



**UNIVERSIDAD
TORCUATO DI TELLA**

**EL CABALLO DE TROYA DE LA
TRANSFORMACIÓN: SCRUM.
TESIS MBA**

Autor: Martín Alejandro Poncio

Tutor: Pablo Roccatagliata

Agradecimientos

A Francisca por estar siempre

A mis compañeros del MBA por enseñarme algo nuevo todos los días

A mi mentores, Ingrid, Tomas, Maria y Martín, por colaborar con este proyecto

A Pablo Roccatagliata y Vanesa Walsh por ayudarme a cerrar este proceso

PALABRAS CLAVES

- Agilidad
- Scrum
- Marcos de Trabajo
- Transformación Digital
- CHAOS Report

Resumen

En los últimos 30 años los Marco de Trabajo Ágiles han ido incrementando su grado de popularidad. En la década del 90, comenzaron a desarrollarse debido a la baja tasa de éxito de los modelos predictivos adaptados a la industria del Software. Durante ese periodo distintos marcos de “livianos” fueron desarrollándose de forma independiente, hasta consolidarse en el año 2001 bajo un mismo paraguas de principios y valores: el manifiesto Ágil.

Desde comienzos del siglo XXI Scrum se ha convertido en el Framework Ágil más popular y se ha vuelto en uno de los marcos de trabajo predilectos de las empresas que han emprendido sus procesos de transformación digital. Ahora bien, Scrum, al igual que los otros marcos de trabajo, requiere de ciertas características del entorno para poder maximizar su adopción. Es por esto que las empresas que buscan implementar Scrum se encuentran en la disyuntiva de modificar el entorno laboral para operar con Scrum, o hacer una implementación más superficial, donde no se rompa el entorno organizacional y cultural.

Durante este trabajo se buscará entender el contexto en el que surge la Agilidad y cuales son las características de Scrum que lo llevaron a convertirse en el marco de trabajo más popular al momento de iniciar transformaciones digitales. Luego entrevistamos a 4 referentes para conocer si desde su experiencia, implementar Scrum ha ayudado a movilizar procesos de Transformación Digital y mejorar la performance de los proyectos de software.

Índice

Agradecimientos	1
Resumen	3
Índice	4
Introducción	5
Capítulo 1 - El Manifiesto Ágil	6
1.2 Del Caos a la complejidad	6
1.2. Marcos Emergentes	7
1.2.1 Dynamic Systems Development Method	7
1.2.2 Adaptive Software Development	8
1.2.3 Crystal Clear	9
1.2.4 Scrum	10
1.2.5 Extreme Programming	12
1.3 Snowbird	14
Capítulo 2 - Scrum como palanca de la transformación	17
2.1 ¿Por qué se transforman las empresas?	17
2.2 El origen de la tradición	18
2.3 Complejidad y empirismo	20
2.4 Scrum, agilidad en la práctica	22
2.4.1 Roles	22
2.4.2 Eventos	23
2.4.3 Artefactos de Scrum	24
2.5 ¿Cómo impactó la Agilidad en la industria del Software?	24
Capítulo 3- Investigación	28
3.1 Objetivo	28
3.2 Entrevistados	28
3.3 Hallazgos	29
Capítulo 4 - Conclusiones	31
Referencias	32
ANEXO	33
Entrevista	34
Notas Adicionales	40

Introducción

Durante las últimas dos décadas, la mayoría de las grandes empresas han comenzado un camino de transformación digital, basándose en la agilidad como forma de organizarse, ahora bien. ¿Realmente se han transformado las grandes empresas?.

La Agilidad nace para dar respuesta a los problemas que había en la década del 90' para construir software. Si lo pensáramos como una enfermedad podemos decir que los síntomas eran que los productos “se rompían”, que los tiempos no se cumplían y los costos se triplicaron (Standish Group, 1995). En los casos más extremos, pasaba eso y además lo que construía no servía o no se terminaba usando.

En febrero del 2001, 17 referentes de la industria se reúnen en Snowbird, con el objetivo de descubrir las mejores formas de desarrollar software (Agile Alliance, 2001), dando como resultado de ese encuentro los 4 valores y 12 principios que conforman el Manifiesto Ágil.

Durante los últimos años, muchas empresas han iniciado su camino de Transformación Digital, intentando apoyarse en el Manifiesto y las buenas prácticas promovidas por sus firmantes. Por este motivo no debería llamar la atención que uno de los marcos de trabajo más utilizados para dar inicio a las transformaciones es Scrum, desarrollados por Schwaber y Sutherland, ambos firmantes del manifiesto.

Ahora bien, la implementación del marco de trabajo no parece ser condición suficiente para transformar una organización, por eso se impone la siguiente pregunta ¿Cuál es el dilema de fondo que tienen los Scrum Masters o Agile Coaches? Muy simple, en el ¿Para qué se utiliza Scrum, para acelerar la entrega o para cambiar el contexto de los equipos de desarrollo?

Durante este trabajo buscaremos explorar porque si se utiliza Scrum para cambiar el contexto de trabajo de los equipos de desarrollo puede ser un gran motor para impulsar la Transformación Digital, pero si el foco está puesto solamente en

acelerar el *Delivery*, no termine generando el impacto necesario para transformar la cultura y la estructura de la organización.

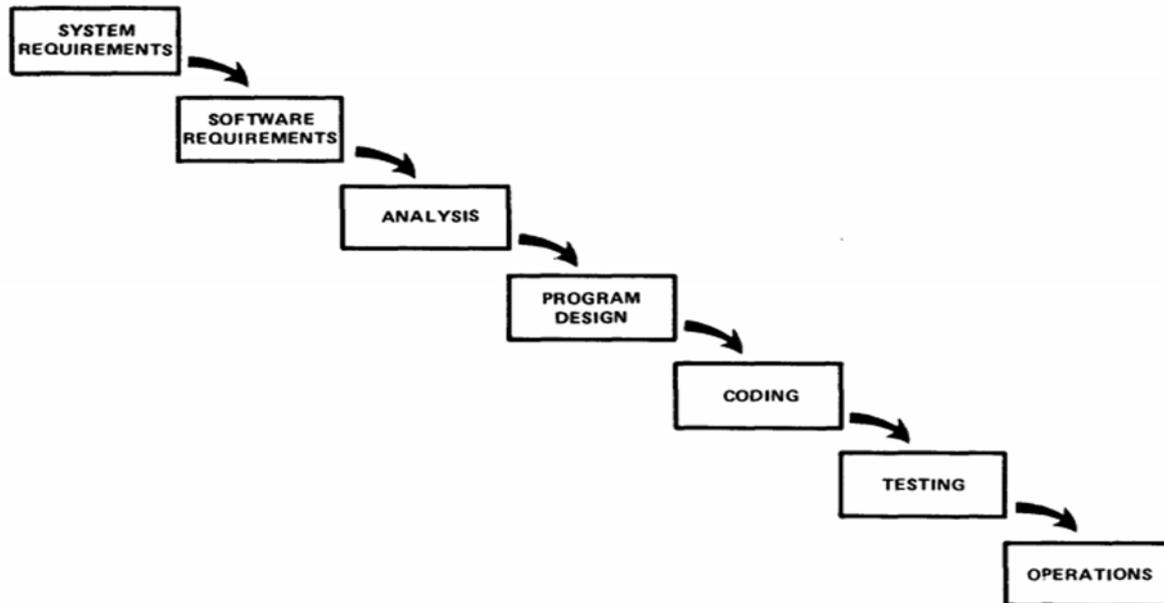
Capítulo 1 - El Manifiesto Ágil

1.1 El origen de la tradición

Como vimos en el primer capítulo la Agilidad nace como respuesta al modelo de trabajo que se volvió dominante para construir software a partir de la década 70, el famoso "Método en cascada". En el año 1970 Winston Royce publicó un paper titulado *Managing the development of large software systems: concepts and techniques*

con el propósito conceptualizar como se estaba construyendo software en ese momento. En su paper Royce propone que los sistemas chicos podrían desarrollarse en dos etapas, pero los grandes requerirán de atravesar 7 etapas consecutivas, que se sucedían una detrás de la otra:

1. Requerimientos del Sistema
2. Requerimientos de software
3. Análisis
4. Diseño del programa
5. Programación
6. Pruebas, y
7. Operación



Fuente: Managing The Development Of Large Software Systems. Recuperado de: <http://www-scf.usc.edu/>

Adicionalmente propone 5 factores claves que tienen que ser tenidos en cuenta para asegurar el éxito del proyecto:

1. Diseñar primero.
2. Documentar el diseño.
3. Construir dos veces: una primera versión que es revisada por los usuarios, una suerte de prototipo funcional, y una segunda versión que es el producto terminado.
4. Planificar, controlar y monitorear las pruebas.
5. Involucrar al cliente.

