

**MBA 2014**

# **Metodologías Ágiles y Neurociencias Cognitivas, caminos encontrados.**

**Autor:** Leandro Lorenzo y Losada.

**Tutor:** Patricio Nelson.

"No es el más fuerte el que sobrevive, ni el más inteligente sino  
aquellos que mejor responden al cambio"

**Charles Darwin**

## **AGRADECIMIENTOS**

Agradezco a mi esposa Mónica que siempre me acompaña en todas mis aventuras. A mis hijos Juan Pablo y Martina, que me enseñaron que todo esto no es lo importante. Y por supuesto a Patricio que desde el primer momento creyó en mis cuentos sobre las metodologías ágiles y las neurociencias cognitivas.

## **RESUMEN**

Las Metodologías Ágiles de desarrollo de software surgen hacia el final de la década del '90 cuando muchos líderes de proyectos e ingenieros en software en contraposición a la gestión tradicional de proyectos, comenzaron a reconocer los beneficios de las comunicaciones cara a cara, involucramiento directo del cliente en el proyecto, equipos pequeños e implementaciones frecuentes en ambientes de producción. Los principales exponentes de cada una de ellas se reunieron para dar forma al Manifiesto for Agile Software Development ([agilemanifesto.org](http://agilemanifesto.org)), el cual enumera una serie de valores y principios que conforman las metodologías ágiles y cuyos firmantes se comprometen a cumplir.

Por otro lado desde la academia se presentan las Neurociencias Cognitivas como un área de investigación científica multidisciplinaria que tienen como foco el entendimiento de los mecanismos neuronales responsables de la actividad humana como la autoconciencia, imaginación, y el comportamiento social. Por lo tanto, estudian cómo el cerebro crea la mente.

Estas dos disciplinas que a priori vienen de dos ámbitos totalmente opuestos la empresa y la academia. Están llegando a las mismas respuestas a la hora de recomendar prácticas para incrementar la productividad, adaptación al cambio y trabajo en equipo dentro de las empresas.

En este trabajo se va poder observar como las Metodologías Ágiles para el desarrollo de software están mejor adaptadas (que las técnicas tradicionales) a la forma de funcionamiento de nuestra mente a partir de los últimos estudios en Neurociencias Cognitivas, y como esto influye positivamente en los proyectos que aplican dichas prácticas.

**Palabras Claves:** Metodologías Ágiles, Neurociencias Cognitivas, Software, Cambio, Equipo.

**CONTENIDO**

AGRADECIMIENTOS .....	1
RESUMEN .....	2
INTRODUCCIÓN .....	4
CONTEXTO .....	4
METODOLOGÍAS ÁGILES VS TOP DOWN.....	5
NEUROCIENCIAS COGNITIVAS .....	10
PROPUESTA DE TRABAJO.....	12
CUERPO TEÓRICO.....	15
CAPITULO I: EXTREME PROGRAMMING (XP) .....	15
CAPITULO II: SCRUM .....	20
CAPITULO III: STATUS .....	24
CAPITULO IV: CERTEZA .....	26
CAPITULO V: AUTONOMÍA .....	29
CAPITULO VI: RELACION.....	31
CAPITULO VII: JUSTICIA.....	34
CUERPO EMPÍRICO .....	38
MARCO EPISTEMOLÓGICO .....	38
DISEÑO DE INVESTIGACIÓN .....	38
ANÁLISIS DE LAS ENTREVISTAS .....	43
CONCLUSIONES.....	57
ANEXO .....	62
ANEXO I: ENTREVISTAS.....	62
ANEXO II: GRADO DE ACERCAMIENTO DEL PROYECTO AL MODELO	79
BIBLIOGRAFÍA .....	

## **INTRODUCCIÓN**

### **CONTEXTO**

En los últimos 30 años han surgido cambios en el campo tecnológico y geoeconómico produciendo un fuerte impacto en la forma en que trabajamos, nos relacionamos, aprendemos y disfrutamos.

En el campo tecnológico la llamada “Revolución de la información”, surgida a principios de la década del ´80 con la aparición y adopción en forma masiva de la computadora personal, posibilita a las clases profesionales en occidente el acceso a una herramienta que era para unos pocos privilegiados solo una década atrás. Siguiendo la cronología de los hechos, en 1989 se produce un acontecimiento político-económico que cambiaría la relación de fuerzas a nivel global. La caída del muro de Berlín con el subsiguiente desmoronamiento del bloque soviético en 1991, lo que posibilitó que una porción significativa de la población mundial se incorporara al mercado global. Ese mismo año, Linus Torvalds crea el sistema operativo Linux y lo sube en forma gratuita para su utilización, ampliación y distribución en lo que es una incipiente Internet. Solo 5 años más tarde la mayor parte de las páginas de Internet están alojadas sobre servidores Linux. Hacia finales de la década los sitios de Internet y usuarios de la red crecen de manera exponencial convirtiéndose en una herramienta de comunicación Global. Con la incorporación de China (2001) en la OMC se creó, por primera vez en la historia un mercado que abarca más del 80% de la población mundial. El surgimiento de los teléfonos inteligentes a muy bajo costo en conjunto con iniciativas para acortar la brecha digital en los países en desarrollo ha provocado que los sectores socioeconómicos de la base de la pirámide de estos países se incorporen a este nuevo mercado global no solo como consumidores de bienes y servicios sino como productores de bienes, servicios y conocimiento. Produciendo lo que Thomas Friedman dio a llamar el “Aplanamiento de la tierra” (Friedman, 2008).

En este contexto de boom de las Tecnologías de la Información y el surgimiento de un mercado global, toma un nuevo impulso la producción científica en general y en el campo de la biotecnología en particular. Es aquí en

donde a mi entender dos disciplinas, las Metodologías Ágiles (de desarrollo de software) y las Neurociencias Cognitivas, por caminos totalmente diferentes y supuestamente opuestos (la empresa y la academia) están llegando a las mismas respuestas a la hora de desarrollar herramientas para mejorar la productividad, la adaptación al cambio e interrelación dentro de los equipos de trabajo.

En lo que sigue de esta introducción, primero se analizará el surgimiento de las Metodologías Ágiles en contra posición a las metodologías TOP-Down. Luego veremos qué son las Neurociencias Cognitivas y porque juegan un papel preponderante al momento de explicar el éxito de las Metodologías Ágiles.

## **METODOLOGÍAS ÁGILES VS TOP DOWN**

### **Metodologías Top Down**

La adopción en el ámbito privado de las computadoras en la década del 60, según el paper *CMMI or Agile* (Glazer, Dalton, Anderson, Konrad, & Shrum, 2008) , en un contexto dominado por el COBOL (Common Business -Oriented Language) y debido a la necesidad de procesamientos de datos en forma masiva, dio como resultado la popularización de metodologías de diseño y desarrollo “Top-Down.” Por otro lado en el ámbito público el Departamento de Defensa, principal consumidor de software de la época, sumó las exigencias del “Plan Driven Development”<sup>1</sup> y de manejar un presupuesto de costo fijo hacia sus proveedores. Esto generó la masificación de las metodologías basadas en visiones “Top-Down.”

¿Cómo era el diseño y desarrollo de software mediante metodología “Top-Down”? En 1970 W. W. Royce escribe su afamado paper “Managing the Development of Large Software Systems” (Royce, 1970) en donde por primera vez se describe el Ciclo de Vida del Software (proceso por el cual se llevan a cabo los proyectos de desarrollo de software) para grandes sistemas. Royce definió el proceso “Top-Down” en 7 etapas.

---

<sup>1</sup> Desarrollo de Software Orientados por un plan. Estos tipos de desarrollos toman

Como primer paso se identifican los requerimientos funcionales del sistema. A partir de estos se estima el tiempo y costo del proyecto, por medio del cálculo del esfuerzo (cantidad de tiempo) que llevaría cada etapa de acuerdo al tamaño y cantidad de requerimientos. Esto es elaborado por un Project Manager, el cual se encarga de controlar estas dos variables, tamaño y cantidad de requerimientos, durante todo el proyecto. A partir de este momento se elabora el diseño completo del sistema, luego se realiza la implementación en código, por último se ejecuta el ciclo de pruebas (volviendo a la etapa de diseño en caso de fallas). Cuando todas las pruebas son satisfactorias pasa a la etapa de operación del sistema.

Desde los años '90 se han hecho muchas críticas a este tipo de metodologías, por la imposibilidad de poder cumplir con los tiempos y los plazos de los proyectos de software de manera recurrente (The Standish Group, 1995). Dichas críticas están centradas en primer lugar que las estimaciones tienden a ser optimistas y los desvíos pueden llegar a superar el 100% de las mismas y en segundo lugar que existe una falta de control de cambios en los requerimientos lo que agrava el tema de los desvíos. Por último no existen medidas de avance. Se suele confundir el tiempo transcurrido con el avance del Proyecto.

Para superar estos inconvenientes a mediados de la década del '90 las empresas de desarrollo de software comenzaron a implementar las prácticas impulsadas por el PMI (Project Management Institute). Este organismo por medio de la Guía PMBOK y la Certificación PMI, convalidan que los Líderes de Proyecto que cuenten con dicha certificación tienen el conocimiento necesario para implementar las mejores prácticas en el gerenciamiento de proyectos. Estas prácticas hacen un fuerte hincapié en la Planificación y Control del Alcance, Tiempos, Costos y Riesgos del proyecto que son responsabilidades del Líder de Proyecto. (Bhargav, 2015)

El cumplimiento de las prácticas y procesos recomendados en el PMBOK, generan una carga elevada de documentación y planificación, que si bien son requisitos fundamentales en los proyectos de ingeniería, en los proyectos de

software suelen ser efectivos en pocas oportunidades. En muchos casos estos proyectos suelen encontrarse con:

- a) Falta de completitud en los requerimientos.
- b) Falta de involucramiento por parte del usuario.
- c) Expectativas poco realistas.
- d) Constante cambios en los requerimientos y especificaciones
- e) Desconocimiento de la tecnología. (Dunbar, 2016)

### **Metodologías Ágiles**

Como contrapartida para poder mitigar estos problemas comunes en los proyectos de software comenzaron a surgir diversas adaptaciones de la metodología IIDD (Iterative and Incremental Design and Development)<sup>2</sup>. Un ejemplo paradigmático es el Modelo de Desarrollo de software en espiral presentado por Barry Boehm en 1988 (B. W. Boehm, 1988) En donde por primera vez se define la planificación, la determinación de objetivos y la detección de riesgos como parte de cada fase, que de forma Iterativa va construyendo el sistema.

Durante la década de los 90 ganaron popularidad nuevas adaptaciones entre las cuales se encuentran Rapad Prototype, Rapad Application Development (RAD) y Rational Unified Process (RUP). Durante esta década surgieron muchos de los mayores exponentes de las metodologías ágiles, dentro de proyectos de gran envergadura. eXtreme Programming (XP) surgió en un proyecto de Chrysler en 1996 y Feature Driven Development en United Overseas Bank en Singapur (uno de los mayores bancos de Asia). Hacia el final de la década muchos líderes de proyectos e ingenieros en software reconocían los beneficios de las comunicaciones cara a cara,

---

<sup>2</sup> Metodología que promover un movimiento orientado hacia el desarrollo ágil, liviano con iteraciones adaptables el cual garantizaría resultados en el corto plazo y una visibilidad mayor en la velocidad de entrega de "Business Value".



involucramiento directo del cliente en el proyecto, equipos pequeños e implementaciones frecuentes en ambientes de producción. En este período es donde surgieron nombres hoy conocidos como Scrum, Crystal, FDD, etc. Con el surgimiento de todas estas nuevas metodologías con raíz común en el modelo de espiral y con un renovado entusiasmo los principales exponentes de cada una de ellas se reunieron para dar forma al Manifiesto for Agile Software Development ([agilemanifesto.org](http://agilemanifesto.org)), el cual deja constancia de una serie de En el Manifiesto Ágil se expresa:

“Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

- Individuos e interacciones por sobre procesos y herramientas
- Software Funcionando por sobre documentación extensiva
- Colaboración con el Cliente por sobre Relación Contractual
- Respuesta al cambio por sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.” (Kent Beck, 2001)

Acto seguido los integrantes del Manifiesto Ágil definen los 12 principios que deben guiar los procesos de desarrollo de software.

- “Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los Procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcionando es la medida principal de progreso.
- Los Procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.”

### **En resumen**

Las Metodologías Ágiles de desarrollo de software surgen como una respuesta a los problemas que las metodologías Top-Down encuentran en los proyectos de desarrollo de software, estas últimas suelen provenir de ámbitos como la Ingeniería Civil o La gestión de proyectos donde el conocimiento y la estandarización de los requerimientos, procesos y herramientas permiten una planificación anticipada y en detalle de todo el proyecto, donde existe un rol de planificador y rol de ejecutor de las tareas.

Los proyectos de software a causa de la inestabilidad de los requerimientos, sumado a los procesos y herramientas en constante evolución, requieren trabajadores especializados capaces de tomar decisiones, probar y experimentar con las soluciones técnicas, colaborar día a día con el cliente en la definición, validación y aceptación de lo producido y finalmente estar

preparados para el cambio como parte del proceso mismo. Lo que explica gran parte del éxito de las Metodologías Ágiles.

## NEUROCIENCIAS COGNITIVAS

Las neurociencias cognitivas son un área de investigación científica multidisciplinaria que reconoce en las neurociencias y las ciencias cognitivas las bases de su metodología de investigación, y el desarrollo de su cuerpo teórico.

Tienen como foco el entendimiento de los mecanismos neuronales responsables de la actividad humana como la autoconciencia, imaginación, o el comportamiento social. Las neurociencias cognitivas estudian cómo el cerebro crea la mente.

En los últimos 30 años se han hecho avances significativos en las neurociencias, gracias a las nuevas tecnologías, como la imagen por resonancia magnética funcional (IRMf), se puede observar en tiempo real como un cerebro reacciona a estímulos, y como las neuronas forman nuevos enlaces a partir de experiencias. Han surgido nuevas teorías sobre la capacidad que tiene la mente de alterar literalmente la fisiología del cerebro (Doidge, 2007). Por otro lado se han desarrollado nuevos esquemas teóricos que permiten aproximaciones a los desafíos planteados por las ciencias cognitivas.

En este contexto David Rock, experto en neurociencias cognitivas, desarrolla un modelo en donde existen 5 conceptos dentro de la experiencia social, que son tratados por el cerebro como si fueran objetos de supervivencia.

Estos conceptos componen el denominado Modelo SCARF (Rock, 2008): Status, Certeza, Autonomía, Relaciones y Justicia. (Rock, YOUR BRAIN AT WORK, 2009). En breve, según Rock, el status representa la importancia relativa de una persona en relación a otros. La certeza es la capacidad de predecir el futuro. La autonomía proporciona una sensación de control sobre los eventos. Las relaciones es una sensación de seguridad con otros

(amigo/enemigo). La justicia es la percepción de intercambios justos (equitativos) entre personas.

Como se mencionó anteriormente las experiencias sociales del modelo SCARF son interpretadas por el cerebro como amenazas o recompensas de manera análogas a lo que podríamos observar en las dicotomías: sed/beber o hambre/comer. Es decir como elementos básicos de supervivencia. Esto tiene una implicancia revolucionaria en como interpretamos los comportamientos sociales e individuales, generando un giro copernicano en la pirámide de Maslow. La Piramide de Maslow fue una teoría desarrollada en la década del 60 la cual describe una jerarquía de necesidades para el ser humano en donde las primeras (base de la pirámide) se centran en las necesidades físicas básicas para la supervivencia como la alimentación, el descanso o el sexo. Hasta llegar al final de la jerarquía en donde se encuentra la autorrealización. Las necesidades sociales se encuentran en un puesto medio.

Si nos basamos en el modelo SCARF el reconocimiento, la confianza y las creencias pasarían a ocupar un lugar en la base de la pirámide. Generando un quiebre en la reconocida teoría de la Psicología Humanista es decir las necesidades sociales serían básicas para la supervivencia del ser humano.

Considero que las empresas, y por sobre todo las orientadas al desarrollo de software, pueden beneficiarse, mejorando la productividad, la adaptación al cambio e interrelación de sus empleados, a partir de metodologías de trabajo que se alineen con las recomendaciones del modelo SCARF. Estableciendo procesos y actividades que protejan el status de los integrantes permitiendo una interacción horizontal, generando claras expectativas y objetivos para incrementar la certeza en sus miembros, con procesos de trabajo que posibiliten a los empleados tomar decisiones y responsabilidades incrementando su autonomía, haciendo hincapié en la comunicación y el contacto cara a cara incrementando las relaciones al interior del grupo, y tomando decisiones en equipo por consenso incrementando la percepción de justicia en ellas.

A mi entender las Metodologías Ágiles se alinean con estas prácticas en contraste a las metodologías Top-Down de manejo de proyectos que tradicionalmente se implementan en las empresas.

## **PROPUESTA DE TRABAJO**

### **Preguntas de investigación:**

1. ¿En qué medida las Metodologías Ágiles sistematizan y ponen en práctica los hábitos y comportamientos que son recomendados por el Modelo SCARF como herramientas para mejorar la productividad, adaptación al cambio y trabajo en equipo?

1.a ¿Cuáles la correspondencia en las prácticas de las Metodologías Ágiles y el Modelo SCARF?

1.b ¿Dentro de los proyectos estudiados cómo y en qué grado se implementan las prácticas y procesos desarrollados en el Cuerpo Teórico?

1.c ¿Cuál es el grado de correspondencia de las recomendaciones del modelo SCARF implementadas por medio de las metodologías ágiles y la efectividad en productividad, adaptación al cambio y trabajo en equipo en los proyectos estudiados, según sus protagonistas?

### **Objetivo General:**

El objetivo general de esta investigación es determinar el sustento científico de las prácticas de las metodologías ágiles por medio de su correspondencia con las últimas investigaciones en Neurociencias Cognitivas. Y el aporte que realizan en términos de productividad, adaptación al cambio y trabajo en equipo dentro de las empresas.

### **Objetivos Específicos:**

Se plantean los siguientes objetivos específicos para llevar a cabo la investigación:

- 1- Describir Procesos, Roles y Prácticas de las dos principales Metodologías Ágiles (XP y Scrum) que se utilizan en la industria del Desarrollo de Software.
- 2- Describir el modelo SCARF creado por David Rock y su impacto en adaptación al cambio y trabajo en equipo dentro de las empresas.
- 3- Vincular las Metodologías Ágiles (XP y Scrum) con el Modelo SCARF teniendo en cuenta como las Metodologías implementan las recomendaciones del Modelo.
- 4- Llevar a cabo un estudio de campo por medio de la selección de 3 casos de proyectos de software que a priori representan 3 puntos en la distancia que existe en la adopción de las metodologías ágiles para la gestión de desarrollo de software (Adopción total, parcial, y nula).
- 5- Realizar un análisis de los casos seleccionados y evaluar el Grado de acercamiento de cada caso a las prácticas planteadas por las Metodologías Ágiles.
- 6- Analizar en cada uno de los casos el impacto de las prácticas implementadas a través del Modelo SCARF.
- 7- Analizar la percepción de los entrevistados del desempeño que cada proyecto tiene en términos de la productividad, adaptación al cambio y trabajo en equipo.
- 8- Establecer un vínculo entre las recomendaciones del modelo SCARF implementadas por medio de las Metodologías Ágiles y la efectividad en productividad, adaptación al cambio y trabajo en equipo en los proyectos estudiados, según sus protagonistas

### **Desarrollo del Trabajo**

El trabajo se divide en 3 partes. Primera parte, el **Cuerpo Teórico** el cual en una primera instancia se realizará una descripción de los Procesos, Roles y Prácticas de dos de las Metodologías Ágiles más utilizadas (XP y Scrum). En una segunda instancia se examinarán los 5 conceptos de la experiencia social

desarrollados por David Rock, y para cada uno de estos conceptos se establecerá un vínculo con los valores, principios y prácticas de las Metodologías Ágiles. Segunda parte el **Cuerpo Empírico**, donde a partir de un Marco Epistemológico y un Diseño de Investigación se desarrollará un trabajo de campo en el cual indagará el vínculo entre las Metodologías Ágiles y los conceptos de David Rock en diferentes proyectos de desarrollo de software. Para finalizar, **Conclusiones** se intentará dar respuestas a las preguntas de investigación por medio de lo expuesto a lo largo del presente trabajo.

