) Offsite: Malasia (~20)

	mundo.		
Distributed Scrum when turning into maintenance: a single case study (43)	Compañía global IT.	no especifica	Dueño del Producto y un miembro del equipo en Estocolmo, Suecia; Scrum Master y 2 miembros en Helsinki, Finlandia, y cuatro miembros en India, Pune.
From RUP to Scrum in Global Software Development: A Case Study (44)	Una firma de consultoría de software, ubicada en Holanda.	no especifica	Holanda e India
A Case Study of Coordination in Distributed Agile Software Development (45)	Compañía de tamaño medio con aproximadamente 150 empleados (software)	India I	3 en Noruega - 5 en India
		India II	1 en Noruega - 4 en India
		Eastern Europe	2 Noruega - 4 a 5 en país Europa del Este
Adoção de SCRUM em uma Fábrica de Desenvolvimento Distribuído de Software (46)	O3S Open Source Software Solutions	ANKOS (A New Kind Of Simulator)	Equipo distribuido (no especifica)

Agile software development in a globally distributed environment. (47)	Es el mismo caso de (44)		
--	--------------------------	--	--

Estudo de Caso da Utilização de Scrum no Desenvolvimento Distribuído de Software (48)	INES (Instituto Nacional de Ciências e Tecnologia para Engenharia de Software) disponible en Sourceforge	FireScrum	No especifica. Están en 4 ciudades distintas de Brasil
---	--	-----------	--

En lo que sigue se analizan los problemas que se detallan en los casos de estudio y las acciones que se tomaron para resolverlos. Los problemas identificados se presentan agrupados en los siguientes grupos:

- Problemas de diferencia horaria con poco o nulo solapamiento temporal entre las distintas locaciones
- Problemas de comunicación y entendimiento
- Problemas de falta de documentación formal para los requerimientos
- Problemas debido a diferencias culturales
- Problemas de capacitación
- Problemas de infraestructura
- Problemas de escala
- Problemas de falta de confianza
- Problema de comprensión y administración de los requerimientos
- Problemas de falta de herramientas para soportar el proyecto
- Problemas de coordinación
- Problemas de administración y control de proyecto
- Problemas varios

En lo que sigue se analiza cada grupo de problemas:

Problemas de diferencia horaria con poco o nulo solapamiento temporal entre las distintas locaciones:

En (17) se considera que esta situación genera un sentimiento de verse menospreciados por parte de los integrantes del equipo remoto, impacta hasta en la vida cotidiana y las costumbres, por no poder, por ejemplo, tener la cena familiar. En otro proyecto, el equipo remoto tiene estrictas costumbres de no trabajar fuera del horario de oficina, dificultando la planificación de reuniones fuera de horario laboral.

Para (12), la diferencia obliga a extender el horario laboral de uno o varios equipos, generando aumento en los costos para coordinación.

Se reduce la posibilidad de comunicación sincrónica según (15), (11), (35), (14), (42), habiendo dificultad para planificar los horarios de las reuniones.

En (20) se observa un problema adicional, donde un equipo se junta en una sala de reuniones, mientras que el otro lo hace cada uno desde sus lugares de trabajo.

Como problema colateral, (17), al verse obligado a dividir la reunión de planificación en 2, hace que el equipo que tiene la primera reunión, elige las historias más interesantes. Además tienen problemas de sincronización al asignarse tareas interdependientes entre los dos equipos.

También (20) deben tener reuniones de revisión de sprint separadas por la diferencia horaria; para (32) la diferencia horaria impide tener reuniones largas de planificación.

Vemos en (40) que hay en los equipos tres husos horarios diferentes, y se obligan a tener retrospectivas asincrónicas

Las acciones para resolverlos consistieron en:

Para (17), reuniones locales de equipo y solamente uno de cada equipo en reunión de coordinación de frecuencia dos o tres veces por semana. En su segundo caso, estricto time box de dos horas para las reuniones de revisión, planeamiento y retrospectiva juntas de 4:00 a 6:00 p.m. en Noruega, y 7:00 a 9:00 a.m. en Sunnyvale/Santa Mónica, luego continúan planificando el sprint en forma separada, y reunión 2/3 veces por semana entre Product Owner (Sunny Vale) y Scrum Master (Noruega).

Similar es en (35), donde unos miembros cambiaron horario de las reuniones diarias para las 9:00 PM desde sus casas, adicionalmente al horario común de trabajo, pero participando solos en las reuniones de planificación y análisis de requerimientos.

Otra solución fue en (12), mediante el aumento de las horas comunes de solapamiento, modificando horarios laborales en las reuniones.

Alternativa de (14) y (20) es la de hacer reuniones diarias separadas entre los subequipos y luego una scrum de scrum entre representantes de cada uno.

Para resolver la selección de historias, (17) exige una selección por estricto orden de prioridad definida por el Product Owner, y los problemas de sincronización definiendo reunión scrum de scrum dos veces por semana.