Royce también advirtió: Creo en este concepto [Waterfall], pero su implementación es riesgosa e invita a fallar. [...] La fase de testing, la cual ocurre al final del ciclo de desarrollo, es el primer evento con tiempo, almacenamiento, y transferencias de inputs y outputs.

Por otro lado lo describió como un proceso poco flexible: Si los requerimientos deben ser modificados o sufren cambios sustanciales en el diseño, el efecto sobre el proceso de desarrollo puede generar un *overrun* del 100% en tiempos de entrega y costos (Royce, 1970)

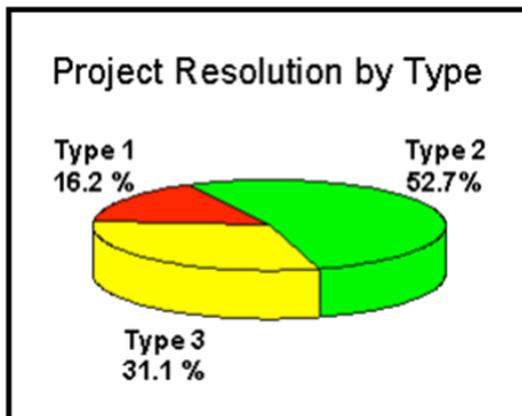
Posiblemente el mayor impacto de esta publicación fue que en el año 1985 fue citado por el Departamento de Defensa de los Estados Unidos en la publicación de su estándar Estándar 2167 establecía un proceso para el desarrollo de software

basado en los paper científicos publicados hasta ese momento: “Incorporar prácticas que demostraron ser eficientes desde la perspectiva del ciclo de vida del desarrollo, basada en la información reunida por el Departamento de Defensa y la industria” (Dod-Std-2167a, Military Standard: Defense System Software Development,1985).

Este estándar influenciará a varios otros estándares significativos de la industria, como el JSP-188 (Gran Bretaña), V-Model (Alemania), GAM-T-17 (Francia), entre otros.

1.2 Del Caos a la complejidad

A comienzo del año 2001, 17 referentes de lo que ese momento se conocía como *Lightweight Methodologist* se dan cita en Snowbird convocados por Kent Beck, el autor de Extreme Programming, para discutir sobre alternativas a los procesos tradicionales de desarrollo de Software (Agile Alliance, 2001). El



CHAOS Report, 1995

disparador fue que durante la década del 90´, con el surgimiento de las empresas nativas digitales y el crecimiento de la industria de software, se profundiza lo que se conoce como la Crisis del Software. Tal es así que en el año 1994, el Standish Group empieza estudiar este fenómeno y en su primera publicación en el año 1995 dice: “En su primer informe el Standish Grup reporta un récord de fallas en los proyectos de Software en los Estado

Unidos, donde se gastó más de 250 billones de dólares en aplicaciones de Tecnología en aproximadamente 175.000 proyectos, el promedio del costo de desarrollo en compañías grandes es de 2.322.00, compañías medianas es 1331.000 y en compañías chicas de 434000. La mayoría de estos informes fallaron (The Standish Group International, 1995, pp3)”.

En el reporte generado arroja que el 31.1% de los proyectos fueron cancelados, 52,7% costaron un 189% más que la estimación original y solo un 16,2% cumplieron el objetivo.

Dentro de las conclusiones que sacaron, el uso de time frames reducidos, con delivery de componentes tempranos y frecuentes incrementan las posibilidades de éxito. A esto lo llamaron growing software. (Standish Group International, 1995).

Los distintos informes de Chaos publicados durante la década del noventa concluyeron que el involucramiento del usuario y el empleo de períodos de tiempo más cortos eran claves para incrementar las tasas de proyectos exitosos (Alaimo, M & Salias M, 2018).

En este contexto no es de extrañar que a lo largo de la década fuesen desarrollándose y tomando relevancia distintos marcos de trabajo enfocados en el proceso Desarrollo, acuñado por los propios equipos, que fueron tomando notoriedad a partir del éxito de sus proyectos. Por esto durante el año 2000 se escribieron varios artículos que hacían referencia a la categoría de procesos "Lightweight". Varios de estos artículos se referían a Marcos como Extreme Programming, Adaptive Software Development, Crystal, y SCRUM" (Agile Alliance, 2001). Parte del acuerdo del manifiesto fue dejarse de llamar Lightweight, para convertirse en Agile Frameworks.

A continuación se expondremos los marcos de trabajo y principios que sirvieron como base y referencia para escribir el Manifiesto Ágil.

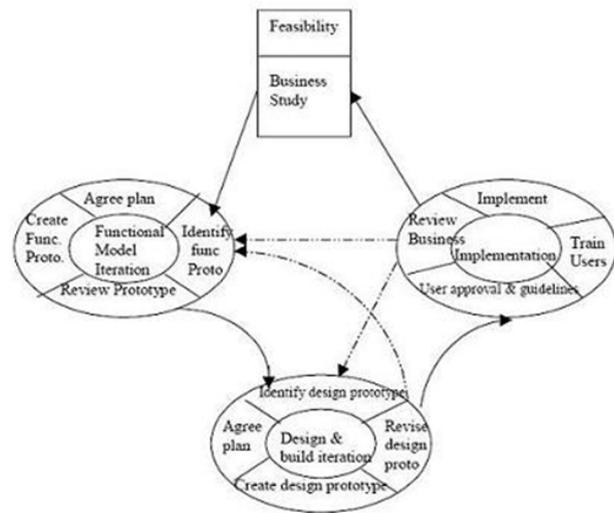
1.2. Marcos Emergentes

1.2.1 Dynamic Systems Development Method

A principios de la década del 90 dentro del Reino Unido se organizó un grupo de expertos de la industria del software que comenzaron a estudiar las mejores prácticas de la industria, con el objetivo de desarrollar y promover un marco de desarrollo de aplicaciones independiente. Esto derivó en la fundación del *Dynamic Systems Development Method Consortium*, donde uno de sus miembros era Arie van Bennekum, firmante del Manifiesto Ágil. En el año 1994, con su primera publicación *Dynamic Systems Development Method Manual*.

En dicha publicación el eje estaba puesto en cómo la falta de compromiso de los usuarios y alta gerencia impacta en comisión superior de la gerencia impacta en el incremento de costos y plazos de entrega.

En la publicación del año 1997, Jennifer Stapleton enumera estos 9 principios, como base del marco propuesto



Fuente: DSDM, Dynamic Systems Development Method: The Method in Practice. Londres, Addison-Wesley.

1. Participación activa del usuario: imperativo.
2. Los equipos deben estar facultados para tomar decisiones.
3. Centrarse en la entrega frecuente.
4. Criterio de entregable aceptado (Aptitud para el Negocio).
5. Desarrollo iterativo e incremental – Obligatorio.
6. Todos los cambios durante el desarrollo deben ser reversibles.
7. Los requisitos se basan en un nivel alto.
8. Las pruebas están integradas a lo largo del ciclo de vida.
9. Enfoque colaborativo y cooperativo.

