

LIGHTNING SCRUM: ADAPTACIÓN DEL FRAMEWORK DE TRABAJO ÁGIL SCRUM A LA REALIDAD DE LOS EMPRENDIMIENTOS TI DE LA REGIÓN DE ARICA Y PARINACOTA

LIGHTNING SCRUM: ADAPTATION OF THE FRAMEWORK OF AGILE SCRUM WORK TO REALITY OF ENTERPRISES OF THE REGION ARICA AND PARINACOTA

Patricio Arias Acosta*
Dr. Ricardo Valdivia Pinto**

*Gerente General y Fundador de ARID Soluciones Inteligentes. Arica, Chile.
Email: patricio.arias@arid.cl

** Doctor en Ciencias de la Ingeniería. Académico de la Escuela Universitaria de Ingeniería Industrial, Informática y Sistemas (EUIIS) de la Universidad de Tarapacá (UTA) Arica, Chile. Email: rvaldivi@uta.cl

RESUMEN

Las tendencias actuales del mercado y la realidad en la que la región de Arica y Parinacota está inmersa, obliga a las pequeñas empresas de desarrollo a ser eficientes e innovadoras al momento de desarrollar soluciones de software. Si bien, el framework Scrum entrega herramientas para la mayoría de los casos y maneras de trabajar, la realidad de las empresas de desarrollo entre USA y Chile difiere en madurez y en escala.

Este trabajo propone medidas de adaptación de este framework acorde a la realidad de los emprendimientos (Startups) del área TI de la región, enfrentando los problemas que se han encontrado en casos reales de empresas de la zona.

Palabras clave: Desarrollo ágil, Desarrollo de Software, Scrum, Arica y Parinacota, Emprendimiento, Startup.

ABSTRACT

The current market trends and the reality in the region of Arica and Parinacota is immersed, forces small development companies to become efficient and innovative when developing software solutions. While the Scrum framework provides tools for most cases and ways of working, the reality of business development between the US and Chile differs in maturity and scale.

This paper proposes measures to adapt this framework according to the reality of the Startups (referred to IT solutions) in the region, facing the problems found in real business cases in the area.

Keywords: Agile Development, Software Development, Scrum, Arica and Parinacota, Entrepreneurship, Startup.

INTRODUCCIÓN

El mundo actual camina y evoluciona a una velocidad impresionante. El ritmo de esta evolución y la constante generación de información obligan a que las metodologías utilizadas y requeridas por empresas se sometan a una constante actualización. La Ingeniería de Software es, dentro de la familia de las Ingenierías, una rama joven y aún inmadura en muchos aspectos. Para poner en perspectiva, la Ingeniería en la construcción de edificaciones es casi tan antigua como el mismo hombre, con una constante evolución formal desde épocas antiguas.

Hoy en día, la Ingeniería Mecánica es capaz predecir con detalle si una estructura es sustentable o no. De este mismo modo, varias ramas de la Ingeniería son capaces de simular, prever, analizar y estimar resultados antes de comenzar. Esta es la evolución que debe alcanzar la Ingeniería de Software. La Ingeniería del Software data de mediados del siglo XX y durante su corta existencia se ha buscado persistentemente la “bala de plata” (aquel santo grial que permita asegurar buenos resultados de manera teórica por sobre el conocimiento empírico, descrito como tal en 1986 en el libro de Fred Brooks “No Silver Bullet”), sin resultados positivos.

Aún, al día de hoy, no es posible predecir con total seguridad el resultado de un proyecto de construcción de Software de una complejidad mediana-elevada (Brooks, 1987). A falta de esta “fórmula secreta” aún no descubierta, es que se han propuesto

metodologías para llevar a cabo esta Ingeniería de una manera eficaz y eficiente, las cuales se basan principalmente en el conocimiento empírico, tanto en experiencias el mercado como en el laboratorio. Las primeras metodologías de desarrollo de Software imitaron las de sus ramas hermanas, es decir la metodología en cascada. (Brooks, 1987).