Con respecto a las reuniones de sprint, (20) considera que es necesario tener una única reunión de revisión, por lo que hay que planificarla como sea para lograrla.

Destaca (32) la importancia de los viajes entre las locaciones para mejorar la comunicación.

Para (40) evaluaron separar los miembros de los equipos de tal manera que no tengan más de dos husos horarios distintos; para las retrospectivas, las hacen separadas y los resultados se comparten

Problemas de comunicación y entendimiento:

En uno de los proyectos descritos por (17), las reuniones debían hacerse utilizando teléfono y video conferencia, obligando a un equipo a estar tarde/noche en la oficina, por no poder hacerse en cada casa al ser la banda ancha domiciliaria insuficiente.

Obviamente, la inexistencia de comunicación cara a cara es un problema recurrente en los distintos casos: (17), (19), (21), (15),(24), (28), (33), (1), (36),(43).Incluso (32) y (42) reconocen que hay gran dificultad de reconocer quién habla al no poder ver la cara.

Para (12), la reducción de contacto informal puede llevar a la falta de conciencia de la criticidad de las tareas.

Para (33), se evidencian problemas de relación entre los miembros de los equipos, y se comete el error de filtrar información al equipo remoto con el sentido de simplificar, provocando moral baja y sentido de frustración. También detecta que al aumentar la cantidad de los miembros de los equipos, la comunicación se hace más compleja, y la conformación de los equipos si no son multidisciplinarios, les imposibilita hacerse dueños del proyecto.

Para (39), la video conferencia no está siempre disponible, las conferencias por teléfono no son buenas por la falta de comunicación cara a cara. El sistema telefónico no funciona correctamente.

En (40) también reportan problemas de comunicaciones.

Para (41), la comunicación informal es dificultosa, encuentran diferentes expectativas y presunciones acerca de cómo implementar scrum debido a las barreras culturales y de lenguaje.

En (45), las discusiones por teléfono entre los dos sub equipos no funcionan, lleva mucho tiempo interactuar y comunicarse.

En (48) las reuniones diarias constituyen un problema.

También en (45) informan que la conferencias por teléfono no se entienden por el lenguaje, por lo que se reemplazó su uso por chat y mail.

Finalmente, para (41) se llegó a plantear problemas de expectativas y lenguaje usado en las reuniones de dirección

Las formas para tratar de resolver este problema son variadas:

Para (17), se hace viajar al equipo de un sitio al otro y viceversa para desarrollar un sprint entero, tendiendo a tener reuniones locales todo lo posible; por otro lado definen sprints cortos, de dos semanas, haciendo reuniones de revisión en conjunto pero las retrospectivas separadas, con posterior reunión de coaches.

En forma similar (36) se buscó generar relaciones personales trasladando a los desarrolladores offshore al lugar onshore y viceversa, temporariamente.

Para (15), la misma dinámica de los eventos de Scrum, reuniones diarias, de revisión, retrospectivas, de refinamiento, ayudan u obligan a minimizar o reducir este problema, al aumentarse la visibilidad, la comunicación entre los interesados, la coordinación de proyecto.

En (24), tratan que los equipos puedan trabajar en un mismo espacio físico todo lo que sea posible, llegando al extremo en que si no hay oficinas, de reunirse en coffee shops, casas o salas de reuniones. En todo caso, es indispensable el uso de tecnología para reuniones vía web.

En(28), Utilizan salas de videoconferencias para las daily meetings, y desarrollan un pizarrón digital (add on de Jira).

En (33), se implementan embajadores para mejorar la comunicación entre los lugares, generar confianza y permitir transferencia de conocimientos .

El problema de comunicación esencial en (1) es en términos de entendimiento de los requerimientos, por lo que deciden desarrollar workshops de diseño y requerimientos.

En (39) se usa una herramienta llamada web demo, también mensajería instantánea.

Se resuelve en (40) reducir la frecuencia de las reuniones diarias a 3 veces por semana, con la posibilidad de hacerlas desde la casa a raíz de las diferencia horarias, y efectuando una superposición de horarios esos días.

Deciden en (41) implementar el rol de co-Scrum Master en el sitio offshore y un aumento de viajes entre los sitios; asimismo, se reemplaza el arquitecto onshore por uno offshore que se muda onshore durante el proyecto para facilitar la comunicación. Para el segundo caso del mismo reporte, la gente onshore está más tiempo offshore y se reemplaza el coordinador local por el manager on site.

El planteo en (45) es el de delegar el rol de SD en una persona en India que viaja a Noruega por un período. Se abandonan las daily meetings. Se dividen los requerimientos en módulos para disminuir las interdependencias. Se empieza a especificar y documentar el product backlog para luego dejar que los detalles lo decidiera el equipo.

En (48), se resuelve simplemente usar herramientas de comunicación y conferencia estando el Scrum Master controlando y gerenciando las reuniones.

Y en (41), se debe efectuar un training inicial acerca del lenguaje y metodología Ágil para el management.