Más allá de haber aportado a construir los cimientos del manifiesto, podemos destacar dos de la prácticas Core del DSM que se encuentran en plena vigencia, y que hoy en día se usan en combinación con otros Marcos de trabajo:

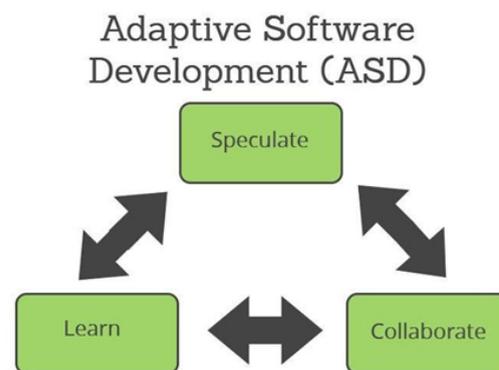
- Timeboxing: principalmente orientados a tener ciclos cortos de Delivery, no más de dos semanas.
- MoScWoW: Como técnica de priorización para poder combinar las necesidades técnicas con las necesidades de los usuario. El acrónimo hace referencia a los términos en Inglés *Must*, todas las características clasificadas en este

grupo deben implementarse y, si no se entregan, el sistema simplemente no funcionará; *Should*, las características de esta prioridad son importantes para el sistema, pero pueden omitirse si las limitaciones de tiempo lo ponen en peligro; *Could*, estas características mejoran el sistema con elementos funcionales que se pueden reasignar fácilmente a una caja de tiempo posterior. *Want to have*, estas funciones solo sirven a un grupo limitado de usuarios y son de poco valor.

1.2.2 Adaptive Software Development

Para la misma época, en Estados Unidos Jim Highsmith y Sam Bayer se encontraban desarrollando *Adaptive Software Development*. Ellos buscan generar un marco con capacidad de adaptarse al cambio en lugar de luchar contra él. Se basa en la adaptación continua a circunstancias cambiantes. Es por esto que basaron sus ciclos en 4 fases que se desarrollaban de forma iterativa: Especular, Colaborar y Aprender.

Al momento de especular se busca establecer la primera fase de iniciación para establecer los principales objetivos y metas del proyecto en su conjunto y comprender las limitaciones con las que operará el proyecto. El foco es establecer los objetivos prioritarios para el proyecto y la cantidad de iteraciones a realizar. La fase de colaboración es donde se centra la mayor parte del desarrollo manteniendo una componente cíclica. Un trabajo importante es la coordinación que asegure que lo aprendido por un equipo se transmita al resto y no tenga que volver a ser aprendido por los otros equipos. Para finalizar, se realizan ciclos de colaboración, donde se busca capturar lo que se ha aprendido, tanto positivo como negativo. Es un elemento crítico para la eficacia de los equipos.



Fuente: Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Nueva York, Dorset House.

Jim Highsmith (1997) enumera cuatro tipos de aprendizaje en esta etapa:

1. Calidad del producto desde un punto de vista del cliente.

2. Calidad del producto desde un punto de vista de los desarrolladores.
3. La gestión del rendimiento.
4. Situación del proyecto.

In adaptive software development, teams operate according to the 4 core values: mutual trust, mutual respect, mutual participation, and mutual commitment.

1. Confianza
2. Respeto
3. Participación
4. Compromiso

1.2.3 Crystal Clear

Crystal es una familia de metodologías que se subdivide en varios tipos de en función a la cantidad de personas que vayan a conformar el proyecto. El nombre de metodologías Crystal viene de que cada proyecto software puede caracterizarse según dos dimensiones, tamaño y criticidad, al igual que los minerales se caracterizan por dos dimensiones, color y dureza. Y esta es una de las bases de las metodologías Crystal, donde hay un marco específico para cada proyecto:

- Clear es para equipos de hasta 6 personas o menos.
- Amarillo para equipos entre 7 a 20 personas.
- Naranja para equipos entre 21 a 40 personas.
- Roja para equipos de entre 41 a 80 personas.
- Marron para equipos entre 81 a 200 personas

		Crystal Methodologies				
		Clear	Yellow	Orange	Red	Maroon
Criticality of the Project	Life (L)	L6	L20	L40	L80	L200
	Essential Money (E)	E6	E20	E40	E80	E200
	Discretionary Money (D)	D6	D20	D40	D80	D200
	Comfort (C)	C6	C20	C40	C80	C200
		1 to 6	7 to 20	21 to 40	41 to 80	81 to 200
		Number of People involved in the Project				

Fuente: Crystal Clear: A Human-Powered Methodology for Small Teams

Crystal da vital importancia a las personas que componen el equipo de un proyecto, y por tanto sus puntos de estudio son: Aspecto humano del equipo; Tamaño de un equipo, Comunicación, Políticas a seguir y Espacio físico de trabajo. Por eso propone 7 propiedades fundamentales:

1. Entregas frecuentes
2. Mejora reflexiva
3. Comunicación osmótica.
4. Seguridad personal.
5. Foco
6. Fácil acceso a usuarios expertos
7. Entorno técnico con pruebas automatizadas

1.2.4 Scrum

En el año 1993 Jeff Sutherland comienza a desarrollar un marco de trabajo con el objetivo de resolver los problemas que tenían los equipos al momento de cumplir con los objetivos de los proyectos en curso. En sus propias palabras “Scrum representa un cambio radical respecto a las metodologías prescriptivas y verticales de gestión de proyectos. Scrum se asemeja a los sistemas evolutivos, adaptativos y capaces de autocorregirse (Sutherland J, 2020).

Los autores definen valores como el corazón de Scrum.

- Compromiso
- Foco
- Franqueza
- Respeto
- Coraje

Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

En Scrum, el equipo se focaliza en construir software de calidad. La gestión de un proyecto Scrum se centra en definir cuáles son las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en vencer

cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo. En el capítulo 2 de desarrollaran en profundidad los 3 roles, 5 eventos y 3 artefactos propuestos por Scrum

1.2.5 Extreme Programming

En el año 1999, Kent Beck publicó el libro *Extreme Programming Explained: Embrace Change*. El libro conceptualiza la experiencia del autor dentro del proyecto C3 en la empresa Chrysler en 1996. En su rol Kent. En su libro Beck (1999) plantea que el principal objetivo de XP es que un equipo de desarrollo pueda producir software de mejor calidad de forma constante y a su vez busca promover una buena calidad de vida para el equipo. Para esto experimentó llevar al extremo las mejores prácticas que habían sido utilizadas históricamente en la industria del desarrollo de software, hasta ese momento. Extreme Programming se sostiene en 5 valores, 15 principios, 11 prácticas:

Valores de XP

1. Comunicación
2. Simplicidad
3. Retroalimentación
4. Coraje:
5. Respeto

A pesar de que XP nace como una práctica de desarrollo de software, se fundamenta en valores que pueden ser abrazados y aplicados por cualquier organización y equipo de trabajo.

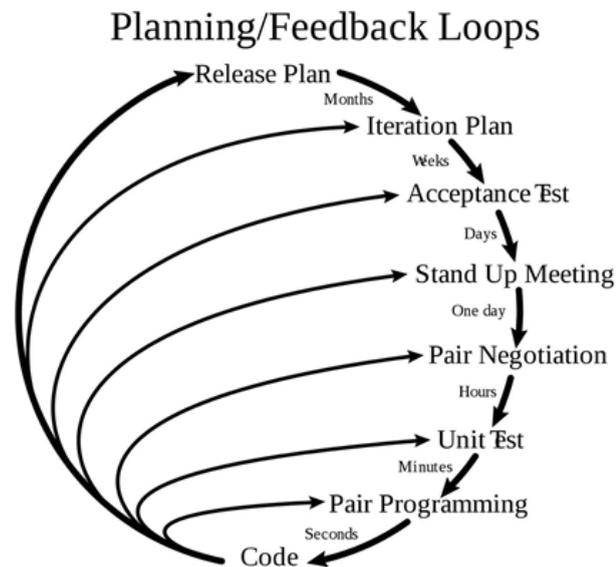
1. Realimentación rápida
2. Asumir simplicidad
3. Cambio incremental
4. Abrazar el cambio
5. Trabajo de calidad
6. Enseñar a aprender
7. Pequeña inversión inicial

8. Juega para ganar
9. Trabajar con los instintos de las personas, no en contra de ellos
10. Responsabilidad
aceptada
11. Experimentos concretos
12. Comunicación abierta y honesta
13. Viaja con poco equipaje
14. Medición honesta

XP es un marco de trabajo centrado en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Por este motivo se encarga de de inventariar las siguiente practicas técnicas a fin de gestionar la complejidad de los proyectos de desarrollo:

- Prácticas de Feedback
 - Test-driven development
 - The Planning Games.
 - On-site Customer
 - Pair Programming
- Proceso Continuo
 - Code Refactoring:
 - Continuous Integration



Fuente: Extreme Programming Explained : Embrace Change. Nueva Jersey, Editorial Pearson

- Small Releases:
- Code Understanding
 - Simple Design
 - Coding Standards
 - Collective Code Ownership
 - System Metaphor
- Programmer's Work Conditions
 - 40-Hour Week

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

1.3 Snowbird

En marzo de 2001, 17 referentes de la industria del software y promotores de lo que hasta entonces se conocía como *"lightweight Methodologies"* (Metodologías Livianas) son convocados por Kent Beck, para reunirse en Salt Lake City. Tras el fortalecimiento de algunos de los marcos de trabajo que emergieron en el siglo pasado, varios de los referentes de la industria sintieron la necesidad de debatir y

buscar alternativas a los procesos tradicionales de desarrollo de software, caracterizados por la rigidez de su carácter normativo y su gran dependencia de la planificación detallada previa al desarrollo (Agile Alliance, 2001).