En la década de 1960, en la llamada “Crisis del Software”, es que se proponen medidas radicales para afrontar el alicaído panorama por el que atravesaba la industria del Software, en donde la mayoría de los proyectos informáticos fracasaban total o parcialmente. Es ahí donde se producen los primeros movimientos por desarrollar metodologías ágiles en la Ingeniería de Software. (Agile Manifesto. (Recuperado el 07-01-15). <http://www.agilemanifesto.org/>)

A la fecha, las metodologías ágiles de desarrollo de Software, se han popularizado y estandarizado, partiendo primero en pequeñas empresas/proyectos, pero hoy utilizadas en las empresas más grandes y productivas del mundo.

Scrum es un Framework (conjunto de herramientas y directrices) para insertar el pensamiento, los valores y principios ágiles dentro de una organización y hacerlos prácticos. Hoy, Scrum es una de las estrategias utilizadas más populares del mundo y diversas empresas -tecnológicas, automovilísticas, universidades y militares- alrededor del mundo lo utilizan (Agile Alliance. (Recuperado el 07-01-15). <http://www.agilealliance.org/>), no tan solo en el ámbito del desarrollo de Software, sino en todo aquel

desarrollo que tenga al producto como principal medida de éxito. (Stacey, 1999)

MATERIALES Y MÉTODOS

SCRUM

Para comenzar hay que definir ¿Qué es Scrum? Scrum es un framework de trabajo ágil y no una metodología específica. Esto es porque el fundamento principal de Scrum es analizar el ¿Por qué? y no el ¿Cómo?, en otras palabras, Scrum se preocupa de los cimientos de las prácticas ágiles en algún grupo o equipo.

Las raíces de Scrum vienen dadas por los autores Hirotaka Takeuchi & Ikojuri Nonaka (The Scrum Alliance® Community. Recuperado el 07-01-15. <http://www.scrumalliance.org>) , quienes se basan en un escenario inestable para la creación de un producto, en donde el personal involucrado en su creación y producción se debe auto-organizar y cumplir con las expectativas del cliente, no importando que el mercado vaya cambiando al mismo tiempo que se desarrolle el producto.

El sprint es el concepto más relevante en Scrum, una ventana de tiempo fijo e inamovible en donde el equipo desarrollará parte del producto que debe ser funcional y contener las características más importantes. Se recomienda utilizar Sprints que vayan entre las 2 y 4 semanas.

Desde la visión del producto se desprenden las user stories (historias de usuario), que describen las características que tendrá el sistema y que harán valioso al mismo. Estas user stories se almacenan en el

product backlog. El product owner (encargado de reunir las necesidades del cliente y generar la visión del producto) debe organizar y priorizar las user stories para que el equipo vaya desarrollando primero aquellas características que son más valiosas.

El equipo de desarrollo se organiza en una reunión llamada sprint planning, en donde las user stories se descomponen en una o más tareas, dividiéndolas y simplificándolas hasta el punto que puedan medirse mediante algún método propio de la organización o según algunas propuestas teóricas. Cada miembro del equipo hace un compromiso y se responsabiliza de las tareas que tendrá listas al final del sprint.

Al finalizar el sprint debe existir siempre un producto entregable. Se realiza un sprint review en donde también puede participar el cliente, para que vaya teniendo una experiencia desde temprano con su software (o producto en general), lo que hará que entregue retroalimentación lo antes posible. Luego el equipo realiza una reunión llamada sprint retrospective en donde se analiza qué anduvo bien y qué anduvo mal, para mejorar para el siguiente sprint.

Cada día de trabajo dentro del sprint tiene una actividad llamada daily scrum que es una reunión de 15 minutos, conducida por el scrum master (el encargado de salvaguardar el proceso completo de Scrum), en donde cada miembro responde las siguientes preguntas relacionadas al trabajo dentro del sprint:

- ¿Qué hiciste desde la última reunión?

- ¿Cuál es el plan para hoy?
- ¿Qué obstáculo tienes?

Cada día se deben actualizar los indicadores de avance, como el project burndown o burnup chart. Estos artefactos indican el nivel de avance del proyecto o cuánto trabajo queda aún por hacer. Scrum involucra equipos pequeños. Se dice que el equipo ideal va desde 5 hasta 9 personas. En el equipo se busca gente capaz de entregar funcionalidad real con gran solidez y que sea capaz de trabajar día a día colaborativamente. Es importante destacar que Scrum será efectivo si y sólo si su equipo está completamente comprometido con sus roles. Se recomienda que el equipo se encuentre geográficamente en el mismo lugar.