## CUERPO TEÓRICO

### CAPITULO I: EXTREME PROGRAMMING (XP)

El XP es una Metodología de desarrollo de software cuyo principal foco es la mejora de calidad del producto (aplicación/sistema) y dar respuesta rápida a los cambios de los requerimientos del cliente. Como parte del movimiento de Metodologías Ágiles de desarrollo de software se concentra en realizar entregas frecuentes, en pequeños ciclos de desarrollo lo que contribuye a incrementar la productividad. Al finalizar cada entrega el dueño del sistema puede solicitar nuevos requerimientos y/o cambios del mismo.

A continuación voy a enumerar las prácticas que componen XP, alineadas con los valores y principios del Manifiesto Ágil presentado en la introducción. Luego describiremos los roles que implementan dichas prácticas y las estrategias que utiliza esta metodología para el desarrollo de software.

#### Prácticas:

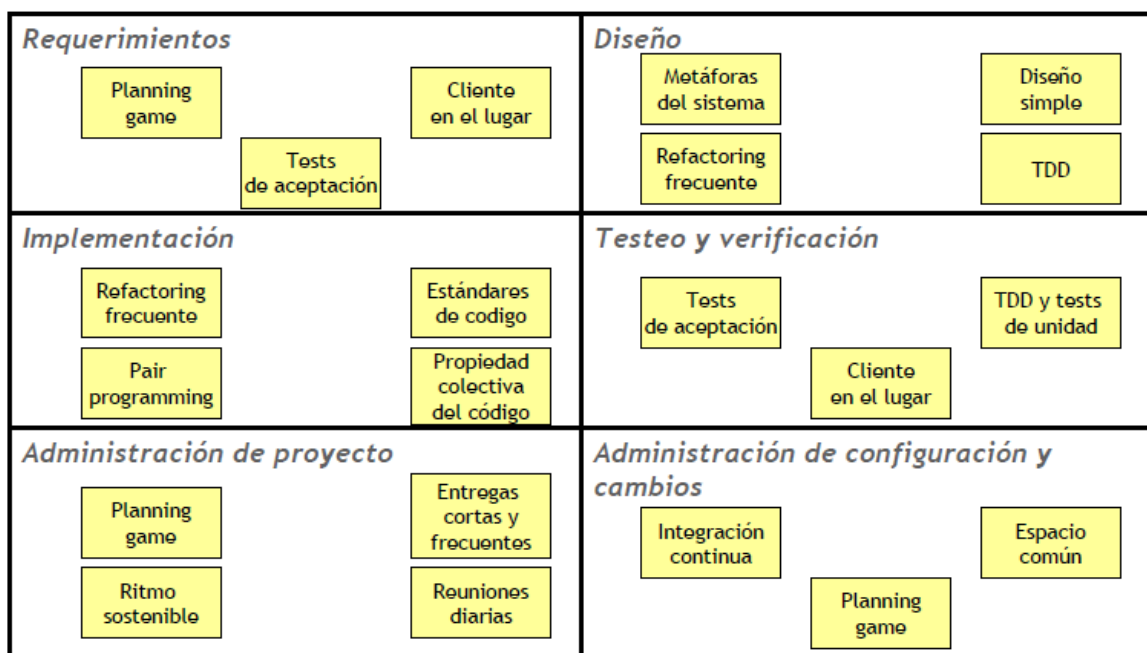


Figura 1. Prácticas llevadas a cabo en XP, agrupadas por disciplina. (Lorenzo y Losada, 2009)

**Diseño simple:** Énfasis sobre el diseño más simple: no implementar ahora cosas que tal vez vaya a necesitar más adelante.



**TDD (test driven development):** Se escribe el test de unidad antes que el código. Es decir, se comienza por diseñar la forma de interacción con la unidad a construir y el comportamiento esperado.

**Refactoring:** Reestructurar el código frecuentemente con el objetivo de mejorar y simplificar el diseño (sin cambiar la funcionalidad).

**Pair programming:** 2 personas trabajan en conjunto en una computadora: uno escribe y el otro realiza revisión de código on-the-fly, detecta defectos y ayuda a resolver bloqueos del que está programando. Se alternan los roles con frecuencia y se rotan las parejas.

**Propiedad colectiva del código:** Cualquiera puede cambiar cualquier parte del código en cualquier momento (siempre verificando que no falle ningún test).

**Integración continua:** Nuevas integraciones a medida que el código está listo. En general se hacen en un servidor dedicado (build server) y en forma automática. Un proceso baja los cambios subidos al repositorio de código, compila el sistema y corre todos los tests. Si falla alguno de estos puntos, notifica a todo el equipo (mail) que el build está roto.

**40 horas semanales (o ritmo sostenible):** Se trabajan 40 horas por semana como máximo. No hay horas extras. Se considera que un proyecto en el cual se realizan muchas horas extras por un período prolongado de tiempo el cansancio y el stress impactan sobre la calidad del producto. Por otro lado, ese ritmo de trabajo no es sostenible a largo plazo ocasionando que los desarrolladores se desmotiven y/o cambien de trabajo.

**Cliente en el lugar (on-site):** El cliente (o un proxy) debe estar presente y disponible full time durante todo el proyecto. Si no es posible, al menos disponible bajo demanda por teléfono.

**Espacio común:** Todo el equipo trabaja en un espacio común (y en lo posible separado de otros equipos).

**Reunión diaria (Daily stand up meeting):** Reuniones cortas, generalmente todos parados, al principio del día para sincronizar las tareas entre los miembros del equipo.

**Roles:**

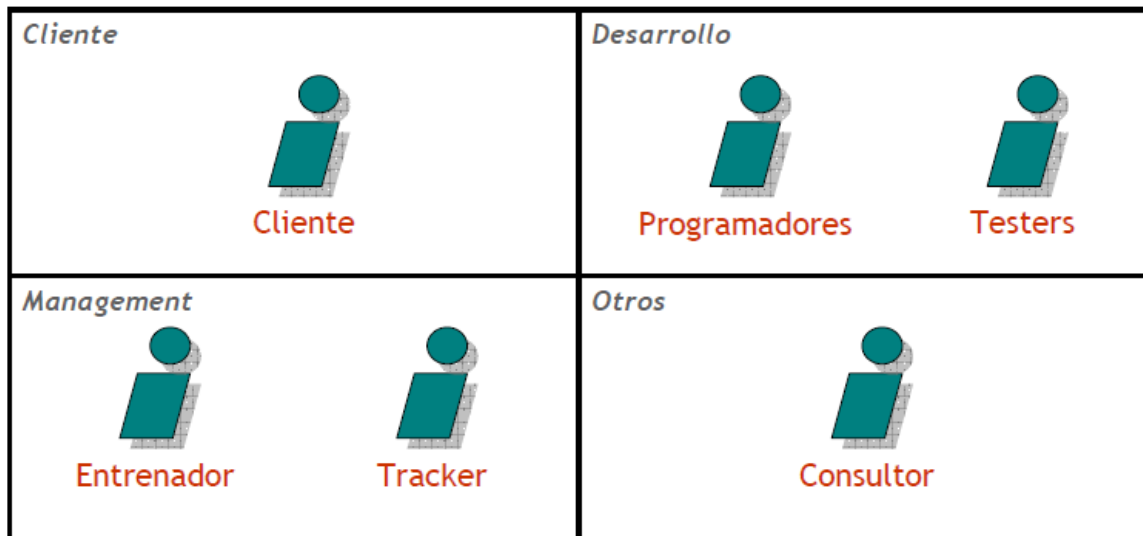


Figura 2. Roles en XP, agrupado por tareas. (Lorenzo y Losada, 2009)

**Programador:** Escribe tests de unidad, diseña y programa. Realiza refactorings del código. Identifica tareas y estima esfuerzo.

**Cliente:** Escribe user stories y tests de aceptación. Elige qué stories serán implementadas en el release y en la siguiente iteración.

**Tester:** Ayuda al cliente a escribir los test funcionales y los programa o ejecuta.

**Tracker:** Colecta métricas, fundamentalmente de progreso durante las iteraciones.

**Entrenador:** Responsable de todo el proceso. Enseña al resto del equipo.

**Consultor:** Consultor en conocimientos técnicos específicos. Coaching.

**Artefactos:**

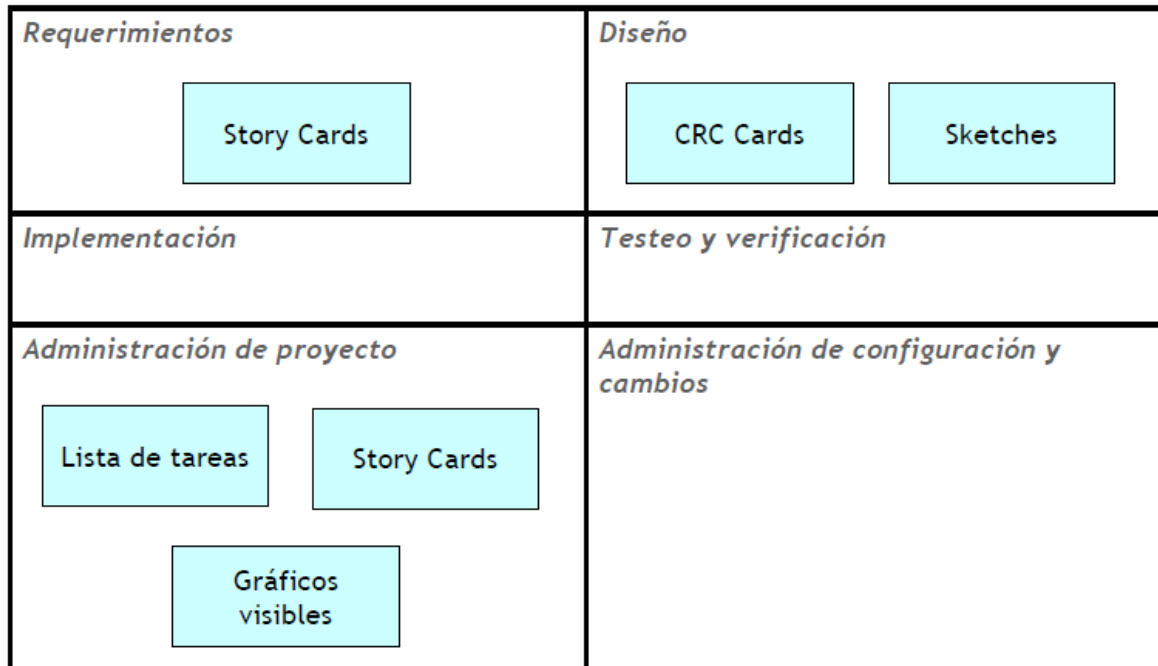


Figura 3. Artefactos en XP, agrupados por disciplina. (Lorenzo y Losada, 2009)

**Story cards:** tarjeta de índice o post-it escrita a mano. Cada *story card* captura una *user story*. Las user stories capturan features, mejoras, requerimientos no funcionales que el cliente / usuario necesita. En general, cada user story debería poder ser implementada en un tiempo que varía entre 1 día y 3 semanas. No es el objetivo de una story card capturar en detalle el requerimiento. Las story cards deben interpretarse como “una promesa de hablar” entre el equipo de desarrollo y el cliente. Las story cards se descartan una vez que han sido implementadas.

**Lista de tareas:** durante la planificación de una iteración, el equipo genera una lista de las tareas necesarias para cada una de las story cards seleccionadas para la iteración. Cada tarea debe ir acompañada de la estimación de esfuerzo en horas, generalmente en el rango entre 1 y 2 días.

**Gráficos visibles:** Se colectan métricas sencillas de tomar y se muestran gráficamente en un lugar visible para todo el equipo y para cualquiera que quiera verlas. Ejemplo: velocidad.

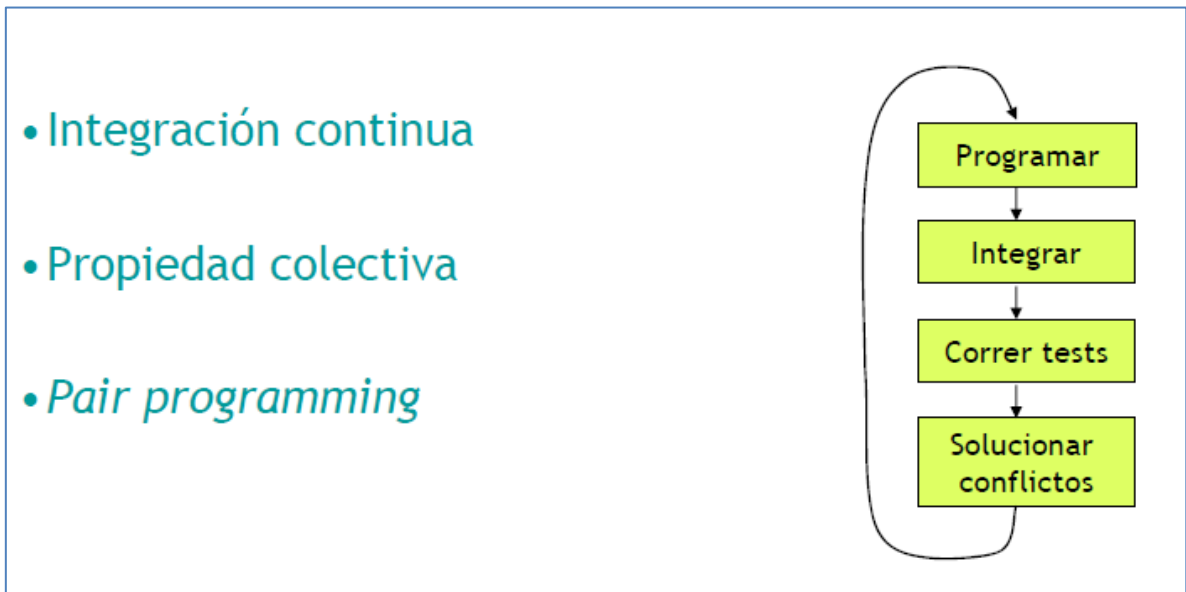
**Proceso de desarrollo:**

Figura 4. Proceso de desarrollo de XP. (Lorenzo y Losada, 2009)

**Integración continua:** El programador trabaja en su entorno aislado de los cambios que están haciendo en simultáneo el resto de sus compañeros. Cuando se termina una porción de código, se la integra con la última versión del sistema y se corren todos los tests. Lo más eficiente es tener soporte de una herramienta para hacerlo automáticamente varias veces por día.

**Propiedad colectiva:** Todo programador puede cambiar cualquier porción del código en cualquier momento. Sólo puede realizarse sin riesgo si están escritos los tests automáticos sobre todo el sistema. Es una práctica necesaria para garantizar que se refine permanentemente el diseño y se mejoren las porciones del código que no hayan quedado bien escritas (complejas de entender). Un refactoring realizado por una herramienta (ej: renombrar un método) puede modificar muchos archivos distintos. Esta práctica también tiene como ventaja fomentar el conocimiento integral de todo el sistema por parte de todo el equipo.

**Pair programming:** Dos programadores trabajan juntos para analizar, diseñar y programar un feature. Mientras uno de los dos escribe, el otro tiene una visión

más amplia y puede descubrir errores (de sintáxis o de diseño) tan pronto como ocurren.

Ambos programadores se ayudan mutuamente para superar bloqueos en la tarea que se está realizando. Puede usarse entre una persona muy experimentada y un junior como forma de coaching. Los roles se invierten frecuentemente: el que estaba escribiendo pasa a observar y el que estaba observando pasa a escribir. Las parejas también rotan en su composición: esto facilita que el conocimiento fluya y se expanda a través de todo el equipo.

El objetivo primario detrás de esta práctica es la calidad del código que se escribe y el aumento de la productividad al reducir el tiempo invertido en corregir.

## **CAPITULO II: SCRUM**

El Scrum es una metodología de desarrollo de software iterativa e incremental, la cual suscribe al manifiesto ágil y encuentra sus raíces en Lean y el Toyotismo. Como todo proceso empírico está pensado de manera que se auto evalúe y mejore en forma constante. Hace foco en la comunicación cara a cara de los miembros del equipo y la auto-organización del mismo en pos de objetivos comunes que se renuevan en cada iteración, su fuerte es la administración y el seguimiento de proyectos ágiles, por estas características sus prácticas se complementan con las de XP.

A continuación voy a enumerar las prácticas que componen Scrum, alineadas con los valores y principios del Manifiesto Ágil presentado en la introducción. Luego describiremos los roles que implementan dichas prácticas y los artefactos que utiliza esta metodología para tal fin.

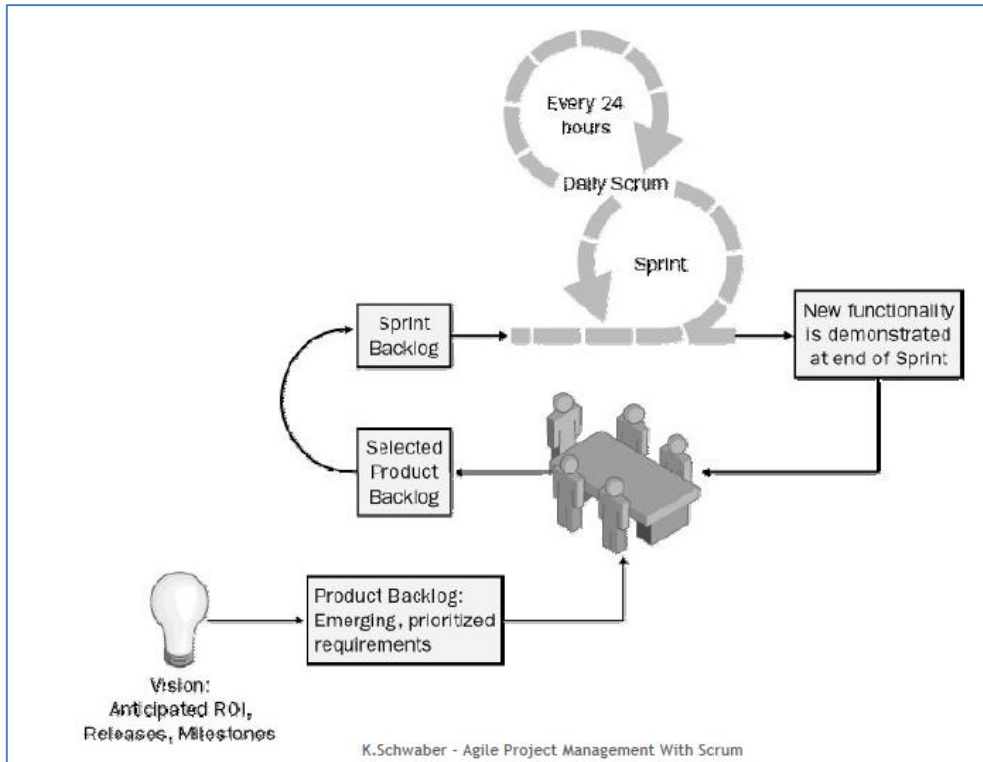
**Prácticas del Proceso:**

Figura 5. Proceso de desarrollo en Scrum. (Schwaber, 2004)

**Sprint planning:** El Product Owner selecciona los requerimientos prioritarios del Product Backlog que desea incluir. El Team determina cuanto puede convertir en funcionalidad al final del Sprint. Se agregan las tareas al Sprint Backlog (que sigue evolucionando a lo largo del Sprint).