Problemas de falta de documentación formal para los requerimientos:

Solamente en un artículo (19) se comenta que la documentación, que no se prioriza en los principios del Manifiesto ágil, es necesaria formalizarla en este contexto para poder disminuir los riesgos de ambigüedades y problemas de entendimiento

Como solución, plantea conservar la documentación valiosa y definir qué documentación es útil para equipos distribuidos y cuál aporta valor para el proyecto

Problemas debido a diferencias culturales:

Los casos presentan distintos aspectos de esta problemática:

En (12) se alega que debido a que las prácticas laborales son distintas en los distintos sitios de trabajo, se lleva a problemas de coordinación entre los equipos; por otro lado, la dificultad de comprenderse, lleva a reducción de la cooperación y del trabajo en equipo.

En (15) tenemos similarmente que se generan malentendidos y adicionalmente que surgen diferentes expectativas respecto de prácticas de liderazgo y problemas en adaptarse a las distintas normas culturales de trabajo.

Para (25), también las normas culturales y el estilo de comunicación son un obstáculo, el equipo offshore es reticente a estar en desacuerdo en público o a debatir ideas, problema planteado también en (35), donde directamente los miembros del equipo no participan de las reuniones, siendo representados por el gerente; coincide (36) al comentar que el equipo de la India siempre le dice “sí” a todo, con lo cual se generan severos problemas de falta de confianza; los casos(14) y (5) tienen una distancia socio cultural importante, con problemas en el lenguaje, normas y cultura.

En (37), severos inconvenientes de lenguaje con el equipo remoto, de procedencia china, y se detecta un problema cultural también relevado en (19): el equipo de desarrolladores no quiere probar, no se logra que sea multidisciplinario.

Tanto para (32) como para (23), las diferencias culturales hacen que en las reuniones se generen silencios y falta de participación.

En (32), se nota un problema de diferencia cultural al reportar los impedimentos, la reunión diaria dura solamente pocos minutos.

En (42), se ven diferencias culturales por ejemplo al reportar impedimentos; lleva tiempo aprender a reportar información que sea útil para otros; también las diferencias culturales y de lenguaje pueden causar silencio de algunos participantes. Reporta (43) problemas culturales sin especificar cuáles.

Para solucionar estos temas se toman las siguientes acciones:

En (12) se presenta por un lado que lograr reuniones cara a cara con viajes de los participantes ayuda, y que las distintas actividades iterativas hacen que se puedan monitorear los problemas para poder resolverlos.

También para (15), las prácticas de Scrum (reuniones diarias, revisión, retrospectivas) permiten resolver estos inconvenientes.

Sin embargo, (25) entiende que los problemas se mantienen dentro del equipo, por lo que es necesario crear un ambiente de trabajo seguro que anime a la gente a expresarse. Se deben respetar los feriados locales. La comunicación debe ser muy explícita, para no dar lugar a malas interpretaciones, pues es muy difícil resolver la forma de respuesta de los equipos principalmente asiáticos.

Los casos de (14) deciden agregar reuniones pre y post planificación y no hacen retrospectiva. El product backlog se define onshore, mientras que el sprint backlog y el control diario se realiza offshore.

Para (5), la clave es el correcto uso de herramientas de comunicación.

Sin embargo (36) reconoce que el problema no se puede resolver, lo que implica falta de confianza en las comunicaciones.

La capacitación en inglés es el único camino para mejorar la comunicación del equipo chino en (37), y se tiene que demostrar que todos los otros lugares los desarrolladores efectúan tareas de testing cuando es necesario.

En (32), la continua práctica de las reuniones diarias hace que se vaya mejorando la comunicación al tener al Scrum Master alentando a los miembros a plantear los problemas.

En (42), el Scrum Master alienta para que participen todos, y en la Daily meeting se mejora la comunicación.

Problemas de capacitación:

Es común la falta de conocimiento de la metodología, tanto del equipo como de la estructura gerencial, como lo remarca (19) para los dos proyectos analizados.

Falta de buenas prácticas de ingeniería remarca (20), no específicamente relativo a GSD, pero común de todas maneras.

Falta de capacitación del equipo offshore es planteado por (25) y por (37), con el agravante de desconocimiento del producto, de la incapacidad de retener a los desarrolladores más experimentados y de la capacidad limitada de los analistas de negocio para escribir buenas historias de usuario.

Tener un gerente de proyecto sin experiencia hace que (5) no existan políticas, procesos y planes de proyecto. Aquí también los desarrolladores experimentados se fueron del proyecto.

En (39) vemos que los empleados nuevos generan muchos bugs por ser el sistema muy complejo, entrenamiento de los nuevos es deficiente, especialmente si es remoto, debería ser cara a cara.

Para resolver estas cuestiones;

Se documenta que para (19), los eventos de retrospectivas ayudan a mejorar los procesos, incorporación de buenas prácticas en (20), mientras que (25) implementa viajes de capacitación del equipo offshore al sitio onshore, atendiendo también a la generación de relaciones personales, importantes para generar confianza y espíritu de equipo.