RESULTADOS Y DISCUSIÓN

En Chile existe un fuerte impulso al emprendimiento e innovación. Desde el año 2010, se han potenciado las instituciones que fomentan el emprendimiento y la creación de empresas en regiones, a través de programas de desarrollo estratégico regionales (PER). Las regiones extremas –entre ellas la región de Arica y Parinacota- son las más beneficiadas por estos instrumentos. En Chile, al igual que en el mundo entero, se vive el llamado “Boom de las StartUps”. Además, la popularidad de aplicaciones móviles y plataformas de distribución de Software como la AppStore provocaron que la industria cambiara totalmente de orientación. Desde que ocurrió esto, el modelo de negocios del Software cambió radicalmente, comenzaron a aparecer aplicaciones de 1 dólar y servicios

informáticos de bajo costo que apostaban a la clientela masiva más que a la selectiva y cautiva.

Por otro lado, en la región de Arica y Parinacota, el mercado de las soluciones TI es aún inmaduro, pero presenta un crecimiento positivo desde hace un par de años. Un mercado inmaduro en este aspecto se puede manifestar de la siguiente manera:

- Las empresas no invierten en soluciones TI o no tienen un presupuesto que las considere de manera regular.
- Se tiende a confundir una solución TI con soporte y mantenimiento de máquinas y hardware.
- La labor del profesional encargado de entregar soluciones TI no es valorada (ni valorizada) como corresponde, en comparación con otras regiones en donde este mercado se encuentra más maduro.
- Los clientes no se comprometen ni responden responsablemente.
- Las barreras de entrada al mercado son muy bajas. No hay exigencia de calidad.
- No existe información o conocimiento.
- Muy pocas empresas tienen un departamento TI.

Scrum es un modo de trabajo bastante potente y que se refleja en un aumento de productividad del equipo desarrollador de aplicaciones informáticas (en general, de cualquier producto o servicio). Sin embargo, para poder sacar máximo beneficio ofrecido por Scrum, hay ciertas circunstancias que se deben cumplir:

- El equipo debe tener cierto grado de habilidad.
- Los horarios de trabajo deben ser fijos y regulares.
- Los equipos deben estar involucrados en un solo proyecto a la vez.
- Todos los miembros que aportan al desarrollo deben estar cohesionados.

MODELADO DE LA PROPUESTA

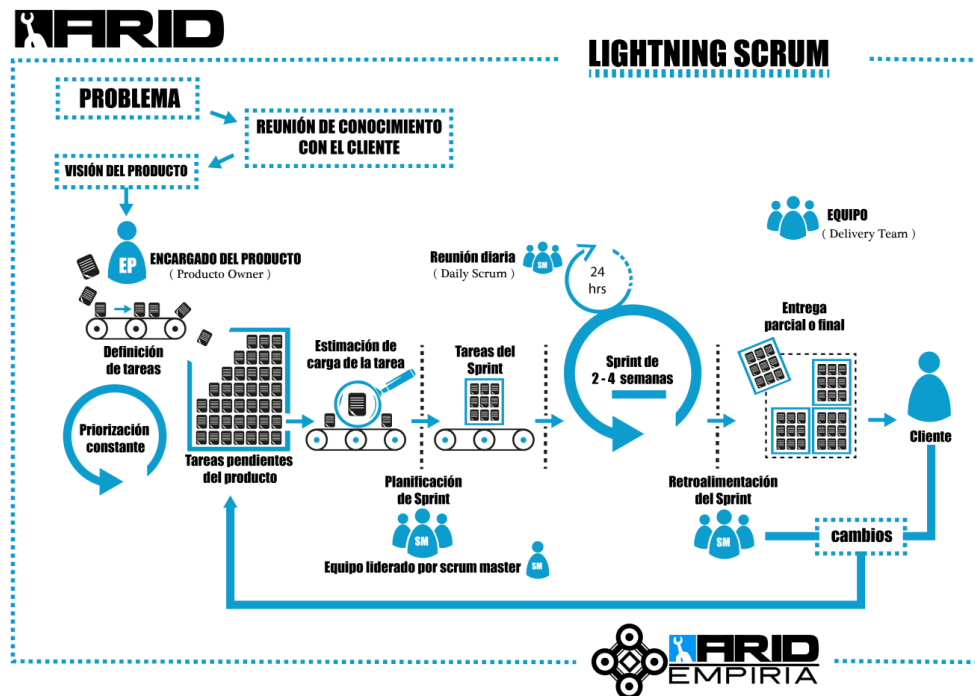
A continuación se presenta una solución para que pequeñas empresas emergentes puedan adoptar Scrum, y en general cualquier metodología ágil, de una manera más efectiva y con resultados más controlables. Lo presentado a continuación es el resultado de la experiencia recopilada a través de proyectos informáticos reales, en la empresa ARID Soluciones Inteligentes, junto a empresas de emergentes del rubro TI, y con clientes de diversa índole de la región de Arica y Parinacota.