Los integrantes de la reunión resumieron en cuatro postulados lo que ha quedado denominado como “Manifiesto Ágil”, que son los valores sobre los que se asientan estos métodos.

“Estamos poniendo al descubierto mejores métodos para desarrollar software, haciéndolo y ayudando a otros a que lo hagan. Con este trabajo hemos llegado a valorar:

A los individuos y su interacción, por encima de los procesos y las herramientas.

El software que funciona, por encima de la documentación exhaustiva.

La colaboración con el cliente, por encima de la negociación contractual.

La respuesta al cambio, por encima del seguimiento de un plan.

Aunque hay valor en los elementos de la derecha, valoramos más los de la izquierda.”

El manifiesto no pone en duda que haya elementos de metodologías de las llamadas tradicionales o “pesadas” que no sean importantes, ahora bien, serán los elementos de “de la izquierda” los que generen el contexto necesario para desarrollar software en óptimas condiciones. Las mismas deberían permitirnos trabajar en un contexto de complejidad, mejorando significativamente los ratios de resultados. A su vez busca generar un entorno de acuerdos donde realmente lo más importante sean las personas, despegándose del concepto tradicional de Recursos, propio de la Industria del SXX. Así lo expreso Bob Martin al cierre del manifiesto:

Las Metodologías Ágiles realmente se trata de “cosas blandas”, para poder crear buenos productos para los clientes y generar conductas más allá de hablar sobre “las personas como el recurso más importante”, y realmente actuar como si fuesen lo más importe y dejar de usar la palabra recurso.

Los valores fueron complementados con 12 principios, que permiten trazar un eje entre los distintos marcos de trabajo, los lineamientos en las organizaciones que quieran moverse hacia la agilidad:

1. Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.
2. Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se dobligan al cambio como ventaja competitiva para el cliente.
3. Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los períodos breves.
4. Las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto.
5. Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.
6. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
7. El software que funciona es la principal medida del progreso.
8. Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica enaltece la agilidad.
10. La simplicidad como arte de maximizar la cantidad de trabajo que se hace, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos que se autoorganizan.
12. En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta en consecuencia.

Luego de una década de la publicación del Manifiesto, en el año 2011 el Standish Group publicó: El proceso Ágil es el remedio universal para el fracaso en los proyectos de desarrollo de software. Las aplicaciones desarrolladas a través de

procesos ágiles tienen tres veces la tasa de éxito del método en cascada tradicional y un porcentaje mejor de demoras y sobrecostos (Standish Group, 2011).

Capítulo 2 - Scrum como palanca de la transformación

2.1 ¿Por qué se transforman las empresas?

Con el comienzo del siglo XXI y el advenimiento de la cuarta Revolución Industrial, gran parte de las empresas que fueron referentes del siglo pasado empezaron a sentir una necesidad profunda de transformación: La cuarta revolución industrial genera un mundo en el que los sistemas de fabricación virtuales y físicos cooperan entre sí de una manera flexible en todo el planeta. Esto permite la absoluta personalización de los productos y la creación de nuevos modelos de operación. (Schwab, 2016, p 21).

Este contexto empezó a impactar profundamente en las expectativas de los consumidores, ya que sus estándares de experiencia se vieron fuertemente influenciados por la digitalización y le comenzaron a exigir a sus viejas empresas que generen experiencias similares a las empresas que son nativas digitales. Tal como afirma El CEO de VMWare Pat Gelineer: Hoy en día todos los negocios son digitales, no tiene sentido hablar de negocios digitales versus negocios digitales (Slotnisky, 2016). Es por este motivo que gran cantidad de empresas comenzaron su camino de Transformación Digital. Independientemente del sector, las empresas deben asumir la revolución del software. Por ejemplo las empresas líderes de la industria musical de tecnología: iTunes, Spotify y Pandora de Apple. Los sellos discográficos tradicionales van perdiendo relevancia año a año (Andressen, 2011).

La creación de productos digitales es un proceso complejo, dado que se deben entender las necesidades del cliente, al mismo tiempo que se diseña e implementa la solución. Tal como plantea David J. Snowden (2007), en un sistema complejo los agentes se restringen mutuamente, especialmente a lo largo del tiempo. Esto significa que no podemos pronosticar ni prever lo que ocurrirá. Es por esto que la clave del éxito se halla en las soluciones emergentes y no por las soluciones diseñadas en el pasado. La complejidad obliga a las empresas digitales a moverse de su zona de confort y estar descubriendo constantemente nuevas y mejores formas de satisfacer las necesidades de sus clientes. Tal como cita Débora Slotnisky (2016) al cofundador de Google Larry Page: En las empresas de tecnología es relevante que las compañías no se queden en su zona de confort.

Por esto es que las empresas que buscan Transformarse Digitalmente buscan transformarse en organizaciones más ágiles, y el paso natural es incorporar

marcos de trabajo que permitan habilitar la Agilidad, como por ejemplo lo es Scrum. Tal como lo definen sus creadores, Jeff Sutherland & Ken Schwaber (2021), Scrum es un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos.

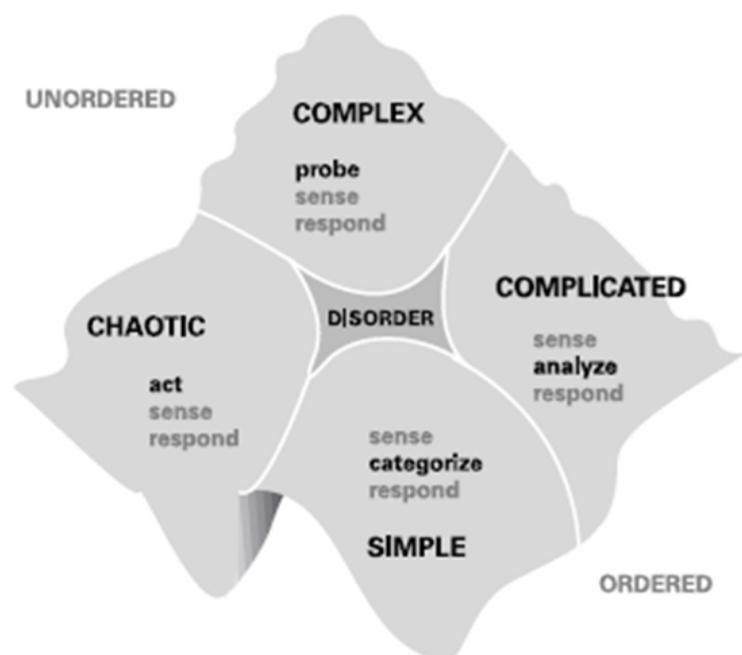
2.2 Complejidad y empirismo

Scrum es un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos (Sutherland J & Schwaber, K, 2021). Para poder comprender cuales son las características de los entornos complejos, podemos usar el Marco Cynefin, desarrollado por C. F. Kurtz y D. J. Snowden :

no son ordenados: no hay una relación inmediatamente aparente entre causa y efecto, y el camino hacia adelante es determinado basándose en los patrones emergentes. El mundo ordenado es el mundo de la gestión basada en hechos; el mundo no ordenado representa la gestión basada en patrones.

El mundo ordenado es el mundo de la gestión basada en hechos; el mundo no ordenado representa la gestión basada en patrones.