En (37) se toma una solución similar, implementando visitas al sitio onshore para capacitación en el lugar, más visitas periódicas y mayores controles de revisión de código y reuniones de tipo

técnico periódicamente; adicionalmente, la participación del equipo onshore tiene que ser más intensa para compensar la falta de seniority del equipo chino, adicionando un líder como arquitecto para revisión de los diseños.

En (5) se precisa definir claramente los procedimientos ágiles, no pudiendo resolver la ida de los recursos.

En (39), se implementa un sistema de compartir información entre los desarrolladores experimentados y los nuevos, como opción, juntan nuevos y experimentados en el mismo equipo pero no siempre es posible, adicionalmente usan una wiki para compartir información

Problemas de infraestructura:

Es muy común encontrarse con estos tipos de problemas, especialmente en países emergentes, donde la disponibilidad de los servicios de internet y banda ancha, es escasa, intermitente, etc.

Así vemos que (23) y (32) no pueden efectuar video conferencias por problemas técnicos, que (25) no cuenta con herramientas de colaboración, de administración, (28) extiende la problemática aduciendo que el uso de video conferencia por períodos largos de tiempo es muy cansador, la calidad de los micrófonos es mala, y las video conferencias se usan solamente para reuniones formales.

En (33) y (42) se tienen problemas de calidad de la comunicación, los participantes no se entienden ni se escuchan.

En (34) hay problemas con las herramientas que usan para la definición de requerimientos, que no están integradas: Word, Excel, Application life manager. Adicionalmente, no pueden utilizar el pizarrón típico de los proyectos locales para las reuniones diarias.

En (35) se reporta que a raíz de los problemas de tecnología de la oficina remota, se han debido suspender reuniones de proyecto, y las concretadas tenían mucha dificultad para escuchar.

En forma similar, (33) tiene problemas de infraestructura generando retrasos .

En (40) vemos que hay problemas para compartir información.

En (41), no pueden tener acceso remoto a la configuración del cliente: en el primer caso porque las políticas de seguridad no lo permitían, y en el segundo debido a que la configuración no estaba lista.

En (42) ven que editor Wiki complejo para usar, por lo que están en búsqueda de otra herramienta.

Las soluciones reportadas son:

Para las herramientas de colaboración, el aumento de la documentación escrita, generando un incremento del costo de comunicación en cascada. Se desarrolla una herramienta interna de visualización del backlog y medición de performance (25).

Para las video conferencias, (28), decidieron limitar la duración y extender su uso para reuniones informales.

En (33) se implementa un equipo dedicado al mantenimiento de la infraestructura.

En (40) implementan una infraestructura común de IT para compartir la información.

En (41) se efectúan en el primer caso acuerdos temporarios que permiten saltar las políticas de seguridad, y en el segundo se usa la red del proveedor en el cliente.

Problemas de escala:

Para (1), en proyectos grandes, aparecen problemas de coordinación de aquellas funcionalidades compartidas por varios equipos; por otro lado hay imposibilidad de desarrollar las demos.

En (22) se reporta que es una organización muy grande, de catorce equipos con catorce Scrum Masters, generando problemas de flujo de información, y coordinación entre los equipos.

En (30) se analiza la problemática de escalar Scrum para proyectos distribuidos grandes, y se plantea si no se presentan demasiados problemas: coordinación entre los distintos grupos, falta de arquitectura, falta de análisis de requerimientos, problemas de proyectos distribuidos, imposibilidad de implementar demos y retrospectivas comunes.

Las soluciones planteadas son:

En (1), buscar la forma de efectuar las demos al considerarlas muy importantes en el marco del proyecto, no se especifica en qué forma.

No reporta solución el caso (22).

Con respecto a (30), se crea el área de dueños de Producto, dividido en arquitectos del sistema y gerencia de producto, diariamente hacen reuniones Scrum de scrum, que no les resultan muy efectivas, tampoco les resultan las retrospectivas comunes, y no ven solución a las demos.

Problemas de falta de confianza:

En (25) se tiene problemas de performance del equipo offshore, y falta de confianza consecuentemente.

En (35), los miembros del equipo remoto no interactúan directamente con el dueño del Producto, sino a través del manager, hay dificultad en generar confianza ya que el equipo offshore en Bangalore se restringe por sí mismo a adoptar Scrum, desfavoreciéndolos y debilitando la productividad, no existe una clara visión del producto en el equipo offshore, y no hay un balance de poder para los equipos, se tuerce hacia donde residen el dueño de producto y el Scrum Master.

En (39) hay problemas de confianza cuando el Scrum Master o el desarrollador del otro sitio no termina una tarea o no hay noticias de ella.

En (40) se reporta falta de confianza entre los miembros.

Para (41), encuentran problemas para establecer una relación de pares entre onshore y offshore, debido a problemas de motivación y compromiso del equipo offshore.