Lightning Scrum o Scrum Relámpago (ver figura 1) es el modelo adaptado desde el original Scrum tomando en cuenta:

- La realidad de las empresas emergentes del rubro TI en la Región de Arica y Parinacota
- Experiencia con Scrum en proyectos informáticos de índole pequeña, mediana y mediana-grande.
- Adaptación de la teoría según vivencias y puntos de vista de desarrolladores de la región.
- Caracterización de los clientes potenciales y reales de la región.

Lightning Scrum adopta la metodología formal propuesta por Scrum y la simplifica a la realidad de las empresas desarrolladoras de Software (considerando las empresas TI) de la región de Arica y Parinacota. Si bien se mantienen los roles, los rituales y la mayoría de los componentes, la manera de operar está simplificada. Se agrega una reunión de conocimiento con el cliente y el ciclo constante de priorizar el product backlog. Además, después de cada sprint se realiza una sprint review que va acompañada de un demo, para que el cliente ponga sus manos en el sistema y pueda entregar retroalimentación real.

Figura 1. Funcionamiento de Lightning Scrum



contacto@arid.cl

www.arid.cl

Fuente: ARID

En el ámbito de este trabajo, se realizó una comparación de dos casos reales de construcción de software por la empresa ARID. En el primero de los casos, la empresa intentó utilizar el Scrum real, pero a causa de no manejar el modelo correctamente la implementación no fue efectiva. En el segundo caso –y con un año de diferencia desde el proyecto anterior– se presenta un proyecto de software desarrollado con un Scrum efectivo: Lightning Scrum.

Durante los años 2012 y 2013 ARID Soluciones Inteligentes, colaboró en un proyecto de software que consistía en la creación de un sistema informático ERP para un

conglomerado empresarial de la región de Arica y Parinacota. Este proyecto contemplaba desde el diseño conceptual del sistema hasta el servicio de post-venta y garantía, pasando por las fases de desarrollo, implementación, marcha blanca, rectificación, etc. El proyecto contaba con un tiempo esperado de 12 meses, para lo cual se formó un equipo de 5 personas.

Para el desarrollo del proyecto se planificó la utilización de Scrum, pero el desconocimiento real que tenía la empresa sobre este framework y la inexperiencia en este tipo de proyectos hizo que la empresa cometiera errores al aplicarlo, como por ejemplo:

- Intentar seguir al pie de la letra la metodología propuesta por Scrum (orientada para empresas más experimentadas y de mayor tamaño).
- La utilización de Software de Scrum que no permite flexibilidad en el modo de trabajar.
- No respetar los compromisos de tiempos y entregas dentro del equipo, sino más bien ir adaptándolos de tiempo en tiempo.
- Priorizar los compromisos con el cliente más que los compromisos internos en el equipo.

Las deficiencias que se notaron fueron algunas prácticas propias del “legado de la cascada”, como la poca comunicación con los clientes, mostrar los avances muy de vez en cuando y otros. Los principales problemas se ocasionaron debido a que el equipo no supo implementar Scrum en su funcionamiento, y tuvo que improvisar para cambiar de una modalidad a otra sobre la marcha del proyecto. Además, las prácticas no ágiles provocaron que en la fase de marcha blanca aparecieran demasiados cambios en algunos módulos, lo que hizo imposible arreglar todo y se tuvo que priorizar los módulos más usados.