A diferencia de los entornos complejos, . Los contextos simples y complicados suponen un universo ordenado, donde las relaciones de causa y efecto son perceptibles y las respuestas correctas pueden ser determinadas basándose en los hechos. Es por esto que los modelo en cascada suelen tener mejores resultados en entorno sexto entrenos. Cuanto más estables son, más posibles es



Fuente: Un marco para la toma de decisiones del líder, Harvard Business Review, Vol 85, N° 11 pp.110 - 120.

predeterminarse y utilizar modelos de organización rígidos y poco amigables con los cambios.

Scrum es un marco de trabajo que permite encontrar prácticas emergentes en dominios complejos (Alaimo, M & Salias, M, 2018). En los contextos complejos no existen soluciones prediseñadas, sino que hay que descubrirlo a partir del conocimiento emergente

Scrum se basa en el empirismo y el pensamiento Lean. El empirismo afirma que el conocimiento proviene de la experiencia y de la toma de decisiones con base en lo observado. El pensamiento Lean reduce el desperdicio y se enfoca en lo esencial (Sutherland J & Schwaber, K, 2021).

El pensamiento Lean tiene su origen en la empresa japonesa Toyota, cuando comienzan a buscar un sistema de producción donde pudiese entregarse el pedido al cliente lo más rápido posible, eliminando los desperdicios del proceso: El *Toyota Production System* (TPS) se basó en dos conceptos: "jidoka" , cuando un problema ocurre, el equipo frena inmediatamente, para evitar la producción de productos defectuosos; y el concepto "*Just-in-Time*", donde cada proceso produce solo lo que es necesario para el proceso siguiente, en un flujo continuo (Toyota Production System, s.f.). El reconocimiento generalizado de TPS como sistema modelo de producción creció rápidamente con la publicación en 1990 de "La máquina que cambió el mundo", basada en una investigación de cinco años dirigida por el Instituto de Tecnología de Massachusetts. Los investigadores del MIT descubrieron que TPS era mucho más efectivo y eficiente que la producción en masa tradicional que representaba un paradigma completamente nuevo y acuñó el término Lean para indicar este enfoque radicalmente diferente de la producción.

El proceso Lean combina las ventajas del trabajo artesanal con la producción en masa evitando los costos del primero y la rigidez del último. El proceso Lean es liviano porque usa menos de todo comparado con la producción en masa, mejorando la productividad. El foco de la filosofía Lean es hacer solo lo que se necesita. Busca producir de calidad de manera eficiente a través de la eliminación completa de desperdicios (Womack, 1990).

- La mitad de esfuerzo humano en la fábrica, la mitad de espacio de fabricación, la mitad de inversión en herramienta y la mitad de horas de ingeniería para desarrollar nuevos productos, en la mitad de tiempo- (Womack,1990, p. 13). Para esto se proponen 5 principios:

1. Identificar el valor desde la perspectiva del cliente
2. Mapear el flujo de valores.
3. Crear el flujo.
4. Establecer un sistema pull
5. Perseguir la perfección con la mejora continua del proceso.

Sobre los principios de Lean se busca constantemente la reducción de los desperdicios dentro del proceso. Este es cualquier cosa que los clientes no creen que agregue valor y por la cual no están dispuestos a pagar. Esto requiere una mejora continua, lo que constituye el corazón de Lean.

Estos son los 7 desperdicios señalados por Womack en la publicación original de 1990:

1. Transporte innecesario
2. Exceso de inventario
3. Movimiento innecesario de personas, equipos o maquinaria
4. Espera,
5. Sobreproducción
6. Sobreprocesamiento
7. Defectos, que requieren esfuerzo y costo para su corrección.

Por otro lado, el empirismo propone que el conocimiento surge de la experiencia. Se basa en tomar decisiones basados en la información concreta obtenida de la observación que muestra el progreso del desarrollo de producto, los cambios en el mercado y los comentarios de los clientes (Sutherland & Schwaber K, 2021). En contraposición, los modelos tradicionales se basan en modelos de control predictivo, ya que toman las decisiones del proyecto basados en el análisis de datos y la proyección de resultados. Estos funcionan bien en contextos estables y predecibles.

Por esto Scrum rompe con el modelo de cascada en particular y los modelos predictivos en general. Tal como dice Sutherland (2011), todo esfuerzo invertido en planear, con intención de restringir cualquier cambio y conocer lo incognoscible, es

inutil. Scrum abraza la incertidumbre y la creatividad. Cuanto más complejo es el entorno, menos predecible es.

2.3 Scrum, agilidad en la práctica

Scrum es un marco de trabajo liviano, que busca generar un entorno de trabajo que permite gestionar la complejidad. Para esto, propone 3 roles, 5 eventos y 3 artefactos. El responsable de la gestión del entorno será el Scrum Master, para guiar al Scrum Team y a la organización a realizar una adecuada adopción del marco de trabajo

2.4.1 Roles

Scrum se basa en tener equipos pequeños, con 3 roles, Scrum Master, un Product Owner y Developers. Esta división no implica jerarquías, ni subequipos. Todas las personas están enfocadas en el objetivo del producto. El conjunto de los 3 roles conforman un Scrum Team. En la Guía de Scrum Sutherland plantea 3 responsabilidades distintas dentro del Scrum Team:

Developers

- Crear un plan para el Sprint, el Sprint Backlog;
- Inculcar calidad al adherirse a una Definición de Terminado;
- Adaptar su plan cada día hacia el Objetivo del Sprint; y,
- Responsabilizarse mutuamente como profesionales.

Product Owner

- Desarrollar y comunicar explícitamente el Objetivo del Producto;
- Crear y comunicar claramente los elementos del Product Backlog;
- Ordenar los elementos del Product Backlog; y,
- Asegurarse de que el Product Backlog sea transparente, visible y se entienda.

El Scrum Master

- Guiar a los miembros del equipo en ser autogestionados y multifuncionales;

- Ayudar al Scrum Team a enfocarse en crear Increments de alto valor que cumplan con la Definición de Terminado;
- Procurar la eliminación de impedimentos para el progreso del Scrum Team; y,
- Asegurarse de que todos los eventos de Scrum se lleven a cabo y sean positivos,
- Ayudar a encontrar técnicas para una definición efectiva de Objetivos del Producto y la gestión del Product Backlog
- Ayudar al Scrum Team a comprender la necesidad de tener elementos del Product Backlog claros y concisos
- Ayudar a establecer una planificación empírica de productos para un entorno complejo; y,
- Facilitar la colaboración de los interesados según se solicite o necesite.
- Liderar, capacitar y guiar a la organización en su adopción de Scrum;
- Planificar y asesorar implementaciones de Scrum dentro de la organización;
- Ayudar a los empleados y los interesados a comprender y aplicar un enfoque empírico para el trabajo complejo
- Eliminar las barreras entre los interesados y los Scrum Teams.

2.4.2 Eventos

Una de las principales diferencias de Scrum con las metodologías tradicionales es que trabaja con *timeboxes* fijos, denominados sprints. Dentro de estos ciclos de trabajo es donde suceden los eventos, donde se busca inspeccionar y adaptar los artefactos

Dentro de la guía se propone los siguientes propósitos para cada uno de los eventos:

- **Sprint:** Los Sprints son el corazón de Scrum, donde las ideas se convierten en valor.
- **Sprint Planning:** Sprint Planning inicia el Sprint al establecer el trabajo que se realizará para el Sprint. El Scrum Team crea este plan resultante mediante trabajo colaborativo. El Product Owner se asegura de que los asistentes estén preparados para discutir los elementos más importantes del Product Backlog y cómo se relacionan con el Objetivo del Producto. El Scrum Team

también puede invitar a otras personas a asistir a la Sprint Planning para brindar asesoramiento.

- **Daily Scrum:** El propósito de la Daily Scrum es inspeccionar el progreso hacia el Objetivo del Sprint y adaptar el Sprint Backlog según sea necesario, ajustando el trabajo planificado entrante.
- **Sprint Review:** El propósito de la Sprint Review es inspeccionar el resultado del Sprint y determinar futuras adaptaciones. El Scrum Team presenta los resultados de su trabajo a los interesados clave y se discute el progreso hacia el Objetivo del Producto.
- **Sprint Retrospective:** El propósito de la Sprint Retrospective es planificar formas de aumentar la calidad y la efectividad.