**Daily Scrum:** El Objetivo es sincronizar el trabajo de los miembros del Team y resolver inconvenientes a medida que surgen. Enfocada en contestar 3 preguntas:

¿Qué hice para este proyecto desde la última reunión?

¿Qué pienso hacer hasta la próxima reunión?

¿Qué impedimentos tuve para cumplir mis compromisos?

**Sprint review:** El Objetivo es validar el cumplimiento de expectativas y definir próximos pasos. Se demuestra la funcionalidad completa al ProductOwner y otros Stakeholders del proyecto. Se presenta qué salió bien y qué salió mal en el Sprint. Se toma nota de los cambios que pide el cliente.

**Sprint retrospective:** El Objetivo es analizar el resultado del Sprint y proponer mejoras en el proceso para el siguiente Sprint. Las reuniones de Planificación de Sprint, Daily Scrum, Sprint Review y Sprint Retrospective constituyen las prácticas empíricas de inspección y adaptación de Scrum.

Roles:

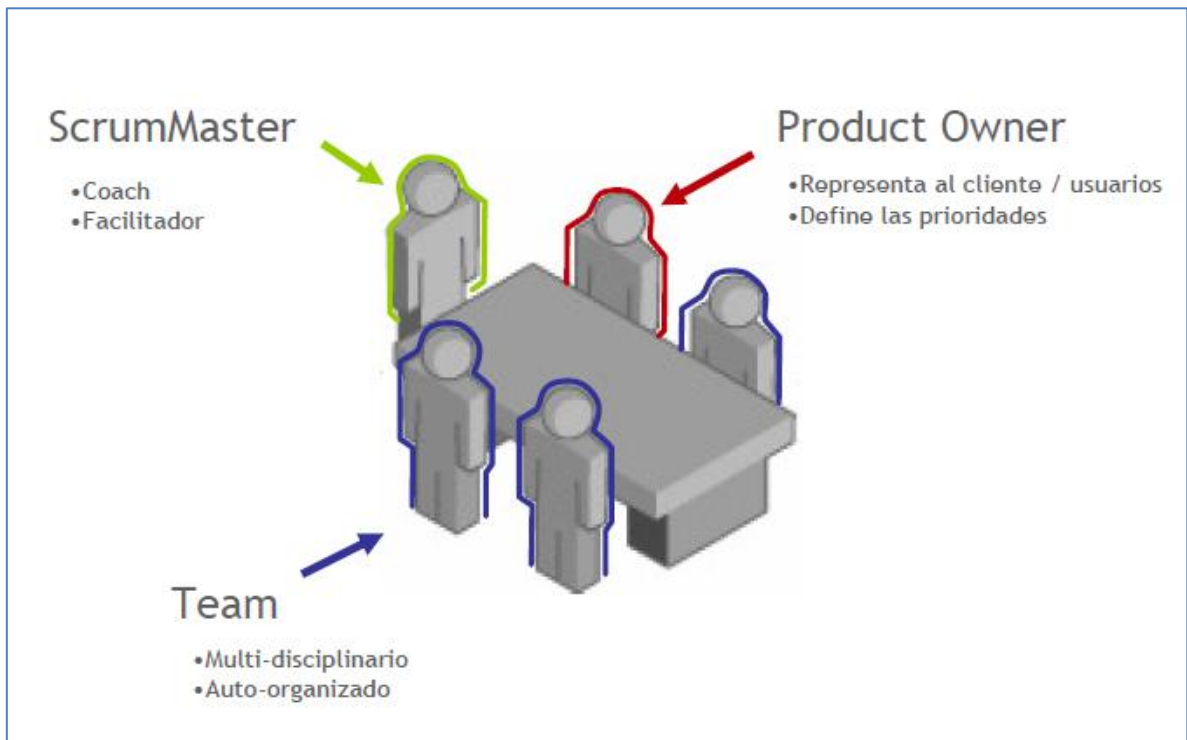


Figura 5. Roles en Scrum . (Lorenzo y Losada, 2009)

**Scrum Master:** Dueño del proceso, es responsable de enseñar Scrum a los involucrados en el proyecto, implementar Scrum de forma que encaje en la cultura de la organización, garantizar que todos cumplan las reglas de Scrum.

**Product Owner:** Dueño de la definición de “éxito” del proyecto, administra el ROI priorizando los requerimientos, crea la lista inicial de requerimientos

(Product Backlog), establece los objetivos de ROI, establece el plan preliminar de entregas, es responsable de usar el Product Backlog para garantizar que la funcionalidad que más valor agrega al negocio sea construida primero.

**Team:** Autogestionado y multidisciplinario, dueño de los procesos de producción e ingeniería. Define colaborativamente cómo transformar el Product Backlog en un incremento de funcionalidad al final de la siguiente iteración, es colectivamente responsable del éxito de cada iteración y del proyecto en su totalidad. Se recomienda que no sea de más de 7 o 10 integrantes.

Artefactos:

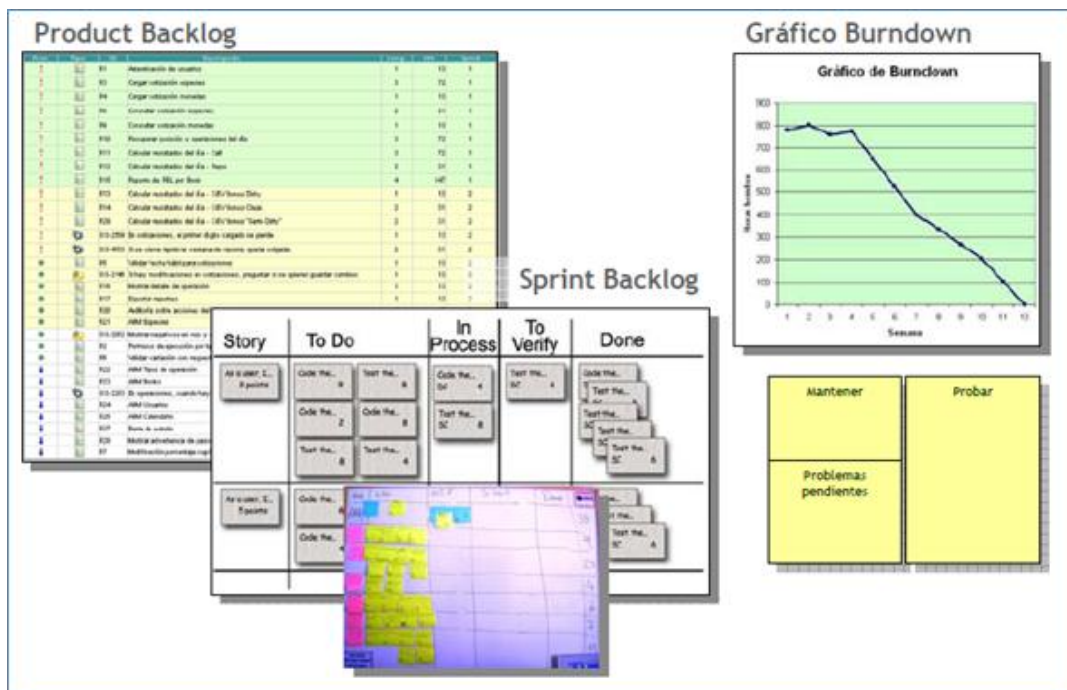


Figura 6. Artefactos en Scrum. (Lorenzo y Losada, 2009)

**Product Backlog:** Requerimientos funcionales y no funcionales de alto nivel, priorizados para maximizar el ROI. División tentativa en Versiones. Estimación inicial de esfuerzo para cada requerimiento. Especialmente para los de mayor prioridad. Evoluciona a lo largo del proyecto.

**Sprint Backlog:** Porción del Product Backlog seleccionada para un Sprint. Tareas que define el Team para implementar los requerimientos seleccionados.



Debe reflejar el avance del equipo día a día. Permanentemente visible. Durante el Sprint, sólo puede ser modificado por el Team.

**Gráfico de Burndown:** A nivel Versión: muestra el trabajo restante al final de cada Sprint en alguna medida de tamaño relativa (ej: story-points). A nivel Sprint: muestra la cantidad de horas hombre restantes al final de cada día.

**Incremento de funcionalidad:** Empaquetado como ejecutable. Sólo se incluye funcionalidad done, el significado de done puede variar de proyecto en proyecto.

### CAPITULO III: STATUS

En los siguientes 5 capítulos voy a describir en profundidad los elementos del modelo SCARF, presentado por David Rock, y los valores, principios y prácticas de las metodologías ágiles presentadas que se interrelacionan con cada uno de ellos.

El primer elemento es el status. Este, según Rock, es uno de los principales conductores del comportamiento tanto en el trabajo como fuera de él. Una percepción de que el propio status está subiendo, aunque sea pequeña, activa los circuitos de recompensa del cerebro. En cambio una percepción de que nuestro status está bajando activa los circuitos de amenaza.

Como señala David Rock (Rock, 2009, págs. 184-202) Un incremento en el status es una sensación muy gratificante. Los niveles de dopamina y serotonina suben, generando una sensación de felicidad y los niveles de cortisol bajan, signo de un nivel bajo de stress. Los niveles de testosterona también se incrementan, los cuales mejoran las habilidades cognitivas y crea confianza en uno mismo. Con los cambios de estas hormonas en el cerebro se incrementa la cantidad de conexiones por hora que este realiza. Esto significa que una recompensa hacia el status ayuda a procesar más información con menor esfuerzo. Con la reducción de las amenazas producto de las emociones positivas se posibilita una utilización plena del córtex prefrontal sin interrupciones.

En mi experiencia, dentro de las organizaciones las personas prestan mucha atención en construir y proteger su status. Cuando este se percibe como amenazado, esto genera tensiones y sensación de amenaza. Viéndose agravado en los lugares de trabajo que poseen una fuerte impronta jerárquica. Como vimos anteriormente uno de los valores que enumera el Manifiesto Ágil es la importancia de individuos e interacciones por sobre procesos y herramientas. Esto se expresa a través del Principio 1-“*Los proyectos deben ser desarrollados en torno a individuos motivados dándoles el apoyo que necesitan y confiándoles la ejecución del trabajo, generando una sensación de igualdad y confianza entre los individuos que forman parte del equipo*”. Como prácticas implementadas por las Metodologías Ágiles que contribuyen con los valores y principios mencionados podemos encontrar al *Stand Up Meeting*, que es una reunión diaria al principio del día entre todos los miembros del equipo para sincronizar las tareas. Por medio de esta práctica los integrantes refuerzan una sensación de igualdad en cuanto a status al estar en presencia de otros miembros y observar los avances y problemáticas encontradas por cada uno de ellos. Al mismo tiempo que posibilita a contribuir en la resolución de problemas de personas con mayor seniority, igualando el nivel de status hacia arriba de los miembros. De esta forma se puede apreciar cómo se reduce el nivel de amenaza incentivando la colaboración y desarmando las barreras que las jerarquías pueden llegar a formar al interior del equipo.

Con el *Sprint planning*, reunión de planificación mensual donde los integrantes del equipo se autoasignan las tareas a ser realizadas sucede algo similar. En el sentido en que todos los integrantes del equipo colaboran como pares. Esto es un contraste muy fuerte con respecto a las metodologías Top-Down donde un Jefe/Líder, asigna las tareas haciendo una diferenciación de status primero entre él y el resto del equipo y segundo al interior del equipo al discriminar quienes son los que están capacitados a realizar que tareas. Al autoasignar la responsabilidad sobre la tarea ya no es dada por otro sino que es asumida como propia elevando el nivel de estatus en las personas con menos autonomía del equipo. Por otro lado se desvanecen las diferencias jerárquicas entre los miembros, ya los individuos dejan de estar relacionados con las

tareas de acuerdo al seniority y comienzan a relacionarse más en torno a las preferencias.

Otra práctica utilizada en Scrum que contribuye a incrementar el nivel de status es el *Sprint Review*, reunión cuyo objetivo es el de validar el cumplimiento de expectativas y definir próximos pasos. En esta práctica todos los miembros del equipo tienen la posibilidad de mostrar y explicar la funcionalidad que fue agregada al producto a los distintos stakeholders del proyecto. Esto produce un incremento del status en todos los miembros al recibir devoluciones positivas (recordemos que solo la funcionalidad que está probada y considerada como “*Done*” se expone en la reunión) siendo todos ellos y cada uno de ellos responsable del éxito. En caso de un mal desempeño en la cantidad y calidad de las funcionalidades entregadas la responsabilidad es compartida por todo el equipo lo que amortigua la sensación de amenaza que podría provocar.

#### **CAPITULO IV: CERTEZA**

El cerebro, según el neurólogo Jeff Hawkins, ansía tener certeza. La sensación de falta de certeza acompañado por la pérdida de control puede generar fuertes respuestas por parte del sistema límbico. Para evitar esto el cerebro actúa como si fuera un dispositivo diseñado para predecir. Recursos neuronales son destinados en forma masiva a predecir que va suceder a cada instante. Como explica Jeff Hawkins (Hawkins & Blakeslee, 2004), el cerebro recibe patrones del mundo exterior, los guarda en la memoria y realiza predicciones combinando lo que ha observado en el pasado y lo que está sucediendo en este momento. Para este autor la predicción no es una función más del cerebro sino la principal función del neo-cortex y la base para la inteligencia.

Uno no ve, solo predecimos lo que vamos a estar viendo. Tampoco escuchamos solo predecimos lo que vamos a estar escuchando. El cerebro es bueno reconociendo patrones aunque estén incompletos, y conjeturar su significado a partir de un contexto dado. Esto lo realiza con todos los sentidos, por eso podemos escuchar frases aunque la música esté muy alta o leer un

texto sin la necesidad de leer letra por letra. Por otro lado siempre existe la información del contexto que a nivel subconsciente el cerebro utiliza para sus predicciones. Por ejemplo el cerebro predice el resultado de nuestros movimientos. Por este motivo tenemos esa sensación de torpeza al tirar un objeto sin intención o que es lo mismo sin haberlo podido predecir. La habilidad de predecir y obtener una respuesta positiva a partir de esta predicción genera una fuerte sensación de recompensa que se vuelve casi adictiva. Esto se puede ver claramente cuando el cerebro predice patrones musicales, resuelve rompecabezas o juega al Candy Crush.

Dentro del ámbito corporativo la predicción juega un papel central tanto en el pensamiento estratégico como en la estimación de venta de algún servicio o producto. Al igual que el cerebro las empresas utilizan información del pasado para predecir el futuro aunque más no sea por medio de tendencias.

La incertidumbre se manifiesta en el cerebro como una amenaza a la propia vida. Cuando no es posible prevenir el desenlace de una situación de forma inmediata la amígdala dispara una alerta que aumenta el nivel de atención. La incertidumbre es como la imposibilidad de crear un mapa completo de la situación creando una sensación de incomodidad por falta de información de contexto. Un ejemplo de esto es el nivel de incertidumbre que generan las comunicaciones vía mail en contraste de las comunicaciones cara a cara. La falta de objetivos claros y la constante redefinición de los mismos es otra fuente de incertidumbre muy importante en el ámbito corporativo. (The Standish Group, 1995)

De los cuatro valores enumerados por el manifiesto Ágil, dos están directamente enfocados a reducir la incertidumbre: “*Colaboración con el Cliente por sobre Relación Contractual*” y “*Individuos e interacciones por sobre procesos y herramientas.*” Estos valores se ven reflejados en los siguientes principios y prácticas: 1-“*Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.*” Tanto en Scrum como en XP existe el rol de *Product Owner* el cual forma parte del equipo de proyecto y sus principales tareas son el de crear la lista de requerimientos, establecer una prioridad en los mismos, trabajar con el equipo

para clarificación de los mismos y elaboración de los casos de prueba. Como se podrá observar este contacto y colaboración permanente que tiene el *Product Owner* con el equipo reduce la incertidumbre del mismo a tal grado que las metodologías Ágiles tienen como principio también: 2- *“Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente”*.

La comunicación cara a cara como se mencionó es otro valor importantísimo expresado a través del principio 3 - *“El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.”* Esto se pone en práctica de forma diaria a través del *Daily stand up meeting*, *Sprint planning* y el *Pair programming*. Todas prácticas en donde se establece una comunicación directa entre los miembros del equipo.

Por último el cambio y la comunicación constante no podría ser posible y generaría un grado alto de incertidumbre si no fuera por el siguiente principio Ágil: 4- *“Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.”* Esto se logra mediante la práctica de *Time Boxing*. Para comenzar, gran parte de las metodologías ágiles promueven una semana laboral de 40hs., un día laboral de 8hs. y Sprint (Iteraciones) de 4 semanas (este último puede variar entre 2 y 6 según el proyecto). Cada una de las actividades tienen un punto y un espacio de tiempo fijo asignado dentro del proyecto. Esto genera un efecto ordenador y optimizador de esfuerzos. Por otro lado aporta seguridad, ya que puedo predecir qué voy a hacer en un contexto complejo, donde cambian las exigencias del cliente. Por ejemplo: Durante el Sprint Planning, que dura 8hs, el equipo se compromete en la funcionalidad a ser desarrollada en las próximas 4 semanas. A partir de ese momento no se podrá agregar o quitar funcionalidades a la planificación del Sprint (a menos que el equipo, no el Product Owner, o el Scrum Master, acepte un trade off.). Al cabo de estas 4 semanas el equipo presentará las funcionalidades que están completas (*Done*). Cualquier cambio o incremento de funcionalidad será analizado en el próximo *Sprint Meeting*. Por último, dentro del Sprint, el equipo cuenta con el *Burn Down Chart* que es un gráfico en el cual el equipo puede

realizar un seguimiento en tiempo real de la funcionalidad finalizada, faltante y días que restan para el fin del Sprint. Esto da una visión de avance y velocidad del equipo elevando el nivel de certeza de todos sus miembros.

## **CAPITULO V: AUTONOMÍA**

La autonomía es similar a la certeza y las dos están interrelacionadas. Esta, según David Rock, se manifiesta por medio de una sensación de falta de control por la incapacidad de influenciar los resultados en una determinada situación. (Rock, 2009, pág. 123) .Una nueva sensación de no poder determinar el futuro surge en la persona, y esto se manifiesta como incertidumbre.

Sin embargo puede haber situaciones en donde las dos, certeza y autonomía, se manifiestan por separado. Uno puede estar estresado por la falta de certidumbre pero al mismo tiempo tener mucha autonomía. Esto es típico de los profesionales que trabajan de manera independiente, en donde ellos tienen el control sobre sus propias acciones pero muy poco por sobre sus clientes y proveedores, creando así un ambiente de trabajo de alta incertidumbre. Se podría mencionar el caso contrario, del trabajador en dependencia que realiza una tarea operativa. Este tiene la certeza de pocos cambios en el trabajo, la tranquilidad de un sueldo a fin de mes, pero debe enfrentarse a situaciones de baja autonomía, por ejemplo es constantemente controlado por procesos y/o jefes.