Por otro lado, en el año 2014 ARID Studios (parte de la empresa ARID Soluciones Inteligentes www.arid.cl), luego de haber adoptado formalmente Scrum, entrenándose a través de capacitaciones y cursos, y habiendo adaptado éste a una versión propia, comenzó el desarrollo de un nuevo producto de software llamado FERIAMarket (www.feriamarket.cl), una versión de ERP ágil para la gran masa de empresas MIPYME en el mercado

chileno y latinoamericano. Para este desarrollo se utilizó un equipo muy similar de desarrollo, que constaba de cinco integrantes.

Para efectos de este desarrollo –y a diferencia del caso contado anteriormente- el equipo si se organizó correctamente con su versión de Scrum (Lightning Scrum), respetando sus acuerdos y manteniendo a las partes siempre presentes, viendo el nivel de desarrollo en cada sprint review (cada dos semanas), abiertos siempre a nuevas características y mejoras, y repriorizando constantemente el product backlog para maximizar el valor del software para el cliente.

En comparación, FERIAMarket es un software más grande y completo que el primero realizado por ARID Soluciones Inteligentes, realizado en menos tiempo y con una conformidad mayor por parte del cliente. La satisfacción del cliente provocó una reinversión de fondos para lograr más funciones en el Software, por lo cual FERIAMarket está nuevamente como proyecto de desarrollo en ARID Studios.

CONCLUSIONES

- Scrum solamente es útil cuando se enfrenta con problemas de magnitud media-elevada. Sobre todo cuando enfrente algún problema que se complica para abordar en equipo.
- No existe una versión unificada de Scrum. Si se realizara una visita por varias empresas que utilizan Scrum se encontrarían variantes y pequeños matices

propios de cada una de las empresas. Esta es la idea de Scrum desde su base: no ser una metodología (para seguir al pie de la letra) sino más bien un framework de trabajo que permita incorporar los valores y la ideología del trabajo ágil.

- Adoptar Scrum es un proceso incremental. El equipo debe estar consciente que no logrará dominar metodologías ágiles de inmediato sino que será un proceso que ira madurando en el tiempo.
- Es necesario que las empresas emergentes del rubro TI en la región de Arica y Parinacota se muestren al mercado, pues de nada sirve que se perfeccionen si el mercado las desconoce. En una región con las características de Arica y Parinacota, en donde existe una creciente demanda tecnológica por parte de las empresas y servicios y una alta interacción transfronteriza, es demandante que el mercado de soluciones TI se consolide en la región, para que de manera conjunta, todas las empresas se puedan potenciar y mejorar su valor.

REFERENCIAS BIBLIOGRÁFICAS

Brooks, F.,1987. "No Silver Bullet – Essence and Accident in Software Engineering", Univerisity publishing.

Agile Manifesto. (Recuperado el 07-01-15).
<http://www.agilemanifesto.org/>

Agile Alliance. (Recuperado el 07-01-15).
<http://www.agilealliance.org/>

Stacey. R.,1999. "Strategic Management & Organizational Dynamics: The Challenge of Complexity".

The Scrum Alliance® Community. Recuperado el 07-01-15.
<http://www.scrumalliance.org>

Ogunnaike, Ray, 1992. "Process Dynamics, Modeling and Control", Oxford University Press.

Takeuchi, H., Nonaka, I., 1986, "The New New Product Development Game", Harvard Business Review, Jan/Feb.

Wegner, P., 1997, "Why Interaction is more powerful than algorithms", Brown University CACM.

Ziv, H., 1996, "The Uncertainty Principle in Software Engineering", ICSE'97.

W. S. Humphrey, 1995, "A Discipline for Software Engineering", Addison-Wesley.

Memorias Corfo 2010-2013. (Recuperado el 07-01-15).
<http://www2.corfo.cl/memoria2010-2013/>

Montoya Suárez, O., 2004. "Schumpeter, Innovación y

Determinismo Tecnológico”,
Scientia et Technica Año X,
Nro.

Prieto, L. September 2014, “Welcome
to Chilecon Valley, Latin
America’s New Center of

Innovation”, Forbes Brand
Voice.

Leavitt, H. J., Whisler T. L.,
1958. “Management in the
1980’s”, Harvard Business
Review.