2.4.3 Artefactos de Scrum

Los artefactos de Scrum buscan generar transparencia y garantizar que el esfuerzo este alineado con el valor del producto. Dentro de la guía se proponen los siguientes artefactos:

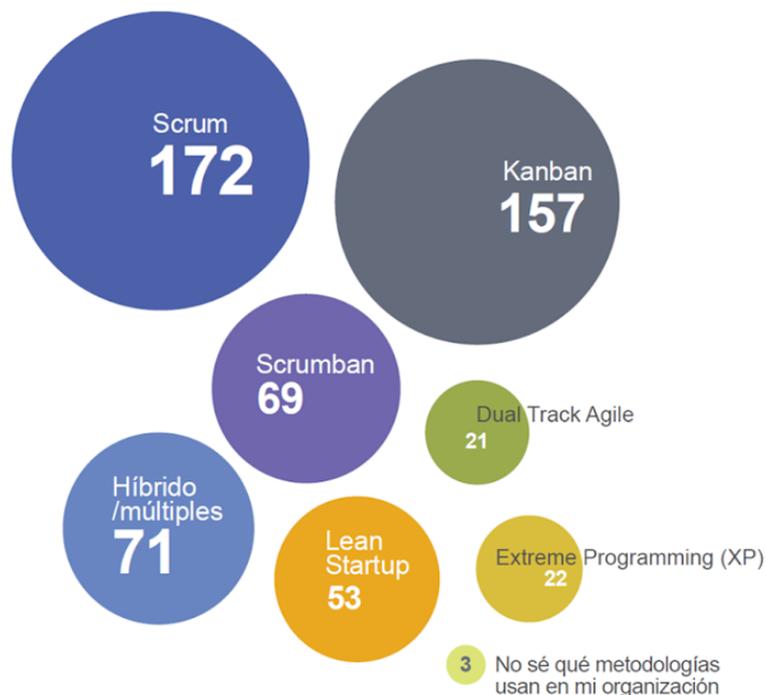
- **Product Backlog:** El Product Backlog es una lista emergente y ordenada de lo que se necesita para mejorar el producto. Es la única fuente del trabajo realizado por el Scrum Team [...] Los Developers que realizarán el trabajo son responsables del dimensionamiento. El Product Owner puede influir en los Developers ayudándolos a entender y seleccionar sus mejores alternativas.
- **Sprint Backlog:** se compone del Objetivo del Sprint (por qué), el conjunto de elementos del Product Backlog seleccionados para el Sprint (qué), así como un plan de acción para entregar el *Increment* (cómo). El Sprint Backlog es un plan realizado por y para los Developers.
- **Increment:** Un *Increment* es un peldaño concreto hacia el Objetivo del Producto. Cada *Increment* se suma a todos los *Increments* anteriores y se verifica minuciosamente, lo que garantiza que todos los *Increments* funcionen juntos. Para proporcionar valor, el *Increment* debe ser utilizable.

Scrum no es ni una metodología ni un marco completo. Esto se debe a que no existen ni mejores ni buenas prácticas en un contexto complejo. Es el equipo de involucrados quien encontrará la mejor manera de resolver los problemas y estas serán soluciones emergentes (Alaimo & Salias, 2015)

2.4 ¿Cómo impactó la Agilidad en la industria del Software?

Desde el comienzo de este trabajo hemos repasado diferentes hitos que han marcado la evolución de la industria del software durante los últimos 50 años: la Crisis del Software, el Paper de Royce, el Estándar 2167, el surgimiento de los marcos livianos durante la década del 90, la creación del CHAOS Report, la proclamación del Manifiesto Ágil del año 2001, y el boom de las Transformaciones Digitales desde el inicio de este siglo. Durante este proceso la Agilidad en general y Scrum en particular han crecido en adopción en distintas organizaciones e industrias. En un reporte del año 2020 elaborado en conjunto entre NTT Data y el MIT, Scrum es el marco de trabajo de mayor adopción en Latam.

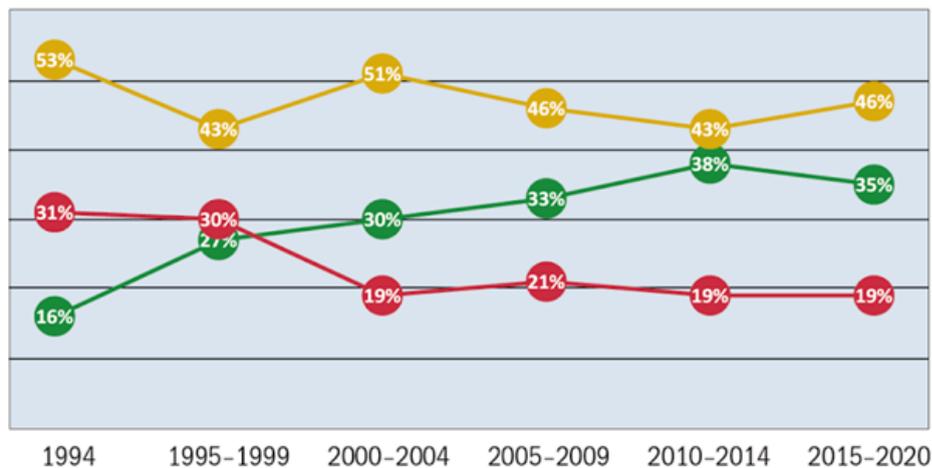
Metodologías más adoptadas en las organizaciones



Fuente: La Agilidad En América Latina 2021. Una Competencia Core de las Organizaciones Adaptativas. Publicado por Opinno.

Desde el año 1994 a la actualidad, el CHAOS Report ha servido como termómetro de excelencia operativa en la industria.

En su primer informe se veía una tasa de fracaso superior al 30% y de éxito inferior



Fuente: Standish Group (2020). CHAOS Report: Beyond Infinity

al 16%. Esos números fueron moviéndose hasta encontrar cierta estabilidad durante los últimos 20 años, encontrando valores promedio de: 30% para proyectos exitosos, 50% para los que sufren modificaciones y un 20% son abandonados

Dentro de las conclusiones del último CHAOS Report se destaca que por más que se usen prácticas DevOps o Agile, siempre el factor humano termina siendo determinante. Si no hay un buen sponsor, un buen equipo y un buen entorno de trabajo, los proyectos tienden al fracaso. Este fenómeno se incrementa cuanto mayor valor busca capturar la iniciativa (CHAOS Report, 2020).

Ahora bien, al momento de estudiar los marcos de trabajo como variable relacionada al éxito de los proyectos se ha visto una buena performance de los modelos Waterfall cuando se tratan de proyectos chicos, pero una alta tasa de fracasos en los proyectos medianos y grandes. Los proyectos que utilizan metodologías Ágiles han preformado bien independientemente de su tamaño. Cabe destacar que durante la investigación se han visto ciento de proyectos que dicen contar con entornos de trabajo ágiles, pero que al mismo tiempo están compuestos por miles de empleados, trabajando por más de tres años sin interacción con los clientes, que no se han entregado al cliente por más de tres años, lo podemos considerar “Agile Name Only”.

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
ALL-SIZE PROJECTS	AGILE	42%	47%	11%
	WATERFALL	13%	59%	28%
LARGE-SIZE PROJECTS	AGILE	19%	56%	25%
	WATERFALL	8%	56%	36%
MEDIUM-SIZE PROJECTS	AGILE	34%	53%	13%
	WATERFALL	9%	66%	25%
SMALL-SIZE PROJECTS	AGILE	59%	36%	5%
	WATERFALL	45%	46%	9%

Fuente: Standish Group (2020). CHAOS Report: Beyond Infinity

Uno de los cambios más significativos del último CHAOS Report fue el cambio en la definición de éxito. El cambio de contexto obligó a los investigadores a dejar de lado el oxidado triángulo de hierro, ya que no daba cuenta de los objetivos perseguidos por las empresas durante los últimos 10 años. Tal como narra Hans Mudle durante la presentación del último reporte “Tuvimos que salir de la definición tradicional de tradicional, donde veíamos cumplimiento de presupuesto y tiempo. Vemos que un proyecto puede no estar en tiempo pero generando mucho más valor, por eso tuvimos que crear una perspectiva moderna de éxito, donde veamos tiempo, presupuesto y satisfacción, donde podamos ver cual es el outcome para los clientes, partner y organización” (Mulder, 2020).