La sensación de autonomía puede ser un importante impulsor de recompensa o amenaza. Según Steve Maier (Maier, 1982) el nivel de control que los organismos pueden ejercer sobre algo que les crea stress determina si el estresor altera el comportamiento del individuo. Esto es: solo los agentes de stress que no pueden ser controlados causan un efecto negativo sobre el organismo.

Otra forma de ver este concepto de autonomía es la capacidad de tener opciones. Una situación a priori puede parecer estresante, pero si la persona

logra hacer un análisis de la misma y observa que tiene varios cursos de acción se reduce los niveles de stress y siente que la misma es más manejable.

Estudios en adolescentes muestran que el típico comportamiento “rebelde” presente en occidente no es algo biológico sino una reacción ante la falta de opciones en su vida diaria. (Epstein, 2007).

Dentro de las organizaciones tener opciones está directamente ligado al tema de la responsabilidad, ya que al elegir las personas tienden a comprometerse con esa decisión.

En los valores de las metodologías ágiles relacionados con la Autonomía podemos encontrar *“Individuos e interacciones por sobre procesos y herramientas.”* Lo que se ve reflejado por los siguientes principios: 1- *“Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.”* Y 2- *“Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados”*. Como podemos apreciar las metodologías ágiles tienen una fuerte convicción en torno a la autonomía de los individuos, sino también al equipo que de forma expresa asume los compromisos de manera colectiva y no a través de un líder o representante.

Esto queda claro cuando el equipo se reúne en el *Sprint Plan Meeting* para analizar las funcionalidades dentro del *Product Backlog*. El equipo descompone cada una de estas funcionalidades en tareas con su correspondiente estimación en horas y un miembro del equipo se responsabiliza en cumplirla. De esta forma pasan a ser parte del *Sprint Backlog*. Este procedimiento se da con cada item del *Product Backlog* hasta que se completa la cantidad de funcionalidad que el equipo puede construir en la próxima iteración.

De esta forma cada miembro del equipo adquiere gran autonomía en primer lugar seleccionando y estimando (en conjunto a sus compañeros) las tareas a realizar. Y por otro lado asumiendo frente al equipo el compromiso de llevarlas a cabo en el tiempo estipulado.

Otro principio indica que 3- “*Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.*” Esto se da a partir de que el *Product Owner* es un integrante del equipo, posibilitando una comunicación clara y directa con el negocio. Dentro de las responsabilidades del *Product Owner*, se encuentra la creación, revisión y priorización del Product BackLog durante todo el proyecto, pero también la de estar disponible para los miembros de manera que estos puedan realizar las consultas que sean necesarias durante el Sprint. Esto clarifica las metas, y tener metas claras incrementa la autonomía al posibilitar que el equipo pueda decidir en cómo alcanzar dicho objetivo.

En resumen, las metodologías ágiles proveen mayor autonomía a los miembros del equipo. En primer lugar logrando la auto asignación de las tareas y asumiendo una responsabilidad ante las mismas y en segundo lugar permitiendo un contacto directo con el responsable del negocio. Lo que da como resultado una relación de confianza, buena predisposición y mayor productividad en el trabajo en equipo.

## **CAPITULO VI: RELACION**

La sensación al establecer relaciones en un ámbito social, según David Rock, es de recompensa, y la ausencia de la misma es interpretada como una amenaza para el cerebro. Las relaciones se establecen cuando una persona siente que pertenece a un grupo con afinidad o es parte de un equipo.

Es muy importante la comunicación cara a cara para crear este tipo de vínculos, ya que el cerebro genera empatía con las personas a partir de poder “copiar” e interpretar internamente las emociones y estados de ánimo de terceros. La forma en que el cerebro realiza esto es por medio de las neuronas espejo. Descubiertas por Giacomo Rizzolatti de la universidad de Padua (Iacoboni, 2005). Éstas han abierto todo un nuevo espectro de posibilidades para entender como las personas se conectan unas con otras. Este tipo de neuronas que forman parte del córtex pre-motor, tienen la característica de activarse cuando se realiza una acción como puede ser patear una pelota de



fútbol o realizar un gesto de desaprobación con los brazos. Se las denominan neuronas espejo debido a que estas se activan de manera similar al observar la misma acción. Lo verdaderamente sorprendente es que en los seres humanos estas neuronas se activan con mayor intensidad cuando la acción observada está rodeada de un contexto y una intención. Es decir, no es lo mismo patear una pelota de fútbol en el vacío que en medio de un partido dando un pase gol. Estos procesos son parte de la generación de empatía en las personas permitiendo reconocer emociones, intenciones y pensamientos en otros.

No solo importa la cantidad de conexiones sociales que generan las personas al establecer relaciones, sino la calidad de las mismas, este es uno de los principales factores que produce la sensación de felicidad en el ser humano. El cerebro se enriquece en un ambiente de conexiones sociales donde las relaciones son estables y positivas. Como en el caso de un núcleo familiar o un grupo de trabajo unido en torno a un objetivo común.

La felicidad que produce este tipo de relaciones libera dopamina y oxitocina lo que contribuye también en una mayor productividad en el ámbito laboral y a extender y mejorar la calidad de vida (Rock, 2009, pág. 163). Por otro lado la soledad puede incrementar el riesgo de infarto y enfermedades del corazón. Ésta produce una sensación de amenaza en el cerebro de la misma manera que la sed, el hambre o el miedo. Es decir que establecer relaciones es una necesidad básica como puede ser beber, comer o dormir. Si se evita esta amenaza al cerebro, este tiene mayor capacidad para pensar, planificar y crear.

Si observamos los valores de las metodologías ágiles encontramos: 1- *“Individuos e interacciones por sobre procesos y herramientas”* y 2- *“Colaboración con el Cliente por sobre Relación Contractual”*. Éstos valores se ven reflejados en los siguientes principios: 1- *“El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara”*. Como podemos apreciar uno de los pilares de éstas metodologías es la comunicación directa, que no solo la hace mucho más efectiva sino que propicia la creación de relaciones en las personas en el ámbito laboral.

Esto lo podemos ver en todas las reuniones del equipo, sobre todo en el *Daily stand up meeting*, donde cada integrante del equipo participa todos los días dando un mini reporte de las tareas que realizó en la última jornada, que tareas tiene para esta jornada y que impedimentos tiene para cumplirlas. Vale la pena aclarar que el reporte se realiza al equipo y no al Scrum Master. 2 - “Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.”

Esto adquiere un carácter esencial en las metodologías ágiles sobre todo teniendo en cuenta que para “adoptar el cambio” el equipo (los desarrolladores) deben incorporar al *Product Owner* y éste a su vez debe incorporar al equipo de tal forma que se construyan relaciones de confianza y empatía donde las problemáticas e intereses del Negocio y producción tiendan a converger en una sola visión. Esto se lleva a la práctica por medio de la participación del *Product Owner* (o un representante del negocio) en un mismo espacio físico, estando junto al equipo durante el transcurso del sprint y participando en forma activa en la planificación del mismo (*Sprint Planning Meeting*). En muchas organizaciones se forman pares de trabajo o programas de mentores por medio de los cuales se incrementan las relaciones al interior de la compañía. Esta práctica es implementada en XP mediante el *Pair Programming*. En donde dos programadores trabajan sobre una misma máquina. Usualmente los pares suelen estar formados por un programador experimentado junto a un junior.

Otra práctica muy difundida es el de establecer espacios comunes de dispersión que fomentan las conexiones a nivel personal entre los empleados. Esto también es fomentado por XP y la mayor parte de las metodologías ágiles donde el espacio de trabajo está acompañado por zonas de esparcimiento en donde los empleados no solo pueden conversar con una taza de café de por medio, sino jugar al Ping-Pong, Metegol o la última versión del FIFA.

Todas estas actividades por un lado refuerzan los lazos y por el otro bajan el nivel de competencia por el status, que es un elemento que puede entorpecer las relaciones dentro y entre los diferentes equipos de la compañía.

## CAPITULO VII: JUSTICIA

La Justicia, “fairness” según David Rock, es una necesidad primaria para el cerebro, la misma es definida como la presencia de intercambios equitativos en una determinada situación del ámbito social . Una sensación de justicia puede provocar una respuesta de recompensa en el cerebro, y una sensación de injusticia puede generar una percepción de amenaza persistente por varios días.

La justicia está presente en todos los ámbitos de la vida. Esto se ve claro en situaciones cotidianas, nos preocupa que el intercambio con otros sea justo. Por ejemplo, cuando en el almacén de la esquina pagamos más por un producto que en el supermercado, por más que la diferencia sea pequeña, sentimos que el intercambio es injusto y muchas veces no estamos dispuestos a pagar el precio.

En torno a este tema Golnaz Tabibnia realizó un estudio, a partir de un Juego llamado Ultimatum. (Tabibnia G. A., 2008) En él, dos personas tienen una cantidad de dinero a repartir, supongamos \$10, y una de ellas tiene la capacidad de seleccionar las cantidades que le corresponden a cada uno y la otra la capacidad de aceptar o no el trato. En caso que este no sea aceptado ninguna de las dos partes recibe dinero alguno. Los resultados nos muestran que cuando el sujeto A (repartidor) reparte \$5 y \$5, en un 97% de los casos la oferta es aceptada por el sujeto B (aceptador). Esto es así ya que repartir al %50-%50 parece lo más justo. En cambio cuando la oferta es de 20%-80% o menor, sólo un 31% de los casos el sujeto B acepta. Siendo que desde el punto de vista de un sujeto económico-racional el sujeto B debería aceptar toda oferta ya que cualquier suma que reciba al aceptar es mayor a nada, que es lo que recibe al rechazar la oferta.

Siguiendo con el mencionado estudio Tabibnia nos explica que cuando se acepta una oferta injusta, la respuesta en la ínsula es regulada por la corteza prefrontal ventrolateral (VLPFC). Es decir a mayor actividad del VLPFC, mayor probabilidad tiene una persona de aceptar una oferta injusta. De esta forma las

personas que tienden a regular sus emociones son las que pueden aceptar con más frecuencias ofertas injustas pero favorables desde el punto de vista económico-racional.

Lo interesante de esto es que la ínsula es la parte del cerebro que reacciona cuando sentimos repugnancia provocada por un mal sabor. Esto es el disgusto por un alimento o por una situación social son procesados en el mismo lugar del cerebro. Más aún las recompensas sociales y gustativas también son procesadas utilizando la misma región que en este caso es el Striatum Ventral.

En relación a que el sabor y la justicia son procesados por las mismas regiones del cerebro, el neurocientífico Stephen Pinker (Pinker, 1997) tiene la teoría que la respuesta a la justicia emergió como un subproducto del comercio eficiente, en los inicios de la humanidad cuando no existía el dinero ni alguna manera para preservar la comida, la mejor forma de acumular recursos era el de dar favores a otros. La deuda en formato de favor era acumulada en la mente del deudor.

Estos intercambios eran de especial utilidad en las sociedades de cazadores-recolectores, donde una presa era mucha comida para una sola familia. Para poder sobrevivir en este sistema las personas tenían que ser buenas detectando a los que se negaban a devolver favores, lo que pudo haber resultado en que estos individuos tuvieran una ventaja evolutiva.

La injusticia genera una fuerte reacción en el sistema límbico, y tiene la característica de generalizarse, es decir si uno siente que una persona está siendo injusta, se despierta una sensación de sospecha sobre las demás personas hasta que se disperse esa sensación. Así mismo las personas son más susceptibles a la injusticia cuando están cansadas o cuando el sistema límbico está sobrecargado, esto puede generar una reacción sobredimensionada ante hechos menores en el ámbito laboral, por ejemplo cuando se traen problemas del hogar o cuando se está mal dormido.

Por otro lado la justicia actúa aportando una alta dosis de recompensa en el cerebro por medio de la liberación de dopamina, de manera similar que lo haría un aumento de sueldo o una buena comida.

Esto tiene fuertes implicancias dentro de las organizaciones, en donde generar un ambiente de “justicia” por medio del establecimiento de normas claras, estables y equitativas para todos aporta de manera positiva al estado de ánimo general y a la productividad.

En el contexto de las metodologías ágiles está claro que el valor “*Individuos e interacciones por sobre procesos y herramientas*” es de suma importancia con respecto a promover la sensación de justicia ya que permiten que los miembros del equipo expresen su disconformidad con respecto a la forma y/o el trabajo que realizan, y que el mismo como un todo tenga la capacidad de actuar al respecto.

Podemos ver los valores expresados en los principios 1- “*El software funcionando es la medida principal de progreso.*” y 2 –“*A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia*”. Estos principios se implementan en Scrum mediante el *Sprint planning* que es, como se mencionó anteriormente, una reunión de planificación donde el equipo analiza en conjunto los requerimientos a ser entregados y cada integrante del equipo escoge cuales realizar entregando una estimación de tiempo para su cumplimiento.

Esto genera una fuerte sensación de justicia en el equipo ya que las tareas y los tiempos no son “dados”, sino que son asumidos mediante un compromiso a ser realizado con el equipo. Con respecto a la mejora y cambios en los procesos siguiendo dentro de Scrum podemos encontrar el *Sprint retrospective* el cual es una práctica que consta de una reunión al finalizar el sprint cuya finalidad es el de analizar el resultado del sprint y proponer mejoras al proceso para que los miembros del equipo puedan trabajar de manera más efectiva. Otra vez esta práctica aumenta la sensación de justicia en el proyecto ya que cada miembro del equipo no solo tiene la capacidad de seleccionar el trabajo a ser realizado si no de mejorar los procesos y actividades que se realizan para alcanzar la completitud de los requerimientos.

Por último quisiera resaltar el valor: “*Respuesta al cambio por sobre seguir un plan*”. En un principio se podría suponer que la aceptación del cambio de manera recurrente tendería a incrementar la sensación de intercambios injustos ya que el equipo trabaja en forma constante en lograr que los requerimientos del cliente se transformen en funcionalidad y la presencia recurrente de pedidos de cambio puede promover una sensación de falta de objetivos claros, disconformidad y menosprecio por el trabajo realizado.

Las metodologías ágiles invierten esta ecuación muy común en los proyectos de software. Teniendo por un lado como principal punto de medida de avance del proyecto el software funcionando y por otro lado aportando como diferencial la integración de los cambios en la propia filosofía de trabajo. De esta forma los cambios son tomados por los miembros del equipo como parte fundamental del servicio que prestan evitándose una sensación de abuso e injusticia ante la aparición de cambios que exceden al plan inicial.

## **CUERPO EMPÍRICO**

### **MARCO EPISTEMOLÓGICO**

Como marco epistemológico del trabajo de campo de la presente investigación se ha optado por un enfoque de “Estudio de caso.” Este enfoque a mí entender cuenta con varias características que lo hace ser el indicado. En primer lugar sigue una vía metodológica común al estudio etnográfico, como lo describe Anthony Giddens (Giddens, 2007), por medio del cual se realiza un estudio directo sobre un grupo de personas realizando una actividad específica en un entorno determinado ya sea por medio de observación o entrevistas.

En segundo lugar el estudio de caso, puede ser utilizado para crear hipótesis y entablar relaciones entre ellas alcanzando niveles explicativos de los hechos sociales en un contexto concreto. Lo que nos daría la posibilidad de profundizar el siguiente trabajo pasando de un estudio descriptivo hacia uno explicativo.

### **DISEÑO DE INVESTIGACIÓN**

A partir del marco epistemológico planteado en el punto anterior se llevó a cabo un estudio de campo por medio de la selección de 3 casos que a priori representan 3 puntos en la distancia que existe en la adopción de las metodologías ágiles para la gestión de desarrollo de software (Adopción total, parcial, y nula).

El objetivo de la investigación como se plantea en la introducción es el indagar comparativamente en qué grado los proyectos estudiados se acercan al modelo teórico planteado para luego establecer el desempeño de los mismos. De esta forma, se desea saber, si en la medida que los proyectos de software se aproximen a las recomendaciones del modelo SCARF por medio de la implantación de metodologías ágiles tenderán a mejorar la productividad, adaptación al cambio y trabajo en equipo al interior de dichos proyectos.

Cada uno de dichos casos fue explorado a través de una entrevista semi-estructurada a un representante del proyecto. Dichas entrevistas fueron divididas en 6 secciones, En las primeras 5 secciones se indaga en los 5 conceptos (Status, Certeza, Autonomía, Relaciones, Justicia) de la experiencia

social presentadas por David Rock como fue descrito en el marco teórico, relacionando cada uno de éstos con indicadores sensibles a las prácticas de las metodologías ágiles.

En la última sección se indagó sobre la productividad, adaptación al cambio y trabajo en equipo en cada uno de los proyectos analizados. Para esto se utilizó una Escala de Osgood, en donde los entrevistados evaluaron el desempeño de cada uno de los proyectos.

Las 6 secciones son:

## **1 - STATUS**

Como se planteó en el marco teórico, el status es uno de los principales conductores del comportamiento tanto en el trabajo como fuera de él. Una percepción de que el status está subiendo, aunque sea pequeña, activa los circuitos de recompensa del cerebro. En cambio una percepción de que el status está bajando activa los circuitos de amenaza.

Dentro de las metodologías ágiles los principios que prevalecen en torno al Status son: La creación de un entorno con individuos motivados, apoyo de parte del entorno del proyecto y confianza plena a los miembros del equipo para la realización de las tareas de manera autónoma.

Como indicadores de estos principios se seleccionó: **a) Roles/ Jerarquía dentro del proyecto, siendo 1 una estructura jerárquica del tipo piramidal y 5 una estructura de equipo horizontal donde no existe la figura del “Jefe/Líder”.**

**b) Instancias de interacciones entre los miembros del equipo siendo 1 que los miembros del equipo trabajen en forma de silos sin interacciones en lo referente a lo laboral y 5 que se privilegie la comunicación cara a cara y todo el equipo de proyecto se encuentre en un mismo lugar físico.**



## 2 - CERTEZA

Como se mencionó en el marco teórico, el cerebro ansía tener certeza. La sensación de falta de certeza acompañado por la pérdida de control puede generar fuertes respuestas por parte del sistema límbico. Para evitar esto el cerebro actúa como si fuera un dispositivo diseñado para predecir, recursos neuronales son destinados en forma masiva a predecir que va suceder a cada instante.

La incertidumbre se manifiesta en el cerebro como una amenaza a la propia vida. Cuando no es posible prevenir el desenlace de una situación de forma inmediata la amígdala dispara una alerta que aumenta el nivel de atención.

Aunque a priori pareciera paradójico la certeza es importante para la adaptación al cambio. El cambio es susceptible de ser planificado y ser tomado como parte del proceso mismo.