Las soluciones planteadas:

En (25) se acota que no todos los equipos están preparados para ser proactivos, auto organizados, profundizar el coaching para las reuniones diarias sería la vía para mejorar; para mejorar la performance se implementa un monitoreo del equipo onshore, pero perjudica la performance de éste.

En (35) percibieron que es mejor el trato directo del equipo offshore con el dueño del Producto, éste a su vez, generó mayor documentación y más detallada. Tratan finalmente de esforzarse para hacer participar al equipo offshore.

En (39), buscan lograr conocimiento mutuo y generarse confianza sin especificar la forma.

En (40), tratan que los recursos se puedan contactar cara a cara para reunirse y crear confianza.

En (41) se plantea la necesidad de crear y fomentar una cultura de mutuo respeto.

Problema de comprensión y administración de los requerimientos:

En (20) se tienen problemas en la definición de requerimientos: administración del backlog, cambios en el backlog.

En (23) se reporta que los requerimientos no son entendidos correctamente.

En (26) asimismo, existen también problemas en la definición de requerimientos, en el seguimiento y control de proyectos, en la dinámica de las reuniones diarias.

En (19) se reportan problemas de aprobación de los requerimientos por parte de los business partners, sin especificar su solución.

Para resolverlo, (20) sigue los procedimientos básicos de Scrum. El resto de las fuentes no plantean solución

Problemas de falta de herramientas para soportar el proyecto:

En (20) se observan problemas de actualización del sprint backlog diariamente por lo que se implementa una herramienta (Jira) para uso de las reuniones diarias y control del burndown chart.

Problemas de coordinación:

Vemos en (39) que tienen un equipo de Quality Assurance que prueban lo de los otros, y no saben qué probar; además cuando contratan un empleado nuevo, los del otro lugar no quieren trabajar con él; adicionalmente no resuelven cómo asignar tareas con el equipo de soporte y posterior seguimiento de bugs.

En (42), diferentes feriados religiosos u otros en países diferentes generan problemas de sincronización, sin reportar como solucionarlo.

Tiene (46) problemas con las reuniones diarias y de acoplamiento de los componentes del sistema y actividades.

Las soluciones planteadas:

En (39) es haciendo que los desarrolladores participen del testing y le muestran que probar a los QA y usan Jira para seguimiento de bugs.

En (46) se decide hacer reuniones asincrónicas con las preguntas básicas y distribuir los requerimientos de tal manera que haya mínimo acoplamiento y dependencia entre las tareas de los distintos miembros

Problemas de administración y control de proyecto:

Se plantea en (39) que no se terminan las tareas comprometidas. Nadie se hace responsable.

En (41) detectan que la mezcla junior/senior inicial está distorsionada. Los seniors insumen mucho tiempo enseñando a los juniors; en el segundo caso evaluado, ven incapacidad del coordinador offshore de dedicarle energía y tiempo al proyecto.

En (42) se detectan responsabilidades no claras.

Para resolverlo estas cuestiones:

En (39) se implementa control de project managers o Scrum Master para que trabajen los desarrolladores.

En (41), se reasignaron recursos para proyecto internos para ganar experiencia, y en el segundo proyecto se decidió el reemplazo del coordinador por el project manager onshore

Problemas varios:

En(20)se considera que sprint de cuatro semanas es muy largo, por lo que decidieron recortarlos a dos, además tiene problemas para definir tareas chicas, no explicando cómo resolverlo.

En (22) hay una variedad amplia de inconvenientes: falta de disponibilidad de recursos, no involucración del dueño del producto, no está claro el rol del arquitecto respecto del de dueño de producto; adicionalmente reporta problemas de planificación y de calidad de la documentación.

En (28) se reporta que el Scrum Master no está siempre disponible, con la generación de retrasos, y es el único en detectar que el equipo tiene necesidad de reunirse.

Para (11), las demos son difíciles de organizar, el diseño compartido es muy limitado, para este caso no es posible viajar por el presupuesto escaso, y el tester del equipo está solamente onshore, por lo que hay retrasos es las devoluciones.

En (36) se reporta problemas de visión en el equipo remoto.

En (41) tenemos el problema de tiempos muertos ya que las respuestas del cliente para clarificaciones o dudas son más lentas de lo esperado, por lo que implementaron un mecanismo de clasificación de las preguntas e implementación de una regla para responderlas, y un mecanismo de compensación ante la falta de respuesta.

Un detalle planteado por (42) es el del cansancio con teleconferencias largas.

En (44) se estima que las iteraciones cortas generan problemas de configurationmanagement, control de versiones y documentación.

En (45) vemos que el software no estaba testeado satisfactoriamente, que hay problemas con los builds diarios, donde quizás el que lo originó está durmiendo, y que hay baja información y conocimiento de lo cotidiano respecto de los equipos locales.

Vemos en (43) que existen interrupciones de proyectos externos, hay problemas de comunicación entre la organización y el proyecto, que el equipo no es estable, falta de habilidades técnicas y problemas en testing y documentación.