Como vimos en esta sección, los proyectos exitosos se caracterizan por tener un buen entorno de trabajo, un buen sponsor y buenos equipos. Evidentemente no todas las organizaciones cuentan con estos, y conseguirlos requiere madurez organizacional. Scrum, por sus características operativas, podría tener las características suficientes para acompañar el proceso de maduración organizacional y acompañar el proceso de transformación, con el objetivo de que las organizaciones alcancen niveles más altos de generación de valor.

Capítulo 3- Investigación

3.1 Objetivo

El presente relevamiento tiene como objetivo conocer la mirada de algunos referentes de la transformación Ágil acerca de si Scrum como marco de trabajo Ágil más popular tiene la capacidad de habilitar de tener buenos equipos, buenos entornos y buen sponsoreo.

Para esto se realizara un muestro no probabilístico con el objetivo de generar un analisis cualitativo, y que sirva como puto de partida para comenzar a relacionar las conclusiones del CHAOS Report con Scrum. Por este motivo se seleccionarán profesionales con un vasto conocimiento teórico y experiencia acompañando procesos de transformación Ágil.

3.2 Entrevistados

Ingrid Asitz



Ingrid tiene 25 años de experiencia en la industria de desarrollo de software y actualmente trabaja en transformaciones digitales.

Es autora del libro “Poder Creativo” y escribió artículos que fueron publicados en libros de otros autores y varios otros materiales. Estudió filosofía en la Universidad de Buenos Aires.

Es Certified Scrum Master (2009) por Scrum Alliance, tiene una certificación en Management 3.0 y PDA Assessment.

Martín Salías



Martín es Coach & Trainer de Kleer, tiene más de 35 años en la industria del software trabajando con varias plataformas diferentes, programación y lenguajes hablados, para organizaciones que van desde las Naciones Unidas hasta Microsoft Corp, en una amplia gama de industrias. Comenzó como un programador de núcleo duro y actuó en muchos roles

como líder de desarrollo, entrenador de XP y arquitecto.

Martía Thompson



Maria se recibió de actriz en la Escuela Municipal de Arte Dramático. Dedicó más de 10 años a dar clases de teatro, realizar obras y performances. Desde el año 2009 es miembro de la comunidad Ágil de Argentina. Es cofundadora de la Consultora Fuerza Tres y capacitadora y facilitadora en procesos de transformación organizacional en empresas, gobierno y ONG's. Está certificada por la Scrum Alliance como Professional Scrum Master y como Coach Profesional.

Tomas Christie



Tomás Ingeniero Informático, con más de 20 años de experiencia en desarrollo de software a la medida, e implementaciones ágiles. Actualmente está enfocado en acompañar a las organizaciones en sus procesos de transformación, colaborando en la creación de planes estratégicos, tácticos y operativos para transitar estos caminos de manera sistémica.

3.3 Hallazgos

Durante las entrevistas se les preguntó a los participantes si creen que la demanda creciente de implementar Scrum en distintas organizaciones era un habilitador para promover el desarrollo de los equipos de alta rendimiento, desarrollar buenos sponsor y generar buenos entornos de trabajo. El objetivo de las mismas no es verificar la hipótesis, sino tener una mejor comprensión de que implica implementar Scrum y tener un mejor contexto de exploración. En líneas generales los 4 entrevistados coincidieron que Scrum tiene que ser considerado un medio para conseguir un objetivo y no un fin en sí mismo. A su vez es clave el expertise y la convicción de la persona que auspicia de guía dentro de este proceso. Lo más común es que esto sea liderado por un Scrum Master o un Agile Coach. María enfatizó que en las ocasiones que le toca liderar implementaciones de Scrum en organizaciones chicas (menos de 150 personas), pudo movilizar al C-level a una

transformación más personal, que rápidamente permeó en los equipos de trabajo. Para ella es clave generar un cambio a nivel personal en los líderes para habilitar la transformación organizacional, y está convencida que Scrum es una buena excusa para poner en marcha ese proceso.

Tomas es claro y directo al plantear que Scrum por sí solo no sirve para nada. Desde su perspectiva, Scrum principalmente tiene mucha más capacidad de impactar sobre el equipo que sobre el contexto y los sponsor. Plantea que para él en las organizaciones existen dos planos distintos, uno donde los equipos construyen los productos y otro donde los sponsor toman decisión y fomentan determinada cultura organizacional. En este sentido Scrum podría impactar en el contexto y los sponsor, pero sería un impacto de mediano o largo plazo, siempre y cuando los resultados de negocio acompañen ese proceso. Ahora bien, esto sólo sería posible contando con un Scrum Master muy senior y con mucha vocación transformadora. Para él el hábitat natural de Scrum es a nivel de los equipos, y será el entorno en el que se desarrollan, el que permita sacarle el máximo provecho. Martín también destaca que para que Scrum impacte a nivel de los equipos se debe apostar por el desarrollo de las personas, y esa debería ser una de las principales prioridades del management. Lo sintetiza diciendo: El management desarrolla a las personas, las personas desarrollan productos.

Dentro de las prácticas propuestas por Scrum la retrospectiva es la que se describe con mayor potencial transformador, ya que permite generar cambios reales en grupos humanos. La retrospectiva tiene la capacidad de provocar a los participantes para romper con las estructuras burocráticas y generar conversaciones sinceras y reales. Cabe destacar que todos los entrevistados coincidieron en que estas son un gran habilitador de la transformación. Ingrid y Maria coinciden que comenzar a generar este espacio con el comité ejecutivo puede ser un factor clave de éxito para poder impactar en el liderazgo y, desde ahí, transformar la organización.

Hacer una real implementación de Scrum implica explícitamente cambiar el contexto en el que trabajan los equipos y, en la mayoría de los casos, también es necesario repensar profundamente como está estructurada la organización. Por esto para Martín es fundamental que la organización que busca implementar Scrum sea consciente que en el fondo va a tener que prescindir de la capa de mandos medios, ya que está delegando gran parte de esas decisiones a los equipos. Eso no quiere

decir necesariamente que los van a tener que despedir, pero sí que van a tener que buscar un lugar en el nuevo esquema organizativo. En esta línea Tomas remarca el cambio producido en la guía de Scrum, donde se cambió el término de “autoorganizados” a “autogestionados”. Desde su perspectiva este cambio enfatiza el poco valor que generan los roles de gestión, e invita a repensar las organizaciones prescindiendo de los mandos medios.

Por la profundidad que requiere toda transformación es fundamental que el proceso esté sponsorado por un C-level. Evidentemente, tal como pasa con el resto de los proyectos o iniciativa organizacional, las transformaciones si no cuentan con un buen sponsor también están destinadas al fracaso. Tomas remarca la importancia de un C-level convencido de hacer las cosas diferentes y tomar riesgos. Al igual que Ingrid, coincide en que un buen Sponsor tiene que tener libertad para administrar el presupuesto de la iniciativa, tener una visión clara, y estar disponible para el equipo. Los cuatro coinciden en que habilitar la autonomía en los equipos, es una capacidad esencial para generar resultados sostenibles a largo plazo. A su vez destacan estas mismas como las características claves de un buen sponsor

Evidentemente la capacidad de influenciar sobre los Comités Ejecutivos o sponsors termina siendo una característica clave de quienes buscan liderar una transformación. Ingrid también plantea que más allá de las herramientas específicas que pueden proponer los distintos Marcos de Trabajo Ágiles, ella suele ver que a los Agilistas les cuesta influenciar sobre el C-level debido a que su foco suele estar puesto en la operación de los equipos, y se alejan de la complejidad que manejan los sponsor. Una de las hipótesis que plantea es que posiblemente esto se deba a que gran parte de los Agilistas vienen de gestionar productos digitales y no empresas, suelen ser efectivos en lo táctico y lo operativo, pero a veces les falta estrategia y abordaje sistémico." A su vez, esto haría que se busque implementar Agilidad como solución para gestionar problemas complejos, pero tal como explica Snowden en la complejidad no existen soluciones lineales, sino que hay que generar prácticas emergentes. Martín remarca que, desde su experiencia, las organizaciones que mejores han logrado habilitar y escalar la Agilidad son aquellas lideradas por *makers*, personas que valoran la acción por sobre la gestión y que tienen experiencia en el negocio o práctica que están trabajando.