Se establecieron los siguientes parámetros como indicadores de certeza en los proyectos analizados: **a) Definición de requerimientos y funcionalidades siendo 1 que no se definen ningún tipo de requerimiento y 5 que los requerimientos son definidos en forma conjunta con el Product Owner. b) Planificación de las tareas a realizar donde 1 indica que no existe un detalle de las tareas a realizar y 5 que las tareas son definidas por el equipo y con el compromiso de ser cumplidas en un período determinado de tiempo. c) Gestión de los cambios en la planificación, 1 representa la ausencia de pedidos de cambios, 5 que los cambios son gestionados como parte de la metodología del proyecto y aceptados de manera constante.**

## 3 - AUTONOMÍA

Como vimos en el marco teórico la autonomía es similar a la certeza y las dos están interrelacionadas. Esta se manifiesta por medio de una sensación de falta de control por la incapacidad de influenciar los resultados en una determinada situación. Una nueva sensación de no poder determinar el futuro

surge, esto se manifiesta como incertidumbre. Las metodologías ágiles no solo tienen una fuerte convicción en torno a la autonomía de los individuos sino también al equipo que de forma expresa asume los compromisos de manera individual y colectiva, y no a través de un líder o representante.

Como indicadores se seleccionaron: **a) asignación de tareas siendo 1 la inexistencia de tareas formalmente definidas y 5 la auto asignación de tareas por parte del equipo.**

**b) Reporte de Status, siendo 1 que las tareas no tienen una instancia formal para ser reportadas y 5 donde existe dicha instancia donde se reporta de manera periódica y es realizada por el miembro del equipo a cargo de la misma.**

#### **4 – RELACIÓN**

Siguiendo con lo observado en el cuerpo teórico, la sensación al establecer relaciones, en un ámbito social, es de recompensa y la ausencia de la misma es interpretada como una amenaza para el cerebro. Las relaciones se establecen cuando una persona siente que pertenece a un grupo con afinidad o es parte de un equipo. El cerebro se enriquece en un ambiente de conexiones sociales de calidad con relaciones estables y positivas.

La felicidad que produce este tipo de relaciones libera dopamina y oxitocina lo que contribuye también en una mayor productividad en el ámbito laboral y a extender y mejorar la calidad de vida.

Dentro de las metodologías ágiles podemos encontrar los siguientes principios: “El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara” y “Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.”

Como indicadores se seleccionaron: **a) Comunicaciones dentro del equipo, siendo 1 que no existan las comunicaciones entre los miembros del equipo que están a un mismo nivel y 5 que las comunicaciones son en**

todos los sentidos y se establecen preferentemente cara a cara. b) **Relación con el Product Owner en donde 1 indica que no existe relación alguna entre los miembros del equipo y el Product Owner y 5 que el PO es parte del equipo.**

## **5 - JUSTICIA**

La justicia actúa aportando una alta dosis de recompensa en el cerebro por medio de la liberación de dopamina, de manera similar que lo haría un aumento de sueldo o una buena comida.

Esto tiene fuertes implicancias dentro de las organizaciones, en donde generar un ambiente de “Justicia” por medio del establecimiento de normas claras, estables y equitativas para todos aporta de manera positiva al estado de ánimo general y a la productividad.

Siguiendo con el marco teórico cuando se habla del hábito de la Justicia en relación a las metodologías ágiles podríamos mencionar los principios “Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida” y “A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.”

Para medir el nivel de aplicación de dichos principios se indagó en: **a) Las estimaciones de esfuerzo de los requerimientos siendo 1 que los tiempos están predefinidos sin participación del miembro del equipo que ejecuta la tarea y b) La mejora continua en donde el 1 representa que no existe una revisión de los mismos y que los cambios provienen de una fuente externa. Y 5 que el equipo tiene la capacidad de mejorar el proceso y que se revisa en forma periódica e intervienen todos los miembros.**

## **6 – Autoevaluación del Proyecto**

Para finalizar se les preguntó a cada uno de los entrevistados su evaluación del proyecto en términos de: **a) Colaboración, b) adaptación al cambio y c) la Productividad. Siendo 1 Muy Malo y 5 Muy bueno.**

## ANÁLISIS DE LAS ENTREVISTAS

A partir del diseño detallado en la sección anterior se llevaron a cabo tres entrevistas Antonio Villarroel (AV), Diego Narvaez (DN) y Fernando Scasserra (FS), ver Anexo I: ENTREVISTAS. Cada uno de ellos relató sus experiencias como miembros de un equipo de desarrollo de software en diferentes empresas, contextos y roles.

Antonio es un Ingeniero en Sistemas de Información (UTN) con amplia experiencia trabajando para Baufest. Dentro de la empresa se especializó como consultor en la implementación de procesos y herramientas que soportan el desarrollo de software. Trabaja con empresas nacionales y multinacionales de primera línea como parte del equipo de proyecto. La mayor parte de estos proyectos utilizan metodologías PMI (Project Management Institute), que se caracterizan por ser metodologías Top-Down. Por otro lado reconoce que existe un movimiento muy importante en torno de la implementación de Metodologías Ágiles en todo tipo de proyectos, y el conflicto que surge cuando se implementan en empresas grandes acostumbradas a gestionar sus procesos de manera Top-Down. La entrevista de Antonio gira en torno a un caso exitoso en el que participó, el cual se utilizó Scrum como metodología de desarrollo de software para una importante empresa multinacional.

Diego Narvárez es un analista y desarrollador de software con mucha experiencia en proyectos de E-learning principalmente desarrollando cursos e implementando plataformas Web que los soporten. Está acostumbrado a trabajar en equipos multidisciplinarios con diseñadores web y diseñadores didácticos coordinando la creación de los cursos. Diego nos cuenta en la entrevista sobre su última experiencia laboral, que fue bastante diferente a lo que estaba habituado, en donde participó de un equipo de desarrollo el cual tenía a su cargo el mantenimiento de varias aplicaciones, entre ellas una de gestión documental en la cual era el encargado del soporte de la misma. No utilizaban ninguna metodología específica para la gestión de los proyectos y el manejo de los mismos era a través de una estructura Top-Down.

Por último Fernando Scasserra que es Ingeniero en Sistemas de Información (UTN) y trabaja hace 10 años en Mercado Libre. Actualmente ocupa el puesto de Senior Manager del Área de Desarrollo. Desde el año pasado también es Manager del área Quality Assurance (Calidad de Producto) dentro de la empresa. Su principal tarea es gestionar el desarrollo de la API (del inglés: Application Programming Interface), que son un conjunto de sistemas que componen el núcleo de Mercado Libre. Para esta tarea existen 5 equipos diferentes y cada uno de ellos tiene un líder de Proyecto. Con respecto a la metodología que utilizan Fernando nos cuenta que por experiencia en Mercado Libre se aprendió que las metodologías ágiles son las mejores para gestionar los desarrollos dentro de la empresa. Cada equipo tiene la libertad de adaptar la metodología como más le resulte conveniente, mientras que sea “Ágil”. Pero por otro lado admite que existe cierta estructura piramidal a partir de la figura del Líder de Proyecto.

A partir de estas tres entrevistas se realizó un análisis comparativo en donde se utilizaron los indicadores descritos en la sección DISEÑO DE INVESTIGACIÓN y se les otorgó un puntaje de acuerdo a la escala definida en dicha sección. A su vez se interpretó en cómo influyen los mecanismos del modelo SCARF en los comportamientos analizados.

Cada análisis es acompañado de una tabla en donde en las columnas figuran los entrevistados: Antonio Villarroel (AV), Diego Narvaez (DN) y Fernando Scasserra (FS). En las filas figuran los indicadores que representan las prácticas de las metodologías ágiles.

## 1 - STATUS

Tabla 1  
*Indicadores de grado de implementación de metodologías ágiles en torno al Status en los proyectos analizados.*

Entrevistas	AV	DN	FS
<b>STATUS</b>			
Roles/jerarquías dentro del proyecto.	5	1	3
Instancias de interacción entre los miembros del equipo.	5	1	4

Cuando se indagó sobre los Roles/Jerarquías, Antonio nos cuenta que los roles del proyecto constaban de un Team, un Product Owner y un Scrum Máster. Estos roles son los que se definen dentro de Scrum, y si bien existen diferentes seniority y responsabilidades, obtiene un puntaje de 5 ya que la estructura es horizontal y la mayor parte de las decisiones son tomadas por consenso en el equipo. Generando, como señala Rock, una recompensa sobre el Status de los miembros con menos seniorities elevando los niveles de dopamina y serotonina y aumentando la productividad en ellos. En el otro extremo Diego con un puntaje de 1, nos cuenta que en su proyecto existía una estructura muy piramidal con un “Jefe” que tomaba las decisiones y subordinados que las ejecutaban de manera individual, cuando diego nos menciona esto nos imaginamos un incremento en los niveles de cortisol y en consecuencia del stress en todo el equipo. Fernando nos cuenta que si bien todos hacen de todo dentro de los proyectos de su gerencia, asumiendo las responsabilidades en cada caso. Existe una estructura piramidal con un líder de proyecto que asume las responsabilidades del equipo. En este caso nos encontramos en un punto medio en donde por un lado existen estímulos positivos, pero la presencia de un líder fuerte como responsable de última instancia puede provocar la falta de compromisos por parte de los miembros del equipo por este motivo se le otorga un 3 de puntaje en la escala.

Cuando se les pregunta a los entrevistados por las instancias de interacción entre los miembros del equipo. De nuevo la respuesta de Antonio es de un 5 ya que afirma que se manejaban “Como en los libros [de metodologías ágiles]”, donde se realizaban reuniones diarias, reuniones de planificación cada

2 semanas y mensualmente reuniones de mejoras del proceso. Antonio remarca también la importancia del Kick Off Meeting, donde participaron todos los integrantes del proyecto, desde el programador más raso hasta los principales sponsors. Todas estas reuniones y puntos de contacto reducen los niveles de amenaza en torno al Status tanto para los más experimentados como los más juniors del proyecto, posibilitando un incremento en la cantidad de conexiones que realiza el cerebro. Por el otro lado Diego declara que no había interacción directa en los miembros del equipo, es decir toda interacción era mediada por el jefe, lo que resulta en un puntaje de 1 para este indicador. Al imposibilitar esta interacción y no haber un conocimiento personal y profesional se tiende a ver a los demás como una posible amenaza y el cerebro tiende a proteger el Status desperdiciando recursos neuronales que podrían estar dedicados a resolver los desafíos inertes al desarrollo de Software. En la gerencia de Fernando tienen reuniones semanales entre los miembros del equipo y entre todos los equipos en donde evalúan el gap que cada proyecto tiene con lo establecido para el Quarter. Todas estas reuniones generan un alto grado de comunicación tanto al interior del proyecto como con los otros proyectos y los Sponsors. En este caso al igual que con Antonio las reuniones estimulan el Status de los Miembros. El puntaje obtenido por Fernando es de un 4 debido a que si bien existen reuniones entre los miembros del equipo el Sponsor no se encuentra en el mismo lugar físico que el equipo de desarrollo.

Como refleja el puntaje asignado para cada caso en la medida que las prácticas se acercan a las metodologías ágiles el nivel de status y por consiguiente sus beneficios se incrementan en los miembros del equipo.

## 2 - CERTEZA

Tabla 2			
<i>Indicadores de grado de implementación de metodologías ágiles en torno a la certeza en los proyectos analizados.</i>			
Entrevistas	AV	DN	FS
<b>CERTEZA</b>			
Definición de requerimientos y funcionalidades	5	2	5
Planificación de las tareas a realizar	5	2	5
Gestión de los cambios en la planificación	5	2	3

Cuando se les pregunta a los entrevistados sobre la definición de los requerimientos y funcionalidades a desarrollar Antonio nos cuenta que si bien los requerimientos del proyectos se encontraban definidos a grandes rasgos, se había generado un Product Backlog que incluía un detalle de cada uno de los requerimientos y una priorización de los mismos, de esta forma se logró un consenso hacia donde se iba. Luego se reparten los Items entre los diferentes equipos para lograr un desarrollo en paralelo de los mismos. Como podemos ver en el caso de Antonio existe una definición clara de los requerimientos en varias etapas y con la participación de los miembros del equipo (Product Backlog), lo que da Certeza sobre los objetivos del proyecto, por este motivo se le otorga un puntaje de 5. Por otro lado Diego nos relata una situación donde el Jefe establecía los requerimientos al mínimo detalle y así mismo estos requerimientos carecían de alguna herramienta para su análisis o seguimiento por los miembros del equipo. Si bien en el caso de Diego existe una planificación Top-Down por parte del líder del proyecto, esto puede generar incertidumbre por falta de una visión global y de contexto, obteniendo un puntaje de 2 para este indicador. En cambio Fernando realiza junto a los Sponsor y a cada uno de los equipos una definición de los requerimientos por “Quaters”. Por la características de los proyectos que gestiona Fernando la mayor parte de los requerimientos llegan una vez ya comenzado el Quater estos se suman al Backlog y se priorizan. Si bien en este caso la definición que nos da Fernando es la que mejor refleja el espíritu de las metodologías ágiles en torno a la definición de los requerimientos aportando certeza al proyecto, como veremos más adelante otros factores influyen para que esto no pueda cumplirse plenamente, pero en este caso se le otorga un puntaje de 5.

Luego de la definición de los requerimientos, se les preguntó a los entrevistados sobre la planificación de las tareas a realizar. En el caso de Antonio se realizaban Sprint cada 2 semanas, en donde se volvía a revisar el Backlog junto al Product Owner, el equipo descomponía en tareas los requerimientos prioritarios y se comprometía a tener listo en 2 semanas un conjunto de los mismos, obteniendo de nuevo un puntaje de 5 en la escala al comportarse “como en los libros”. En el proyecto de Diego como se afirmó en el análisis del indicador anterior el Jefe daba instrucciones de qué se debía



hacer sin dar mucho contexto sobre las mismas, lo que resulta en un 2. En el caso de Fernando al principio de cada Iteración, que son semanales, se escogen que requerimientos atacar y como realizarlo dentro de cada equipo. Por este motivo se le otorga un puntaje de 5.

Al igual que en el ítem anterior podemos ver reflejado que a mayor participación del equipo, genera una mayor claridad y certeza sobre el futuro. Evitando “Sorpresas” en los tiempos y/o tareas a realizar que disparen las alertas del Sistema Límbico.

Para finalizar, con respecto a la gestión de los cambios, Antonio nos comenta que en su proyecto los cambios eran constantes, había casos en que el Product Owner quería implementar cambios durante el sprint, entonces se negociaba la salida de otros requerimientos para que se pueda cumplir con la funcionalidad comprometida en la Iteración tomando como parte de la metodología de trabajo a los cambios de requerimiento, por esto obtiene un 5. Diego por otro lado, con un puntaje de 2 casi no tenía pedido de cambios sobre las nuevas funcionalidades, una vez planteadas se hacían y que uno andaba más preocupado por las incidencias del día a día. Fernando también nos cuenta que en su proyecto los pedidos de cambio eran constantes al no tener todos los requerimientos definidos al principio del Quater, muchas veces esto lo obligaba a tener que priorizar entre los diferentes sponsors y frenar cambios que ya estaban en curso. Si bien los pedidos de cambios son tomados por el equipo de Fernando, no se contemplan como parte de la metodología. Generando disrupciones en el flujo de trabajo del proyecto que puede afectar a los diferentes sponsors y hasta los miembros del equipo. Por este motivo se le otorga un puntaje de 3 para este indicador.

Con respecto a la gestión de los cambios podemos observar que el manejo de Antonio al mantenerse firme en las tareas definidas para el Sprint y negociar pequeños cambios en forma de Trade-off es el que mayor aporta a generar certidumbre al equipo. En contraste Diego nos señala que no se consideraban “cambios” en el sentido formal de las metodologías ágiles o con Fernando, que debido a la propia lógica del negocio a veces los cambios eran planteados y

luego dados marcha atrás en la misma iteración, lo que genera falta de certeza y stress en los miembros del equipo.

### 3 – AUTONOMÍA

Tabla 3  
*Indicadores de grado de implementación de metodologías ágiles en torno a la autonomía en los proyectos analizados.*

Entrevistas	AV	DN	FS
<b>AUTONOMÍA</b>			
Asignación de Tareas	5	2	2
Reporte de Status	5	1	3

En el caso de la Autonomía se indagó a los entrevistados sobre la Asignación de Tareas, Antonio nos comenta que cada miembro del equipo se autogestionaba y se comprometía a terminar las funcionalidades durante el transcurso del Sprint, lo que le otorga un puntaje de 5 para este indicador. Los miembros más juniors del equipo eran asistidos por el Scrum Master para dicha tarea. Como podemos observar existe confianza en los miembros del equipo otorgándole plena autonomía y ayuda para aquellos con menos experiencia, de esta forma se incrementan la confianza y el compromiso con las tareas a cumplir. En el caso de Diego nos menciona dos fuentes de asignación de tareas, el primero como se marcó en el punto anterior el jefe las asigna directamente cuando son nuevos desarrollos. En segundo lugar se le asignaba de forma automática por medio de la herramienta de tickets en los casos de las incidencias (errores críticos), si bien existía una tarea definida formalmente no había una autoasignación de tareas lo que le otorga un 2 para este indicador. A Fernando también se le asigna un 2 ya que nos cuenta que cuando llega un nuevo requerimiento es analizado por el líder del proyecto, este le otorga una prioridad y decide si entra para la próxima semana, próximo mes o Quarter. Estos requerimientos son asignados semanalmente por el Líder del proyecto a los miembros del equipo.

Tanto Diego como Fernando están fuertemente influenciados por las metodologías Top-Down, en lo que respecta a la asignación de tareas. Esto es,

que un líder/jefe tiene el conocimiento suficiente sobre las capacidades de los recursos a cargo para asignar las tareas. Esto por un lado genera falta de compromiso y Stress en los miembros del equipo al no tener el control sobre las tareas a realizar. Y por el otro tiende a limitar las capacidades de los mismos.

Con respecto al reporte de Status o avance de las tareas Antonio nos cuenta que habían dos fuentes donde se reportaban, por un lado por medio de una herramienta donde se reflejaba el Sprint Backlog y el Product Backlog, los integrantes del equipo actualizaban las tareas y esto era reflejado en la herramienta que podía ser observado ingresando al sitio del proyecto y en los monitores del piso. Por otro lado se realizaban las Daily Meeting donde cada integrante del equipo reportaba el avance en sus tareas. Como relata Antonio, no solo los miembros del equipo manejaban las herramientas de avance del proyecto sino que entregaban el reporte de status de forma diaria al resto del equipo reafirmando su autonomía y control sobre sus actos, y por ende elevando el nivel de compromisos con los mismos, resultando en un puntaje de 5 para este indicador. En cambio Diego reportaba al jefe el status de las tareas, que a cada momento le preguntaba por el avance de las mismas. A su vez el usuario se enteraba de la incidencia resuelta a través de la herramienta. En este caso se da lo que se conoce como “Micromanagement” lo que está en las antípodas de la autonomía, generando falta de compromiso y un constante stress por cumplir con expectativas ajenas, por estos motivos se le otorga un puntaje de 1. Por último en el proyecto de Fernando el reporte de status lo dan los desarrolladores al líder de proyecto una vez finalizada la tarea, y este se comunica con el Sponsor para las pruebas y el ok, final. En la reunión semanal de todos los equipos cada uno reporta al resto el avance de los requerimientos a su cargo. Con respecto a Fernando nos encontramos en un punto medio si bien no hay manejo de la herramienta o contacto con el usuario final, sí existe un reporte y compromiso asumido con el resto del equipo. Dando como resultado un 3 en nuestra escala para este indicador. En el caso de la autonomía podemos ver como a medida que las prácticas se acercan a las Metodologías Ágiles, se incrementa el nivel de autonomía de los miembros del equipo incrementando el compromiso y reduciendo el Stress.