Tiene (46) cuestiones de imposibilidad de participación del Dueño de producto en las distintas ceremonias, por lo que el equipo tomó decisiones para no depender de él .

En (48), se cuestiona el rol del Scrum Master, que debería extenderse del de auxilio del equipo y PO, y resolución de impedimentos, para tomar como prioridad la de estar atento a la eficiente interacción de los miembros, finalmente, tiene problemas de visualización y manipulación del cuadro de actividades, implementando una planilla compartida para que todos los miembros tengan acceso y la puedan actualizar status.

Artículos de casos donde no reportan problemas:

Finalmente, los artículos (18) y (16), no plantean problemas específicos de los casos analizados, sino que reportan que el uso de Scrum es beneficioso para la problemática de GSD, donde Scrum permite la generación de una red de vínculos entre los equipos, las daily meetings y la entrega permanente genera confianza basada en la performance, el product backlog único permite una visión compartida del proyecto, están claros los roles y responsabilidades, la articulación del trabajo funciona a través de las sprint planning al armar el sprint backlog; para la coordinación, Scrum provee un marco de trabajo; Scrum provee 5 mecanismos diferentes para lograr la comunicación: user stories, Product backlog, burndown charts, rol de PO y de SM, productividad casi similar a los equipos locales, la reuniones de scrum facilitan casi toda la comunicación necesaria, ratio errores menor al de la industria, correctas estimaciones y velocidad estimada

dieron transparencia. Tampoco (38) hace mención de inconvenientes, enumera mejores prácticas utilizadas para este tipo de entornos.

5.- Conclusiones

Después del análisis de todos estos artículos, podemos inferir que para los entornos geográficamente dispersos, el marco de trabajo Scrum logra que inmediatamente se manifiesten los problemas de distancia geográfica, socio-cultural y temporal típicos en estos contextos, y por otro lado proporciona las herramientas procedimentales para detectarlos, por ejemplo a través de las reuniones retrospectivas, mientras los proyectos están activos, y depende de las distintas características de los mismos y de la capacidad del management para poder resolverlos.

Vemos por otro lado que las formas de resolución son sumamente variadas y dependen de cada uno de los contextos en los que los proyectos se desenvuelven.

Así encontramos que varios inconvenientes suelen intentar ser resueltos a través del viaje de los miembros de los equipos entre las distintas locaciones, ya sea de “onshore” a “offshore” o viceversa, solución que está fuertemente restringida por el presupuesto del proyecto, y en consecuencia no siempre es posible.

Otro problema que se torna severo en este marco de trabajo es el de la diferencia horaria entre los equipos, fundamentalmente cuando es amplia, pues al exigirse una importante participación en reuniones, afecta la vida personal de los miembros de los equipos cuando tienen que trabajar frecuentemente fuera de sus horarios normales, y las soluciones planteadas son en algunos casos discutibles.

Esta frecuente necesidad de comunicación asimismo requiere de la disponibilidad de una infraestructura tecnológica que suministre servicios de web conference, buena calidad de sonido para teleconferencias, entornos con herramientas para compartir información, que en muchas oportunidades no se tienen o son limitadas, especialmente cuando se trabaja con equipos situados en países emergentes de Asia o Latinoamérica.

Otra cuestión que podemos destacar es que la mayoría de los casos se remiten a un grupo “onshore” que trabaja con otro equipo “offshore”. De allí que los problemas reportados son en general referidos a la relación de los miembros como equipo, estando cada uno de ellos en una misma oficina. No se reportan prácticamente casos en donde parcial o totalmente los miembros de cada equipo están distribuidos geográficamente, por ejemplo en la modalidad de “home office”, en donde estaría ausente en su totalidad toda comunicación informal cotidiana cara a cara, complejizándose el marco de trabajo, siendo que está fuertemente basado en la continua comunicación entre los participantes. El autor de este trabajo estima que esta alternativa de trabajo, debería profundizarse en su análisis para obtener información que permita evaluarla.

Por otro lado, los artículos que analizan la problemática de cambio de escala de proyecto, concluyen que Scrum tiene muchos inconvenientes que obligan a modificarlo de alguna manera pues se hace imposible implementarlo según el marco teórico definido.

De todas maneras, muchos de los reportes afirman que el marco de trabajo en sí mismo resuelve o bien mitiga los problemas intrínsecos de GSD.