Para poder trabajar de forma descentralizada y autogestionada es clave la claridad en los objetivos. En esta línea, Martín y María destacan la importancia de

tener un propósito real y concreto, orientado a resolver un problema a la sociedad, como motor de tener equipos de alta performance. La claridad en el propósito, orienta el foco de la organización en la mejora continua del producto, lo que decanta en la generación continua de valor por parte de los equipos. En los casos que esto no sucede, los equipos tienden a la desmotivación y a generar elevados niveles de rotación, lo que termina impactando en la calidad y tiempos de desarrollo.

	Ingrid	Martín	Tomas	Maria
¿Crees que el involucramiento del C-level impacta de forma directa en el éxito de la transformación?	Siempre	Siempre	Siempre	Siempre
¿Scrum tiene el potencial de promover buenos equipos?	Siempre	Siempre	Siempre	Siempre
¿Incide la expertise del Scrum Master o Agile Coach en los resultados?	Casi Siempre	Casi Siempre	Casi Siempre	Siempre
¿Sirve comenzar una transformación desarrollando equipos Scrum=	A veces	A veces	A veces	A veces
¿Scrum tiene el potencial de promover buenos contextos?	Casi Nunca	A veces	Nunca	A veces
¿Scrum tiene el potencial de promover buenos sponsors?	Nunca	Casi Nunca	Nunca	A veces

Capítulo 4 - Conclusiones

Durante los últimos 30 años la industria del software fue creciendo tanto que terminó revolucionando la manera de concebir los negocios en el siglo XXI. Todas las experiencias de consumo de hoy en día se encuentran atravesadas por la digitalización, y detrás de todo eso hay software. Al mismo tiempo, las empresas no aprendieron a dominar el software a la misma velocidad que los clientes lo adoptaron para su vida cotidiana, lo que hizo que las organizaciones que crecieron en un mundo más analógico, busquen volverse empresas tecnológicas, en un mundo marcado por el liderazgo de las empresas nativas digitales (Alphabet, Meta y Amazon).

Como vimos durante el desarrollo de todo el trabajo de investigación encontrar una manera de hacer software de calidad y que cumpla con los criterios de éxito no ha sido tarea sencilla. La industria sigue buscando mejorar sus prácticas, al mismo tiempo que busca escalar sus soluciones, lo que parece ser el combo perfecto para desatar el caos organizacional. En este contexto pareciera normal que las empresas demanden a los expertos una receta que les permita lograr una transformación de forma eficiente, aunque esto no sea más que un oxímoron. Tal como vimos anteriormente, las transformaciones son procesos complejos, y como tales, solo pueden alcanzar el éxito si se basan en prácticas emergentes, y no en buenas prácticas. Esto quiere decir que toda organización que busca transformarse está obligada a desarrollar su capacidad de experimentación y aprendizaje.

Indudablemente la popularidad creciente de Scrum hizo que las organizaciones que iniciaron su camino de Transformación Digital, tuviesen como paso natural buscar desarrollar sus primeras iniciativas Digitales bajo este marco. Ahora bien, tal como lo dice la Guía de Scrum (Sutherland, 2021), Scrum es un marco incompleto de manera intencional, sólo define las partes necesarias para implementar su teoría. En este sentido se vuelve fundamental la expertise y técnica de las personas que lideran el proceso. Los líderes no deberían repetir recetas, sino utilizar la implementación de Scrum para pasar de un paradigma de control predictivo a uno empírico de control. La propia implementación de Scrum debería ser el proyecto insignia de la organización con base en el empirismo y en la gestión de la complejidad.

Queda claro que Scrum puede ser un buen marco de trabajo para construir producto, pero no es suficiente para transformar y gestionar la estructura y cultura organizacional. Scrum podría llegar a servir para visibilizar los desafíos que tiene una organización para mejorar su capacidad adaptativa, pero serán las decisiones que tomen sus líderes las que permitan soltar viejas prácticas y habilitar una transformación real.

Ver la Agilidad como un fin en sí mismo supone un gran riesgo para la organización y para el negocio. Ahora bien, una implementación consciente de Scrum puede ser la llama que encienda la transformación organizacional y permita el desarrollo de equipos de alto rendimiento, grandes sponsors y excelentes entornos laborales.

Referencias

- Agile Alliance (2001). Manifiesto for Agile Software Development. Recuperado de: <https://agilemanifesto.org/>
- Alaimo, M. (2020). Gestión de proyectos: ¿Qué es el modelo secuencial Waterfall y cómo funciona este método en cascada?. Recuperado de: alaimolabs.com/
- Alaimo, M & Salias, M (2015). Proyectos Ágiles con #Scrum, Editorial Dunken.
- Andressen, M. (2011). Why Software Is Eating the World. The Wall Street Journal 20 de Agosto de 2011.
- Beck, K (1999) Extreme Programming Explained : Embrace Change. Editorial Pearson
- Cockburn, A (2007). Crystal Clear: A Human-Powered Methodology for Small Teams. Editorial: Pearson Education, Inc.
- Dod-Std-2167a, Military Standard: Defense System Software Development,1985. Recuperado de: http://everyspec.com/DoD/DoD-STD/DOD-STD-2167A_8470/.
- Galloway, S (2018). Four. Penguin Random.
- Highsmith, J. (1997) Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Dorset House.
- MIT Technology Review (2022). La Agilidad En América Latina 2021. Una Competencia Core de las Organizaciones Adaptativas. Publicado por Opinno.
- Mulder, H (2020) CHAOS 2020: Beyond Infinity. Recuperado de: <https://www.youtube.com/watch?v=W4tmgk2QZBw&t=1153s>
- Royce, W (1970). Managing The Development Of Large Software Systems. Recuperado de: <http://www-scf.usc.edu/>
- Schwab, K. (2017). La cuarta revolución industrial. Penguin Random House.
- Slotnisky, D. (2016).Transformación Digital. Editorial Distal.
- Snowden, D y Boone M. (2007, Noviembre 07).Un marco para la toma de decisiones del líder, Harvard Business Review, Vol 85, N° 11 pp.110 - 120.
- Stapleton, J (1997). DSDM, Dynamic Systems Development Method: The Method in Practice. Londres, Addison–Wesley.

- Standish Group (1995). The CHAOS Report 1994. Recuperado de: <https://www.standishgroup.com/>
- Standish Group (2011). The CHAOS Report 1994. Recuperado de: <https://www.standishgroup.com/>
- Standish Group (2020). CHAOS Report: Beyond Infinity. Recuperado de: <https://www.standishgroup.com/>
- Sutherland, J & Schwaber, K.(2021). Scrum Guides <https://scrumguides.org>.
- Sutherland, J & Sutherland, J (2019). Scrum. El arte de hacer el doble de trabajo en la mitad del tiempo. Editorial Oceano.
- Toyota Production System (s.f). Recuperado de: <https://global.toyota/en/company/vision-and-philosophy/production-system/>.
- Womack, J; Jonnes, D; & Roos, (1990). The Machine That Changed the World: The Story of Lean Production. Simon & Shuster.

ANEXO

Preguntas de las entrevistas

1. ¿Crees que el involucramiento del C-level impacta de forma directa en el éxito de la transformación?
2. ¿Scrum tiene el potencial de promover buenos equipos?
3. ¿Incide la expertise del Scrum Master o Agile Coach en los resultados?
4. ¿Sirve comenzar una transformación desarrollando equipos Scrum?
5. ¿Scrum tiene el potencial de promover buenos contextos?
6. ¿Scrum tiene el potencial de promover buenos sponsors?
7. ¿Pensas que los agilistas que lideran esas transformaciones están preparados para ese desafío?
8. ¿Que viste en los líderes que logran transformaciones exitosas?
9. ¿Por qué no siempre funciona Scrum?
10. ¿Cómo impacta Scrum en el equipo, el contexto y los sponsors?
11. ¿Qué rol tiene HR en las transformaciones?
12. ¿Cómo cambian las organizaciones?