## 4 - RELACIÓN

Tabla 4			
<i>Indicadores de grado de implementación de metodologías ágiles en torno a la relación en los proyectos analizados.</i>			
Entrevistas	AV	DN	FS
<b>RELACIÓN</b>			
Comunicaciones dentro del equipo	5	1	4
Relación con el Product Owner	5	3	3

Con respecto a las relaciones en el proyecto de Antonio, más allá de las reuniones típicas de las metodologías ágiles como el Stand Up Meeting, o el Sprint planning la empresa hizo una inversión importante en ubicar a todos los involucrados del proyecto en dos pisos contiguos facilitando la comunicación cara a cara y la familiaridad entre los miembros del proyecto. Resalta también la reunión de kick off inicial del proyecto con todos los miembros, donde se les realizó una introducción al mundo de las metodologías ágiles. Esto no solo cumplió el rol de una capacitación sino también en romper un poco el hielo y que los diferentes integrantes se reconocieran como miembros del equipo. Como bien lo marca Antonio la importancia de las relaciones es la clave para la fortaleza en los Proyectos que Utilizan Metodologías Ágiles, por estos motivos se le otorga un 5 en este indicador. Diego, con un puntaje de 1, no tenía comunicación con los otros miembros del grupo de trabajo ya que manejaban aplicaciones diferentes y si bien compartían un mismo jefe, área y espacio físico no existían comunicaciones a nivel laboral, este es un típico caso que no solo puede afectar la performance del equipo por falta de comunicación sino producir en términos individuales una sensación de aislamiento que afecte la performance. Como se mencionó anteriormente en el proyecto de Fernando tienen una reunión semanal dentro de cada equipo y una con todos los equipos donde se reporta el Status de los proyectos y se decide con qué seguir.

También se apoyan mucho en un chat (Telegram) del equipo el cual se reserva para comunicaciones importantes para todos los miembros. En el caso de Fernando si bien no todas las comunicaciones de importancia son cara a cara, se le otorga un puntaje de 4 al cumplir con los valores de las metodologías ágiles generando reuniones periódicas que permiten la integración en los miembros elevando la productividad a nivel individual y grupal.

Si hablamos de la relación con el Product Owner en el caso de Antonio esta se apoyaba en la propia metodología, donde el PO era parte integrante del equipo, por esto obtiene un puntaje de 5. Esta participación activa en el equipo provocó que a medida que el proyecto fue avanzando entendiera las problemáticas del equipo de desarrollo ajustando los pedidos y cambios a las posibilidades reales del equipo. Una vez más Antonio nos muestra "El tipo Ideal" de las relaciones con el PO, al formar parte del equipo se genera una empatía gracias a las neuronas espejo la cual sería casi imposible sin un contacto personal. En el caso de Diego las comunicaciones con el usuario eran por medio de la herramienta y en caso de ser necesario se programaban reuniones por temas particulares. Si bien no era parte del equipo existía una comunicación constante con el PO y contactos cara a cara con el mismo por eso obtiene un puntaje de 3. Fernando también obtiene un 3 en este ítem al tener una relación similar con el Sponsor donde los requerimientos se plantean por mail, y se cargan en una herramienta para el seguimiento. Las reuniones no son programadas pero suelen ocurrir en un primer momento con el líder de proyecto para un análisis inicial y luego se pueden programar otras en caso que hallan dudas. Tanto Fernando como Diego se apartan del modelo de las Metodologías Ágiles lo que dificulta el mecanismo de empatía como en el caso de Antonio, pero tienen la posibilidad de un contacto directo entre los miembros del equipo y el PO lo que facilita la claridad en los requerimientos.

Como mencioné al principio de este análisis las Metodologías Ágiles hacen de las relaciones entre los miembros del equipo un factor central para el éxito de los proyectos. En los tres casos analizados podemos observar que a medida que se implementan las prácticas de las Metodologías Ágiles se establecen mecanismos que, siguiendo a David Rock, incrementan la productividad y la empatía entre los miembros del equipo.

## 5 - JUSTICIA

Tabla 5  
*Indicadores de grado de implementación de metodologías ágiles en torno a la justicia en los proyectos analizados.*

Entrevistas	AV	DN	FS
<b>JUSTICIA</b>			
Estimaciones esfuerzo de los requerimientos	5	1	5
Mejora continua	3	1	3

Para finalizar con los 5 conceptos de la experiencia social dentro de los proyectos de software analizaremos la Justicia, que en el caso de Antonio cuando hacemos referencia a la estimación de los requerimientos, existía una primera estimación para dimensionar el tamaño del mismo y luego en cada Sprint Meeting se refinaba dicha estimación con el integrante del equipo encargado de realizarlo. Este mecanismo que tienen las Metodologías Ágiles crea un ambiente en el cual la estimación (es decir el tiempo en que deben realizarse) las tareas sea recibido por los participantes como algo justo aumentando el compromiso con el cumplimiento de las mismas, Antonio al seguir los principios de las Metodologías Ágiles Obtiene un 5 de puntaje. En el caso de Diego los tiempos de cumplimiento de las tareas estaban predefinidos. Para los incidentes que venían por medio de la herramienta existía un acuerdo de servicio el cual el equipo técnico se comprometía a resolverlo en un tiempo determinado según la criticidad del mismo. Para las nuevas funcionalidades el tiempo lo determinaba el jefe. Estos son mecanismos muy utilizados en las metodologías Top-Down, por este motivo se le otorga un 1, en donde la imposición de los tiempos sin consultar al equipo de desarrollo suele ser visto como algo arbitrario y por consiguiente injusto. En el caso de Fernando el procedimiento era muy similar al de Antonio, obtiene un 5 también, había una primera revisión con el Product Owner y luego se analizaba en detalle con el developer encargado de implementarlo.

Con respecto a la mejora de procesos de trabajo, Antonio nos comenta que en su proyecto no se producían muchas modificaciones, y que las reuniones de mejora apuntaban a mejorar las relaciones con las otras áreas que contribuían

pero no eran parte directa del proyecto, si bien no se trabajaba directamente sobre los procesos existía un espacio formal de mejora por este motivo se le otorga un 3. Diego nos cuenta que en un primer momento no había una metodología, la forma de trabajo era establecida por el jefe. En una segunda etapa se implementó una metodología que estaba orientada a la documentación y que podía no ser respetada si el “negocio” lo demandaba, el puntaje obtenido es un 1 al no existir ningún tipo de revisión sobre los procesos. En el caso de Fernando se realizan revisiones al proceso, pero solo participan de las mismas el manager y el líder de proyecto que en todo caso puede o no transmitir al manager los comentarios de su propio equipo, en este caso si existe la revisión del proceso y la mejora del mismo pero al no participar todos los miembros del equipo en dicha revisión el puntaje en este ítem es de un 3.

En este punto si bien Antonio y Fernando implementan en sus proyectos algún tipo de revisión sobre los procesos, ninguno de los tres implementa dichas reuniones de forma constante y con la participación de todo el equipo como recomiendan las Metodologías Ágiles. Esto puede derivar en los tres casos no solo una baja de productividad por no poder percibir ineficiencias en el proceso de desarrollo sino también a nivel individual en los miembros del equipo al sentir como una injusticia ver que su trabajo se desperdicia a causa de procesos ineficientes.

## **6 - Productividad, Adaptación al cambio y Trabajo en equipo.**

Como cierre de esta sección se analizará las respuestas dadas por los entrevistados en torno a la Productividad, Adaptación al cambio, Trabajo en equipo.

Tabla 6  
*Evaluación dada al desempeño de los proyectos en términos de Productividad, Adaptación al cambio y Trabajo en equipo por parte de los entrevistados.*

Entrevistas				AV	DN	FS
Dimensiones						
Productividad				3,5	1	3,5
Adaptación al cambio				5	2	3
Trabajo en equipo				5	1	5

Comenzaremos con la Productividad donde se les pidió que comparen el proyecto analizado con otras experiencias. En el caso de Antonio, le otorga un puntaje de 3,5 a esta dimensión ya que si bien las metodologías ágiles generan una alta cohesión y motivación entre los miembros del equipo, puede llegar a darse la situación de un freno por parte del Product Owner por falta de definición del negocio en torno de cómo y por donde seguir, paralizando el proyecto por completo. En este sentido las metodologías ágiles funcionan de manera muy similar a una línea de montaje (recordemos que la mayoría de las metodologías ágiles tienen una raíz fuerte en los procesos industriales) si se frena un punto en la cadena de montaje se frena la producción de nuevas unidades. Que en este caso son las funcionalidades del sistema. Es decir que para que funcionen estas metodologías se necesita de la participación constante de todos los miembros del equipo. Diego dice que la productividad era muy baja por falta de colaboración, los miembros del equipo realizaban tareas muy parecidas pero no había un intercambio ni de tareas ni experiencias. Fernando evalúa la productividad del equipo con un 3,5. Como lo explica él, que si bien se llega a cumplir los objetivos y los Sponsors están contentos, muchas veces sucede esto porque el Líder de proyecto se pone los objetivos al hombro y es el encargado de cumplir con las tareas pendientes. Es decir se cumple pero no se llega de la mejor forma.

Con respecto a la adaptación al cambio, Antonio le otorga un puntaje de 5, afirmando “era parte de la naturaleza del proyecto”. Sobre todo si se tiene en cuenta la metodología utilizada (Scrum). En el caso de Diego le otorga un puntaje de 2, comenta que las nuevas versiones de la aplicación las iban planchando. Es decir no eran actualizadas por temor al cambio. Y cuando surgió la posibilidad de implementar una nueva plataforma fue descartada. Vale



la pena aclarar que si bien estos cambios parecen meramente tecnológicos, un cambio de sistema en muchos casos requiere adaptación para incorporar nuevos conocimientos y nuevas formas de trabajar. Fernando piensa que en su caso la adaptación al cambio es un 3, debido a que al equipo le cuesta reaccionar de forma positiva a los cambios, cuando son descartadas las funcionalidades recién finalizadas por nuevas necesidades surgidas por el negocio. Esto produce en los desarrolladores la sensación de estar trabajando sin propósito alguno.

Cuando se les preguntó sobre el trabajo en equipo. Antonio le otorga un 5 ya que afirma que este factor fue fundamental para el éxito del proyecto. Diego desde el otro extremo afirma que no había colaboración entre los miembros del equipo ya que eran todos silos independientes. Tampoco existía una colaboración por parte del jefe, sino simplemente control. Fernando le otorga un 5 resaltando la “buena onda” y colaboración entre los miembros del equipo. Sobre todo la del líder de proyecto que funciona como un par más.

## **CONCLUSIONES**

En la primera Etapa del cuerpo teórico, Capítulo I y II, del presente trabajo, respondiendo al Objetivo N° 1, se pudo ver en detalle los Procesos, Roles y Prácticas de dos de las Metodologías Ágiles de desarrollo de software más utilizadas en la industria hoy en día. En ambos casos se puede observar como dichas metodologías se encuentran alineadas con los Valores y Principios del Manifiesto Ágil que fueron explicitados en la introducción del presente trabajo. Siguiendo con el Cuerpo teórico en los Capítulos III, IV, V, VI y VII se profundizó en detalle en cada uno de los 5 conceptos de la experiencia social de David Rock y cómo estos influyen en la productividad, adaptación al cambio y trabajo en equipo dentro de las empresas, como respuesta a lo planteado en el Objetivo N°2.

En respuesta al Objetivo de Investigación N°3, a partir de la descripción de cada uno de los conceptos del Modelo SCARF, se detalla el vínculo con las Metodologías Ágiles. Para luego determinar cuáles son las prácticas de las metodologías que mejor reflejan los comportamientos que generan, como señala David Rock, una recompensa para la mente. A partir de este análisis se elaboró *la tabla número 7 Cruce Modelo SCARF con Prácticas, Principios y valores de las Metodologías Ágiles*, que sintetizan el vínculo entre los dos.

Tabla 7

*Cruce Modelo SCARF con Prácticas, Principios y valores de las Metodologías Ágiles*

Practica Ágil	Metodología	Modelo SCARF				
		Status	Certeza	Autonomía	Relación	Justicia
Diseño simple	XP					
Pair programming	XP		X		X	
40 horas semanales (cXP)			X			
Espacio común	XP				X	
Daily stand up meeting	XP	X	X			
Story cards	XP					
Gráficos visibles	XP		X			
Lista de tareas	XP					
Refactoring	XP					
Sprint planning	Scrum	X	X	X	X	X
Daily Scrum	Scrum					
Sprint review	Scrum	X	X			
Sprint retrospective	Scrum					X
Team (Autogestionado)	Scrum			X		
Product Backlog	Scrum					
Sprint Backlog	Scrum					
Gráfico de burndown	Scrum		X			
Rol de Product Owner			X	X		
TimeBoxing			X			X
Zonas de espacimientocomún					X	
<b>Principios Ágiles</b>						
Nuestra mayor prioridad es satisfacer al cliente mediante el software funcionando			X			X
Aceptamos que los requisitos cambien, incluso en etapas tardías			X			
Entregamos software funcional frecuentemente, entre otros beneficios			X	X	X	
Los responsables de negocio y los desarrolladores trabajan juntos diariamente		X		X		
Los proyectos se desarrollan en torno a individuos motivados			X		X	
El método más eficiente y efectivo de comunicar información es el software funcionando			X		X	
El software funcionando es la medida principal de progreso			X			X
Los procesos Ágiles promueven el desarrollo sostenible			X			
La atención continua a la excelencia técnica y al buen diseño						
La simplicidad, o el arte de maximizar la cantidad de trabajo con la menor cantidad de esfuerzo						
Las mejores arquitecturas, requisitos y diseños emergen de la colaboración				X		
A intervalos regulares el equipo reflexiona sobre cómo mejorar						X
<b>Valores ágiles</b>						
Individuos e interacciones por sobre procesos y herramientas		X	X	X	X	X
Software Funcionando por sobre documentación extensiva						
Colaboración con el Cliente por sobre Relación Contractual			X	X	X	
Respuesta al cambio por sobre seguir un plan			X		X	X

Como se mostró a lo largo de este trabajo, la relación entre las Metodologías Ágiles y el Modelo SCARF están asociados en que dichas Metodologías incorporan prácticas, dentro de los proyectos, que son interpretadas por la mente como recompensas. El cerebro reacciona de manera positiva elevando los niveles de Dopamina y Oxitocina, posibilitando crear nuevas conexiones

que literalmente son nuevos enlaces disponibles para resolver viejos problemas y nuevos desafíos.

El modelo SCARF de David Rock nos brinda el marco teórico necesario para entender el contexto en el cual dichas metodologías tienen su mayor éxito. Estos proyectos requieren por un lado el uso intensivo del Córtex prefrontal para poder diseñar e implementar el software, y al mismo tiempo un sistema límbico libre de amenazas para una cooperación plena de los miembros del equipo. Esta mayor cooperación facilita un esquema de trabajo horizontal característico de las Metodologías Ágiles.

En la segunda parte del presente trabajo (Cuerpo Empírico), para poder cumplir con el objetivo de investigación N°4 se procedió a definir un Marco Epistemológico y un Diseño de investigación por medio de los cuales se desarrolló un trabajo de campo. En dicho trabajo se realizaron entrevistas semi-estructuradas a 3 miembros de la industria donde cada uno de ellos relató sus experiencias como parte de un equipo de desarrollo de software en diferentes empresas, contextos y roles. Dichas entrevistas fueron divididas en 6 secciones, En las primeras 5 secciones se indaga en los 5 conceptos (Status, Certeza, Autonomía, Relaciones, Justicia) de la experiencia social, relacionando cada uno de éstos con indicadores sensibles a las prácticas de las metodologías ágiles para poder determinar el grado de adopción de las mismas. Por último se les pidió que realizaran una evaluación de acuerdo a su percepción y experiencia sobre Productividad, Adaptación al cambio y Trabajo en equipo del proyecto.

Siguiendo con el Cuerpo Empírico del trabajo se encuentra la sección Análisis de las Entrevistas. En dicho apartado se evalúa el grado de acercamiento de cada proyecto a las prácticas planteadas por las Metodologías Ágiles, cumpliendo con el objetivo N°5, cuyo resumen se puede ver en el *Anexo II : Grado de Acercamiento del Proyecto al Modelo*. A su vez en el análisis se detalla en cada uno de los casos cómo impacta la implementación de las diferentes prácticas en el equipo de proyecto a través del modelo SCARF, es decir como la mente de los miembros del equipo puede interpretar cada

situación como amenaza o recompensa. De esta forma se alcanza el objetivo de investigación N°6.

Para finalizar en esta sección, en respuesta al objetivo N°7, se analizó la percepción de los entrevistados sobre el desempeño de sus respectivos proyectos en términos de productividad, adaptación al cambio y trabajo en equipo. La síntesis del mismo se puede observar en la *Tabla 6 Evaluación dada al desempeño de los proyectos en términos de Productividad, Adaptación al cambio y Trabajo en equipo por parte de los entrevistados.*

El objetivo de investigación N°8 como se plantea en la introducción es el indagar comparativamente en qué grado los proyectos estudiados se acercan al modelo teórico planteado para luego establecer el desempeño de los mismos. De esta forma, se desea saber, si en la medida que los proyectos de software se aproximen a las recomendaciones del modelo SCARF por medio de la implantación de metodologías ágiles tenderán a mejorar la productividad, adaptación al cambio y trabajo en equipo al interior de dichos proyectos

Para esto se puede analizar la Tabla 6: *Evaluación dada al desempeño de los proyectos en términos de Productividad, Adaptación al cambio y Trabajo en equipo por parte de los entrevistados.* En conjunto con la Tabla 7: *Indicadores del grado de implementación de Metodologías Ágiles promediados por Conceptos de la Experiencia Social (SCARF).*

**Tabla 6**

*Evaluación dada al desempeño de los proyectos en términos de Productividad, Adaptación al cambio y Trabajo en equipo por parte de los entrevistados.*

	AV	DN	FS
<b>Dimensiones</b>			
Productividad	3,5	1	3,5
Adaptación al cambio	5	2	3
Trabajo en equipo	5	1	5

**Tabla 7**

*Indicadores del grado de implementación de Metodologías Ágiles promediados por Conceptos de la Experiencia Social (SCARF)*

<b>Conceptos de la Experiencia Social</b>	AV	DN	FS
Status	5	1	3,5
Certeza	5	2	5
Autonomía	5	1,5	2,5
Relación	5	2	3,5
Justicia	4	1	4

Se puede apreciar que a medida que los conceptos se acercan a una experiencia de recompensa para el cerebro (puntaje de 5), que es en este contexto la implementación de las prácticas de las Metodologías Ágiles dentro de los proyectos, se ve un incremento en la percepción que tienen los entrevistados en torno a la productividad, adaptación al cambio y trabajo en equipo. Por otro lado en la medida que los entrevistados señalan prácticas relacionadas con metodologías Top-Down (puntaje de 1) dentro de los proyectos y por ende estableciendo una percepción de amenaza para el cerebro se puede observar como los indicadores disminuyen en las Dimensiones descriptas.

Por ende se puede decir que en los casos estudiados existe una relación incremental directa entre las recomendaciones del modelo SCARF implementadas por medio de las metodologías ágiles y la percepción de efectividad en productividad, adaptación al cambio y trabajo en equipo por parte de los entrevistados.

### **Palabras finales**

Me gustaría resaltar que mi intención sobre el presente trabajo es que no sea un fin en sí mismo. Sino un puntapié inicial para el entendimiento y mejora de los procesos de desarrollo de software y la gestión de empresas en general. Espero también que sirva para los especialistas en Neurociencias Cognitivas al señalar un ámbito en donde nuevas hipótesis y teorías puedan ser puestas a prueba.