Como consecuencia de este estudio, el autor considera como importantes líneas de investigación básicamente dos temas:

- Scrum bajo la modalidad de “home office” total, donde funcionan los denominados equipos virtuales, o sea que hay nula interacción personal cara a cara entre los miembros del equipo de desarrollo.
- Problemática de escala, donde los casos de estudio relevados no consiguen resolver satisfactoriamente este tema, y en donde este marco de trabajo suele ser cuestionado por la industria

6.-Bibliografía

1. Paasivaara M, Heikkilä VT, Lassenius C. Experiences in scaling the product owner role in large-scale globally distributed scrum. En: Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on. IEEE; 2012. p. 174-8.
2. Takeuchi H, Nonaka I. The new new product development game. Harv Bus Rev. 1986;64(1):137-46.
3. Schwaber K, Sutherland J. La Guía de Scrum. La guía definitiva de Scrum. Las Reglas del Juego. 2013.
4. Manifiesto por el Desarrollo Ágil de Software [Internet]. Recuperado a partir de: <http://agilemanifesto.org/iso/es/>
5. Hossain E, Babar MA, Verner J. Towards a framework for using agile approaches in global software development. En: Product-Focused Software Process Improvement. Springer; 2009. p. 126-40.
6. Deemer P, Benefield G, Larman C, Vodde B. Información básica de SCRUM. California: Scrum Training Institute; 2009.
7. Carmel E. Global Software Teams. Collaborating Across Borders and Time Zones. Prentice Hall PTR; 269 p.
8. Luz M, Gazineu D, Teófilo M. Challenges on adopting scrum for distributed teams in home office environments. World Acad Sci Eng Technol. 2009;59:308-11.
9. Herbsleb JD, Moitra D. Global software development. Softw IEEE. 2001;18(2):16-20.
10. Lipnack J, Stamps J. Virtual teams: people working across boundaries with technology. 2nd ed. New York: Wiley; 2000. 317 p.
11. Sutherland J, Schoonheim G, Kumar N, Pandey V, Vishal S. Fully distributed Scrum: Linear scalability of production between San Francisco and India. En: Agile Conference, 2009 AGILE'09. IEEE; 2009. p. 277-82.
12. Bannerman PL, Hossain E, Jeffery R. Scrum practice mitigation of global software development coordination challenges: A distinctive advantage? En: System Science (HICSS), 2012 45th Hawaii International Conference on. IEEE; 2012. p. 5309-18.
13. Leffingwell D. Scaling software agility: best practices for large enterprises. Upper Saddle River, NJ: Addison-Wesley; 2007. 349 p.

14. Hossain E, Bannerman PL, Jeffery R. Towards an understanding of tailoring scrum in global software development: a multi-case study. En: Proceedings of the 2011 International Conference on Software and Systems Process. ACM; 2011. p. 110-9.
15. Hossain E, Babar MA, Verner J. How Can Agile Practices Minimize Global Software Development Co-ordination Risks? En: O'Connor RV, Baddoo N, Gallego JC, Muslera RR, Smolander K, Messnarz R, editores. Software Process Improvement. Springer Berlin Heidelberg; 2009. p. 81-92.
16. Sutherland J, Schoonheim G, Rustenburg E, Rijk M. Fully distributed scrum: The secret sauce for hyperproductive offshored development teams. En: Agile, 2008 AGILE'08 Conference. IEEE; 2008. p. 339-44.
17. Drummond B, Unson JF. Yahoo! Distributed Agile: Notes from the world over. En: Agile, 2008 AGILE'08 Conference. IEEE; 2008. p. 315-21.
18. Pries-Heje L, Pries-Heje J. Why Scrum works: A case study from an agile distributed project in Denmark and India. En: Agile Conference (AGILE), 2011. IEEE; 2011. p. 20-8.
19. Cristal M, Wildt D, Prikladnicki R. Usage of Scrum practices within a global company. En: Global Software Engineering, 2008 ICGSE 2008 IEEE International Conference on. IEEE; 2008. p. 222-6.
20. Berczuk S. Back to basics: The role of agile principles in success with an distributed scrum team. En: Agile Conference (AGILE), 2007. IEEE; 2007. p. 382-8.
21. Jensen B, Zilmer A. Cross-continent development using Scrum and XP. En: Extreme Programming and Agile Processes in Software Engineering. Springer; 2003. p. 146-53.
22. Smits H, Pshigoda G. Implementing scrum in a distributed software development organization. En: Agile Conference (AGILE), 2007. IEEE; 2007. p. 371-5.
23. Paasivaara M, Durasiewicz S, Lassenius C. Distributed agile development: Using Scrum in a large project. En: Global Software Engineering, 2008 ICGSE 2008 IEEE International Conference on. IEEE; 2008. p. 87-95.
24. Karsten P, Cannizzo F. The creation of a distributed agile team. En: Agile Processes in Software Engineering and Extreme Programming. Springer; 2007. p. 235-9.
25. Cottmeyer M. The good and bad of Agile offshore development. En: Agile, 2008 AGILE'08 Conference. IEEE; 2008. p. 362-7.
26. Sutherland J, Viktorov A, Blount J. Adaptive Engineering of Large Software Projects with Distributed/Outsourced Teams. En: Proc International Conference on Complex Systems, Boston, MA, USA. 2006. p. 25-30.