## **ANEXO**

### **ANEXO I: ENTREVISTAS**

En este anexo se adjunta la desgravación completa de las entrevistas analizadas en el cuerpo empírico.

#### **ENTREVISTA: ANTONIO VILLARRUEL**

##### **BAKGROUD/PROYECTOS**

###### **1 - ¿Contame un poco de vos? ¿Experiencia Laboral? ¿Estudios?**

**AV:** Mi experiencia en Baufest, en los proyectos que estuve trabajando no eran netamente de software eran más proyectos de consultoría y de procesos. En los proyectos que estuve involucrado sea de implementación de herramientas o de software siempre tenían una pata fuerte de procesos. Siempre tenés que tener los procesos bien definidos más allá de las herramientas. Si la empresa no está bien organizada no hay herramienta que te lo resuelva.

Lo que tiene la parte de procesos es como... muchos clientes lo ven como que estás en el aire que venís con el librito de las buenas prácticas, pero si eran proyectos que la parte de procesos llevaban varios meses.

**LLL:** ¿En estos trabajaban en equipos?

**AV:** Si, trabajábamos en equipos con empresas multinacionales o con empresas nacionales de primera línea.

###### **2 - ¿Cómo clasificarías los proyectos en que trabajas (TopDown/BotomUp)?**

**LLL:** ¿Para gestionar los proyectos, qué metodologías utilizaban?

**AV:** Mayormente las definidas por el PMI, pero con la salvedad como yo lo veo que todo el mundo está probando Agile (Metodologías Ágiles de desarrollo de software). Se está tratando de hacer una gestión mixta de proyectos Ágiles-PMI, debido a que las organizaciones tienen una serie de requerimientos internos que las obligan a seguir con una metodología más estructurada por lo que no pueden abandonar este tipo de manejo de proyectos. También en muchas organizaciones hay personas más grandes que están acostumbradas a trabajar con el Project y el concepto de “te voy a cambiar el alcance porque el backlog te deja” no lo terminan de ver.

**LLL:** ¿Podés recordar un proyecto que hayas utilizado metodologías Ágiles? Hablemos sobre ese proyecto y tu experiencia en base a ese proyecto.

**AV:** Sí, hubo un proyecto en Tenaris que hacían esa combinación de Scrum para la gestión de proyecto y utilizaban herramientas del PMI para la planificación a alto nivel y el control.

## STATUS

### 3 - Roles/jerarquías dentro del proyecto.

**LLL:** ¿Cómo eran los roles/jerarquías dentro de ese proyecto?

**AV:** Volviendo a lo que decía recién tenías una gestión tradicional que se basaba en un Project y se interactuaba con varias áreas: Desarrollo, Arquitectura, Change Management, etc. El área de desarrollo se manejaba con Scrum (Metodologías Ágiles), en realidad con un Scrum que era más escalado. Lo que se conoce como Scrum de Scrum<sup>3</sup>. Utilizábamos la herramienta de Microsoft TFS el cual te permitía hacer el seguimiento de acuerdo a scrum en cuanto a las iteraciones, los Sprints, el Burn Down Chart. Por un lado tenías todo ese tipo de seguimientos y diagramas del tipo Ágil y por el otro tenías el Project.

**LLL:** OK, como estabas dentro de una empresa formal había que conservar la parte formal, pero para un mejor manejo interno del proyecto se aplicaba Scrum.

**AV:** Exacto.

**LLL:** ¿Qué roles había dentro de ese proyecto y vos que rol ocupabas?

**AV:** Dentro del área de desarrollo, vos tenías las Dailys, tenías a los Product Owners, tenías Scrum Master. Como PM era importante hacer el seguimiento. La mejor parte de la experiencia era que pudieron convivir las dos partes. Si bien yo tengo formación en Metodologías Ágiles, como PM por lo menos en las empresas donde trabajé es como que nunca te vas a poder alejar de una gestión tradicional, quizá en un entorno más chico puede ser. Me imagino en las empresas más dinámicas del mundo “Google”, bueno quizás. Pero no es el caso de una Telco ni un Banco.

---

<sup>3</sup> El Scrum es Scrum es una forma de escalar Scrum en donde existen varios equipos desarrollando en paralelo en donde cada Scrum Master participa de otro scrum encargado de coordinar y reportar del avance de cada equipo.



**LLL:** Entonces estaban los roles de PM, Scrum Master y Product Owner ¿El que gobernaba las prioridades del proyecto era el Product Owner?

**AV:** Sí, con la salvedad, que por la complejidad que tenía había distintos módulos funcionales que tenían como distintos sub-Products Ownes

**LLL:** El “proxy” del product owner.

**AV:** Exacto el proxy, que a la vez tenía otro subproxy que es el que podía seguir al equipo de desarrollo durante toda la semana.

#### **4 - Instancias de interacción entre los miembros del equipo.**

**LLL:** ¿Cómo era la interacción dentro de esos miembros del equipo?

**AV:** Como en los libros, había reuniones diarias donde se revisaban los pendientes, que problemas encontraban cada uno. Había problemas que había que escalarlos por problemas de definiciones y verlos con el negocio. Eso podía demorar un sprint entonces se repriorizaba hasta despejarse las dudas. Lo que sí requería de muchas reuniones, porque en un momento tenías seis equipos, entonces tenías 6 dailys todos los días, o sea 30 reuniones dailys por semana. Tenías reuniones con los Scrum Masters de Scrum de Scrum. Por otro lado la gestión tradicional, la reunión con los sponsors, las reuniones con el comité, etc. Había un tema de aprobación de presupuestos, el proyecto requería viajes y capacitaciones, eso tenía un costo para esos desvíos de costos había que pedir autorización al comité.

Una de las claves del éxito del proyecto, en relación a la interacción y comunicación fue que al principio en el kick off del proyecto se realizó una actividad que era entender cómo iba a ser cada iteración, o sea se presentó el esquema de trabajo para que entendieran la dinámica. Era un grupo enorme, había una reunión tipo auditorio donde se presentó el diagrama de cómo iba ser la célula de trabajo, cómo se pensó en ese momento que iba a hacer la agenda de reuniones y quienes tenían que participar en cada tipo de reunión. Se hizo una presentación y se difundió. Obviamente hubo variaciones después en la realidad de cómo fueron, pero ya por lo menos dejó en claro a los distintos equipos técnicos y equipos de negocios que iba a ser una integración fuerte.

Que quizás en un proyecto 100% metodología tradicional no era necesaria. El típico ejemplo que siempre se cuenta que en esquema tradicional el líder del

negocio aparece al principio del proyecto, define los requerimientos y al final para aprobar. Acá había palo y palo semana a semana había necesidad de estar con el analista del negocio. Pero yo creo que una vez aclarado el esquema de trabajo la gente se fue amoldando, inclusive las personas que entraron al proyecto con poca experiencia trabajando de esta forma.

## CERTEZA

### 5 – Definición de requerimientos y funcionalidades

**LLL** - ¿Quién y cómo se definen los requerimientos / funcionalidades?

**AV** - Si bien al inicio del proyecto los requisitos a alto nivel ya estaban definidos, cuando hubo que bajarlo se elaboró un product backlog que se repartía entre los diferentes equipos. Lo que se utilizó como ayuda para detallar los requerimientos es una técnica llamada *Design thinking*, en donde gente del negocio y gente del equipo de desarrollo se juntaron con unos coordinadores y con herramientas visuales ayudaron a definir las prioridades y funcionalidades de los requisitos.

Por ejemplo en un momento se habían pedido un montón de cosas como un *nice to have* de la aplicación. Se logró priorizar por medio del *design thinking* todos estos nuevos requerimientos y lograr una toma de conciencia del costo de estos nuevos requisitos y lograr un consenso hacia donde se iba. Por otro lado algunas personas del negocio no trabajaban en Buenos Aires, así que se aprovechaban esas jornadas para despejar dudas adicionales. Para mí esas jornadas fueron determinantes para que entiendan como era la metodología.

### 6 – Planificación de las tareas a realizar

**LLL** - ¿Cómo se produce la planificación de las tareas a realizar? ¿Cada cuánto?

**AV** - Los Sprints eran de 2 semanas, se revisaba con el negocio el *Product Owner* las nuevas funcionalidades y prioridades. Se iban bajando a detalle cada una de esas funcionalidades y el equipo evaluaba la probabilidad que el equipo pueda cumplir dentro del Sprint el desarrollo de esas funcionalidades.

### 7 – Gestión de los cambios en la planificación

**LLL**- ¿Cómo se manejan los cambios?

**AV** - Los cambios eran constantes. Algunos eran nuevas funcionalidades y otros eran como redefiniciones. Se negociaba con el *Product Owner* si el requerimiento entraba en este Sprint y ocupaba el lugar de otro o si quedaba para más adelante. Siempre intentando que no se vayan los tiempos. En un principio iban a parar a un Sprint. Para mí esto es una mala práctica. Al final de eso se tuvo que replanificar y no hubo tiempo para todo.

**LLL**: ¿El clásico sprint sin fecha?

**AV**: Claro, porque todo el mundo quiere agregar cosas y nadie quiere sacar.

## **AUTONOMÍA**

### **8 – Asignación de Tareas**

**LLL** - ¿Cómo se asignaban las tareas?

**AV** - Cada miembro del equipo se autogestionaba y se comprometía a terminar algo. El Scrum Master cumplía el rol de orientador a los miembros del equipo de menor seniority.

### **9 – Reporte de Status**

**LLL** - ¿A quién se reportaban las tareas?

**AV** - Al equipo, con herramienta. El product backlog y el sprint backlog estaba cargado en una herramienta que se asignaba a los distintos integrantes del equipo, y a medida que un miembro del equipo avanzaba en una tarea iba actualizando la herramienta, así que era un proceso en tiempo real. Yo podía ver online el estado de los distintos ítems de un sprint dado.

**LLL**: ¿También se realizaba el Daily Meeting?

**AV**: Si claro.

## **RELACIONES**

### **10 – Comunicaciones dentro del equipo**

**LLL** - ¿Cómo eran las comunicaciones dentro del equipo?

**AV** - Otra práctica que se apoyaba en las metodologías ágiles era que los equipos de trabajo estuvieran juntos. No era lo normal en este tipo de empresas porque está todo el mundo trabajando en distintas áreas o gerencias. Pero para este proyecto se gestionó reservar dos pisos del edificio para los equipos del proyecto y que todos trabajen juntos.

Esto facilitaba muchísimo las interacciones entre los involucrados del proyecto, las reuniones con los líderes cuando se necesitaba reunirse para hacer revisiones se organizaban en 2 minutos.

Por otro lado tenían desarrollado el concepto de *War Room* vos entrabas a un piso y tenías ahí las carteleras, carteleras digitales. Que eran televisores enormes que se pusieron a disposición del proyecto. En donde vos podías ver el *Burn Down Chart* y otras métricas que le daban al lugar de trabajo una atmosfera ágil que antes no existía. Vos ibas caminando por el piso y al ver una tele te dabas cuenta que parte del proyecto estaba complicado. Visualmente te dabas cuenta que varios módulos avanzaban con un ritmo y había uno que se estaba quedando, y decías:- Claro!! Problemas de infraestructura, problemas de definiciones, todo le está pegando al mismo módulo. Y eso lo veías cuando ibas caminando y ¡¡¡eso era comunicación!!! Al momento de sentarte en una reunión ya tenías un panorama del estado de las cosas.

Me gustaría volver a comentar que lo que hizo efectivo todo el proceso fue esa reunión, que se hizo al principio porque si no, podría a haber sido un caos.

## 11 – Relación con el Product Owner

**LLL** - ¿Y con los usuarios/Product Owner?

**AV**- La importancia de la relación con el *Product Owner*, más cuando este viene de otro enfoque un poco más tradicional es que desde las primeras iteraciones ya empieza a entender cuál es la realidad de un equipo de desarrollo. En principio no puede pedir cualquier cosa en cualquier momento, sabe que los requerimientos se planifican se estiman y hay una capacidad limitada. Entonces una vez que los *Products Owners*, entendían eso sabían que no podían pedir por pedir que todo lo que pidieran competía contra otra cosa. Vos me pedís esto pero yo te elimino otra funcionalidad de la iteración. Una vez que eso estuvo aceptado en un par de iteraciones el proceso fluía.

Creo que un error hubiera sido implementar esto a medias, es decir que el *Product Owner* participe de vez en cuando o permitir que priorice el Scrum Master.

**LLL**: ¿El Product Owner era parte del proyecto?

**AV**: Si, era parte. En muchos casos no podía estar físicamente pero delegaba el rol en un *Proxy*.

De todas formas cada 4 o 5 Sprints se realizaban reuniones especiales donde se le mostraba el avance a estos *Product Owners*.

## JUSTICIA

### 12 – Estimaciones esfuerzo de los requerimientos

LLL - ¿Cómo se establecían la magnitud y tiempos para la ejecución de las tareas?

AV - Como te comentaba antes primero se realizaban las estimaciones a alto nivel y después el equipo de desarrollo lo refinaba en cada *Sprint Plan Meeting*.

### 13 – Mejora continua

LLL - ¿Quién y cómo se definen los procesos de trabajo?

AV - El proceso ya estaba definido y era un Scrum bastante tradicional. No se modificaba mucho, pero sí había reuniones retrospectivas donde se planteaban mejoras externas, como la relación con las otras áreas.

## EXPERIENCIAS

### 14 – LLL : ¿Del 1 al 5 que puntaje le darías a la colaboración entre los miembros del equipo?¿Porque?

AV - 5, sencillamente el proyecto no hubiera funcionado si no hubiera habido el alto grado de colaboración entre los miembros del equipo como lo hubo.

### 15 – LLL: ¿Del 1 al 5 que puntaje le darías a la adaptación al cambio del equipo?¿Porque?

AV - 5, era en parte la naturaleza del proyecto.

### 16 – LLL: ¿Del 1 al 5 que puntaje le darías a la performance del equipo?¿Porque?

AV - 3,5 Nos pasó que en un punto del proyecto se rompe la cadena. Es decir el proceso se frena en algún punto por falta de definiciones. A diferencia de los proyectos tradicionales no hay una persona que pueda ver el todo y señalar por donde avanzar cuando ocurren estas situaciones.

Como resultado podés tener varios equipos del proyecto sin trabajar por semanas hasta que se resuelva la situación.

**ENTREVISTA: DIEGO NARVAEZ****BAKGROUD/PROYECTOS****1 - ¿Contame un poco de vos? ¿Experiencia Laboral? ¿Estudios?**

**DN:** Bueno yo estudié sistemas, nunca me recibí. Tengo mucha experiencia en proyectos de e-learning, más que nada en desarrollo de cursos e implementación de plataformas web. Con equipos conformados con diseñadores web, diseñadores didácticos coordinando el desarrollo de los cursos.

**2 - ¿Cómo clasificarías los proyectos en que trabajaste en tu última experiencia laboral (TopDown/BotomUp)?**

**DN:** Bueno era un proyecto de soporte de una aplicación de gestión documental. Una aplicación desarrollada por un proveedor que nos daba soporte a nosotros y nosotros al usuario. En parte era un pasamano del usuario al proveedor. Y en el medio tratábamos de resolver algunas cuestiones.

**LLL:** ¿Cuál era tu rol dentro del proyecto?

**DN:** Nunca estuvo claro el objetivo, o sea los usuarios cargaban incidencias y yo las tenía que resolver. Eso era, resolverlas yo o pedirle al proveedor que las resuelva.

**LLL:** ¿Específicamente cómo definirías la metodología que utilizaban?

**DN:** Era una estructura hiperpiramidal rígida, donde el jefe tomaba todas las decisiones y no escuchaba opiniones.

**LLL:** ¿Pero había una planificación previa o era solamente responder a los incidentes que podrían ir surgiendo?

**DN:** Por un lado si era responder a los incidentes, pero por otro lado había una planificación de un upgrade de funcionalidades (cambio de versión de la aplicación). Los plazos del Upgrade estaban puestos por el líder de proyecto.

**STATUS****3 - Roles/jerarquías dentro del proyecto.**

**LLL -** ¿Cómo eran los roles/jerarquías dentro del proyecto?

**DN -** Cómo mencioné anteriormente era muy piramidal.

#### **4 - Instancias de interacción entre los miembros del equipo.**

**LLL** - ¿Que instancias de interacción hay dentro de los miembros del equipo?

**DN** - No había una interacción directa. Nos reuníamos todos una vez por semana donde el jefe les hacía las preguntas a cada uno y cada uno le respondía dando indicaciones de qué hacer y cómo hacer.

#### **CERTEZA**

#### **5 – Definición de requerimientos y funcionalidades**

**LLL** - ¿Quién y cómo se definen los requerimientos / funcionalidades?

**DN** - Llegaba una incidencia del usuario, la analizaba y la ejecutaba bajo la supervisión de mi jefe.

**LLL** - ¿Había alguna instancia de planificación de las actividades diarias?

**DN** - No, los requerimientos se encolaban y se procesaban con la metodología FIFO (Fisrt In First Out).

#### **6 – Planificación de las tareas a realizar**

**LLL** - ¿Cómo se produce la planificación de las tareas a realizar? ¿Cada cuánto?

**DN** - Como te mencionaba antes, el jefe ponía los plazos y decía que hacer al mínimo detalle, lo que daba poco margen a que uno participe del diseño de la solución. Obviamente esto generaba contradicciones en los pedidos.

**LLL** - ¿Esta planificación se volcaba en alguna lista o aplicativo?

**DN** – SI, para los nuevos desarrollos se volcaba en un Project, pero yo como programador no tenía acceso al mismo. Solo podía ver el inventar ¿ que era la herramienta en donde llegaban las incidencias y que debían ser solucionadas al momento.

#### **7 – Gestión de los cambios en la planificación**

**LLL** - ¿Cómo se manejan los cambios?

**DN** - Sobre las nuevas funcionalidades no había cambios, una vez que se planteaba salía. Era mucho de integrar funcionalidades de la aplicación y básicamente hacerla andar. Uno andaba más preocupado por las incidencias del día a día que por las mejoras a futuro.

## **AUTONOMÍA**

### **8 – Asignación de Tareas**

**LLL** - ¿Cómo se asignaban las tareas?

**DN** - Las tareas ingresaban por la herramienta de gestión de incidentes (Inventar) y de forma automáticas eran asignadas. En mi caso por mi rol de soporte.

### **9 – Reporte de Status**

**LLL** - ¿A quién se reportaban las tareas?

**DN** - A mi jefe, todo el tiempo estaba preguntando como andaba el asunto. ¿Cuándo la incidencia estaba resuelta? Yo cerraba el ticket y el usuario se enteraba a través del ticket cerrado.

## **RELACIONES**

### **10 – Comunicaciones dentro del equipo**

**LLL** - ¿Cómo eran las comunicaciones dentro del equipo?

**DN** - No había, las otras personas estaban con aplicaciones diferentes y no era necesaria una comunicación a nivel laboral.

### **11 – Relación con el Product Owner**

**LLL** - ¿Y con los usuarios/Product Owner?

**DN** - Por medio de la herramienta (Inventar) y por teléfono también si había alguna consulta sobre incidentes particulares.

**LLL** - ¿Había reuniones programadas con anterioridad?

**DN** - No, solo se trataban temas por incidentes particulares.

## **JUSTICIA**

### **12 – Estimaciones esfuerzo de los requerimientos**

**LLL** - ¿Cómo se establecen la magnitud tiempos para la ejecución de las tareas?

**DN** - Había un SLA que cumplir, según la criticidad de la tarea.

**LLL** - ¿Y para los nuevos requerimientos?



**DN** - También tenías plazos predefinidos de un máximo de 3 meses. Pero mi jefe predefinía el tiempo estimado de implementación de cada uno de ellos.

### **13 – Mejora continua**

LLL – ¿Quién y cómo se definen los procesos de trabajo?

DN – En un principio los procesos los definía el jefe. Después en la última etapa se comenzó a implementar una metodología, que hacía enfoque en la documentación. El cumplimiento de la misma era relativa a las necesidades del negocio. Lo importante de la misma era mantener una documentación antes de implementar algún cambio en el sistema.

### **EXPERIENCIAS**

#### **14 - ¿Del 1 al 5 que puntaje le darías a la colaboración entre los miembros del equipo? ¿Porque?**

1, no había colaboración eran todos silos independientes. Y el jefe tampoco colaboraba con ninguno de los proyectos. Sino que controlaba.

#### **15 - ¿Del 1 al 5 que puntaje le darías a la adaptación al cambio del equipo? ¿Porque?**

2, Bueno no hubo muchos casos. Hubo una posibilidad de migrar a una nueva plataforma para la gestión documental. En un principio me habían asignado el proyecto. Pero luego fue tomado por el jefe y no hizo nada dejándolo morir. Las nuevas versiones de la aplicación en la mayoría de casos también las iban planchando.

#### **16 - ¿Del 1 al 5 que puntaje le darías a la performance del equipo? ¿Porque?**

1, No había colaboración, había cosas muy parecidas que se duplicaba el trabajo. No se compartía el know how.

### **ENTREVISTA: FERNANDO SCASSERRA**

#### **BAKGROUD/PROYECTOS**

#### **1 - ¿Contame un poco de vos? ¿Experiencia Laboral? ¿Estudios?**

**FS:** Yo laburo en Mercado Libre (ML) hace 10 años. Actualmente trabajo como Senior Manager del Área de Desarrollo. Hace un mes tengo el área de QA (Quality Assurance) a cargo, y bueno ahí tengo un desafío, mejorar la performance de la gerencia. Estoy también a cargo de la API de ML, donde pasa todo el tráfico y donde está montada la aplicación. Es un laburo de Back-end, pero bastante crítico. En mi proyecto las personas son más bien ingenieros bastante seniors, que tienen un esquema de desarrollo Dev-Ops que corre todo en el Cloud.

Este puesto lo tengo hace un año antes de eso estuve como Manager que era algo similar pero con un equipo más chico.

También en la empresa soy el Focal Point de las entrevistas técnicas y de la parte de ISO, cuando hay que certificar. En el laburo tenemos que usar metodologías ágiles. No es que sea una obligación pero la historia nos llevó a ver que eso era práctico. Cada equipo lo puede implementar de la forma que crea correcto. Haciendo Scrum como dicen los libros o haciendo una adaptación de lo que consideren necesario mientras que sea ágil está bien.

## **2 - ¿Cómo clasificarías los proyectos en que trabajaste en tu última experiencia laboral (TopDown/BottomUp)?**

**FS:** A mi entender mi proyecto no es ninguna de las dos cosas. ¿Qué sería TopDown y BottomUp?

**LLL:** TopDown, es que hay un PM que planifica y que tiene robotitos que hacen lo que dice.

**FS:** BottomUp, sería que entre todo el mundo se define todo?

**LLL:** Sí, más que nada hace hincapié en un ambiente más colaborativo.

**FS:** Es una cosa intermedia una mezcla de los dos. No tenemos un PM. A ver, básicamente mis sponsors son gerentes de otros proyectos que vienen a pedir a mi proyecto una funcionalidad nueva. Ej. viene la persona encargada de la Homepage de ML y me dice, "Quiero que la api me dé un listado de los productos que se están por vencer ahora".

**LLL:** Está bien, ¿Eso es lo que sería para vos un Product Owner?

**FS:** Sí, le decimos Sponsor. Pero en realidad tenemos muchos Sponsors, que son todos los proyectos de FrontEnd de ML.

## STATUS

### 3 - Roles/jerarquías dentro del proyecto.

LLL - ¿Cómo son los roles/jerarquías dentro del proyecto?

El proyecto tiene muchas partes, en realidad está dividido en 4 subproyectos, dentro de los cuales hay desarrolladores que se dividen en 4 niveles: Junior , Ingeniero, Semi-senior y Senior. Todos hacen todo pero con distinto impacto y/o tamaño de lo que hacen.

Todo significa: Estar en una reunión con el Sponsors intentando entender que quiere, después hacer una mini documentación, después lo programa, después lo testea y finalmente lo sube a producción. Si es un Junior hará un ABM sencillo y si es un Senior puede hacer un sistema mucho más grande con mucho más impacto. Eso es coordinado por un Líder de Proyecto, que es un desarrollador senior que además coordina a otros desarrolladores. Con lo cual depende de cuanta gente programando tenga, tendrá un 50% del tiempo programando y un 50% del tiempo ayudando al resto, y si es un equipo mucho más grande está casi todo el tiempo ayudando a los demás , “Seguí con esto, seguí con lo otro”.

### 4 - Instancias de interacción entre los miembros del equipo.

LLL - ¿Que instancias de interacción hay dentro de los miembros del equipo?

**FS:** Por lo general el líder de proyecto se junta una vez por semana para ver cómo está cada programador y si tiene algún problema, y reasignar prioridades en caso que sea necesario. Y después está el manager que es un líder de proyecto mucho más senior. Yo como mánager me reúno con todos los líderes y desarrolladores (20 personas en total) los lunes más o menos una hora y media. Cada equipo presenta lo que hizo la semana pasada y con lo que va a seguir.

La idea es analizar cómo está el equipo con respecto al plan “seteado” al principio del Quarter.

Si se detecta un atraso significativo se les avisa a los Sponsors.

Cada Sub-proyecto se reúne una vez por semana. A veces estoy en esas reuniones y durante el día están sentados en la misma mesa así que si hay algún tema lo charlan ahí.

## CERTEZA

### 5 – Definición de requerimientos y funcionalidades

**LLL** - ¿Quién y cómo se definen los requerimientos / funcionalidades?

**FE** - Los Sponsors del proyecto son los que definen los requerimientos. Yo me junto con los Sponsors y definimos que quiere cada uno, participan de esta reunión el Gerente y líder del equipo. Otro input que hay es la reunión que tenemos con los líderes y el equipo de calidad que reportan bugs y vulnerabilidades también una vez por semana, que incrementan la cantidad de tareas en el Backlog.

Resumiendo los requerimientos pueden ser cosas que nos pidió un Sponsor, que es una necesidad del negocio. Un ejemplo: Viene el Gerente de Marketing y dice: “Queremos hacer una campaña de cupones”, se lo dice al líder encargado de Front-End que es al que le va terminar impactando, ellos definen lo que van a necesitar y eso nos impacta en un requerimiento en la API de nosotros. Por otro lado hay requerimientos que son bugs que se detectan porque llamó un usuario quejándose de un error en una funcionalidad de ML. El Call Center toma el pedido. Y lo reporta a otra área que es la encargada de derivar el problema al equipo de desarrollo que corresponda.

Otra fuente puede ser seguridad informática que detecta una vulnerabilidad en el sistema. Y por último hay requisitos autogenerados por nosotros. Que pueden prevenir de alertas de performance en nuestras aplicaciones donde nosotros generamos un requerimiento para la mejora de los mismos.

En resumen todo termina como una tarea en el Backlog de alguno de los 4 proyectos.

### 6 – Planificación de las tareas a realizar

**LLL** - ¿Cómo se produce la planificación de las tareas a realizar? ¿Cada cuánto?

**FE** - Tenemos objetivos por Quarters, con lo cual el primero de enero planificamos todo lo que vamos a hacer hasta el primero de abril. En la planificación por ejemplo nos caen como 60 requerimientos en la reunión con los Sponsors y luego lo vemos con el equipo. Ahí se hace una estimación más fina, y después se les da un feedback a los sponsors. Así que es una mezcla de los dos. Una vez que está el Ok, comienzan los ciclos iterativos mucho más

cortos. La particularidad que tiene mi proyecto es que nuestros sponsors son a su vez otras áreas técnicas que tienen sponsors del negocio, por lo cual los requerimientos nos llegan un poco tarde. Lo que provoca que al principio del Quarter solo tenemos el 30% de las cosas y a medida que va pasando el loop van cayendo el resto de los pedidos. Que son a demanda, no hay ninguna reunión específica para esto solo tienen que completar un documento con el requerimiento.

## **7 – Gestión de los cambios en la planificación**

**LLL** - ¿Cómo se manejan los cambios?

**FE** - El proceso anterior genera que haya una replanificación constante de los requerimientos, teniendo que priorizar entre los diferentes sponsors.

## **AUTONOMÍA**

### **8 – Asignación de Tareas**

**LLL** - ¿Cómo se asignan las tareas?

**FS**- Cuando llega un requerimiento nuevo, el líder de proyecto le solicita al sponsor que defina el requerimiento en un documento donde se detalla la funcionalidad requerida, y la importancia que tiene para el negocio. Analiza la prioridad y decide si el desarrollo entra para la próxima semana, el próximo mes, o Quarter llegado el caso.

**LLL** - ¿El líder de proyecto lo asigna?

**FE** - Sí.

**LLL** - ¿No hay auto asignación de tareas?

**FS**- No, no creo en la autoasignación de tareas. Bueno depende... Hay autoasignación de tareas pero de cosas sencillitas. No sé, la típica es que se va de vacaciones el líder de proyecto y dejó todo el Backlog armado con las prioridades y bueno en esos casos, si terminan con las tareas asignadas siguen con otras cosas del Backlog.

### **9 – Reporte de Status**

**LLL** - ¿A quién se reportan las tareas?

**FS** – Los integrantes del proyecto cuando finalizan las tareas le dicen al líder que ya están listas, y tiene la aprobación del Sponsor. Luego el líder de

proyecto las pasa a producción y el Sponsor da el ok final. Yo me entero en la reunión semanal.

## RELACIONES

### 10 – Comunicaciones dentro del equipo

**LLL** - ¿Cómo son las comunicaciones dentro del equipo?

Bueno las reuniones que por lo general son reuniones de status y con qué seguimos. También tenemos una cuenta del mail del equipo en donde enviamos los pedidos y se comunican a todos. Después tenemos un grupo en Telegram y nos escribimos por ahí. Cada vez que hay que hacer un deploy se avisa por la herramienta y también se utiliza de guardia por si se rompió algo. Es decir si alguien me escribe por esa herramienta yo lo leo, porque sé que es importante. La comunicación de los requerimientos pueden llegar por varias vías: Chat, mail, pasillo. Pero siempre le pedimos que me llene el documento es como para que piensen un poco el requerimiento (Que aporta al negocio, cuan urgente es, para cuando lo necesitan).

### 11 – Relación con el Product Owner

**LLL** - ¿Y con los usuarios/Product Owner?

**FS:** Como ya mencioné antes de que comience el Quarter le enviamos a los Sponsors un mail diciendo por favor que todos los proyectos que requieran una modificación en la API nos envíen los requerimientos.

En el mejor de los casos el manager muy prolijo te dice: “Che miren vamos a necesitar estas 10 cosas en este Quarter.”

**LLL** - No es un miembro del equipo del Sponsor, pero ¿está accesible?

**FS:** - Sí, le mandás un mail y coordinás una reunión sin problemas para aclarar cualquier tema. Al principio del mes le enviamos un mail con un aviso si vamos a poder iniciar o no con lo programado para el mes. Y después cuando comenzamos a trabajar sobre el requerimiento ahí te juntas más o le preguntas más cosas. Antes de subirlo a producción lo subimos a un ambiente donde el Sponsor que lo pidió lo puede probar. Y si da el ok lo terminamos subiendo. Muchas veces se hace a la noche.

## JUSTICIA

## **12 – Estimaciones esfuerzo de los requerimientos**

**LLL** - ¿Cómo se establecen la magnitud y tiempos para la ejecución de las tareas?

**FS:** El líder de proyecto dice lo que puede llegar a demorar en una primera instancia con el Sponsor. Luego lo chequea con el developer que lo va realizar pactando una estimación más refinada.

Como comentario adicional, me gustaría agregarle que el éxito de los proyectos está en que tenemos personas muy capaces. Si no fuera así sería imposible llevar a cabo los proyectos. Y tendríamos que poner mucha más metodología y muchos más controles.

## **13 – Mejora continua**

**LLL** - ¿Quién y cómo se definen los procesos de trabajo?

**FS** - Lo hacen entre el líder del proyecto y el Manager.

**LLL** - ¿Y hay algún tipo de revisión de esos procesos?

**FS** – Si, tenemos ese tipo de charlas donde se proponen nuevas mejoras que son consensuadas entre el líder de proyecto y el manager, pero no se hacen con todo el equipo.

## **EXPERIENCIAS**

**14 - ¿Del 1 al 5 que puntaje le darías a la colaboración entre los miembros del equipo? ¿Porque?**

5, son súper colaborativos. Son todos buena onda y están todos comprometidos con el proyecto, a todos les gusta lo que hacen. La comunicación es muy buena entre ellos, se llevan super bien con su líder. Si se piden ayuda se dan ayuda. Yo creo que la clave es que hay muy buena onda en el proyecto, están re cómodos. El líder es uno más de ellos.

**15 - ¿Del 1 al 5 que puntaje le darías a la adaptación al cambio del equipo? ¿Porque?**

3, porque a la gente le molesta mucho que le cambien las cosas. Lo que más le impacta es cuando los miembros del equipo vienen trabajando con una funcionalidad que ya está lista y le dicen que eso ya no es más importante

porque lo dice el Sponsor, eso les molesta mucho. Eso pasa todo el tiempo por la dinámica del negocio de mercado libre.

### 16 - ¿Del 1 al 5 que puntaje le darías a la performance del equipo? ¿Porque?

No creo que sea un 5 porque siempre hay un problema. Entre un 3 y un 4... Bueno le daría un 3,5. Porque el equipo tiene muy buena performance llegan a cumplir con los objetivos, pero no es la forma más feliz que me gustaría que los hayan cumplido. Por lo general cumplen porque el líder se puso algo al hombro que si no, no se llegaba. Pero de todas formas cumplimos más de lo que se nos pide y los Sponsor están contentos con el proyecto.

### ANEXO II: GRADO DE ACERCAMIENTO DEL PROYECTO AL MODELO

Entrevistas	AV	DN	FS
<b>STATUS</b>			
Roles/jerarquías dentro del proyecto.	5	1	3
Instancias de interacción entre los miembros del equipo.	5	1	4
<b>CERTEZA</b>			
Definición de requerimientos y funcionalidades	5	2	5
Planificación de las tareas a realizar	5	2	5
Gestión de los cambios en la planificación	5	2	3
<b>AUTONOMÍA</b>			
Asignación de Tareas	5	2	2
Reporte de Status	5	1	3
<b>RELACIÓN</b>			
Comunicaciones dentro del equipo	5	1	4
Relación con el Product Owner	5	3	3
<b>JUSTICIA</b>			
Estimaciones esfuerzo de los requerimientos	5	1	5
Mejora continua	3	1	3



## **BIBLIOGRAFÍA**

- Amthor, F. (2012). *Neurociencia for Dummies*. Ontario: Wiley.
- B. W. Boehm, T. D. (1988). A spiral model of software development and enhancement. *Computer (Volume:21 , Issue: 5 )*, 61 - 72.
- Bear, M. F., Connors, B. W., & Paradiso, M. A. (2016). *Neurociencia Exploring the brain*. Philadelphia: Wolters Kluwer.
- Beck, K. (1999). *Extreme Programming Explained*. Boston: Addison-Wesley.
- Bhargav, R. (15 de 06 de 2015). *www.simplilearn.com*. Recuperado el 14 de 03 de 2017, de *www.simplilearn.com*: <https://www.simplilearn.com/history-evolution-of-pmp-certification-article>
- Cacioppo, J. T. (2008). *Loneliness: Human Nature and the Need for Social Connection*. New York: W. W. Norton and Company.
- Doidge, N. (2007). *The brain that Changes Itself*. New York: Penguin Group.
- Dunbar, G. (20 de 01 de 2016). *linkedin.com*. Recuperado el 15 de 03 de 2017, de *linkedin.com*: <https://www.linkedin.com/pulse/project-management-failures-standish-chaos-report-2015-dunbar>
- Epstein, R. (2007). *The Case Against Adolescence: Rediscovering the Adult in Every Teen*. Fresno: Quill Driver Books.
- Flyvbjerg, B. (2004). Cinco malentendidos acerca de la investigación mediante los estudios de caso. *REIS*, 33-62.
- Giddens, A. (2007). *Sociología*. Madrid: Alianza editorial.
- Gilb, T. (1977). *Software metrics*. Cambridge: Winthrop Publishers.
- Glazer, H., Dalton, J., Anderson, D., Konrad, M., & Shrum, S. (2008). *CMMI® or Agile: Why Not Embrace Both!* Hanscom AFB: SEI Administrative Agent.
- Hawkins, J., & Blakeslee, S. (2004). *On Intelligence*. New York: Times Books.

- Iacoboni, M. I.-S. (2005). Grasping the intentions of others with one's own mirror neuron system. *PloS Biology*3, no. 3, 79.
- Kent Beck, y. O. (2001). *agilemanifesto.org*. Recuperado el 15 de 03 de 2017, de *agilemanifesto.org*: <http://agilemanifesto.org/iso/es/manifesto.html>
- Lieberman, M. (2007). Social cognitive neuroscience: a review of core processes. *Annual Review of Psychology*, 259–289.
- Lorenzo y Losada, L. (2009). CMMI y Metodologías Ágiles – Un Caso Práctico (tesis de postgrado). Buenos Aires, Argentina: Universidad Católica Argentina.
- Maier, S. F. (1982). Controllability, coping behavior, and stress-induced analgesia in the rat. *Pain*12, 47–56.
- Pinker, S. (1997). *How the Mind Works*. New York: W. W. Norton and Company.
- Rock, D. (2008). SCARF: a brain-based model for collaborating with and influencing others. *NeuroLeadership Journal*.
- Rock, D. (2009). *YOUR BRAIN AT WORK*. New York: HarperCollins ebooks.
- Royce, W. (1970). Managing the Development of Large Software Systems. *Proceedings of IEEE WESCON 26 (August)*, 1-9.
- Schwaber, K. (2004). *Agile Project Management*. Redmond: Microsoft Press.
- Tabibnia, G. a. (2007). Fairness and cooperation are rewarding: Evidence from social cognitive neuroscience. *Annals of the New York Academy of Sciences* ([http://www.scn.ucla.edu/pdf/Tabibnia%20\(2007\).pdf](http://www.scn.ucla.edu/pdf/Tabibnia%20(2007).pdf)), 90-101.
- Tabibnia, G. A. (2008). The sunny side of fairness: Preference for fairness activates reward circuitry (and disregarding unfairness activates self-control circuitry). *Psychological Science*19, no. 4, 339–47.
- The Standish Group. (1 de 1 de 1995). *www.projectsart.co.uk*. Recuperado el 13 de 03 de 2017, de *www.projectsart.co.uk*: <https://www.projectsart.co.uk/white-papers/chaos-report.pdf>