27. Sutherland J, Viktorov A, Blount J, Puntikov N. Distributed scrum: Agile project management with outsourced development teams. En: System Sciences, 2007 HICSS 2007 40th Annual Hawaii International Conference on. IEEE; 2007. p. 274a - 274a.
28. TIETZ V, MÖNCH A. Facing Fake-to-Fake: Lessons Learned from Distributed Scrum. 2015;
29. Sutherland J, Schoonheim G, Rijk M. Fully distributed scrum: Replicating local productivity and quality with offshore teams. En: System Sciences, 2009 HICSS'09 42nd Hawaii International Conference on. IEEE; 2009. p. 1-8.
30. Paasivaara M, Lassenius C. Scaling scrum in a large distributed project. En: Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on. IEEE; 2011. p. 363-7.
31. Kussmaul C, Jack R, Sponsler B. Outsourcing and offshoring with agility: A case study. En: Extreme Programming and Agile Methods-XP/Agile Universe 2004. Springer; 2004. p. 147-54.
32. Paasivaara M, Durasiewicz S, Lassenius C. Using scrum in a globally distributed project: a case study. *Softw Process Improv Pract.* 2008;13(6):527-44.
33. Hogan B. Lessons learned from an extremely distributed project. En: Agile Conference, 2006. IEEE; 2006. p. 6 - pp.
34. Esbensen M, Bardram JE. Tool Support for Globally Distributed Scrum. En 2014. p. 3.
35. Therrien E. Overcoming the challenges of building a distributed agile organization. En: Agile 2008 Conference. IEEE; 2008. p. 368-72.
36. Summers M. Insights into an agile adventure with offshore partners. En: Agile, 2008 AGILE'08 Conference. IEEE; 2008. p. 333-8.
37. Uy E, Ioannou N. Growing and sustaining an offshore Scrum engagement. En: Agile, 2008 AGILE'08 Conference. IEEE; 2008. p. 345-50.
38. Sureshchandra K, Shrinivasavadhani J. Adopting agile in distributed development. En: Global Software Engineering, 2008 ICGSE 2008 IEEE International Conference on. IEEE; 2008. p. 217-21.
39. Cho J. Distributed Scrum for large-scale and mission-critical projects. *AMCIS 2007 Proc.* 2007;235.
40. Vax M, Michaud S. Distributed Agile: Growing a practice together. En: Agile, 2008 AGILE'08 Conference [Internet]. IEEE; 2008 [citado 3 de febrero de 2016]. p. 310-4. Recuperado a partir de: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4599497

41. Sadun C. Scrum and Global Delivery: Pitfalls and Lessons Learned. En: Agility Across Time and Space [Internet]. Springer; 2010 [citado 3 de febrero de 2016]. p. 71-89. Recuperado a partir de: http://link.springer.com/chapter/10.1007/978-3-642-12442-6_5
42. Paasivaara M, Durasiewicz S, Lassenius C. Using scrum in distributed agile development: A multiple case study. En: Global Software Engineering, 2009 ICGSE 2009 Fourth IEEE International Conference on [Internet]. IEEE; 2009 [citado 3 de febrero de 2016]. p. 195-204. Recuperado a partir de: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5196933
43. Kilpala M, Kärkkäinen T. Distributed scrum when turning into maintenance: A single case study. En: Second International Conference on Computer Science and Information Technology (COSIT 2015) [Internet]. 2015 [citado 3 de febrero de 2016]. p. 55-67. Recuperado a partir de: <http://airccj.org/CSCP/vol5/csit53706.pdf>
44. Noordeloos R, Manteli C, Van Vliet H. From RUP to Scrum in global software development: A case study. En: Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on [Internet]. IEEE; 2012 [citado 3 de febrero de 2016]. p. 31-40. Recuperado a partir de: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6337395
45. Hole S, Moe NB. A case study of coordination in distributed agile software development. En: Software process improvement [Internet]. Springer; 2008 [citado 3 de febrero de 2016]. p. 189-200. Recuperado a partir de: http://link.springer.com/chapter/10.1007/978-3-540-85936-9_17
46. Soares FSF, de Sousa Mariz LMR, Cavalcanti YC, Rodrigues JP, Neto MG, Bastos PR, et al. Adoção de SCRUM em uma Fábrica de Desenvolvimento Distribuído de Software. Cent Informática—UFPE—Recife—PE—Brasil [Internet]. 2007 [citado 3 de febrero de 2016]; Recuperado a partir de: <http://afinaimagem.googlecode.com/svn/trunk/PGP/Rodrigo/Documentacao%20Scrum/adocao-de-scrum-em-uma-fabrica-de-desenvolvimento.pdf>
47. Noordeloos R. AGILE SOFTWARE DEVELOPMENT IN A GLOBALLY DISTRIBUTED ENVIRONMENT. 2012 [citado 3 de febrero de 2016]; Recuperado a partir de: http://www.cs.vu.nl/~hans/publications/y2012/ICGSE2012-RUPtoScrum/MasterThesis_RNoordeloos%20.pdf
48. Chalegre V gínia C, Santos WB, de Souza LO, Muñoz HJ.,Romer o de Lemos Meir a S. Estudo de Caso da Utilização de Scrum no Desenvolvimento Distribuído de Software. En PUC - RS; 2010. p. 129-36.